

TOWARDS EFFICIENT 3D RECONSTRUCTIONS FROM
HIGH-RESOLUTION SATELLITE IMAGERY

Ke Wang

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2018

Approved by:

Jan-Michael Frahm

Enrique Dunn

Alexander C. Berg

Dinesh Manocha

Marc Niethammer

© 2018
Ke Wang
ALL RIGHTS RESERVED

ABSTRACT

Ke Wang: Towards Efficient 3D Reconstructions from High-Resolution Satellite Imagery
(Under the direction of Jan-Michael Frahm and Enrique Dunn)

Recent years have witnessed the rapid growth of commercial satellite imagery. Compared with other imaging products, such as aerial or streetview imagery, modern satellite images are captured at high resolution and with multiple spectral bands, thus provide unique viewing angles, global coverage, and frequent updates of the Earth surfaces. With automated processing and intelligent analysis algorithms, satellite images can enable global-scale 3D modeling applications.

This dissertation explores computer vision algorithms to reconstruct 3D models from satellite images at different levels: geometric, semantic, and parametric reconstructions. However, reconstructing satellite imagery is particularly challenging for the following reasons: 1) Satellite images typically contain an enormous amount of raw pixels. Efficient algorithms are needed to minimize the substantial computational burden. 2) The ground sampling distances of satellite images are comparatively low. Visual entities, such as buildings, appear visually small and cluttered, thus posing difficulties for 3D modeling. 3) Satellite images usually have complex camera models and inaccurate vendor-provided camera calibrations. Rational polynomial coefficients (RPC) camera models, although widely used, need to be appropriately handled to ensure high-quality reconstructions.

To obtain geometric reconstructions efficiently, we propose an edge-aware interpolation-based algorithm to obtain 3D point clouds from satellite image pairs. Initial 2D pixel matches are first established and triangulated to compensate the RPC calibration errors. Noisy dense correspondences can then be estimated by interpolating the inlier matches in an edge-aware manner. After refining the correspondence map with a fast bilateral solver, we can obtain dense 3D point clouds via triangulation.

Pixel-wise semantic classification results for satellite images are usually noisy due to the negligence of spatial neighborhood information. Thus, we propose to aggregate multiple corresponding observations of the same 3D point to obtain high-quality semantic models. Instead of just leveraging geometric reconstructions to provide such correspondences, we formulate geometric modeling and semantic reasoning in a joint Markov Random Field (MRF) model. Our experiments show that both tasks can benefit from the joint inference.

Finally, we propose a novel deep learning based approach to perform single-view parametric reconstructions from satellite imagery. By parametrizing buildings as 3D cuboids, our method simultaneously localizes building instances visible in the image and estimates their corresponding cuboid models. Aerial LiDAR and vectorized GIS maps are utilized as supervision. Our network upsamples CNN features to detect small but cluttered building instances. In addition, we estimate building contours through a separate fully convolutional network to avoid overlapping building cuboids.

To my family

ACKNOWLEDGEMENTS

This dissertation would not be possible without my advisors: Prof Jan-Michael Frahm and Prof Enrique Dunn. You are not only just great academic advisors, but also great life mentors. I thank you for your patience and kind support to lead me through the hard but fruitful PhD career.

I also want to express my gratitude to my dissertation committee members: Prof Alexander C. Berg, Prof Dinesh Manocha, and Prof Marc Niethammer. I would like to thank you for your support and help in writing and defending this dissertation.

Special thanks to all lab mates from the computer vision research group at UNC. Thank you for all the meaningful and interesting conversations. Thank you for all the help, and letting me take that windowed office.

Special thanks to Tianxiang Gao and his family. Thank you for sharing with me your happiness, and walk me through my gloomy days.

Special thanks to Lu Chen and all other friends from the Computer Science Department. You made my life in graduate school unforgettable.

Special thanks to my parents. Thank you for the freedom you gave me to study abroad for years. Thank you for the unconditional understanding and support.

Especially, I would like to express my gratitude for my wife Lei Miao. You made all my efforts worthwhile.

壬辰年秋至戊戌年春，教堂山下，六载光阴，转瞬即逝。

博士这几年，学会了独在异乡为异客时当如何面对生活寂寥，通晓了负篋曳屣挑灯夜读时应如何看淡世事喧嚣，领悟了实验挫败意冷心灰时又当如何坦然一笑。读博这些年，是人生中十分特别的一段经历。这段人生所积淀的，不仅是安身立命的技能与知识，更是如何踏实做人认真做事的处世哲学。博士阶段的学业结束，也标志着漫长的求学阶段即将

画上一个短暂的休止符。这些年来所经历和感受的，痛苦和彷徨也好，心酸与喜悦也罢，此处挂一漏万，难以表全。惟愿在以后的人生里，能初心不忘，努力前行。

最后，铁甲依然在

王珂

戊戌年春

于教堂山

TABLE OF CONTENTS

LIST OF TABLES	xii
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS	xvi
CHAPTER 1: INTRODUCTION	1
1.1 Dissertation Statement	5
1.2 Outline of Innovations	5
CHAPTER 2: RELATED WORK	7
2.1 Data Sources for Urban Reconstruction	9
2.1.1 Ground-Level Imagery	9
2.1.2 Aerial Imagery	10
2.1.3 Satellite Imagery	11
2.1.4 LiDAR	12
2.2 Sparse Reconstruction	14
2.3 Dense Reconstruction	15
2.3.1 Multi-View Stereo	15
2.3.2 Volumetric reconstruction	20
2.3.3 Monocular reconstructions	21
2.4 Data-driven Methods for 3D Reconstructions	21
2.4.1 Deep Learning	22
2.4.2 Deep Learning for 3D Vision	26
2.4.3 Datasets for Learning	29

2.4.4	Learning SfM/SLAM	30
2.4.5	Learning Feature Matching	34
2.4.6	Learning Stereo	35
2.4.7	Learning Optical Flow	37
2.4.8	Learning Surface Normals	37
2.4.9	Learning Monocular View Tasks	38
2.4.10	Learning Multi-view Tasks	38
CHAPTER 3: SATELLITE IMAGING CAMERA MODEL		40
3.1	Spatial Reference Systems	40
3.1.1	Geodetic Datum	41
3.2	Satellite Images	42
3.3	RPC Camera Models	44
3.3.1	Forward and Inverse Model	45
3.3.2	Triangulation	47
3.3.3	Bias Compensation	48
3.4	Radiometric Calibration	50
CHAPTER 4: GEOMETRIC RECONSTRUCTION VIA EDGE-AWARE INTER- POLATION		51
4.1	Introduction	51
4.2	Geo-guided Sparse Feature Matching	53
4.3	Edge-Aware Interpolation	54
4.4	Confidence Refinement	56
4.5	Bilateral Refinement	58
4.6	Point Cloud Alignment	59
4.7	Experiments	60
4.7.1	Evaluation	62
4.7.2	Runtime	62

4.7.3	Ablation Study	63
4.8	Discussion	65
4.8.1	Feature Comparison	65
4.8.2	Occlusion Handling	65
4.8.3	Future Work.....	65
CHAPTER 5: JOINT GEOMETRIC AND SEMANTIC RECONSTRUCTION		66
5.1	Introduction.....	66
5.2	Local 3D Plane Formulation	67
5.3	Land Use Classification	70
5.4	Inference	70
5.5	Experiments	71
5.5.1	Implementation.....	71
5.5.2	Ground Level Stereo Experiments	71
5.5.3	Satellite Images Experiments.....	74
5.6	Discussion	78
CHAPTER 6: PARAMETRIC CUBOID MODEL RECONSTRUCTION		80
6.1	Introduction.....	80
6.2	Cuboid Models.....	82
6.3	Cuboid Model Fitting as 3D Object Detection	83
6.3.1	Network Architecture	84
6.3.2	Vector Map Pre-processing.....	87
6.3.3	Training.....	88
6.4	Overlapping Refinement.....	91
6.4.1	Network Architecture	92
6.4.2	Training.....	92
6.5	Experiments	93

6.5.1	Dataset.....	93
6.5.2	Evaluation	94
6.6	Discussion	97
6.6.1	Input data	97
6.6.2	Radiometric correction.....	97
6.6.3	Feature map resolution.....	98
6.6.4	Instance-aware segmentation.....	98
6.6.5	Limitations of Cuboids	98
6.6.6	Future work	99
CHAPTER 7: DISCUSSION		100
7.1	Conclusion	100
7.2	Future Work	101
7.2.1	Efficient and Robust RPC Solvers	101
7.2.2	Multi-Modal 3D Reconstructions	102
7.2.3	Multi-Modal Machine Learning	102
7.2.4	Machine Learning with Noisy Supervision	103
7.2.5	Systematic Design and Engineering	104
REFERENCES		105

LIST OF TABLES

Table 2.1 – Comparision for 3D reconstruction data modalities.	9
Table 2.2 – Activation functions	27
Table 3.1 – Earth reference ellipsoids	42
Table 3.2 – Geo-location in different Geodetic Datum.....	42
Table 3.3 – Configuration comparisons of recently launched commercial imaging satellites. .	43
Table 3.4 – RPC bias compensation model comparison.....	50
Table 4.1 – Quantitative Evaluation of Satellite Image Stereo.....	60
Table 4.2 – Runtime statistics for each stage of the stereo pipeline.	63
Table 4.3 – Ablation study for individual stages of the stereo pipeline.	64
Table 4.4 – Local image feature comparison for geo-guided feature matching	64
Table 5.1 – Stereo accuracy evaluation on Leuven dataset.	72
Table 5.2 – Stereo accuracy evaluation on the KITTI dataset	73
Table 5.3 – Stereo semantic classification accuracy.....	73
Table 5.4 – Kitti semantic classification accuracy evaluation.....	74
Table 5.5 – Satellite reconstruction examples.	75
Table 6.1 – Base feature network architecture.....	85
Table 6.2 – Detection network detailed architecture.	86
Table 6.3 – Signed-distance map network detailed architecture.....	93
Table 6.4 – Detailed dataset statistics.	94
Table 6.5 – Quantitative evaluations of cuboid parametric reconstructions.	95
Table 6.6 – Visualization of single view parametric reconstructions	96

Table 6.7 – Comparison of single-view reconstructions.	97
Table 6.8 – Evaluation of feature map resolution.	98

LIST OF FIGURES

Figure 1.1 – Early satellite images. Source: Wikipedia.	1
Figure 2.1 – Relationship between images, geometry, and photometry.	7
Figure 2.2 – Typical 3D reconstruction pipeline.	8
Figure 4.1 – Stereo pipeline overview	51
Figure 4.2 – Illustration for geo-guided feature matches	52
Figure 4.3 – Example of geo-guided feature matching	54
Figure 4.4 – Illustration of edge-aware interpolation	56
Figure 4.5 – Refinement of flow field using bilateral solver.	59
Figure 4.6 – Height map visualization	61
Figure 5.1 – Grid representation visualization.	68
Figure 5.2 – Qualitative illustration of PMBP convergence.	70
Figure 5.3 – Illustration of PMBP inference process.	71
Figure 5.4 – Joint reconstruction visualization on the KITTI dataset	72
Figure 5.5 – Joint reconstruction visualization on satellite images	75
Figure 5.6 – Comparison of semantic labels	76
Figure 5.7 – Zoom-in comparison of height maps reconstructed from satellite images	78
Figure 6.1 – Overview of proposed parametric reconstruction pipeline.	81
Figure 6.2 – Difference between satellite imagery and ordinal images.	82
Figure 6.3 – 3D cuboid parameterization of buildings	84
Figure 6.4 – Proposed network architectures	85
Figure 6.5 – Architecture for 3D detection network.	87

Figure 6.6 – Default cuboid design.....	90
Figure 6.7 – Signed distance field.....	92
Figure 6.8 – Architecture for signed-distance map prediction network.....	92
Figure 6.9 – Comparison with instance-aware segmentation methods	99

LIST OF ABBREVIATIONS

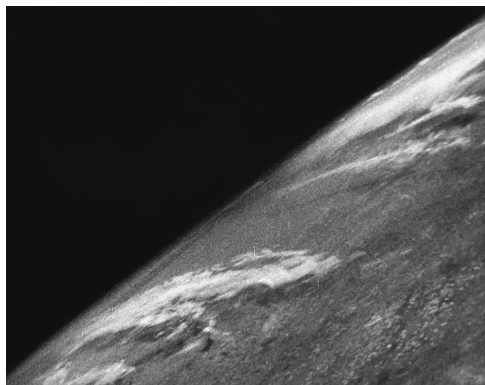
CNN	Convolutional Neural Networks
GCP	Ground control points
IMU	Inertial measurement unit
LSTM	Long-Short Term Memory
LiDAR	Light Detection And Ranging
MRF	Markov Random Field
MVS	Multiple-View Stereo
RFM	Rational Function Model
RPC	Rational Polynomial Coefficients
SfM	Structure-from-Motion

CHAPTER 1: INTRODUCTION

Human's curiosity of the space and the Earth never ends. Numerous spacecrafts have been sent to the space to explore the unknown. The U.S. took the first image of the Earth (see Figure 1.1a) from the space on a sub-orbital V-2 rocket on October 24, 1946. On August 14, 1959, the U.S. Explorer 6 satellite shot the first orbital satellite photograph (Figure 1.1b). In 1972, the Apollo 17 spacecraft crew took the famous Blue Marble photo of the Earth (Figure 1.1c). These are early attempts to collect visual observation data of the Earth from the space.

Also in 1972, the U.S. launched the Landsat program (NASA, 1972) to systemically acquire imagery of the Earth from space. The Landsat program marks the start of the modern era of Earth observation and surveying using satellite imagery.

Since then, imaging satellites have thrived to enable a comprehensive range of applications, covering military, government, and commercial scenarios. State-of-the-art commercial image satellites are capable of taking images with ground sampling distances up to 30 centimeters per pixel. Modern satellite images can cover as many as 16 spectral channels, and pixels are captured



(a) First image from space



(b) First satellite image



(c) The Blue Marble

Figure 1.1: Early satellite images. Source: Wikipedia.

with up to 12 bits of dynamic range. Thus, modern imaging satellites have become reliable platforms for gathering high-quality Earth observation data.

Imaging satellites usually orbit the Earth with a radius of ~ 600 kilometers and a period of ~ 100 minutes. The unique viewing angle and position of imaging satellites make them capable of photographing the Earth surface with broad coverage and high update frequency, which other imaging platforms usually cannot provide. Commercial satellite imaging vendors, such as Planet Labs (Schingler et al., 2010), have developed inexpensive miniaturized satellites that can provide daily updates of specific Earth surface areas.

Commercial space transportation services, such as Space-X and Blue Origin, have significantly lowered the cost of launching satellites into space with reusable rocket platforms. Besides the commercial efforts, different governments and research institutes also heavily invest in next-generation imaging satellite platforms. In the next two decades, a much larger number of modern imaging satellites are scheduled to be launched to better serve the mankind by continuously gathering Earth observation data.

All such advances have evolved imaging satellites from survey-only to surveillance-capable platforms. Large-scale and high-quality visual datasets can be continuously collected from the satellite imaging platforms. Satellite imagery, if properly processed, can benefit many different applications. For example, detection and identification of small features from satellite images, *e.g.* vehicles and roads, can benefit the civil and land planning (Weng, 2002). Example use in the agriculture industry is estimating the crop yields, the location of potential crop diseases, the tree count, from overhead satellite imagery (Kussul et al., 2017). Satellite imagery derived intelligence and tactical planning in urban areas are critical for defense purposes. Homeland security can also benefit from satellite images by monitoring mitigation and assessing crisis events such as earthquakes (Gueguen and Hamid, 2015). Topography and drainage basin gradient studies in hydrology can use satellite images as reliable data sources.

Especially, automatically reconstructing large-scale geospatial areas (both urban and rural areas) from satellite imagery is particularly useful. For example, mobile navigation and au-

onomous driving need high-definition 2D/3D maps; the entertainment industry such as gaming, filming, and virtual reality, requires high-fidelity 3D models for photo-realistic rendering and visual effects; urban planning also benefits from accurate virtual models of the current environment.

However, many of the aforementioned applications still rely on manual inspections, analysis, and modeling. Automated and intelligent processing is thus critical to fully unleash the wealthy information buried in the massive satellite image collections. In this dissertation, we focus on utilizing satellite imagery as a global data source to research the automatic 3D reconstruction of large-scale geospatial areas. But reconstructing satellite imagery is particularly challenging for the following reasons:

1. **The enormous absolute pixel count:** satellite images typically contain several hundred million pixels while each pixel can contain up to sixteen channels. The high pixel count implies a substantial computational complexity (both time and space complexity) for satellite imagery processing. For example, traditional multi-view stereo algorithms would require searching a large disparity space in satellite images (Wang et al., 2014b) to establish stereo correspondences. Thus, 3D modeling from satellite images requires efficient and scalable algorithms.
2. **The low ground sampling resolution:** despite the large geospatial coverage of satellite images, each pixel only covers an area no smaller than $30 \times 30\text{cm}^2$ on the ground. Thus, small-scale objects on the ground are only covered by a few pixels or only contribute to the appearance of a single pixel. Such a low ground sampling resolution poses great challenges for detecting building instances and approximating their shapes.
3. **Complicated and inaccurate sensor imaging models:** image sensors can vary across different satellite image vendors. Formulating precise physical camera models to describe the geometric imaging process is complex. Usually, only approximate mathematical models are provided (Hartley and Saxena, 1997) by the image vendors. Rational polynomial

coefficients (RPC) models (Hu et al., 2004) are a common choice for modeling the satellite imaging process. In addition, due to the approximation errors of the orbital model and the systematic errors of the onboard sensors, satellite images exhibit registration errors when projected onto the Earth surfaces (Ozcanli et al., 2014). Such registration errors, if not properly corrected, can significantly degrade the geometry estimation.

The challenges mentioned above must be addressed appropriately in order to obtain high-quality geometric reconstructions from satellite imagery. This dissertation explores methods to improve the reconstruction efficiency while tackling such challenges to maintain model fidelity.

Beyond geometric reconstructions, semantic analysis of the Earth surface can also benefit many applications. For example, to navigate a drone or an autonomous vehicle, knowing geometry (where are things) alone is not enough. Additional semantic information (what are things) must be obtained to guarantee safety movements, *e.g.*, not driving/landing on water surfaces. However, in the context of satellite image-based land usage analysis, most existing land usage datasets neither provide enough coverage nor enough resolution. For example, land use data provided by USGS have ground sampling distance of at least one kilometer per pixel. The potential usage of such coarse resolution data is rather limited. On the other hand, satellite imagery, captured at high-resolution and with multiple spectral bands, is an ideal data source to build high-resolution land use data at a global scale.

Nonetheless, high-resolution semantic annotations on satellite images are rare to find. The limited availability of ground-truth data prohibits data-driven methods to be applied to achieve such goals. In this dissertation, we explored the possibility of joining geometric reconstruction together with semantic land use classification. We utilize the multiple spectral bands of satellite images to obtain noisy pixel-wise classifications. Such initial semantic maps are then refined through the joint reconstruction process to obtain high-quality semantic land usage classification results.

Compared with high-fidelity 3D point cloud models, parametric models are simpler but still have their unique advantages: (1) parametric models are compact, thus easy to store and trans-

mit; (2) parametric models can provide instance level semantic information. To obtain parametric models, existing map and GIS data sources can be utilized as ground-truth annotations. The coarse geo-registration of satellite imagery provides a direct link between the rich semantic annotations of vector geodetic maps and the raster pixels of satellite images. Such geo-registration, available at a global scale, largely removed the need for manual data annotations. Thus, in this dissertation, we also explored data-driven methods to perform single-view parametric building reconstructions, using satellite imagery and their corresponding 2D vector maps as supervision.

1.1 Dissertation Statement

Satellite imagery serves as global-scale sources for 3D modeling. Efficient geometric reconstructions can be computed by interpolating reliable sparse matches in image space. Semantic modeling can be attained through (1) pixel-level reasoning combining semantic cues and appearance information, or (2) directly estimating instance-level parametric models.

1.2 Outline of Innovations

To summarize, we have made the following innovations to support our dissertation statement. Supporting works have been published as (Wang et al., 2014a,b; Zheng et al., 2015; Wang et al., 2016b; Wang and Frahm, 2017a,b):

1. **Understanding of satellite imaging model:** The imaging process of satellite images are usually described by a Rational Polynomial Coefficients (RPC) model. We propose to use accurate minimal solvers (Zheng et al., 2015) and sparse feature matches to establish reliable pixel correspondences and compensate for extrinsic calibration errors. Sensor calibration errors can then be compensated using bundle adjustment (Wang and Frahm, 2017a).
2. **Efficient stereo reconstruction from satellite imagery:** Satellite images are of high pixel count but low ground sampling rate. To increase the computational efficiency, we propose

to use edge-aware interpolation to propagate reliable sparse feature matches into dense pixel-wise height maps (Wang and Frahm, 2017a).

3. **Joint semantic and geometric reconstruction from satellite imagery:** we propose to utilize the multi-view pixel correspondence information to refine pixel-level semantic classification results. The refinement process aggregates multiple views of the same pixel to improve upon both the accuracy and the smoothness of the final land usage classification results. The semantic refinement objective can be seamlessly integrated into the geometric reconstruction process (Wang et al., 2016b).
4. **Parametric reconstruction from satellite imagery:** We propose to use a data-driven method to perform parametric reconstructions from single-view satellite imagery (Wang and Frahm, 2017b). Buildings can be parameterized as simple 3D cuboid models and can be detected using convolutional neural networks.

In this dissertation, we first review related areas and background knowledge in Chapter 2. Characteristics of satellite images and camera models are covered in Chapter 3. Chapter 4 proposes a fast multi-view stereo pipeline for satellite images. Chapter 5 proposes an algorithm to refine land usage maps by jointly recovering geometry and semantics from satellite imagery. Chapter 6 proposes a data-driven method to perform single-view parametric reconstructions on satellite images. Chapter 7 concludes this dissertation and discusses future research directions.

CHAPTER 2: RELATED WORK

Obtaining high-quality 3D reconstructions from satellite imagery is related to many research fields, for example, computer vision, remote sensing, machine learning, *etc.* In this chapter, we briefly discuss the associated areas and pioneering work.

The goal of 3D reconstructions from the visual input is to recover both the shape and appearance from visual input. Thus, 3D vision pursues the opposite goal of graphics, as indicated in Figure 2.1. The 3D reconstruction process can be accomplished by active (such as depth sensors or LiDAR) or passive sensors (mostly cameras). If the targeted model can change its shape or appearance over time, the 3D reconstruction process is referred as non-rigid 3D reconstruction. In this dissertation, we mostly explore and discuss 3D reconstruction using passive sensors for mostly static and urban scenes.

Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) are two popular 3D reconstruction frameworks. SfM is the process of estimating the 3D structure of a scene from a set of 2D images, while SLAM is the process of constructing/updating a map of unknown environments while simultaneously localizing the agent. SfM and SLAM solve very similar problems with different emphases. SfM traditionally uses on images only, while SLAM allows and integrates additional sensors, such as inertial and active depth sensors. SfM usually performs in an offline fashion, while SLAM aims for low-power and real-time operations. Considering their similarities, we illustrate a common 3D reconstruction pipeline in Figure 2.2.

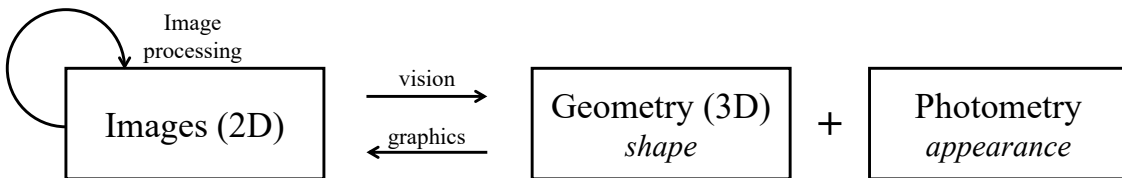


Figure 2.1: Relationship between images, geometry, and photometry.

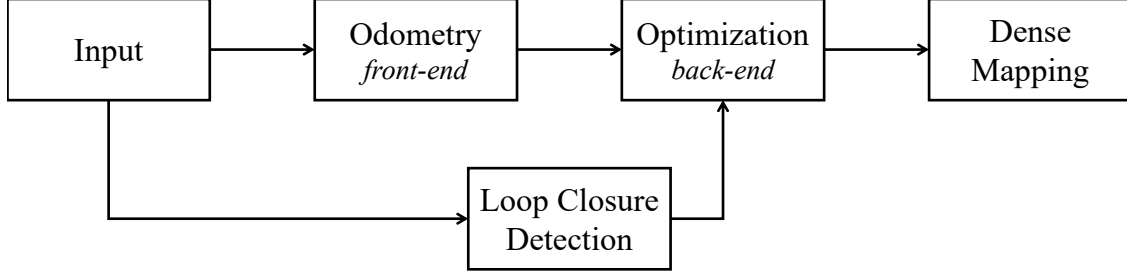


Figure 2.2: Typical 3D reconstruction pipeline.

Depending on available input data and desired output results, some modules in the pipeline may be simplified or even discarded. Images are the most typical input. Additional dense depth maps or 3D point clouds can be obtained via active sensors. Characteristics of different data modalities suitable for 3D reconstruction tasks are presented in Section 2.1.

The odometry module in Figure 2.2 estimates the relative sensor motion between different data captures, for example, the relative camera pose between adjacent video frames or image pairs. Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) systems both incorporate visual odometry in their processing pipelines. The back-end optimization stage takes the recovered 3D structure, and the estimated camera poses to obtain globally consistent reconstructions. Large-scale non-linear optimization techniques are key to scale up the 3D reconstructions. The loop closure module detects if the scene depicted by an image has been visited during the reconstruction process before. If a loop is detected, such information can be helpful for the back-end optimization process to correct the drifting problem. We detail the discussion of SfM/SLAM in Section 2.2.

SfM pipelines and most SLAM pipelines usually generate sparse 3D point clouds and camera poses as output. The subsequent dense mapping stage takes the camera poses as input to generate dense 3D models. Different dense reconstruction methods are covered in Section 2.3.

Besides, we also cover data-driven methods, especially deep learning based techniques, in Section 2.4. Such data-driven methods have been successfully applied to many 3D vision tasks, and are discussed in Section 2.4.2.

Table 2.1: Comparison of major data sources for urban reconstruction tasks. Among passive sensors, ground-level and aerial imagery can provide the highest resolution, while satellite imagery can provide the highest coverage.

Data Source	Ground Imagery	Aerial Imagery	Satellite Imagery	LiDAR	SAR
Sensor	Passive	Passive	Passive	Active	Active
Camera Model	Pin-hole	Pin-hole	Push-broom		
Pixel Count (Million)	20	20	1000	n/a	n/a
Physical Resolution (cm)		5-10	30	30	300
Data Type	RGB	RGB	Hyper-spectral	3D point	3D point
GPS Prior	Partial	✓	✓	✓	✓

2.1 Data Sources for Urban Reconstruction

Urban reconstructions can utilize many different data sources. For example, other than satellite images, common everyday images and LiDAR are also widely used to obtain 3D models for urban environments. For a given area of interest, different data modalities provide highly correlated observations, but usually in vastly different perspectives or formats. Domain-specific priors must be appropriately taken into consideration when utilizing or fusing various data sources. Table 2.1 summarizes the different primary characteristics of available data sources.

2.1.1 Ground-Level Imagery

Images taken on the ground level are very easy to obtain, store, and exchange. However, large-scale ground-level image datasets are usually collected in a crowd-sourcing fashion, for example, by querying GPS locations or landmark names on public photo sharing websites. Such user provided photos can be of varying quality, resolution, and may contain noisy or even wrong text labels.

One major disadvantage of crowd-sourced ground-level image collection is the lacking of association information, especially for large-scale Internet photo collections. For example, one can query the keyword “Rome” on photo sharing websites such as Flickr. The query can return

millions of photos, but only a small fraction of image pairs have visual overlap. Non-overlapping images cannot easily contribute to the 3D reconstruction process. For image pairs with visual overlap, there exist mature computer vision algorithms to find their relative camera poses and to triangulate corresponding 3D points. But the establishment of such image matching relationships is non-trivial. One naive idea is to perform exhaustive trials on all possible image pairs. Such $O(N^2)$ strategy is infeasible on large-scale image collections. Recovering inter-image relationships efficiently on massive-scale photo collections has been a major research topic for computer vision for a long time (Agarwal et al., 2011; Frahm et al., 2010; Heinly et al., 2015; Li et al., 2008; Snavely et al., 2006, 2008).

Additionally, crowd-sourced ground-level imagery suffers from insufficient coverage problem. Intuitively, tourists tend to take photos of famous landmarks but pay less attention to the connecting paths between geographically adjacent landmarks. Such geo-spatially neighboring landmarks can end up as disjoint 3D models. Such data deficiency problem can be alleviated by introducing additional data sources. For example, in Wang et al. (2016a, 2018b), we utilized crowd-sourced video collections to discover geo-spatially adjacent landmark groups and reconstruct the connecting camera trajectory to align disjoint 3D models together.

2.1.2 Aerial Imagery

Aerial Images are collected by airplanes or unmanned aerial vehicles (UAVs). Usually, multiple high-resolution cameras with different viewing angles are mounted on the imaging platform. Inertial sensors and GPS units are typically available on such platforms. Prior pose information can be very beneficial when applying 3D modeling techniques like structure-from-motion on the collected visual datasets.

Compared with ground-level images, aerial images are first used for photogrammetry and city-planning purposes. Thus, airborne image datasets are usually captured in well-planned trajectories, providing better coverage than ground-level photos.

Aerial images also have their inherent flaws. One major limitation of aerial imagery is the data actuation cost. High-quality camera rigs and inertial sensors are expensive, let alone the usually much more costly aviation platforms. In addition to the hardware platform cost, airplane operation costs are also significant. Flying is forbidden in some geospatial areas for security reasons. With the consumer grade drones becoming more and more popular, lower quality aerial images will become more available. Another disadvantage of aerial imagery is the weather conditions. Cloudy or rainy conditions not only pose challenges to the data collection process but also reduce the visibility and quality of the collected images.

Despite the limitations, aerial images have seen great commercial success. For example, the 3D map from Google, Apple, and Microsoft, all utilize aerial images as their major data sources. The high-resolution, high-quality imagery and simple camera models are the main reasons for such choices.

2.1.3 Satellite Imagery

Satellite imagery provides another alternative for urban reconstructions. Imaging devices are mounted on satellites orbiting around the earth. Using Earth-orbiting satellites as imaging platforms, satellite imagery can cover the entire globe within a few days. Satellites usually carry precise GPS receiver and star trackers on board. The position and orientation of imaging sensors can then be precisely calculated when images are taken. Such known camera poses information can be used to directly associate images to the actual imaged region on the Earth surface, usually in the form of GPS coordinates.

In addition, satellite images can provide hyperspectral images of the earth surfaces at a lower resolution. Such hyper-spectral images are useful for many applications, for example, mineral discovery, land use classification, agricultural monitoring, *etc.*

One disadvantage of satellite images is the limited ground-sampling distances. The ground-sampling distances (GSD) describe how detailed a satellite can “see” the Earth surface within a single pixel. State-of-the-art panchromatic satellite images provide ground sampling distance

no better than 30 centimeters per pixel. Hyper-spectral images are usually four times lower than this, at around 1.2 meters per pixel. Such data must be handled with care to obtain accurate 3D models.

Pan-sharpening algorithms (Padwick et al., 2010; Shah et al., 2008; Vivone et al., 2015; Yang et al., 2017a) are usually utilized to borrow information from the higher resolution panchromatic images to increase the discriminating power of the hyper-spectral channels. For example, 30 cm/pixel RGB color images can be obtained by pan-sharpening the 120 cm/pixel hyper-spectral images using the 30 cm/pixel panchromatic images as guidance.

Although the ground-sampling resolution is rather low, satellite images usually contain enormous numbers of pixels. With over thousands of million pixels, each pixel with high dynamic range (up to 12 bits) and multiple spectral bands, the storage and efficient processing of satellite images are a bottleneck to utilize such data modality at a larger-scale.

The visual appearance of satellite images can be severely affected by weather and sun positions. Clouds presented in satellite images are useful for weather forecasting but render the image almost useless for 3D reconstruction purposes. Changes in lighting conditions also make the reconstruction prone to failure.

The most challenging characteristic of satellite imagery is the imaging camera model. Unlike traditional images which can be well described by a pinhole camera, satellite images are taken by push-broom cameras. What's worse, the rigid physical camera calibration information and detailed imaging model is usually not provided by the vendors. A mathematical approximation model is usually provided as a proxy, for example, the widely adopted rational polynomial coefficient (RPC) model. Existing computer vision algorithms need to incorporate such camera models. Details on the satellite camera model are described in Chapter 3.

2.1.4 LiDAR

Light Detection and Ranging (LiDAR), is an active sensing method. LiDAR projects laser light onto surfaces and captures reflected backscattering. The LiDAR sensor then calculates the

distance from the sensor to the hit 3D point in space, based on the time of the round-trip laser flight. LiDAR sensors can be mounted on ground vehicles or aerial platforms, both ground-level LiDAR scans and aerial LiDAR scans can be useful for the urban reconstruction tasks. Here we do not differentiate ground-level and aerial LiDAR. LiDAR can provide accurate semi-dense 3D point clouds directly. Thus, reconstruction methods utilizing LiDAR data usually focus on point cloud registration, segmentation and parametrized high-level polyhedral model fitting. Zhou and Neumann (2008) proposed a data-driven algorithm to learn the principal directions of roof boundaries in airborne LiDAR data. Vanegas et al. (2012) take the Manhattan World building assumption to reconstruct buildings from 3D LiDAR point clouds.

Compared with other active sensors, LiDAR has obvious advantages. LiDAR emits light actively, so it works independently of the ambient lighting conditions. Night, cloudy, or shadows, don't affect the perception capability of LiDAR sensors.

LiDAR's major problems include: (1) High-resolution LiDAR can be very expensive. Until the next-generation solid-state LiDAR sensors are available, the cost of high-quality mechanical LiDAR sensors is limiting their broad commercial applications. (2) The resolution is rather modest. For example, the 64 beam Velodyne LiDAR can only scan 64 rows, generating an image only 64 pixels high. (3) LiDAR should be mounted outside carrying platforms to ensure their perception functionality; thus the device is more prone to operation damage. (4) LiDAR is very sensitive to weather conditions. Heavy rain, fog, and snow can all lead to the failure of LiDAR sensors. Cameras suffer from such severe weather problem too but are less affected. (5) Nowadays commercially available LiDAR are mostly mechanical rotating LiDAR sensors. LiDAR with moving parts is more likely to fail and the mechanical implementation constraints the sensor refresh rate. Low refresh rates can be catastrophic for autonomous vehicles operating at high speed. Solid-state LiDAR which doesn't have mechanical rotating parts thus can have higher refresh rates and lower failure rate, is not widely available yet. (6) LiDAR provides accurate geometry but not much appearance information. Any reconstruction requiring appearance needs

additional data sources and potentially sensor extrinsic calibration (*e.g.* between LiDAR sensors and cameras) to add textures or color information onto the LiDAR point cloud.

2.2 Sparse Reconstruction

3D modeling from images has progressed significantly with the growth of available digital images. Structure-from-Motion (SfM) is the primary technique in computer vision to recover the 3D structure of a mostly static scene from a set of 2D images.

In essence, SfM involves three main stages: (1) extraction of features in images (*e.g.*, points of interest, lines, *etc.*) and matching these features between images, (2) camera motion estimation (*e.g.*, recovering relative pairwise camera positions estimated from the extracted features), and (3) recovery of the 3D structure using the estimated motion and features (*e.g.*, by minimizing the reprojection error). The first two stages described here roughly correspond to the odometry module in Figure 2.2, while the third stage involves both loop closure detection and back-end optimization in Figure 2.2.

Snavely et al. (2006) first proposed a practical incremental Structure-from-Motion pipeline, named Bundler, for crowd-sourced Internet photo collections. Bundler starts by detecting and matching SIFT features (Lowe, 2004) exhaustively for every image pair. For each image pair, the relative camera pose can be solved from the putative feature matches. RANSAC algorithm is applied for outlier removal. Starting with an initial image pair, more images and structures are gradually added to the reconstruction. Bundle adjustment is repeatedly solved to improve the reconstruction quality. Although Bundler is computationally demanding, it successfully demonstrated that large-scale 3D modeling from unordered Internet photo collections is possible, thus sparked a massive interest in the development of efficient large-scale SfM algorithms.

Starting from a few thousand images (Snavely et al., 2006, 2008), large-scale incremental structure-from-motion systems have made great advances over time. Using image retrieval techniques for overlap prediction, Agarwal et al. (2011) processed 150 thousand images in a single day on a computer cluster. Frahm et al. (2010) reconstructed from 3 million images in one day

on a single computer utilizing a compact binary image representation for clustering. Recently, Heinly et al. (2015) pushed the envelope to tackle a world-scale dataset (100 million images) by using a streaming paradigm to identify connected images by looking at each image only once. One of the core computational challenges and the key to improved scalability for large-scale image-based 3D reconstruction systems is the efficient mining for element connectivities within photo collections. Li et al. (2008) introduced the concept of *iconic images* to model the relationship between different image clusters via iconic scene graphs. Frahm et al. (2010) and Heinly et al. (2015) further utilized the iconic representation for better scalability.

2.3 Dense Reconstruction

Structure-from-Motion generates camera poses and sparse point clouds as output. Dense reconstructions are usually performed with such obtained camera poses as input to obtain dense geometry from the given visual inputs. Surface meshes or volumetric occupancy grids are common output results.

Multi-view stereo techniques aim to find the most likely 3D geometry, usually in the form of depth maps or disparity maps, given multiple image observations. Volumetric reconstruction methods represent the target scene directly as a 3D volume. Image appearance evidence or geometry evidence (depth map) is used to determine the occupancy information for each voxel. On the contrary, monocular reconstruction methods only require one image as input to determine dense geometry.

2.3.1 Multi-View Stereo

Multi-view stereo (MVS) methods rely on image-to-image appearance correlation to obtain depth maps or 3D point clouds. We refer interested readers to the excellent tutorial by Furukawa et al. (2015) for a much more detailed introduction.

According to (Scharstein and Szeliski, 2002), common multi-view stereo algorithms can be divided into four stages: (i) *Matching cost computation*: this stage involves designing and computing the similarity of pixel observations from different views. Commonly used cost metrics include the sum of absolute difference (SAD), the sum of squared difference (SSD), and the normalized cross-correlation (NCC). (ii) *Cost aggregation*: a pure pixel-wise matching cost is not robust to mismatches, repetitive textures, or homogeneous regions. The cost aggregation stage tries to mitigate such problem by incorporating the neighborhood information of each matching pixel, for example, by using adaptive support windows (Yoon and Kweon, 2006; Hosni et al., 2013). (iii) *Disparity computation*: a cost volume is constructed from the input images after the first two stages. This stage selects the optimal disparity at each pixel location, either by simply selecting the disparity with the optimal matching cost, or fitting curves to find sub-pixel accurate disparities. (iv) *Disparity refinement*: this stage mainly uses post-processing techniques such as median filters to improve the disparity map quality.

The major computation overhead of stereo computation is usually in the first stage. If the image resolution is high, or the disparity search range is high (for example when the baseline between cameras is big), exhaustively evaluating each disparity hypothesis at each pixel location can be very computationally demanding. Different strategies have been proposed to reduce the matching cost computation overhead. Wang et al. (2014b) proposed a sampling-based algorithm to minimize the potentially huge disparity search range to a much smaller set of disparity hypotheses. After such a reduction, each pixel only needs to evaluate a few disparity hypotheses. Thus this strategy dramatically speeds up the cost computation stage. Wang et al. (2014b) successfully demonstrated the usefulness of such sampling method on high-resolution satellite imagery.

Another strategy to reduce the matching cost computation overhead is to allow costly exhaustive disparity evaluations but only at selected pixel locations. Geiger et al. (2010) proposed to first establish costly matches sparsely across the image. A probabilistic graph model is then formulated on the entire image to fill all unmatched pixels, with the probabilistic model condi-

tioned on the established stereo matches. By using the sparse-to-dense paradigm, Geiger et al. (2010) amortized higher time complexity of sparse matching with the lower time complexity of graphical model inference. Using such strategy, relatively expensive sparse matching methods can be adopted to obtain good initializations.

Similarly, Sinha et al. (2014) first perform sparse feature matches as the first step. Such initial feature matches are clustered into groups to form local 3D planes. Disparity maps are finally obtained by sweeping the local 3D planes through the disparity space. The local plane assumption not only reduced the average pixel-wise computation overhead but also enforces the local smoothness of the final disparity map.

Propagation based approaches, such as PatchMatch (Barnes et al., 2009, 2010), also gained great popularity for their lightweight overhead for matching cost computation. PatchMatch was first proposed by Barnes et al. (2009) as a fast algorithm to find approximate correspondence fields between two images. Starting with randomly initialized correspondence fields, PatchMatch works by propagating the current candidate correspondence at each pixel location to their spatial neighbors, if the candidate is better than the neighbors' current estimation. Empirically PatchMatch based solvers can converge to good solutions very efficiently, usually with only a few propagation iterations.

Notice that for PatchMatch based methods, the time complexity is less dependent on the size of the solution space. With a disparity search range of d pixels, for n propagation iterations, each pixel only evaluates $O(n \log d)$ candidate solutions. If the random sampling of the solution space is only performed at the initialization stage, the time complexity can be further reduced to $O(n)$. Thus, PatchMatch based methods can be very promising to deal with large disparity ranges for stereo estimation problems. Bleyer et al. (2011) over-parameterized each pixel by a small local 3D plane and used PatchMatch to solve for the plane parameters at each pixel location. Such formulation can perfectly model slanted surfaces but is very hard to optimize, considering that each pixel needs to estimate four parameters of a 3D plane instead of just one disparity value. Besides, each 3D plane can have an arbitrary orientation. The solution space is infinite in theory. Since

the time complexity of PatchMatch is independent of the size of the solution space, such formulation can be very efficiently optimized with only a few propagation iterations.

Similarly, Heise et al. (2013) proposed to add a Huber regularizer to the patch match propagation scheme. The variational smoothing formulation with quadratic relaxation allows the explicit regularization of both the disparity and normal gradients using the estimated plane parameters. To better address occlusion problems, Zheng et al. (2014) designed a pixel-wise view selection algorithm for multi-view depth map estimation tasks. However, the scene structure and visibility problem is a “chicken-and-egg” problem. Thus, Zheng et al. (2014) adopted an EM-like (Expectation-Maximization) approach to iteratively update the scene geometry and visibility. Schönberger et al. (2016) further extended Zheng et al. (2014) to unstructured environments by considering not only photometric consistency but also geometric consistencies. The original PatchMatch stereo algorithm mostly optimizes for the pixel-wise objective function, without considering the spatial neighborhood smoothness. Besse et al. (2013) unifies PatchMatch propagation and belief propagation into a common framework to support explicit optimization of pairwise energy functions, thus allowing better spatial smoothness in the results. Galliani et al. (2015) proposed a different hypothesis propagation scheme to accelerate the PatchMatch estimation process.

As a general correspondence searching algorithm, PatchMatch can be applied to problems other than just multi-view stereo. For example, Bao et al. (2014) utilized PatchMatch for optical flow estimation. Since the time complexity of PatchMatch is logarithmic with respect to the solution space, PatchMatch based method can be very powerful to deal with large displacements in optical flow problems. Similar to the PatchMatch stereo algorithm (Bleyer et al., 2011), Hornáček et al. (2014) over-parameterized the optical flow estimation problem by locally oriented planes and used PatchMatch Belief Propagation algorithm to solve for the flow field. To further speed up the estimation of optical flow fields, Hu et al. (2016) embedded PatchMatch scheme within a coarse-to-fine framework, allowing large displacements while maintaining low computation overhead.

Context information is important to recover correct stereo disparity maps from images. Thus, global optimization methods have shown better results compared with methods just incorporating local matching costs. But global formulations, such as graph cut (Kolmogorov and Zabih, 2001) are usually slow to optimize. Hirschmuller (2008) proposed a semi-global aggregation strategy that preserves the low computation cost of local methods but mimics the output quality of global methods. Rather than using expensive solvers to optimize for the best disparity map subject to some user-defined cost function, the semi-global matching algorithm approximates the global optimization process by multiple dynamic programming inference processes. By explicitly modeling slanted surfaces and depth discontinuities in the dynamic programming process, Hirschmuller (2008) achieved good disparity quality at acceptable computation cost.

Plane-sweeping is another well-known technique in 3D computer vision to estimate dense depth maps from multi-view images (Collins, 1996; Gallup et al., 2007; Sinha et al., 2014; Yang and Pollefeys, 2003). Compared with other multi-view stereo algorithms, the plane-sweeping algorithm does not require image rectification. Note that rectification is in general not possible for the multi view scenario. Rectification makes the optical axes of two overlapping images parallel, such that the epipolar lines become horizontal. Horizontal epipolar lines are much easier to search. Given a plane hypotheses (usually parameterized by a plane normal direction and an associated depth d), a 2D homography transformation induced by this hypothesizing homography can be formulated to warp the matching views onto the reference views. When the hypothesizing plane is sweeping through the 3D space, the photometric consistency will be achieved at the correct depth hypotheses. Gallup et al. (2007) proposed to use multiple sweeping directions to better model the slanted surfaces.

Many satellite stereo methods extend traditional MVS approaches to account for the challenges posed by satellite imagery. For example, Wang et al. (2014b) reduces the disparity search space using statistical sampling to accelerate satellite MVS computation. Our work (Wang et al., 2016b) combined satellite-based stereo and scene semantic labels within a PatchMatch propagation framework (Barnes et al., 2009) to achieve more reliable scene geometry and semantic label-

ing. Duan and Lafarge (2016) can efficiently reconstruct city-level scenes from satellite stereo pairs, by jointly estimating geometry and semantics on superpixels.

2.3.2 Volumetric reconstruction

Volumetric reconstruction methods represent the scene space as a collection of voxels. Visual evidence is then used to determine the voxels' occupancy (Kutulakos and Seitz, 2000; Seitz and Dyer, 1999). Compared with surface modeling which only tracks voxels along the surface of the geometric structures, volumetric reconstruction explicitly reasons about the occupancy/emptiness of each voxel.

Volumetric reconstruction methods first discretize the scene space into a voxel grid. Depending on the discretization granularity, hard occupancy and empty decisions for the voxels might lead to structure artifacts. To avoid such artifacts, probabilistic volumetric representations have been proposed. Broadhurst et al. (2001) assign each voxel an occupancy probability and empty space probability. Pollard et al. (2010) model each voxel's occupancy probability as well as its appearance distribution to cope with changing scene illumination. Smooth surfaces or structures can be extracted from the voxel grid by incorporating the probabilistic formulation.

One problem with volumetric reconstruction methods is their poor memory scalability. Naive implementations of the voxel grids imply cubic memory requirements. Given the extremely high resolution of satellite images, often containing tens of thousands of pixels along each image direction, volumetric methods are poorly scalable due to their vast memory complexity. To handle this limitation, Crispell et al. (2012) proposed an adaptive volumetric representation that only samples with high resolution around reconstructed surfaces, but uses very coarse voxels in other areas. This reduces the memory requirements by orders of magnitude while maintaining the probabilistic occupancy and appearance model for the voxels. In general, it is observed that accurate volumetric models often require dozens of images (Crispell et al., 2012). In summary, volumetric methods are not widely used for 3D reconstructions from satellite images.

2.3.3 Monocular reconstructions

Monocular reconstruction estimates dense structures from one input view. Notice monocular SLAM is different, because multiple different views obtained at different time instances are available to the system.

Without correspondences from other views, monocular reconstruction methods need to incorporate prior knowledge to *guess* the scene geometry. Thus data-driven methods are popular for such single view geometric estimations. Saxena et al. (2009) used a Markov Random Field (MRF) approach to estimate the 3D location of small patches. 3D LiDAR point clouds are used as ground truth for training the model. Wang et al. (2014a) used semantic information to refine the geometric estimation. More recently, Eigen and Fergus (2015) combined multiple-scale convolutional features to jointly infer the depth, surface normal and the semantic labels of a given scene. Our work aims at instance level building reconstructions. Thus we unify building height estimation into the detection framework, rather than predicting the pixel-wise height map separately.

Similar to such monocular reconstruction tasks, 3D building reconstruction from one single image is an important and challenging research topic. Most existing single view methods exploit shadow information to estimate the building height by a geometric analysis (Ok et al., 2013). Such methods require precise metadata like sun illumination and sun-earth positions for geometric estimation. Differently, our work (Wang and Frahm, 2017b) performed parametric reconstructions from single-view satellite images in a data-driven approach, requiring only image as input.

2.4 Data-driven Methods for 3D Reconstructions

Computers are good at tasks that can be explicitly defined but need large-amounts of computation. Humans, on the other hand, are good at high-level reasoning and perception, which are hard to describe explicitly using formal rules. Many vision tasks are non-trivial because the tasks cannot be easily solved by defining simple formal rules.

Learning based methods have become the mainstream approaches for high level knowledge extraction. Instead of explicitly describing the rules to achieve a specific goal, a mathematical model is first defined to map visual input, usually images, to the desired high-level outputs, for example, bounding boxes and categorical labels for the desired object. By optimizing the model with respect to the desired objective function, high-level tasks can be achieved implicitly without programming the rules explicitly.

2.4.1 Deep Learning

Deep learning is a representative machine learning techniques that has revolutionized many computer vision fields. Deep neural networks can automatically build hierarchical feature representations from the input. By chaining non-linear functions as layers into a network, deep learning models can compute increasingly abstract and high-level features automatically from large amounts of training data. By using powerful back-propagation optimization techniques, deep learning models can be efficiently and effectively trained in an end-to-end fashion. We refer interested readers to the excellent book by Goodfellow et al. (2016) for a systematic introduction.

First popularized for visual recognition tasks, deep learning methods can also improve many 3D geometric vision tasks. We first review the essential deep learning concepts in Sec 2.4, then discuss recent advances on using deep learning methods for 3D vision tasks in Sec 2.4.2.

2.4.1.1 Convolutional Neural Network

Convolutional neural networks (CNNs) are a specialized type of neural network. CNNs, as the name indicates, utilize *convolution* operations to process data with grid-like topologies. Thus, CNNs are very powerful and popular for images with pixels stored in 2D grids. The core convolution operation applies a linear transformation at every spatial location of the input grid and outputs a weighted average value. Weighting coefficients are aggregated as *kernels*.

Modern CNN models can carry huge capacity. Having achieved significant improvements for many vision tasks with modern CNN models, the research community now starts to trade off

between runtime efficiency and inference accuracy. The better understanding and design of basic convolution operations are fundamental to enable such a trend.

Simonyan and Zisserman (2015) proposed to use a series of small convolution kernels to replace larger kernels. For example, two 3×3 convolution kernels have the same receptive field as one 5×5 kernel, but contain fewer parameters to train ($9 + 9 < 25$), more non-linearities, and better computational efficiency.

Especially, the use of 1×1 kernels interests the community. The 1×1 kernel cannot change the spatial resolution of the output feature map, but can change the number of feature channels. Thus, 1×1 kernels are widely used to reduce/increase the dimensionality of feature maps.

The Inception module proposed in GoogLeNet (Szegedy et al., 2015) aggregates input features at different scales to achieve a better visual understanding of the input images. The 1×1 *bottleneck* filters are used to reduce the multi-scale feature map to manage the overall computations needed.

For traditional convolution operations, the theoretical receptive field of the output neuron increases linearly with the number of preceding layers. Yu and Koltun (2015) proposed dilated convolution to introduce spacing between the convoluted spatial locations. For dilated convolution layers, the receptive field of output neurons can increase exponentially with respect to the number of layers. Although the actual learned receptive field of output neurons might be much smaller than the theoretical upper bound.

Standard convolutions are performed on multiple input feature channels at the same time. Group-wise convolutions first divide the input channels into disjoint groups. Each group is convolved separately then concatenated together. Chollet (2016) pushed such idea to an extreme and proposed *depth-wise* convolution: instead of convolving over every input channel, use convolutional filters to process each input channel individually first, then use 1×1 filters to combine them. Such Xception network can be more computationally efficient.

Besides group-wise/depth-wise convolution, channel shuffling can also benefit performance at minimal computation cost. For example, ShuffleNet (Zhang et al., 2017) randomly shuffles

the order of feature maps before the depth-wise convolution operation. ShuffleNet can achieve state-of-the-art performance on many vision benchmarks with significantly fewer parameters.

Standard convolution kernels are of regular shapes. Thus CNNs are inherently limited to model geometric transformations due to the structure/shape constraints of the underlying convolution kernels. To better accommodate the shape of the actual objects in the input data, Dai et al. (2017) proposed deformable convolution kernels as well as deformable RoI (region of interest) pooling modules. Both spatial convolution and pooling involve spatial sampling operations. The proposed deformable module introduced learnable parameters to model the offsets of the spatial sampling locations. Such deformable convolution and pooling operations can directly replace traditional convolution/pooling modules.

2.4.1.2 Recurrent Neural Networks

CNN models are largely “stateless” functions, which transform input data to desired output. There exists a different type of neural networks, recursive neural networks (RNNs), which have cyclic connections inside the network. Such connection loops allow information to persist in the network, making RNNs able to “memorize” information. Plain RNN models, although quite powerful, cannot handle long-term dependencies in the data very well. A special variant, long-short term memory (LSTM) network Hochreiter and Schmidhuber (1997), demonstrated its effectiveness in handling long-term dependencies, thus has become quite popular in computer vision/natural language processing research. The LSTM network introduced the concepts of “gates”: which control the information flowing through the neural cell. The gating mechanism has proven its effectiveness in not only handling long-term dependencies but also in controlling the gradient explosion problems.

With such “memory” capability, RNNs are especially powerful for processing sequential data, such as texts written in natural languages. The success of modern convolutional and recursive network models has enabled many new research topics combining vision and language. Image captioning is a perfect example: generating a corresponding natural language description for

the given visual image input. Vinyals et al. (2017) proposed a generative model to maximize the likelihood of the target description given the input image. Visual features extracted by convolutional neural networks are used as input to the LSTM module, while the LSTM model generates the description in a recurrent fashion, one word at a time.

Combining visual and textual information is a great way to achieve a better semantic understanding of the input data, thus improving the performance of the desired tasks. For example, to utilize all available information within an online tweet, in Wang et al. (2017a, 2018a), we utilized CNNs to extract visual features and LSTM-RNN models to extract textual features. Such neural features are jointly embedded together with social and temporal information. By employing a Poisson regression model, the jointly learned feature representation can achieve state-of-the-art retweet prediction performances.

2.4.1.3 Activation Function

Convolution, after all, is still a linear transformation, but the objective function of most neural networks are highly non-linear. To learn such non-linear mappings, non-linearities must be introduced into the networks. Non-linear activation functions are widely used after convolutional operations to add non-linearities to the neural networks. In fact, advances in the activation functions play an important role in the success of modern neural networks. Commonly used activation functions are summarized in Table 2.2.

The traditional Sigmoid and tanh activation functions suffer from gradient saturation problems when $|x|$ becomes large, which can lead to the gradient vanishing problem in very deep neural networks.

The Rectified Linear Unit (ReLU) function significantly reduced the problem of gradient vanishing because ReLU has a constant gradient for positive inputs. However, for negative inputs, ReLU always outputs a zero. If the previous layer is not initialized properly or has learned a very large negative bias term along the way, the ReLU unit can make the neuron “dead” by always outputting a zero gradient. The pReLU unit was then proposed (He et al., 2015a) to avoid such

“dead” ReLU problems by introducing a learnable negative slope in the original ReLU function. LeakyReLU and eLU functions have motivations similar to pReLU but different mathematical formulations.

2.4.1.4 Architecture

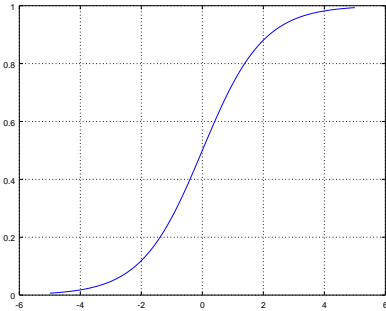
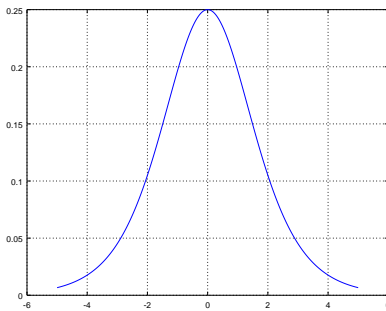
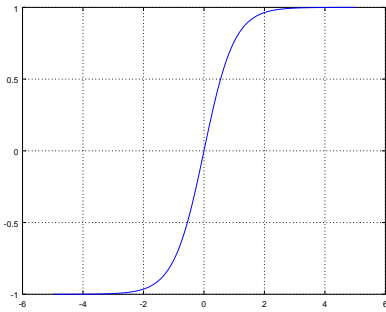
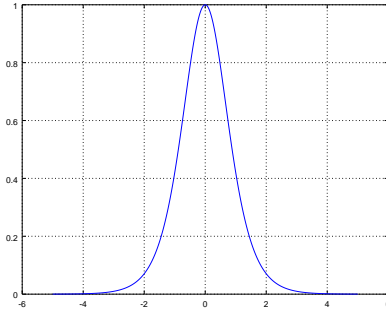
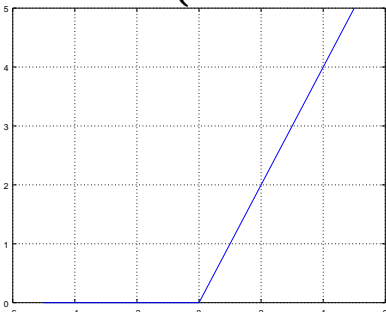
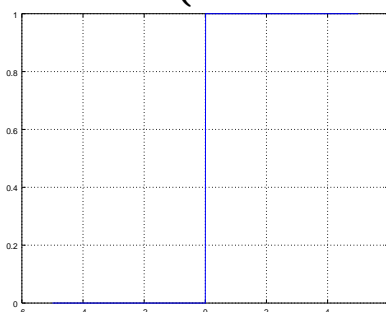
Starting from AlexNet (Krizhevsky et al., 2012), utilizing GPUs to train deep convolutional neural networks has become very effective methods to solve vision tasks. The architectures of CNNs have rapidly evolved since then. For example, Simonyan and Zisserman (2015) proposed the well-known VGG architecture, which popularized the use of small kernels to train very deep convolutional networks. Szegedy et al. (2015) proposed the Inception module to aggregate information at different scales in the GoogLeNet architecture. Xception (Chollet, 2016) utilized depth-wise convolution to optimize the inference computation efficiency. MobileNet (Howard et al., 2017) reduced the computation overhead of modern CNNs by streamlined architectures and depth-wise separable convolutions. ShuffleNet (Zhang et al., 2017) proposed to shuffle the depth-wise convoluted feature channels to further improve the inference accuracy with minimal computation overhead.

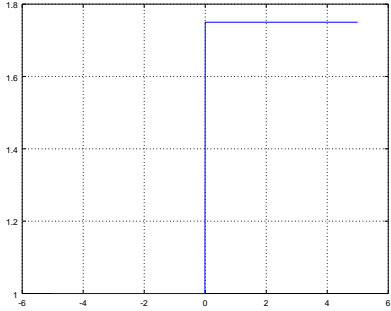
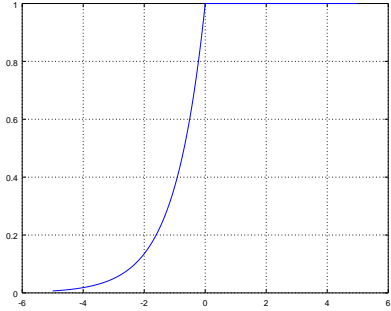
One of the most important advances of network architectures is the discovery of using skip-connections to create other pathways to connect layers. ResNet (He et al., 2016) proposed to learn the “residual” of the input data. The skip-connections used in the ResNet architecture allows both the data and gradient to flow forward and backward without any difficulties. Such architectural changes successfully enabled the training of very deep networks, with hundreds of layers compared to tens of layers before. Densely-connected networks (Huang et al., 2017) used more skip connections to let every layer fuse the feature outputs from different layers.

2.4.2 Deep Learning for 3D Vision

Geometric vision needs extra information or prior knowledge to deal with ambiguous scenarios, for example, homogeneous regions or repetitive textures. Data-driven methods could auto-

Table 2.2: Activation functions commonly used in neural networks

Name	Equation	Gradient
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x) \cdot (1 - f(x))$
		
tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
		
ReLU	$f(x) = \begin{cases} x & x \geq 0 \\ 0, & x < 0 \end{cases}$	$f(x) = \begin{cases} 1 & x \geq 0 \\ 0, & x < 0 \end{cases}$
		
pReLU	$f(x) = \begin{cases} \alpha x & x \geq 0 \\ x, & x < 0 \end{cases}$	$f(x) = \begin{cases} \alpha & x \geq 0 \\ 1, & x < 0 \end{cases}$

Name	Equation	Gradient
leakyReLU	$f(x) = \begin{cases} x & x \geq 0 \\ 0.01x, & x < 0 \end{cases}$	
ELU	$f(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$	

matically learn such prior knowledge through large amounts of training data. Thus, deep learning technologies have greatly advanced the progress of many 3D vision tasks. Compared to many

visual recognition tasks, prior knowledge has been well studied for geometric 3D vision problems (Hartley and Zisserman, 2003). How to effectively utilize such existing knowledge is very important to train successful CNN models for 3D vision tasks.

2.4.3 Datasets for Learning

High-quality data play a central role in enabling many computer vision technologies. For example, the evolution of stereo algorithms greatly benefited from the Middlebury dataset (Scharstein and Szeliski, 2003). However, the collection and labeling process of large-scale datasets for 3D vision tasks are tedious and costly. KITTI (Geiger et al., 2012; Menze and Geiger, 2015), DTU (Jensen et al., 2014), and ETH3D (Schops et al., 2017) all require carefully calibrated camera rigs to capture images, expensive Laser scanner or structured light sensor to obtain the ground truth. The Cityscapes dataset (Cordts et al., 2016) even requires human labeling. Even worse, many existing datasets cannot be easily adapted for new tasks. Thus, gathering new datasets in a cost-effective fashion has always interested researchers in the current deep learning era.

One popular way to collect training data is to utilize photo-realistic rendering engines to synthesize large amounts of data with known ground-truth. Models trained on such synthetic datasets demonstrated on-par or even better performances with respect to models trained on real-world datasets. For example, Dosovitskiy et al. (2015) and Ilg et al. (2017) utilized synthesized images with known camera motion to train CNNs for optical flow estimation. Mayer et al. (2016) employed rendering software to generate data for scene flow estimation tasks. Richter et al. (2016) and Richter et al. (2017) exploited game engines to obtain pixel-wise semantic labels with minimal human efforts. Existing CAD models are also proven useful (Chang et al., 2015). Shah et al. (2017) proposed a complete game simulator to synthesize training images with ground-truth geometry. A physical collision detection engine is also provided, thus enabling reinforcement learning and robotics research. Ley et al. (2016) proposed a synthetic dataset generated for 3D reconstruction evaluation. Ros et al. (2016) proposed a synthetic dataset created for urban scene

semantic segmentation tasks. Richter et al. (2016) utilized a game engine to easily obtain large amounts of training data for visual perception.

In the meantime, ground-truth information is much harder to obtain. Although utilizing unlabeled data in unsupervised or semi-supervised fashion is challenging, specific domain knowledge can be utilized to *interpolate* the labeled data or provide information for the unlabeled data. For example, the left-right consistency check can be used to supervise the monocular depth estimation task (Godard et al., 2017) and the stereo matching task (Zhou et al., 2017a). Zhou et al. (2017b) trained a CNN model to estimate camera motion by minimizing the re-projection errors of the reconstructed 3D models. Tonioni et al. (2017) proposed an unsupervised way to adapt trained stereo estimation networks to new datasets.

2.4.4 Learning SfM/SLAM

Structure-from-Motion (SfM) and Simultaneous Localization and Mapping (SLAM) are large pipelines that involve many different vision tasks, including feature matching and tracking, camera pose estimation, and many more (see Figure 2.2 for details). Some components, for example, bundle adjustment in SfM and pose graph optimization in SLAM, can be very efficiently solved analytically. Thus there's less need to use data-driven methods to fit/mimic such modules. On the other hand, some components can be greatly improved by CNNs. For example, identifying already visited places and closing the loop can benefit from better visual feature representation of the visual images. Increasing the robustness of feature point detection and tracking, incorporating semantic information to handle dynamic objects in the scene, extracting features at different levels (point, line, facades, objects), and generating dense/semi-dense maps efficiently, can all be exciting problems suitable for CNNs to solve.

2.4.4.1 Learning Odometry

Visual odometry is the task of determining the camera pose from visual input. The determined camera pose is usually relative to some world coordinate system, or some initial camera

poses. Thus it is natural to use recurrent networks to model the visual odometry problem. Recurrent networks can learn and memorize the visited scenes, thus having the capability to determine the camera pose of new input data.

PoseNet, proposed by Kendall et al. (2015), is one of the earliest work in using deep neural networks for camera localization problems. Kendall et al. (2015) proposed to use a CNN network to predict the camera rotation and translation, using Structure-from-Motion reconstructions as supervised training data. PoseNet directly regresses the camera rotation and translation using a CNN model. Although conceptually simple, PoseNet demonstrated the feasibility of using data-driven methods to solve traditionally well-studied geometric vision problems. VINet (Clark et al., 2017b) reformulate the Visual-inertial odometry problem as a sequence-to-sequence learning problem. Sequence-to-sequence learning was originally introduced by Sutskever et al. (2014) for machine translation tasks, where parallel text descriptions are given in different languages. The sequence-to-sequence model is trained to translate one language description into the other language description. Here in the visual odometry setting, given a sequence of images and ground-truth IMU data, the VINet extracts visual features using CNNs, then employs an RNN model to “translate” the CNN features to camera poses. Similarly, Wang et al. (2017b) also utilized recurrent neural networks to perform visual odometry. By modeling the entire visual odometry problem as an end-to-end neural network pipeline, the DeepVO network greatly simplified the need to individually develop and tune each stage of the visual odometry pipeline. Li et al. (2017) then extends DeepVO (Wang et al., 2017b) to enable unsupervised learning by using stereo image pairs for training. Walch et al. (2017) proposed a CNN+LSTM architecture to regress the camera pose. LSTM performed structured dimensionality reduction on the CNN feature vectors, bringing drastic improvements in the localization accuracy. Wu et al. (2017) suggested using Euler angles to represent the camera orientation. Camera poses are usually sparse and clustered in many reconstructions. Data augmentation is used in (Wu et al., 2017) to cope with overfitting. Clark et al. (2017a) proposed the VidLoc network to utilize the temporal con-

constraints in short video clips (20 frames) to estimate the 6 DoF camera poses. The temporal constraints helped to smooth the camera poses and also decrease the localization error.

2.4.4.2 Learning Map Generation

In the SLAM context, the map is the condensed knowledge of the surrounding area obtained by the robot agent. Depending on applications, different maps are needed: for cleaning robots, 2D maps are sufficient, for drones, 3D occupancy voxel grids are required. Hereby the map generation is the process of combining previous knowledge with current visual inputs.

CNN-SLAM (Tateno et al., 2017) employed a monocular depth estimation CNN module to augment traditional SLAM, which greatly increased the density of the obtained map. In addition, the dense geometry can also help to produce semantically coherent reconstructions from single view observations.

Depth estimation from monocular images is inherently ambiguous. To obtain robust and accurate depth estimation, Ma and Karaman (2017) proposed to use a regression network to predict the dense depth map, with monocular image and sparse depth samples as input. The depth samples can be obtained either from low-resolution depth sensors or SLAM triangulations. Neural networks can be very powerful to learn prior knowledge from image data. Such sparse-to-dense strategy utilized such prior knowledge to fill in the missing depth values using monocular images as guidance.

Running neural network based semantic segmentation algorithms can be expensive for robotic platforms. Li and Belaroussi (2016) proposed to utilize the temporal and spatial smoothness of the SLAM map to obtain semi-dense 3D semantic maps. 2D semantic information is propagated to 3D mapping via geometric correspondences, without the need to run semantic segmentation for each frame.

Although neural networks provide promising methods to generate dense depth maps and semantic maps, how to intelligently integrate existing prior knowledge to reduce the computation

overhead of heavy neural networks, is still an important and open-ended question to be solved for neural network based map generation algorithms.

2.4.4.3 Learning End-to-End SfM/SLAM Pipelines

Learning based methods have been proposed to replace/optimize many modules of the SfM/SLAM pipeline, but few learning-based systems have demonstrated the capacity of performing structure-from-motion or SLAM end-to-end and efficiently.

Zhou et al. (2017b) proposed an unsupervised framework to train neural networks to predict the depth and pose from video sequences. SfM-Net Vijayanarasimhan et al. (2017) performed similar tasks using supervised learning. Given a sequence of frames, SfM-Net predicts the dense pixel-wise frame-to-frame motion field (optical flow). By warping images in the time domain in a differentiable manner, the SfM-Net can be trained by minimizing the pixel matching errors.

Although success has been achieved by porting CNNs to the SLAM/SfM domain, the limitation of widely available training data severely limits the practicality of such methods. Training a robust network that generalizes well to unseen datasets and scenarios is still challenging. In addition, SLAM requires real-time efficiency in most application scenarios. Most of the learning based CNN methods are far behind this criterion.

Another limitation of using neural networks to do geometry-aware tasks, such as SfM, is the training data. Several strategies have demonstrated their successes. By completely modeling the perspective projection process, self-supervision can be achieved by minimizing the re-projection error (either geometric or photometric) Zhou et al. (2017b); Vijayanarasimhan et al. (2017). If ground-truth camera motion is available, for example, through IMU, neural networks can be trained to learn the camera motion (Clark et al., 2017b,a; Kendall et al., 2015; Li et al., 2017; Walch et al., 2017; Wang et al., 2017b; Wu et al., 2017; Sutskever et al., 2014). Depth sensors, such as RGBD cameras or LiDAR sensors, can also be used to enable supervised training of geometry-aware networks (Tateno et al., 2017; Ma and Karaman, 2017). Among all the afore-

mentioned strategies, to enable neural-network based SLAM systems for higher robustness and scalability, un-supervised learning approaches have more potentials.

2.4.5 Learning Feature Matching

Correspondences are essential to solve 3D vision problems. Machine learning provides an alternative when no such correspondences can be reliably obtained with traditional methods. Currently, there are several different approaches to applying data-driven techniques to establish correspondences: (i) learning a better feature representation directly from the visual input; (ii) learning a better cost function or similarity measure, better known as *metric learning* in the literature; (iii) formulating the correspondence estimation problem into the entire problem-solving framework, such as stereo, and perform end-to-end learning on the entire problem. We briefly review the first two approaches in this section and discuss correspondence estimation in the stereo context in Sec 2.4.6.

Feature point detection and matching are critical for real-time SLAM applications. Modern CNNs require heavy computation on powerful GPUs. To alleviate the computational overhead, DeTone et al. (2017) proposed very efficient networks that can run in real time on CPUs to perform feature point extraction and matching using CNNs. The MagicWarp network directly estimates the homography transformation between an image pair with only keypoint locations. Such strategy is very different from the traditional descriptor matching based transformation estimation algorithms.

Data-driven methods can also be used to build better local feature representations or patch similarity measures.

Zbontar and LeCun (2016) used convolutional neural networks to compute stereo matching cost and outperform traditional measures like sum of squared distances (SSD) and normalized cross-correlation (NCC). Knobelreiter et al. (2017) also utilized CNNs to compute the matching cost but unified the CNN together with a conditional random field (CRF) to compute the final disparity map. Yang et al. (2017b) proposed the concept of complementary descriptors to obtain

better patch representations using networks. Tulyakov et al. (2017) learns a cost function with weakly supervision. Yi et al. (2016) proposed an end-to-end system to perform interest point detection, orientation estimation, descriptor extraction using CNNs.

Such patch similarity measures can be embedded into many different 3D vision tasks, for example, stereo matching or structure-from-motion. By embedding the learned matching function into traditional methods or pipelines, existing algorithms and prior domain knowledge can be incorporated very easily. For example, many existing stereo refinement algorithms can still be applied. Although good accuracy can be achieved, exhaustive computation is required on many image patches, or even all possible matching patch pairs. Moreover, patch based systems operate on groups of sampled patches, they lack the capabilities to incorporate larger global contextual information to resolve ambiguities. One strategy is to use multi-scale patches. However, global context information is critical to differentiating local ambiguities, as shown in modern semantic segmentation systems (Chen et al., 2016). Thus, working only at patch level might not be sufficient.

2.4.6 Learning Stereo

Computing the photo-consistency or similarity score between two image patches is at the core for two or multi-view stereo tasks. Chen et al. (2015) trained a multi-scale CNN to learn a better cost function. Luo et al. (2016) treated disparities as discrete class labels, thus solving the stereo matching problem efficiently as a classification problem. To better incorporate global context information, Kendall et al. (2017) explicitly computed the cost volume to let the network capture more spatial cues. On the KITTI dataset (Geiger et al., 2012; Menze and Geiger, 2015), ground-truth disparity maps are sparse, thus weakly supervised methods (Kuznetsov et al., 2017; Tulyakov et al., 2017) are proposed to better utilize the training data. Knobelreiter et al. (2017) proposed a joint CNN-CRF model that produces stereo estimations. CNN modules are simply used to compute the unary and pairwise costs for nodes in the CRF. Rather small CNNs are used to compute the unary and pairwise costs. Thus the CNN matching costs are rather noisy for di-

rect stereo prediction. The successful joint training of CNNs and CRFs proves that incorporating higher-order context is very helpful for structured prediction tasks. Other common ways of doing such things include multi-scale CNNs, basically running multi-scale networks on the same input data and fuse the resulting multiple feature maps. Post-processing using CRF/MRF is another commonly used method. Pooling feature maps at different scale is another option, for example, spatial pyramid pooling (He et al., 2015b) and feature pyramid network (Lin et al., 2017).

Although significant progress has been made for the stereo estimation task, certain errors still exist in the predicted disparity maps. Confidence measures are thus used to quantify the probability of erroneous predictions. Haeusler et al. (2013) aggregated multiple hand-engineered confidence measures and trained random forest classifiers based on such measures. Pfeiffer et al. (2013) utilized stereo confidence to refine high-level tasks such as object detection in real-world traffic scenarios. Park and Yoon (2015) leveraged the confidence score predicted by the trained random forest to refine the stereo estimation. Mostegel et al. (2016) utilized multi-view conflicts to identify reliable and unreliable pixels to enable data-driven methods to learn confidence measures for stereo estimations. Poggi and Mattoccia (2017) utilized a convolutional neural network to exploit spatially local consistency to improve upon conventional pixel-wise confidence measures. Shaked and Wolf (2017) proposed the highway network to estimate stereo disparities and used confidence to further refine the predicted results. Marin et al. (2016) utilized stereo confidence measures to reliably fuse sparse active sensor collected depth map with stereo estimations for accurate and high resolution depth maps. Poggi et al. (2017) quantitatively evaluated stereo confidence measures.

Ummenhofer et al. (2017) proposed a network (DeMoN) to predict depth and motion at the same time from a pair of unconstrained images. DeMoN establishes dense correspondences between two images via predicting the optical flow between the image pair. Spatial relative differences predicted via optical flow are minimized iteratively through multiple network modules to infer the output. DeMoN demonstrated state-of-the-art performance on predicting camera poses and depth maps.

2.4.7 Learning Optical Flow

Stereo matching establishes dense pixel-wise correspondence across images. With the epipolar constraints, the correspondence search is applied on a 2D line instead of a general 2D area, thus can be relatively efficient.

For traditional pinhole camera models, epipolar constraints (Hartley and Zisserman, 2003) reduce the correspondence search space to an epipolar line. However, for satellite images with RPC models, such simple 1D constraints are not easily obtainable. Thus, the general correspondence search problem resembles 2D optical flow estimations. Revaud et al. (2015) proposed an edge-preserving interpolation method to greatly improve the optical flow accuracy, especially for situations with large displacements. Anderson et al. (2016) only computed coarse 2D correspondences and relied on bilateral smoothing (Barron and Poole, 2016) to obtain high fidelity optical flow fields.

FlowNet (Dosovitskiy et al., 2015) proposed to use neural networks to learn to predict flow field in an end-to-end fashion. Synthetic datasets are used to train the neural network. FlowNet 2.0 (Ilg et al., 2017) improves the performance of optical flow estimation on top of FlowNet. Hierarchical coarse-to-fine variational refinements are the key to the high performance of many traditional methods. FlowNet 2.0 mimics such coarse-to-fine refinements with a stack of network modules.

2.4.8 Learning Surface Normals

Surface normals can be useful cues for not only geometric 3D reconstructions but also for scene understanding and visual recognition. Galliani and Schindler (2016) utilized multi-view stereo reconstructions as ground truth to train CNNs to predict surface normals, and surface normals are used to improve upon 3D reconstruction quality. Wang et al. (2015) trained convolutional neural networks to predict surface normals from single view images. Chen et al. (2017a) proposed novel loss functions utilizing the annotated surface normal data to improve upon depth predictions.

2.4.9 Learning Monocular View Tasks

Significant advances have been achieved by convolutional neural networks on many challenging single-view tasks. Eigen et al. (2014) first introduced a multi-scale CNN to regress scene depth from a single image, then extended the network (Eigen and Fergus, 2015) to predict pixel-wise depth, surface normal, and semantic labels at the same time. Roy and Todorovic (2016) embedded shallow CNNs within the nodes of random forests. Depth values are then regressed on the leaf nodes. Kim et al. (2016) unified the monocular depth estimation and intrinsic image decomposition tasks and solved them jointly using a CNN. Chen et al. (2017a) proposed novel loss functions utilizing the annotated surface normal data to improve upon depth predictions.

Pixel-wise prediction results can sometimes be noisy if contextual constraints are not enforced on the network. Li et al. (2015) thus adopted a conditional random field (CRF) to refine the network outputs. Xu et al. (2017) fused the intermediate outputs at different stages of the network with multiple continuous CRFs for final prediction.

Besides contextual constraints, motion information is also used to train network for monocular depth estimation tasks. Ummenhofer et al. (2017) predict the depth map and camera motion for a given image pair. By learning the concept of matching, DeMoN can better generalize to unseen scenarios. Garg et al. (2016) used geometry information to train CNN for monocular depth estimation in an unsupervised fashion. Ranftl et al. (2016) recognized two consecutive frames and utilized such temporal information to improve upon monocular depth estimation results.

2.4.10 Learning Multi-view Tasks

Multiple overlapping images can be used for 3D visual recognition tasks in addition to traditional 3D reconstructions. Su et al. (2015) utilized view pooling to aggregate multi-view convolutional features to perform 3D shape retrieval and recognition tasks. Qi et al. (2016) utilized multi-view and volumetric representations to perform 3D shape recognition. Choy et al. (2016) relied on an LSTM network to unify single-view and multi-view reconstructions. Ji et al. (2017) proposed an end-to-end 3D CNN to directly produce MVS reconstructions. However, each view

is fed through the recurrent network separately, the recurrent network is required to aggregate information from different views to generate better reconstructions. Li et al. (2016) adopted the LSTM layer to memorize and fuse information from different data sources and modalities. Chen et al. (2017b) simply used fixed viewpoints and fixed architectures for 3D object detection. Hartmann et al. (2017) use feature averaging to reach a consensus on photo-consistency for multiple input patches. Kar et al. (2017) proposed to learn multi-view reconstructions by performing convolution features projection and unprojection in a volumetric grid representation.

However, a volumetric representation is often utilized for deep learning based methods , thus limiting the scalability of the generated output. Riegler et al. (2017a,b) proposed to use octrees instead of regular volumetric grids to reduce memory footprints so that higher resolution 3D models can be learned.

CHAPTER 3: SATELLITE IMAGING CAMERA MODEL

In this chapter, we describe the unique characteristics of the satellite imaging models.

3.1 Spatial Reference Systems

Modern imaging satellites carry GPS receiver that localizes the satellite, as well as star trackers to identify its orientations. Star trackers are essentially cameras that look up at the sky and determine the orientation of the satellite telescope. The combination of satellite position (from GPS) and satellite orientation enables the accurate geo-registration of satellite images.

In addition to knowing where the image is taken, every pixel in the image can be roughly located. A bidirectional mapping exists between image pixel locations and actual spatial locations on the Earth's surface. There are two widely used approaches to represent such bi-directional mapping: affine geo-transformation and ground control points (GCP).

The six-degree-of-freedom affine geo-transformation $\mathbf{A} \in \mathbb{R}^{2 \times 3}$ establishes a mapping between the raster pixel coordinates (row y and column x) and the geo-referenced coordinates (usually in the form of latitude lat and longitude lng):

$$lng = a_{11} + a_{12}x + a_{13}y \quad (3.1)$$

$$lat = a_{21} + a_{22}x + a_{23}y \quad (3.2)$$

The above affine transformation can be written in the following equivalent matrix form:

$$\begin{bmatrix} lng \\ lat \end{bmatrix} = \mathbf{A} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} \quad (3.3)$$

In the case of north up images without any in-plane rotation, the affine transformation is perfectly interpretable: a_{13} and a_{22} are zero, a_{12} is the pixel width, a_{23} is the pixel height, while (a_{11}, a_{21}) is the GPS location of the top left corner of the image raster.

Ground control points (GCP) provide another form of geo-registration. Each ground control point offers ground-truth correspondence between a pixel location (x, y) and a 3D point location (lat, lng, alt) . Usually, it is up to the application to fit a transformation model that establishes dense pixel-to-3D correspondences. Low order polynomials (1st to 5th) are common choices.

With such mapping, different layers of geo-spatial data can be overlaid together for visualization and processing. A valid spatial reference system is required to represent spatial locations on the Earth's surface. Thus, the ground control points from one satellite image must share the same georeferencing coordinate system.

Traditional Cartesian coordinate systems provide a bijection between coordinates and locations. For example, in the 2D Euclidean space, one can use $\{(x, y) | x, y \in \mathbb{R}\}$ to represent all points on the 2D plane. Similarly, georeferencing coordinate systems aim to provide a two-dimensional coordinate representation (lat, lng) for each point on the Earth surface. But the Earth is an ellipsoid, representing a 3D sphere with 2D encodings has certain drawbacks. For example, many georeferencing coordinate systems do not preserve the original shape/angle/distance in the encoded space.

3.1.1 Geodetic Datum

Coordinate systems need references. For example, a 2D Euclidean coordinate system needs an origin point. For a geo-referencing system, a reference geodetic datum acts as the origin point. The geodetic datum is an approximation of the Earth's surface against which positional measurements are made for computing locations. Horizontal datums are used for describing a point on the Earth's surface, in latitude and longitude or another coordinate system. Vertical datums are used to measure elevations or underwater depths.

Table 3.1: Earth reference ellipsoids

Ellipsoid	Year	Semamajor axis (m)	Seminimor axis (m)	Inverse flattening
GRS	1980	6,378,137.0	6,356,752.314 140	298.257222101
WGS	1984	6,378,137.0	6,356,752.314 245	298.257223563
IERS	1989	6,378,136	6,356,751.302	298.257
IERS	2003	6,378,136.6	6,356,751.9	298.25642

Table 3.2: Exmaple geo-location in different geodetic datum.

Reference	Longitude	Latitude
NAD 1927	-122.46690368652	48.7440490722656
NAD 1983	-122.46818353793	48.7438798543649
WGS 1984	-122.46818353793	48.7438798534299

The Earth can be approximated by an ellipsoid. The ellipsoid is completely parameterized by the semi-major axis a and the flattening f . From a and f it is possible to derive the semi-minor axis:

$$b = a(1 - f) \quad (3.4)$$

As the geology technology advances, more precise and global reference ellipsoids of the Earth were obtained. See Table 3.1 for example. Using different ellipsoids as the reference datum, the same geo-location on the Earth surface can be represented in different coordinate systems. See Table 3.2 for example.

Satellite images are usually geo-registered. However, different spatial reference systems can be used for such registration. It is very important to make sure that the same underlying geodetic datum and the same geographic reference system are used before utilizing such geodetic prior information.

3.2 Satellite Images

The concept of “*resolution*” in the satellite image context is multi-fold. The *spatial* resolution is defined as the pixel size of an image representing the size of the surface area (*i.e.* m²) being

Table 3.3: Configuration comparisons of recently launched commercial imaging satellites.

Satellite	Orbit				Sensor		
	Launch Date	Period (minute)	Perigee (km)	Apogee (km)	Swath (km)	Panchromatic (cm)	Multispectral (cm)
Ikonos-2	09/24/1999	98.40	678	682	11.3	82	320
QuickBird	10/18/2001	93.40	450	482	16.8	61	240
WorldView-1	09/18/2007	94.6	497	504	17.6	50	—
GeoEye-1	09/06/2008	98.33	678	693	15.2	41	165
WorldView-2	10/08/2009	100.16	772	773	16.4	46	184
Pleiades	12/17/2011	98.64	695	695	20	50	200
SPOT-7	06/30/2014	98.64	695	695	60	150	600
WorldView-3	08/13/2014	96.98	619	622	13.1	31	124
WorldView-4	11/11/2016	98.33	610	613	13.1	31	124

measured on the ground, determined by the sensors' instantaneous field of view (IFOV). The *spectral* resolution is defined by the wavelength interval size (discrete segment of the electromagnetic spectrum) and the number of intervals that the sensor is measuring. The *temporal* resolution is defined by the amount of time (*e.g.* days) that passes between imagery collection periods for a given surface location. The *radiometric* resolution is defined as the ability of an imaging system to record many levels of brightness (contrast for example) and to the effective bit-depth of the sensor (number of grayscale levels) and is typically expressed as 8-bit (0--255), 11-bit (0--2047), 12-bit (0--4095) or 16-bit (0--65,535). The *geometric* resolution refers to the satellite sensor's ability to effectively image a portion of the Earth's surface in a single pixel and is typically expressed in terms of Ground sample distance, or GSD. GSD is a term containing the overall optical and systemic noise sources and is useful for comparing how well one sensor can "see" an object on the ground within a single pixel. For example, the GSD of Landsat is $\sim 30\text{m}$, which means the smallest unit that maps to a single pixel within an image is $\sim 30\text{m} \times 30\text{m}$. The latest commercial satellite (WorldView-4) has a GSD of 0.31m.

3.3 RPC Camera Models

To be able to recover 3D geometry from multiple overlapping satellite images, we first need to formalize the satellite forming process. The camera imaging model links the 3D scene geometry and the 2D pixel observations together. Pin-hole camera models (Hartley and Zisserman, 2003) can well describe common images taken by cameras and mobile devices. Pin-hole camera models express the perspective projection by a linear system in homogeneous coordinate systems. *Epipolar* constraints (Hartley and Zisserman, 2003) well describe the geometric relationships between cameras depicting the same scene, thus enables efficient reconstruction from images using structure-from-motion (SfM) and multi-view stereo (MVS).

However, the imaging model for satellite imagery is rather different and complicated. Instead of using separate *intrinsic* parameters to describe the camera and lens properties and *extrinsic* parameters to define the camera poses, a unified model is provided by the satellite image vendors. Such models are usually provided in the form of rational polynomial functions (Dial and Grodecki, 2005; Fraser and Hanley, 2003; Grodecki and Dial, 2003; Hartley and Saxena, 1997; Hu et al., 2004). As a replacement model, the rational polynomial coefficients (RPC) model is mathematically fitted to perform the imaging process equivalently, thus barely carrying any physical interpretations. Such camera models complicate the 3D reconstruction problem because:

1. *epipolar* constraints generalize from 1D lines for pin-hole camera models to general curves for RPC models. Without a closed form solution for the general curve, one has to evaluate the expensive RPC camera model to follow the curve to establish pixel correspondences.
2. Many satellite imagery vendors only provide 3D-to-pixel mapping (also known as the forward mapping). The missing inverse (pixel-to-3D) mapping also makes reconstruction harder. Such absent inverse mapping must be fitted via either least square methods (Tao and Hu, 2002) or minimal solvers (Zheng et al., 2015).
3. Extrinsic calibrations of the RPC camera model are derived from the satellite orbital model, thus sometimes having accuracy problems. Such inaccurate extrinsic calibrations

will also propagate errors to the geo-registration described in Sec 3.1. Ozcanli et al. (2014) proposed to add a simple 2D translational compensation to the RPC model to fix such extrinsic calibration errors.

Using the rational polynomial coefficients (RPC) model to replace physical sensor models has become standard practice for satellite image processing. First introduced by Hartley and Saxena (1997), the RPC model quickly gained its popularity due to its numerical accuracy, independence of physical sensor parameters, and real-time computations (Hu et al., 2004). One common issue for RPC models is that many satellite imagery vendors only provide 3D-to-pixel mapping (also known as the forward mapping). The missing inverse (pixel-to-3D) mapping can be fitted via least square methods (Tao and Hu, 2002) or the more accurate minimal solvers (Zheng et al., 2015). Besides, overlapping satellite images are usually misaligned with respect to each other when projected onto the Earth surfaces due to sensor calibration errors. Ozcanli et al. (2014) proposed to add a simple translational compensation to the RPC model to fix such registration errors. We used the same translational compensation to calibrate the provided RPC models.

Beyond geometric 3D reconstructions, people are also interested in high-level semantic understanding of the satellite imagery, for example, building detection, land usage classification, *etc.* Large amounts of annotated data are needed to apply modern machine learning methods to achieve such goal. However, existing data annotations, such as OpenStreetMap (OpenStreetMap Foundation, 2006), needs precise geo-registration to be useful. The extrinsic calibration error from satellite sensors must be minimized in order to obtain better geo-registration. The RPC calibration process, which can reduce the calibration issue, is thus vital to enable the use of existing data annotations for machine learning tasks.

3.3.1 Forward and Inverse Model

The RPC camera model provides functionality to relate the 3D geographic coordinates (lat, lng, alt) and the image pixel coordinates (x, y) . The 3D-to-pixel model, also known as the “*forward*” model, is usually provided by most satellite imagery vendors. The pixel-to-3D model,

also known as the “*inverse*” model, is rarely provided. Through out this dissertation, we assume the forward model when we discuss RPCs unless otherwise stated.

Both the 3D geographic coordinates (lat, lng, alt) and the image pixel coordinates (x, y) are normalized to the $[-1, 1]$ range before the RPC projection process. The bounding volume (parameterized by the scale parameter $lat_s, lng_s, alt_s, x_s, y_s$ and offset parameter $lat_o, lng_o, alt_o, x_o, y_o$) is part of the RPC metadata, determined by the actual camera pose when the image is taken:

$$\tilde{lat} = \frac{lat - lat_o}{lat_s} \quad (3.5)$$

$$\tilde{lng} = \frac{lng - lng_o}{lng_s} \quad (3.6)$$

$$\tilde{alt} = \frac{alt - alt_o}{alt_s} \quad (3.7)$$

$$\tilde{x} = \frac{x - x_o}{x_s} \quad (3.8)$$

$$\tilde{y} = \frac{y - y_o}{y_s} \quad (3.9)$$

For the forward model, the normalized 3D points $(\tilde{lat}, \tilde{lng}, \tilde{alt})$ are then projected into the normalized image space by the following rational polynomial equations:

$$\tilde{x} = \frac{p_1(\tilde{lat}, \tilde{lng}, \tilde{alt})}{p_2(\tilde{lat}, \tilde{lng}, \tilde{alt})} \quad (3.10)$$

$$\tilde{y} = \frac{p_3(\tilde{lat}, \tilde{lng}, \tilde{alt})}{p_4(\tilde{lat}, \tilde{lng}, \tilde{alt})} \quad (3.11)$$

For the inverse model, the normalized 3D point location $(\tilde{lat}, \tilde{lng}, \tilde{alt})$ can be found from the normalized pixel location (\tilde{x}, \tilde{y}) , if a depth hypothesis \tilde{alt} is provided:

$$\tilde{lng} = \frac{p_5(\tilde{x}, \tilde{y}, \tilde{alt})}{p_6(\tilde{x}, \tilde{y}, \tilde{alt})} \quad (3.12)$$

$$\tilde{lat} = \frac{p_7(\tilde{x}, \tilde{y}, \tilde{alt})}{p_8(\tilde{x}, \tilde{y}, \tilde{alt})} \quad (3.13)$$

$p_1, p_2, p_3, p_4, p_5, p_6, p_7$ and p_8 are cubic polynomials of the same form with different coefficients $\mathbf{a} \in \mathbb{R}^{20}$:

$$\begin{aligned}
p(X, Y, Z) = & a_1 + a_2Y + a_3X + a_4Z + a_5YX \\
& + a_6YZ + a_7XZ + a_8Y^2 + a_9X^2 + a_{10}Z^2 \\
& + a_{11}XYZ + a_{12}Y^3 + a_{13}YX^2 + a_{14}YZ^2 + a_{15}Y^2X \\
& + a_{16}X^3 + a_{17}XZ^2 + a_{18}Y^2Z + a_{19}X^2Z + a_{20}Z^3
\end{aligned} \tag{3.14}$$

The normalized image pixel location (\tilde{x}, \tilde{y}) or 3D point location can then be re-scaled to the original space using the known scale and offset parameters.

3.3.2 Triangulation

Given multiple overlapping satellite images, triangulation is the process of finding the *best* 3D point location (lat, lng, alt) for the corresponding pixel locations $(x_1, y_1), \dots, (x_n, y_n)$ observed in multiple images. Here we aim to find the 3D point that has the minimal reprojection errors with respect to all the observed pixel locations.

The RPC projection process (Equation 3.10) can be re-written as polynomial equation systems:

$$\tilde{x} \cdot p_2(\tilde{lat}, \tilde{lng}, \tilde{alt}) - p_1(\tilde{lat}, \tilde{lng}, \tilde{alt}) = 0 \tag{3.15}$$

$$\tilde{y} \cdot p_3(\tilde{lat}, \tilde{lng}, \tilde{alt}) - p_4(\tilde{lat}, \tilde{lng}, \tilde{alt}) = 0 \tag{3.16}$$

For the triangulation problem, we know at least two pixel locations from two different images $(x_1, y_1), (x_2, y_2)$. Similarly, we can write the two RPC projection process as polynomial equation

systems:

$$\tilde{x}_1 \cdot p_{2,1}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) - p_{1,1}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) = 0 \quad (3.17)$$

$$\tilde{y}_1 \cdot p_{3,1}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) - p_{4,1}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) = 0 \quad (3.18)$$

$$\tilde{x}_2 \cdot p_{2,2}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) - p_{1,2}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) = 0 \quad (3.19)$$

$$\tilde{y}_2 \cdot p_{3,2}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) - p_{4,2}(\tilde{l}at, \tilde{l}ng, \tilde{a}lt) = 0 \quad (3.20)$$

Notice from the above polynomial equation system that, we have four known values x_1, y_1, x_2, y_2 but only three unknown variables lat, lng, alt to solve. All RPC coefficients are known and fixed. Our previous work (Zheng et al., 2015) shows that by using a Gröbner basis solver, three known values are sufficient to solve for the above triangulation problem. Thus, we can use the forth remaining pixel coordinate to validate the pixel correspondences: if the projected pixel location is very far from the ground truth forth pixel coordinate value, the pixel correspondence is highly likely bogus.

3.3.3 Bias Compensation

RPC camera models provided by most satellite imagery vendors encode both the intrinsic and extrinsic calibration parameters of the actual physical camera. Accurate sensor calibration is critical in obtaining accurate 3D reconstructions from images. Depending on the quality of satellite geopositioning sensors, most satellite images exhibit some degree of geolocation errors when projected on the ground (Ozcanli et al., 2014). Accordingly, 3D reconstructions for satellite images might fail if such registration errors are not properly corrected.

Given a normalized 3D point $(\tilde{l}at, \tilde{l}ng, \tilde{a}lt)$ and its corresponding normalized 2D pixel location (\tilde{x}, \tilde{y}) in image i , a bias compensation model $F(x, y)$ minimizes the extrinsic calibration

error and estimates the correct normalized 2D pixel location by:

$$\begin{bmatrix} \tilde{x}' \\ \tilde{y}' \end{bmatrix} = F \left(\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} \right) \quad (3.21)$$

Given ground-truth 2D correspondence locations and their corresponding 3D points, estimating the transformation F is equivalent to minimizing the 3D-to-2D reprojection error:

$$\min_{\{F, \mathbf{p}_k\}} \sum_k \sum_j F \left(\tilde{R}(P_k, \mathbf{p}_k) \right) \quad (3.22)$$

where $\mathbf{p}_k = (x_j, y_j)_k$ is the observed pixel location of the k -th normalized 3D point P_k in image j and \tilde{R} is the RPC projection error. Initial 3D point locations are obtained from the 2D feature correspondences using the above mentioned triangulation solver (Zheng et al., 2015). Equation (3.22) defines a bundle adjustment problem, which can be optimized efficiently by the Levenberg-Marquardt algorithm. We minimize the reprojection error to obtain the transformation F for each image respectively.

Different transformation models are evaluated in Table 3.4. The simple 2D translational correction model works reasonable well compared with other more complicated models. Intuitively, imaging satellites are typically far from the Earth's surface, thus the viewing rays for individual pixels are almost parallel to each other. Thus, the geopositioning errors can be corrected by adding a small translation. Please refer to Table 3.4 for details.

Compared with Ozcanli et al. (2014), our method does not require complex line patterns. In addition, by using accurate numerical solver, the bundle adjustment has a much better initialization, thus is numerically more stable and efficient to solve. The obtained translation (δ_x, δ_y) can be directly added to the RPC normalizing offsets (x_o, y_o) respectively. The translation only affects the pixel coordinate normalization, thus triangulation solver can be directly applied without modifications.

Table 3.4: Quantitative evaluation of different RPC compensation models. Reprojection error from triangulated 3D points to images observations are reported.

Model	DoF	Form	Avg-Error	Median Error	STD
Uncorrected	0	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$	0.4574	0.3296	0.7333
Translational	2	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	0.3856	0.2595	0.6642
Affine	6	$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	0.3887	0.2579	0.6073
Homography	8	$\begin{bmatrix} \omega x' \\ \omega y' \\ \omega \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	0.3898	0.2579	0.5987

3.4 Radiometric Calibration

Satellite images are typically collected at different times of day, different seasons, and different altitudes, all of which can lead to significant changes in the imaged illumination conditions. We apply the following per-pixel calibration to transform the radiance value to the top-of-atmosphere (TOA) reflectance value to correct for such radiometric changes:

$$\rho_{\lambda_{pixel},band} = \frac{L_{\lambda_{pixel},band} \cdot d_{ES}^2 \cdot \pi}{E_{sun_{\lambda_{band}}} \cdot \cos(\theta_s)} \quad (3.23)$$

In Equation (3.23), the TOA reflectance per pixel per band $\rho_{\lambda_{pixel},band}$ is unknown. $L_{\lambda_{pixel},band}$ is the band-averaged radiance. The earth-sun distance d_{ES} , the band-averaged solar spectral irradiance $E_{sun_{\lambda_{band}}}$, and the solar zenith angle θ_s can be obtained from metadata. After the pixel-wise radiometric correction, pixel values fall into the range of $[-1, 1]$. We further normalize the pixel values to $[0, 1]$.

Radiometric information from the metadata of satellite images may also contains errors. Correcting such radiometric metadata is beyond the scope of this thesis.

CHAPTER 4: GEOMETRIC RECONSTRUCTION VIA EDGE-AWARE INTERPOLATION

4.1 Introduction

As discussed in Chapter 1, many applications of satellite imagery require the accurate relationship between different images, or 2D image observations and underlying 3D scene geometry, to be known. Multi-view stereo (MVS) techniques (Furukawa et al., 2015) estimate the most likely 3D geometry given multiple image observations, thus are ideal to establish such dense correspondences.

However, estimating multi-view stereo correspondences on satellite imagery is particularly challenging, as described in Chapter 1, for the following reasons: (i) the enormous absolute pixel count; (ii) the low ground sampling rate; (iii) radiometric changes across images; (iv) complicated sensor imaging models; (v) inaccurate sensor calibration.

In this chapter, we propose a complete 3D reconstruction pipeline for satellite images that addresses the above challenges. An illustration of our proposed pipeline can be found in Figure 4.1. Our novelties include:

1. We utilize coarse geo-registration to guide the 2D feature matching to establish initial correspondences across images.
2. We utilize accurate minimal solvers (Zheng et al., 2015) and sparse feature matches to refine satellite sensor calibrations.

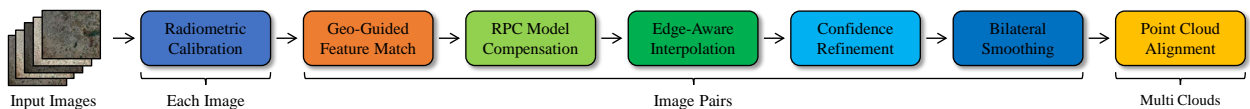


Figure 4.1: Overview of the satellite stereo pipeline.

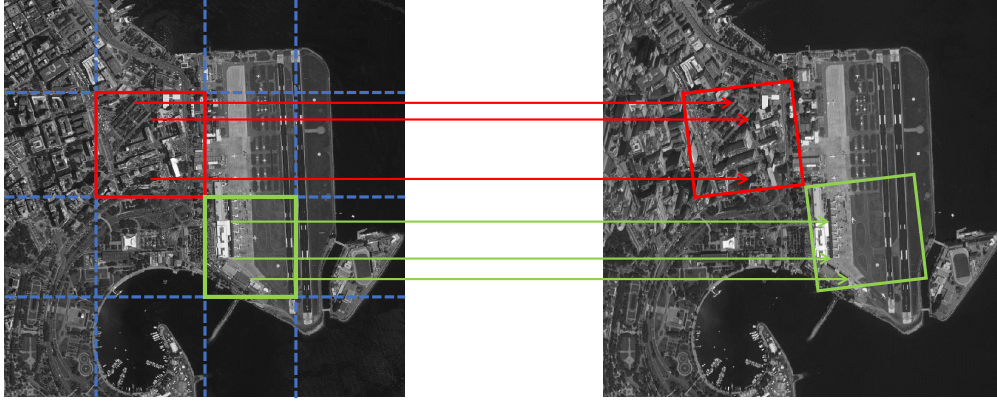


Figure 4.2: Illustration for geo-guided feature matches. For the given image pair, we divide each image into blocks. For image features detected in the red block of the left image, we only match them to the features detected in the red block of the right image.

3. We estimate dense correspondences by interpolating refined sparse matches, thus eliminating the need for per-pixel correspondence search.
4. We further improve the dense height map quality via probabilistic confidence modeling and efficient bilateral smoothing.

Given a set of geographically overlapping satellite images, our goal is to establish pixel-wise dense correspondences between image pairs and obtain 3D point clouds from such correspondences. We first obtain reliable sparse feature matches and initial triangulated 3D points (Sec 4.2). Geographical priors are utilized to accelerate the feature matching process. The satellite camera calibration biases are further refined by minimizing the reprojection between the triangulated 3D points and the feature locations (please refer to Chapter 3 for more details). Sparse matches are then interpolated into pixel-wise dense correspondences via edge-aware interpolation (Sec 4.3). A probabilistic model is built to evaluate the quality of such preliminary dense correspondences (Section 4.4). We next apply bilateral smoothing to refine low-confidence matches (Section 4.5). Finally, multiple point clouds obtained from the different image pairs are further merged and aligned (Section 4.6). The proposed pipeline is illustrated in Figure 4.1.

4.2 Geo-guided Sparse Feature Matching

Despite the fact that satellite images exhibit geospatial registration errors (Ozcanli et al., 2014), satellite images are roughly geo-registered to a common geodetic coordinate frame. For each image, an affine transformation $\mathbf{A} \in \mathbb{R}^{2 \times 3}$ maps pixel locations $\mathbf{p} = (x, y)$ into geodetic locations $[long, lat]^T = \mathbf{A}[x, y, 1]^T$. The affine transformation \mathbf{A} can be obtained either from image metadata or RPC models. Such coarse geo-registration can be applied as regularization for 2D feature matching: feature points should only match to feature points that are close in the geodetic coordinate frame. To apply such regularization efficiently, we first divide the reference image into blocks. For each block, we map the block corner pixel location \mathbf{p} into the matching image by $\mathbf{p}' = \mathbf{A}_{match}^{-1} \mathbf{A}_{ref} \mathbf{p}^1$. Sparse feature matching is then only performed within corresponding image blocks. An illustration of geo-guided feature matching can be found in Figure 4.2. An example of inlier matches and associated 3D points can be found in Figure 4.3.

Our geo-guided feature matching method is superior to regular exhaustive feature matching for an image pair:

1. Block-wise feature matching is much more efficient. Given N features and m blocks, exhaustive feature match has a time complexity of $O(N^2)$ while block-wise matching only has a time complexity of $O(N^2/m)$.
2. By reducing images into blocks, we regularize the matching process with the prior geospatial knowledge, which can reduce outlier matches.
3. Blocks are independent of each other, thus block-wise matching can be fully parallelized to increase computational efficiency.
4. With block-wise feature extraction, we can obtain more spatially uniform-distributed features. Depending on the visual content, some blocks may contain more visual features while other blocks may contain fewer visual features. The sorting or non-maximum sup-

¹ Augment \mathbf{A} with $[0, 0, 1]$ to compute the inverse matrix.

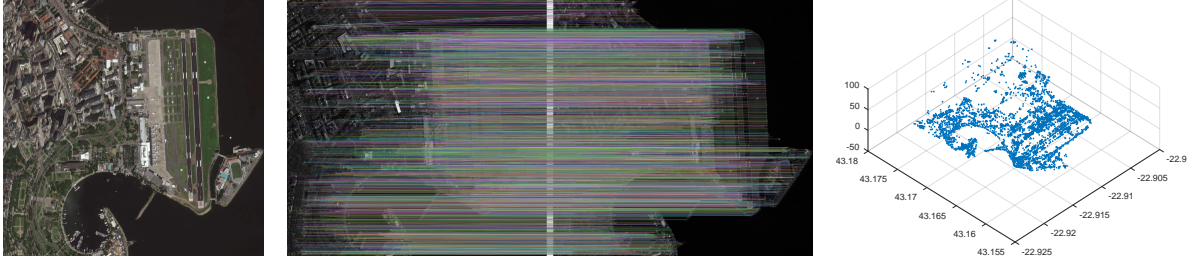


Figure 4.3: Example of geo-guided feature matching. From left to right: reference image, inlier matches, triangulated 3D point cloud.

pression during the global image feature extraction process will favor more salient/distinct visual features across the whole image. However, if extraction is done individually in each block, less distinct features in some blocks won't be suppressed by stronger features in other blocks. Having features covering more image regions is important for the later interpolation process. Notice this usually lead to more features being detected across the entire image, thus not contradicting the algorithm complexity mentioned above.

Within each block-pair, SIFT (Lowe, 2004) keypoints are detected on a regular grid and SIFT (Lowe, 2004) descriptors are extracted. We enforce the left-right consistency and apply the ratio test for the feature matching process. For a putative feature match (x_1, y_1) and (x_2, y_2) , the triangulation minimal solver by Zheng et al. (2015) uses three pixel coordinates (for example, x_1, y_1, x_2) to triangulate the 3D point in space (lat, lng, alt) . The remaining coordinate y_2 is used for numerical verification to eliminate outlier matches. Considering the potential geo-location error of satellite RPC models, we use a loose threshold for outlier removal. Thus, we can further prune outlier matches and obtain a set of reliable feature matches and their corresponding 3D points within each image block.

4.3 Edge-Aware Interpolation

Having obtained a set of matches $\mathcal{M} = \{(\mathbf{p}_m, \mathbf{p}'_m)\}$ between image I and I' , we now estimate a pixel-wise dense correspondence field CF . For each matched pixel pair $(\mathbf{p}_m, \mathbf{p}'_m)$, we first use

the reprojection error $\tilde{R}(F, P, \mathbf{p})$ to model the confidence for both matching pixels:

$$C(\mathbf{p}_m) = \exp\left(-\frac{\tilde{R}(F, P, \mathbf{p})}{\sigma_R}\right) \quad (4.1)$$

where F is the bias compensation model (see Section 3.3.3), P is the 3D point, and σ_R is the weighting parameter.

For a pixel $\mathbf{p} \in I$, we interpolate its correspondence pixel $CF(\mathbf{p}) \in I'$ using the Nadaraya-Watson (NW) estimator (Revaud et al., 2015):

$$CF(\mathbf{p}) = \frac{\sum_{\mathbf{p}_m \in \mathcal{N}_K(\mathbf{p})} C(\mathbf{p}_m) \cdot k_D(\mathbf{p}_m, \mathbf{p}) \cdot \mathbf{p}_m'}{\sum_{\mathbf{p}_m \in \mathcal{N}_K(\mathbf{p})} C(\mathbf{p}_m) \cdot k_D(\mathbf{p}_m, \mathbf{p})} \quad (4.2)$$

where $k_D(\mathbf{p}_m, \mathbf{p}) = \exp(-a \cdot D(\mathbf{p}_m, \mathbf{p}))$, D is the distance between two pixels, and $\mathcal{N}_K(\mathbf{p})$ represents the K nearest neighbors with respect to the distance D .

Using Euclidean distances between two pixels \mathbf{p} and \mathbf{q} in Equation (4.2), we simply interpolate the correspondence for un-matched pixels based on the spatial location of the established feature matches. However, such a strategy does not respect image boundaries and depth discontinuities. Similar to Revaud et al. (2015), we used the geodesic distance $D_G(\mathbf{p}, \mathbf{q})$, *i.e.* the shortest distance between pixel \mathbf{p} and pixel \mathbf{q} with respect to image edge responses E . A pixel belongs to the same object is close to all other pixels within the same object defined by the image edges, but far away from pixels across object boundaries. Thus the interpolation will respect image edges and depth discontinuities.

Edge responses are computed using the structured edge detector (SED) (Dollár and Zitnick, 2013). The approximated geodesic distances $D_G(p, q)$ are used to improve computational efficiency. For each correspondence pair $(\mathbf{p}, \mathbf{p}') \in \mathcal{M}$, we first compute the flow vector $(u_{\mathbf{p}}, v_{\mathbf{p}}) = (x_{\mathbf{p}'} - x_{\mathbf{p}}, y_{\mathbf{p}'} - y_{\mathbf{p}})$. We then apply the Nadaraya-Watson estimator to interpolate the sparse flow vectors into a dense per-pixel flow field. An example of edge-aware interpolation can be found in Figure 4.4.



Figure 4.4: Illustration of edge-aware interpolation. From left to right: input image, edge response, interpolated flow field, triangulated height.

4.4 Confidence Refinement

Multi-view stereo computation is usually expensive for high-resolution images. With higher image resolutions, not only absolute pixel count increases, potential disparity search space for each pixel also increases, all of which incurs higher computational complexities. Efforts have been made to directly reduce the disparity search space (Wang et al., 2014b), or to use sampling and propagation to mitigate the computational overhead (Wang et al., 2016b). On the contrary, our proposed method avoids the efforts to sample or search for each pixel. By using edge-aware interpolation, the dense correspondence problem is reduced to efficient sparse feature matching and light-weight interpolation for individual pixels.

However, the quality of the interpolated dense correspondence field CF heavily depends on the sparse feature matching results. We adopted multiple constraints in the previous stages of our proposed algorithm to increase the quality of the initial sparse feature matches:

1. We constrain initial feature matches to reside in geospatially local regions rather than the entire image, which helps to reduce bogus matches.
2. We perform bundle adjustment to correct the positioning errors of the RPC camera calibrations. The refined camera calibration is used to re-verify putative feature matches for outlier removal.
3. Unlike the planar homography assumption used in (Bosch et al., 2016), we use the accurate RPC triangulation minimal solver to prune outlier matches (Zheng et al., 2015).

For traditional pinhole camera models, epipolar constraints reduce the MVS search space from a general 2D field to a 1D epipolar line. But RPC camera models do not have such well-defined geometric constraints. With less regularization, the interpolated correspondence field is more error-prone to photo-consistency ambiguities. Thus we propose a probabilistic model to lessen the impact of potentially bogus pixel flows.

For interpolated pixels, the confidence of their interpolated flow vectors can also be defined with the Nadaraya-Watson estimator:

$$C(\mathbf{p}) = \frac{\sum_{\mathcal{N}_K(\mathbf{p})} k_D(\mathbf{p}_m, \mathbf{p}) \cdot C(\mathbf{p}_m)}{\sum_{\mathcal{N}_K(\mathbf{p})} k_D(\mathbf{p}_m, \mathbf{p})} \quad (4.3)$$

Given a reference image I and a matching image I' , we compute the “forward” flow $\{U_f, V_f\} : I \rightarrow I'$ and “backward” flow $\{U_b, V_b\} : I' \rightarrow I$ simultaneously. For correct correspondences, if a pixel \mathbf{p} in image I maps to the pixel \mathbf{p}' in the image I' , we would expect the pixel \mathbf{p}' in the image I' to map to the pixel \mathbf{p} in the image I in the backward flow field. Otherwise, the disagreement between forward flow field and backward flow field indicates erroneous correspondences. We penalize such inconsistencies by:

$$\begin{aligned} a(\mathbf{p}) &= [U_f(\mathbf{p}) + U_b(x_{\mathbf{p}} + U_f(\mathbf{p}))]^2 \\ &\quad + [V_f(\mathbf{p}) + V_b(y_{\mathbf{p}} + V_f(\mathbf{p}))]^2 \\ C(\mathbf{p}) &\leftarrow C(\mathbf{p}) \cdot \exp\left(-\frac{a(\mathbf{p})}{\sigma_{sym}}\right) \end{aligned} \quad (4.4)$$

where $a(\mathbf{p})$ is the squared residual between forward flow and backward flow vectors of pixel \mathbf{p} and $\sigma_{sym} = 4$.

Homogeneous textures like water surfaces are generally hard to match. Compared with traditional RGB images, extra channels, if multispectral images are available, can provide information to identify such regions. We compute the normalized difference vegetation index (NDVI) (Weier

and Herring, 2000) to identify homogeneous surfaces:

$$N(\mathbf{p}) = \frac{NIR(\mathbf{p}) - Red(\mathbf{p})}{NIR(\mathbf{p}) + Red(\mathbf{p})} \quad (4.5)$$

where Red and NIR stand for the spectral reflectance measurements acquired in the visible red and the near-infrared channel. NDVI values fall into the range of $[-1.0, 1.0]$. Negative values correspond to water, values close to zero ($[-0.1, 0.1]$) correspond to barren surfaces, for example, rocks, and, or snow. Thus, we further decrease the matching confidence of pixel \mathbf{p} if its NDVI value is smaller than 0.1:

$$C(\mathbf{p}) \leftarrow C(\mathbf{p}) \cdot \exp\left(-\frac{N(\mathbf{p}) - 0.1}{\sigma_{ndvi}}\right) \quad \text{if } N(\mathbf{p}) \leq 0.1 \quad (4.6)$$

where $\sigma_{ndvi} = 10.0$.

4.5 Bilateral Refinement

Given a pair of satellite images (I, I') , we now have estimated a flow field (U, V) from I to I' , as well as a per-pixel confidence measure C . The flow field can contain incorrect correspondences, as indicated by asymmetric flow, or ambiguous matches on homogeneous regions. We minimize the following objective by the bilateral solver (Barron and Poole, 2016) to produce a smooth flow field that respects both image edges and highly confident matches:

$$\min_{u_i, v_j} \frac{\lambda}{2} \sum_{i,j} \hat{W}_{i,j} \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \begin{bmatrix} u_j \\ v_j \end{bmatrix} \right\|_2^2 + \sum_i \hat{c}_i \left\| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} \right\|_2^2 \quad (4.7)$$

where $\hat{u}_i, \hat{v}_i, \hat{c}_i$ are the flow and confidence estimates at pixel i , i and j are neighboring pixels, and u_i, v_i are the smoothed flow field. The bistochasized bilateral affinity matrix W is constructed as in (Barron and Poole, 2016) with neighborhood luma, chroma, and pixel location in consideration.

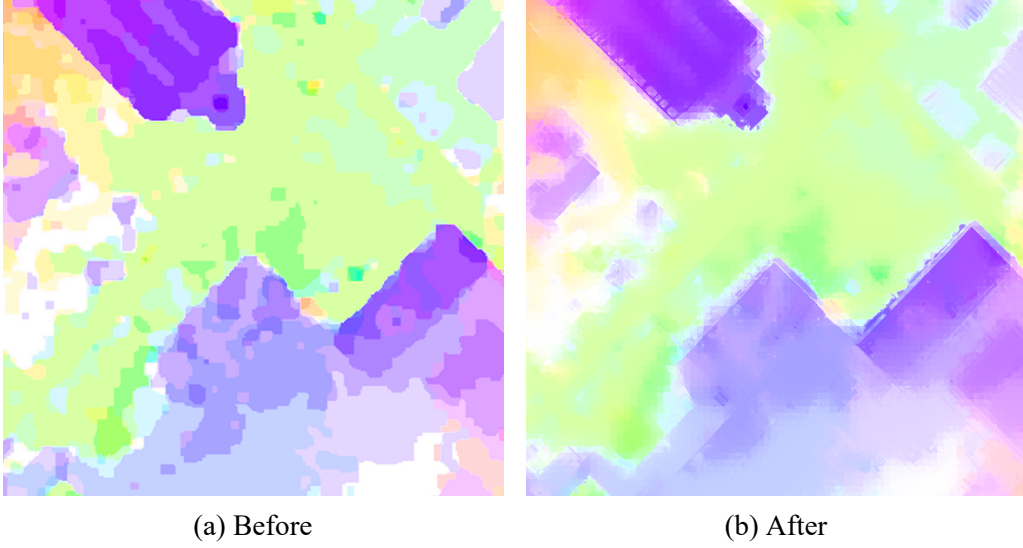


Figure 4.5: Refinement of flow field using bilateral solver.

A bistochasized bilateral affinity matrix W is constructed as follows to build the smoothness term in the smoothing objective in Equation (4.7):

$$W_{i,j} = \exp \left(-\frac{\|p_i^l - p_j^l\|_2^2}{2\sigma_l^2} - \frac{\left\| \begin{bmatrix} p_i^u \\ p_i^v \end{bmatrix} - \begin{bmatrix} p_j^u \\ p_j^v \end{bmatrix} \right\|_2^2}{2\sigma_{uv}^2} - \frac{\left\| \begin{bmatrix} p_i^x \\ p_i^y \end{bmatrix} - \begin{bmatrix} p_j^x \\ p_j^y \end{bmatrix} \right\|_2^2}{2\sigma_{xy}^2} \right) \quad (4.8)$$

where for pixel i , p_i^l is luma, (p_i^u, p_i^v) is chroma, (p_i^x, p_i^y) is spatial pixel location. $\sigma_l = 0.0625$, $\sigma_{uv} = 0.03$, $\sigma_{xy} = 12$ determines the support size of the solver.

With the smoothed flow field, we can estimate a point cloud for reference image I : the 3D point (lat, lng, alt) for pixel p can be triangulated from the smoothed correspondences $[(p_x, p_y), (p_x + u_p, p_y + v_p)]$ using the triangulation minimal solver (Zheng et al., 2015).

4.6 Point Cloud Alignment

Our proposed algorithm estimates a dense pixel-wise 3D reconstruction for a given satellite image pair. Given a collection of overlapping satellite images $\mathcal{I} = \{I_0, I_1, \dots, I_N\}$, there exist

Table 4.1: Quantitative Evaluation of Satellite Image Stereo

Method	Completeness	Registration	Accuracy	Time
	%	Meters	Meters	Minutes
SiftFlow (Bosch et al., 2016)	72.3	0.322	2.689	57.8
S2P (De Franchis et al., 2014)	79.0	0.278	2.727	34.9
PMBP (Wang et al., 2016b)	74.1	0.479	3.819	45.7
SGM (Hirschmuller, 2008)	67.4	0.492	2.6	97.6
Ours	79.3	0.25	2.57	27.5

$O(N^2)$ image pairs. Considering the large image resolutions of satellite images, it is computationally prohibitive to compute dense 3D reconstructions exhaustively for each image pair.

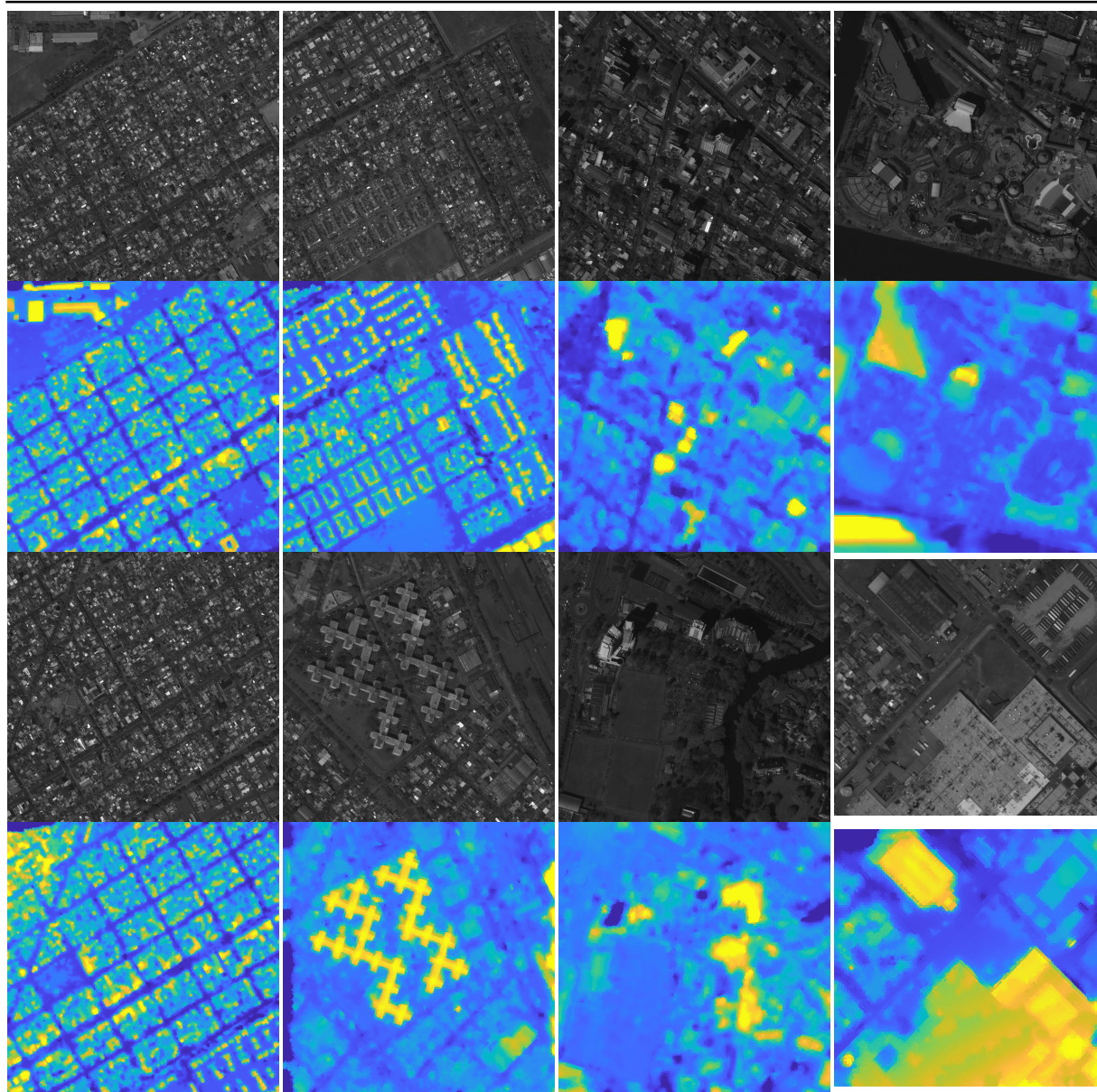
To increase the scalability of our proposed algorithm, we propose to select only k best matching views for each image. Matching views are selected by the RPC reprojection error defined in Equation (3.22). Thus for an image collection with N images, we only need to compute kN pairwise reconstructions. For each reference view, the resulting k dense height maps are fused together and a median filter is applied at each reference pixel location to reject outliers.

We then align the N separate point clouds together. Our RPC sensor compensation only performs pair-wise sensor alignment. To account for the possible sensor misalignment between different image pairs, we compute a rigid transformation (rotation $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ and translation $\mathbf{t} \in \mathbb{R}^3$) to align new point clouds to existing 3D models using iterative closest point (ICP). We further refine the fused 3D point cloud by a radius-based outlier filter: any 3D points that don't have at least 10 neighbors within 1.5 meters are discarded.

4.7 Experiments

In this section, we evaluate the performance of our proposed algorithm, and compare our results with state-of-the-art baselines.

Figure 4.6: Visualizations of satellite imagery and their computed height map.



4.7.1 Evaluation

Our experiments were performed on the JHU/APL satellite MVS benchmark dataset (Bosch et al., 2016). The benchmark dataset contains 50 WorldView-3 images covering San Fernando, Argentina. We only use the panchromatic images with 30-cm ground sampling distances (GSD) for stereo reconstructions. Multi-spectral images with 1.3-meter GSD is upsampled to compute NDVI only. Airborne Lidar was used as ground-truth for evaluation.

To evaluate the reconstructed MVS point cloud against ground-truth Lidar, the input point cloud is first registered to the ground truth point cloud by estimating a simple x-y shift. 3D points from the input point cloud are placed into cells of a predefined regular grid based on their lat/long coordinates. The accuracy of an input point cloud is then defined as the root mean squared error (RMSE) over all non-empty cells. The completeness for an input point cloud is the fraction of non-empty input cells with height errors less than 1-meter. The registration error is the median value of the per-cell height error after alignment.

We compare our proposed algorithm with several state-of-the-art baseline methods. Detailed results can be found in Table 4.1. Our proposed algorithm exceeded the baseline methods in both completeness and accuracy. Qualitative examples of our results can be found in Figure 4.6.

Our choice of using image feature matching to bootstrap the correspondence estimation process is critical to the overall good performance. Compared with traditional photo-consistency metrics, for example, SSD, NCC, and census transform, image descriptors like SIFT have better discriminating power. In addition, local feature descriptors are matched without considering geometric constraints, thus are more robust to initial sensor calibration errors.

4.7.2 Runtime

We implemented our proposed algorithm in C++. Runtime statistics are collected on a machine with a 2.4GHz 4-core Intel Xeon CPU. Comparison of the computation efficiency of our method against baseline methods can also be found in Table 4.1. The runtime is collected on an image pair of $25K \times 20K$ pixel resolution.

Table 4.2: Runtime statistics for each stage of the stereo pipeline.

Stage	Time (seconds)	Parallelizable
Radiometric Correction	92.34	✓
Feature Match	223.78	✓
RPC Compensation	249.32	
Interpolation	267.79	✓
Confidence Refinement	156.98	✓
Bilateral Smoothing	674.08	

Our method has better computation efficiency compared to the baseline methods. MVS computation is heavy because a large number of depth hypotheses needs to be tested for each pixel. By exploiting the local smoothness properties in local image regions, our proposed method, however, totally discarded the pixel-wise search/sampling process, and relies on light-weight interpolation and edge-aware smoothing to obtain high-quality geometry. Thus our method demonstrated great computational efficiency.

We further list the runtime for each stage of our proposed algorithm in Table 4.2. Most stages of our method are implemented and executed in parallel. To summarize, our method is very efficient both in theory and in practice.

4.7.3 Ablation Study

Our algorithm consists of multiple processing stages. In this section, we analyze the contributions of each stage. We use the same completeness and accuracy metric to evaluate the point cloud reconstructed with individual stages removed from the pipeline if possible. We replace the geo-guided feature matching stage with a normal feature matching stage.

Detailed comparisons can be found in Table 4.3. Removing radiometric calibration caused the slightest performance degrade when compared with other processing stages. One reason is the radiometric robustness of local image features. Also, the interpolation of matching feature points is computed with respect to the reference image only. Compared to Wang et al. (2016b),

Table 4.3: Ablation study for individual stages of the stereo pipeline.

Removed Stage	Completeness (%)	Accuracy (m)
Radiometric Correction	78.2	2.93
Feature Match	72.1	3.05
RPC Compensation	68.7	3.89
Confidence Refinement	73.3	3.12
Bilateral Smoothing	69.1	3.56
Full Pipeline	79.3	2.57

Table 4.4: Local image feature comparison for geo-guided feature matching

Feature	Match Number	Completeness	Accuracy
Unit	%	%	Meters
SIFT	100	79.3	2.57
Root-SIFT	102.8	79.4	2.57
A-SIFT	105.8	79.6	2.42
MODS	110.7	80.1	2.39
BRISK	47.3	59.1	3.19
ORB	57.7	64.6	2.98
FREAK	53.1	63.1	3.07

where removing radiometric calibration caused the completeness to drop from 74.1% to 68.7%, our algorithm is fairly robust to radiometric changes.

Replacing geo-guided feature matching with normal image feature matching caused a noticeable drop in completeness. Fewer features are detected in less salient image regions. Without enough accurate initial matches, interpolation in such areas will lead to erroneous estimations.

Removing the RPC camera model compensation led to the worst performance decrease. Inaccurate sensor calibration can cause errors when using minimal solvers for outlier removal and triangulation, jeopardizing all later processing.

Confidence refinement mainly contributed to improving the accuracy by suppressing occlusions or ambiguous matches in texture-less regions. Bilateral smoothing helps to improve both the completeness and accuracy by filling “holes” and low-confidence regions with smooth values.

4.8 Discussion

4.8.1 Feature Comparison

We further evaluate different image local features for our sparse correspondence stage. Detailed results can be found in Table 4.4. Binary features (BRISK, ORB, FREAK) showed inferior performances compared to SIFT. Root-SIFT gets slightly more initial matches but led to similar completeness and accuracy. A-SIFT and MODS (Mishkin et al., 2015) generated higher initial match counts and slight improvements in final completeness and accuracy metric, at the cost of higher computational overhead. A-SIFT or MODS can be used as a replacement if needed.

We also noticed that images coming from the same satellite usually have the same ground sampling distance. Fixed ground sampling rates mean object scales do not change dramatically across images. If needed, scale-invariance constraints can be loosened when computing image feature descriptors, for example, using fewer octave layers in SIFT descriptor extraction.

4.8.2 Occlusion Handling

Both the interpolation and bilateral smoothing stages respect image boundaries, thus preventing the foreground geometries from bleeding into the background. In addition, asymmetric flow described in Equation (4.4) also indicates geometric occlusion and lower the confidence for occluded regions. The bilateral solver will then fill in the smoothed background geometry.

4.8.3 Future Work

Learning-based feature representations have recently become popular (Yi et al., 2016). We leave it as future work to apply such learned features to our proposed satellite MVS pipeline. In addition, our current RPC correction procedure is applied to each image pair. Joint RPC correction proposed by (Ozcanli et al., 2014) can be used to align multiple satellite images together. We leave as future work to compare the joint correction approach and our current ICP alignment approach.

CHAPTER 5: JOINT GEOMETRIC AND SEMANTIC RECONSTRUCTION

5.1 Introduction

Multi-view stereo estimation task obtains 3D structures by minimizing the inconsistency of visual appearances of the hypothesizing structures. However, certain photometric ambiguities cannot be easily resolved, even with larger spatial context, for example, homogeneous regions and repetitive textures.

On the other hand, land use classification on satellite images tries to infer semantic categories on a pixel-wise basis. Without consideration of spatial smoothness and context, such pixel-wise labeling can be noisy. But if dense correspondences can be established across multiple satellite images, both geometric and semantic ambiguities can be jointly resolved.

Thus, given no less than two satellite image observations for the region of interest, our goal is to establish dense correspondences across the reference view image and all the rest matching images, so that geometry and semantic information can be reliably extracted on top of such correspondences.

In this chapter, we propose an efficient dense multiple view stereo algorithm for satellite images, which solves the geometry and semantic estimation problem directly in the 3D space (see Figure 5.1). Utilizing only the provided space-to-image mapping, our method effectively bypasses the limitation of a missing or inaccurate image-to-space mapping. By using an efficient inference framework, our proposed method can achieve state-of-the-art results in significantly less time. To summarize, our innovations are:

1. We propose to solve the dense stereo problem directly in a 2.5D volumetric representation of the 3D space, effectively by-passing the limitation of missing/inaccurate image-to-space model.

2. We propose an efficient PatchMatch based inference framework to address the high-resolution dense satellite stereo problem.
3. We demonstrate that utilizing semantic land usage information can help to improve the accuracy of dense stereo in satellite problems

5.2 Local 3D Plane Formulation

We propose to jointly estimate the geometry structure and land usage information for a geographical region observed by multiple satellite images. Such a geographical region of interest is represented by an uniform vector grid $\{\mathbf{u}\}_{s=1}^n$. Each node corresponds to a geographical point in 3D space, with known latitude lat and longitude lng , but unknown altitude alt . Following the slanted surface formulation proposed in Bleyer et al. (2011), we associate an unknown local 3D plane $\mathbf{f} = (a_f, b_f, c_f)$ to each geographical point (lat, lng, alt) . The altitude of the 3D point satisfies:

$$alt = a_f * lat + b_f * lng + c_f \quad (5.1)$$

Once the 3D planes are estimated, 3D points lying on the plane can be projected onto a specific satellite image I by using the image specific RPC camera models (see Figure 5.1 for example). In addition to the geometry 3D plane, each node \mathbf{u}_s also encodes land usage information for the given geographical 3D point.

Under our formulation, our goal of extracting geometry structure can be achieved by finding the optimal parameterization for each of the vector grid points that minimizes photometric and semantic observation conflicts between the designated reference view r and matching views, while maintaining local smoothness in the underlying geometry structures and semantic representations. Pixel-level view selection has been proved successful in increasing stereo accuracy and robustness by Zheng et al. (2014). In order to account for the drastic view and capture condition changes across multiple satellite images, we include a similar pixel-level view selection scheme in our formulation.

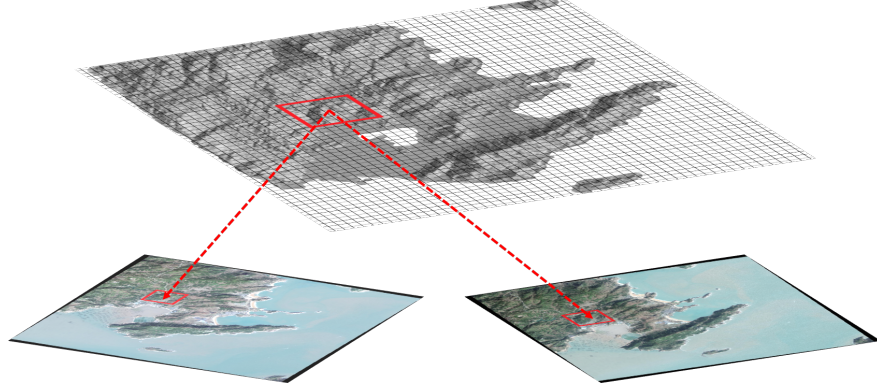


Figure 5.1: Visualization of the grid representation of our proposed method. Each grid point is parameterized by a local 3D plane and land usage information. Photo consistencies are accumulated across multiple images.

The state of each grid point $\mathbf{u}_s = (\mathbf{f}_s, v_s, l_s)$ encodes information for an unknown local 3D plane $\mathbf{f}_s \in \mathbb{R}^3$, a unknown matching view index v_s , and an unknown land usage categories label l_s . Thus our goal is to optimize the following energy function:

$$E(\mathbf{u}_1, \dots, \mathbf{u}_n) = \sum_{s=1}^n \psi_s(\mathbf{u}_s) + \beta_1 \sum_{s=1}^n \left[\sum_{t \in N(s)} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) \right] \quad (5.2)$$

where $N(s)$ being the pairwise neighborhood of the node s . The unary energy $\psi_s(\mathbf{u}_s)$ measures consistency of image appearance and semantic labeling of the associated 3D plane between different image views. To compute the unary energy for a node \mathbf{u}_s , we first collect a set of nearby 3D points P_s lying on its local 3D plane. Each 3D point $p_s \in P_s$, is projected onto the reference view r and the selected match view v_s . The obtained image pixel color/intensity (I_r, I_{v_s}) and image pixel labeling (L_r, L_{v_s}) are compared to build the following cost:

$$C(p_s) = \min(\tau_c, \|I_r - I_{v_s}\|) + \beta_2 \delta(L_r, L_{v_s}) \quad (5.3)$$

where τ_c is truncation threshold for stereo cost, $\|I_r - I_{v_s}\|$ is the L_1 color difference, and δ is a Kronecker delta function defined as:

$$\delta(L_r, L_{v_s}) = \begin{cases} 0 & \text{if } L_r = L_{v_s} \\ 1 & \text{if } L_r \neq L_{v_s} \end{cases} \quad (5.4)$$

The unary data term is then defined over the 3D point set P_s :

$$\psi_s(\mathbf{u}_s) = \sum_{p_s \in P_s} \omega(p, p_s) [C(p_s) + \gamma \mathcal{L}(l_s)] \quad (5.5)$$

where $\omega(p, p_s)$ is the adaptive weight between the center point p and the neighboring 3D point p_s (Yoon and Kweon, 2006), and $\mathcal{L}(l_s)$ is the cost of classifying node \mathbf{u}_s as label l_s , which can be obtained through classifier output (see Section 5.3).

On the other hand, the pairwise term $\psi_{st}(\mathbf{u}_s, \mathbf{u}_t)$ explicitly considers smoothness between adjacent 3D planes, matching view selections, and semantic label assignment:

$$\begin{aligned} \psi_{st}(\mathbf{u}_s, \mathbf{u}_t) = & (|\mathbf{n}_s \cdot (\mathbf{x}_s - \mathbf{x}_t)| + |\mathbf{n}_t \cdot (\mathbf{x}_t - \mathbf{x}_s)|) \\ & + \omega_1 \delta(v_s, v_t) + \omega_2 \delta(l_s, l_t) \end{aligned} \quad (5.6)$$

where \mathbf{n}_s represents the unit normal vector of plane \mathbf{f}_s , and \mathbf{x}_s is a point on the plane \mathbf{f}_s . The smoothness term $\psi_{st}(\mathbf{u}_s, \mathbf{u}_t)$ will have a zero cost value if and only if two nodes lie on the same plane, have the same semantic label and match view selection. In our experiments, only selecting best view can better handle occlusion. In case multiple equally good views exist for a given MRF node, the matching view that's more consistent with adjacent nodes would be selected via MRF inference.

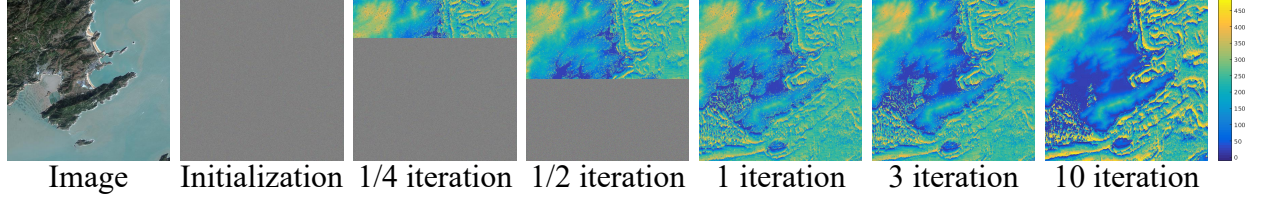


Figure 5.2: Qualitative illustration of convergence. We show the height map obtained at different stages of the optimization process. The first iteration already extracted coarse geometrical structures from the random initializations. Results obtained after 3 iterations are visually very close to the final results with 10 iterations.

5.3 Land Use Classification

Captured at multiple spectral bands, pixel values from satellite images naturally carry physical response information from different land types, thus providing strong clues on land usage information. Thus, we propose to use a simple three-layer fully-connected neural-network to perform pixel-wise land usage classification based on pixel characteristics.

The land usage classification neural network consists of an input layer with eight neurons to take input from the eight channel pixel values, forty neurons for the hidden layer, and 13 output neurons in the output layer. The neural network output is inverted and serves as the labeling cost term $\mathcal{L}(l_s)$ in the unary energy function (see Equation 5.5).

5.4 Inference

Our proposed MRF formulation in Equation 5.2 can be efficiently solved using the Patch-Match Belief Propagation method (PBMP) (Besse et al., 2013). We initialize each node \mathbf{u}_s with a random plane \mathbf{f}_s , a random semantic label l_s , and a random match view choice v_s . MRF is then inferred through spatial-propagation and local resampling. We empirically set the number of particles at each node \mathbf{u}_s to 3, which achieved a good balance between computation overhead and converging speed.

We show a quantitative illustration of inference process in Figure 5.3, and a qualitative visualization in Figure 5.2.

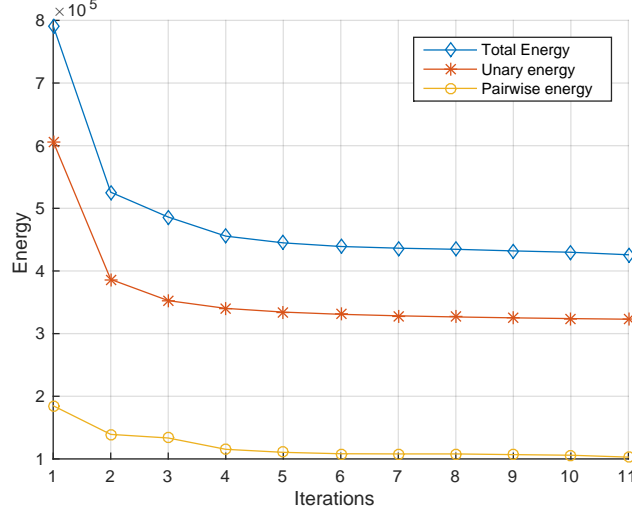


Figure 5.3: An example of PMBP inference on the Xiapu dataset (See Table 5.5 for dataset details). PMBP solver shows fast convergence speed in practice.

5.5 Experiments

5.5.1 Implementation

We evaluate our proposed method in a parallel C++ implementation. All the benchmark numbers are collected on a 32 core Intel Xeon CPU running at 2GHz. We use 5×5 patches for stereo matching. PMBP solver is run for three iterations with three particles at each node. Parameters are set as $\beta_1 = 0.4$, $\beta_2 = 15.0$, $\tau_c = 30.0$, $\omega_1 = 0.2$, $\omega_2 = 0.2$, $\lambda = 20.0$.

5.5.2 Ground Level Stereo Experiments

To justify the incorporation of semantic information in dense stereo problems, we quantitatively evaluate our proposed formulation of joint dense stereo and semantic object classification on the Leuven dataset (Ladický et al., 2011) and the Kitti dataset (Geiger et al., 2012). Both datasets provide rectified stereo image pairs, simplifying the correspondence search space to horizontal lines. Accordingly, we define a corresponding grid node over each pixel in the reference image.

On both datasets, we compare our method against one of the state-of-art joint estimation framework, Automatic Labelling Environment (ALE) (Ladický et al., 2011). Especially, we used

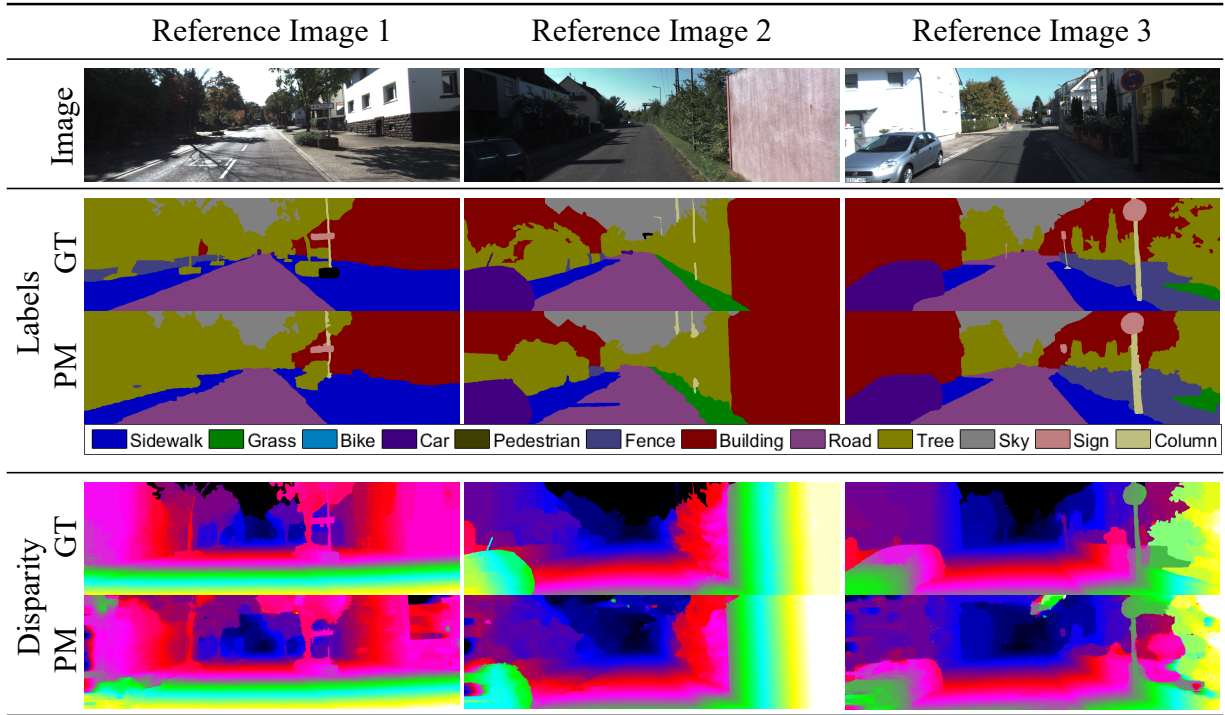


Figure 5.4: Visualization of semantic and stereo results of our proposed method on the Kitti dataset (Geiger et al., 2012). Through joint-optimization, clear boundaries are preserved in semantic segmentations, while stereo ambiguity can be resolved through high-order semantic information.

Table 5.1: Stereo accuracy on Leuven dataset. Out-All: percentage of erroneous pixels in total; Avg-All: average disparity in total.

	ALE (Ladický et al., 2011)	PMBP Stereo	Joint PMBP
Out-All	0.3766%	0.2930%	0.2714%
Avg-All	5.1849px	3.8885px	3.5608px

Table 5.2: Stereo experimentation results computed at 3 pixel error threshold. Outperforming comparison baselines, our proposed joint estimation method shows close stereo accuracy to the state-of-art results on Kitti benchmark list. (As of August 2015). Out-Noc: percentage of erroneous pixels in non-occluded areas; Out-All: percentage of erroneous pixels in total; Avg-Noc: average disparity in non-occluded areas; Avg-All: average disparity in total.

Method	Out-Noc	Out-All	Avg-Noc	Avg-All
ALE (Ladický et al., 2011)	5.27%	8.48%	1.4607px	1.9871px
PMBP Stereo	4.64%	5.96%	1.1820px	1.6693px
Joint PMBP	3.82%	5.69%	0.9932px	1.2650px

the ground-truth semantic labels from (Ladický et al., 2014) to train the ALE classifier on the Kitti dataset.¹ We directly use the semantic classification output as the semantic label initialization of our MRF formulation. Quantitative evaluations on semantic classification accuracy on two datasets can be found in Table 5.3 and Table 5.4 respectively.

Though quantitative improvements upon semantic classifications being modest, Table 5.1 and Table 5.2 show that stereo matching can benefit from semantic regularization, as our proposed method outperforms ALE and pure PatchMatch stereo method. We attribute the insignificant semantic improvements to the highly accurate initialization with over 99% accuracy.

Table 5.3: Semantic classification accuracy on the Leuven test dataset. Seven semantic categories are defined for typical outdoor scenes. Despite being simple, our method can slightly improve upon state-of-the-art by simply smoothing the classification results together with stereo disparity maps.

Method	All	Pavement	Person	Bike	Car	Building	Road	Sky
ALE	0.9948	0.6118	0	0.6765	0.9042	0.9729	0.9885	0.9967
Joint PMBP	0.9949	0.6110	0	0.6767	0.9046	0.9733	0.9887	0.9968

¹ Notice we perform quantitatively evaluation on this subset of labeled stereo image pairs provided in (Ladický et al., 2014), instead of the original Kitti stereo dataset.

Table 5.4: Kitti semantic classification accuracy evaluation.

Class	Overall	Sidewalk	Grass	Bike	Car	Pedestrian	Fence
ALE	0.9370	0.9683	0.9520	0.9645	0.9081	0.9364	0.9427
Joint PMBP	0.9378	0.9694	0.9520	0.9653	0.9132	0.9377	0.9433
Class	Building	Road	Tree	Sky	Sign	Column	
ALE	0.9569	0.9280	0.9709	0.9036	0.4770	0.0630	
Joint PMBP	0.9576	0.9289	0.9721	0.9050	0.4649	0.0622	

5.5.3 Satellite Images Experiments

5.5.3.1 Datasets

The possible drastic change in solar illuminance, weather, and atmosphere conditions for a designated region, poses great challenges for the dense stereo problem on satellite images. To counter such difficulties, commercial satellite image vendors provide multiple image captures within the same orbit pass over a given region of interest. Taken within very short time-intervals, such one-pass captures simplify stereo matching by providing similar illuminance conditions and image appearances.

In order to show the robustness and effectiveness of our proposed satellite stereo method, we collect multi-pass satellite image datasets, with images taken at different time and on different orbits (see Table 5.5 for details). Compared to one-pass captures, such datasets have higher availability but contain greater variety in viewing angles, solar illuminance, and scene contents, thus posing greater challenges for the stereo matching solver. For example in Figure 5.5, the Zarqa dataset is captured within a time span of three years.

Without loss of generality, we used multi-spectral images from WorldView-2 satellite sensors. Such images have ground sampling distance as low as 0.2 meters. Eight spectral bands provide spectrum coverage from optical wavelength to infrared wavelength. The 11 bits dynamic range also enriches the discrimination between different image pixels, which can potentially improve stereo results. Example results of our joint estimation can be found in Figure 5.5.

Table 5.5: Quantitative evaluation on multiple satellite datasets. Empirically, our proposed method expands linearly in terms of memory and computation time.

	Dataset	Resolution	Height range	Pass	View	SGM	BidPMBP	GridPMBP
Time	Xiapu	26.2 Mpx	500 m	1	2	0.48 Hours	0.57 Hours	0.18 Hours
	Bengaluru	66.1 Mpx	1000 m	5	5	9.73 Hours	5.75 Hours	1.78 Hours
	Zarqa	58.7 Mpx	1000 m	7	7	12.96 Hours	7.66 Hours	2.37 Hours
Memory	Xiapu	26.2 Mpx	500 m	1	2	1.8 GB	1.4 GB	1.4 GB
	Bengaluru	66.1 Mpx	1000 m	5	5	11.3 GB	8.8 GB	8.8 GB
	Zarqa	58.7 Mpx	1000 m	7	7	12.0 GB	9.4 GB	9.4 GB

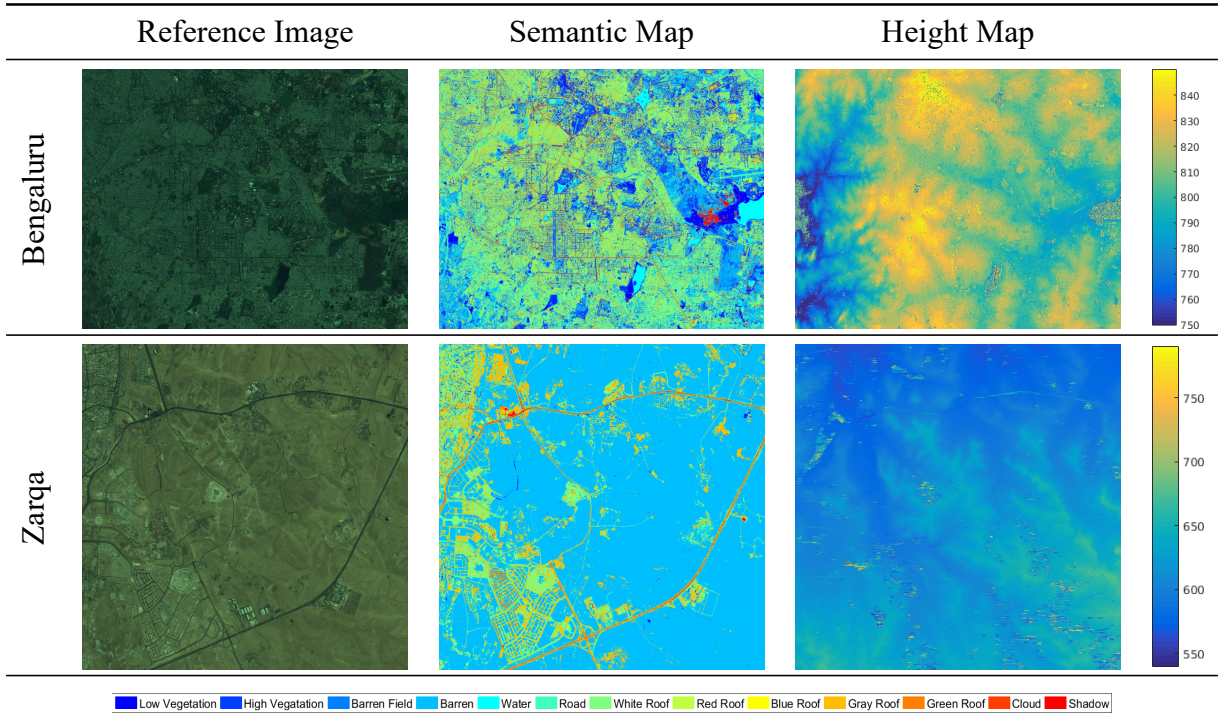


Figure 5.5: Example results obtained via GridPMBP solver on selected datasets. (Best view in color.)

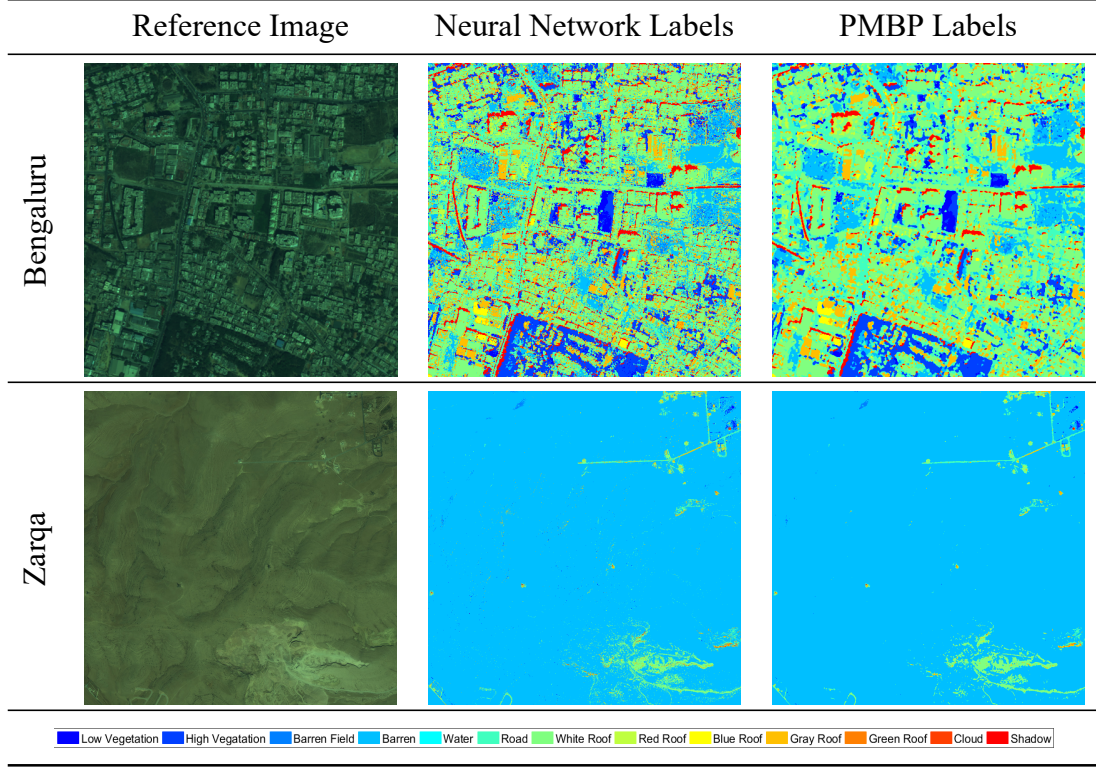


Figure 5.6: Comparison of semantic labels. Per-pixel based classification can lead to noisy semantic labels, see column 2. By joining dense stereo estimation together with per-pixel classifications, noisy semantic maps can be smoothed, see column 3. (Best view in color.)

5.5.3.2 Land Usage Classification

In order to train the land usage classification neural network, 12,000 samples are manually collected for each of the 13 semantic classes. A random split of 70% is used for training, 15% for cross-validation, and 15% for testing. The three-layer neural network is trained by scaled gradient back-propagation algorithm.

A simple per-pixel classification can be obtained by selecting the most likely semantic label for each pixel. Such maximum likelihood classification can lead to noisy semantic maps, as shown in Figure 5.6. By combining semantic classification with stereo correspondence estimation, we can effectively smooth the semantic map, leading to less noisy predictions.

5.5.3.3 Dense Stereo Benchmark

To evaluate our performance on dense stereo estimation tasks, we apply traditional image-space multiple-view stereo methods on our testing satellite image dataset as baselines.

As discussed in chapter 2, commercial satellite images usually either lack image-to-3D RPC camera model, or suffer from inaccuracies issues. We incorporated the minimal solver from (Zheng et al., 2015) to establish accurate bi-directional correspondences between 3D points and pixels. We embedded this minimal solver into traditional image space dense stereo solvers to ensure their functionality on satellite images.

Semi-Global matching (SGM) approximates a globally optimal solution for dense stereo problems by aggregating pixel-wise matching cost from multiple directions. SGM has been proved robust and successful for reconstructions in both standard pin-hole camera images, as well as aerial/satellite images (d’Ángelo and Kuschik, 2012; Gehrke et al., 2010).

The original SGM method proposed in (Hirschmuller, 2008) needs to store the entire matching cost volume for multi-direction cost aggregation. For a reference image with width D , height H , and disparity range D , such $O(WHD)$ memory requirement can hardly be fulfilled for high-resolution satellite images. So we adopted a memory efficient variant of the original SGM (Hirschmüller et al., 2012), which decreases the memory complexity to $O(kWH)$ where k is the number of aggregation directions.

We also compare our proposed method to standard image-space multiple view PatchMatch stereo method. In this case, we define each node \mathbf{u}_s over each pixel instead of a 3D space grid point. The unary and pairwise energy naturally follows the formulation in Section 5.2. We run bidirectional PatchMatch (BidPMBP) stereo on satellite image datasets with same patch radius and optimization iterations.

A detailed comparison of memory usage and runtime can be found in Table 5.5. Our proposed method (GridPMBP) achieved on average 3.2X speedup against bidirectional PatchMatch, and at least 5 times faster than SGM. Since the PatchMatch based method smartly traverses the disparity/altitude search space without iterating through the entire cost volume, our proposed

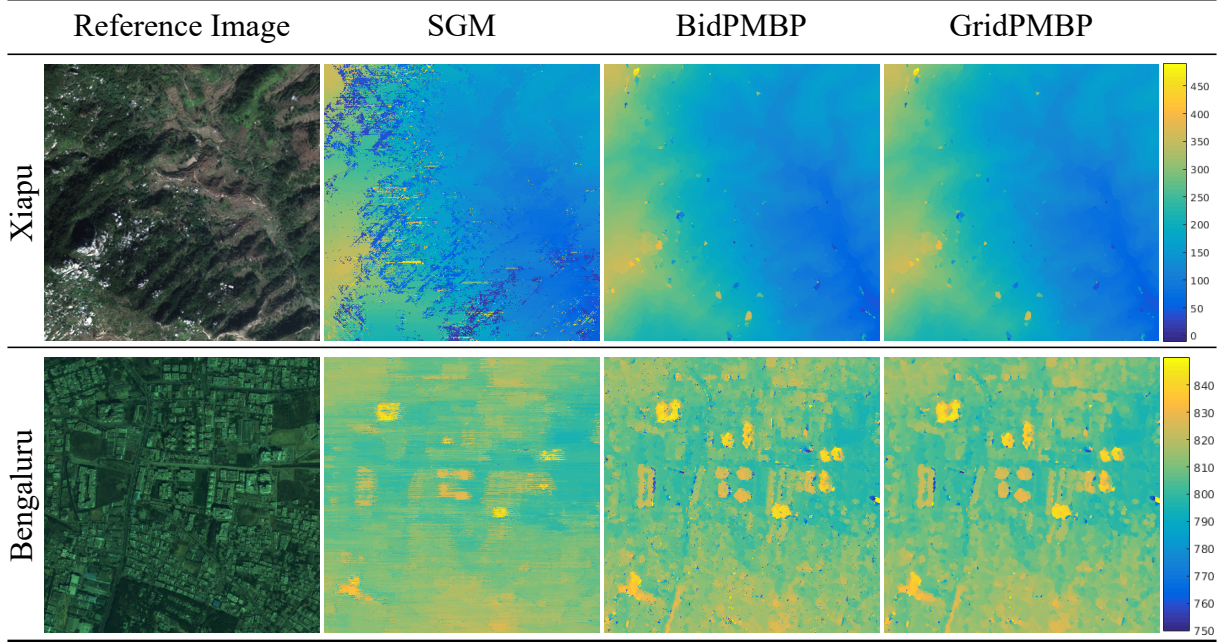


Figure 5.7: Zoom in comparisons of satellite height map estimation results. GridPMBP provides cleaner results with much smaller computation and memory overhead. (Best view in color.)

method inherits the lower memory requirement. Both BidPMBP and GridPMBP shows less memory usage than SGM. Qualitative zoom-in comparisons for three methods can be found in Figure 5.7.

As Figure 5.7 shows, our datasets cover different terrain types, including mountainous and urban areas. Compared with SGM, GridPMBP accurately captured finer details of the terrain shape on different terrain types, demonstrating great robustness and application ability. GridPMBP also achieved good stereo results on both one-pass stereo captures, and multi-pass captures. Thus our proposed method enables the use of satellite images captured from multiple-passes for accurate dense stereo problems.

5.6 Discussion

Overall, our proposed method achieved good results on satellite image datasets, as well as on standard stereo benchmark datasets. Especially, our proposed method has lower memory footprint and higher computation efficiency, which is more suitable for multi-view high resolution satellite image applications.

In our MRF formulation, the unary term $\psi_s(\mathbf{u}_s)$ is only evaluated between the reference view r and the per-node selected match view v_s , thus our proposed method has a low complexity with matching view numbers. Increasing dataset cardinality won't lead to catastrophic runtime increase. Also, the piecewise smoothness in a typical image domain, and spatial propagation scheme employed by PatchMatch based methods, greatly amortizes the optimization burden amongst neighboring pixels. This also makes the optimization of complex state space and unary cost function feasible. Last but not least, by using a grid space representation, we effectively by-pass the limitation of missing image-to-space RPC models, and thus save the overhead of computing such inverse mapping on-the-fly.

As can be seen in Figure 5.6, pixel-based local classification can be error-prone. However, our PMBP solver is independent of the methods used to attain these labels (though we do require a confidence value for each label). Hence, improved classification results can be easily integrated for better results. We consider this as part of the future work.

By adopting the space grid representation, our method has advantages that are not seen in the baseline methods:

1. Compared with the traditional multiple-view-stereo pipeline, which first establishes dense correspondences to extract depth/height maps, then does fusion to extract dense geometry, our space grid representation directly optimizes for 3D geometry structure. Huge computation overhead can be thus saved in later fusion stage.
2. Depth maps can be obtained by projecting the estimated geometry structure to the desired view. Our space-based representation also creates possibilities for virtual view synthesis.
3. The spatial resolution of the MRF grids can be changed to achieve a good balance between geometry fine-details and computation resource budget.

CHAPTER 6: PARAMETRIC CUBOID MODEL RECONSTRUCTION

6.1 Introduction

Despite its wide availability and increasing popularity, satellite imagery exhibits two significant drawbacks for 3D modeling. First, although satellite images contain enormous amounts of pixels, they are quite limited in content resolution. For instance, state-of-the-art WorldView-4 images can only provide images with ground sampling resolutions no smaller than 30 centimeters per pixel. Hence, models reconstructed from satellite imagery often exhibit undesired smoothing artifacts along structural edges and depth discontinuities. Second, satellite image vendors typically provide only one view of an area at each time-point as satellite orbits around the Earth. Lacking multiple overlapping views inhibits direct use of stereo methods for surface estimation. Thus, in this chapter, we focus on using single view satellite imagery to perform automatic urban reconstructions.

Nevertheless, single view reconstruction from the top-down bird’s eye view is very challenging. Regularizations or prior knowledge are necessary to solve such ill-posed problems. Satellite images are usually geo-registered to geodetic coordinate systems. Such geo-registration is usually utilized to project satellite images onto geodetic maps like OpenStreetMap (OpenStreetMap-Foundation, 2006), but it also provides a direct link between the rich semantic annotations of vector geodetic maps and the raster pixels of satellite images. Thus, we propose a deep learning based method to perform single-view parametric building reconstructions, using satellite imagery and their corresponding 2D vector maps as supervision. Our method uses a convolutional neural network (CNN) to directly localize all instances of buildings visible within a single satellite image and simultaneously estimate their skeletal 3D models. LiDAR information is utilized during training to learn to predict such 3D information. Note that GIS annotations and LiDAR are only

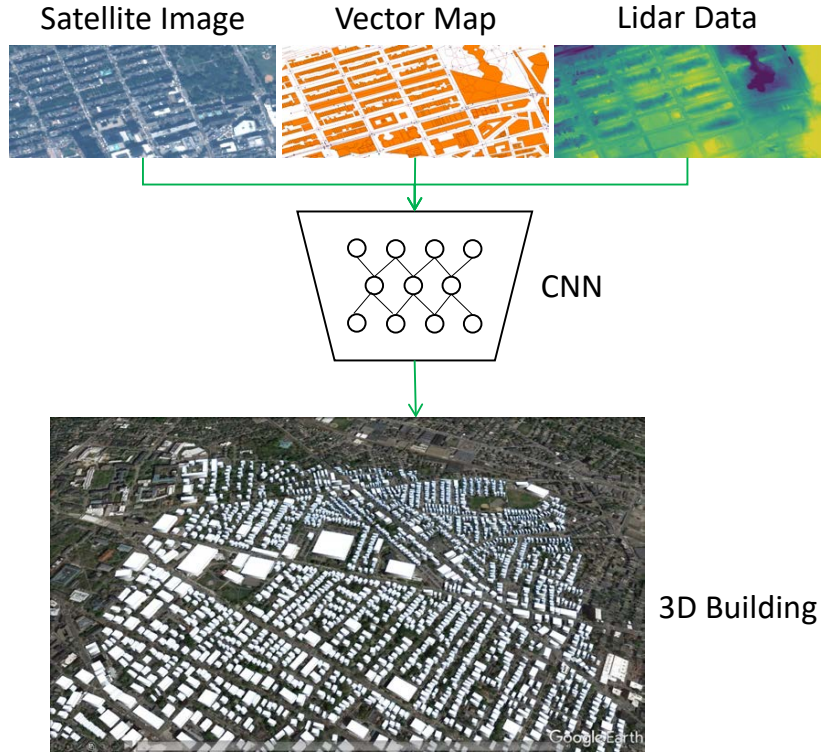


Figure 6.1: Overview of our proposed algorithm. Vector map and LiDAR data are only needed during training.

used during training. Requiring only images at application time makes our method much more broadly applicable.

To summarize, we propose a data-driven approach to perform direct parametric model fitting for single view reconstructions. Compared to traditional single-view satellite reconstruction methods which utilize shadow information and metadata like earth-sun positions (Izadi and Saeedi, 2012), our method learns useful visual cues from the abundant training data automatically. Our main novelties include:

1. We propose a simple but effective 3D cuboid parameterization for buildings, which allows easy integration into the detection formulation.
2. We upsample convolutional feature maps to a higher resolution to account for the low ground sampling rates of satellite imagery.



Figure 6.2: Comparison of traditional object detection data and our satellite reconstruction data. The left two images are from the MS-COCO dataset. Ground-truth object instance and their segmentation masks are visualized. The right two images are images of Boston, MA. Building labels are obtained from OpenStreetMap (OpenStreetMap-Foundation, 2006) and are overlaid on the satellite images. Note the significantly higher object density and small object size in the satellite image in the right two images.

3. We design a novel signed distance metric to predict building boundaries, and address detection overlaps.
4. We train the model utilizing GIS and LiDAR for supervision, but only images are needed at inference time, making our method much more applicable.

6.2 Cuboid Models

Given a single satellite image, we introduce a unified deep CNN to identify all buildings and extract their corresponding models as 3D cuboids. Our method is inspired by state-of-the-art object detection systems but addresses the additional challenge of handling high object density in images with low content resolution, which is not commonly seen in traditional visual recognition tasks. Figure 6.2 shows an example comparison of traditional object densities and the object densities observed in the targeted satellite data of urban areas.

Compared to the images traditionally used for object detection (for example ImageNet (Russakovsky et al., 2015), MS COCO (Lin et al., 2014)), the content density in typical satellite data is significantly higher. This is a consequence of sensor resolution constraints of current satellite cameras. While the images have extraordinarily high pixel counts, their actual ground sampling resolution is coarse relative to the scale of most objects. Hence, the objects of interest, buildings in our case, are represented by only a few pixels. For example, a 100 m^2 house is no larger than

34×34 pixels when captured by the highest commercially available resolution (0.3 meters per pixel ground sampling distance). In multi-spectral images, size is even further reduced to approximately 8×8 pixels. State-of-the-art object detection systems usually focus on localizing objects covering large spatial areas within the given image, and thus are not satisfactory for this task.

In addition to the resolution challenge, the building sizes and orientations vary significantly even within the same environment. Moreover, in standard object detection systems, overlapping bounding boxes are well within the range of valid outputs. However, for our application, we need to maintain the constraints of the physical world, *i.e.*, buildings cannot overlap.

In short, our task is to design a system that can accurately detect and model a large number of disjoint, highly-varied buildings from heavily cluttered, low-resolution data.

6.3 Cuboid Model Fitting as 3D Object Detection

Although buildings are often in different shapes, their underlying architectures share many similarities. For example, most buildings have a fairly rectangular footprint. They are also generally perpendicular to gravity (with a few notable exceptions, such as the Leaning Tower of Pisa). Our method leverages these characteristics to derive a general approximated model for buildings in urban areas: a 3D cuboid. Its orientation can thus be described by its azimuth angle, *i.e.* its rotation wrt the gravity direction.

An example model is shown in Figure 6.3. It is parameterized by its base width w_b , base length l_b , roof height h_r , azimuth angle θ , and its base center location (x_c, y_c) on the ground. We found that even such a simple model provides sufficient information for applications like city planning and monitoring. Interestingly, due to its inherent semantics, the fitted structures are often more useful than the standard point cloud representations from LiDAR scans, stereo depth maps, or volumetric representations. Our cuboid building parametrization shares several parameters with 2D image bounding boxes. In particular, l_b , w_b , and (x_c, y_c) naturally correspond to a bounding box in a 2D image or ground plane. Thus, our algorithm extends the state-of-the-art

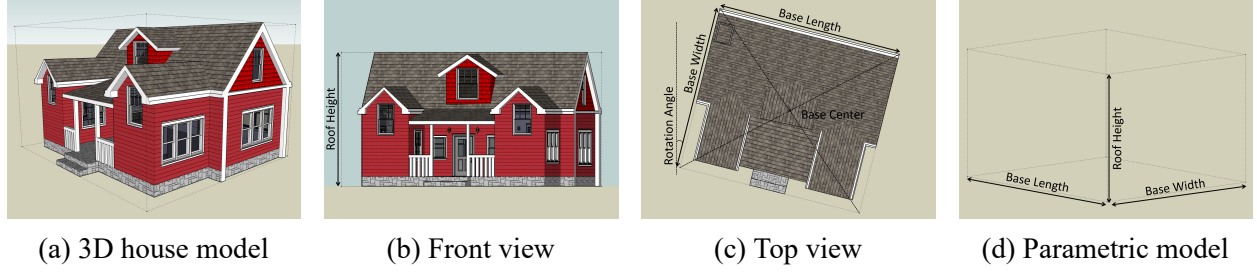


Figure 6.3: Visualization of the 3D cuboid parameterization for an example house.

2D object detection pipelines into 3D space, identifying each building instance and estimating its parameters $(w_b, l_b, h_r, \theta, x_c, y_c)$.

6.3.1 Network Architecture

Base feature network SSD and FCN both use a high quality convolutional neural network for visual feature extraction. We modify the Residual-101 (He et al., 2016) as the base feature extraction network. We follow the design and architecture of Residual-101 but added the following changes. Convolutional feature maps from Residual-101 are upsampled by deconvolutional layers. Such upsampled high level feature maps are concatenated with lower level feature maps. Please refer to Figure 6.4 for a detailed illustration and Table 6.1 for a detailed configuration. Notice that depending on the satellite image data, the input image can have a different number of spectral channels C . Each convolutional and deconvolutional layer is followed by a batch normalization layer.

We follow the spirit of the single-shot detection (SSD) network (Liu et al., 2016) in designing our architecture. A high quality convolutional neural network is used as the base network to extract visual features. In our case, we chose the Resnet-101 architecture (He et al., 2016). The conv5_x feature before the final average pooling and the fully connected layer is used as the visual feature for given input images. Compared with the VGG16 (Simonyan and Zisserman, 2015) base network used in the original SSD framework, the much deeper Resnet-101 network has been shown to extract better feature representations, as indicated by its superior performance in different vision tasks.

Table 6.1: Base feature network detailed architecture. conv3_1, conv4_1, conv5_1 have a stride of 2 to downsample the feature map.

Layer	Input dim	Output dim	Kernel	Stride
Conv1	$224 \times 224 \times C$	$112 \times 112 \times 64$	$7 \times 7 \times 64$	2
Max pool	$112 \times 112 \times 64$	$56 \times 56 \times 64$	3×3	2
Conv2x	$56 \times 56 \times 64$	$56 \times 56 \times 256$	$\begin{bmatrix} 1 \times 1 \times 64 \\ 3 \times 3 \times 64 \\ 1 \times 1 \times 256 \end{bmatrix} \times 3$	1
Conv3x	$56 \times 56 \times 256$	$28 \times 28 \times 512$	$\begin{bmatrix} 1 \times 1 \times 128 \\ 3 \times 3 \times 128 \\ 1 \times 1 \times 512 \end{bmatrix} \times 4$	1
Conv4x	$28 \times 28 \times 512$	$14 \times 14 \times 1024$	$\begin{bmatrix} 1 \times 1 \times 256 \\ 3 \times 3 \times 256 \\ 1 \times 1 \times 1024 \end{bmatrix} \times 6$	1
Conv5x	$14 \times 14 \times 1024$	$7 \times 7 \times 2048$	$\begin{bmatrix} 1 \times 1 \times 512 \\ 3 \times 3 \times 512 \\ 1 \times 1 \times 2048 \end{bmatrix} \times 3$	1
deconv1	$7 \times 7 \times 2048$	$14 \times 14 \times 512$	$3 \times 3 \times 512$	
deconv2	$14 \times 14 \times 1536$	$28 \times 28 \times 256$	$3 \times 3 \times 256$	
deconv3	$28 \times 28 \times 768$	$56 \times 56 \times 128$	$3 \times 3 \times 128$	
deconv4	$56 \times 56 \times 384$	$112 \times 112 \times 64$	$3 \times 3 \times 64$	

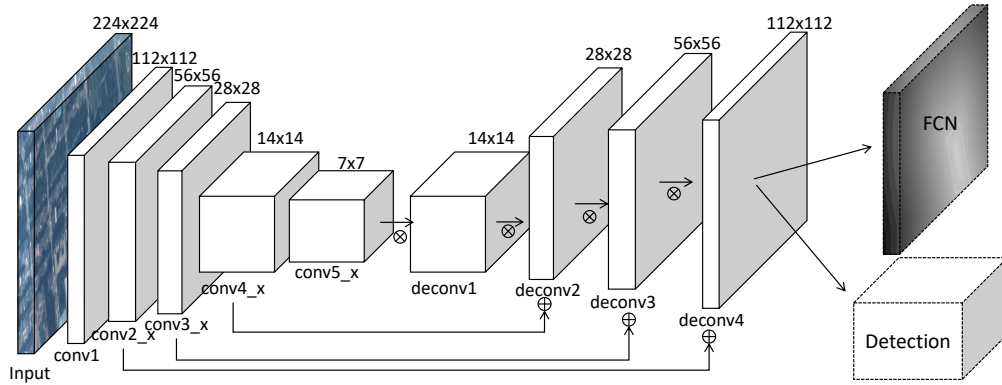


Figure 6.4: Proposed network architectures. We first use Resnet-101 (He et al., 2016) as the base feature extraction network. Feature maps at different levels are upsampled by deconvolutional layers (\otimes). We use skip connections to concatenate (\oplus) higher resolution but lower level features with lower resolution but higher level features. For an given input image of resolution 224×224 , we upsample the feature map to 112×112 before the final detection head and FCN head. The SSD detection head and FCN signed-distance head share the same base feature network, but they are separately trained and don't share weights. deconv1 has 2048 filters, deconv2 1024, deconv3 512, deconv4 256.

Table 6.2: Detection network detailed architecture.

Layer	Input dim	Output dim	Kernel	Stride
ssd_conv1	$112 \times 112 \times 64$	$56 \times 56 \times 128$	$3 \times 3 \times 128$	2
ssd_conv2	$56 \times 56 \times 128$	$28 \times 28 \times 256$	$3 \times 3 \times 256$	2
ssd_conv3	$28 \times 28 \times 256$	$14 \times 14 \times 128$	$3 \times 3 \times 128$	2
ssd_conv4	$14 \times 14 \times 128$	$7 \times 7 \times 64$	$3 \times 3 \times 64$	2
avg_pool	$7 \times 7 \times 64$	$1 \times 1 \times 64$	7×7	

For a 224×224 input image, the original Resnet-101 architecture will produce a 7×7 convolutional feature map, which is 32 times smaller in spatial resolution. Considering our 100 m^2 house appears no larger than 34×34 pixels even in the best satellite imagery, reducing its spatial resolution 32 times will likely destroy any useful features, making accurate detection near impossible. Thus, higher resolution feature maps are necessary for our task. Consequently, we use low-level feature maps with a larger spatial resolution to capture fine details and high-level features with a smaller spatial resolution to encode high-level semantics. Inspired by fully convolutional networks (FCN) (Shelhamer et al., 2016), we upsample feature maps at different feature extraction stages of the network and concatenate the resulting feature maps together as input to later detection stages. See Figure 6.4 for example.

We use the `deconv_4` feature as the input to the SSD detection sub-network. SSD then applies different convolutional layers to allow object prediction at multiple scales. A fixed set of 3D cuboids are estimated at each cell on each feature map level. To detect one building in the feature map, we need to estimate its six geometric parameters $(w_b, l_b, h_r, \theta, x_c, y_c)$ and one semantic class score *conf* (building or not). For a feature layer of size $m \times n$ with p channels, we apply $7k$ small $3 \times 3 \times p$ convolution kernels to estimate k default 3D cuboids. This leads to $7kmn$ outputs for the $m \times n$ shaped feature map. Using a convolutional filter to fit cuboids at each feature map location provides a natural solution for our heavily cluttered scenario.

Feature maps from the base feature network are fed into the multi-scale detection framework to find buildings and estimate their parameters. Please refer to Figure 6.5 and Table 6.2 for detailed configurations.

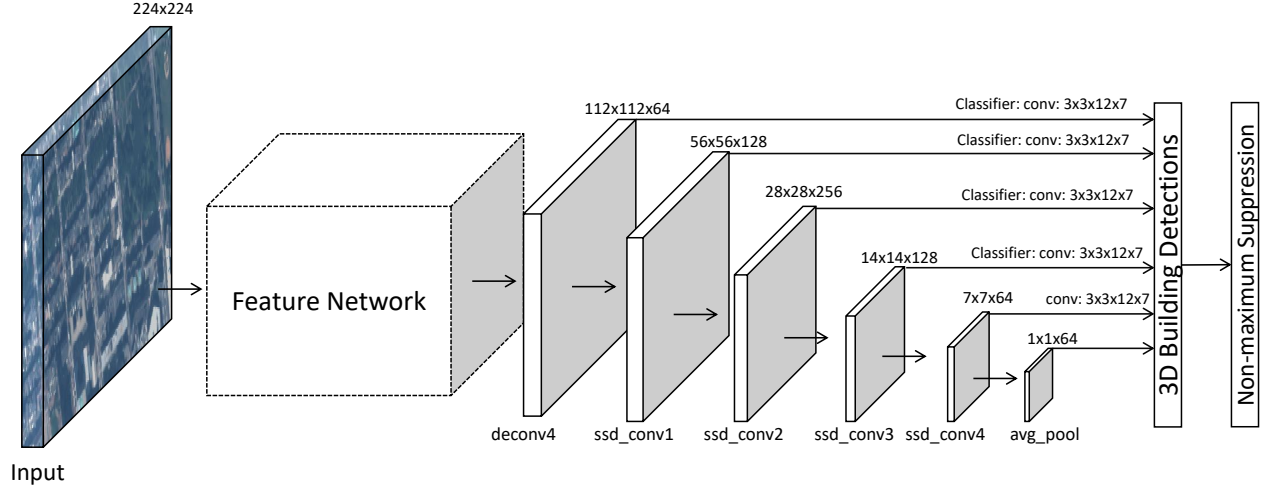


Figure 6.5: Architecture for 3D detection network.

6.3.2 Vector Map Pre-processing

We want to use public GIS information, such as OpenStreetMap, as supervision to train the neural network. 2D building footprints are generally represented as polygons in OpenStreetMap. Polygon vertex coordinates are given in the geo-spatial latitude/longitude system, while the neural network needs to know precise pixel locations in the satellite images in order to learn about buildings. To train our neural network, we first need to align the raster satellite image and LiDAR data wrt the OpenStreetMap vector labeling data. Geo-referenced satellite images and LiDAR data are first reprojected to the same WGS84 coordinate system (DoD, 1984).

Once geo-referenced wrt WGS84, an affine transformation $\mathbf{A} \in \mathbb{R}^{2 \times 3}$ can be obtained for each satellite image and LiDAR map to relate raster position (in pixel/line coordinates) and geo-referenced locations (in latitude/longitude coordinates) $[lng, lat]^T = \mathbf{A}[x, y, 1]^T$.

Each building contour can then be projected into the image space with \mathbf{A} . We then calculate the ground-truth building cuboid parameters $(w_b, l_b, x_c, y_c, \theta)$ in the image space. Given an irregular polygon, we first obtain all pixel locations M falling within the polygon, and the tightest upright enclosing box around the polygon. Orientation is estimated via the moments of the polygon $m_{pq} = \sum_{x,y \in M} x^p y^q I(x, y)$, where $I = 1$ for pixels within the building boundary and 0

otherwise. Similar to ORB (Rublee et al., 2011), we can obtain the orientation as:

$$\theta = \tan^{-1}(m_{01}, m_{10}) \quad (6.1)$$

and the building center as:

$$(x_c, y_c) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (6.2)$$

The oriented building polygons are then rotated to the up-right directly based on the estimated angle θ . The bounding box size l_b, w_b is measured as the pixel dimension of the tightest enclosing bounding box in the up-right direction. The ground-truth roof-height h_r is computed as the average height of the LiDAR data within the polygon. In summary, this obtains all the ground-truth parameters of the labeled buildings from OpenStreetMap.

6.3.3 Training

Ground-truth cuboid matching Following Liu et al. (2016), our detection network performs bounding box regression at each feature map location without using region proposals (Ren et al., 2016). Thus during training, we need to know which default cuboid box correspond to ground-truth detections to allow end-to-end training. We match each ground-truth cuboid to the default cuboid with the highest IoU overlap. In addition, ground truth cuboid boxes are also assigned to default cuboids with higher than 0.5 IoU overlap.

Objective function At each feature map location, the detection network needs to identify if a cuboid contains a building, and, if so, what the geometric parameters of the building are. x_{ij} is an indicator function when default cuboid i matches ground-truth cuboid j for building detection. We formulate the training objective as a weighted sum between the classification confidence loss L_{conf} and the geometric prediction errors L_{geo} .

$$L_{det}(x, l, g) = \frac{1}{N} (L_{conf} + \alpha L_{geo}(x, l, g)) \quad (6.3)$$

We set the weight, α , to 2.5 via cross-validation. The confidence loss L_{conf} is a single-class cross-entropy loss for buildings. We adopt a cuboid regression parameterization strategy similar to Faster R-CNN (Ren et al., 2016). The convolutional regressors output the predicted cuboid l center location as the offset wrt the default cuboid g center location, predicted cuboid length l_b , and width w_b , rotation angle θ , and roof height h_r , as relative scales wrt the default cuboid.

$$\tilde{g}_j^{x_c} = (g_j^{x_c} - d_i^{x_c})/d_i^{x_c} \quad \tilde{g}_j^{y_c} = (g_j^{y_c} - d_i^{y_c})/d_i^{y_c} \quad (6.4)$$

$$\tilde{g}_j^{w_b} = \log(g_j^{w_b}/d_i^{w_b}) \quad \tilde{g}_j^{l_b} = \log(g_j^{l_b}/d_i^{l_b}) \quad (6.5)$$

$$\tilde{g}_j^{h_r} = \log(g_j^{h_r}/d_i^{h_r}) \quad \tilde{g}_j^{\theta} = \log(g_j^{\theta}/d_i^{\theta}) \quad (6.6)$$

The geometric loss models the difference between the predicted cuboid l and the ground truth cuboid g :

$$L_{geo}(x) = \begin{cases} 0.5x^2 & \text{if } 0 \leq x \leq 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \quad (6.7)$$

$$L_{geo}(x, l, g) = \sum_{i \in Pos}^N \sum_{m \in \{x, y, w, h, l, \theta\}} x_{ij} L_{geo}(l_i^m - \tilde{g}_j^m) \quad (6.8)$$

Default cuboid design We use the same scaling and aspect ratio setting as the original SSD framework (Liu et al., 2016). Differently, each default cuboid is also responsible for regressing angle θ and the roof height h_r . The original six default cuboids are placed at one feature map location without rotation. We add another six default cuboids with the same scale and aspect ratios but rotated clockwise by $\pi/4$.

Data augmentation To augment our image set during training, we randomly sample from among the following strategies:

1. randomly sample a patch,
2. random horizontal flip with 0.5 probability,
3. random vertical flip with 0.5 probability,

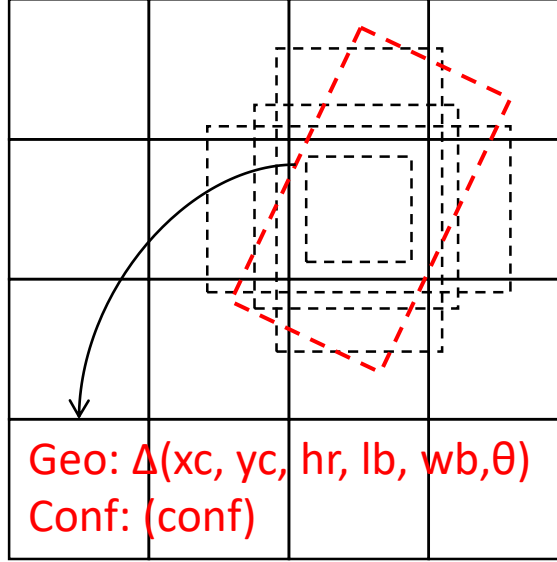


Figure 6.6: Default cuboid design. In addition to bounding box center and size, our default cuboids also regress the rotation and roof height. Thus we added another group of six rotated default cuboids with the same scale and aspect ratio.

4. random rotation of the image.

Unlike images used in other visual detection tasks, the visual semantics of the aerial image do not change after horizontally/vertically flipping or arbitrarily rotation. Allowing flipping and rotation in data augmentation can let the network be more sufficiently trained to better estimate building orientation.

Hard negative mining Many default cuboids are negative examples during training, which leads to a significant imbalance between positive and negative training samples. We follow (Liu et al., 2016) to pick highly confident negative samples and constrain the negative-to-positive ratio up to 3.

Optimization Deconvolutional (transposed convolutional) layers are initialized with bilinear interpolation weights. Later convolutional layers are randomly initialized with the Xavier uniform initializer (Glorot and Bengio, 2010). We train the detection network using Adam (Kingma and Ba, 2015) optimizer with an initial learning rate of 10^{-4} and batch size 8.

6.4 Overlapping Refinement

Outputs from each detection convolutional layer are aggregated to produce raw 3D building detection results. In most detection pipelines, a non-maximum suppression with a small intersection-over-union (IoU) threshold is applied to discard less confident overlapping detections. Since small amounts of overlap are valid for standard object detection tasks, a small IoU threshold is allowed in non-maximum suppression to ensure higher recall. However, buildings do not overlap in the real world. Thus, in our task, predicted 3D cuboids could not overlap with each other. Naïvely, setting the IoU threshold to 0 might eliminate all overlapping outputs, but empirically this turns out to dramatically reduce detection recall. Due to the limited ground sampling resolution of satellite images, smaller cuboids are usually predicted less confident. More confident larger cuboids can eliminate all surrounding neighbor detections with zero IoU threshold during non-maximum suppression.

To resolve the overlap problem, we introduce another fully convolutional network (FCN) that estimates pixel distances wrt to building boundaries. As with our detection network, we use the Resnet-101 feature maps as the base network. We attach another deconvolution layer followed by two convolution layers after the upsampled Resnet-101 feature map to predict a pixel-wise signed distance map. At each pixel location, the predicted value is the smallest distance wrt the building boundaries. Pixel locations inside the building boundaries have positive values while pixel locations outside the building boundaries have negative values. With these outputs, bounding contours are clearly defined as the zero level set in the predicted signed distance map. See Figure 6.7 for an example.

For a given image, the forward pass of our network will produce a set of 3D cuboids around each building, as well as a pixel-wise signed distance map. For each cuboid box overlapping with other detections, we extract its corresponding building contour as indicated by the zero level-set on the signed distance map. We then use RANSAC to find the approximate cuboid within the enclosing contour that has the largest IoU value wrt the contour. h_r is not modified during fitting.



Figure 6.7: (Left) Example satellite image with labeled building polygons. (Right) Ground-truth signed distance field with ground-truth building polygons overlaid on top.

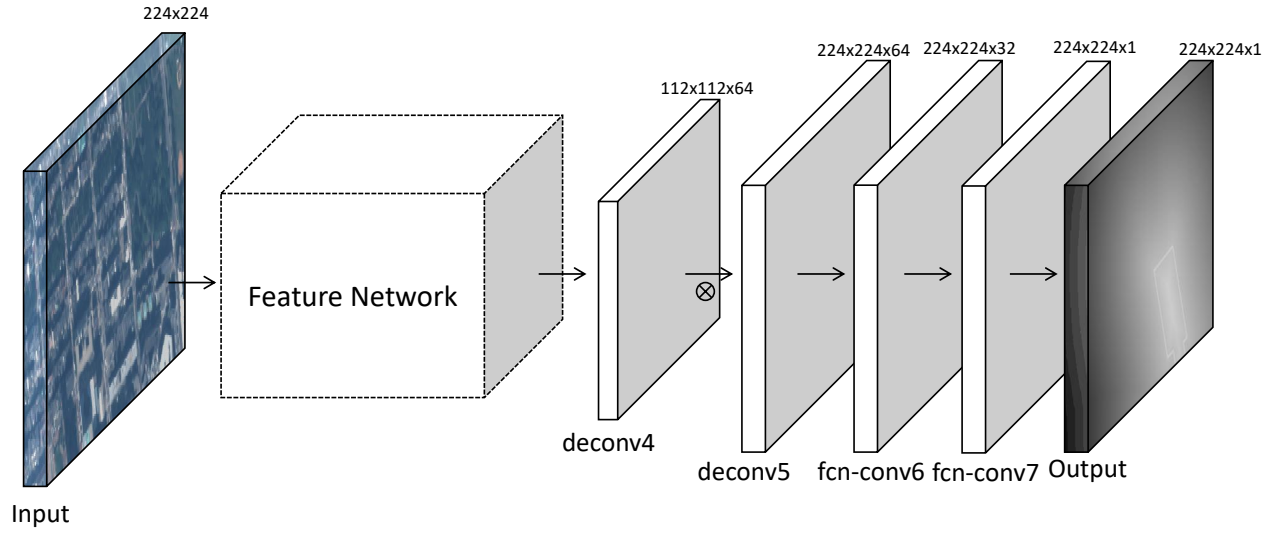


Figure 6.8: Architecture for signed-distance map prediction network.

6.4.1 Network Architecture

Please refer to Figure 6.8 and Table 6.3 for detailed configurations. Notice that we add a batch normalization layer after deconv5 and fcn-conv6.

6.4.2 Training

Our distance-map network uses a fully convolutional architecture, which allows image-to-image training. That is, for each input image and ground-truth signed distance map, the forward pass of the FCN network will produce an estimated signed-distance map at the same resolution

Table 6.3: Signed-distance map network detailed architecture.

Layer	Input dim	Output dim	Kernel	Stride
deconv5	$112 \times 112 \times 64$	$224 \times 224 \times 128$	$3 \times 3 \times 128$	
fcn-conv6	$224 \times 224 \times 128$	$56 \times 56 \times 64$	$3 \times 3 \times 64$	1
fcn-conv7	$224 \times 224 \times 64$	$56 \times 56 \times 1$	$3 \times 3 \times 1$	1

of the input image. The training objective, therefore, minimizes the sum of the squared error between the distance prediction and the ground-truth distance for all pixel locations.

$$L_{dis} = \sum_p \|d(p) - \tilde{d}(p)\|^2 \quad (6.9)$$

We adopted the same initialization strategy as our detection network. We used Adam with an initial learning rate of 5×10^{-4} and batch size 32.

6.5 Experiments

6.5.1 Dataset

Dataset We obtained satellite images and LiDAR data for the Boston area through public source. Vector labelings of the corresponding area are obtained through OpenStreetMap (OpenStreetMap-Foundation, 2006). In the Boston dataset, no overlapping views are available.

We also conducted experiments on a multi-view stereo satellite dataset (Bosch et al., 2016). The MVS dataset contains multiple satellite views of the same area taken at different times, and ground-truth LiDAR data is provided to evaluate the performance of the stereo reconstructions. We also made the ambitious move to compare our single-view based method against state-of-the-art multi-view stereo based methods (Wang et al., 2016b). SpaceNet (DigitalGlobe, 2016) is another open satellite image dataset with high quality labeled training data. We only use SpaceNet to test the generalization ability of our model.

Table 6.4: Detailed dataset statistics.

Dataset	Satellite	OpenStreetMap	LiDAR	Google Earth	Tiles
Boston	✓	✓	✓	✓	1327
MVS	✓	✓	✓	×	2132
SpaceNet	✓	✓	×	×	6094

Dataset split Satellite images contain huge numbers of pixels. Thus we split source images at a fixed 224×224 resolution to fit data into GPU memory. Specifically, bounding boxes intersecting image patch boundaries are discarded. We divide the satellite images into non-overlapping patches so that we can randomly split the patch collections into a disjoint training, validation, and test set.

We used three datasets in total for our experiments. Please refer to Table 6.4 for detailed data modalities about each dataset. In the Boston and SpaceNet dataset, we first withheld an area for visualization of large scale reconstructions. For MVS dataset and the remaining areas in Boston and SpaceNet, images are divided into non-overlapping tiles, and split into 80% training, 10% validation, and 10% test sets for training and evaluation.

Compared to ImageNet or MS-COCO, we have a smaller number of training image tiles. But each tile can contain hundreds of buildings, thus providing a sufficient number of building instances. In addition, we only need to learn one semantic category, building or not building, which also simplifies the learning task. We only evaluated the height estimation when ground-truth LiDAR is available. Since no LiDAR is available for the SpaceNet dataset, we cannot directly train models on the SpaceNet dataset. We used the ensemble of two models, one trained on Boston and the other trained on MVS, to test the generalization capability of our methods.

6.5.2 Evaluation

We show qualitative examples of our estimated 3D model from satellite images in Table 6.6. We also quantitatively evaluated our proposed system on two datasets. We adopted mean average precision (mAP) to evaluate detection performance, and root-mean-squared-error (RMSE) to

Table 6.5: Quantitative evaluations. Mean average precision (%): higher is better. Height RMSE (meters) lower is better. Rotation RMSE (radial) lower is better.

Dataset	mAP	Height RMSE	Rotation RMSE
Boston	0.51	2.72	0.17
MVS	0.49	2.10	0.19

evaluate geometric performance. Especially, we report the RMSE of roof height r_h and rotation angle θ for our true positive predictions. Details can be found in Table 6.5.

On the MVS dataset (Bosch et al., 2016), we further compare our height estimation performance against state-of-the-art multi-view satellite stereo methods (Wang et al., 2016b). For a fair comparison, we only evaluated the positive detected areas. LiDAR ground-truth is used as the evaluation metric. Not surprisingly, mvs method achieves a lower height RMSE (0.79) compared to the 2.10 RMSE reported for our method. Most importantly though our proposed method can estimate the 3D building primitives from a single image whereas our work (Wang et al., 2016b) requires multiple images, which are not always available. Additional limitations of our current training are that the network is trained with the average height value of a building, which can cause confusion.

We further compared our method with Sun et al. (2014) and Izadi and Saeedi (2012) on the MVS dataset. Sun et al. (2014) estimates 2D polygonal building footprints from a single view while Izadi and Saeedi (2012) utilizes shadow information to perform geometric analyses. Detailed results can be found in Table 6.7. Table 6.7 demonstrates that our method achieves significant improvements over the state-of-the-art. Intuitively, in residential areas, many buildings are cluttered and have similar height. Shadows are not obvious in such regions. Izadi and Saeedi (2012) thus performed poorly. We also report the IoU between predicted footprint and ground-truth on true-positive detections. Higher IoU means more ground truth pixels are covered. The limitation of cuboid representation leads to our lower IoU. We also noticed that our method is fairly robust to tree occlusions while Sun et al. (2014) and Izadi and Saeedi (2012) often fails.

Table 6.6: Visualizations of our single view reconstructed parametric models tested on the SpaceNet (DigitalGlobe, 2016) dataset. First row: detected buildings are visualized in red. Second row: estimated models Please refer to the supplementary materials for more visualizations.

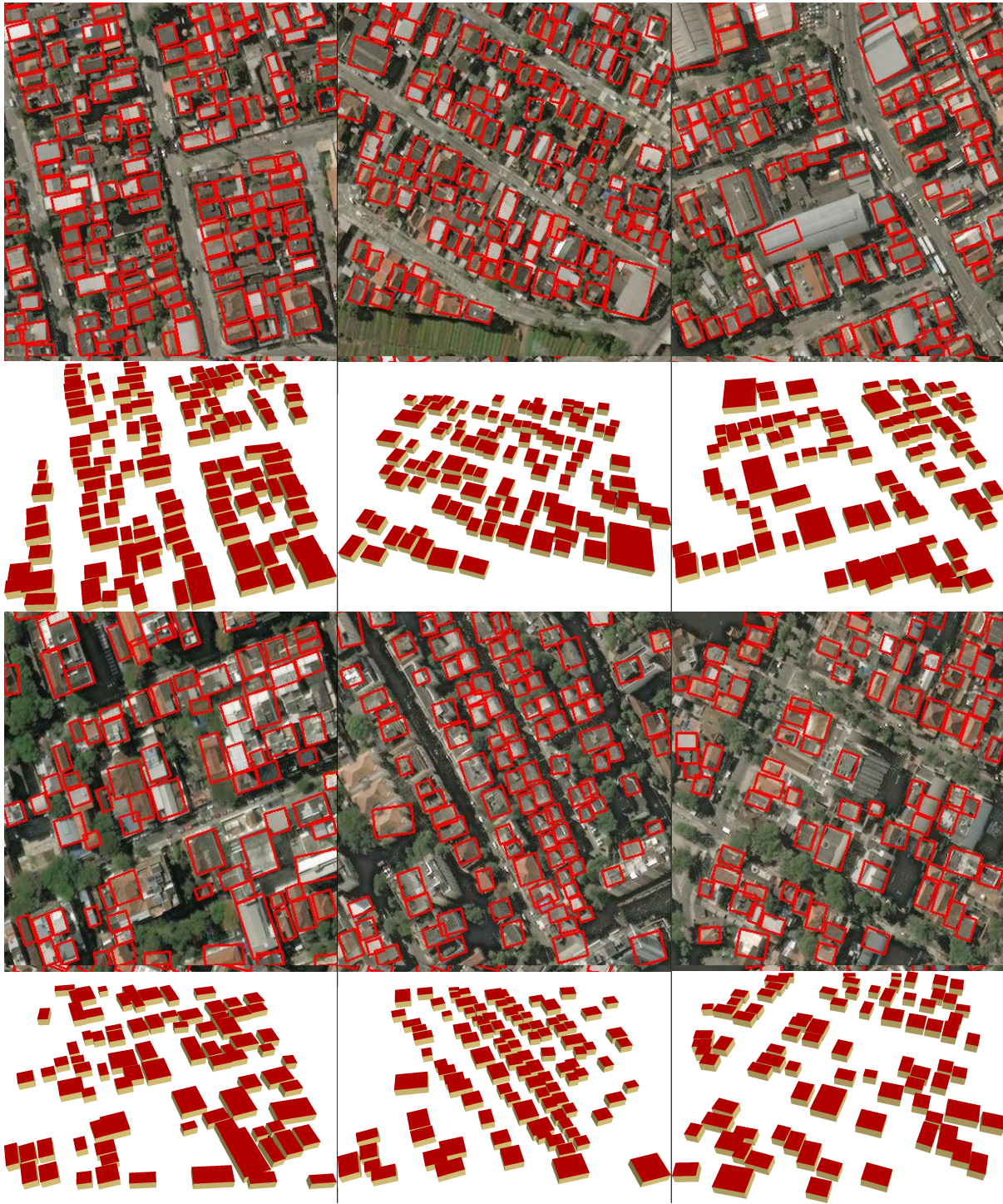


Table 6.7: Comparison with baseline methods.

Method	Sun et al. (2014)	Izadi and Saeedi (2012)	Ours
mAP	0.29	0.21	0.49
Recall	0.51	0.43	0.85
IoU	0.85	0.87	0.73
Height	NA	1.92	2.10

Our method achieved height estimation accuracy similar to Izadi but recovered significantly more buildings.

6.6 Discussion

6.6.1 Input data

Satellite images usually come in two forms: higher resolution single channel grayscale panchromatic images, and lower resolution multi-channel multi-spectral images. We found the best scores are achieved by pansharpening (Vivone et al., 2015) the low-resolution images to the panchromatic resolution and then concatenating them as input to the neural network. Intuitively, panchromatic images have finer details but multi-spectral pixel values from different frequencies carry more information.

6.6.2 Radiometric correction

Radiometric correction of satellite images is important for multi-view stereo, as indicated by (Wang et al., 2016b). However, radiometric correction causes negligible performance difference for our methods. Radiometric correction, in essence, is a linear transformation of the pixel values. Neural networks can easily learn such a transformation to mimic the radiometric correction. Hence, our proposed method shows significantly improved the robustness of its estimation to this typical class of satellite image disturbances.

Table 6.8: Performance comparison at different feature map resolution. Mean average precision (%): higher is better. Height RMSE (meters) lower is better. Rotation RMSE (radial) lower is better. Increasing the feature map resolutions can lead to improvements for all regression tasks. We use the largest 112×112 feature map in our benchmark.

Metric	Dataset	14×14	28×28	56×56	112×112
mAP	Boston	0.21	0.37	0.47	0.51
	MVS	0.19	0.32	0.41	0.49
Height	Boston	4.12	3.97	3.15	2.72
	MVS	4.19	3.83	3.08	2.10
Rotation	Boston	0.33	0.27	0.23	0.17
	MVS	0.32	0.29	0.25	0.19

6.6.3 Feature map resolution

The original SSD framework has reduced performance wrt small objects. A special “zoom-out” data augmentation strategy is adopted during training to overcome this. We took a different approach, *i.e.*, enlarging the feature map input to the detection module. We observed consistently improving results, as shown in Table 6.8.

6.6.4 Instance-aware segmentation

Instance-aware semantic segmentation systems can find all instances of objects together with their belonging pixels. We compare our detection based system against state-of-the-art instance-aware segmentation system (Multi-network cascade (Dai et al., 2016)). Since the original MNC network cannot predict building rotation θ and roof height r_h , we only compare mAP metric on detection results. On the Boston test set, our method (52.1% mAP) outperforms the MNC system (46.2%). Qualitative results are shown in Figure 6.9.

6.6.5 Limitations of Cuboids

The cuboid representation we used has its limitations, for example as shown in the above IoU comparison. But it also has the following advantages. Cuboid representation is simple yet effective. Instance-aware segmentation or FCN gives irregular boundaries. Comparatively, regu-



Figure 6.9: (Left) Example detection using our proposed system. (Right) Results from instance-aware segmentation system (Dai et al., 2016). Note that MNC detects fewer buildings and produces irregular boundaries.

lar shapes are easier to process for later applications for example in GIS data extraction or large scale 3D maps as our cuboid representation is compact and efficient to store/process. Existing methods that estimated polygonal models, usually require additional data other than images (Ortner et al., 2007). Our cuboid achieved a balance between accuracy and the required input data.

6.6.6 Future work

Currently, our proposed system outputs a flat top for all building instances. Simple rooftop models can add much more realism to the reconstructed models. Our method can be readily extended to include root-type classification: instead of judging if an image patch is building or not, we let the network learn what kind of rooftop it is. In addition, misalignment between raster data and vector labeling hinders the performance of our model. We leave the improvement of the training data to future work.

CHAPTER 7: DISCUSSION

Satellite images, due to their unique characteristics, bring both challenges and opportunities for many potential applications. Throughout this dissertation, we have demonstrated the possibility of geometric, semantic, and parametric 3D reconstructions from satellite imagery. In this chapter, we conclude this dissertation and briefly explore possible future research directions related to satellite imagery.

7.1 Conclusion

In this dissertation, we systematically studied the characteristics of modern satellite images and the challenges they posed for 3D reconstructions. We proposed efficient and effective methods to automatically compensate for the calibration errors embedded in the satellite RPC camera model. Beyond just geometric reconstructions, we take full advantage of the available data to explore high-order semantic reconstructions as well.

Considering the huge number of disparity candidates, we proposed stereo reconstruction algorithms that are highly efficient. Both our stereo algorithm and joint reconstruction algorithm discarded the exhaustive disparity search during the reconstruction process. Relying on image space edge-aware interpolation (Chapter 4), or 3D space local structure sampling (Chapter 5), our satellite reconstruction algorithm has demonstrated unprecedented computation efficiency. The success of our algorithms heavily utilized the local smoothness priors of natural images. For edge-aware interpolation, reliable but sparse correspondences are obtained first by rejecting outliers using the triangulation solver. The local smoothness prior ensures that interpolating neighboring correspondences can lead to meaningful and accurate dense, pixel-wise correspondence field. For the 3D scene sampling, the local smoothness prior distribute the huge optimization

burden across the local neighborhood, thus boosting the computational efficiency of the inference process. We demonstrated that by carefully incorporating such useful image priors with the special characteristics of satellite images, we could significantly increase both the accuracy and efficiency of satellite imagery based 3D reconstruction pipeline.

Besides point cloud reconstructions and semantic models, to the best of our knowledge, we are the first to consider the usefulness of parametric models for urban reconstruction tasks. Compared with traditional point cloud models, parametric models are compact for data transfer but still preserve certain geometric information for many applications, for example, 3D mapping. The instance-level semantic information also makes parametric models useful for direct interaction and manipulation. The geo-registration between satellite images and the public GIS information (vectorized map data) provides the opportunity for applying machine learning methods to extract parametric models from a single view. Our work already demonstrated the possibility of utilizing such geo-registration information to train data-hungry machine learning models. Last but not least, we believe our work, although limited in its cuboid representation capabilities, shed light on the importance and practicalness of single view parametric model reconstructions. We strongly believe that parametric modeling, as well as utilizing geo-registration to train data-hungry models, will prove useful for many more applications on satellite imagery.

7.2 Future Work

7.2.1 Efficient and Robust RPC Solvers

In Chapter 3, a Groebner basis based solver is proposed to solve the RPC triangulation problem. Although the Groebner basis based solver empirically generates good solutions in a fairly robust and efficient manner, other alternative solutions exist. In its essence, the RPC projection process can be re-written as a set of cubic equations. A combination of resultants and numeric eigen solvers can be used to compute the solutions for such polynomial equation systems. Such alternative methods have demonstrated their success in other vision problems, as shown in

(Manocha, 1994; Wallack and Manocha, 1998). For three cubic equations, good resultants are known based on the Dixon formulation (Kapur et al., 1994; Manocha, 1994). Thus, only three out of the four known pixel coordinates are needed to solve the polynomial equation system. The unused fourth pixel coordinate can be used for verification and outlier rejection, as described in Chapter 3.

7.2.2 Multi-Modal 3D Reconstructions

Satellite images mostly provide a top-down view of the Earth surface. Reconstructed 3D models can capture the rooftops but barely have any coverage for vertical facades. On the other hand, ground-level imagery, or street view images, provide detailed visual observation of the vertical building facades/walls, but no observation of building roofs.

More complete 3D models can be obtained if top-down satellite/aerial view and the ground-level imagery can be utilized together during the reconstruction process. However, since the two data modalities share very limited visual overlap, registering and fusing them together can be very hard.

7.2.3 Multi-Modal Machine Learning

Satellite images are registered to geographical coordinate systems, but only with limited registration accuracy. Many other data modalities, such as synthetic aperture radar (SAR) data, vectorized map data, coarse digital elevation models (DEM), and land usage data, have been collected all over the globe and registered geographically.

Such data, if utilized together, can achieve many useful and challenging tasks. For example, by combining high-resolution satellite images and low-resolution land usage classification map together, refined land usage classification map can be easily obtained and updated. By registering vectorized maps with raster images, they can be utilized as ground-truth supervision to enable many visual recognition tasks on satellite imagery. Using such registration as ground-truth supervision, machine learning models can be trained to update vectorized maps from raster image

observations. The Pixel2Pixel system (Isola et al., 2016) has demonstrated the preliminary success of such translation.

However, the accurate and automatic registration of geo-spatial data across different modalities is fundamental yet unsolved for all the aforementioned tasks. Many other factors other than inherent cross-modal differences make the registration problem harder, for example, different data might exhibit very different scales. Being able to register satellite imagery with other modalities enables many data-driven methods to be applied on satellite imagery without the tedious efforts of manual labeling.

Another potential direction is registration across the temporal domain. For example, with the frequent updates from satellite images, economical analysis for certain business entities can be easily obtained by counting the number of parked vehicles. Registration of the current visual observation and historical data can also benefit not only historic studies but also studies requiring long-term evidence such as climate changes. Intuitively, historical and modern observations should exhibit dramatic visual differences. How to establish reliable correspondences in the presence of such visual appearance changes is challenging but critical.

7.2.4 Machine Learning with Noisy Supervision

Long before Columbus started his course to discover America, people have been charting maps and recording/measuring Earth. Such manual annotation, in the modern age, continues in the form of online crowdsourcing maps, such as OpenStreetMaps (OpenStreetMap-Foundation, 2006).

Such annotation, if properly utilized, can enable many data-driven methods on geo-spatial datasets, including satellite imagery. Compared with many other visual recognition tasks, such as object detection and instance-aware segmentations, vectorized maps can be utilized as ground-truth guidances without the efforts of manual labeling. However, there are difficulties extruding the usage of such geo-spatial “free” annotations: (i) reliable and automatic cross-modality reg-

istration remains to be solved; (ii) crowd-sourced map suffers from inaccurate and inconsistent annotations.

Crowd-sourced map annotations are usually obtained by showing voluntary users an aerial/satellite image and letting the user provide annotations. Ensuring the quality of such labeling process is challenging. Thus the final obtained labeling can vary: for urban and popular areas, quality tends to be higher since more people pay attention. But for less interesting areas, annotations might be of low quality or not even exist.

To utilize such unreliable annotation for data-driven methods, machine learning models must explicitly model the uncertainty of the ground-truth data. Unsupervised or semi-supervised learning methods are promising approaches to better utilize such noisy annotations.

7.2.5 Systematic Design and Engineering

Good system design and engineering are very important aspects of successful applications. Large-scale 3D modeling techniques, as well as deep learning systems, all significantly benefited from excellent engineering design and implementation.

Considering the amount of potentially available data from satellite imaging platform and the ever-increasing update frequency, how to effectively store, query, and process such large amount of data, is a non-negligible direction of future efforts.

In addition to proposing more efficient algorithms, designing distributed algorithms to efficiently process large geospatial datasets to improve scalability can be of great significance when transforming academic prototypes into real-world productions.

Similar to the concept of “*level of details*” in computer graphics, different applications might require 3D models or outputs at different quality levels. How to effectively cache data at different levels and design algorithms and systems that can fully utilize such layered storage hierarchy can be of significant practical importance regarding real-world applications.

REFERENCES

- Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Communications of the ACM*.
- Anderson, R., Gallup, D., Barron, J. T., Kontkanen, J., Snavely, N., Hernández, C., Agarwal, S., and Seitz, S. M. (2016). Jump: Virtual reality video. In *SIGGRAPH Asia*.
- Bao, L., Yang, Q., and Jin, H. (2014). Fast edge-preserving patchmatch for large displacement optical flow. *IEEE Transactions on Image Processing*, 23(12):4996–5006.
- Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *Transactions on Graphics*.
- Barnes, C., Shechtman, E., Goldman, D. B., and Finkelstein, A. (2010). The generalized patch-match correspondence algorithm. In *European Conference on Computer Vision*.
- Barron, J. T. and Poole, B. (2016). The fast bilateral solver. In *European Conference on Computer Vision*.
- Besse, F., Rother, C., Fitzgibbon, A., and Kautz, J. (2013). Pmbp: Patchmatch belief propagation for correspondence field estimation. *International Journal of Computer Vision*.
- Bleyer, M., Rhemann, C., and Rother, C. (2011). Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference*.
- Bosch, M., Kurtz, Z., Hagstrom, S., and Brown, M. (2016). A multiple view stereo benchmark for satellite imagery. In *IEEE Applied Imagery Pattern Recognition (AIPR) Workshop*.
- Broadhurst, A., Drummond, T. W., and Cipolla, R. (2001). A probabilistic framework for space carving. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Chen, W., Xiang, D., and Deng, J. (2017a). Surface normals in the wild. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. (2017b). Multi-view 3d object detection network for autonomous driving. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Chen, Z., Sun, X., Wang, L., Yu, Y., and Huang, C. (2015). A deep visual correspondence embedding model for stereo matching costs. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*.
- Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer.
- Clark, R., Wang, S., Markham, A., Trigoni, N., and Wen, H. (2017a). Vidloc: A deep spatio-temporal model for 6-dof video-clip relocation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017b). Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *AAAI*.
- Collins, R. T. (1996). A space-sweep approach to true multi-image matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Crispell, D., Mundy, J., and Taubin, G. (2012). A variable-resolution probabilistic three-dimensional model for change detection. *IEEE Transactions on Geoscience and Remote Sensing*.
- Dai, J., He, K., and Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*.
- d’Ángelo, P. and Kusch, G. (2012). Dense multi-view stereo from satellite imagery. In *The IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*.
- De Franchis, C., Meinhardt-Llopis, E., Michel, J., Morel, J.-M., and Facciolo, G. (2014). An automatic and modular stereo pipeline for pushbroom images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):49.
- DeTone, D., Malisiewicz, T., and Rabinovich, A. (2017). Toward geometric deep slam. *arXiv:1707.07410 [cs.CV]*.
- Dial, G. and Grodecki, J. (2005). Rpc replacement camera models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.
- DigitalGlobe (2016). Spacenet on aws. <https://spacenetchallenge.github.io/>.

- DoD, U. S. (1984). World geodetic system 1984. https://en.wikipedia.org/wiki/World_Geodetic_System.
- Dollár, P. and Zitnick, C. L. (2013). Structured forests for fast edge detection. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1841–1848.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Duan, L. and Lafarge, F. (2016). Towards large-scale city reconstruction from satellites. In *European Conference on Computer Vision*.
- Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*.
- Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., et al. (2010). Building rome on a cloudless day. In *European Conference on Computer Vision*.
- Fraser, C. S. and Hanley, H. B. (2003). Bias compensation in rational functions for ikonos satellite imagery. *Photogrammetric Engineering & Remote Sensing*.
- Furukawa, Y., Hernández, C., et al. (2015). Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*.
- Galliani, S., Lasinger, K., and Schindler, K. (2015). Massively parallel multiview stereopsis by surface normal diffusion. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Galliani, S. and Schindler, K. (2016). Just look at the image: Viewpoint-specific surface normal prediction for improved multi-view reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q., and Pollefeys, M. (2007). Real-time plane-sweeping stereo with multiple sweeping directions. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE.
- Garg, R., B.G., V. K., Carneiro, G., and Reid, I. (2016). Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer.
- Gehrke, S., Morin, K., Downey, M., Boehrer, N., and Fuchs, T. (2010). Semi-global matching: An alternative to lidar for dsm generation. In *Proceedings of the 2010 Canadian Geomatics Conference and Symposium of Commission I*.

- Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient large-scale stereo matching. In *Asian Conference on Computer Vision*, pages 25–38. Springer.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*.
- Godard, C., Mac Aodha, O., and Brostow, G. J. (2017). Unsupervised monocular depth estimation with left-right consistency. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Grodecki, J. and Dial, G. (2003). Block adjustment of high-resolution satellite images described by rational polynomials. *Photogrammetric Engineering & Remote Sensing*.
- Gueguen, L. and Hamid, R. (2015). Large-scale damage detection using satellite imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1321–1328.
- Haeusler, R., Nair, R., and Kondermann, D. (2013). Ensemble learning for confidence measures in stereo vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.
- Hartley, R. I. and Saxena, T. (1997). The cubic rational polynomial camera model. In *Image Understanding Workshop*.
- Hartmann, W., Galliani, S., Havlena, M., Van Gool, L., and Schindler, K. (2017). Learned multi-patch similarity. In *The IEEE International Conference on Computer Vision (ICCV)*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015b). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Heinly, J., Schonberger, J. L., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Heise, P., Klose, S., Jensen, B., and Knoll, A. (2013). Pm-huber: Patchmatch with huber regularization for stereo matching. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Hirschmüller, H., Buder, M., and Ernst, I. (2012). Memory efficient semi-global matching. *ISPRS*.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Hornáček, M., Besse, F., Kautz, J., Fitzgibbon, A., and Rother, C. (2014). *Highly Overparameterized Optical Flow Using PatchMatch Belief Propagation*, pages 220–234. Springer International Publishing, Cham.
- Hosni, A., Bleyer, M., and Gelautz, M. (2013). Secrets of adaptive support weight techniques for local stereo matching. *Computer Vision and Image Understanding*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861 [cs]*.
- Hu, Y., Song, R., and Li, Y. (2016). Efficient coarse-to-fine patchmatch for large displacement optical flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Hu, Y., Tao, V., and Croitoru, A. (2004). Understanding the rational function model: methods and applications. *International Archives of Photogrammetry and Remote Sensing*.
- Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2016). Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*.
- Izadi, M. and Saeedi, P. (2012). Three-dimensional polygonal building model estimation from single satellite images. *IEEE Transactions on Geoscience and Remote Sensing*.

- Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., and Aanaes, H. (2014). Large scale multi-view stereopsis evaluation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ji, M., Gall, J., Zheng, H., Liu, Y., and Fang, L. (2017). Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Kapur, D., Saxena, T., and Yang, L. (1994). Algebraic and geometric reasoning using dixon resultants. In *Proceedings of the international symposium on Symbolic and algebraic computation*, pages 99–107. ACM.
- Kar, A., Hane, C., and Malik, J. (2017). Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*.
- Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946.
- Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., and Bry, A. (2017). End-to-end learning of geometry and context for deep stereo regression. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Kim, S., Park, K., Sohn, K., and Lin, S. (2016). *Unified Depth Prediction and Intrinsic Image Decomposition from a Single Image via Joint Convolutional Neural Fields*, pages 143–159. Springer International Publishing, Cham.
- Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
- Knobelreiter, P., Reinbacher, C., Shekhovtsov, A., and Pock, T. (2017). End-to-end training of hybrid cnn-crf models for stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515. IEEE.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.
- Kutulakos, K. N. and Seitz, S. M. (2000). A theory of shape by space carving. *International Journal of Computer Vision*.

- Kuznetsov, Y., Stuckler, J., and Leibe, B. (2017). Semi-supervised deep learning for monocular depth map prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ladicky, L., Shi, J., and Pollefeys, M. (2014). Pulling things out of perspective. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., and Torr, P. H. S. (2011). Joint optimization for object class segmentation and dense stereo reconstruction. *International Journal of Computer Vision*.
- Ley, A., Hänsch, R., and Hellwich, O. (2016). Syb3r: A realistic synthetic benchmark for 3d reconstruction from images. In *European Conference on Computer Vision*, pages 236–251. Springer.
- Li, B., Shen, C., Dai, Y., van den Hengel, A., and He, M. (2015). Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Li, R., Wang, S., Long, Z., and Gu, D. (2017). Undeepvo: Monocular visual odometry through unsupervised deep learning. *arXiv preprint arXiv:1709.06841*.
- Li, X. and Belaroussi, R. (2016). Semi-dense 3d semantic mapping from monocular slam. *arXiv preprint arXiv:1611.04144*.
- Li, X., Wu, C., Zach, C., Lazebnik, S., and Frahm, J.-M. (2008). Modeling and recognition of landmark image collections using iconic scene graphs. In *European Conference on Computer Vision*, pages 427–440. Springer.
- Li, Z., Gan, Y., Liang, X., Yu, Y., Cheng, H., and Lin, L. (2016). Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *European Conference on Computer Vision*, pages 541–557. Springer.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European Conference on Computer Vision*.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*.
- Luo, W., Schwing, A. G., and Urtasun, R. (2016). Efficient deep learning for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Ma, F. and Karaman, S. (2017). Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *arXiv preprint arXiv:1709.07492*.
- Manocha, D. (1994). Solving systems of polynomial equations. *IEEE Computer Graphics and Applications*, 14(2):46–55.
- Marin, G., Zanuttigh, P., and Mattoccia, S. (2016). Reliable fusion of tof and stereo depth driven by confidence measures. In *European Conference on Computer Vision*, pages 386–401. Springer.
- Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Menze, M. and Geiger, A. (2015). Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Mishkin, D., Matas, J., and Perdoch, M. (2015). Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*.
- Mostegel, C., Rumpler, M., Fraundorfer, F., and Bischof, H. (2016). Using self-contradiction to learn confidence measures in stereo vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- NASA, U. . (1972). Landsat program. <https://landsat.usgs.gov/>.
- Ok, A. O., Senaras, C., and Yuksel, B. (2013). Automated detection of arbitrarily shaped buildings in complex environments from monocular vhr optical satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(3):1701–1717.
- OpenStreetMap-Foundation (2006). Openstreetmap. <https://www.openstreetmap.org>.
- Ortner, M., Descombes, X., and Zerubia, J. (2007). Building outline extraction from digital elevation models using marked point processes. *International Journal of Computer Vision*.
- Ozcanli, O. C., Dong, Y., Mundy, J. L., Webb, H., Hammoud, R., and Victor, T. (2014). Automatic geo-location correction of satellite imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 307–314.
- Padwick, C., Deskevich, M., Pacifici, F., and Smallwood, S. (2010). Worldview-2 pan-sharpening. In *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA*, volume 2630.
- Park, M.-G. and Yoon, K.-J. (2015). Leveraging stereo matching with learning-based confidence measures. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Pfeiffer, D., Gehrig, S., and Schneider, N. (2013). Exploiting the power of stereo confidences. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Poggi, M. and Mattoccia, S. (2017). Learning to predict stereo reliability enforcing local consistency of confidence maps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Poggi, M., Tosi, F., and Mattoccia, S. (2017). Quantitative evaluation of confidence measures in a machine learning world. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Pollard, T. B., Eden, I., Mundy, J. L., and Cooper, D. B. (2010). A volumetric approach to change detection in satellite images. *Photogrammetric Engineering & Remote Sensing*, 76(7):817–831.
- Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., and Guibas, L. J. (2016). Volumetric and multi-view cnns for object classification on 3d data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ranftl, R., Vineet, V., Chen, Q., and Koltun, V. (2016). Dense monocular depth estimation in complex dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2015). Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1172.
- Richter, S. R., Hayder, Z., and Koltun, V. (2017). Playing for benchmarks. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Richter, S. R., Vineet, V., Roth, S., and Koltun, V. (2016). Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer.
- Riegler, G., Osman Ulusoy, A., and Geiger, A. (2017a). Octnet: Learning deep 3d representations at high resolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Riegler, G., Ulusoy, A. O., Bischof, H., and Geiger, A. (2017b). Octnetfusion: Learning depth fusion from data. In *Proceedings of the International Conference on 3D Vision*.
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Roy, A. and Todorovic, S. (2016). Monocular depth estimation using neural regression forest. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *The IEEE International Conference on Computer Vision (ICCV)*.

- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*.
- Saxena, A., Sun, M., and Ng, A. Y. (2009). Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42.
- Scharstein, D. and Szeliski, R. (2003). High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.
- Schlinger, R., Marshall, W., and Boshuizen, C. (2010). Planet labs. <https://www.planet.com/>.
- Schönberger, J. L., Zheng, E., Pollefeys, M., and Frahm, J.-M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*.
- Schops, T., Schönberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., and Geiger, A. (2017). A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Seitz, S. M. and Dyer, C. R. (1999). Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*.
- Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2017). Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*.
- Shah, V. P., Younan, N. H., and King, R. L. (2008). An efficient pan-sharpening method via a combined adaptive pca approach and contourlets. *IEEE transactions on geoscience and remote sensing*, 46(5):1323–1335.
- Shaked, A. and Wolf, L. (2017). Improved stereo matching with constant highway networks and reflective confidence learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Shelhamer, E., Long, J., and Darrell, T. (2016). Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Sinha, S. N., Scharstein, D., and Szeliski, R. (2014). Efficient high-resolution stereo matching using local plane sweeps. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics*.

- Snavely, N., Seitz, S. M., and Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210.
- Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3d shape recognition. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Sun, X., Christoudias, C. M., and Fua, P. (2014). Free-shape polygonal object localization. In *European Conference on Computer Vision*.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tao, C. V. and Hu, Y. (2002). 3d reconstruction methods based on the rational function model. *Photogrammetric Engineering & Remote Sensing*, 68(7):705–714.
- Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Tonioni, A., Poggi, M., Mattoccia, S., and Di Stefano, L. (2017). Unsupervised adaptation for deep stereo. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Tulyakov, S., Ivanov, A., and Fleuret, F. (2017). Weakly supervised learning of deep metrics for stereo reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., and Brox, T. (2017). Demon: Depth and motion network for learning monocular stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Vanegas, C. A., Aliaga, D. G., and Benes, B. (2012). Automatic extraction of manhattan-world building masses from 3d laser range scans. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1627–1637.
- Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., and Fragkiadaki, K. (2017). Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*.
- Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2017). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.
- Vivone, G., Alparone, L., Chanussot, J., Dalla Mura, M., Garzelli, A., Licciardi, G. A., Restaino, R., and Wald, L. (2015). A critical comparison among pansharpening algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2565–2586.

- Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., and Cremers, D. (2017). Image-based localization using lstms for structured feature correlation. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Wallack, A. and Manocha, D. (1998). Robust algorithms for object localization. *International Journal of Computer Vision*, 27(3):243–262.
- Wang, K., Bansal, M., and Frahm, J.-M. (2017a). Retweet wars: Tweet popularity prediction via multimodal regression. In *Advances in Neural Information Processing Systems (NIPS) Workshop*.
- Wang, K., Bansal, M., and Frahm, J.-M. (2018a). Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- Wang, K., Dunn, E., Rodriguez, M., and Frahm, J.-M. (2016a). Bringing 3d models together: Mining video liaisons in crowdsourced reconstructions. In *Asian Conference on Computer Vision*.
- Wang, K., Dunn, E., Rodriguez, M., and Frahm, J.-M. (2018b). Efficient video collection association using geometry-aware bag-of-iconics representations. *IPSJ Transactions on Computer Vision and Applications*, 31:325–337.
- Wang, K., Dunn, E., Tighe, J., and Frahm, J.-M. (2014a). Combining semantic scene priors and haze removal for single image depth estimation. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- Wang, K. and Frahm, J.-M. (2017a). Fast and accurate satellite multi-view stereo using edge-aware interpolation. In *The IEEE International Conference on 3D Vision (3DV)*.
- Wang, K. and Frahm, J.-M. (2017b). Single image parametric building model estimation from satellite imagery. In *The IEEE International Conference on 3D Vision (3DV)*.
- Wang, K., Stutts, C., Dunn, E., and Frahm, J.-M. (2016b). Efficient joint stereo estimation and land usage classification for multiview satellite data. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.
- Wang, S., Clark, R., Wen, H., and Trigoni, N. (2017b). Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050.
- Wang, X., Fouhey, D., and Gupta, A. (2015). Designing deep networks for surface normal estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wang, Y., Wang, K., Dunn, E., and Frahm, J.-M. (2014b). Stereo under sequential optimal sampling: A statistical analysis framework for search space reduction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Weier, J. and Herring, D. (2000). Measuring vegetation (ndvi & evi). <https://earthobservatory.nasa.gov/Features/MeasuringVegetation/>.
- Weng, Q. (2002). Land use change analysis in the zhujiang delta of china using satellite remote sensing, gis and stochastic modelling. *Journal of environmental management*, 64(3):273–284.
- Wu, J., Ma, L., and Hu, X. (2017). Delving deeper into convolutional neural networks for camera relocalization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5651.
- Xu, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. (2017). Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Yang, J., Fu, X., Hu, Y., Huang, Y., Ding, X., and Paisley, J. (2017a). Pannet: A deep network architecture for pan-sharpening. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE.
- Yang, T.-Y., Hsu, J.-H., Lin, Y.-Y., and Chuang, Y.-Y. (2017b). Deepcd: Learning deep complementary descriptors for patch representations. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. (2016). LIFT: Learned Invariant Feature Transform. In *Proceedings of the European Conference on Computer Vision*.
- Yoon, K.-J. and Kweon, I. S. (2006). Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.
- Zbontar, J. and LeCun, Y. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*.
- Zhang, X., Zhou, X., Lin, M., and Sun, J. (2017). Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv:1707.01083 [cs]*.
- Zheng, E., Dunn, E., Jojic, V., and Frahm, J.-M. (2014). Patchmatch based joint view selection and depthmap estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Zheng, E., Wang, K., Dunn, E., and Frahm, J.-M. (2015). Minimal solvers for 3d geometry from satellite imagery. In *The IEEE International Conference on Computer Vision (ICCV)*.

- Zhou, C., Zhang, H., Shen, X., and Jia, J. (2017a). Unsupervised learning of stereo matching. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Zhou, Q.-Y. and Neumann, U. (2008). Fast and extensible building modeling from airborne lidar data. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 7. ACM.
- Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017b). Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.