University of California
Santa Barbara

Automated Indoor Mapping With Point Clouds

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Geography

by

Jorge Chen

Committee in charge:
Professor Keith C. Clarke, Chair
Professor Alan Murray
Professor Marcos Novak

January 2018

The Dissertation of Jorge Chen is approved.

Alan Murray

Marcos Novak

Keith C. Clarke, Committee Chair

December 2017

Automated Indoor Mapping With Point Clouds

by

Jorge Chen

To my wife and daughters

Acknowledgements

I would first like to thank the financial supporters who made my journey through the Ph.D. program a reality. These include the U.S. National Geospatial-Intelligence Agency for funding the last year of my studies as well as two follow-on years as a postdoc[1]; the UCSB Graduate Division for providing the Graduate Opportunity Fellowship during my first year; the UCSB Department of Geography for providing assistantships through the years; the Ralph M. Parsons Foundation for its graduate student fellowship in 2015; and Uncle Sam for the Post-9/11 G.I. Bill as well as funding a big chunk of my education prior to coming to UCSB. These funding sources together made it financially feasible to return to school while supporting a growing family, with two new little ones who joined along the way.

While the road to the Ph.D. involved a great deal of independent study and research, it also involved many who helped guide, teach, and encourage along the way. First and foremost, I'd like to thank the members of my committee: Prof. Keith C. Clarke, who brought me on board as his student, funded some early experiments, and provided valuable mentorship on becoming a professional researcher and academic — seldom providing the easy answer but always showing the way; Prof. Marcos Novak, who inspired me to look beyond the readily apparent through his Transvergence seminars — some of my most memorable times at UCSB were during those thought-provoking discussions; and finally Prof. Alan Murray, who gladly joined my committee at the last minute and provided valuable guidance on working through an impasse in my research. Then there are my lab mates: Dr. Ian Irmischer, my office mate, who not only provided valuable discussions on all things academic but also moral support during difficult times; Dr. Kitty Currier,

[1]NURI grant: "Automated Level of Detail and Feature Extraction for 3D Indoor Mapping and Modeling"

who got me started with SfM-MVS and introduced me to kite aerial photography; and (soon to be) Dr. Bo Romero, who was just always there to chat about anything.

I'd also like to thank my former professors at California State University, Northridge — it was truly an outstanding education: Prof. Jim Craine, Prof. Regan Maas, Prof. Shawna Dark, and Prof. Soheil Boroushaki, all of whom prepared me for returning to academia and helped pave the way to UCSB. This list would be incomplete without mentioning two other professors who perhaps had the greatest influence on my aspiration towards a Ph.D.: Profs. Greg and Wita Wojtkowski, husband and wife, at Boise State University, who taught the evening classes I attended while stationed in rural Idaho with the Air Force. They lit the fire of engaging in scholarly inquiry through not only their classes but also their friendship and mentoring outside the university. They were the first to suggest doing a Ph.D. an entire decade before I decided to leave industry and return to school for the doctorate.

Finally and most importantly, I would like to thank my family. To my wife, Szu-Jung, who has been my constant helpmate, always bringing joy and support, always keeping me grounded, and always reminding me to eat and exercise. Also to my two daughters, Hsin-En and Yi-En, who bring delight and happiness to each new day. You are each a blessing from God and a reminder that there's far more to life than just work... thank you for always being there even when I've had to disappear for hours on end working on the Ph.D. Then there is my brother, Dr. Jaime Chen, MD, who has always encouraged me to make the most of my abilities; my cousin, Prof. Roy Chen, who inspired me with his mid-career move to academia despite monumental challenges; and my cousin, Prof. Grace Wang, who, along with Roy, provided guidance early on in the journey. Finally, I would like to thank my parents, who took the long journey to the U.S. with two little boys and their meager life savings and through God's providence, hard work, and tremendous personal sacrifice made all this possible. 多謝老爸老媽 for everything!

Curriculum Vitæ

Jorge Chen

Education and Professional Registration

2018	Ph.D. in Geography, University of California, Santa Barbara
2012	M.A. in Geography, California State University, Northridge
2004	M.S. in Civil Engineering, University of California, Los Angeles
1999	M.S. in Engineering Management, University of Missouri-Rolla
1996	B.S. in Civil Engineering, United States Air Force Academy
2004	Professional Civil Engineer, State of California

Publications

2018	Chen, Jorge. 2018. Grid Referencing of Buildings. In *Proceedings of the 14th Conference on Location Based Services*. Zurich, Switzerland.
2017	Chen, Jorge. 2017. *Air Force Reserve S-Team History: 1978 to 2014*. Robins Air Force Base, Georgia: Headquarters United States Air Force Reserve.
2017	Chen, Jorge, and Keith C. Clarke. 2017. Modeling Standards and File Formats for Indoor Mapping. In *Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management* - Volume 1, 26875. Porto, Portugal.
2016	Chen, Jorge, and Keith C. Clarke. 2016. Rapid 3D Modeling Using Photogrammetry Applied to Google Earth. In *The 21st International Research Symposium on Computer-Based Cartography and GIScience*. Albuquerque, New Mexico.

Academic Employment

2017	Graduate Student Researcher, "Automated Level of Detail and Feature Extraction for 3D Indoor Mapping and Modeling," funded by a grant from the National-Geospatial Intelligence Agency (NGA) University Research Initiative, UCSB
2015	Teaching Assistant, Geog W12: Maps and Spatial Reasoning, UCSB
2014	Graduate Student Researcher, developed the first fully-online version of Geog W12, UCSB
2013	Teaching Assistant, Geog 176A: Introduction to Geographic Information Systems, UCSB

2011–2012	Graduate Research Assistant, developed geodatabase for the California State Water Board, CSUN
2011–2012	Graduate Research Assistant, developed a campus web map implemented on ArcGIS Server, CSUN

Research Grants

2017–2019	National Geospatial-Intelligence Agency University Research Initiative (NURI), "Automated Level of Detail and Feature Extraction for 3D Indoor Mapping & Modeling." Co-authored proposal with Dr. Keith C. Clarke (PI).

Honors and Awards

2015	Ralph M. Parsons Foundation Fellowship
2012–2013	UCSB Graduate Opportunity Fellowship
2001	Computer Information Systems Scholarship, Boise State University
1998–1999	U.S. Air Force Graduate Civilian Institute Scholarship
1996	Distinguished Graduate with Academic Distinction, United States Air Force Academy

Military Service

2015–Present	Officer Historian, United States Air Force Reserve
2008–2015	Civil Engineer Officer, United States Air Force Reserve
2004–2008	Civil Engineer Officer, California Air National Guard
1996–2002	Civil Engineer Officer, United States Air Force

Abstract

Automated Indoor Mapping With Point Clouds

by

Jorge Chen

This dissertation examines the current state of automated indoor mapping and modeling using point cloud data produced by close range remote sensing systems. The first part looks at reality capture techniques that convert the physical form of indoor spaces into *point clouds* of millions of measured points, each with an (x, y, z) coordinate value. The second part examines methods for teasing out geometries from these point clouds — often complicated by noise and voids — and converting them into 3D geometric models. The final part examines techniques for merging the coordinate reference systems of these indoor maps and models with those of the outdoor world, resulting in a seamless representation of space. Lessons learned in this study revealed that theories, techniques, and practices in indoor mapping remain relatively elementary compared to those for the outdoors, yet they also present significant opportunities for future research propelled by emerging developments in remote sensing and a growing demand for indoor maps.

Contents

Part I

Introduction

Chapter 1

Mapping the Great Indoors

1.1 Research vision

This dissertation examines basic concepts and practices for automatically generating 3D maps of indoor spaces from point clouds — large collections of *xyz* coordinates corresponding to the locations of surfaces. The three main areas of interest include automation (i.e., minimizing human involvement), the exploitation of point cloud data, and the mapping and modeling of interior spaces. Unlike the thoroughly researched and highly-developed field of outdoor mapping, indoor mapping and modeling has remained in a relatively primitive state up until very recent times. This study covers only the basic elements of this emerging field and provides a starting point for follow-on research beyond the dissertation.

1.2 Why indoors?

Maps have helped people navigate and understand the world around them since prehistoric times (Harley and Woodward 1987; Clarke 2013). Nearly all of these maps

depicted the "Great Outdoors" with few exceptions, and their various forms and numbers have more recently proliferated with revolutions in digital computing, high speed communications networks, air- and space-borne remote sensing, and global positioning technology. Maps can now be made with remarkable precision and accuracy for nearly every place on Earth with one exception . . . the "Great Indoors." Until very recently, indoor maps have remained in a relatively unsophisticated two-dimensional line drawing form as they have for millennia (see Section 1.3). However, advancements in close-range remote sensing, the processing of remotely sensed data, and computer graphics have now significantly reduced the barriers that once prevented the creation and use of *three-dimensional* indoor maps.

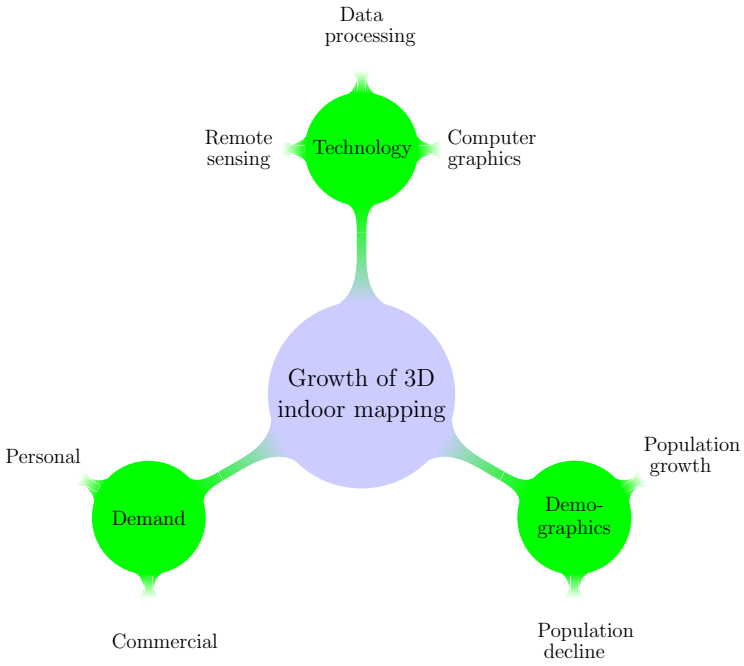


Figure 1.1: Factors influencing the growth of 3D indoor mapping

A cursory look at current and future trends in demographics, demand for indoor mapping data, and technology point to significant future growth for the use of 3D indoor maps

as illustrated in Figure 1.1. Among these, demographics stands out as a notable catalyst for growth. In a 2014 study, the United Nations estimated that the global population will grow by about 2.3 billion people over the next four decades, representing a one-third increase in the total population in 2015, with most of this growth occurring in newly urbanized areas in Asia and Africa (United Nations Population Fund 2014; United Nations 2014). This growth will fuel a boom in housing and other urban structures that "could roughly equal the entire volume of such construction to date in world history" (National Intelligence Council 2012). Growing populations and diminishing resources means that more and more of these future populations will inhabit ever-larger structures that range from high-rise apartment buildings to mega-sized shopping centers — something already happening in many parts of China. While grand mega-structures have existed in the past, they will become more commonplace in the future resulting in a growing need for more effective methods of indoor navigation.

The booming population will also fuel a booming marketplace with many potential uses for 3D indoor maps. The most obvious application will involve the extension of location-based services (LBS) from the outdoors to the indoors, first in 2D then expanding to 3D, for activities such as location-aware marketing, finding products within stores, and the automated delivery of mail and packages using robots. Less obvious will be applications in back-end activities powering that marketplace, such as the use of map-aware autonomous robots for managing retail space and warehouse inventories.

Highly developed nations, however, will experience declining but aging populations, which will drive a different set of needs (United Nations 2017). A declining workforce will likely result in greater reliance on automation, not just for commerce as just mentioned but also for care of the aged and elderly. Indoor maps can play an important role with assistive technologies such as monitoring of the elderly — many of whom will likely live alone due to declining birthrates — and map-enabled autonomous wheelchairs.

Many elements of the third factor, indoor mapping technology, already exist for making primitive 3D indoor maps. As mentioned earlier, these include close-range remote sensing technology, techniques for processing the remotely sensed data, and theories and techniques in computer graphics. Noteworthy future trends include the miniaturization and commoditization of 3D sensors, improved methods for extracting geometric features from sensed data, and the growing adoption of augmented reality and virtual reality systems. This dissertation focuses on the second trend in technology — the extraction of 3D geometric features from remotely sensed data.

1.3 A brief history of indoor maps

Utrilla et al. (2009) examined one of the earliest known *outdoor* maps found engraved in Abauntz Cave, Spain, dating back to 13,660 calBP. Clarke (2013) also surveyed ancient maps and contended that humans began mapping at a much earlier time in history, going as far back as 55,000 calBP. This earlier date coincides with the presumed start of worldwide human migration and the presumed development of higher functioning cognitive abilities, which included the ability to think spatially. Figure 1.2 provides a brief visual survey of outdoor maps through the ages. A comprehensive study on the history of maps can be found in *The History of Cartography* series (Harley and Woodward 1987).

While most prehistoric and ancient maps depicted the outdoor world, archaeologists have also unearthed maps of indoor spaces that date back to ancient times. Talbert (2014) documented archaeological finds at ancient Girsu — an ancient Mesopotamian city — that showed floor plans dating to c.3,000 B.C. The Forma Urbis Romae — a massive 18.1 m by 13 m map etched in marble — served as a map of the city of Rome during the third century A.D. with enough detail to show building interiors (Taub 1993).

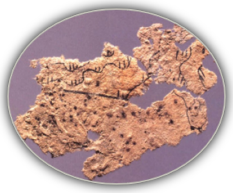
Indoor maps gained greater prominence during the Age of Enlightenment in Europe in the form of architectural and technical drawings (Hewitt 1985), which coincided with the onset of industrialization in Western societies and accompanying urbanization. In due time, continued advancements in civil engineering would enable the construction of ever-larger structures to accommodate these masses of people — e.g., skyscrapers, subways, underground cities, etc. — and these large structures would often use indoor maps to help occupants with indoor navigation. Figure 1.3 provides a brief look at historic indoor maps through the ages.

(a) Abauntz Cave, Spain, c.11,710 BC.

(b) Catalhoyuk, Turkey, c.7,000 BC.

(c) World Map, Babylon, c.600 BC.

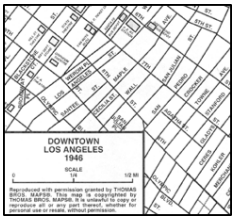
(d) Anaximander's Map, c.575 BC.



(e) Qin Kingdom, China, c.250 BC.

(f) Hondias World Map, 1630.

(g) St. Denis, France, 1706.



(h) Thomas Bros., Los Angeles, 1947

Figure 1.2: Outdoor maps throughout history[1]

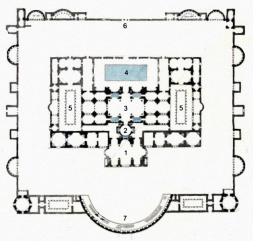
[1]a. Utrilla et al. 2009; b. Clarke 2013; c. British Museum, n.d.(b); d. Saint-Pol 2006; e. Mobojiang 2009; f. Hondius II 1631; g. Stones 2006b; h. Metcalfe, n.d.

1.4 Defining indoor space

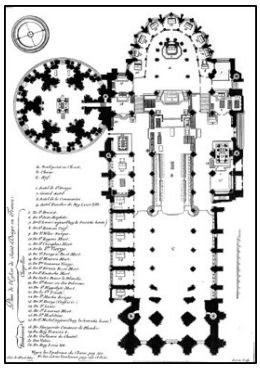
What differentiates indoor from outdoor space? Li (2016, 2008) writing on behalf of the IndoorGML working group, defined indoor space as locations contrained by "architectural components, such as doors, corridors, floors, walls, and stairs," while outdoor space faced no such constraints. Giudice, Walton, and Worboys (2010) noted that indoor space contains regular geometries and poses challenges to conventional outdoor mapping techniques, e.g., geographic (latitude-longitude) coordinates and multi-level routing. Walton and Worboys (2010) provided another perspective based on whether spatial elements are built or natural, their level of enclosure, and the viewability of objects within a certain visual range.

(a) Girsu floor plan, c.3,000 BC

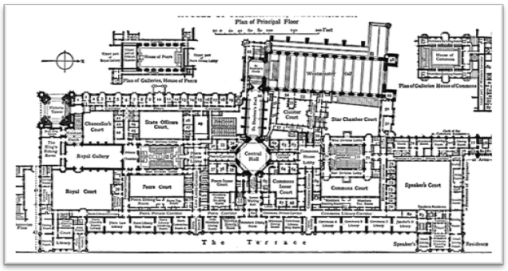
(b) Ningirsu Temple plan, c.2,150 BC



(c) Forma Urbis Romae, c.210



(d) Abbey of St. Denis, France, c.1700



(e) House of Parliament, England, c.1800s

(f) Westfield Culver City, Google Maps, 2015

Figure 1.3: Indoor maps throughout history[2]

Each definition touches on different aspects of indoor space, the most obvious and universal being a state of total enclosure and isolation from the outside world. Beyond that, however, attempts to capture the nature of indoor space branch off into domain-specific paradigms. For instance, the three sources just referenced approach the subject from the standpoint of buildings, e.g., presence of architectural components, regular geometry, and visual range. Walton and Worboys take this a step further by also considering suitable coordinate reference systems and computerized routing techniques. But these definitions quickly break down outside of the typical building context; for example, caves possess no built-up elements nor regular geometry, the vast length of some megastructures can exceed the limits of human vision, and large open-air courtyards and plazas straddle the indoor-outdoor divide (Figure 1.4).

(a) Boeing Everett Factory

(b) Carlsbad Caverns

(c) Kyoto Imperial Palace

Figure 1.4: Spaces that defy conventional definitions of indoor space[3]

The recalcitrant nature of some indoor spaces means that the concept of *indoors* has as much a constructivist nature as it does essentialist, characterized not only by definitions but also by usage (Davidson 2008). For instance, the Alaskan

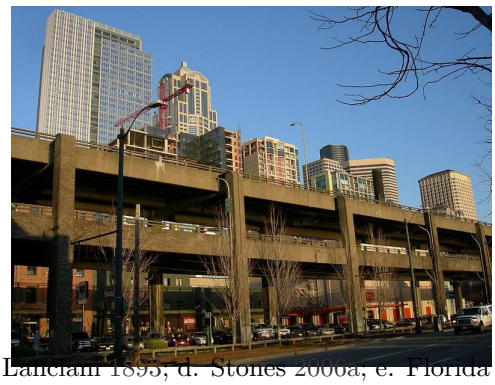


Figure 1.5: Alaskan Way Viaduct[4]

[2]a. Musée du Louvre, n.d.; b. British Museum, n.d.(a); c. Lanciani 1893; d. Stones 2006a; e. Florida Center for Instructional Technology, n.d.; f. Google 2015

[3]a. Jetstar Airways 2013; b. Chen 2006; c. Chen 2011

[4]Mabel 2008

Way Viaduct running through Seattle, Washington, shown in Figure 1.5, has many of the elements of indoor space using the earlier definitions — built-up structure, floors, ceilings, low walls, resistant to multi-level 2D computerized routing, etc. — yet it is a transportation corridor that functions more as an outdoor bridge than a building. Nonetheless, if this section of highway ever closed down, the same structure could be converted into waterfront office space, a shopping center, or a parking garage. Here, usage would result in a change of semantics.

Instead of attempting to put forward an authoritative definition of indoor space, this dissertation will use the rules-of-thumb shown in Figure 1.6 to distinguish indoor from outdoor spaces, with the added criteria that these spaces generally do not include transportation corridors.[5] The remainder of this dissertation will focus exclusively on indoor spaces in the context of buildings.

[5]Even this is an imperfect definition since many people would consider a long highway tunnel as indoor space.

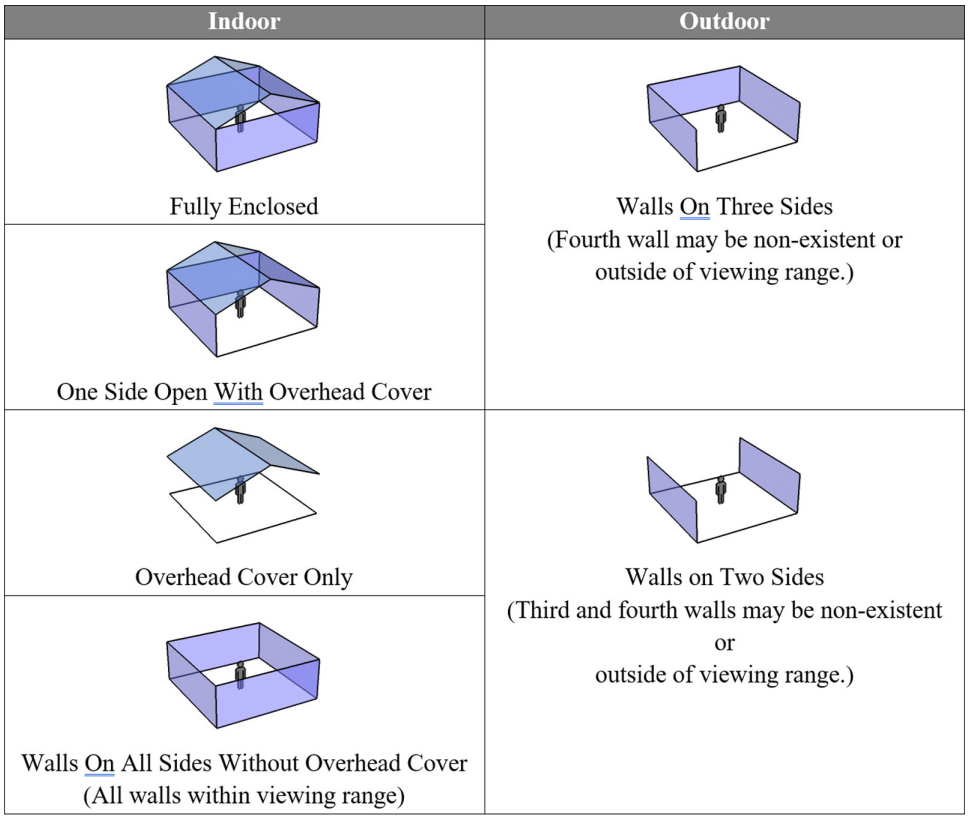


Figure 1.6: Proposed definitions for indoor and outdoor spaces.

1.5 Defining indoor maps

While seemingly simple in appearance, maps represent complex instruments of language that form a sort of spatial linguistics (National Research Council 2006). Bertin (2010) called this a "'language' for the eye" and noted that graphic representation "constitutes one of the basic sign systems conceived by the human mind for the purposes of storing, understanding and communicating essential information." Since maps — in the language sense — transcend the indoor-outdoor divide, an understanding of the two major schools of thought in cartography can help shape the future form and usage of indoor maps. These two schools of thought consist of the cartography as communication and

analytical cartography paradigms (Robinson and Petchenik 1975; Nyerges 1991; Kitchin, Dodge, and Perkins 2011).

The *cartography as communication* paradigm emerged in the mid-twentieth century in response to the positivist approach to mapping that accompanied the quantitative revolution in geography and was advanced by the works of geographers such as Robinson (1952), Board (1967), Robinson and Petchenik (1975), and Morrison (1978). It focused on the communication process from the perspectives of both the cartographer and the map reader, and it later expanded into the constructivist realm of human geography, exploring issues such as place and power structures (Kitchin, Dodge, and Perkins 2011). The other school of thought, *analytical cartography*, was spearheaded by Waldo Tobler and looked at maps as tools for spatial analysis (Tobler 1959, 1976)[6]. Nyerges (1991) proposed that these two cartographic concepts represented the two end points of a complementary, rather than competing, continuum. While these concepts were originally developed for outdoor maps, their ideas transcend the indoor-outdoor divide and can provide useful models for understanding mapping of the indoors.

Indoor maps can thus represent tools for symbolizing the geometry of indoor spaces for purposes of communication or spatial analysis. Another term frequently used for the geometric representation of indoor space is *indoor modeling*, such as in the combined term "indoor mapping and modeling" used by Zlatanova et al. (2013). Since models provide representations of things in the real world, indoor maps can thus be seen as a subset of models used for communicating or analyzing the spatial nature of indoor space in terms of both form and usage.

[6]Clarke and Cloud (2000) provide a fascinating history on the appropriation of state-of-the-art cartographic techniques from Nazi Germany at the end of World War II.

1.6 3D indoor maps of the future

With development still in a formative stage, indoor mapping provides many challenges and opportunities for future research and development. One of the most intriguing and thought-provoking areas involves mapping in all three dimensions (3D) — something that outdoor mapping had never fully developed for practical purposes due largely to technological limitations. However, the advent of small powerful computers, augmented reality systems, and the miniaturization and commoditization of high-accuracy remote sensing systems have now made true 3D mapping possible for both indoors and out.

For visualization, 3D indoor maps can function as fully immersive models or as the source material for augmented reality systems. 3D maps can also find uses in indoor navigation when paired with emerging indoor localization technologies; whereas indoor positioning provides location in space, indoor maps provide the spatial context to make location meaningful. Another potential area is in supply chain management. 3D indoor maps coupled with other indoor technologies have the potential to fully automate activities such as warehousing and long assembly lines using indoor map-enabled robots. Marketing channels[7] can also benefit from 3D indoor maps (Kotler and Keller 2006). The growth of online shopping has resulted in the construction of enormous fulfillment centers staffed mostly by humans who do the picking, sorting, and packing of orders; 3D indoor maps can provide the spatial component for automating this entire process with robots.

3D indoor maps can also serve important social purposes. They can be used in rescue operations involving either very large structures or dangerous indoor environments, such as the capsizing of the Costa Concordia cruise ship or the meltdown of the Fukushima

[7] I.e., the last leg of the supply chain that takes a product from manufacturing to the customer.

Daiichi Power Plant in Japan. Indoor maps can also serve as the framework for studying people and indoor environments as large metropolises begin to form in indoor spaces. These ideas only represent a small fraction of the possible applications of 3D indoor maps; the rest is only limited by the imagination.

1.7 Goal and organization of this dissertation

This dissertation explores a selection of key principles relevant to the automatic generation of 3D indoor maps and models from point clouds. These principles fall under the three general steps of the modeling work flow consisting of *point cloud acquisition*, *feature extraction*, and *indoor-outdoor coordinate integration* that form the basis of parts two to four of this disseration (Figure 1.7), with the added constraint of working with rectangularly shaped (cuboidal) indoor spaces.

Each part begins with a chapter covering general concepts followed by more detailed treatment of key topics and applications with case studies, as illustrated in Figure 1.7. Part II covers the first step of modeling, which involves taking automated measurements of the environment using either laser scanning or photogrammetry to produce point clouds. Its two case studies document early experiments with photogrammetry. With the point cloud acquired, Part III examines the process for classifying points based in terms of the structural features of a room (i.e., floor, ceiling, and walls) before converting the point cloud into simplified geometric form, e.g., planes. Part III's case study improves on a previously developed approach for automatically deriving cuboidal geometric forms from rectangular point clouds, while a separate chapter examines detailed information on standards and file formats used for storing, processing, and displaying 3D models. Finally, Part IV discusses the process of integrating indoor and outdoor coordinates — a necessary step for ubiquitous global positioning and navigation. This final part

begins with a review of spatial referencing followed by three special topics for indoors: a proposed method for establishing indoor coordinates, a method for integrating indoor and outdoor coordinates, and an analysis of the limits of the 3D Cartesian coordinate system for geographic applications.

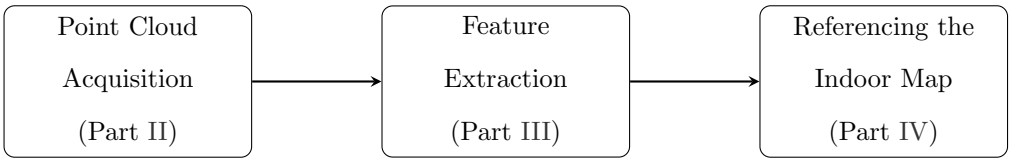


Figure 1.7: Organization of dissertation

Part II

Point cloud acquisition

Chapter 2

Point cloud acquisition

The first step in digitally reconstructing an indoor space involves taking its measurements (Héno and Chandelier 2014). In the not-too-distant past, this involved taking only a handful of measurements of a room — say, of its corners or between key reference points — which were then used to deduce its shape and form. In more recent times, powerful computers and advanced sensors have automated the measurement process, making it possible to capture, store, and process massive amounts of measured coordinate data in a short period of time. These measured points collectively form a *point cloud* — i.e., a collection of three-dimensional xyz coordinates corresponding to locations on solid surfaces in a room.

Many techniques exist for performing three dimensional remote sensing[1], nearly all of which use some form of energy. Figure 2.1 divides 3D remote sensing into methods that use either mechanical sound waves (e.g., sonar), the properties of light, or non-visible electromagnetic energy (e.g., radar). Most remote sensing systems for building modeling use light-based approaches, which can further be divided into light detection

[1]The term *reality capture* is sometimes used to refer to 3D remote sensing, especially for a non-technical audience.

and ranging (LiDAR) or image-based photogrammetric methods. Luhmann et al. (2011), Mikhail (2001), McGlone (2013), and Renslow (2012) provide detailed technical discussions on these approaches. This dissertation focuses exclusively on the use of LiDAR and photogrammetry using structure-from-motion/multi-view stereo (SfM-MVS).

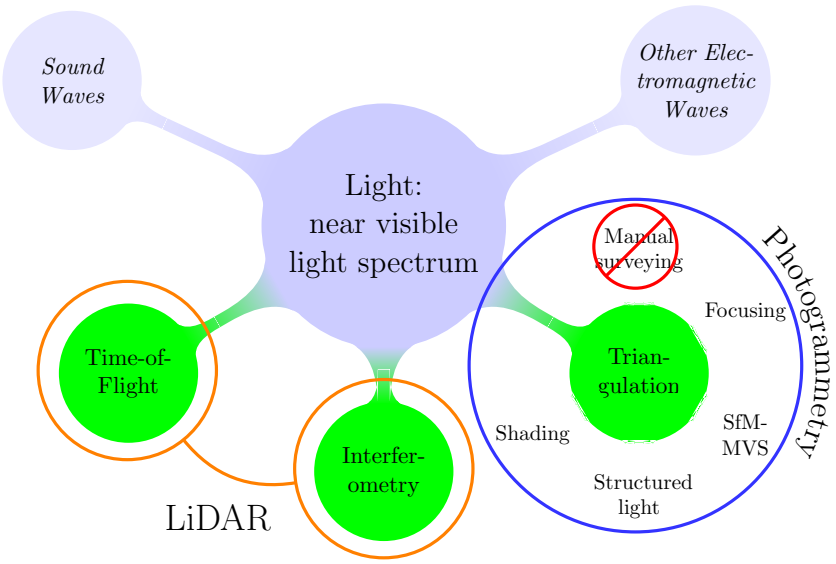


Figure 2.1: Approaches to 3D remote sensing

2.1 Light detection and ranging (LiDAR)

LiDAR uses the properties of light to determine the distance between the measurement instrument and the target. The *light* portion of LiDAR can use any type of light source, but the widespread use of lasers has made LiDAR synonymous with the term *laser scanning*.[2] The *ranging* portion of LiDAR uses techniques similar to those of radar (radio detection and ranging) and involves either directly or indirectly determining the

[2]However, some low-cost LiDAR sensors can also use non-laser light sources for very close range applications.

time-of-flight (ToF) of light as it travels from the sensor to the target and back. Other less used terms for LiDAR include lidar (all lower case), LADAR (laser detection and ranging), and OpDAR (optical RADAR).

2.1.1 Brief history of LiDAR

Synge (1930) published the earliest known theoretical work on LiDAR that proposed using one hundred search lights to measure the heights of overhead clouds. Tuve, Johnson, and Wulf (1935) modified Synge's idea by substituting the one hundred search lights with a pulsed light source, but their proposal was also purely theoretical. The first successful demonstration of LiDAR is credited to Bureau (1946) who successfully used pulsed lights to measure the heights of clouds in 1938. However, it was only after the development of pulsed lasers in the 1960s that LiDAR became a practical method of measurement, with initial deployments on aircraft for airborne measurements of the terrain below followed by terrestrial applications as the instruments shrank in size.

2.1.2 Calculating distances with LiDAR

LiDAR sensors measure distance using the time it takes for light to travel from the sensor to the target and back, with the method of determining the ToF serving as the main differentiator between the various types of systems. The two approaches most commonly used in remote sensing involve directly timing the ToF of pulsed lights or indirectly deducing the ToF of frequency-modulated light, with the former often referred to as *pulsed* LiDAR and the latter as *phase-based* LiDAR. Measurement results are often reported as a collection of Cartesian coordinates or spherical coordinates using fixed units of measurement that reflect each light signal's ranging distance and orientation (Shan

and Toth 2009; El-Omari and Moselhi 2008). Since LiDAR uses emitted light, they can function in both lit and unlit environments.

Pulsed LiDAR

Pulsed LiDAR systems determine distance by measuring the travel time of a short pulse of light emitted from the scanner. Since the light beam will often diverge (widen) as it travels to the target, a single pulse will often detect multiple objects or different parts of the same object at different depths, with the return signal showing a continuous wave form with peaks and valleys as shown in Figure 2.2. For instance, a single LiDAR pulse traveling through vegetation will often capture the canopy (1st return in the example), various parts of the intermediate foliage (second return in the example), and the ground (third return in the example) as peaks within the waveform.

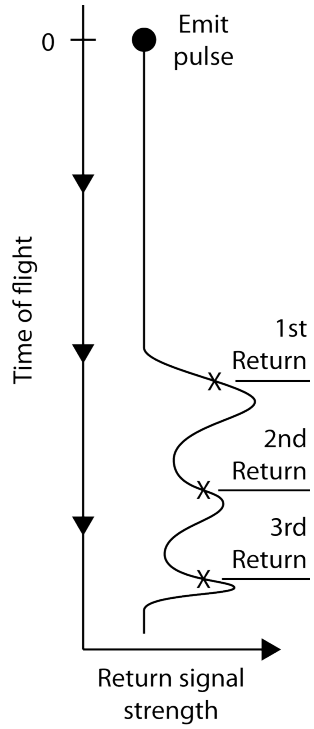


Figure 2.2: Example of a return signal from an airborne LiDAR scanner pointed down towards the ground

The conventional approach to processing LiDAR signals involves using a discretization algorithm that treats each peak of the wave form as a single *return*, as indicated by the Xs in the example. As a result, different objects may show up as multiple returns with the "first return" representing the object nearest the scanner. Techniques that use this discretization method are often called *discrete LiDAR*, since the resulting measurements are presented as discrete and unambiguous. Other techniques use the the entire wave form and are referred to as *full waveform LiDAR*, which are more difficult to work with and reflect actual ambiguities in the real signal. In both instances, the method for determining distance involves multiplying the speed of light (c) by the light pulse's travel time (t) and dividing the product in half to account for the round-trip route (Equation 2.1); in practice, this equation is further adjusted for refraction and other factors.

$$\text{Distance} \approx \frac{c \times t}{2} \tag{2.1}$$

Phase-based

Phase-based approaches calculate distance based on a shift in phase of a light signal's sinusoidal wave form — detected using a technique called interferometry — as it hits the target and returns to the measuring device (Zheng 2005; Duh 2012; Baghmisheh 2017). Two common implementations of phase-based approaches are the chirped pulse method and the frequency-modulated continuous wave (FMCW) method. In the chirped pulse method, the scanner sends a pulse of light with a fixed frequency (f) and finds the ToF using the phase shift angle (ϕ) of the return signal. Distance is then calculated by multiplying the derived ToF by the speed of light and dividing by two. Equations 2.2 and 2.3 show the derivation of ToF from ϕ, and Equation 2.4 shows the general equation for distance.

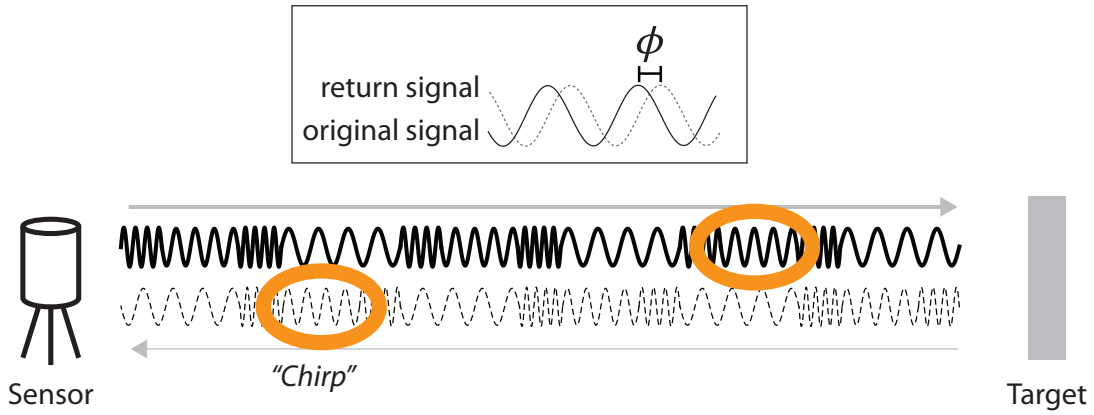


Figure 2.3: Chirped pulse method

$$\text{rev} = \frac{\phi}{2\pi} \text{ (fraction of wavelength)}; \;\; T = \frac{1}{f} \text{ (period of wavelength)} \tag{2.2}$$

$$\text{ToF} \approx \text{rev} \times T = \frac{\phi}{2\pi f} \tag{2.3}$$

$$\text{Distance} \approx \frac{c}{2}\frac{\phi}{2\pi f} \tag{2.4}$$

In the FMCW method, the scanner sends the outgoing light with a time-based frequency signature and finds the ToF by comparing it with the return signature as illustrated in Figure 2.4. Distance is calculated by multiplying the ToF by the speed of light and dividing by two in the same manner as with the other methods.

Comparison

Pulsed LiDAR and phase-based LiDAR each have unique capabilities and limitations. Pulsed systems have a greater detection distance than phase-based systems, since they can concentrate more energy into producing short bursts of light as opposed to phase-based systems that must distribute that same energy into powering a continuous light

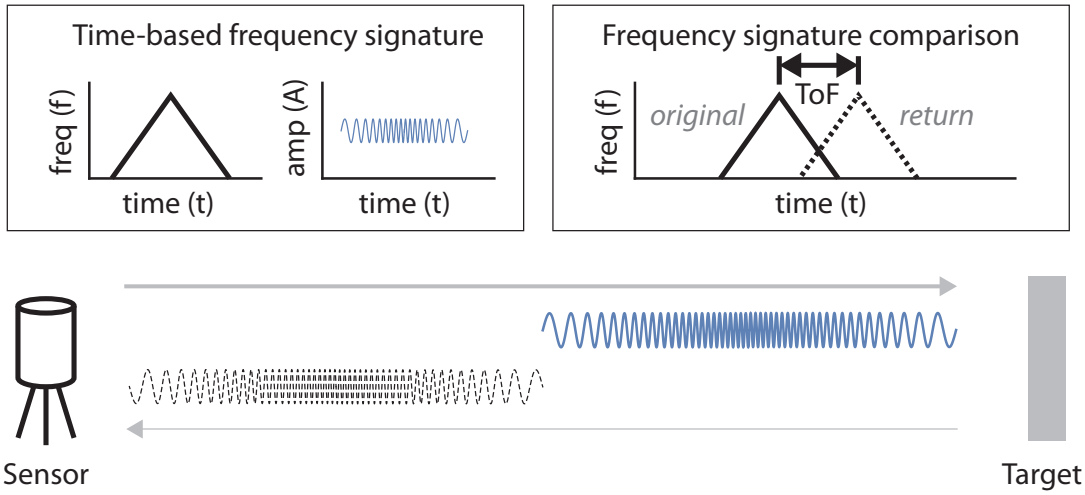


Figure 2.4: Frequency-modulated continuous wave method

stream. Pulsed systems usually have a maximum range of over one kilometer[3] while many phase-based systems top out at about 200 meters at the time of writing this dissertation. For close range work that includes indoor scanning, however, phase-based systems generally provide superior precision and point cloud density compared to pulsed LiDAR. The precision advantage comes from the less ambiguous nature of phase-based timing and the density advantage comes from the ability to send multiple signals in a constant stream of light, whereas pulsed systems require a minimum separation time between pulses to prevent ambiguous signals. Additionally, phase-based instruments often cost less and are more portable than pulsed LiDAR instruments.

2.2 Digital photogrammetry

Recent growth in the 3D remote sensing marketplace has resulted in a multitude of smaller, cheaper, and faster ways to capture 3D point cloud measurements. One such

[3]One of the earliest experiments with laser-based LiDAR involved measuring the distance to the Moon (Smullin and Fiocco 1962).

family of techniques is photogrammetry, which literally means "taking measurements from something drawn or written with light" from the Greek root words *fotos* (genetive form of light), *grámma* (something drawn or written), and *métron* (measurements) (McGlone 2013). At its core, photogrammetry reverses the photographic process[4] by reconstructing three-dimensional scenes using information from two-dimensional images. Luhmann et al. (2014), McGlone (2013), and Mikhail (2001) provide detailed discussions on how photogrammetry works, and the term *digital* photogrammetry describes those techniques that can take advantage of electronically generated data. Of the various photogrammetric techniques, the two most commonly used for building measurements are structure-from-motion/multi-view stereo and structured light.

2.2.1 SfM-MVS

Structure-from-motion/multi-view stereo (SfM-MVS) has emerged as perhaps the single most popular approach to photogrammetry due to its low cost, versatility, and relative ease-of-use (Kersten and Lindstaedt 2012; Smith, Carrivick, and Quincey 2016; Verhoeven 2011; Westoby et al. 2012). It has a very low entry barrier in terms of cost and user skills and can be found in a wide range of software implementations ranging from free open source packages to "freemium" programs to proprietary professional software programs, all of which only require digital photographs taken with any type of digital camera for creating 3D point clouds.

Calculating distances with SfM-MVS

SfM-MVS uses a series of overlapping 2D digital photographs to recover 3D information via the principle of parallax, i.e. the phenomenon of how photographed objects

[4]Photography converts 3D information into a 2D form.

appear to laterally shift at different rates when viewed from different laterally-shifted vantage points (Smith, Carrivick, and Quincey 2016). However, measurements based on conventional 2D photographs have no connection to real world distances making it necessary to scale SfM-MVS models using supplemental information from other sources such as georeferenced camera locations or measured distances between fixed ground control points (GCPs) (Falkingham, Bates, and Farlow 2014).

The SfM-MVS workflow first uses SfM to estimate camera locations and produces a sparse (i.e., low density) 3D point cloud of shared key points detected in the input photographs. This information is then used by MVS to fine tune the camera parameters and pair images together to produce the final dense 3D point cloud, which can be scaled to the correct unit of measurement using supplemental information.

Software

Nearly all SfM-MVS software hide the complex SfM-MVS algorithms from end users, resulting in automated or semi-automated processes in which users provide sets of overlapping photos and the software returns 3D point clouds. The recent proliferation of affordable aerial drones has resulted in a growth in the number of SfM-MVS software providers, which has driven down software prices to the point where some vendors provide SfM-MVS services for free or on a freemium basis[5]. Popular SfM-MVS software or services include Agisoft Photoscan, Autodesk ReCap, Bentley ContextCapture, Bundler, DroneDeploy, Photomodeler, and Pix4D. Some of these programs semi-automate the SfM-MVS process by allowing users to simply upload photos and process them through

[5]The term freemium describes a business model that provides free basic services but charges for premium services (Kumar 2014).

mediated workflows, while others provide full off-site services through their online programs.

Limitations

Image acquisition represents one of the main limitations of SfM-MVS. As a passive remote sensing technique, photographing a scene requires an external lighting source to illuminate the environment. This means that SfM-MVS will struggle in places with poor lighting, e.g., dark environments or areas with excessively high light-to-dark contrast. SfM-MVS also requires a series of photos to capture parallax — a problem in confined spaces or poor access to overhead views of the ground. Additionally, SfM-MVS requires texture to detect image movement and will often fail with textureless or uniformly textured surfaces, a common challenge in indoor spaces. Image resolution, camera lens distortion, and depth-of-field also impact SfM-MVS quality. When using low-cost cameras or video recorders, a rolling shutter system (versus a full frame or mechanical shutter) can distort images taken from moving platforms, resulting in scan-line distortions or a "jelloing effect" that degrades point cloud quality.

SfM-MVS generated point clouds also suffer two other drawbacks. First, these point clouds are inherently unitless making it necessary to use control points and reference scales for proper scaling. The location of all other points are interpolated, which can introduce a level of uncertainty to the data. Second, since SfM-MVS deduces measurements based on parallax, with insufficient parallax contributing to a distorted point cloud unless caught and fixed with control points. Finally, producing high quality point clouds of indoor spaces using SfM-MVS often comes with a significant burden of labor, time, and computing resources.

2.2.2 Structured light

Structured light uses calibrated patterns of projected light to calculate distances using triangulation. Among the different types of structured light systems (see Luhmann et al. (2014)), one popular approach pairs an infrared (IR) projector with two cameras — one IR and one color — to produce a colorized 3D point cloud (Khoshelham and Elberink 2012). The IR camera captures the projected IR pattern and calculates distance for each pixel by comparing it to a calibration pattern stored in memory, and the resulting point cloud is then colorized using the image from the RGB camera. This technique, called RGB-D (D for depth), was initially implemented in inexpensive consumer grade cameras designed for computer games, namely, the Microsoft Xbox 360 Kinect, PrimeSense Sensor, and Asus Xtion. A handful of companies soon converted these cameras into indoor mapping platforms; one such company was Matterport, Inc., which placed three RGB-D sensors (one facing forward and one each facing up and down) on a mechanically rotating base.

RGB-D has the advantages of being low cost, easy-to-use, and able to operate in dark environments due to use of emitted light. For example, the Kinect compared favorably with LiDAR at up to 3 m (Khoshelham and Elberink 2012). However, it has several disadvantages. First, RGB-D has a range of about 5 m, making it suitable only for small spaces when used statically. Second, bright lights such as sunlight can obscure the relatively weak IR signal and introduce noise to the data or even cause a break in measurements. Third, unmodified consumer grade scanners can experience drift as they rotate or move around when scanning larger spaces, introducing additional errors in the data and creating improperly aligned point clouds.

2.3 Other methods

Any type of automated or semi-automated distance measuring technique can conceivably be used to produce point cloud measurements of indoor spaces, including the use of sonar. While sonar lacks the spatial resolution of optical methods, it can provide useful information for surfaces that optical methods traditionally struggle with, such as windows, mirrors, and monochromatic reflective surfaces. Sonar instruments can take the form of simple and inexpensive single beam formats or more complex forms such as the system developed by Swiss researchers that can map an entire room with sound from a single snap of the fingers (Dokmanić et al. 2013).

2.4 Conclusion

Among the three methods covered in this chapter, only LiDAR and structured light have practical value for measuring indoor spaces due to their use of an internal light source. SfM-MVS is generally impractical for indoor use due to the need for scene preparation, external lighting, and its long processing times. However, the ability of SfM-MVS to capture very fine details may make it suitable for modeling intricate architectural features. While structured light provides an affordable way to scan small indoor spaces under most lighting conditions, it also suffers from a few practical limitations that include weak light signals subject to interference and alignment drift. LiDAR stands out as the most promising technique due to its consistency, accuracy, precision, and speed.

Until recently, the price of 3D LiDAR scanners exceeded $30,000 USD for a basic phase-based unit, making it prohibitively expensive for most users. Even a handheld LiDAR, the GeoSlam ZEB1, based on the Zebedee prototype developed in Australia, had a retail price of over $15,000 in 2015. However, rapid growth in consumer robotics

and autonomous vehicles have led to the development of low-cost miniaturized LiDAR sensors. In 2014, PulsedLight released what was perhaps the first consumer grade phase-based LiDAR sensor, the LIDAR-Lite, at a price of only $90 USD and with a range of 40 m (Higgins 2015; PulsedLight 2014). Less than three years later, Scanse used a newer version of the LIDAR-Lite — since acquired by Garmin, Inc. — to create the first low cost 3D LiDAR scanner, the Scanse Sweep 3D, that provided panoramic scanning at up to 40 m distances; it had a price of $400 for a self-assembly kit or $1,000 fully assembled (Higgins 2016, 2017). Developments such as the LIDAR-Lite and Scanse Sense 3D point to a future where the generation of high quality point clouds of indoor spaces could become more commonplace, perhaps even contributing to a future of crowdsourcing.

Chapter 3

Using SfM-MVS with Google Earth

Material from this chapter was published in the AutoCarto 2016 conference proceedings (Chen and Clarke 2016).

3.1 Introduction

3.1.1 An important limitation of SfM-MVS

Digital photogrammetry using structure-from-motion/multi-view stereo (MVS) techniques provides cartographers with an economical and accessible way to measure and model the physical environment using a simple digital camera, a handful of physical baseline measurements, and SfM-MVS software. It can serve as an alternative to laser scanning, which has a higher entry cost, or as a complementary technique for densifying close-range laser scans that have superior accuracy. However, effective use of SfM-MVS requires photographs to be taken from different vantage points, which can present problems when attempting to get images from overhead. For non-critical applications, the 3D models in Google Earth can provide a no-cost alternative to real-world image gathering.

This study investigated the feasibility of using Google Earth's 3D models with SfM-MVS software.

3.1.2 Google Earth 3D models

One workaround to the overhead imagery collection problem involves the use of Google Earth. Google Earth, a 3D virtual globe (Goodchild et al. 2012), includes photorealistic 3D models of Earth that Google has created using a variety of reality capture techniques. These 3D models include coarse resolution 2.5D topography for the entirety of Earth and high resolution 3D models for select urban areas. Google's early attempts at urban modeling involved crowdsourcing with its former SketchUp software. However, crowdsourcing produced inconsistent results prompting Google to sell SketchUp to Trimble Navigation, Ltd., in 2012, and move its urban modeling and reality capture program in-house (McClendon 2012).

Google Earth permits viewing of its 3D models from multiple perspectives, which allows for the simulation of terrestrial, aerial, and space-based image capture. However, the convenience of this approach comes at a cost. First, Google's proprietary 3D models have no measures of reliability; as a result, derived models will inherit all the inherent uncertainties in the original Google Earth models. Second, high resolution models only exist for areas that Google has modeled using high resolution imagery, laser scans, etc. This means metropolitan areas such as Tokyo or New York would qualify but areas with limited high resolution data such as Pyongyang would not.

3.2 Methodology

This study examined three test areas in Google Earth: two high resolution urban areas and one low resolution rural area. Laser scanning data supplemented one urban area and

the rural area for comparison purposes. The goal of this study was to test the feasibility of using SfM-MVS to model different types of environments using Google Earth's 3D data and to quantify their discrepancies relative to accurate LiDAR measurements available for UCSB and Mount Herard.

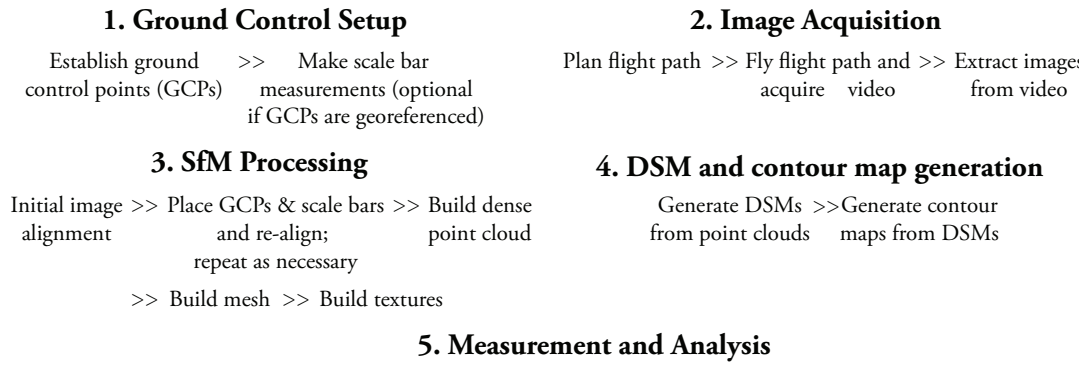
The three study areas consisted of a dense urban environment, a mixed environment, and a natural environment that had Google's proprietary high resolution 3D models: a residential neighborhood in the Shinjuku ward in Tokyo, Japan (UTM Zone 54S 384900mE, 3949700mN); portions of the University of California, Santa Barbara (UCSB) campus (UTM Zone 11S 239000mE, 3811900mN); and Mount Herard in Colorado (UTM Zone 13S 456500mE, 4189190mN)[1].

The SfM-MVS workflow consisted of five phases: ground control setup, image acquisition, SfM-MVS processing, digital surface model (DSM) generation, and measurement and analysis as shown in Figure 3.1. The SfM-MVS processing was performed using Agisoft Photoscan Professional while the DSM was created using ArcGIS.

3.2.1 Ground control setup

Ground control setup involved selecting ground control points (GCPs) from Google Earth to simulate the acquisition of GCPs in the real world. These GCPs used UTM (WGS84) for planimetric/horizontal measurements and presumably EGM96(WGS84) for vertical measurements (although Google does not clearly state the vertical coordinate reference system). Horizontal measurements had centimeter precision and vertical measurements had a precision of one meter. In addition to point coordinates, pairs of GCPs were also used to create scaling distances for the SfM-MVS software.

[1]Note that the UTM designator uses zone bands. See Section 9.2.4.



5. Measurement and Analysis

	Horizontal distances between point-pairs	Vertical distances between point-pairs	Building dimensions	DSM differencing with LiDAR (not Google)	Contour maps with LiDAR (not Google)	Visual representation
Tokyo	X	X	X			X
UCSB	X	X	X	X		X
Mount Herard		X		X	X	X

Figure 3.1: Workflow overview

3.2.2 Image acquisition

Image acquisition involved recording video images of the ground during virtual fly-overs of the study areas, which involved a back-and-forth sweeping configuration to ensure image overlap (Figure 3.2). While flights at multiple altitudes and cross-paths would have been ideal, time constraints limited flights to flight lines oriented in a single direction at one altitude per study area: 300 m above ground level for Tokyo and UCSB and 1,000 m for Mount Herard. To make the most of these single-direction, single-altitude flight lines, the camera was set to a 45° angle from nadir to provide simultaneous coverage of horizontal and vertical surfaces to maximize parallax in both directions.

Microsoft Expression 4 Screen Capture software was used to capture video images of the simulated aerial photos at a resolution of WQHD (2560x1440 pixel), which delivered an effective 3.2 megapixel (MP) resolution (2464x1312 pixel) after accounting for the removal of borders and non-image elements. Even though Agisoft recommended using images with a minimum resolution of 5 MP (Agisoft LLC 2016), the study from Chapter

Figure 3.2: Flight lines

4 showed that a minimum image resolution of 2 MP was sufficient for PhotoScan to produce useable point clouds. The open source VLC media player was used to extract individual still images from the videos for use in SfM-MVS.

3.2.3 Point cloud generation

The modeling workflow involved five steps in PhotoScan as shown in Figure 3.3. First, images were aligned using structure-from-motion techniques. This process involved detecting key points in each image using computer vision algorithms and estimating camera lens parameters to correct image distortions. If successfully performed, the initial image alignment placed some or all images into their correct locations and generated a sparse point cloud of tie points. If this process failed, tie points were manually added to the images to assist with the registration process.

The second step involved placing GCPs and scale bars in a subset of images and further refining the alignment of all images. The software automatically estimated the location of GCPs in all other images from a handful of manually placed GCPs. (Scale bars were only used for Tokyo and Mount Herard.) Upon reaching a satisfactory state, the camera parameters were calibrated one final time to optimize their parameters based on the refined image locations.

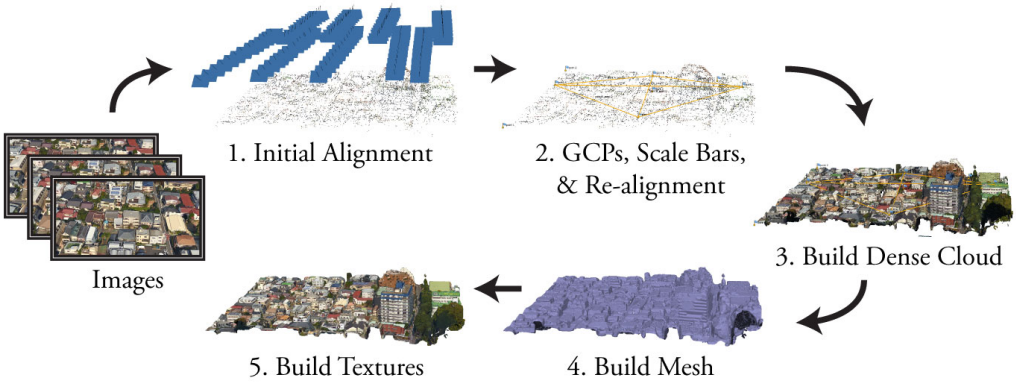


Figure 3.3: PhotoScan workflow

With all images correctly placed and all GCPs and scale bars established, the third step used multi-view stereo techniques to generate a dense point cloud. At this point, the resulting point cloud contained millions of points that collectively provided a photorealistic 3D rendering of the modeled environment when converted to voxels — something that point cloud viewers such as CloudCompare and MeshLab automatically perform. At this point, the point cloud could be used for visualization or further processed into a mesh model.

Generating a textured mesh of the point cloud addresses the pixelation issue and provides a photo-realistic 3D model of the mapped environment. The last two steps of the workflow used PhotoScan to generate a meshed surface and overlay it with image-based texture for photorealism.

3.2.4 Digital surface modeling and contour map generation

For UCSB and Mount Herard, comparing the SfM-MVS point cloud with LiDAR involved transforming the point measurements into a rasterized digital surface model (DSM) and calculating the difference in elevation values between overlapping cells. The Esri ArcGIS "LAS Dataset to Raster" tool was used for the DSM generation, with 2 m

cells for UCSB and 30 m cells for Mount Herard. The DSM for UCSB used the maximum elevation value per cell due to points existing along vertical features, e.g., walls; otherwise, the edges of buildings would be located midway between ground and roof. The DSM for Mount Herard used average elevation values.

Google Earth used UTM (WGS84) and EGM96 (WGS84) as the horizontal and vertical coordinate reference systems (CRSs) but LiDAR data for Mount Herard and UCSB used different CRSs[2]. With its much smaller geographic area and discernible key features, the UCSB LiDAR data was manually transformed using key point registration in CloudCompare. Without any salient features for manual registration, the Mount Herard LiDAR data was transformed to UTM/EGM96 (WGS84) using the U.S. National Oceanic and Atmospheric Administration (NOAA) VDatum software. Contour maps were also generated for Mount Herard using ArcGIS to provide a qualitative assessment of horizontal accuracy.

3.2.5 Measurements and analysis

Ideally, analysis would involve quantifying the elevation differences between the SfM-MVS point cloud and Google Earth for all locations throughout the study areas. Unfortunately, Google did not make its proprietary data publicly available, which made it necessary to use alternative approaches. These approaches involved comparing horizontal and vertical distances between point pairs and comparing digital surface models.

[2]Mount Herard LiDAR used UTM (NAD83)/NAVD88 and UCSB LiDAR used SPCS CA Zone 5 (NAD83)/NAVD88.

Horizontal and vertical distances

Horizontal distances were measured between pairs of GCPs and non-GCP random points, with the latter accounting for the anchoring effect of GCPs in the point cloud generation process. These measurements were separated into horizontal (planimetric) and vertical components, instead of straight-line Euclidean distances, to prevent large horizontal distance values from masking significant deviations in the vertical components. Vertical distances consisted of the elevation differences between two points while horizontal distances consisted of the Euclidean distance as shown in Equation 3.1.

$$D_h = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.1}$$

Point clouds generated by SfM-MVS for Tokyo and UCSB contained enough detail to place GCPs and random points on salient features, e.g., building corners, for assessing horizontal deviations from Google Earth. Additionally, building measurements were also performed for Tokyo and UCSB to assess small area deviations. However, the natural features of Mount Herard made it impossible to identify sufficient numbers of unambiguous points; instead, a qualitative assessment of horizontal deviation was performed using contour maps. Differences in elevation values were calculated in ArcGIS by finding the elevation value of the nearest neighbor in the point clouds for each individual point, while elevation values from Google Earth were manually determined by reading the on-screen value associated with each individual control point.

Digital surface models

DSM analysis involved applying map algebra to determine elevation differences on a cell-by-cell basis using Equation 3.2. The UCSB assessment used a qualitative approach due to the abundance of elements that had changed between the time of LiDAR acquisi-

Table 3.1: Horizontal distances between pairs of points

	Tokyo					UCSB				
Pair #	Type G/R	PhotoScan meters	Google meters	Difference meters	%	Type G/R	PhotoScan meters	Google meters	Difference meters	%
1	G	112.45	113.44	-1.00	-0.9%	G	167.86	166.33	1.53	0.9%
2	G	207.47	208.53	-1.07	-0.5%	G	72.07	72.13	-0.06	-0.1%
3	G	143.67	145.02	-1.35	-0.9%	G	122.59	122.01	0.57	0.5%
4	G	107.98	108.06	-0.08	-0.1%	R	241.02	241.97	-0.95	-0.4%
5	G	120.41	121.42	-1.00	-0.8%	R	218.24	219.04	-0.79	-0.4%
6	G	122.15	122.80	-0.65	-0.5%	R	276.11	274.23	1.88	0.7
7	R	46.03	45.68	0.35	0.8%					
8	R	89.17	89.23	-0.06	-0.1%					
9	R	60.08	59.96	0.12	0.2%					

tion (c.2007) and Google Earth's data (c.2015), such as vegetation, marshlands, and the beach. Mount Herard data was assessed using a quantitative approach involving descriptive statistics, a histogram, a cumulative distribution function (CDF), and visualization of the geographic distribution of errors.

$$DSM_{Diff} = DSM_{SfM.MVS} - DSM_{LiDAR} \tag{3.2}$$

3.3 Results

3.3.1 Horizontal measurements

Horizontal measurements for the Tokyo and UCSB data sets showed differences of less than one percent between Google Earth and the SfM-MVS point clouds (Tables 3.1 and 3.2). In the column "Type, G/R," G identifies distances between pairs of GCPs while R identifies point-pair distances between random points.

Table 3.2: Measurements of building dimensions

		Tokyo				**UCSB**			
		PhotoScan	Google	Difference		PhotoScan	Google	Difference	
Bldg	Dim	meters	meters	meters	%	meters	meters	meters	%
1	L	24.3	24.5	-0.2	-0.8%	105.6	106.2	-0.6	-0.6%
	W	26.1	26.4	-0.3	-1.1%	92.4	93.2	-0.8	-0.9%
	H	21.0	20.1	0.9	4.5%	13.3	13.4	-0.1	-0.7%
2	L	25.8	25.7	0.1	0.5%	46.0	46.1	-0.1	-0.2%
	W	27.6	27.8	-0.2	-0.8%	66.2	66.7	-0.5	-0.7%
	H	5.3	5.1	0.2	4.3%	12.7	12.6	0.1	0.8%
3	L	7.9	7.9	0.0	0.0%	49.9	50.2	-0.3	-0.6%
	W	12.3	12.3	0.0	0.0%	55.5	55.7	-0.2	-0.4%
	H	8.3	8.1	0.2	2.5%	11.1	11.2	-0.1	-0.9%

GCPs could not be precisely located in the Mount Herard data, so visual analysis using isolines — - i.e., contour maps — - was used to compare the SfM-MVS data to LiDAR (Figure 3.4). Even though deviations at specific locations varied widely, analysis of the overall map showed a general conformance of horizontal positions to actual conditions based on LiDAR, though not necessarily to Google Earth itself.

3.3.2 Vertical measurements

The SfM-MVS process produced mixed results in vertical measurements for the sampled point locations. Vertical measurements in the built-up areas of Tokyo and UCSB showed little to no deviation from Google Earth within Google Earth's fine sub-centimeter resolution for vertical point-to-point distances and coarse 1 meter resolution for elevations (Tables 3.2 and 3.3). However, the all-natural area of Mount Herard exhibited significant vertical deviations with values approaching or exceeding 100 m, representing over 10% of the overall height range for the entire modeled area.

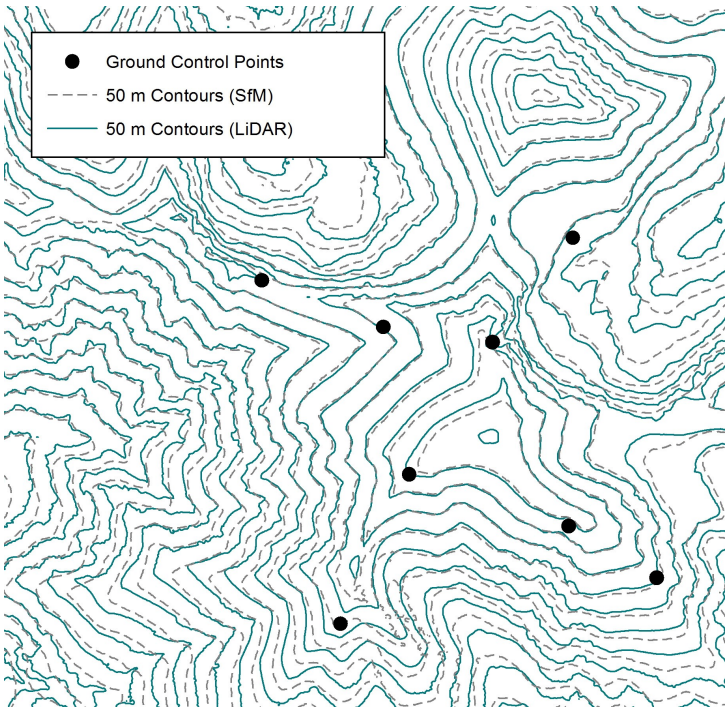


Figure 3.4: Contour map of Mount Herard comparing SfM-MVS to LiDAR

Table 3.3: Elevation differences between pairs of points

	Tokyo				**UCSB**				**Mount Herard**			
Pair #	Type G/R	PhotoScan meters	Google meters	Diff meters	Type G/R	PhotoScan meters	Google meters	Diff meters	Type G/R	PhotoScan meters	Google meters	Diff meters
1	G	-2	-2	0	G	0	0	0	G	119	22	97
2	G	-4	-4	0	G	0	-1	1	G	-72	-153	81
3	G	-1	-1	0	G	-1	-1	0	G	-33	-53	20
4	G	-2	-2	0	R	1	2	-1	G	-151	-75	-76
5	G	1	1	0	R	1	1	0	G	-63	-32	-31
6	G	-3	-3	0	R	0	-1	1	G	-89	-175	86
7	R	0	0	0					G	-36	22	-58
8	R	0	1	-1					G	27	54	-27
9	R	0	0	0					R	38	149	-111
10									R	64	96	-32
11									R	25	-53	78

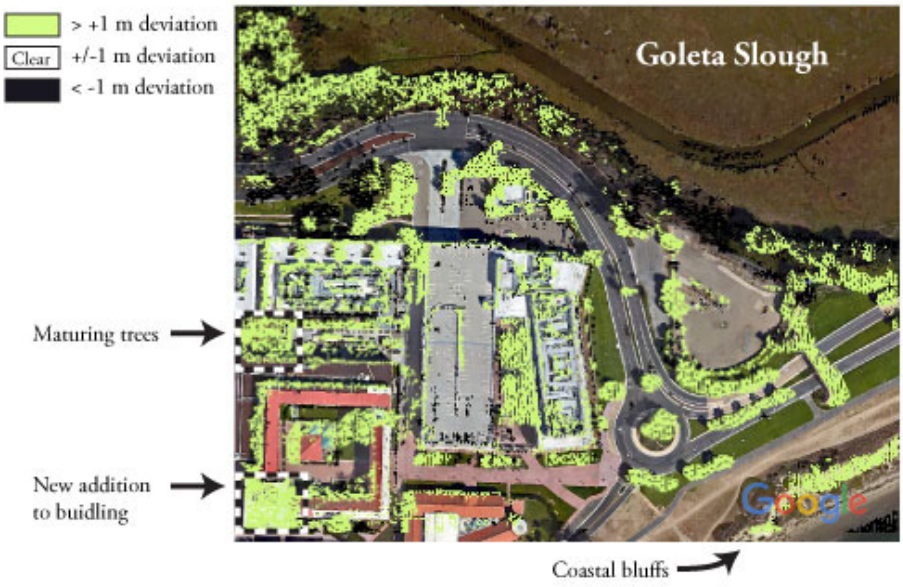


Figure 3.5: Visualized elevation differences between SfM-MVS and LiDAR at UCSB using 2 m DSM cells

3.3.3 DSM analysis

An extended time difference between the LiDAR data and Google Maps prevented use of the UCSB DSM data for descriptive statistics. Nonetheless, a visual inspection of the DSMs provided some insights into how well the two data sets matched. Figure 3.5 shows the time-varying changes which would have significantly skewed results of numerical analysis — most of these changes appeared to be due to vegetative growth, erosion, or land subsidence. Additionally, a new building addition can be seen in the bottom left of the figure. Removing these time-varying elements provides a more promising picture showing that the SfM-MVS point cloud conformed to actual site conditions by +/- 1 m, within the range of the vertical resolution of Google Earth.

The large geographic area of the Mount Herard study area allowed use of a comprehensive cell-by-cell comparison of the data. Figure 3.6 shows a histogram of differences in elevation values between SFM-MVS and LiDAR. This graph shows that on average

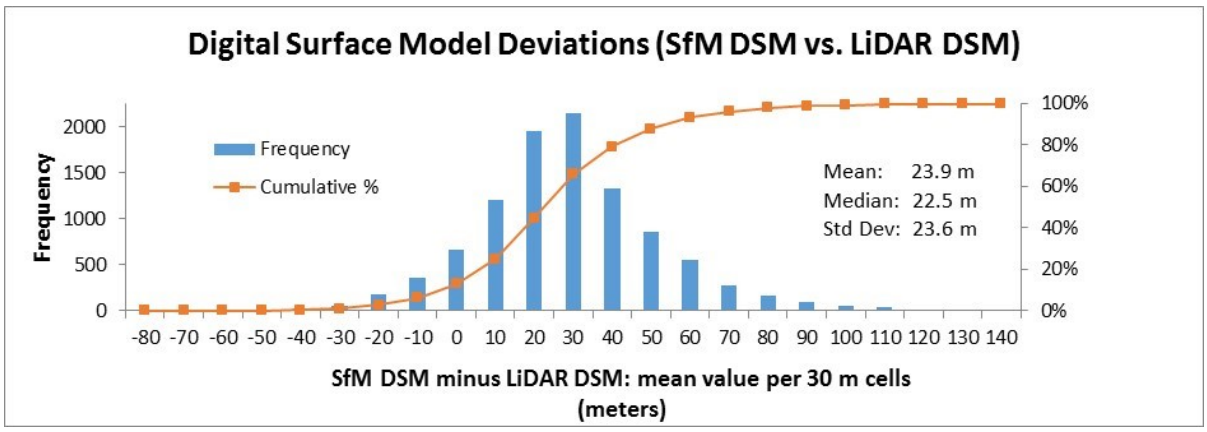


Figure 3.6: Elevation differences between SfM-MVS and LiDAR DSMs at Mount Herard using 30 m DSM cells

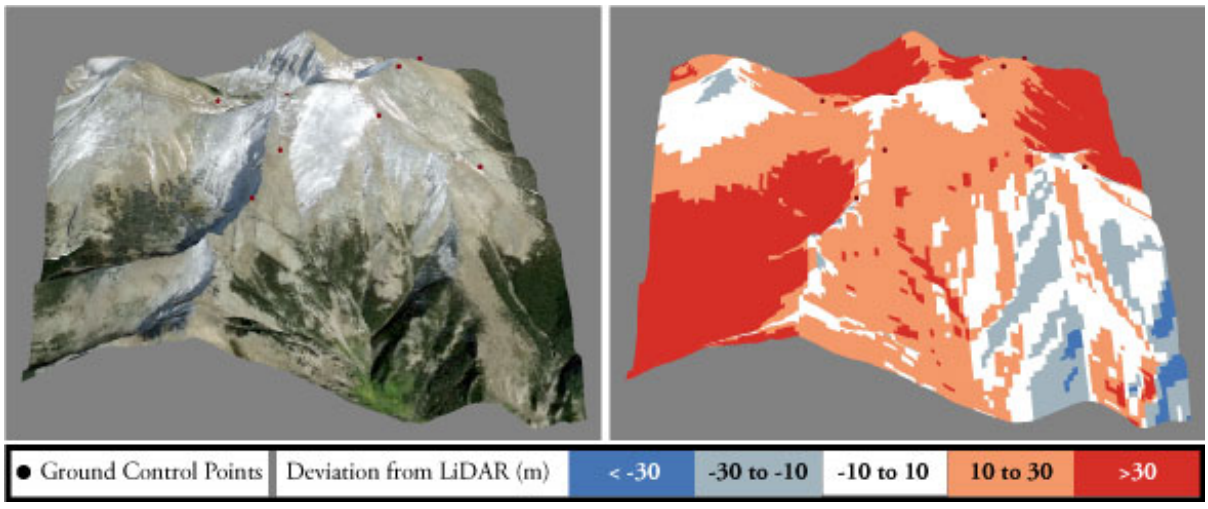


Figure 3.7: Visualization of DSM differences between SfM-MVS and LiDAR at Mount Herard

Table 3.4: Horizontal distances between pairs of points

	Mount Herard				
Pair	Type	Google	LiDAR	Difference	
#	G/R	meters	meters	meters	%
1	G	22	28	-6	-22%
2	G	-153	-140	-13	9%
3	G	-53	-46	-7	16%
4	G	-75	-74	-1	1%
5	G	-32	-7	-25	372%
6	G	-175	-172	-3	2%
7	G	22	23	-1	-6%
8	G	54	30	24	79%
9	R	149	149	0	0%
10	R	96	122	-26	-21%
11	R	-53	-27	-26	95%

SfM-MVS produced elevations values exceeding those in the LiDAR data by a mean value of 23.9 m, although with a standard deviation of 23.6 m. Figure 3.7 shows the geographic distribution of these elevation differences; areas with greater texture or more topographic relief had the least amount of error while low textured or topographically homogeneous areas had the greatest amount of error, as expected when using SfM-MVS.

3.3.4 Google versus LiDAR

Availability of LiDAR data for Mount Herard made it possible to examine errors in Google Earth's model compared to actual physical measurements. Table 3.4 shows vertical distance errors between point-pairs in Google Earth and the LiDAR data for Mount Herard using the same point-pairs shown in Table 3.3.

3.3.5 Visual representation

Applying meshing and texture mapping to the point clouds produced models that appeared nearly identical to the Google Earth models (Figures 3.8 to 3.10). In each case, PhotoScan accurately captured very fine resolution elements including the shape of small features such as trees and cars as well as model imperfections. Achieving this level of fidelity required numerous hours of manual adjustments to remove artifacts from misaligned images; it did not occur automatically using the automated process.

Figure 3.8: Visual comparison of Google Earth and PhotoScan models for Tokyo

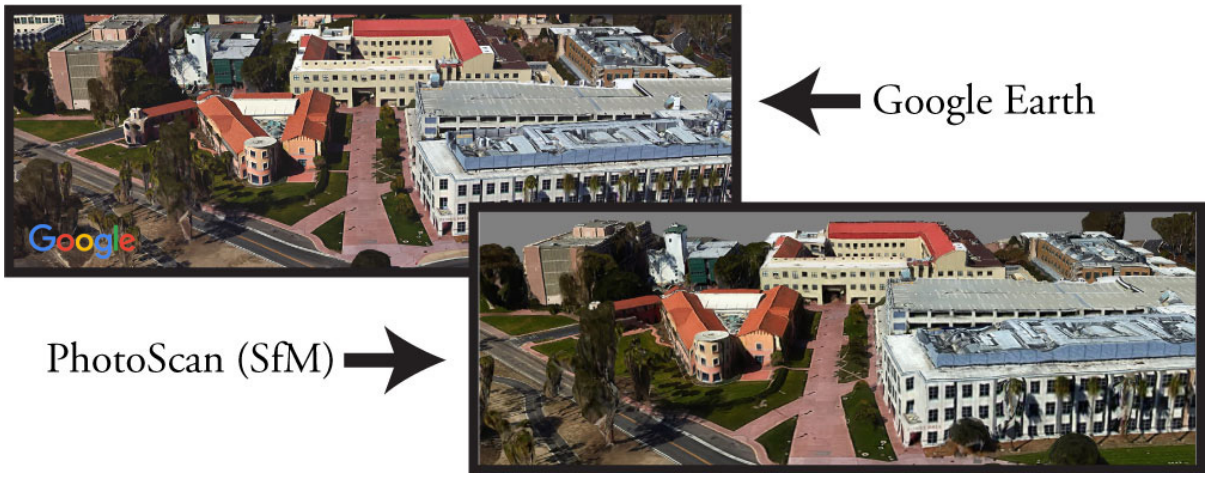


Figure 3.9: Visual comparison of Google Earth and PhotoScan models for UCSB

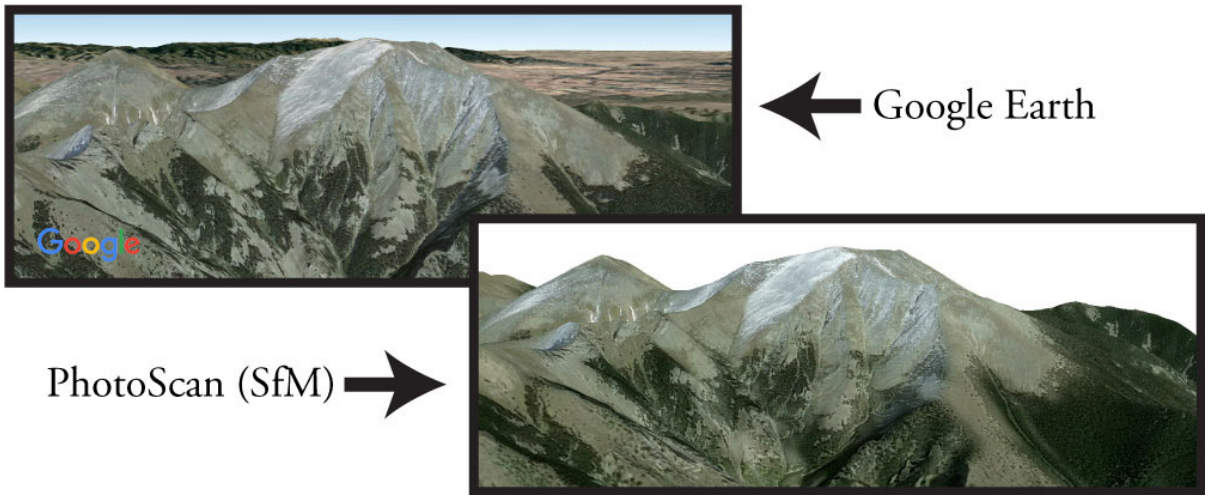


Figure 3.10: Visual comparison of Google Earth and PhotoScan models for Mount Herard

3.4 Discussion

While PhotoScan successfully reconstructed Google Earth's 3D models for all three study areas, analysis of the resulting models showed significant variability in their accuracies compared to the original model and to real-world LiDAR data. The following sections provide a discussion on the limitations of using this approach for scientific applications.

3.4.1 Information content

Since SfM-MVS relies on image content to derive 3D measurements, certain qualities of these images can impact the quality of the final point clouds. These factors include the quality of the camera and its images, the different views of the environment captured by those images, and qualities of the environment itself (Agisoft LLC 2016). While the 3.2 MP resolution of the images used in this study was sufficient for PhotoScan, higher resolution images would have likely produced higher resolution point clouds with possibly greater levels of precision.

However, higher precision does not necessarily translate into higher accuracy. The limited viewing angles of the aerial photos perhaps had greater influence on accuracy than image resolution. All three case studies used single flight lines oriented in a single direction and flown at a single altitude: 300 m above ground for Tokyo and UCSB and 1,000 m above ground for Mount Herard. Moving the "camera" along a horizontal plane produced significantly greater parallax in the horizontal direction than the vertical direction, resulting in lower horizontal error values. Providing a wider range of perspectives would have likely improved the quality of the models. For example, images from multiple altitudes could have improved vertical accuracy.

Surface properties such as texture, color, and reflectiveness played an equally important role in determining model quality. SfM-MVS does not perform well with lightly or

untextured surfaces due to an inability to detect key points in those areas (Agisoft LLC 2016). For this reason, the built-up environments of Tokyo and UCSB with their complex geometries and varied colors produced significantly better results than the all-natural environment of Mount Herard. At Mount Herard, areas with high topographic relief — such as peaks and sharp ridges — produced accurate measurements (±10 m compared to LiDAR) while smooth, rounded, and featureless valleys produced errors of up to 100 m. Use of additional ground control points could have mitigated errors in those problematic areas.

3.4.2 Computation

Producing the point clouds using SfM-MVS required a significant amount of computing resources on a single workstation[3] and took about six hours of processing for each of the six point clouds. Even with the available computing resources and processing times, the software could only generate one point cloud (Tokyo) at the highest quality level while the UCSB and Mount Herard point clouds were generated at the next lower level (i.e., high quality) due to resource limitations. Alternative strategies for producing large point clouds, such as processing and merging smaller areas or using distributed computing, were not explored.

3.4.3 Errors and uncertainties

Models produced using this technique will always contain two types of errors and uncertainties: one between the SfM-MVS point cloud and Google Earth and the other between Google Earth and reality. Errors between the SfM-MVS model and Google Earth

[3]Windows workstation with Intel i7-4790 processor, 32 GB RAM, AMD R9 280 graphics card with 3 GB RAM.

can be characterized through sampling of locations in the Google Earth software, such as those used in this study, although the coarse resolution of vertical measurements can introduce additional uncertainty. However, the absence of quality measures comparing Google's model to reality makes it impossible to characterize the second type of error and uncertainty. The sampling of errors in Google Earth, shown in Table 3.4, showed the degree to which Google's model deviated from reality at Mount Herard, providing an indication of the imperfections — - usually to an unknown degree — - that exist in Google Earth.

3.5 Conclusion

This study demonstrated the feasibility of reconstructing Google Earth's 3D models using SfM-MVS software and screen images from Google Earth. While all three case studies produced point clouds and models that appeared nearly identical to Google's, sample measurements showed varying degrees of errors in the SfM-MVS models: the built-up environments of Tokyo and UCSB exhibited the lowest error values while the natural environment of Mount Herard had a wide range of error values with some exceeding 100 m. Errors in SfM-MVS modeling could be attributed to and mitigated at specific steps in the SfM-MVS workflow. However, errors inherent in Google Earth itself were impossible to quantify without taking actual physical measurements. The variability of SfM-MVS errors and the unknown errors and uncertainties in Google's models makes this technique unsuitable for applications that require accurate measurements. Nevertheless, for many users who have no such requirement, the technique used in this paper can provide a fast, convenient, and affordable approach to modeling the physical world.

Chapter 4

Photogrammetry with inexpensive webcams

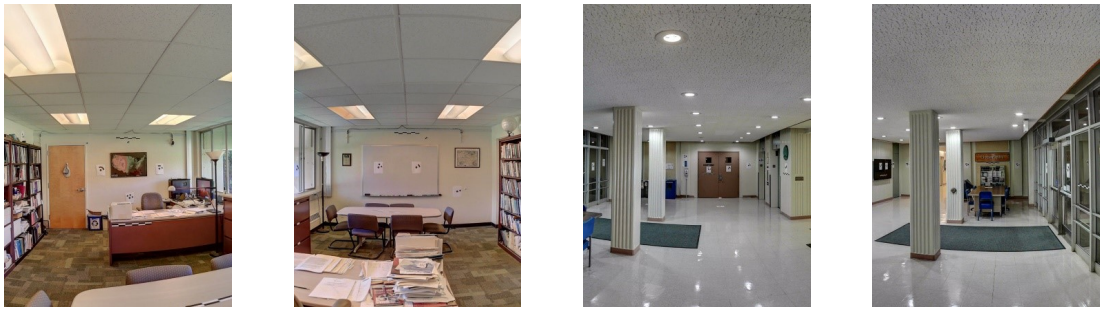
4.1 Introduction

This study explored the feasibility of using structure-from-motion/multi-view stereo (SfM-MVS) for deriving precise 3D indoor measurements using cameras at the lowest end of the consumer digital camera spectrum, namely webcams. It also investigated SfM-MVS performance for a smartphone camera and low- to mid-level consumer digital cameras, as representative models of future miniaturized cameras. Prior studies on use of webcams in photogrammetry have mainly centered on the areas of robotics and simultaneous localization and mapping (SLAM). These studies — such as those from Mansley et al. (2010), Taylor, Boles, and Geva (2007), and Budiharto, Jazidie, and Purwanto (2010) — produced limited-use point clouds that met the expedient needs of robotics at the cost of completeness and precise geometric form. Only two other studies were found to use webcams for precise mapping — Wong (2012) used webcams to map the human head and King (2012) used them to take precise measurements of the heads of newborn

babies. Other studies such as those from Yastikli (2007), Bhatla et al. (2012), and Klein, Li, and Becerik-Gerber (2012) used high quality digital single lens reflex (SLR) cameras — expensive professional or "prosumer" grade cameras with superior optics and sensors. However, no prior study could be found on the use of SfM-MVS with low-cost consumer grade webcams for measuring indoor spaces.

4.2 Study areas

The two areas used for this study included an office space measuring 4 m x 6 m and a lobby area measuring 12 m x 6 m, both selected for their wide variety of physical objects and varied lighting conditions (Figure 4.1). These spaces were in Ellison Hall at the University of California, Santa Barbara, and were accessible at all hours of the night for uninterrupted data collection. Physical objects in both locations such as desks, bookshelves, windows, and memorabilia provided a basis for assessing the geometric accuracy of the resulting point clouds, while a variety of textures and lighting conditions allowed testing of SfM-MVS under various environmental conditions. Noteworthy textures in either of the study areas included polished floor tiles, patterned low-pile carpeting, textured walls, reflective white boards, and transparent windows with reflections. Photographs of the office were taken during both day and night to explore the effects of ambient exterior lighting; however, photos of the lobby were only taken at night due to time constraints. According to the PhotoScan User Manual, these two study areas pushed the limits of where SfM-MVS can operate due to the presence of untextured, shiny, or transparent objects; numerous foreground objects; and very flat objects with low texture (Agisoft LLC 2016).



(a) Office, front (b) Office, back (c) Lobby, left (d) Lobby, right

Figure 4.1: Views of the two study areas

4.3 Equipment and software

Instruments used for this study included two webcams, five digital cameras, an industrial laser scanner, and a laser distance measurer (LDM). The webcams and digital cameras were chosen to represent the spectrum of inexpensive consumer grade cameras, while the LiDAR scanner and LDM provided baseline data for evaluating the resulting point clouds.

4.3.1 Webcams

The two webcams used for this study were the Microsoft LifeCam Studio and the Logitech QuickCam for Notebooks Deluxe, both chosen to represent inexpensive consumer grade webcams at the high- and low-ends. The LifeCam Studio had a maximum native high definition (HD) image resolution of 1920 x 1080 pixels, while the QuickCam had a maximum native VGA resolution of 640 x 480. Both cameras had a market price below $60 in 2014. Specialized high-end webcams were not considered due to their high prices starting at around $500.

4.3.2 Digital cameras, LiDAR, and LDM

This study used five readily-available digital cameras that represented mid-level consumer grade cameras. These included the Nikon D3100 SLR, Canon PowerShot SD1000, Google/LG Nexus 4 smartphone, Sony DSC-TX10, and Panasonic DMC-FZ28. Only the D3100, SD1000, and Nexus 4 were used in both study areas due to equipment and time constraints. Measurements from the Riegl LMS-Z420i light detection and ranging (LiDAR) laser scanner and a Bosch DLR130K handheld LDM served as the comparison baseline for evaluating SfM-MVS performance.

4.3.3 Processing hardware and software

All point cloud generation was done on a laptop computer (Intel i7-4700MQ CPU, 16 GB of RAM, and a dedicated Nvidia GeForce GTX 770M graphics card with 3 GB of video RAM) running Microsoft Windows 7 and the PhotoScan Professional SfM-MVS software produced by Agisoft, LLC. The Riegl RiSCAN PRO software operated the LMS-Z420i laser scanner and provided coarse point cloud registration, while the open source CloudCompare software was used for point cloud visualization, manual fine registration, point cloud cleaning, and taking measurements.

4.4 Workflow

The PhotoScan SfM-MVS process involved study area preparation, photo capture, photo alignment, point cloud generation, point cloud alignment, and data analysis (Figure 4.2). An optional step of camera lens calibration was also performed to account for the unique properties of each camera's lens.

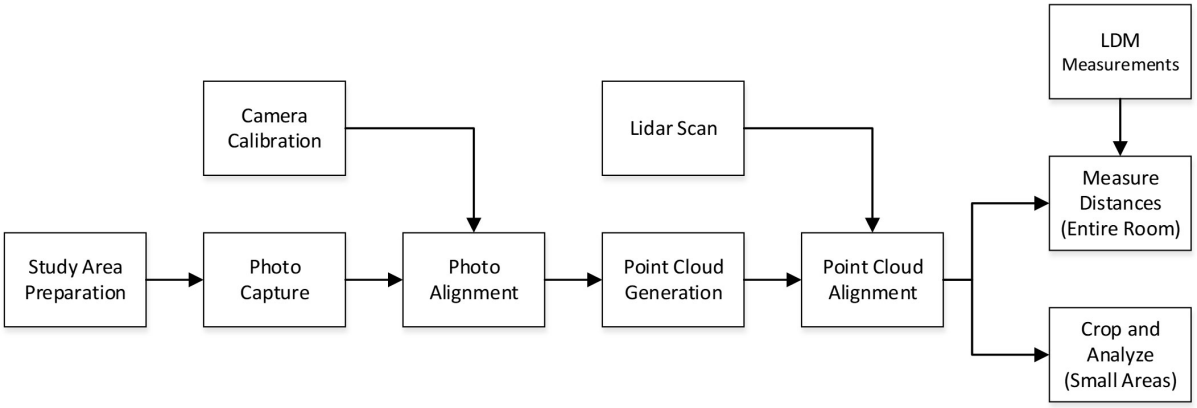


Figure 4.2: Workflow for this study

4.4.1 Camera calibration

Camera calibration corrected for imperfections in the camera's imaging system to improve photo alignment and point cloud generation. While the SfM process automatically provided initial camera calibration during image alignment, a dedicated camera calibration procedure using Agisoft's Lens software provided more comprehensive calibration-level lens adjustments. The Lens software used the Brown pinhole camera distortion model to estimate focal length (f_x, f_y), principal point (c_x, c_y), radial lens distortion (K_1, K_2, K_3, K_4), and the tangential lens distortion (P_1, P_2) parameters using a series of photos of a checkerboard calibration pattern taken at various vantage points (Agisoft LLC 2013).

4.4.2 Reference markers and scale bars

Study area preparation involved placing reference markers and scale bars throughout the office and lobby. While not required, reference markers can assist with generating point clouds of challenging locations. Insufficient texture, reflective surfaces, poor illumi-

nation, and camera noise can impede the software's ability to detect key points in images resulting in poor or failed alignments. Printed reference markers served as *a priori* control points, especially in challenging areas such as white boards and windows. These markers took the form of coded computer-readable markers generated by PhotoScan — printed on letter size paper (8.5 in x 11 in) with a 25 mm center point radius — as well as uncoded markers and scale bars for manual use. Scale bars measuring 60 cm in length with 10 cm subdivisions provided the necessary *a priori* external scaling information to size the point cloud in the correct unit of measurement. In PhotoScan, point-pairs on the scale bars — usually the end points — were manually selected to establish correct distances in the point cloud. Figure 4.3 shows examples of an uncoded marker, a coded marker, and a scale bar.

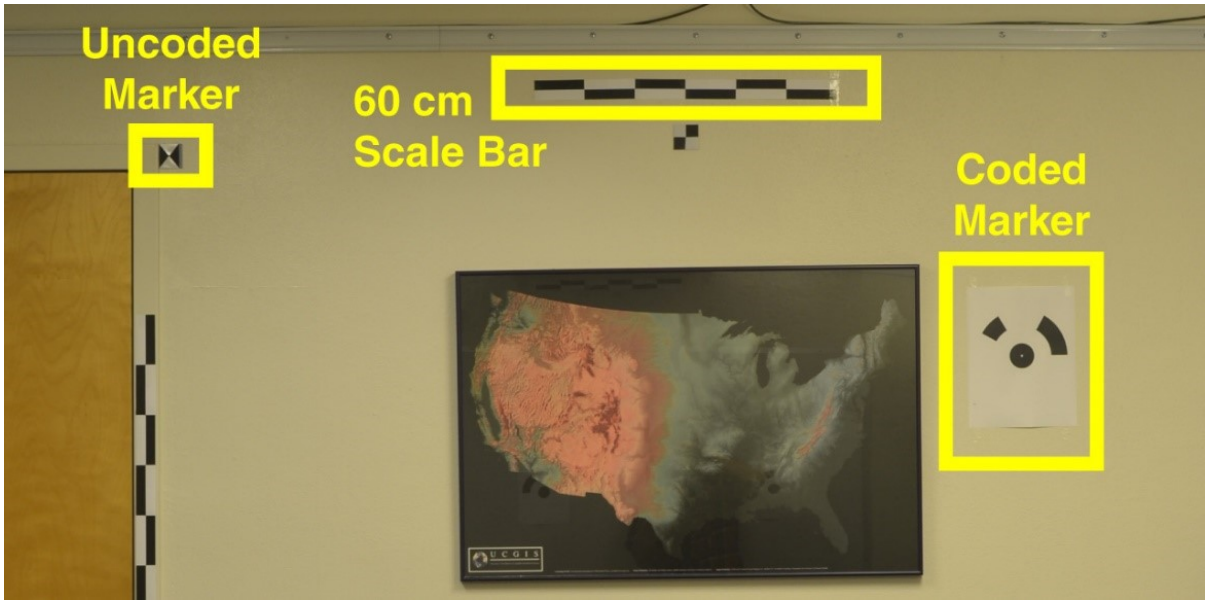
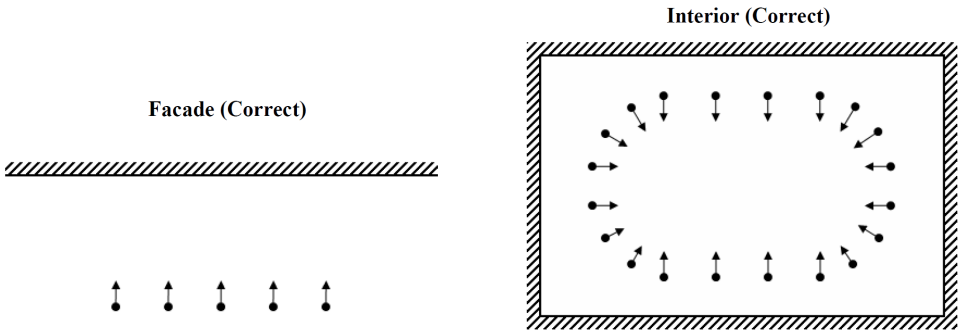


Figure 4.3: Types of printed targets

4.4.3 Photo capture

Images were taken following the camera placement pattern recommended by Agisoft (Figure 4.4), which involved moving the camera in a lateral motion along each wall while pointed at the opposite wall. Corner photos were roughly aimed diagonally across the room and taken in pairs in an attempt to introduce parallax at those locations; these corner photos helped to link photos from one wall to another.



(a) For facade. (b) For room.

Figure 4.4: Recommended camera placement for indoor photogrammetry[1]

Multi-camera setup

Use of a custom-built camera rig shown in Figure 4.5 helped ensure spatial and temporal consistency between the different cameras. This allowed all cameras, except the Nikon D3100, to take images from similar positions, at similar angles, and under similar lighting conditions at about the same point in time. The Nikon D3100 was not attached to this rig due to its size and weight and was attached to a tripod that trailed the camera rig.

[1] Agisoft LLC (2016)

The camera rig consisted of an aluminum angle with camera attachment points, torpedo levels, a tripod, and a tripod dolly to carry a notebook computer and facilitate movement. Three LifeCam cameras were mounted pointing horizontally and 20 degrees from horizontal to provide about 20% vertical overlap in the images. The LifeCam and QuickCam were mounted in landscape mode while the other cameras were mounted in portrait mode.

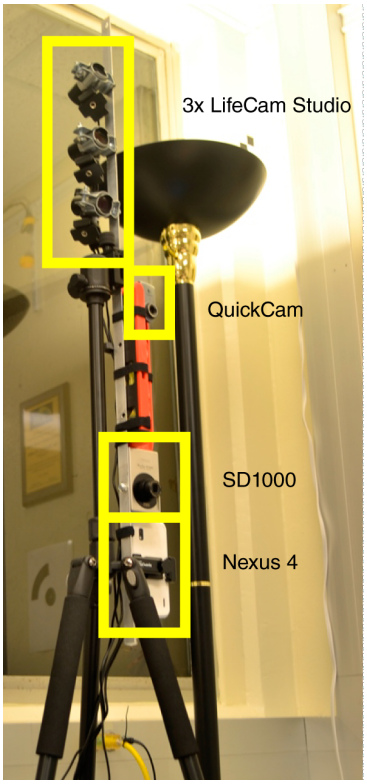


Figure 4.5: Camera mount

Image acquisition software

Software used for capturing the images included a combination of built-in camera software and third-party software. The purpose of using third-party software was to capture an extended range of luminance values for studying the effects of high dynamic

Table 4.1: Cameras and camera settings

Camera	Software	Image Format	ISO	Resolution	Auto-exposure
LifeCam Studio	MATLAB IMAQ	JPG	Auto	2.1 MP	Yes
QuickCam	MATLAB IMAQ	JPG	Auto	0.3 MP	Yes
D3100	Built-In	RAW, JPG	100	14.2 MP	Yes
SD1000	CHDK	JPG	80	7.1 MP	No
Nexus 4	Camera FV-5	JPG	100	8 MP	No
FZ28	Built-In	JPG	100	10.1 MP	Yes
TX10	Built-In	JPG	Auto/Multi	16.2 MP	Tonemapped

range (HDR) photography in a proposed follow-on study; HDR imaging was not used in this study with the exception of photos from the Sony TX10 using its built-in HDR capability. Table 10.7 shows the software and settings used for capturing images from each platform. When possible, this study used photographs taken with autoexposure and auto white balance settings at the lowest ISO setting; where autoexposure photos were not available, the middle exposure of bracketed shots was manually selected through visual inspection.

Image capture

All photos taken for this study were single exposure compressed 8-bit per channel JPEG images, a format which captured luminance data at 256 levels per RGB color, with the exception of photos from the D3100 and TX10. While Agisoft recommended capturing photos in an uncompressed raw format for best results, this study used compressed JPEGs since most webcams and consumer cameras cannot save in raw format.

The D3100 images for the lobby consisted of two different sets of images taken on two separate nights. The first set of images contained images with a short depth of field due to a wide lens aperture resulting in some areas being out of focus. The second set of images contained sharper details produced using a smaller lens aperture, but they lacked the reference markers from the first photo session. Both image sets were combined so that the

resulting image set included reference markers from the first session and sharper details from the second session. These images were captured in the uncompressed Nikon raw 12-bit format and converted to 16-bit TIFs for use in PhotoScan. The final photogrammetric processing for the D3100 images used a combination of TIF and JPEG format images.

4.4.4 Image alignment

The PhotoScan image alignment process involved software detection of reference markers and key points and use of SfM for initial image alignment. Prior to performing photo alignment, PhotoScan detected coded reference markers to assist with the initial SfM alignment step. Next, the software used computer vision to detect key points in each image — this is where the printed markers can help — and applied SfM techniques to estimate the positions from where each image was taken as well as the lens parameters. This step resulted in a sparse 3D point cloud composed of key points that contained errors due to misalignment. Manual alignment was then performed to refine the initial alignment of individual images, which involved manually adding reference points and removing false points caused by reflections and false detections (Figure 4.6). Additional reference markers were added to correspond to uncoded reference markers, prominent features, and coded reference markers that were not automatically detected by the software (Figure 4.7).

In this study, automatic alignment produced mixed results with higher resolution images generally performing better. However, automatic alignment still proved too problematic for the challenging indoor environments, so all image alignment was performed manually.

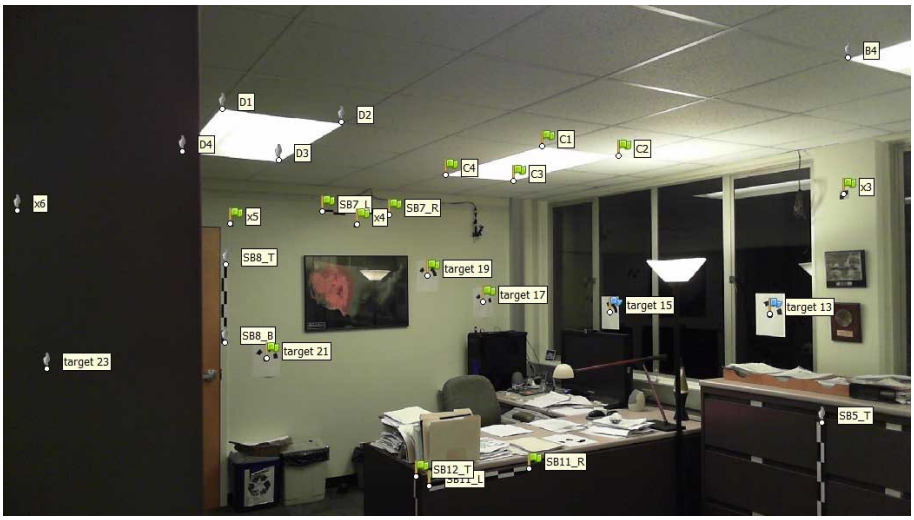
(a) Misidentified marker in reflection (b) Misidentified marker

Figure 4.6: Misidentified markers

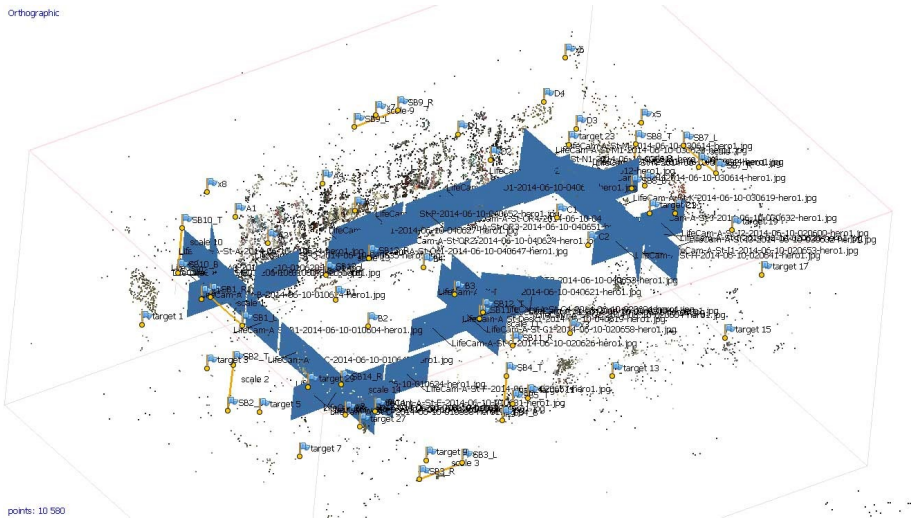
4.4.5 Point cloud generation

Upon completion of image alignment, PhotoScan generated dense point clouds using multi-view stereo (MVS) techniques. PhotoScan provided two simple parameters for dense point cloud generation: quality and depth filtering. Higher quality settings produce denser and more accurate point clouds but require more processing time. Depth filtering impacts the treatment of outlier points which may represent either noise or detailed geometry. The PhotoScan User Manual recommended using mild filtering for scenes with foreground details and aggressive filtering for scenes where small details have low importance.

Photos for all cameras used the highest "ultra high" quality setting and "aggressive" depth filtering, with the exception of the D3100 images for the lobby. Even though the PhotoScan User Manual recommended using mild depth filtering to improve foreground details, aggressive depth filtering was eventually used due to the presence of excessive noise from low quality images, e.g. low resolution or poorly lit images. Insufficient RAM memory on the workstation prevented processing of the D3100 images at "ultra high" quality, so the next lower "high" quality setting with "aggressive" depth filtering was used.



(a) Photo with markers



(b) SfM points with camera positions and markers

Figure 4.7: Single photo with markers and resulting SfM sparse point cloud

4.4.6 Point cloud alignment

Point cloud alignment involved using the CloudCompare software to register the SfM-MVS point clouds with their respective LiDAR point clouds (Figure 4.9). Each alignment required a minimum of four manually selected pairs of points corresponding to the same features in each point cloud, with scale adjustment disabled.

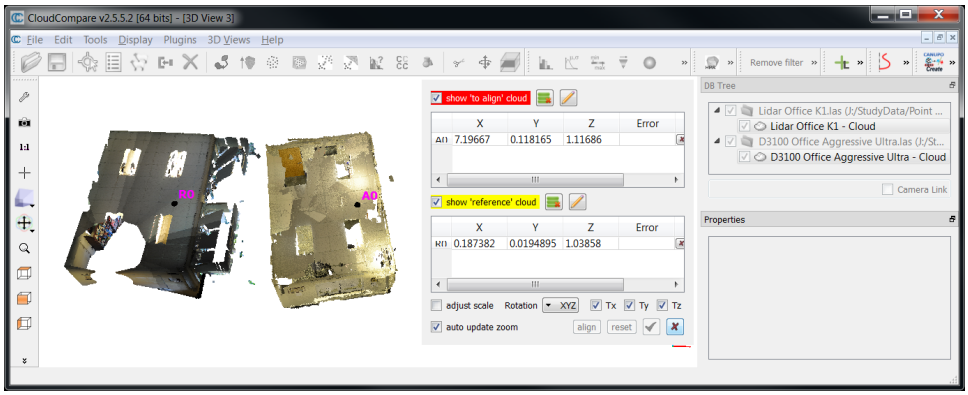


Figure 4.8: Manual point cloud alignment in CloudCompare

4.4.7 Measurement and analysis

Cross-room distances

Cross-room measurements involved using the CloudCompare Point Picking tool to measure linear distances between two manually selected points (Figures 4.9 and 4.10). These measurements were then recorded in a Microsoft Excel spreadsheet for analysis.

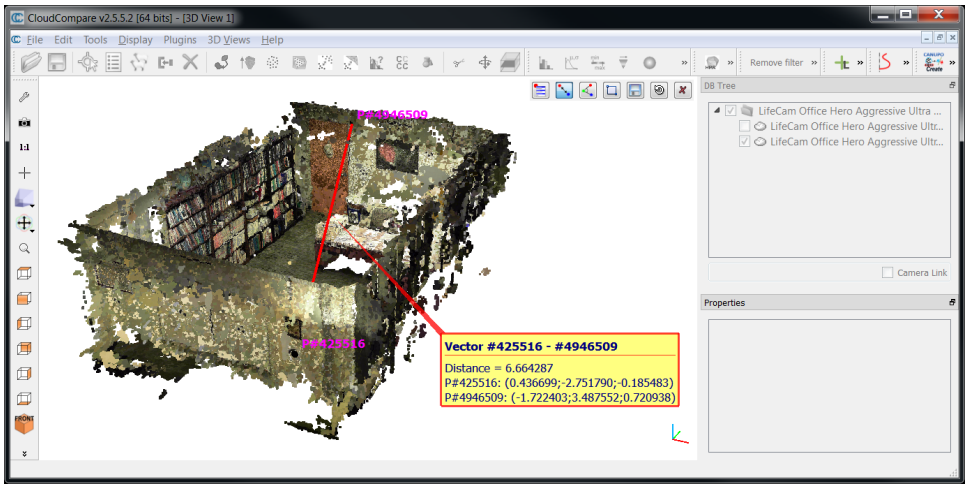
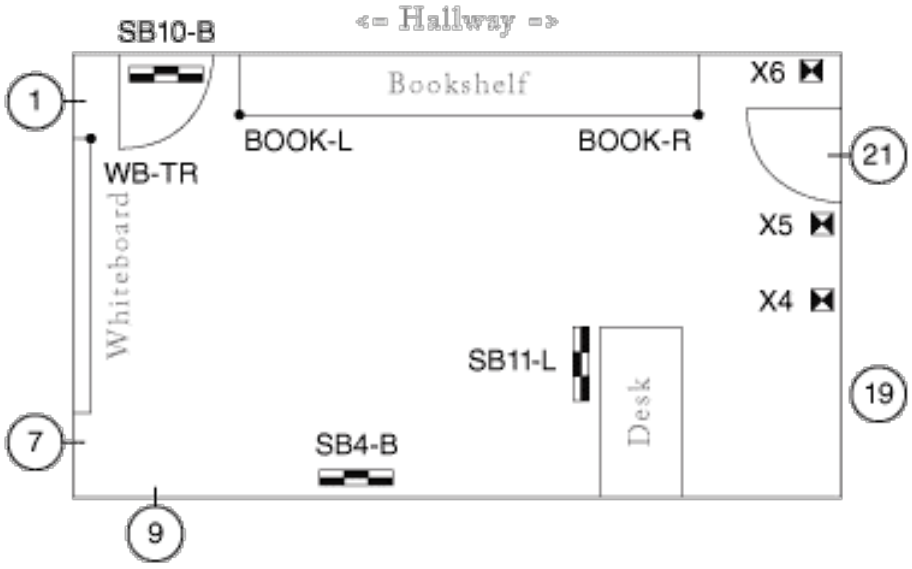
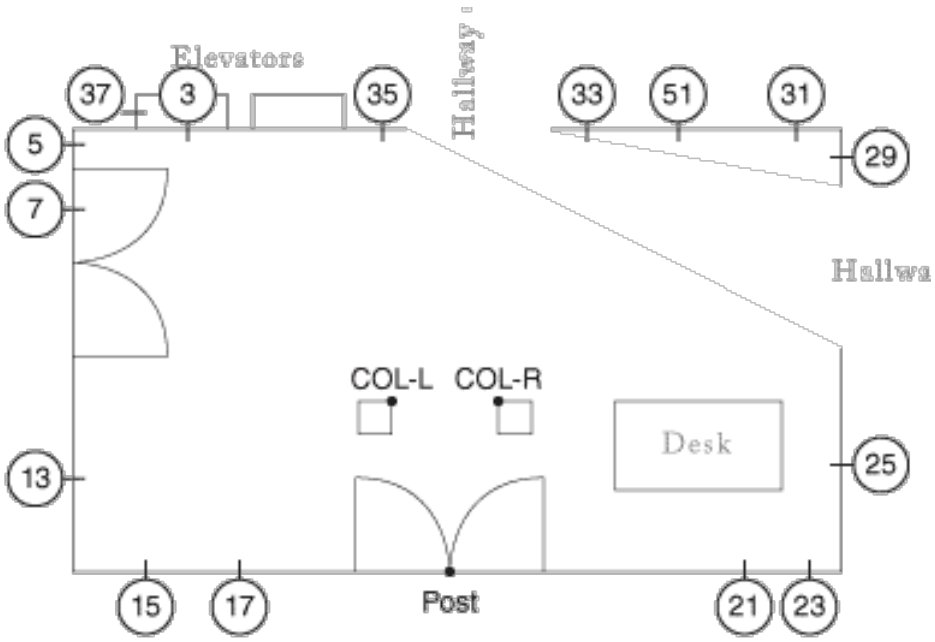


Figure 4.9: Measuring distances in CloudCompare



(a) Office



(b) Lobby

Figure 4.10: Locations of measurement points

Coverage, surface relief, and point density

In addition to evaluating point-to-point distances, this study examined other quality factors in the form of coverage, surface relief reconstruction, and point density. Coverage examined the completeness of point clouds over certain surfaces, surface relief examined the software's ability to capture variations in 3D surfaces, and point density examined the density of points produced.

Four surfaces were converted to digital surface models (DSMs) with 2.5 cm x 2.5 cm cells using ArcGIS Desktop and the "Point to Raster" conversion tool (Figure 4.11). For surface relief, mean out-of-plane[2] point "height" was used as the z-axis in the ArcGIS tool. The Raster Calculator tool then subtracted the LiDAR DSM values from each camera's DSM to produce a raster of residuals. Point density involved taking the number of points in each cell, dividing by the cell's area, and recording the value as the cell's z-axis for every cell; the mean "height" of these cells reflected the overall mean value. Coverage involved using the "IsNull" tool on the surface relief DSM to identify voids in the raster; statistics for the percents of occupied and empty cells was found in the resulting datasets' properties. All three evaluations omitted the bordering cells to prevent boundary effects.

[2]I.e., in the direction normal to the plane.

Table 4.2: SfM-MVS data summary, office (night)

Camera	Resolution	Number of Images	Quality/ Filtering	Processing Time	Number of Points
LifeCam Single	2.1 MP	45	Ultra / Aggressive	0h:16m:32s	5.0M
LifeCam Triple	2.1 MP	135	Ultra / Aggressive	1h:1m:47s	11.0M
QuickCam	0.3 MP	40	Ultra / Aggressive	0h:1m:18s	1.1M
D3100	14.2 MP	68	Ultra / Aggressive	6h:4m:10s	116.0M
SD1000	7.1 MP	41	Ultra / Aggressive	2h:39m:38s	15.5M
Nexus 4	8 MP	41	Ultra / Aggressive	2h:57m:17s	12.3M

Table 4.3: SfM-MVS data summary, office (day)

Camera	Resolution	Number of Images	Quality/ Filtering	Processing Time	Number of Points
LifeCam Triple	2.1 MP	144	Ultra / Aggressive	N/A	11.4M
QuickCam	0.3 MP	46	Ultra / Aggressive	0h:3m:24s	1.3M
FZ28	10.1 MP	46	Ultra / Aggressive	4h:50m:9s	36.5M
TX10	16.2 MP	50	Ultra / Aggressive	3h:37m:25s	18.1M

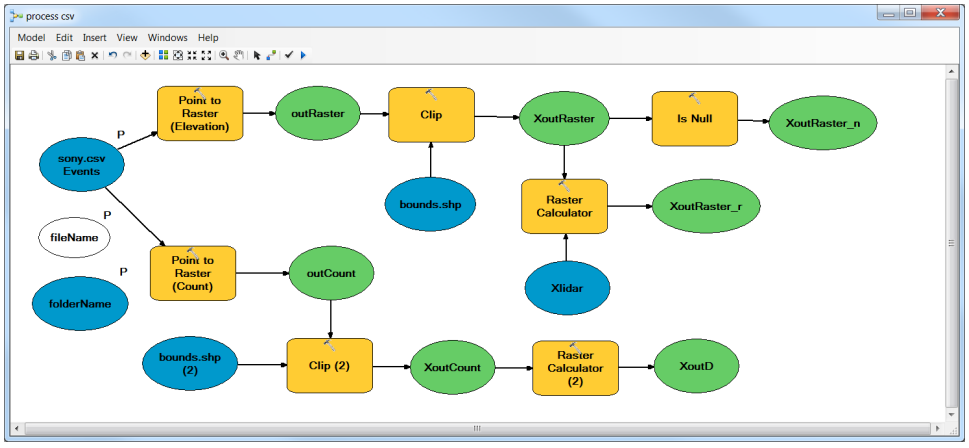
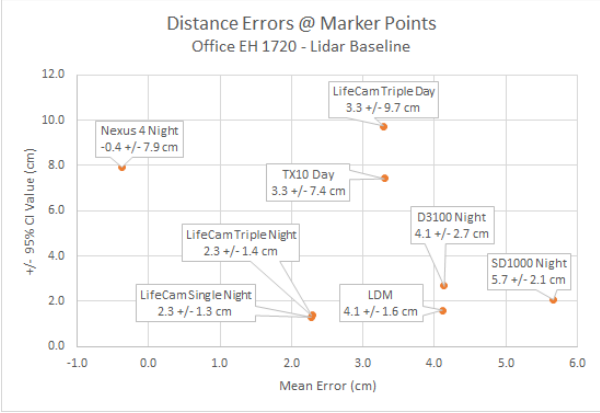


Figure 4.11: ArcGIS ModelBuilder workflow for coverage, surface relief, and point density

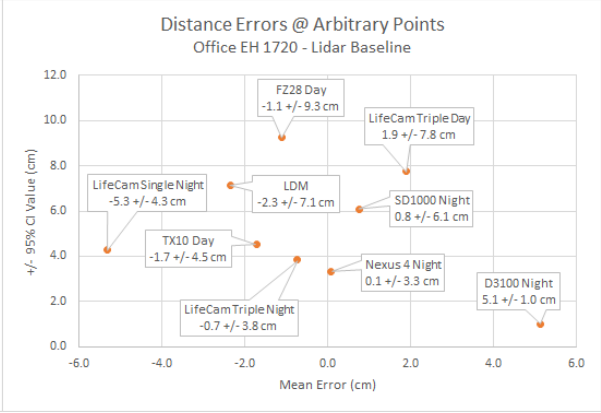
Table 4.4: SfM-MVS data summary, lobby (night)

Camera	Resolution	Number of Images	Quality/ Filtering	Processing Time	Number of Points
LifeCam Single	2.1 MP	51	Ultra / Aggressive	1h:6m:13s	4.5M
LifeCam Triple	2.1 MP	153	Ultra / Aggressive	1h:16m:56s	12.3M
D3100	14.2 MP	226	High / Aggressive	3h:35m:12s	52.3M
SD1000	7.1 MP	51	Ultra / Aggressive	2h:53m:10s	20.9M
Nexus 4	8 MP	50	Ultra / Aggressive	6h:36m:32s	13.7M

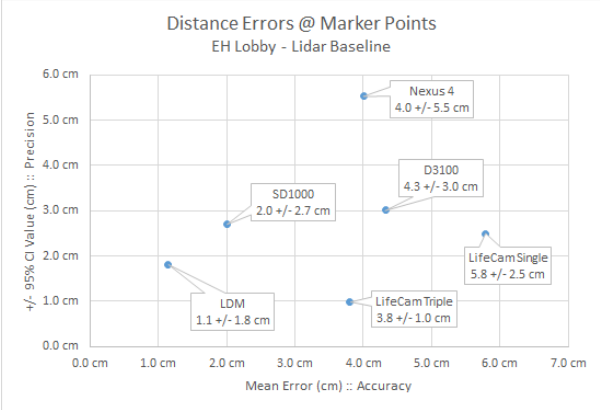
4.5 Results



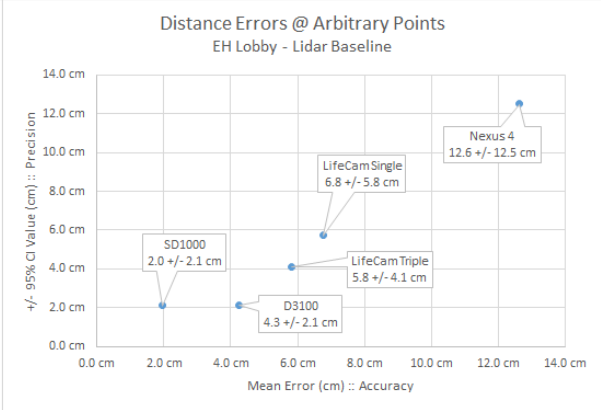
(a) Office, anchor points



(b) Office, non-anchor points



(c) Lobby, anchor points

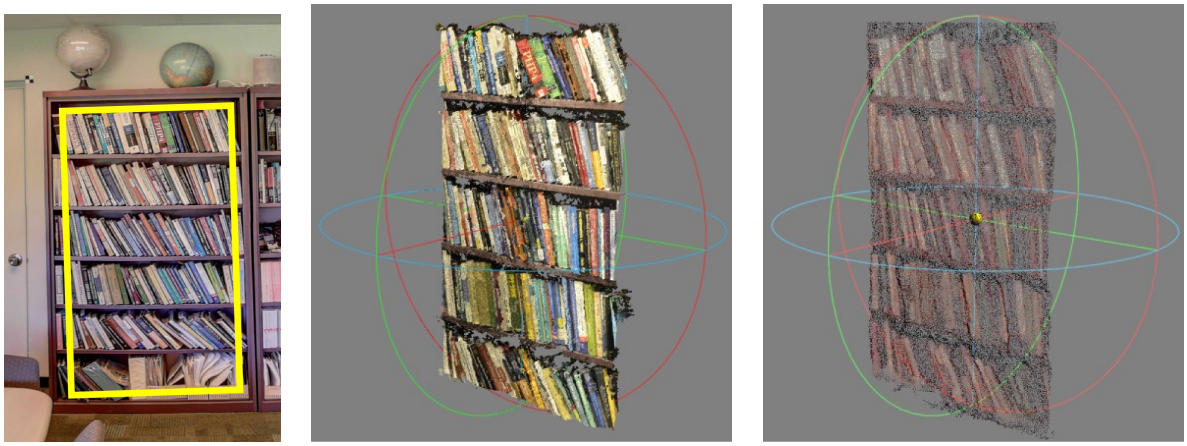


(d) Lobby, non-anchor points

Figure 4.12: Cross-room distance metrics

Table 4.5: Point cloud statistics for bookshelf

Camera	Coverage (%)	Point Density (pts/cm^2)
LiDAR	99.6%	3
LifeCam Single Night	100.0%	26
LifeCam Triple Night	100.0%	36
QuickCam Night	99.2%	5
Nexus 4 Night	99.8%	89
SD1000 Night	99.4%	81
D3100 Night	97.7%	252
TX10 Day	100.0%	46
LifeCam Triple Day	100.0%	33
QuickCam Day	99.9%	5
Panasonic Day	100.0%	111

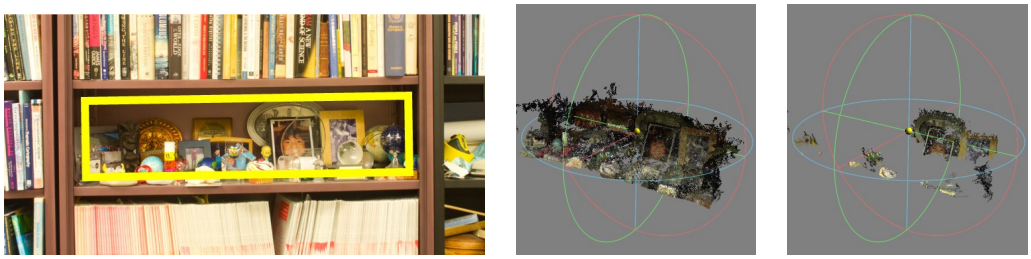


(a) Image (b) Nikon D3100 (c) QuickCam Night

Figure 4.13: Bookshelf point clouds

Table 4.6: Point cloud statistics for display shelf

Camera	Coverage (%)	Point Density (pts/cm^2)
LiDAR	100.0%	1
LifeCam Single Night	96.3%	7
LifeCam Triple Night	99.7%	13
QuickCam Night	98.7%	1
Nexus 4 Night	58.9%	10
SD1000 Night	56.1%	13
D3100 Night	84.2%	52
TX10 Day	99.5%	9
LifeCam Triple Day	100.0%	9
QuickCam Day	100.0%	2
Panasonic Day	61.1%	16

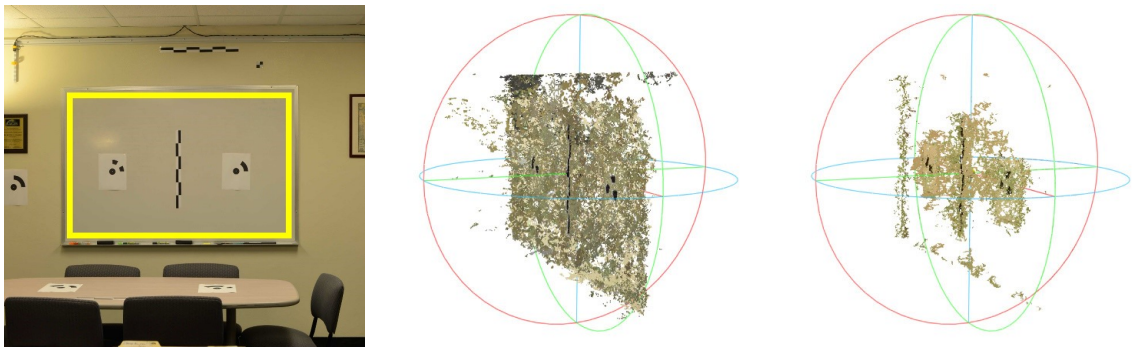


(a) Image (b) LifeCam Triple Night (c) Canon SD1000

Figure 4.14: Display shelf point clouds

Table 4.7: Point cloud statistics for whiteboard

Camera	Coverage (%)	Point Density (pts/cm^2)	Residual STD (cm)
LiDAR	100.0%	3	Baseline
LifeCam Single Night	84.7%	8	2.7
LifeCam Triple Night	96.6%	10	3.3
QuickCam Night	90.9%	2	7.5
Nexus 4 Night	44.7%	30	2.1
SD1000 Night	54.9%	21	1.5
D3100 Night	65.6%	71	1.1
TX10 Day	64.3%	9	2.4
LifeCam Triple Day	84.0%	7	4.5
QuickCam Day	64.2%	1	7.7
Panasonic Day	26.6%	21	1.8

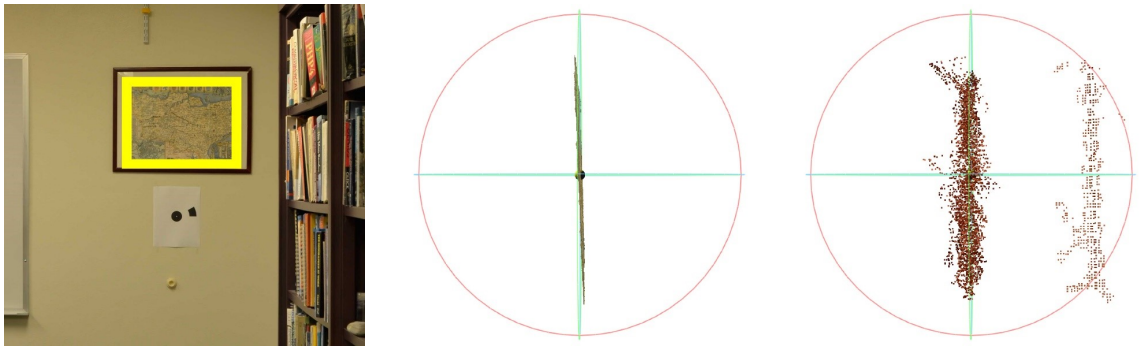


(a) Image (b) LifeCam Triple Night (c) Nexus 4

Figure 4.15: Whiteboard point clouds

Table 4.8: Point cloud statistics for picture

Camera	Coverage (%)	Point Density (pts/cm^2)	Residual STD (cm)
LiDAR	100.0%	2	Baseline
LifeCam Single Night	100.0%	15	1.5
LifeCam Triple Night	100.0%	22	0.9
QuickCam Night	100.0%	3	3.1
Nexus 4 Night	100.0%	33	0.7
SD1000 Night	100.0%	56	0.8
D3100 Night	100.0%	167	0.4
TX10 Day	100.0%	24	0.5
LifeCam Triple Day	100.0%	16	0.7
QuickCam Day	100.0%	2	0.8
Panasonic Day	100.0%	55	0.5

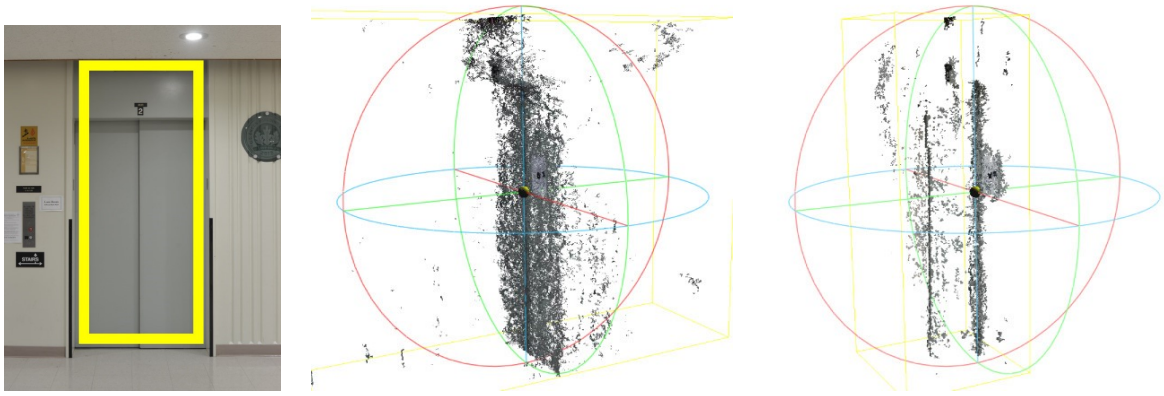


(a) Image (b) Nikon D3100 (c) QuickCam Night

Figure 4.16: Picture point clouds

Table 4.9: Point cloud statistics for elevator

Camera	Coverage (%)	Point Density (pts/cm^2)	Residual STD (cm)
LiDAR	100.0%	1	Baseline
LifeCam Single	68.7%	5	17.3
LifeCam Triple	92.5%	7	17.8
D3100	56.1%	16	9.2
SD1000	64.2%	17	26.5
Nexus 4	40.6%	12	23.0

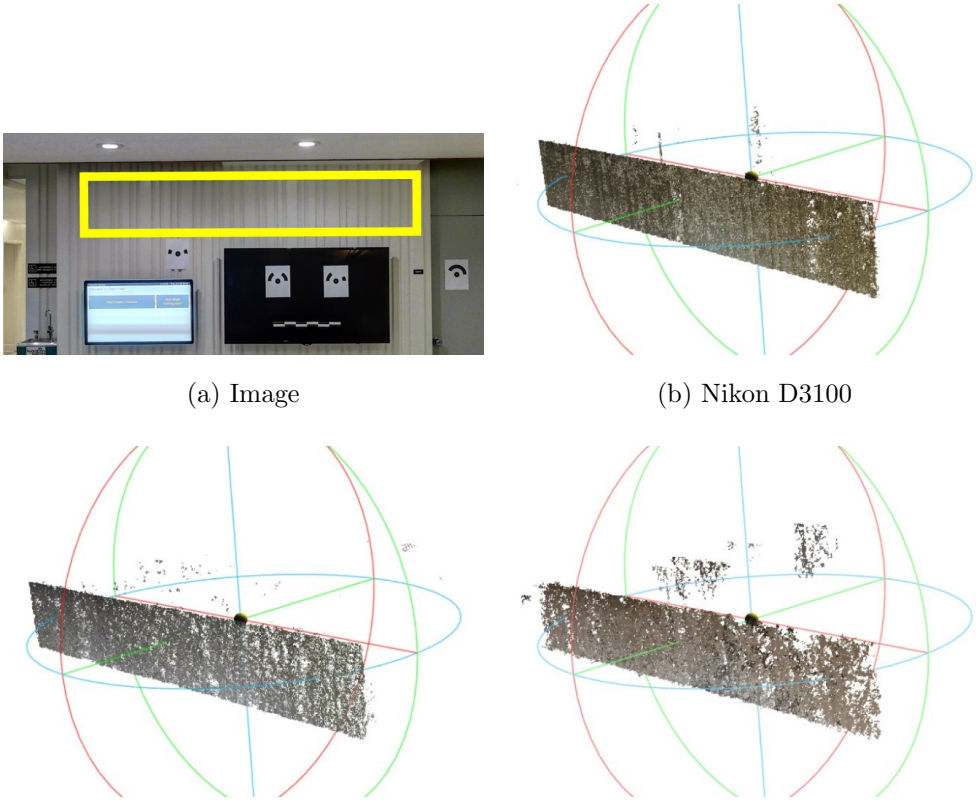


(a) Image (b) LifeCam Triple (c) Nexus 4

Figure 4.17: Elevator point clouds

Table 4.10: Point cloud statistics for ribbed wall

Camera	Coverage (%)	Point Density (pts/cm^2)	Residual STD (cm)
LiDAR	100.0%	1	Baseline
LifeCam Single	99.1%	4	6.6
LifeCam Triple	100.0%	10	6.8
D3100	100.0%	34	2.4
SD1000	100.0%	32	7.7
Nexus 4	99.6%	24	18.1



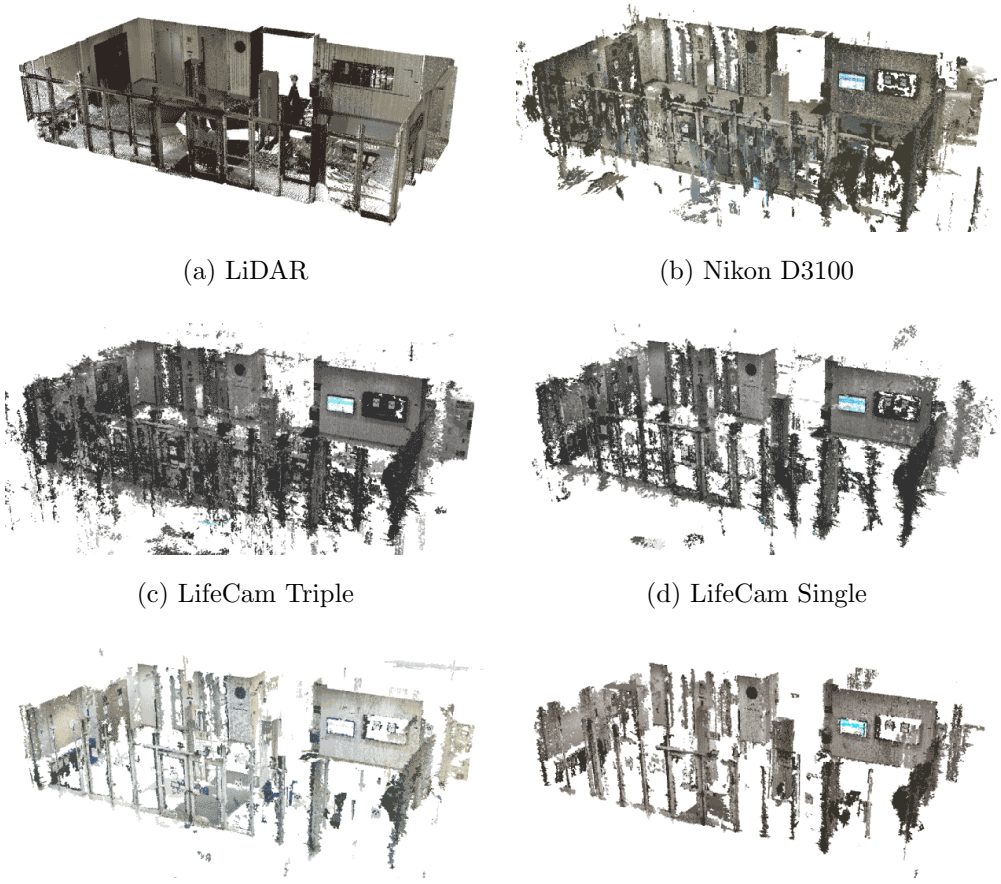
(a) Image

(b) Nikon D3100

(c) LifeCam Triple

(d) Nexus 4.

Figure 4.18: Ribbed wall point clouds



(a) LiDAR

(b) Nikon D3100

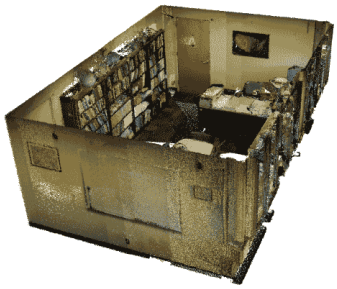
(c) LifeCam Triple

(d) LifeCam Single

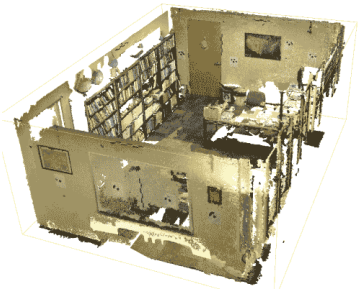
(e) Canon SD1000

(f) Nexus 4

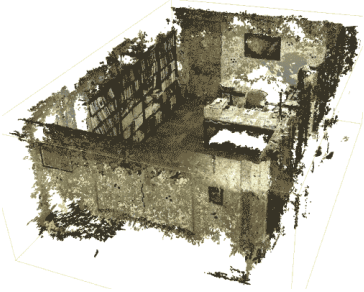
Figure 4.19: Point clouds of the lobby



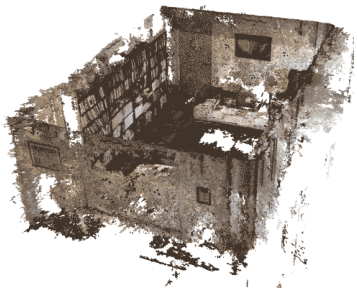
(a) LiDAR



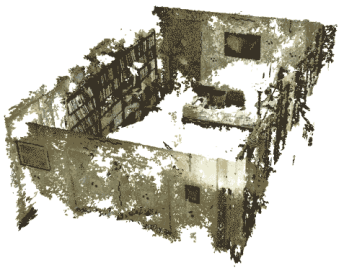
(b) Nikon D3100



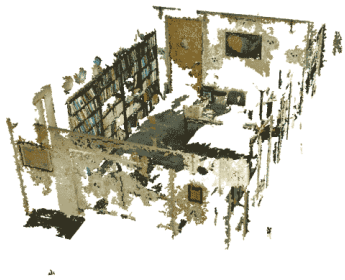
(c) LifeCam Triple Night



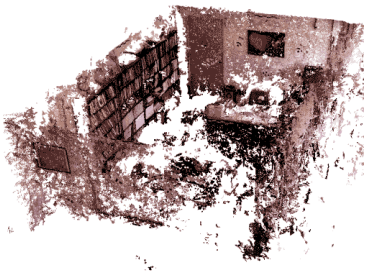
(d) LifeCam Triple Day



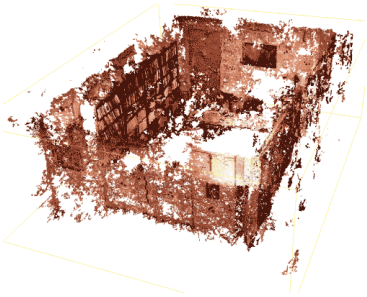
(e) LifeCam Single Night



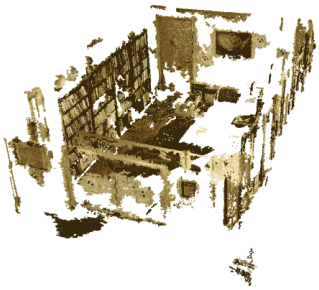
(f) Canon SD1000



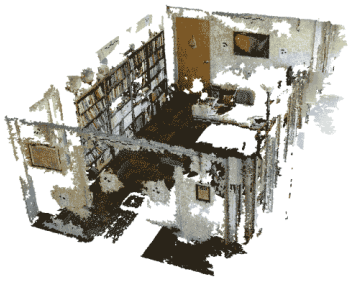
(g) QuickCam Day



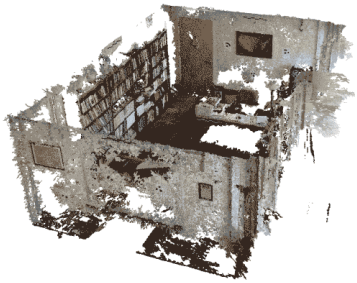
(h) QuickCam Night



(i) Nexus 4



(j) Panasonic FZ28



(k) Sony TX10

Figure 4.20: Point clouds of the office

4.6 Discussion

Results of this study showed that the high resolution Microsoft LifeCam Studio webcam produced point clouds with accuracy and coverage values matching those produced using more expensive digital cameras. However, the low resolution webcam (640x480 pixels) produced coarse point clouds with excessive levels of noise (low precision) and errors in the smaller office environment and failed to produce a point cloud for the larger lobby.

4.6.1 Data assessment

Cross-room distances

LiDAR measurements served as the baseline for assessing quality of distances measurements. Six of the eight camera platforms (LifeCam Single, LifeCam Triple Night, LifeCam Triple Day, D3100, SD1000, and TX10) produced measurements with mean distance errors below 0.6% and precision values below 0.8% at the 95% confidence level, or roughly 6 cm (2.4 in) accuracy and 8 cm (3.2 in) precision for a 10 m (33 ft) distance measurement. The low resolution QuickCam webcam consistently produced measurements with distance errors in excess of 7% in low light conditions and 2% in daylight, or 70 cm and 20 cm per 10 m, respectively. The Nexus 4 produced inconsistent results with accuracies ranging between 0.01% and 1.3% and precision values between 0.3% and 1.3%. There was insufficient data from the FZ28 to analyze its capabilities, although a partial analysis showed accuracy and precision of around 0.1% and 0.9%, respectively. Figure 4.12 provides a summary of these results.

The LifeCam Studio webcam (2.1 MP) produced accuracy and precision values across both study areas that matched or exceeded the values from the traditional digital cameras with larger lenses and higher resolution sensors (Table 4.11). However, the very low

Table 4.11: Estimated distance accuracy and precision values for the LifeCam Studio webcam with 95% CIs using t-values

	Office	Lobby	Cumulative
	(Mean Distance Error ± 95% CI in cm per 10 m)		
LifeCam Studio (Single) — Night	-5.3 ± 4.3	6.8 ± 5.8	1.7 ± 5.0
LifeCam Studio (Triple) — Night	-0.7 ± 3.8	5.8 ± 4.1	3.1 ± 3.2
LifeCam Studio (Triple) — Day	1.9 ± 7.8	N/A	1.9 ± 7.8
Cumulative	-1.4 ± 3.0	6.3 ± 3.1	2.3 ± 2.5

Table 4.12: Estimated distance accuracy and precision values for the QuickCam webcam with 95% CIs using t-values

Mean Distance Error	Office	Lobby
(Mean Distance Error ± 95% CI in cm per 10 m)		
QuickCam — Night	-70.8 ± 3.2	N/A — Failed
QuickCam — Day	20.5 ± 8.1	N/A

resolution QuickCam webcam (0.3 MP) produced the highest error values by underestimating office distances by 8%, and its images of the lobby failed to work in PhotoScan (Table 4.12).

Coverage

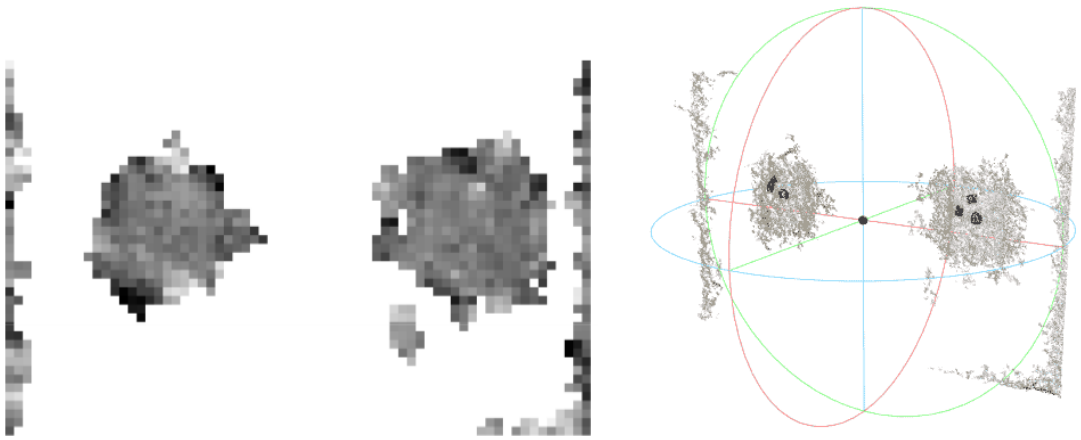
Coverage measured the level at which a point cloud covered a surface after being converted into a 2.5D digital surface model (DSM) with 2.5 cm x 2.5 cm grids. In general, surfaces with greater texture — either in surface roughness or color variation — consistently had greater coverage for all camera models with nearly 100% coverage for highly textured surfaces (Table 4.13). Conversely, smooth surfaces, uniformly colored surfaces, and poorly lit areas resulted in poor coverage with a high percentage of voids and gaps, with nearly 0% coverage if the effects of printed reference markers are removed. These results highlight the challenges posed by low texture and poor lighting conditions for using passive methods such as SfM-MVS in indoor environments.

Table 4.13: Coverage

Mean Coverage (%)	Bookshelf	Display Shelf	White-board	Picture	Elevator	Ribbed Wall
LifeCam Studio (Single) — Night	100.0%	96.3%	84.7%	100.0%	68.7%	99.1%
LifeCam Studio (Triple) — Night	100.0%	99.7%	96.6%	100.0%	92.5%	100.0%
LifeCam Studio (Triple) — Day	100.0%	100.0%	84.0%	100.0%	N/A	N/A
QuickCam — Night	99.2%	98.7%	90.9%	100.0%	N/A	N/A
QuickCam — Day	99.9%	100.0%	64.2%	100.0%	N/A	N/A

Surface relief

Surface relief measurements assessed the ability of SfM-MVS to measure out-of-plane surfaces. Since the metrics omitted voids in the DSM, the surface relief metrics provided more accurate assessments for surfaces with higher coverage values; results with high-void surfaces are suspect. Additionally, the use of physical reference markers skewed results for surfaces that otherwise would have produced large voids; for example, the featureless white board would likely have turned into a large void without the paper targets as shown in Figure 4.14.



(a) Residuals (b) Point cloud

Figure 4.21: Marker bias on whiteboard data

Table 4.14: Surface relief residuals

RMSE of Surface Relief (cm) Compared to LiDAR Baseline	Bookshelf	Display Shelf	White-board	Picture	Elevator	Ribbed Wall
LifeCam Studio (Single) — Night	N/A	N/A	2.7	1.5	17.3	6.6
LifeCam Studio (Triple) — Night	N/A	N/A	3.3	0.9	18.8	6.8
LifeCam Studio (Triple) — Day	N/A	N/A	4.5	0.7	N/A	N/A
QuickCam — Night	N/A	N/A	7.5	3.1	N/A	N/A
QuickCam — Day	N/A	N/A	7.7	0.8	N/A	N/A

Table 4.15: Mean point density

Mean Point Density (Points per cm^2)	Bookshelf	Display Shelf	White-board	Picture	Elevator	Ribbed Wall
LifeCam Studio (Single) — Night	26	7	8	15	5	4
LifeCam Studio (Triple) — Night	36	13	10	22	7	10
LifeCam Studio (Triple) — Day	33	9	7	16	N/A	N/A
QuickCam — Night	5	1	2	3	N/A	N/A
QuickCam — Day	5	2	1	2	N/A	N/A

Point density

The same limitations on surface relief metrics apply to the point density metrics, i.e. surfaces with high percentages of voids may be measuring reference targets and boundary noise instead of the surface itself (Table 4.15). In all cases, the mean point density value should be viewed with caution since a given surface can have a wide density range with extreme outlier values. Nonetheless, the overall mean value can still provide a rough indication of how different environmental conditions can impact the density of point clouds.

Point cloud generation times

Generating point clouds in PhotoScan required a significant amount of time on the order of hours (Figure 4.22) using a notebook computer equipped with a high-end Intel Core i7-4700MQ quad core processor, 16 GB of RAM, and a dedicated Nvidia GeForce GTX 770M graphics card with 3 GB of dedicated video RAM. Enabling OpenCL in

PhotoScan reduced processing times by distributing some of the point cloud workload to the graphics processing unit (GPU) on the video card; however, each iteration of building a point cloud still required over two hours on average to complete. The use of a desktop workstation with more powerful CPU and GPU processors could have shortened processing times.

While the very low resolution QuickCam (not shown in Figure 4.23) had the shortest processing times, it produced the most errors and failed to work in the lobby .

Figure 4.23 shows that the LifeCam Studio — a high quality, high resolution webcam — was a more efficient platform for creating point clouds of the office and lobby compared to the other more capable digital cameras. LifeCam photos required less than half the processing time of the point-and-shoot digital cameras and less than a quarter of the processing time for the SLR camera, an instrument of choice in many close-range photogrammetry studies to date.

4.6.2 Workflow discussion

Duration of entire workflow

While computation required a significant amount of time, the manual portions of the PhotoScan workflow — site preparation, photo capture, and photo alignment — took even longer. Each photo capture session required about six hours due to the number of cameras involved, the need to capture multiple exposures for high dynamic range (HDR) analysis, and time spent resolving technical difficulties with equipment. Manual photo alignment, including the semi-automated and manual placement of reference markers, required about nine hours to perform for the first photo set at each location and about six hours for all other photo sets once the initial reference markers were set.

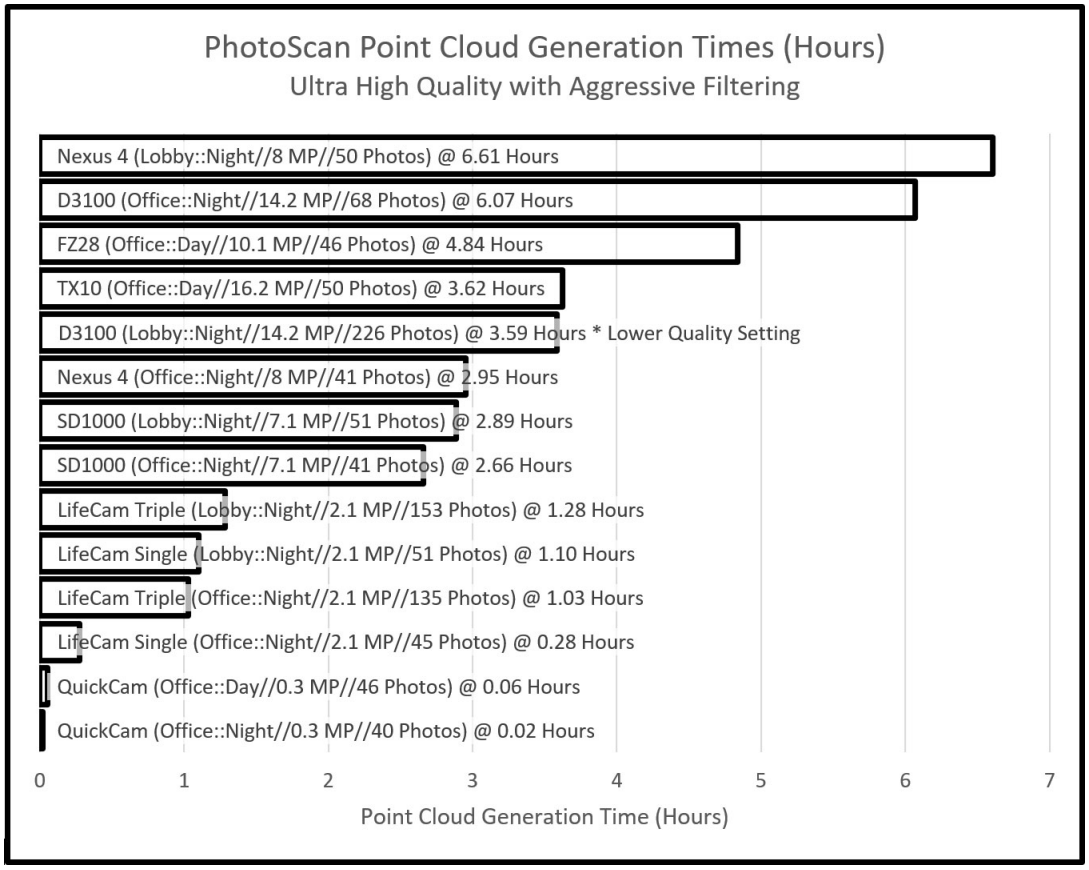


Figure 4.22: PhotoScan point cloud generation times

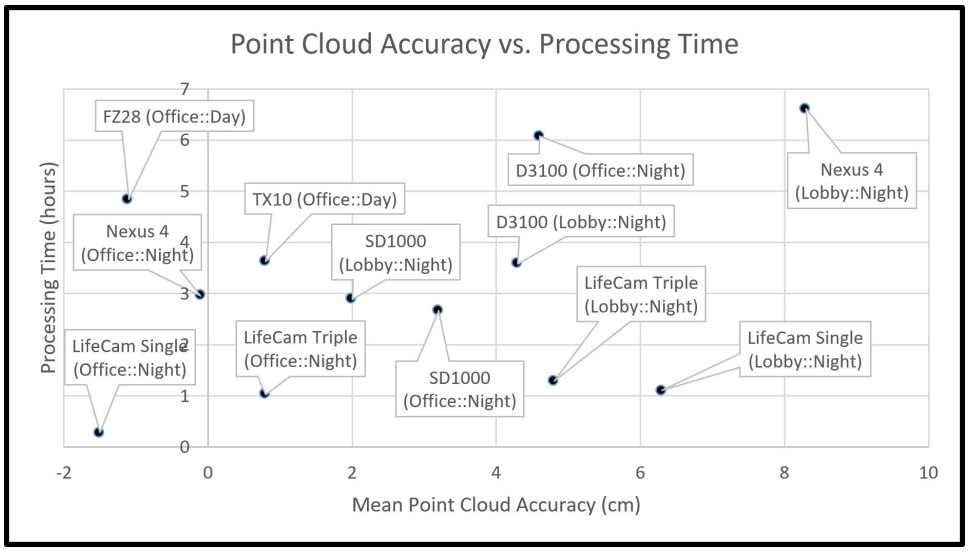


Figure 4.23: Point cloud accuracy vs. processing times

The total time spent on the entire workflow for each individual photo shoot was over 50 hours, not including mobilization/demobilization times nor the time required to correct errors and regenerate point clouds. Therefore, the entire work flow for the three photo shoots performed for this study took over 150 hours to perform.

Reducing the camera to one LifeCam Studio webcam taking a single autoexposure shot at each station would have reduced the workflow time to about 9 hours, i.e. 1 hour for site preparation, 1 hour for the photo shoot, 6 hours for the manual alignment, and 1 hour for point cloud generation. Using the three-camera setup would have resulted in a workflow time of about 12 hours.

Skill factor

As "automated" as the SfM-MVS process may have appeared, the final quality of the resulting point cloud depended largely on non-computer human factors. These factors included the effective planning of shooting locations, photography skills to ensure sharp and well-exposed images, strategic placement of reference markers and scale bars, selection of marker points in the software, and understanding the order in which to manually align photo pairs.

For example, improper planning resulted in a misaligned section of the floor in the D3100 office point cloud due to insufficient photo coverage of that area (Figure 4.24); this problem was mitigated through the extensive use of manually inserted reference points and several iterations of manually aligning individual photos. In another example with the D3100, using an overly wide camera lens aperture caused portions of photos to lose focus, requiring a re-shoot of the entire lobby; reducing the aperture resulted in sharper photos (Figure 4.25).

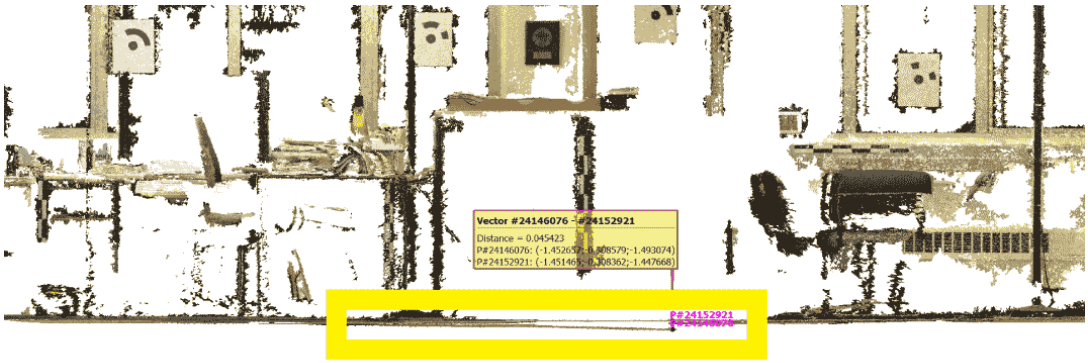


Figure 4.24: Side view of office point cloud from D3100 photos. Note the 4.5 cm vertical gap in the floor.

(a) Foreground and background out of focus (f/4.8 aperture)

(b) Foreground and background in focus (f/16 aperture)

Figure 4.25: Effect of lens aperture on depth of field

Cost and alternatives

The total cost of the setup using three LifeCam Studio webcams was approximately $850 USD, which included the cost of the webcams, mounting hardware, tripod and dolly, and Agisoft PhotoScan Professional (educational license). Use of the commercial PhotoScan Professional license would have increased the cost to $3,800 USD. These costs compared favorably with terrestrial LiDAR scanners which started at about $70,000 USD for a new instrument or $15,000 for a used one in 2014, excluding routine maintenance costs. The webcam system also costed many times less than the portable ZEB1 handheld LiDAR system ($22,000 in 2014).

However, other low cost alternatives existed for 3D point cloud generation at the time of the study. Websites such as Microsoft Photosynth and AutoDesk 123D Catch provided basic SfM-MVS services on a "freemium" basis. Additionally, structured light solutions such as the Matterport 3D camera ($4,500) and Paracosm Scanning Kit ($2,999) cost about the same as this study's webcam system with a commercial license. While the structured light solutions required minimal user involvement in data processing, the technology at the time introduced a whole other set of limitations such as limited range and interference with intense sunlight.

4.6.3 Study limitations and recommendations

Changes in scenery between photo captures

Since data was acquired on four separate days over the span of four months, changes in the scenery impacted the ability to provide consistent measurements between the various point clouds. Major changes included the repositioning of a desk with a reference marker in the office, the repositioning of various furniture including a large conference table in

each location, and the removal of paper reference markers between data captures (which make it difficult if not impossible to find baseline measurement points).

Small sample sizes

The removal or movement of physical reference markers resulted in a very small dataset for comparing photogrammetry data with LiDAR data for reference marker locations. This required estimating the location of corresponding points in the LiDAR point cloud to match locations from photogrammetry and reduced the sample sizes to five or fewer points for the lobby and to three points for the office.

Confidence intervals for these small datasets were determined using simple t-scores which may have overestimated the precision but were required to capture the uncertainty associated with the mean error value. Monte Carlo simulation was considered as an alternative to using the t-score but was not performed in this study.

Univariate statistics

For the sake of simplicity and time, this study used simple univariate statistics to characterize point clouds which exist in 3D space. A more rigorous approach would have been to use multivariate spatial statistics for characterizing the nature and relationship of the photogrammetry point clouds to the LiDAR point clouds.

Automatic camera calibration

Camera calibration allows PhotoScan to correct for distortions in photos caused by imperfections in camera lens and imaging systems. However, the decision to disable automatic camera calibration likely resulted in a greater amount of errors in the point clouds in this study.

High dynamic range (HDR) photography

One of the major challenges of taking photos during daylight hours was the extreme contrast between bright and dark areas near windows. HDR can significantly expand the luminance range of photos to capture details in both dark and bright areas in the same photograph (Figure 4.26). A future study can examine the impact of HDR on close-range photogrammetry using HDR data collected during this study.

(a) Conventional single exposure image

(b) High dynamic range image

Figure 4.26: Comparison of conventional exposure and high dynamic range images

4.7 Conclusion

As indoor mapping and modeling becomes increasingly integrated with everyday life, the demand for gathering measurements of indoor spaces will continue to grow. This study explored one approach to generating point cloud measurements using simple inexpensive webcams and a popular SfM-MVS photogrammetry software package. The results showed that the Microsoft LifeCam Studio (a high quality mass market webcam costing under $60 USD) rivaled more capable and more expensive digital cameras and SLRs in producing point clouds of moderate sized rooms with accuracies and precisions of about 1% at the 95% confidence level compared to measurements from an industrial LiDAR scanner. Point clouds from LifeCam photos also required fifty percent less time to generate in PhotoScan compared to point-and-shoot cameras and over seventy-five percent less time compared to the SLR — a significant advantage considering that this process can take hours to complete.

Although webcam photogrammetry has its advantages, it is also a very labor intensive and computationally demanding process compared to LiDAR or structured light. Additionally, visual inspection showed that point cloud surfaces from the LifeCam were generally not as smooth as those from cameras with better lenses such as the Panasonic FZ28 and the D3100 SLR. Another important finding from this study is that the low resolution QuickCam webcam was able to produce a point cloud for the office (but not the lobby) in under five minutes, but its resulting point cloud would have limited use due to excessive noise and poor accuracy errors of 7% compared to LiDAR.

The miniaturization and commoditization of LiDAR may make LiDAR a more suitable technology for taking indoor measurements in the near future; however, SfM-MVS has certain strengths that can complement LiDAR. This study demonstrated the feasibility of SfM-MVS as a technique that can supplement LiDAR data in special use cases.

Part III

Feature extraction

Chapter 5

A review of segmentation techniques

5.1 Overview

Point clouds made through 3D remote sensing will often produce impressive visual renderings of the physical environment, but each individual point contains little useful information other than coordinate values and possibly a few other quantifiable attributes. Producing a 3D map requires using the point cloud to provide geometric and semantic information about the physical environment through a process known as *segmentation*. Many of the concepts used in the segmentation of 2D images also apply to 3D point clouds (Héno and Chandelier 2014; Gonzalez and Woods 2002). For indoor mapping, segmentation divides a point cloud into regions representing entities or subcomponent parts — e.g., walls, floors, and furniture — which can later serve as the source material for model reconstruction.

For example, Figure 5.1 shows a point cloud for part of a room inside a museum. Each point has an identification number, coordinates, and color information, which provide no insight into the physical make-up of the room. When examined collectively, though, the points reveal walls, a floor, and individual furnishings, with the points segmented or

grouped based on their context; a closer look also shows voids and errors in the data (i.e., noise). The goal of segmentation involves finding algorithms that can correctly classify these points.

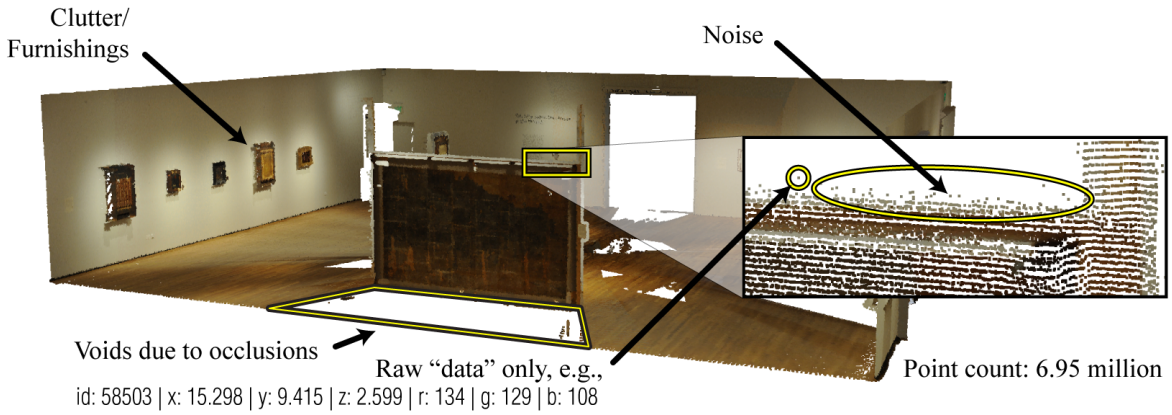


Figure 5.1: Some limitations of raw point cloud data for indoor mapping

5.2 Approaches to segmentation

Héno and Chandelier (2014) divided point cloud segmentation methods into geometric and semantic approaches, as shown in Figure 5.2. Geometric approaches attempt to group points based on geometric shapes, such as planes and cylinders, while semantic approaches seek to extract abstract entities, such as rooms and furniture. Earlier studies on indoor feature extraction focused almost exclusively on simple geometric and semantic segmentation with the goals of identifying primitive indoor structural elements (e.g., wall, floors, ceilings, windows, and doors) from point clouds, with use cases growing increasingly complex to include incomplete point clouds and occlusions. While these structural elements continue to present challenges, especially in non-Manhattan world

scenes, more recent studies have broadened the domain to include smaller elements such as furnishings and larger elements such as entire buildings with multiple rooms.

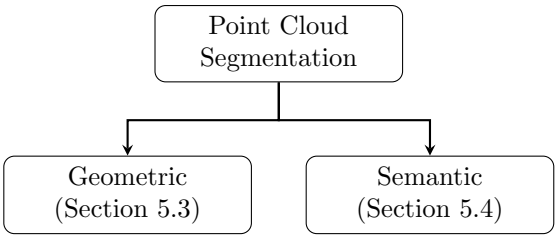


Figure 5.2: Héno and Chandelier (2014) model for point cloud segmentation

5.3 Geometric segmentation

Geometric segmentation algorithms generally partition a point cloud based on either discontinuities or similarities between groups of points (Gonzalez and Woods 2002). In image processing, the former often partitions an image based on distinct edges that show abrupt changes in pixel values while the latter looks for regions with similar pixel values. For these reasons, Gonzalez and Woods (2002) and Héno and Chandelier (2014) classified these as "edge based" and "region based" approaches. Gonzalez and Woods called a third classification method *thresholding*, which Héno and Chandelier called *clustering*, that looks for spatial clusters of similar values, often implemented through statistical analysis. A fourth method uses *shape fitting* to find the best fit for a geometric shape to a point cloud. These four approaches can also be combined to leverage the strengths of each individual approach.

5.3.1 Edge based methods

In 2D image processing, edge based methods detect edge pixels, or edgels, in an image that are then linked together to form the edges of partitioned objects. These methods

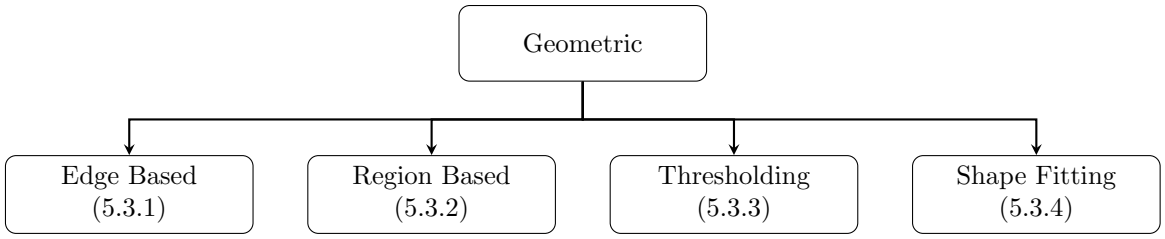


Figure 5.3: Héno and Chandelier (2014) model for point cloud segmentation

closely approximate how humans segment images and work well with images that have relatively high levels of contrast (Solberg 2016). However, they tend to struggle with images that have smooth color gradients or low contrast, are sensitive to noise, and face challenges in forming robust edges.

Unlike the pixels of a 2D image that combine to form edges, the points of a 3D point cloud represent surfaces that can be viewed from multiple perspectives, making edge based methods more complicated in a 3D setting. Other factors complicating use of edge based methods include the noise and uneven distributions of points in a point cloud (Nguyen and Le 2013). The closest application of edge based methods for point clouds appears to be in the area of photogrammetry, specifically in RGB-D (see Section 2.2.2) or range images, where the source data for the point clouds come from 2D images (Nguyen and Le 2013; Choi, Trevor, and Christensen 2013). Otherwise, most approaches for segmenting indoor spaces use either the region based approach or thresholding.

5.3.2 Region based methods

Region based methods work directly on point clouds to find segments with similar predefined properties. These can take a bottom-up approach starting with a single point that grows into a region (i.e., the region growing method) or a top-down approach starting with the entire point cloud data set that is divided and merged until reaching an end state (i.e., the splitting and merging method).

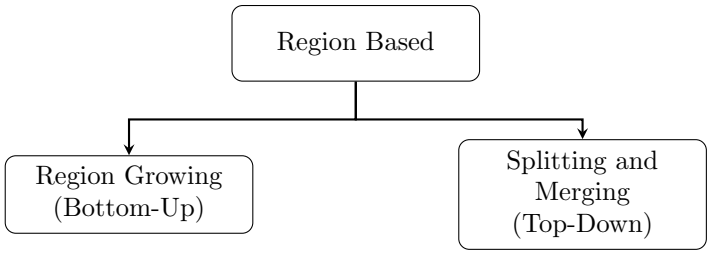


Figure 5.4: Héno and Chandelier (2014) model for point cloud segmentation

Region growing

Segmentation that uses region growing starts with a point and appends neighboring points with similar properties — such as the direction of the normal vector — to form a region of similar points (Nguyen and Le 2013; Solberg 2016). Region growing does not always produce consistent results since the final regions depend on selection of the initial seeds. In situations with no given seed points, the measured properties are computed for every point in an image; if this results in clusters of points, the centroids of these clusters can act as the seed points (Gonzalez and Woods 2002). Prior indoor mapping work that used region growing usually compared a point's estimated surface normal vector to that of the seed point. These studies include Xiong and Huber (2010), Sanchez and Zakhor (2012), Turner and Zakhor (2012), Stambler and Huber (2014), and Turner, Cheng, and Zakhor (2015).

Splitting and merging

Splitting and merging repeatedly splits a point cloud using the octree structure until reaching some criteria for all four of the smallest leaf nodes, which are then compared with neighboring regions and merged if they have the same value (Gonzalez and Woods 2002; Solberg 2016). This process is repeated until no further merging or splitting can be made.

5.3.3 Thresholding

Applied to image processing, thresholding represents a method that discretizes an image based on natural breaks — i.e., the thresholds — in the data, which can then be used to group together similar values (Gonzalez and Woods 2002). Thresholding provides a simple approach to segmentation as long as the number of partitions remains low.

For indoor point clouds, thresholding stands out as a favored approach due to its simplicity and effectiveness, since any type of analysis that can generate discernible breaks in the data can use thresholding. A rudimentary form uses histograms of *coordinate values* to detect planar surfaces, such as floors, ceilings, and walls. Budroni and Boehm (2009) swept a vertical plane both laterally and rotationally through a point cloud to detect peaks in the histogram of points intersecting the plane; these peaks corresponded to surfaces. In their highly cited work, Okorn et al. (2010) used the histogram approach for detecting floors and ceilings but used two variations of the Hough transform for detecting irregularly shaped walls. Khoshelham and Díaz-Vilariño (2014) used an approach for modeling multiple rooms that relied on the Manhattan world assumption and axis-aligned point clouds for placing and merge cuboids based on histogram peaks.

A simplified form of Khoshelham and Díaz-Vilariño (2014) technique was used in this dissertation that involved taking histograms of points projected onto axis-aligned orthogonal planes for detecting the floor, ceiling, and outer walls of a room. This approach first isolates two of the three coordinate axes to produce flattened 2D projections of the point cloud (Sinton 1978). It then plots the number of points along one axis within bins distributed throughout the other axis. This plot reveals the 1D coordinates of walls, floors, or ceilings as spikes in the histogram, resulting in six coordinates defining the corner points of the room.

5.3.4 Shape fitting

Shape fitting involves attempting to fit primitive shapes, such as planes, to the data often using the random sample consensus (RANSAC) approach and classifying the nearest points using that shape. Fischler and Bolles (1981) used RANSAC to fit 2D shapes to images while Bolles and Fischler (1981) used to to fit cylinders to point clouds. Schnabel, Wahl, and Klein (2007) used this approach to segment architectural features and Ochmann et al. (2014), Jung et al. (2014), and Kim, Kang, and Lee (2014) extended it to indoor modeling.

5.3.5 Other methods

Other segmentation methods exist in addition to the four mentioned in this chapter. For example, Adán and Huber (2011) supplemented room-based thresholding with radio frequency identification (RFID) tags to distinguish clutter (i.e., furniture) from structure. In another instance, Valero, Adán, and Cerrada (2012) and Valero, Adán, and Bosché (2016) used projected 2D binary images instead of histograms followed by the Hough transform to identify a room's perimeter.

5.4 Semantic segmentation

Semantic segmentation gives meaning to the groups of points identified through geometric segmentation. For instance, points representing vertical planar surfaces can be differentiated as walls versus the horizontal surfaces of floors and ceilings. Not surprisingly, progress in indoor semantic segmentation has developed alongside geometric segmentation. Earlier studies involved the semantic identification of primitive structural elements — such as walls, floors, ceilings, doors, and windows — with the remainder

grouped under the category of clutter. More recent studies have expanded the spatial scope of semantics in both directions, with one looking at the macro-view of buildings, rooms, and corridors while the other looks at the micro-view of segmenting what was once formerly considered clutter, i.e., furniture, etc.

5.4.1 Structure versus clutter

Semantic segmentation developed alongside work in geometric segmentation, with early studies seeking to identify the main elements of rooms, i.e., walls, floors, and ceilings (Schnabel et al. 2007; Budroni and Boehm 2009; Xiong and Huber 2010). Later studies expanded structural segmentation by adding methods for detecting openings (e.g., doors and windows) and voids due to occlusions. Adán and Huber (2010) used the method proposed by Okorn et al. (2010) to identify wall surfaces, applied edge segmentation to 2D range images to identify candidate openings, and classified openings as either occupied, empty, or occluded based on ray-tracing analysis.

5.4.2 Furniture segmentation

Further refinement of indoor point cloud segmentation sought to classify the elements of clutter, i.e., furniture. For example, Valero, Adán, and Cerrada (2012) and Adán and Valero (2014) used an occupied-unoccupied binarized projection of the point cloud to segment a room's walls and then used RFID labels attached to non-structural point cloud groupings to identify furniture.

5.4.3 Room segmentation

Later developments in indoor point cloud segmentation also looked at the macro-view, i.e., automatically segmenting parts of a building as a prerequisite for room-based

segmentation. Ochmann et al. (2014) took a multi-room point cloud and used RANSAC as proposed by Schnabel et al. (2007) for initially segmenting the structural components of each room; they then determined room segments using an empirically derived visibility function. Jung, Stachniss, and Kim (2017) took a different approach by first applying binary map segmentation to a single floor, multi-room point cloud, which left an outline of all the walls with openings for doorways. They then implemented an opening-closure routine to close the intra-room pathways resulting in the segmentation of the entire floor into rooms and a corridor. Armeni et al. (2016) used perhaps the simplest method for room segmentation. They first used the histogram thresholding method to segment all walls within an axis-aligned point cloud of an entire floor of a building. Since walls have a certain thickness, they then looked for consecutive peaks (i.e., walls) in the histogram separated by a certain distance to determine room segments.

5.5 Conclusion

This chapter provided a high level overview of the state of research on the segmentation of point clouds of indoor spaces. It covered the four general approaches to geometric segmentation (edge based, region based, thresholding, and shape fitting) and considerations for the semantic segmentation of the insides of buildings. While most research to date have looked at segmentation at the room level and continue to refine those methods, more recent research has branched out in two opposite directions at the micro- and macro-levels. The micro-level looks at segmenting clutter, which consists of all those elements previously ignored for finding the shape of a room; these consist of such things as furniture and equipment. The macro-level goes in the opposite direction and looks at segmentation at the building level, with an emphasis on identifying individual rooms and corridors. Together, these two emerging areas will improve the quality and capabilities

of 3D indoor mapping by facilitating automation and providing the basis for multiple levels of detail of indoor space.

Chapter 6

Segmentation and modeling via thresholding

6.1 Introduction

A simple and effective approach to segmenting point clouds of rectangular rooms involves the use of thresholding with histograms, where coordinates for bins with the highest counts represent the coordinates of boundary surfaces. This approach assumes a Manhattan world scene with rooms consisting only of planar and orthogonal surfaces and requires that these yet-to-be segmented "surfaces" be aligned with the Cartesian coordinate axes. Budroni and Boehm (2009), Okorn et al. (2010), and Huber et al. (2010) provided early examples of using different forms of thresholding to segment point clouds of rooms, with several follow-on studies either applying or refining these method. In particular, Khoshelham and Díaz-Vilariño (2014) and Díaz-Vilariño et al. (2015) addressed the problem of axis alignment by using directional statistics applied to an extended Gaussian image (EGI).

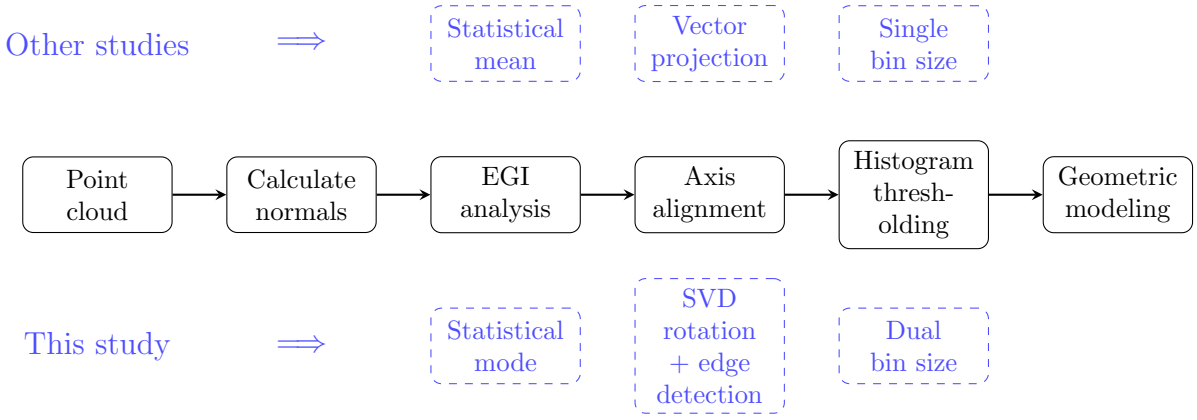


Figure 6.1: A point cloud segmentation and modeling process for a rectangular room. Top row shows methods used by Khoshelham and Díaz-Vilariño (2014) and Díaz-Vilariño et al. (2015) and bottom row shows methods used in this study.

This study provides refinements in both the EGI and histogram thresholding approaches proposed by Khoshelham and Díaz-Vilariño (2014) and Díaz-Vilariño et al. (2015) as summarized in Figure 6.1. In their use of the EGI, Díaz-Vilariño et al. used the mean values of the EGI clusters to deduce the three orthogonal axes of the point cloud. However, the mean value has a high sensitivity to outliers which makes this approach less than optimal. This study uses the robust statistical mode for approximating the point cloud axes, as will be discussed later. Furthermore, Díaz-Vilariño et al. used direct vector projections, instead of pure rotations, to re-orient the point cloud, which can lead to imperceptible distortions in the original data. To ensure pure rotations, this study used singular value decomposition followed by an edge-based refinement step. For floor, ceiling, and wall detection, Khoshelham and Díaz-Vilariño used a fixed histogram bin size but smaller bin sizes require longer times to compute. An alternative method proposed in this study uses a two-step approach with a general search using large bins followed by small bins to pinpoint locations.

6.2 Methodology

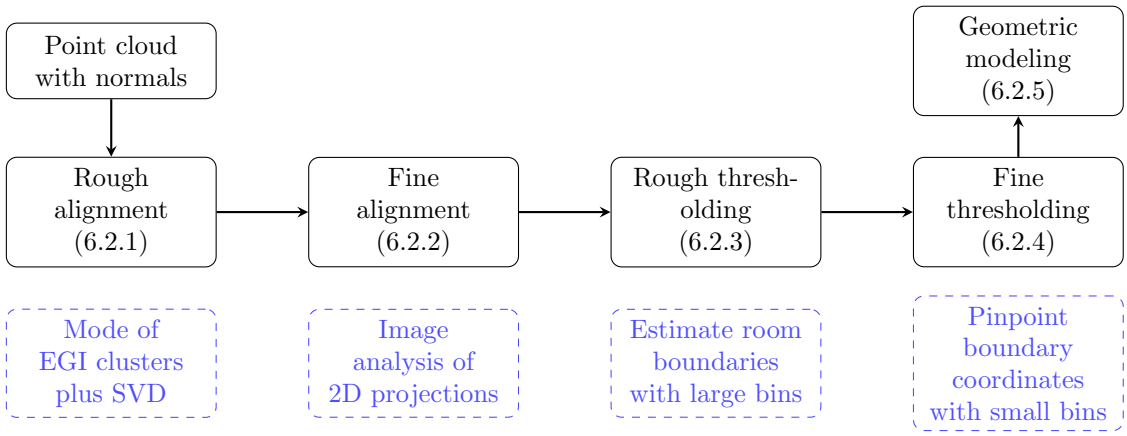


Figure 6.2: Point cloud segmentation and modeling process used in this study

The overall segmentation process used in this study has five phases after calculating normals. These consist of rough axis alignment, fine axis alignment, rough thresholding, fine thresholding, and geometric modeling as illustrated in Figure 6.2. The rough alignment phase exploits properties of the EGI as demonstrated by Khoshelham and Díaz-Vilariño and Díaz-Vilariño et al. However, instead of using the mean, this study uses the mode — a more accurate and reliable predictor of central location for this particular application. The second phase uses an image-based alignment process to refine the initial EGI alignment; in practice, it serves only as a safeguard since the first phase usually produces very accurate results. With the point cloud axis-aligned, the third and fourth phases use histogram thresholding to search for and refine the locations of floor, ceiling, and walls in the point cloud. Finally, the six coordinates derived from thresholding are used to create the simple 3D polyhedron of the room.

For illustration purposes, this study uses the LiDAR point cloud of a room in the UCSB Art, Design, and Architecture Museum to demonstrate each step of the process.

This point cloud has been intentionally rotated off-axis to make the effects of the alignment process more obvious as well as to demonstrate the robustness of the EGI approach.

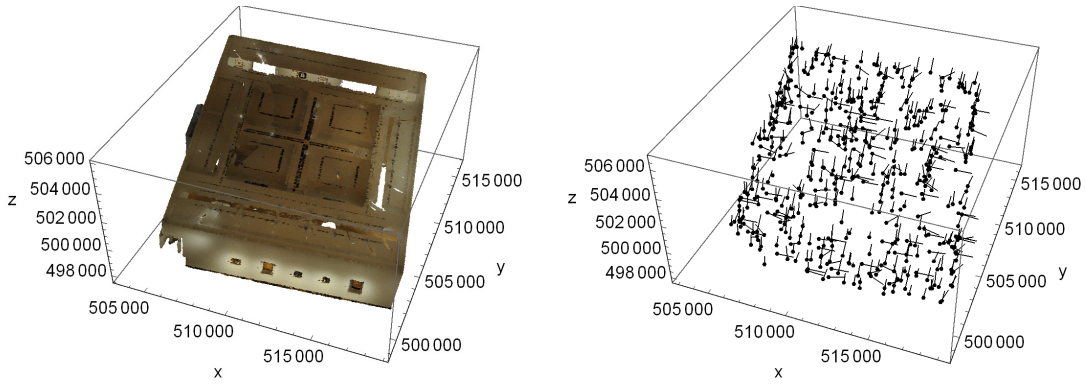
6.2.1 Phase 1: Rough axis alignment

Rough axis alignment works by estimating the orientation of a rectangular room and then finding the rotation matrix required to align its quasi-orthogonal axes with the xyz axes. While simple in concept, the execution is far from trivial. This process involves calculating the normals for every point in the point cloud, plotting the unit normals in a extended Gaussian image (EGI), clustering the EGI points into three groups, finding the vectors that point toward the "center" of each cluster, finding the rotation matrix to align these vectors with the xyz axes, and applying this rotation matrix to the point cloud. Translation of the point cloud's centroid to the origin and optional scaling precedes this phase.

Point cloud normals

Normal calculations estimate the direction of a vector that runs orthogonal relative to the k nearest neighbors (knn) of a point for each point in a point cloud, in which the knn provides an estimate of the modeled surface. These vectors require post-processing to correctly orient their directions, since a normal vector can arbitrarily point in one of two opposing directions. In many applications, normal vectors are oriented to emanate away from a point cloud's centroid; for a rectangular point cloud, this results in six principal directions. For axis alignment, however, this study requires three and only three principal directions made possible by orienting the normals away from a false origin point located in the $-x, -y$, and $-z$ directions relative to the point cloud. For a rectangular point cloud, this results in three quasi-orthogonal principal directions, which makes it possible

to calculate the rotation required to align these directions with the xyz axes. Figure 6.3 shows the sample point cloud and a random sampling of its oriented normal vectors.



(a) Sample point cloud (b) Oriented normals

Figure 6.3: Sample point cloud and random sample of oriented surface normals

Extended Gaussian image

When plotted with a common origin, unit normal vectors form a distinct spherical pattern called an extended gaussian image (EGI). Patterns in the EGI can serve as a unique "fingerprint" for a scanned object, but more importantly, they can also provide clues on the point cloud's orientation (Ikeuchi 1981; Horn 1984; Brou 1984). For a perfectly rectangular cuboid with normal vectors oriented away from a false origin, the EGI will show three points corresponding to the six orthogonal planar surfaces. When used with empirical data, furniture, non-orthogonal room partitions, and errors in estimated normal vectors will introduce noise to the EGI, as illustrated in Figure 6.4, but the six principal planes should still produce three distinct clusters of points.

EGI clusters

Finding the three vectors corresponding to the rectangular point cloud's axes involves finding the locations where the normal vectors cluster together in the EGI — illustrated

in Figure 6.4 — and then finding the "central" location within each cluster. These three central locations point in the directions of the point cloud's quasi-orthogonal axes[1] that will be used to find the rotations for matching the xyz axes.

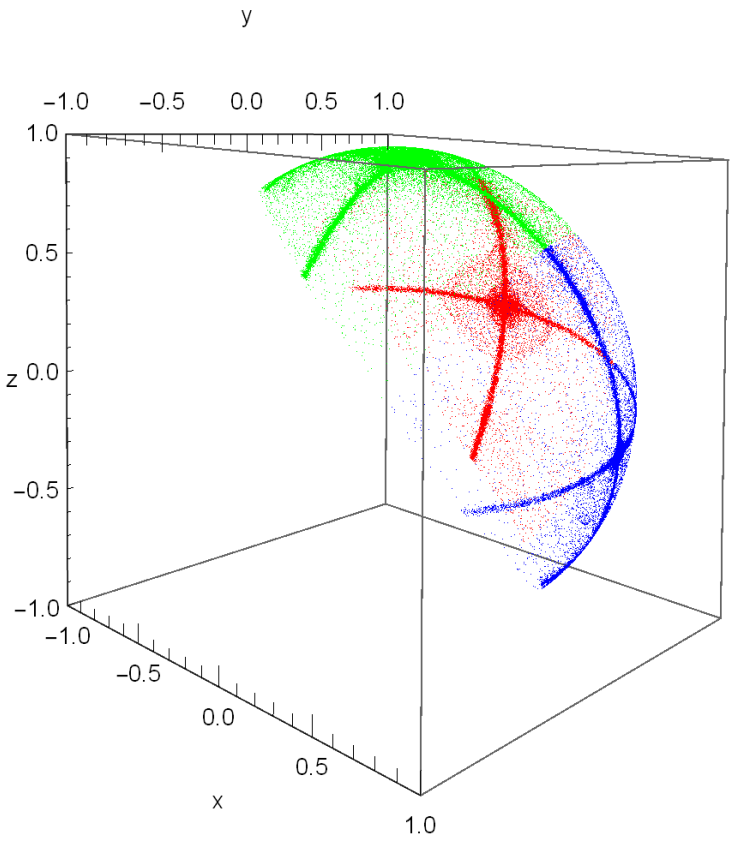


Figure 6.4: Extended gaussian image of sample data clustered into three groups

Mean and median not ideal Many different approaches exist for estimating the central location of each EGI cluster. The simplest and most straightforward involves calculating either the mean or median of each cluster in the original Cartesian coordinate system. Díaz-Vilariño et al. (2015) used the k-means in their study; when using the mean, the mean radius provides a measure of dispersion (Leong and Carlile 1998). Compared to

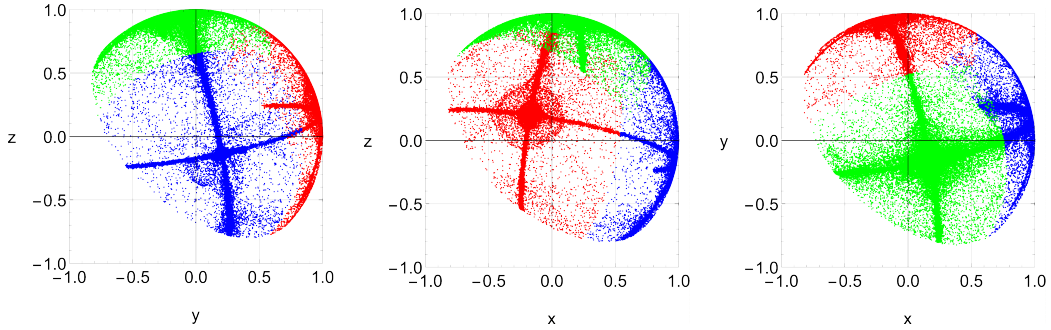
[1]The three empirically derived vectors will usually be slightly off-angle from perfectly orthogonal axes.

the mean for this specific use case, the median will often provide a more accurate measure of centrality due to the mean's sensitivity to outliers. However, neither approach provides the optimal solution since they take into account irrelevant data. Since this process is only concerned with the principal surfaces (i.e., floor, ceiling, and walls), normal vectors for all other non-principal surfaces — as well as erroneous vectors — provide no informational value and can be disregarded.

Mode as the central location Since the floor, ceiling, and walls will often have the highest number of points in a point cloud — thus, the highest number of normal vectors pointing in only one of three principal directions — the most reliable approach involves using the mode, which requires performing statistical analysis on a spherical surface. A classical approach to this problem involves discretizing the sphere using some form of tessellation, e.g., triangles or dodecahedra, and selecting the three polyhedra with the highest number of points, with their central coordinates representing the end points of the three principal vectors (Ikeuchi 1981; Horn 1984; Brou 1984). However, this approach can be computationally burdensome when working with tens of thousands to millions of points.

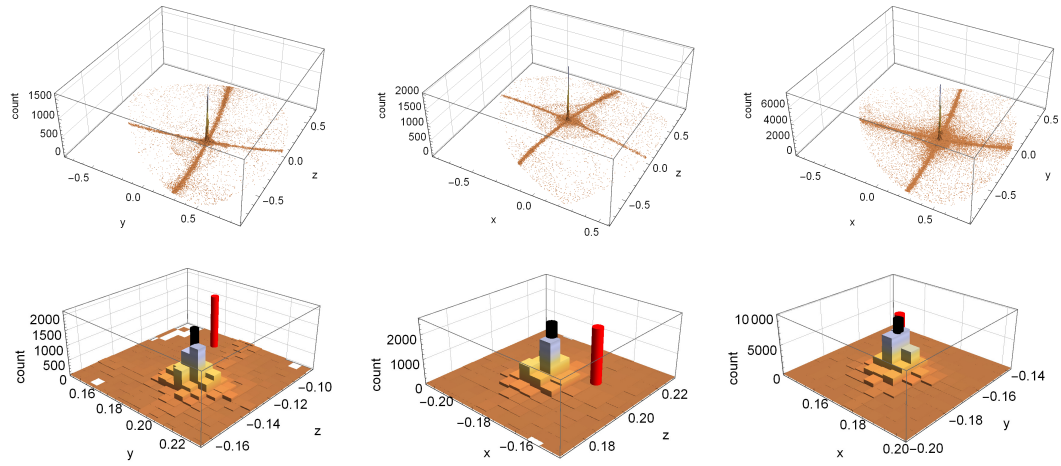
This study uses a cartographic approach that projects the EGI clusters onto their respective orthogonal planes — i.e., yz plane for the x-axis, etc. — and applies histogram analysis to the resulting 2D points to find the location of each cluster's mode, where the bin with the highest number of points represents the mode. Since this first phase seeks only a rough alignment, errors from using an *orthographic projection* should not adversely impact the final results.

For the sample data, finding the mode for each of the three clusters, illustrated in Figure 6.6, produced the unit normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ shown below. Note how the mean (red tube) provides a poor estimate of the central location in Figure 6.6; even



(a) View along x-axis (b) View along y-axis (c) View along z-axis

Figure 6.5: Orthographic projections of the EGI



(a) x-axis location (b) y-axis location (c) z-axis location

Figure 6.6: Histograms of the EGI clusters. Red tubes represent the mean values and black tubes the median values.

though the median (black tube) provides a better estimate, it still deviates from the optimal mode value in certain situations.

$$\mathbf{n}_1 = \begin{bmatrix} 0.968523 \\ 0.2075 \\ -0.1375 \end{bmatrix}; \quad \mathbf{n}_2 = \begin{bmatrix} -0.1725 \\ 0.965006 \\ 0.1975 \end{bmatrix}; \quad \mathbf{n}_3 = \begin{bmatrix} 0.1725 \\ -0.1675 \\ 0.970663 \end{bmatrix}$$

First rotation matrix, $\mathbf{R}_1$

The conventional rotation matrix $\mathbf{r}_c$ for rotating the xyz vectors to match the *perfectly* orthogonal point cloud axes vectors $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ consists of $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ arranged in row form, with each row representing a simple projection of xyz coordinates onto $\mathbf{n}_i$ (Hearn, Baker, and Carithers 2010). Inverting $\mathbf{r}_c$ by taking its transpose, i.e., $\mathbf{r}_i^\top$, does the reverse by projecting point cloud coordinates on to the xyz axes.

$$\mathbf{n}_1 = \begin{bmatrix} n_{1,x} \\ n_{1,y} \\ n_{1,z} \end{bmatrix}; \quad \mathbf{n}_2 = \begin{bmatrix} n_{2,x} \\ n_{2,y} \\ n_{2,z} \end{bmatrix}; \quad \mathbf{n}_3 = \begin{bmatrix} n_{3,x} \\ n_{3,y} \\ n_{3,z} \end{bmatrix} \tag{6.1}$$

$$\mathbf{r}_c = \begin{bmatrix} n_{1,x} & n_{1,y} & n_{1,z} \\ n_{2,x} & n_{2,y} & n_{2,z} \\ n_{3,x} & n_{3,y} & n_{3,z} \end{bmatrix} \tag{6.2}$$

However, the empirically derived principal vectors may not be perfectly orthogonal, so the direct use of Equation 6.2 will result in a distorted point cloud. Instead, the derived vectors $\mathbf{n}_i$ should be used to somehow find a pure rotation that best fits those vectors with the xyz axes. A test for invertibility can show whether a matrix represents a *pure* rotation matrix ($\mathbf{r}^\top \cdot \mathbf{r} = \mathbf{I}$) or contains other undesirable transformations ($\mathbf{r}^\top \cdot \mathbf{r} \neq \mathbf{I}$). For the sample data, $\mathbf{r}_c^\top \cdot \mathbf{r}_c \neq \mathbf{I}$, so another approach is required to find the rotation matrix.

$$\mathbf{r}_c = \begin{bmatrix} 0.968523 & 0.2075 & -0.1375 \\ -0.1725 & 0.965006 & 0.1975 \\ 0.1725 & -0.1675 & 0.970663 \end{bmatrix} \quad \mathbf{r}_c^\top = \begin{bmatrix} 0.968523 & -0.1725 & 0.1725 \\ 0.2075 & 0.965006 & -0.1675 \\ -0.1375 & 0.1975 & 0.970663 \end{bmatrix}$$

$$\mathbf{r}_c^\top \cdot \mathbf{r}_c = \begin{bmatrix} 0.99755 & 0.00561123 & 0.00019873 \\ 0.00561123 & 1.00235 & -0.000528596 \\ 0.00019873 & -0.000528596 & 1.0001 \end{bmatrix}$$

Finding $\mathbf{R}_1$ using singular value decomposition Least squares adjustment through singular value decomposition (SVD) provides one solution for optimally aligning the imperfect empirically-derived axes with the xyz axes. SVD minimizes the least squares distances between matching pairs of points while simultaneously determining the rotation matrix (Umeyama 1991). By treating vector coordinates as points, $\mathbf{P}$ can represent the *source* end points of the point cloud axes and $\mathbf{Q}$ can represent the end points of the target xyz axes as shown in Equation 6.3. Here, the goal is to find the rotation from $\mathbf{P}$ to $\mathbf{Q}$.

$$\mathbf{P} = \begin{bmatrix} n_{1,x} & n_{1,y} & n_{1,z} \\ n_{2,x} & n_{2,y} & n_{2,z} \\ n_{3,x} & n_{3,y} & n_{3,z} \end{bmatrix}; \quad \mathbf{Q} = \begin{bmatrix} x_x & x_y & x_z \\ y_x & y_y & y_z \\ z_x & z_y & z_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

The first step involves calculating the cross-covariance matrix $\mathbf{A}$ using Equation 6.4.

$$\mathbf{A} = \mathbf{Q}^\top \cdot \mathbf{P} \tag{6.4}$$

Next, an SVD operation — widely available in many math processing software packages — is performed to decompose $\mathbf{A}$ into three matrices as shown in Equation 6.5, where $\mathbf{D}$ can be interpreted as a scaling or skewing matrix.

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^{\top} \tag{6.5}$$

Since the transformation involves only rotations, an identity-like matrix $\mathbf{S}$, as defined in Equation 6.9, replaces matrix $\mathbf{D}$ to isolate the operation to rotations only. Use of $\mathbf{S}_{SVD}$ also ensures that the rotation preserves the right-hand rule for the coordinate system, where a value of $d = 1$ signifies a right-hand system.

$$\mathbf{S}_{SVD} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} \quad \text{where} \quad d = \det(\mathbf{U}) \cdot \det(\mathbf{V}) \tag{6.6}$$

The initial 3x3 rotation matrix $\mathbf{r}_1$ takes the form shown in Equation 6.7, with $\mathbf{R}_1$ representing the 4x4 homogeneous form.

$$\mathbf{r}_1 = \mathbf{U} \cdot \mathbf{S}_{SVD} \cdot \mathbf{V}^{\top} \tag{6.7}$$

$$\mathbf{R}_1 = \left[\begin{array}{ccc|c} & & & 0 \\ & \mathbf{r}_1 & & 0 \\ & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{6.8}$$

For the sample data, $\mathbf{r}_1$ passes the invertibility test showing that it is a pure rotation matrix.

$$\mathbf{r}_1 = \begin{bmatrix} 0.969155 & 0.204505 & -0.137536 \\ -0.175441 & 0.964422 & 0.197763 \\ 0.173086 & -0.167533 & 0.970553 \end{bmatrix}$$

$$\mathbf{r}_1^\top \cdot \mathbf{r}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Applying the first transformation

Solving for $\mathbf{R}_1$ used the origin-centered EGI, which is invariant to the point cloud's actual location. However, applying $\mathbf{R}_1$ to the point cloud requires first translating the point cloud's centroid to the origin using $\mathbf{T}_1$ before applying the rotation. This study also uses a unit conversion scaling matrix $\mathbf{S}$ to convert all units to meters to allow use of a single histogram analysis algorithm.

$$\mathbf{S} = \begin{bmatrix} scale & 0 & 0 & 1 \\ 0 & scale & 0 & 1 \\ 0 & 0 & scale & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.9}$$

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & x_{centroid} \\ 0 & 1 & 0 & y_{centroid} \\ 0 & 0 & 1 & z_{centroid} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.10}$$

Combining $\mathbf{T}_1$, $\mathbf{S}$, and $\mathbf{R}_1$ produces the consolidated transformation matrix $\mathbf{M}_1$ for initial alignment (Equation 6.11). Note that $\mathbf{M}_1$ leaves the point cloud at the origin $(0, 0, 0)$ for follow-on operations.

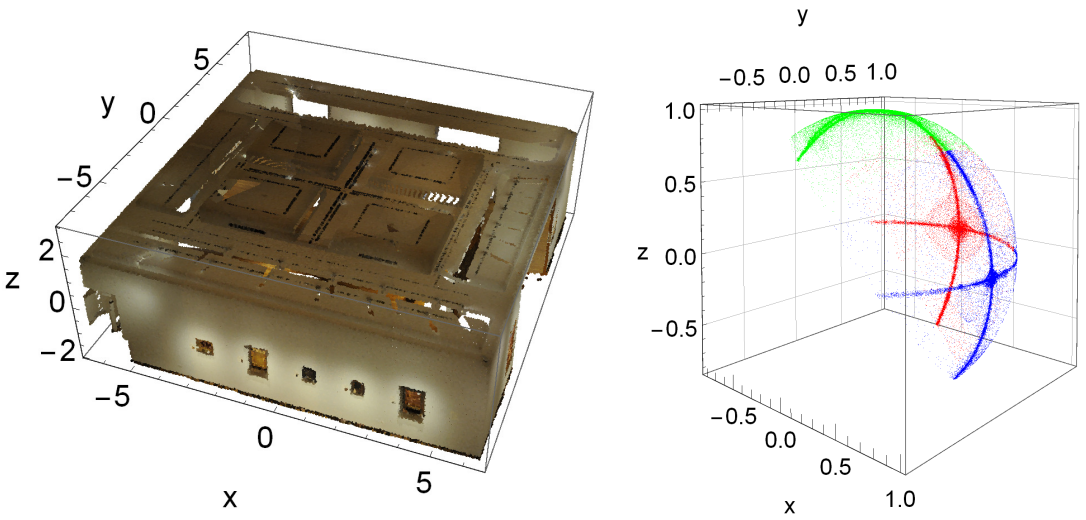
$$\mathbf{M}_1 = \mathbf{S} \cdot \mathbf{R}_1 \cdot \mathbf{T}_1 \tag{6.11}$$

For the sample data, $\mathbf{R}_1$, $\mathbf{S}$, $\mathbf{T}_1$, and $\mathbf{M}_1$ have the values shown below, with Figures 6.7, 6.8, and 6.9 showing visualizations of initial alignment. Note that the sample data used units of millimeters requiring a conversion factor of 0.001 to meters.

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & -511125.28058904150854 \\ 0 & 1 & 0 & -507744.26572262365594 \\ 0 & 0 & 1 & -501798.16323047575226 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad \mathbf{S} = \begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.001 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

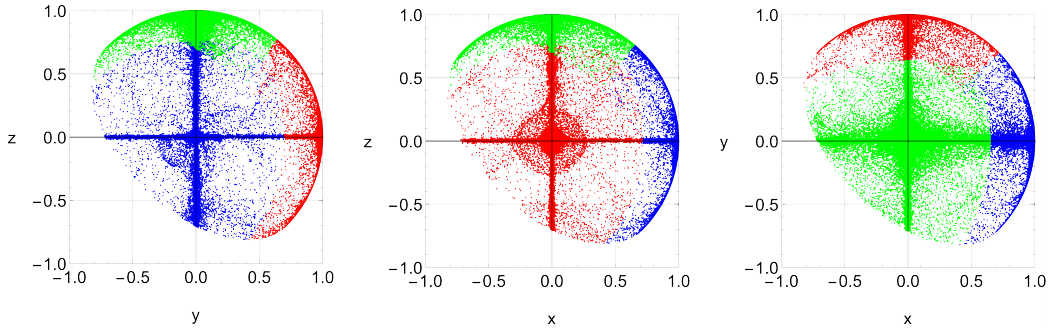
$$\mathbf{R}_1 = \begin{bmatrix} 0.969155 & 0.204505 & -0.137536 & 0 \\ -0.175441 & 0.964422 & 0.197763 & 0 \\ 0.173086 & -0.167533 & 0.970553 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}_1 = \begin{bmatrix} 0.000969155 & 0.000204505 & -0.000137536 & -530.181 \\ -0.000175441 & 0.000964422 & 0.000197763 & -499.244 \\ 0.000173086 & -0.000167533 & 0.000970553 & -490.427 \\ 0. & 0. & 0. & 1 \end{bmatrix}$$



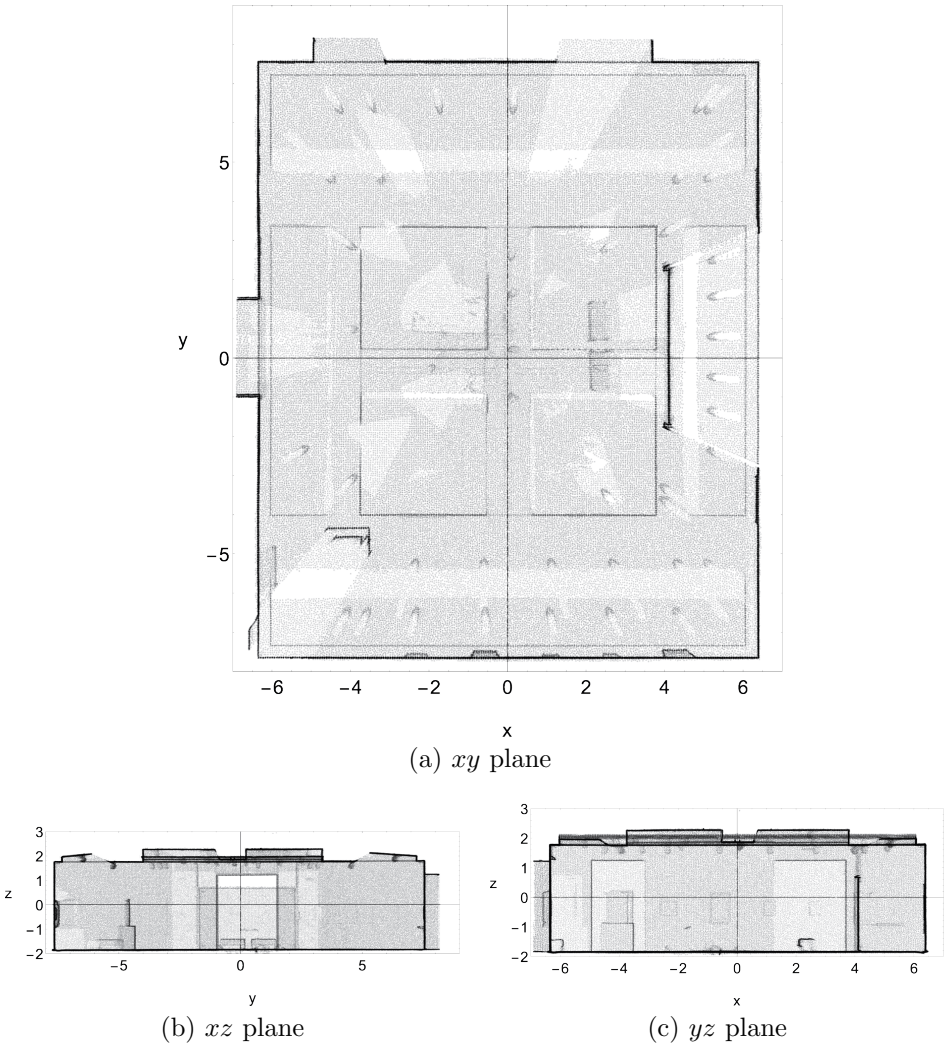
(a) Point cloud (b) EGI

Figure 6.7: Point cloud and EGI after rough alignment



(a) View along x-axis (b) View along y-axis (c) View along z-axis

Figure 6.8: Orthographic projections of the EGI after rough alignment



(a) xy plane

(b) xz plane

(c) yz plane

Figure 6.9: Orthographic views of point cloud after rough alignment

6.2.2 Phase 2: Fine axis alignment

The second phase fine tunes the initial axis alignment by using image processing routines applied to 2D projected images of the point cloud. A common rasterization approach uses occupancy grids, where cells of a certain size are treated as either occupied or unoccupied, and are often used in areas such as data reduction (Valero, Adán, and Bosché 2016; Oesau 2015; Xiong et al. 2013), boundary tracing (Hong et al. 2015; Jung et al. 2014), feature detection (Ochmann et al. 2016), and voxel carving (Turner 2015).

This study uses a simpler approach by directly projecting orthographic views of the point cloud as 2D images and exploiting the phenomenon where higher numbers of overlapping semi-opaque points will result in darker lines that make structural features more salient. Line detection algorithms can then be used to find the orientations of the floor, ceiling, and walls. While this approach remains a rasterization process at its very core, it avoids the need for computationally burdensome grid discretization and bin counting routines; the only two parameters consist of the opacity level and point size. Since the process adjusts rotations one axis at a time, it involves a sequence of three rotations along the z, y, and x axes, respectively. The component rotation matrices $\mathbf{R}_{2z}$, $\mathbf{R}_{2z}$, and $\mathbf{R}_{2z}$ comprise the second and final set of rotations.

Project image onto orthogonal planes and detect lines

Creating 2D images of the point cloud requires setting up three orthographic views by fixing coordinate values for one axis at a time and rendering images of the plane-projected points using software. The opacity value controls the salience of the linear features; too high a value would result in a saturated image while too low a value a washed out image, both of which would make line detection difficult. Figure 6.9 shows the three projected images for the sample data using a 5% opacity value.

The three projected images are then processed through a gradient filter to highlight areas of sharp color changes that usually correspond to the darker lines of structural features, as illustrated in Figure 6.10. Geometric lines are then extracted by applying the Hough transform to the gradient filtered images, as illustrated in Figure 6.11. Note that the 3D point cloud and 2D projected images use two different coordinate reference systems, with the former using a world-based system and the latter an image-based system; this has no impact on calculation of rotation angles.

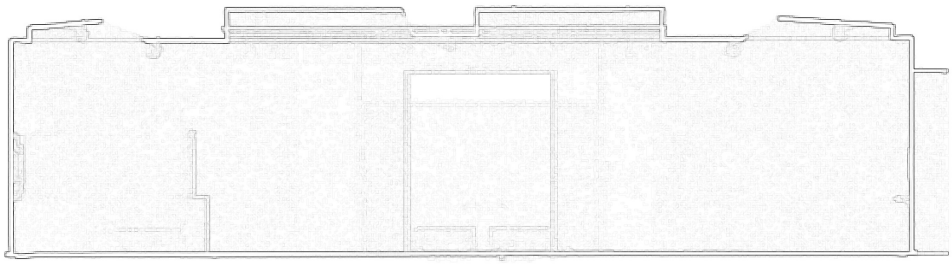


Figure 6.10: Point cloud image processed through a gradient filter

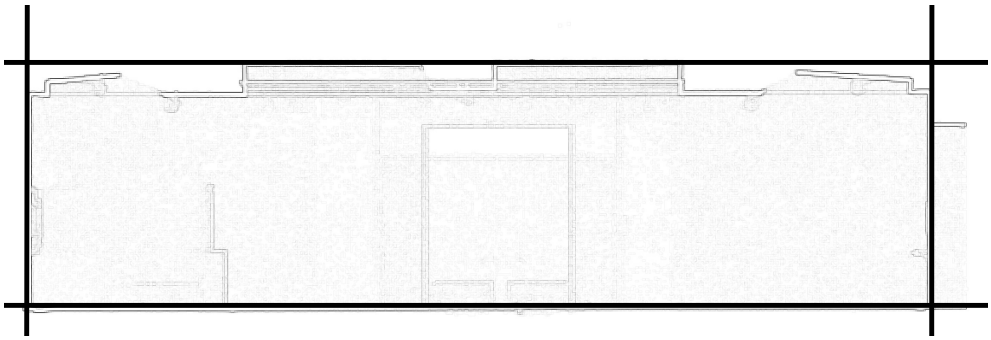


Figure 6.11: Lines detected using the Hough transform

Calculate rotation angles based on line slopes

This study assumed that all floors have a horizontal orientation, all walls have a vertical orientation, and ceilings may have irregular shapes or arbitrary orientations. As a result, only the floor was used to measure rotational deviation from the horizontal plane about the x and y axes while the mean value of up to four walls[2] was used about the z axis.

Performing the transformation operation

Figure 6.12 provides an overview of the process for fine rotational alignment. While applying this process to the sample data resulted in an identity matrix — indicating that

[2]This depends on the success of line detection.

the initial rough alignment produced accurate results — it still serves a useful safeguard against inaccurate rough alignments, e.g., those produced with very noisy data.

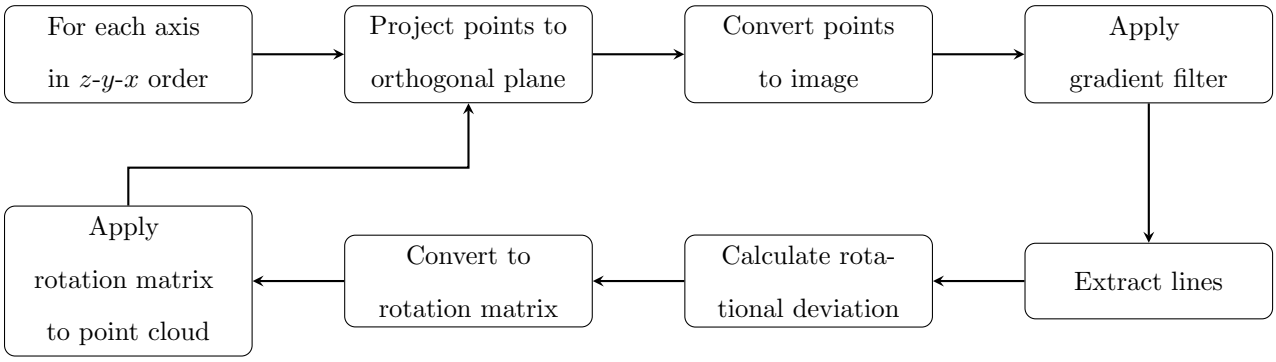


Figure 6.12: Coarse Registration.

6.2.3 Phase 3: Initial thresholding

This study used a basic approach to thresholding based on a method used by Khoshelham and Díaz-Vilariño (2014) with two changes. First, it adds a two-step progressive search for the thresholds that consists of an initial low resolution histogram to quickly identify the general vicinity of floor, ceiling, or walls, and a high resolution histogram to pinpoint their locations. Second, it adds a search algorithm that differentiates walls from wall-like surfaces, such as large doors.

Project point cloud onto planes

The first step of histogram analysis involves projecting the 3D point cloud onto 2D planes, as discussed in 6.2.2. These consist of the xz, yz, and zx planes that run orthogonal to the y, x, and y axes, respectively. Here, the first variable of the axis-pair describes the coordinate value for binning while the second describes the direction for counting points. The xz and yz planes bin along the x and y axes, respectively, and

count vertical points to identify walls; using xy and yx projections produce identical results even though they use a slightly different counting approach. The zx plane uses the z axis for binning and counts horizontally to identify the floor and ceiling; using the zy plane produces identical results.

Build histogram along each axis

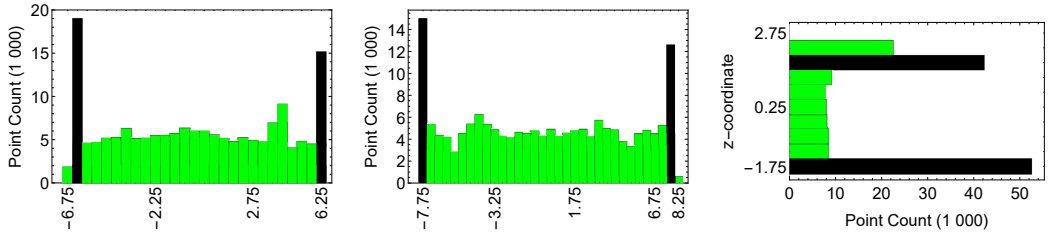
While building the histograms with fine resolution bins may seem ideal, this approach may be computationally burdensome and unnecessary when dealing with hundreds of thousands to millions of points since the goal of histogram analysis involves finding only two bins per histogram. A potentially more efficient approach uses two passes: the first with coarse resolution bins to identify the vicinities of structural features and a second with fine resolution bins to pinpoint their locations.

When using coarse resolution bins, the possibility exists that intra-room partition walls may produce bins with higher counts than the room's actual walls, such as in Figure 6.13, in which the highest count bin may not represent the room's boundary wall. This study addresses the problem by first isolating bins within a fixed distance from the point cloud's boundary and then selecting the bin with the highest count per feature. This results in the selection of six coarse resolution bins. Figure 6.14 shows the six bins for the sample data using a bin size of 50 cm.

6.2.4 Phase 4: Fine thresholding

After initial thresholding, each of the six coarse bins are partitioned into smaller bins to pinpoint the location of the floor, ceiling, and walls. Figure 6.15 shows the fine resolution histograms, which use a bin size of 1 cm. These six histograms produce six coordinates that combine to describe the eight corner points of the rectangular room.

Figure 6.13: This thoroughly scanned partition wall may have a higher point count than the partially scanned boundary wall in the background. Note that this point cloud shows a different room from the sample data used in this study.

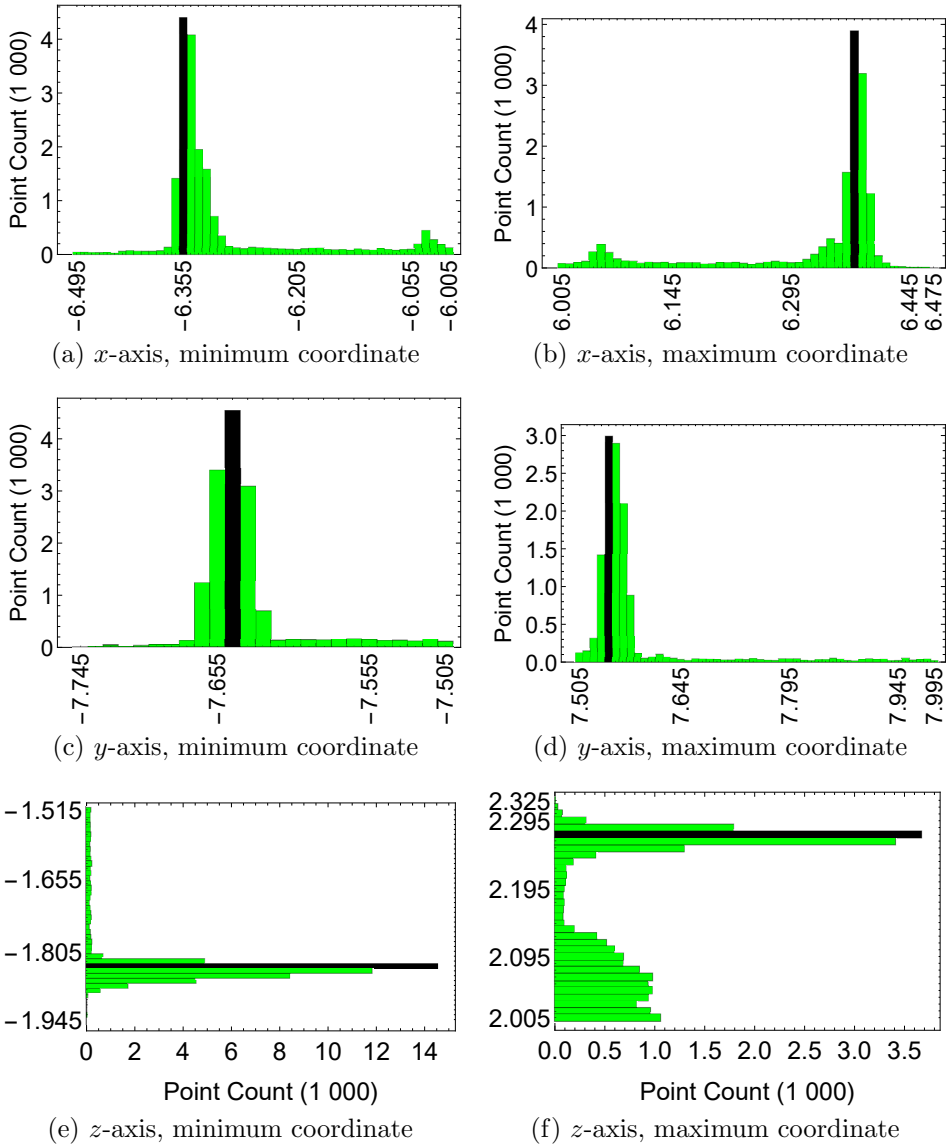


(a) Along x-axis (b) Along y-axis (c) Along z-axis

Figure 6.14: Coarse resolution histograms of the sample data using 50 cm bins

6.2.5 Phase 5: Building and placing the model

The six coordinates from fine thresholding provide all the information needed to build a simple rectangular model of the room, but it still remains at the origin $(0, 0, 0)$ as illustrated by Figures 6.16a and 6.16b for the sample data. Recall that moving the point cloud to the origin used the following transformation matrices, applied in the following order: $\mathbf{T}_1$; $\mathbf{R}_1$; $\mathbf{S}$ from SVD; and $\mathbf{R}_{2z}, \mathbf{R}_{2y}$, and $\mathbf{R}_{2x}$ from image-based alignment. Moving the model to the point cloud's original location requires a reverse transformation involving



(a) x-axis, minimum coordinate (b) x-axis, maximum coordinate

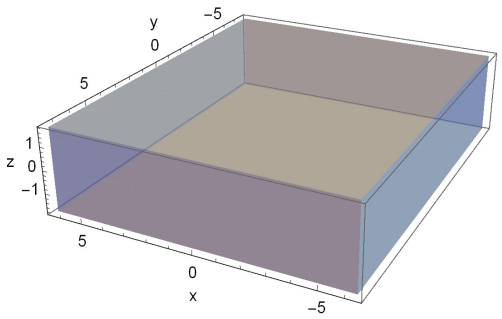
(c) y-axis, minimum coordinate (d) y-axis, maximum coordinate

(e) z-axis, minimum coordinate (f) z-axis, maximum coordinate

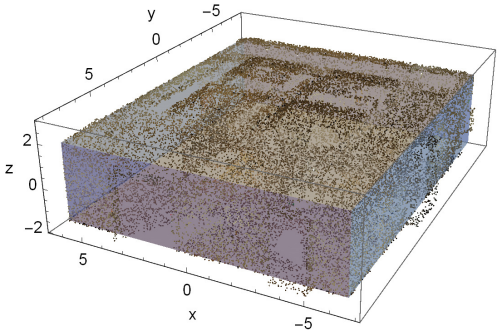
Figure 6.15: Fine resolution histograms of the sample data using 1 cm bins

the inverse matrices applied in reverse order as shown in Equation 6.12 to form the combined matrix $\mathbf{M}_{reverse}$, where the transpose and inverse are equivalent for the rotation matrices. Applying $\mathbf{M}_{reverse}$ to the model places it in the point cloud's original location as illustrated in Figure 6.16c for the sample data.

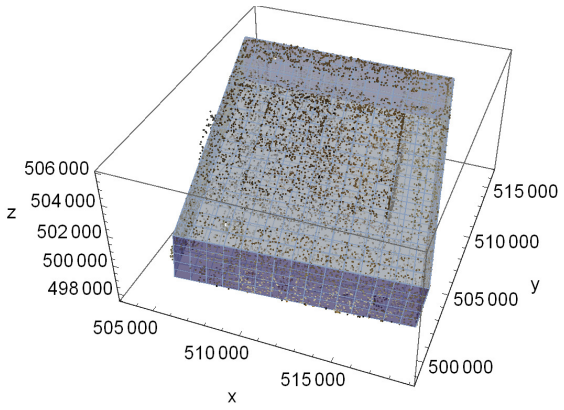
$$\mathbf{M}_{reverse} = \mathbf{T}_1^{-1} \cdot \mathbf{R}_1^{\top} \cdot \mathbf{S}^{-1} \cdot \mathbf{R}_{2z}^{\top} \cdot \mathbf{R}_{2y}^{\top} \cdot \mathbf{R}_{2x}^{\top} \tag{6.12}$$



(a) Geometric model at origin



(b) Model with point cloud at origin



(c) Model placed at point cloud's original location and orientation

Figure 6.16: Model of rectangular room derived from point cloud

6.3 Conclusion

This chapter examined a simple implementation of histogram thresholding for identifying the floor, ceiling, and walls in the point cloud of a rectangular room based on techniques developed by Khoshelham and Díaz-Vilariño (2014) and Díaz-Vilariño et al. (2015). Whereas Díaz-Vilariño et al. (2015) used cluster means from the EGI and direct vector projection for the alignment, this study used the more precise method of the statistical mode for each cluster via gridded orthographic projections and used singular value decomposition to provide pure rotations that preserve the original shape of the

point cloud. Once axis-aligned, the point cloud is processed using a simple histogram approach to identify the outer boundaries of the room — i.e., floor, ceiling, and walls — in contrast to Khoshelham and Díaz-Vilariño (2014)'s more advanced method that also identified interior walls. Whereas Khoshelham and Díaz-Vilariño (2014) used fixed sized bins, this study used a two-step process that starts with large bins to identify the general vicinity of the boundary features followed by small bins to pinpoint their locations.

While the technique used in this study demonstrates several key concepts for segmenting point clouds of rooms, it has several major limitations. First, this technique only works with rectangular rooms having orthogonal floors, ceilings, and walls; it will not work with angled or curved structural members. Second, it only identifies the room envelope but does not identify any type of openings such as doors and windows. Finally, this technique works only on the point cloud of a single room; it will not segment structural features for a multi-room or multi-level point cloud. Nonetheless, the simplicity of this technique may make it suitable for use as a teaching tool for introducing concepts relevant to segmenting point clouds for indoor space modeling.

Chapter 7

From point cloud to geometric model

The modeling standards and file formats section uses material from a paper published in the Geographical Information Systems Theory, Applications and Management (GISTAM) 2017 conference proceedings (Chen and Clarke 2017).

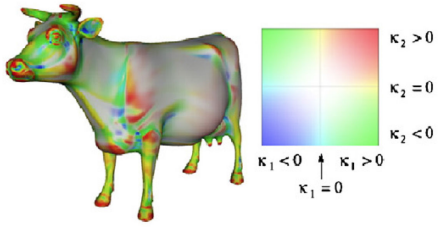
Chapter 5 covered point cloud segmentation, which classified the point clouds of indoor spaces into subgroups based on geometry and semantics. This chapter reviews automated methods for transforming those segmented indoor point clouds into geometric models followed by a review of some widely accessible file formats for the storage, transmission, analysis, and rendering of those models. As opposed to a point cloud, a geometric model uses a math-based geometric form that is suitable for digital processing (Remondino 2003). They offer several advantages over point clouds: they have a more compact form, can be manipulated as coherent objects, can be rendered for photorealistic visualizations, and can use numerous standard formats optimized for different applications.

7.1 Approaches to geometric modeling

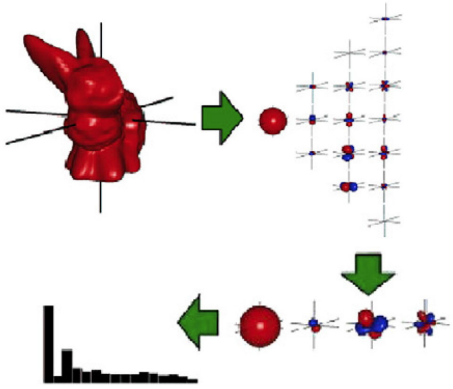
Geometric modeling usually takes a surface-based or volume-based approach that defines shapes either implicitly or explicitly (Ganovelli et al. 2015; Tang et al. 2010). The surfaced-based approach models only the outer shell of objects, while the volume-based approach models objects as solid volumes. Since the shell of an object also represents it physical boundary, the surface-based approach is often referred to as the boundary representation or *B-Rep* approach. Indoor mapping applications can use both types of representations, but models derived from point clouds often use B-Rep since current remote sensing technology can only detect object surfaces but not what lies underneath. Four of the most common types of B-Rep models are implicit surfaces, polygon meshes, parametric surfaces, and subdivision surfaces, with the latter three representing explicit methods (Ganovelli et al. 2015; Tang et al. 2010). Implicit methods imply the shape of an object through intermediate values but inherently do not directly describe its physical form.

7.1.1 Implicit surfaces

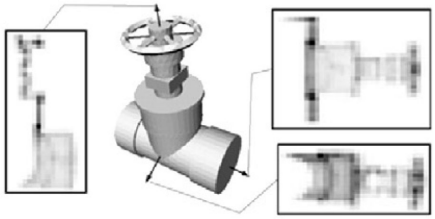
Implicit surfaces use intermediate descriptors rather than explicit descriptions of a surface, making them ill-suited for generating 3D models. However, their generic nature makes them well-suited for segmentation and object recognition, since the properties of the point cloud can be compared to implied descriptors, such as curvature or normal vector directions, as illustrated in Figure 7.1. The extended Gaussian image (EGI) covered in Chapter 6 is an example of an implicit surface descriptor. Implicit surfaces can be derived from measured objects or from libraries of models, such as a library of furnishings for furniture recognition (Tang et al. 2010).



(a) Curvature



(b) Spherical harmonics



(c) Spin images

Figure 7.1: Examples of implicit surface descriptors from Tang et al. (2010)[1]

7.1.2 Explicit surfaces

Explicit surface modeling directly describe the modeled surface using polygon meshes — e.g., lines and surfaces defined by vertices — or idealized parametric surfaces. These methods can sometimes be combined or used as steps within a larger modeling process.

Polygon meshes

Polygon meshes partition the surface of an object into a continuous surface of connected polygons, such as triangles or quadrilaterals, with the term *tessellation* describing

[1] a. Rusinkiewicz (2004); b. Kazhdan, Bolitho, and Hoppe (2006); c. Johnson et al. (1997)

a special case where the polygons are equally sized. For point clouds, the simplest approach involves directly connecting each point with its nearest neighbors to form a series of triangular surfaces (Marton, Rusu, and Beetz 2009). However, this naive approach can result in excessive or insufficient point cloud densities, noise, and voids — all of which require various post-processing measures to address. While direct meshing may work well for visualization, it can result in irregular surfaces that have limited value for indoor mapping and modeling.

Another approach to meshing a point cloud involves first modeling with B-Rep or volume-based techniques to achieve a smooth surface and then meshing the resulting surface instead of the point cloud itself to produce a smooth mesh. This approach is often used in situations where the original model exists in parametric form but some external requirement calls for the use of meshes. For example, in the past, the Industry Foundation Classes (IFC) for building information modeling (BIM) did not support parametric surfaces created in proprietary BIM software; these had to be converted to mesh format for transfer as an IFC-compliant file.

Parametric surfaces

A parametric surface describes a curved 2D surface in 3D space that uses only a small set of parameters — determined by control points in mesh form — that can oftentimes model shapes not possible with other methods (Moore and Smith 2002; Ganovelli et al. 2015). Examples range from simple shapes such as spheres, cylinders, cones, and ellipsoids to more complex surfaces. These parameterized surfaces can take an interpolation form or an approximate form, with the surface of the former passing through all control points and the latter using the control points only as a guide. Two widely used approximate approaches include Bézier patches and NURBS (non-uniform rational basis

splines), with the former more closely following control points and the latter exhibiting more gradual changes in shape.

Subdivision surfaces

Subdivision surfaces provide a compromise between the highly discrete meshes and the smooth and continuous parametric surfaces (Ganovelli et al. 2015). As with parametric surfaces, subdivision surfaces use a control mesh for surface definition. However, subdivision uses various schemes for subdividing the surface into progressively smaller cells that can significantly loosen the geometric constraints of parametric modeling, allowing the modeling of complex features using relatively small meshes (Ganovelli et al. 2015; Botsch 2010).

7.2 Approaches to geometric modeling from indoor point clouds

A sample of 57 publications dealing with the processing of indoor point clouds showed that about half the papers stopped at segmentation while the other half continued with geometric modeling using some form of B-Rep surface meshing. Of those that went on to geometric modeling, many used some form of surface extrusion from simplified 2D representations, others directly derived planar patches from the 3D points, and a handful of others used other methods.

Direct from region growing

Sanchez and Zakhor (2012) used RANSAC during segmentation to deduce alpha shapes, computed the vertices of their concave hulls, and then used split and merge to

remove redundant vertices to the unspecified mesh. Adán and Valero (2014) and Valero, Adán, and Bosché (2016) segmented walls, floors, ceilings, windows, and doors using RANSAC plane fitting and directly converted these planes into *quadrilateral meshes*, with the plane corners forming the vertices. For columns, they used a least-squares best fit for a circle using a projection view on the xy-plane and extruded from floor to ceiling using an unspecified B-Rep method. They then used ready-made mesh furniture models from a facilities management database to match and place furniture in the model. Mura et al. (2014) used a simple region growing method to form planar patches which they later merged for form boundary edges. Instead of extruding the walls from the resulting floor plan, they used a more robust statistical approach to height estimation and represented the model using what appeared to look like a quadrilateral mesh. Díaz-Vilariño et al. (2015) used region growing to segment planar surfaces from which they derived vertices for the quadrilateral mesh model. Oesau (2015) used region growing with voxels to deduce room structure then modeled with triangular meshes.

2.5D extrusions from thresholding

Budroni (2013) first created floor plans using Okorn et al.'s thresholding method and extruded the lines from floor to ceiling using an unspecified B-Rep method. Turner (2015) also extruded floor plans, created during segmentation using voxel carving, from floor to ceiling for the walls and represented all surfaces using *triangular meshing* applied to a quadtree structure. They then applied texture mapping to add an element of photorealism. Ochmann et al. (2016) likewise constructed piecewise linear graphs of the walls and extruded them from floor to ceiling using an unspecified B-Rep method.

Other thresholding method

Khoshelham and Díaz-Vilariño (2014) took a different approach to thresholding by aligning cuboids with histogram peaks and converting the merged cuboids into walls using an unspecified B-Rep method.

Wireframe

Researchers affiliated with Yonsei University used a wireframe approach to geometric modeling, which can be seen as a variation of mesh modeling. Kim, Son, and Kim (2013) segmented the point cloud with RANSAC and projected the resulting planes onto a horizontal occupancy grid from which they traced the structures boundary and apparently extruded for the wireframe model. Yoon, Jung, and Heo (2015) built a wireframe model of two complex intertwined staircases by combining Okorn et al.'s method for planes and RANSAC for all other surfaces and tracing the features to create the wireframe. Jung et al. (2016) projected points onto horizontal and vertical planes to binary map and performed tracing using Douglas-Peucker segmentation to form the wireframe.

True BIM modeling

Thomson and Boehm (2015) used an algorithm called RANSAC in the C++ Point Cloud Library for producing planar patches which they later merged for final segmentation. They then used this information to extrude BIM structures using IFC standard models for walls (IfcWallStandardCase) and floors (IfcSlab) using assumed thicknesses. This was the only instance in the 57 papers that provided a scan-to-BIM conversion for indoor spaces. It is unlikely that these were volume-based constructive solid geometry (CSG) models, since older versions of IFC almost exclusively supported B-Rep models.

7.3 Modeling standards and file formats for indoor mapping and modeling

Once geometric modeling has been finished, the resulting model must be saved in a certain format for storage, dissemination, rendering, and analysis. This section reviews prevailing open source and widely accessible modeling standards and data formats for computer-aided design (CAD), building information modeling (BIM), geographic information systems (GIS), and computer graphics (CG) systems and examines their suitability for indoor mapping and modeling. Material in this section builds on the work of Karimi and Akinci (2010), Kolbe (2009), and Zlatanova, Stoter, and Isikdag (2012).

7.3.1 Computer-aided design and geographic information systems

Computer-aided design and geographic information systems represent two distinct approaches to making and using spatial data, each with its own set of domain-specific applications, where each term describes a system or process rather than specific software. Facilities-based CAD has theoretical underpinnings from architecture and civil engineering while theories that drive GIS come largely from geography, geographic information science, and geoinformatics.

Computer-aided design

The construction and manufacturing industries developed CAD for producing electronic versions of previously hand drawn technical drawings. It was first called computer-aided drafting and later redesignated as computer-aided design as the software and processes grew more sophisticated (Akin 2009). Practitioners in the architecture, engineer-

ing, and construction (AEC) industry use CAD for designing and detailing built-up infrastructure, which can range from small residential houses to highways. Characteristics of AEC-based CAD include providing very fine levels of detail, use of a locally-referenced Cartesian coordinate system for buildings, and a predominant use of 2D space even though 3D capabilities exist in CAD.

Over the past decade, building information modeling (BIM) has emerged as the next generation of CAD that takes advantage of advances in computing and object-oriented processes (Barnes and Davies 2015; Succar 2009; Suermann 2009). As with CAD, BIM describes a process rather than specific software and, like CAD, it also uses a local Cartesian coordinate system for spatial referencing. Three features distinguishing BIM from CAD consist of its use of an object-oriented model, its use of databases for object management, and its native use of 3D space. Whereas conventional CAD uses 2D line drawings to graphically depict physical objects, BIM uses solid 3D geometries, which it can then render into 2D CAD line drawings as needed. Another advantage of BIM over conventional CAD involves the ability to capture dimensions of data beyond 3D, e.g., time, resources, and cost. These elements make BIM suitable for use in post-construction activities such as building maintenance and resource management, i.e., building operations. As a result, BIM is often associated with the term AECO where the "O" can stand for operations or owner.

Some sources treat BIM and CAD as two separate concepts while others treat BIM as a subset of CAD. While this paper regards BIM as a form of CAD, it will refer to the two terms separately with the understanding that CAD refers to the traditional 2D CAD process and BIM to the 3D process.

Geographic information systems

GIS emerged in the 1960s as a computer-enabled approach to analyzing land use and rapidly expanded to other areas in the 1980s with the advent of affordable personal computers. Within two decades, GIS became the de facto technology for all things geospatial, to include the storage, manipulation, visualization, and analysis of anything with geographic information (Goodchild 2000). GIS has since taken many forms that impact numerous aspects of modern life, such as in vehicle navigation, customizable computer maps, and virtual globes such as Google Earth. It has also led to a massive growth in all forms of geospatially-enabled data ranging from online government-furnished maps to geotagged photos and social media.

Geographic information science (GIScience or GISc) — also known as geoinformatics — provides the underpinnings for technological GIS systems, and the growth of GIS in the 1980s and 1990s fueled a parallel rise in GIScience's prominence as a scientific discipline (Goodchild 1992, 2000), incorporating ideas from other disciplines such as statistics, information systems, and graphic design to explore spatially-based phenomena. In turn, it led to both empirical and theoretic discoveries that have provided greater insights into the geographic world, an especially important contribution in a world experiencing climate change and major shifts in human migration and demographics. However, GIS and GISc in their present form almost exclusively deal with the outdoor world outside of enclosed areas such as buildings and underground facilities. Considering that most people spend 80% to 90% of their lives indoors, GIS and GISc have much to offer in the study of indoor worlds (Klepeis et al. 2001; Roberts 2016; Goodchild 2011).

7.3.2 Characterizing file formats and standards for indoor use

Four properties were used to evaluate the suitability of file formats and modeling standards for indoor applications. These four properties consisted of geometric representation, level of abstraction, spatial referencing method, and level of detail support. While these properties could apply equally to indoor and outdoor environments, they were selected to specifically evaluate indoor applications in this dissertation.

Geometric representation

Geometric representation describes the method used to mathematically represent objects. Section 7.1.2 identified three primary types of boundary representations or B-Rep: polygon meshes, parametric surfaces, and subdivision surfaces. This section adds volumetric representations in the form of constructive solid geometry (CSG) and voxels (Ganovelli et al. 2015). Whereas parametric surface modeling uses smooth 2D mathematical surfaces in 3D space, CSG represents 3D objects as combinations of solid geometric primitives, such as spheres, cuboids, and cylinders. Voxels (volume elements), another volumetric method, are the 3D equivalent of 2D rasters and provide another way to model 3D volumes.

Abstraction level

In computer programming, abstraction describes the process of revealing only relevant information required to run a computer program and hiding all other implementation details to reduce clutter and prevent confusion. Abstraction level thus describes the extent to which a program can interact with natural human language, with higher levels indicating nearness to natural language and lower levels to machine language (Stroustrup 2014). When dealing with building models, this paper defines *abstraction level* as the

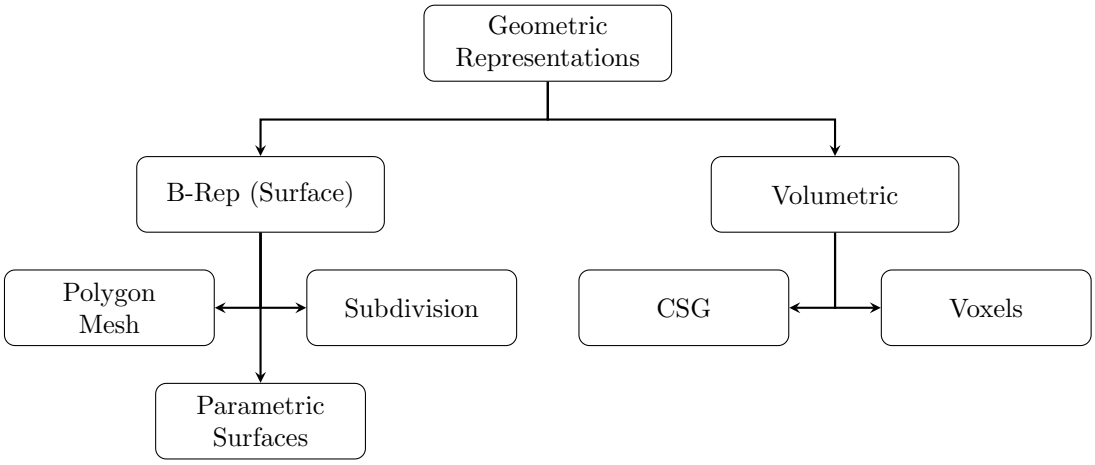


Figure 7.2: Methods of geometric representation

extent to which a geometric model represents an object in natural language terms. For instance, a low level representation of a door can require the description of mesh vertex coordinates and their red-green-blue color values while a high level representation can involve invoking a "door" descriptor with attributes of material and dimensions, hiding the details of mathematical implementation. High level abstraction also allows objects to be generalized and used as templates for other objects via inheritance and polymorphism (Ibrahim and Krawczyk 2003; Rüppel, Meissner, and Möller 1993). For example, a double panel oak door can be an object based on a generic door class. The door itself, with all its component parts and attributes, forms a single object, and copies of that door object placed within a building model represent its instances.

This study uses four levels of geometric abstraction: none, low, medium, and high. No geometric abstraction applies only to the IndoorGML standard that deals mainly with topology. Low level abstraction describes formats and standards that support only simple geometries without semantics, such as simple vertices, lines, and polygons. Medium level abstraction provide more advanced capabilities — such as the grouping of geometric shapes or limited semantics — that fall short of true object-oriented representation.

Finally, high level abstraction represents true object-oriented capabilities with inheritance and semantic capabilities.

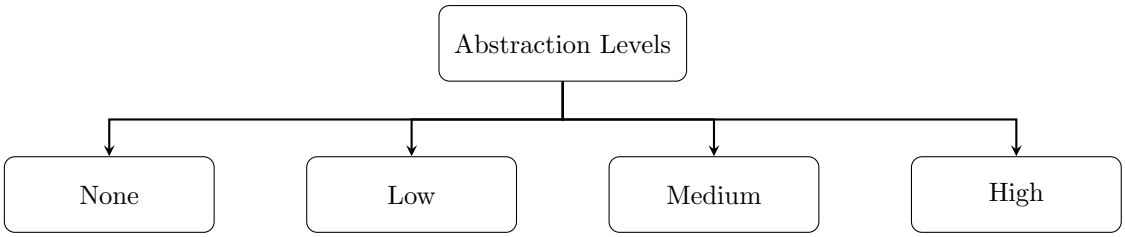


Figure 7.3: Abstraction levels

Spatial referencing support

Coordinate referencing describes the measurement system used for placing a geometric model in space, which consists of a coordinate system (i.e., measurement conventions) and datum (i.e., starting point) that together form a unique coordinate reference system or CRS (International Organization for Standardization 2002; Bernhardsen 2002; Van Sickle 2010). Building and indoor space modeling usually use CRSs based on the Cartesian coordinate system paired with a locally defined datum due to the relative ease of working with linear measurements. However, maps of the outdoor environment use a wide variety of systems that can include linear horizontal and vertical measurements as well as angular geographic measurements. The International Association of Oil and Gas Producers (IOGP) (2017) identified over 5,700 unique CRSs in the 2017 edition of its widely referenced EPSG CRS database. With no limit to the number of available CRSs, flexibility in accommodating different types of CRSs can play an important role when integrating indoor with outdoor models. Thus, the level of support for coordinate referencing provides an indication of a format or standard's flexibility in accommodating these different types of CRSs. Another approach to spatial referencing involves the use of natural language descriptors in the form of geographic identifiers. Identified places

often have associated coordinates although some ambiguity may exist with the object's actual location, such as when a single coordinate tuple is used to specify the location of a large building or city.

This study evaluates whether a standard or file format supports coordinate referencing or identifiers. It further divides the coordinate referencing category into local engineering coordinates and Earth-based geodetic coordinates. While the term *geodetic coordinates* is traditionally used to describe horizontal locations, it is loosely used here to describe Earth based horizontal and vertical locations.

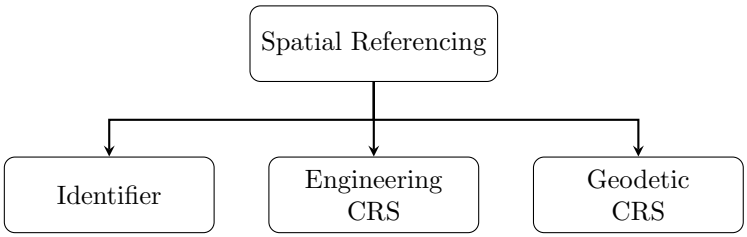


Figure 7.4: Spatial referencing support

Level of detail

The ability for a model to have multiple levels-of-detail (LODs) describes the ability to use different geometric forms to represent the same object at different viewing levels, akin to preset map scales for outdoor maps (Pham, Ruas, and Libourel 2015; Tolmer et al. 2013). However, the actual implementation of LOD varies by discipline. In computer graphics, LOD is often associated with polygon counts where a low LOD model is a pared down version of a high LOD model in terms of the number of vertices and faces. In BIM, LOD takes on a completely different meaning where LOD stands for *level-of-development*, rather than detail, that defines geometric form based on the level of certainty during the design and construction process — some authors refer to this as LODt to prevent confusion (Reinhardt and Bedrick 2016). Geometric form and LODt are only generally

but loosely coupled in BIM. In cartography, semantics plays a greater role in determining geometric form for multiple LODs where vastly different geometric shapes can be used to represent the same object.

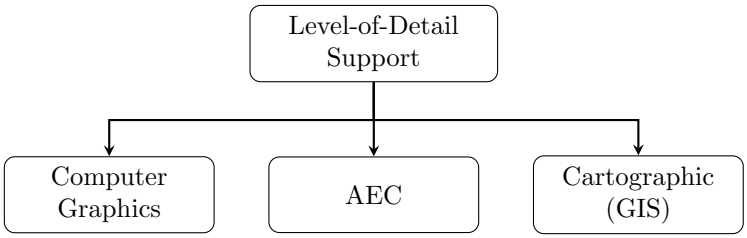


Figure 7.5: Level-of-detail support

7.3.3 Review of BIM and GIS formats and standards

This section provides a review of open-source or widely accessible BIM and GIS formats and standards that have potential relevance to 3D indoor mapping and modeling. In many instances, standards are tightly coupled with file formats so that a modeling standard also provides specifications for the storage and transmission of data. Additionally, many other proprietary formats exist that may provide greater capabilities for indoor mapping and modeling, but they have been omitted since they often require expensive proprietary software.

Industry Foundation Classes (IFC)

Background Building information modeling (BIM) is a generic term that describes processes that use 3D digital models for coordinating all aspects of work on buildings and infrastructure throughout their lifecycles from concept to disposal (Barnes and Davies 2015; NBS 2017). Over the past decade, openBIM has increasingly gained acceptance as the de facto set of guidelines governing the BIM process as well as the products produced through BIM (Johnson 2015).

Work towards openBIM began in 1995 when CAD software maker Autodesk led a private alliance of twelve companies to develop an open and international standard for the interchange of building industry data (buildingSMART 2016). In 1996, this group formally became the International Alliance for Interoperability (IAI) and was renamed buildingSMART in 2008. In its on-going efforts to make building data universally accessible, buildingSMART also established partnerships with the International Organization for Standardization (ISO) to make its standards internationally recognized and the Open Geospatial Consortium (OGC) to integrate BIM with geospatial data.

IFC standard The concept of BIM is widely associated with IFC since it was the first open standard published for BIM that also enabled BIM software to openly exchange data (buildingSMART 2016). IFC provides a rich standard for representing, storing, and transferring building, infrastructure, and site data, with a singular focus on supporting the AECO industry. It uses a 3D object-based model for abstracting real-world entities; can use CSG and B-rep geometric representations (buildingSMART 2015); has robust and standardized semantics, especially when paired with the buildingSMART Data Dictionary (bSDD); uses an engineering CRS, i.e., assumes a flat Earth; and supports LODt but not LOD. IFC data can be transmitted in one of three ways (file extensions shown in parentheses): the ISO STEP file structure (.ifc), a text file formatted using extensible markup language, XML (.ifcXML), or a compressed file (.ifcZIP) containing a .ifc or .ifcXML file in the main directory.

LODt for openBIM refers to the level of design refinement of building objects along various stages of the design and construction process (Bedrick 2013) and can serve as an indirect indication of a model's reliability. openBIM has largely adopted LODt specifications from the American Institute of Architects (AIA), with the most recent version of the specification, dated 2016, outlining five LODts (LOD 100, 200, 300, 400, and 500)

following AIA protocol G202-2013 and a sixth for openBIM (LOD 350) (Reinhardt and Bedrick 2016).

Other related openBIM standards In its present form, openBIM consists of three basic standards: IFC, the international framework for dictionaries (IFD) implemented as the bSDD, and information delivery manuals (IDMs) (buildingSMART 2017c). buildingSmart also recognizes external specifications such as the BIM Collaboration Format (BCF). bSDD serves as the openBIM implementation of the IFD using a multi-lingual data dictionary to provide ontologies and characteristics of objects in IFC (Petrie 2016). IDM provides guidance on documenting processes and building information shared among multiple organizations (Karlshøj 2011). Subsets of IFC identified using IDM are captured in official multi-view definitions (MVDs), which provide the subset schemas (buildingSMART 2017b). The BCF is a recent addition to the openBIM standards that was developed to improve communication capabilities using the BIM model itself as the medium (buildingSMART 2017a). Prior to BCF, users who wanted to communicate issues using a BIM model needed to exchange the entire model as bulk data. BCF provides an open format that exchanges only vital elements between different software applications.

City Geography Markup Language (CityGML)

Background CityGML serves as a standard data model and an XML-based data exchange format for the modeling of cities in 3D. It exists as one of several application-specific application schemas of the Open Geospatial Consortium (OGC) Geography Markup Language (GML) standard, an umbrella international standard for describing real world phenomena that have geospatial properties. Similar to how openBIM and IFC emerged from the efforts of a consortium of companies in the AECO industry, CityGML emerged from the work of a group called the Special Interest Group 3D (SIG3D) con-

sisting of companies, academic institutions, and government entities in Germany. In 1999, the German state of North Rhine-Westphalia established the Geodata Infrastructure North Rhine-Westphalia (GDI NRW) to create a market for realizing the potential economic benefits of geographic information (Brüggemann and Liebig 2000). SIG3D was formed from this initiative in May 2002 with the goals of advancing standards for the open exchange and visualization of 3D geographic data (GDI-DE 2016). SIG3D's work on the open data exchange standard resulted in an early version of CityGML (Open Geospatial Consortium 2012). OGC began examining CityGML in 2004 within its 3D Information Management (3DIM) Working Group, which eventually led to the adoption of CityGML as an OGC standard in 2008. SIG3D continues to play a major role in developing CityGML along with other 3DIM members including the U.S. National Institute for Building Sciences (NIBS) and buildingSMART.

CityGML standard With its origins in GIS, CityGML has a strong focus on urban modeling with an emphasis on buildings. CityGML uses a 3D object-based model for abstracting real-world entities, similar to IFC; exclusively uses B-rep with standardized semantics; supports use of both engineering and geodetic CRSs; and supports five LODs (Buyukaslih, Isikdag, and Zlatanova 2013; Kolbe, Gröger, and Plümer 2005). When using an engineering CRS, the CityGML 2.0 specifications recommend also identifying an anchor point based on a geodetic CRS (Open Geospatial Consortium 2012).

LODs for CityGML closely follow the cartographic concept of LOD. CityGML has five LODs sequentially numbered from LOD0 (most generalized) to LOD4 (most detailed). Since the standard treats features and geometric representations separately, a single feature can have different geometric appearances at different LODs and it can even have different appearances based on different themes. LOD4 has particular relevance to indoor

cartography since it provides the only LOD that reveals indoor features; building interiors remain void and empty from LOD0 to LOD3.

Indoor Geography Markup Language (IndoorGML)

Background Whereas IFC and CityGML address 3D geometries and the semantic aspects of buildings, the IndoorGML standard was developed for the sole purpose of supporting indoor navigation. OGC officially approved IndoorGML as an OGC standard in 2015 and it serves to complement, rather than supplant, other standards such as CityGML and IFC.

IndoorGML standard IndoorGML uses a non-visual object-based model for abstracting real-world entities; has minimal support for geometric representation, instead using topological network graphs; has standardized semantics; supports use of both engineering and geodetic CRSs; but does not have LODs (Kim, Yoo, and Li 2014; Nagel et al. 2010).

As a standard for topological modeling, IndoorGML uses non-overlapping cells to represent indoor spaces, such as rooms. These object-based cells need neither dimension nor spatial location, although the standard supports adding either or both as an option. Cells can have topological relationships and can relate to geometry by linking to CityGML or IFC. While the standard does not require geometry-based LODs, it uses the concept of multi-layering to represent different uses of the same space. For instance, one layer can represent pedestrian travel while another can represent wireless internet coverage.

Other CAD standards

DXF was initially released by CAD software maker Autodesk in 1982 as an open file format to facilitate interoperability of its AutoCAD software with other CAD programs. It provides a visualization-only approach to abstraction using simple geometric repre-

sentations (e.g., points, lines, arcs/circles, polygons, etc.) with no semantics; uses an engineering CRS; and has no support for LODs (Autodesk 2011; Ibrahim and Krawczyk 2003). Although the DXF standard uses the terms *object* and *entity*, it defines entities as actual geometric forms (e.g., lines and hatches) and objects as non-graphical and non-geometric properties (e.g., line types and groups) (Autodesk 2011). Autodesk's native DWG CAD format for its AutoCAD software shares many of the same properties as DXF with additional support for BRep in the form of tesselation (meshes) through sweeping and extrusion (Library of Congress 2016).

The DGN format is Bentley's native CAD format for its Microstation software and supports both engineering and geodetic CRSs, with no LOD capabilities and no semantics, but supports points, lines (including curves), and areas in 2D/3D and extruded B-Reps (Bentley Systems 2016).

7.3.4 Other GIS standards

Shapefile

GIS software maker Esri released the shapefile format in the 1990s as a way to represent, store, and transfer simple 2D geometric structures of geographic entities for its once state-of-the-art ArcView GIS software (Library of Congress 2011b). Though often associated with the .shp file extension, the shapefile format actually describes a collection of multiple files. Shapefiles require a minimum of three files stored in the same directory: feature geometries or shapes (.shp), shape index (.shx), and attribute tables (.dbf). They can also include various other optional files, the most relevant being the projection file (.prj) for storing coordinate reference system specifications. Shapefiles represent real-world entities using simple 2D and 3D vector geometries based on points, lines, polygons, and isohedrons. Shapefile features have no topological capabilities, limited semantic ca-

pabilities through the use of attributes, can use either engineering or geodetic CRSs through the optional .prj file, and have no built-in capabilities for defining LOD. For 3D representation, shapefiles use multipatch geometry, a type of B-rep representation (Esri 2008).

Geodatabase

Esri released the geodatabase format in 1999 based on an object-relational data model. Geodatabases use objects called features to represent real world entities and can take advantage of object-based inheritance and semantics through the use of feature classes. In terms of geometry, the geodatabase format uses the same vector representations as shapefiles — i.e., points, lines, polygons, and 3D multipatches — with additional support for rasters in 2D (Library of Congress 2011a). The format currently does not support voxels although individuals at Esri have discussed the idea (*Ten Questions for Esri: 3D Expert, Nathan Shepard* 2015). For spatial referencing, the geodatabase supports both engineering and local CRSs as well as referencing by identifiers. This format also supports different user-defined LODs due to the separation of features and geometries. Esri has three different implementations of the geodatabase — personal, file, and enterprise — each with increasing sets of features and capabilities (Esri 2016). Personal geodatabases use Microsoft Access data files to store data, file geodatabases use a directory-base file structure, and enterprise geodatabases use database servers.

FISDM

The open source Facilities Information Spatial Data Model (FISDM) emerged in 2014 from the Building Interior Space Data Model (BISDM) released by Esri and several other organizations in 2007 (Rich and Smith 2014). FISDM is not a standard but an integration platform for CAD, BIM, and GIS data built on Esri's geodatabase format

that can fuse GIS data with CityGML, IFC, and other formats. It includes a data model, cartographic templates, sample data interoperability kits, and sample starter kits. While BISDM focused exclusively on interior spaces, FISDM extended those capabilities to include exterior spaces and the outside built environment. FISDM has the same 2D and 3D capabilities of its underlying geodatabase file format.

KML

KML is an XML-based OGC international standard for providing 2D and 3D geographic visualizations in online mapping and virtual globe browsers. It was originally developed as the Keyhole[2] Markup Language for use with the Keyhole Earth Viewer, an innovative virtual globe partially funded by the U.S. Intelligence Community and acquired by Google in 2004 to become Google Earth (Garfield 2015; In-Q-Tel 2003). KML became an official OGC international standard in 2008. KML provides a lightweight but extendable format for representing 3D features using B-rep (Isikdag and Zlatanova 2010). Although mostly a visualization format, the KML standard provides limited ability to add user-defined attributes to features. KML exclusively uses two specific geodetic CRSs: World Geodetic System of 1984 (WGS84) for horizontal coordinates and Earth Gravitational Model 1996 based on the WGS84 ellipsoid for elevations (WGS84 EGM96 Geoid). It also supports user-defined levels of detail through the use of regions (Burggraf 2015).

[2]The *KEYHOLE* codeword was used by the United States government to designate highly classified space-based intelligence during the Cold War.

7.3.5 Other graphics formats

While many of the previously reviewed BIM and GIS formats and standards work well for data collection, storage, and transmission, they may not provide the best performance in terms of rendering and visualization. This section reviews open source or widely accessible general purpose graphics formats that may be useful for visualizing models of indoor spaces.

X3D

X3D consists of a family of open ISO standards for representing and exchanging 3D scenes and objects and is managed by the Web3D Consortium (Web3D Consortium 2017). The X3D standards emerged from an earlier standard called the Virtual Reality Modeling Language (VRML), which had a geographic component called GeoVRML. X3D offers a quasi-object-based capability called grouping that allows for the definition and re-use of constructed geometries called groups. Other properties of X3D include collision detection and the representation of multiple LODs, as well as support for both local engineering and a limited number of geodetic CRSs (Web3D Consortium 2008). X3D data can be presented in three forms: XML, Javascript, and Java. With its goal of becoming the default 3D modeling language for the web, members of X3D continue to develop a growing set of capabilities. One such effort comes from the X3D CAD working group, which has continued to develop support for CAD models including the use of B-rep geometries (Brutzman 2012).

SVG

SVG is an XML-based 2D graphics format supported by many modern web browsers. Version 1.1, the latest version, supports line and area geometries (i.e., rectangle, circle,

ellipse, and polygon) and rasters, although it still lacks support for point geometry. Version 2, currently in draft form, has provisional mesh geometry capabilities that enable shape distortions, but it still omits point geometry. Similar to X3D, re-usable groups can be made in SVG, and the SVG data can link to other XML files for acquiring custom attributes (Adams 2005; Geroimenko and Geroimenko 2006). It natively uses a Cartesian coordinate based engineering CRS but has the flexibility to accommodate geodetic CRSs using one of three methods: a web-based uniform resource identifier (URI), a well-known CRS identifier, or directly defining the CRS within the xml document. SVG also supports multiple-representation LODs (Chang, Chuang, and Wang 2004).

COLLADA

COLLADA is an XML-based ISO-adopted standard for the exchange of 3D digital assets among different software applications (Barnes and Finch 2008; Khronos Group 2017). It uses an object-based approach for abstracting real-world entities, uses B-rep geometric representation, supports custom semantics, and supports multiple LODs. It supports use of a local engineering CRS and one geodetic CRS (WGS84 and WGS84-EGM96, following KML specifications).

3D PDF

3D PDF (portable document format) provides a convenient format for delivering 3D models using widely available PDF readers. 3D models do not reside in the PDF document itself but rather in an embedded 3D file saved in one of only two supported formats, U3D and PRC.

The U3D format only supports the visualization of geometric models, which can have attached attributes; however, it lacks object-based capabilities such as inheritance. U3D only supports the use of tessellated meshes, uses a local engineering CRS, and its imple-

mentation of LOD uses reduced polygon counts for the same geometric representation (Klawonn 2012), instead of using multiple representations.

The ISO-adopted PRC format similarly supports visualization of models with attached attributes but falls short of being truly object-based. In addition to tessellated meshes, PRC also supports other forms of B-rep (PDF3D 2015). PRC natively uses a local engineering CRS but some PDF implementations can provide software-based transformations to a geodetic CRS.

Minecraft

Minecraft is a simple, voxel-based game that achieved the distinction of becoming the second best selling videogame in history by 2016, behind the best-selling game Tetris (Peckham 2016). As an open-ended game, players can explore the many worlds of Minecraft and interact with their objects. Due to its simplicity and allure, Minecraft has found various uses in school curriculum, community planning, and academic research (Ertiö and Bhagwatwar 2017). Microsoft acquired Minecraft for over $2 billion USD in 2014 (Wingfield 2014).

While not a modeling standard per se, the Minecraft map format, called Anvil, allows users to create custom worlds, called creation maps, for importing into the software. As an example of the power of this feature, the Danish Geodata Agency released a full scale (1:1) Minecraft map of the entire country in 2014 allowing players to explore the country using Minecraft (Geodatastyrelsen 2014).

Minecraft uses object-oriented blocks as the most basic map element, with over 150 block types existing in the Minecraft library (Minecraft Wiki 2016). While object-oriented in nature, the attributes and behaviors of these blocks are largely constrained by the proprietary software program. Most blocks have a dimension of 1 meter on all sides, although smaller blocks exist for special applications or can be specially built using

Table 7.1: Summary of building modeling standards and file formats

Standard/Format	Geometry	Abstraction	Spatial Referencing	LOD Type	LOD Levels
IFC	**M** P | **C** V	N L M **H**	I **E** **G**	**AE** CG GIS User	Indoors: 6 / Total: 6
CityGML	**M** P | C V	N L M **H**	I **E** **G**	AE CG **GIS** User	Indoors: 1 / Total: 5
IndoorGML	N/A	N L M **H**	**I** **E** **G**	None	N/A
Shapefile	**M** P | C V	N L **M** H	I **E** **G**	None	N/A
Geodatabase	**M** P | C V	N L M **H**	**I** **E** **G**	AE CG **GIS** User	User-defined
FISDM	See geodatabase				
DXF/DWG	**M** P | C V **—** **▱** **●**	**N** L M H	I **E** G	None	N/A
DGN	**M** P | C V **—** **▱** ●	**N** L M H	I **E** **G**	None	N/A
COLLADA	**M** P | C V	N L M **H**	I **E** **G**	AE CG GIS **User**	User-defined
X3D	**M** P | C **V**	N **L** M H	I **E** **G**	AE CG GIS **User**	User-defined
SVG	**—** **▱** ●	N **L** M H	I **E** **G**	AE CG GIS **User**	User-defined
KML	**M** P | C V **—** **▱** **●**	N L **M** H	I E **G**	AE CG **GIS** User	User-defined
PDF (U3D)	**M** P | C V	N L **M** H	I **E** G	AE **CG** GIS User	User-defined
PDF (PRC)	**M** P | C V	N L **M** H	I **E** **G**	AE **CG** GIS User	User-defined

Level of support: **#** Full # Partial or with workaround # None
Geometry: **M**esh **P**arametric **C**SG **V**oxels **●** Point **—** Line **▱** Area
Abstraction level: **N** None (simple geometry) **L** Low **M** Moderate **H** High (object-oriented)
Spatial referencing: **I** Identifier **E** Engineering CRS **G** Geodetic CRS
Level-of-detail type: **AE** AEC/BIM **CG** Computer graphics **GIS** GIS/cartography **User** User-defined

pixel measurements (1 block face = 16 x 16 pixels). The program uses an engineering CRS (i.e., flat Earth), has horizontal dimensions of +/-29,999,999 blocks (i.e., meters) from the origin, a depth limit of 64 blocks, and a height limit of 191 blocks. The software automatically sets the LOD for the map.

7.3.6 Summary

Table 7.1 summarizes the building modeling standards and file formats covered in this paper based on a format used by T.H. Kolbe for Billen et al. (2014).

7.4 Evaluation of standards

Evaluation of the formats and standards looked at three areas: software accessibility, data sourcing, and indoor modeling capabilities. Software compatibility examined practical issues of working with and sharing indoor maps, data sourcing examined the relative importance of the formats themselves serving as a data source, and modeling capabilities assessed capabilities and potential limitations when used for indoor mapping and modeling.

7.4.1 Software accessibility

An open specification, vendor-neutral format allows software makers to incorporate it into their products for cross-platform exchanges. In contrast, closed formats such as DWG and DGN only work with proprietary software programs, such as AutoCAD and Microstation, that may be cost prohibitive for many users. All file formats covered in this review, except DWG and DGN, were open specification formats and had high levels of compatibility and accessibility with various software. In particular, IFC, DXF, and COLLADA were specifically designed as open exchange formats for maximizing compatibility and minimizing conflicts between different software programs. Nonetheless, some data loss can still occur when the open format does not fully support features in proprietary software. For example, the Autodesk Revit software for BIM supports parametric geometries while IFC does not; to export to IFC, Revit must convert parametric components into meshes, which results in a loss of information.

7.4.2 Data sourcing

While this dissertation focused exclusively on point clouds, existing building designs can also serve as convenient sources of data for creating indoor maps although they may

contain an unknown degree of uncertainty. Among the reviewed models and formats, IFC-based BIM models have the greatest potential use in indoor mapping due to their very native 3D geometry, high level of detail, and rich object-oriented semantics. However, few buildings currently have BIM models due to its relatively slow adoption in AEC. For older structures, CAD drawings provide the next best data source followed by printed drawings, which can be digitized through scanning or manual transcription. Most CAD and printed drawings use a 2D line format, which require manual interpretation to convert to 3D.

Even though CityGML has LOD4 for indoor modeling, few CityGML models actually take advantage of the standard at that level of detail. Of the 15 publicly-accessible urban models listed on the official CityGML website[3] in 2017, only one used LOD3 and none used LOD4. This may be due to a lack of use cases for indoor maps or lack of methods for economically capturing the geometry of indoor spaces — the subject of this research. Meanwhile, BIM will continue to serve as the most reliable source for detailed indoor data driven by a commercial need in the AEC industry.

7.4.3 Suitability for indoor modeling

Indoor mapping and modeling occupies an area between BIM and GIS, where it does not require the detailing capabilities of BIM but requires the ability to deal with finer resolutions than traditionally provided by GIS systems as illustrated in Figure 7.4.3. Since no such standard or format currently exists, the most suitable alternatives among the standards and formats reviewed in this chapter are IFC, CityGML, X3D, and SVG.

IFC provides a feature-rich platform for indoor modeling that enables the modeling of building components to the most minute detail. However, it has four potential draw-

[3]https://www.citygml.org/3dcities/

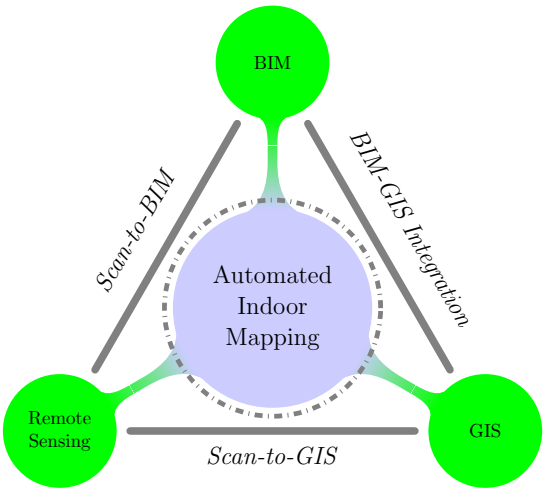


Figure 7.6: Indoor mapping in the context of existing approaches to indoor modeling

backs. First, BIM works best with volumetric building components rather than surfaces, making it necessary to make assumptions about structural features such as thicknesses and materials of walls, floors, and ceilings, as was done by Thomson and Boehm (2015). This adds a cumbersome and time-consuming step to the workflow, especially for non-BIM applications. Second, IFC uses the BIM concept of LODt, which prevents effective use of cartographic LOD in mapping applications. Finally, the feature-rich nature of IFC also makes it a data-heavy format — made heavier with the need to specify volumetric properties — that may impede data sharing and fast visualization. Finally, IFC has very limited support for geodetic coordinates, although this may represent a minor drawback.

CityGML, X3D, SVG, and GIS formats provide viable alternatives to IFC. CityGML provides the greatest level of support for the diverse requirements of indoor mapping in terms of data generation, storage, and transmission with its support for rich semantics, engineering and geodetic CRSs, the cartographic concept of LOD, and the ability to extend to other GIS domains using application domain extensions (ADEs). CityGML can also exchange data with IFC through BIM-GIS integration techniques that are actively being developed by openBIM and OGC. However, it falls short of providing full

support for indoor mapping due to its all-or-nothing single LOD for indoor spaces, where LOD4 shows everything indoors while LOD3 only shows the hollow shell of a building. Nonetheless, indoor LODs may be added at a later time either through a revision of the standard itself or an ADE once the concepts of indoor LOD have been developed.

CityGML works well for generating, storing, and transmitting indoor models, but it is not optimized for visualization. For visualization purposes, CityGML can export to X3D, SVG, or Esri's geodatabase formats which can all work with most modern web browsers. Among these, X3D has the greatest capabilities for displaying 3D indoor maps. X3D supports rich semantics, engineering and a handful of geodetic CRSs, and cartographic LODs. It can also display voxels, which may provide an alternative form of Minecraft-like visualization. While the relatively mature SVG standard can provide 2D visualizations, mature GIS formats used with web mapping servers — such as the open source OSGeo MapServer[4] or proprietary Esri ArcGIS Server — may provide a more practical solution, especially with an intermediary format such as FISDM. Using a GIS system for 2D indoor maps can also allow the indoor maps to seamlessly integrate with outdoor maps and take advantage of pre-existing GIS functionality.

While relatively new and not previously covered in this chapter, the partially proprietary WRLD web maps[5] provide a glimpse of the future of 3D indoor web mapping that fills the gap between BIM and GIS as illustrated in Figure 7.4.3. WRLD maps focus on 3D indoor space but place them in the context of a 3D outdoor GIS map rendered using game engine technology. Although impressive, a closer examination shows the limitations of their indoor maps and current indoor mapping technology in general. The current state of WRLD's indoor maps reflects the early years of Google's 3D mapping

[4]http://www.mapserver.org/
[5]https://maps.wrld3d.com/

efforts, when it manually built 3D models and invited the public to crowdsource models using its SketchUp software. At the time of writing, WRLD also manually builds many — if not most — of its indoor maps and has an on-going program for crowdsourcing. Indoor maps that appear to have been automatically generated use simple extrusions of floor plans for walls and lack other architectural features such as doors, windows, and furnishings.

7.5 Conclusion

This chapter covered the process of converting segmented point clouds of indoor spaces into geometric models and then reviewed a selection of modeling standards and formats for model generation, storage, transmission, rendering, and analysis. While geometric modeling itself has well-established theoretical and practical bases, the automated generation of building models from imperfect point clouds of indoor spaces remains relatively under-developed and an area of active research. Modeling standards and file formats present another challenge to the geometric modeling of indoor spaces. Presently, no standard or format exists that can fully support the anticipated requirements of indoor mapping. CityGML and IFC/BIM — which can inter-operate — provide perhaps the most capable frameworks for indoor mapping, limited most notably by weak support for multiple indoor cartographic LODs. However, to prevent putting the cart before the horse, theories and concepts of indoor cartography must first be developed before modeling standards can be established. Thus, the indoor cartography presents a potentially new field of study distinct from conventional outdoor cartography, architecture, and urban modeling.

Part IV

Indoor coordinate referencing

Chapter 8

Spatial referencing review

8.1 Overview of spatial referencing

Spatial referencing describes the process of specifying locations of objects in space. According to Kuhn (2012), the concept of location describes a spatial *relationship* between objects rather than an intrinsic property, meaning that any description of location must include an arbitrary ground truth, e.g., another object, a geographic region, or a street network. Since location has an arbitrary nature and can be described in an infinite number of ways, the use of standardized language in the form of a *spatial reference system* (SRS) helps ensure that location descriptions remain consistent and repeatable (International Organization for Standardization 2002; Drewes 2009; Plag 2011). However, an SRS is purely theoretical. Realizing the SRS in the physical world requires associating the SRS with meticulously measured physical and tangible reference objects in the real world — such as monuments and calibration stations — that make up the *spatial reference frame* (SRF). Therefore, the SRS specifies the methods of describing location while the SRF puts the SRS into practice in the physical environment.

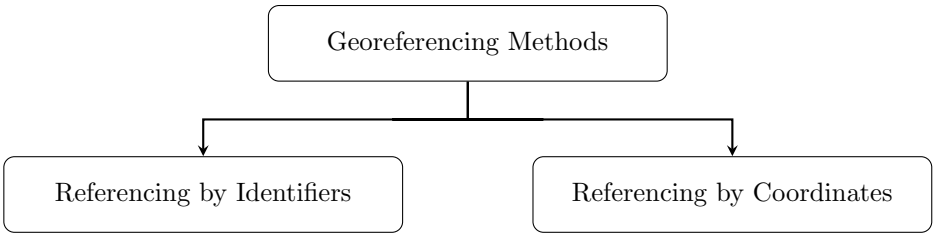


Figure 8.1: Georeferencing methods

In an Earth-based context, spatial referencing is also referred to as *georeferencing*. Since Earth is in a constant state of motion — rotation about its axis, orbit around the Sun, etc. — georeferencing uses Earth itself as the reference object, whether that involves its geocenter or arbitrary locations on its surface. This chapter provides a detailed review of the two general approaches to georeferencing: referencing by identifiers and referencing by coordinates (Figure 8.1; Bernhardsen 2002; International Organization for Standardization 2002).

8.1.1 Referencing by identifiers

Referencing by identifiers uses names, labels, or codes to identify locations in space and provides a high level abstraction that the average person can more easily understand. Examples include street addresses, postal codes, building numbers, and grids on a shopping mall floor plan. Use of identifiers often requires use of a standard list of locations, called a *gazetteer*, that can range from informal locally-maintained lists to internationally coordinated lists (United Nations Group of Experts on Geographical Names 2017). Advantages of identifiers over coordinates include greater simplicity, ease-of-use, and understandability. For instance, the identifier "Ellison Hall, Room 1720" provides a more understandable description than the coordinates "34°24′55.512″N 119°50′10.752″W." Disadvantages include ambiguities in describing physical locations and the possibility that multiple locations can have the same name.

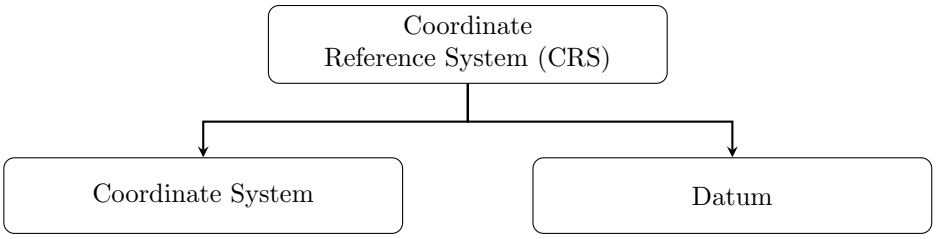


Figure 8.2: Components of a coordinate reference system

8.1.2 Referencing by coordinates

Referencing by coordinates uses numerical measurements from some given ground truth location and presents these measurements in the form of a list of coordinates[1]. Compared to identifiers, coordinates provide unambiguous and precise descriptions of locations in space. Familiar examples include the use of latitude and longitude coordinates for locations on Earth and (x, y) coordinates for referencing locations on graph paper. The term *coordinate reference system* (CRS) describes any SRS that uses coordinate referencing, which specifies use of a specific coordinate system and a unique datum[2]. (Junkins and Garrard 1998; Becker 2011; Torge and Müller 2012). The coordinate system specifies the methods of measurement while the datum specifies properties of the ground truth or starting point for measurements (Drewes et al. 2007; Drewes 2009). The remainder of this chapter will examine details of coordinate reference systems based on ISO and OGC conventions.

[1]According to International Organization for Standardization (2002) and Open Geospatial Consortium (2010), the singular term coordinate defines an individual measurement value in a coordinate tuple; the term coordinate tuple defines the ordered list of coordinates for a single location in space; and the term coordinate set defines a set of coordinate tuples for multiple locations. The size of a coordinate tuple corresponds to its spatial dimension — that is, a 1-tuple coordinate represents a location in one-dimensional space, 2-tuple coordinates represent a location in two-dimensional space, and 3-tuple coordinates represent a location in three-dimensional space. In colloquial usage, the general term coordinates is often used synonymously with coordinate tuples, such as in the expression, "What are the coordinates for Point X?".

[2]However, some organizations such as Esri use a slightly different approach to classifying reference systems and use the term "coordinate system" to describe a CRS.

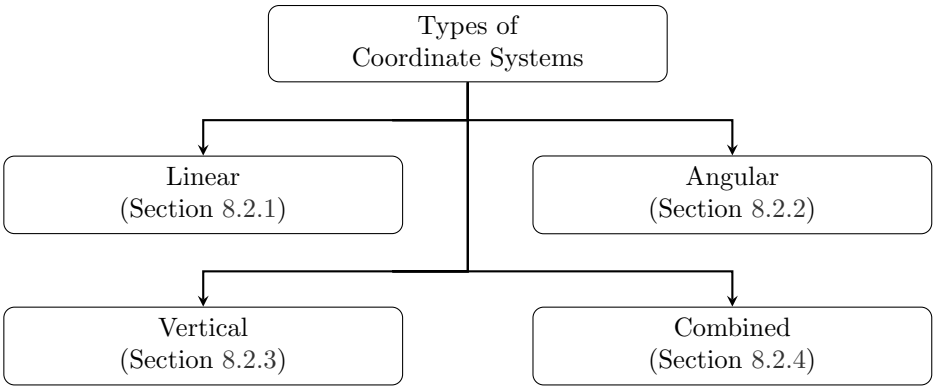


Figure 8.3: Types of coordinate systems

8.2 Coordinate systems

Coordinate systems — not to be confused with coordinate reference systems — define conventions for describing the measurement of points in space and can include elements such as the number of spatial dimensions, names given to each of those dimensions, units of measurement, and the directions and sequence of the coordinate axes (Open Geospatial Consortium 2010; Maling 1992; International Organization for Standardization 2002; Van Sickle 2010). All coordinate systems used in georeferencing fall under one of four categories: linear systems, angular systems, vertical systems, and combined systems. Linear systems measure distances along straight or curved lines, angular systems measure angles, vertical systems are unique to georeferencing and measure distances from some arbitrary surface such as mean sea level, and combined systems combine measurement conventions from the previous three systems.

8.2.1 Linear systems

Linear systems describe the locations of points along straight or curvilinear lines using linear units of measurement such as meters, feet, kilometers, and miles. A *linear coordinate system* defines a location along a one-dimensional line using a single-valued

coordinate (Open Geospatial Consortium 2010) — examples include the measurement of distances along highways or pipelines. *Cartesian and affine coordinate systems* define locations using measurements along two or three straight lines — axes — intersecting at an origin. Cartesian coordinate systems have all axes intersecting at right angles while affine coordinate systems do not. *Multi-dimensional curvilinear coordinate systems* also exist and are used in map projections (Bugayevskiy and Snyder 1995) but are beyond the scope of this paper; however, lines of latitude and longitude provide a convenient alternative which will be covered next.

8.2.2 Angular systems

Angular systems use angular units of measurement to specify locations. In georeferencing, these angular measurements originate at or near Earth's center and describe locations on or near Earth's surface using coordinate pairs matched to an ellipsoid — a geometric reference surface representing the surface of Earth. Three major considerations in using angular systems include the choice of ellipsoid parameters, the choice of the unit of measurement, and the method for determining central angles.

Ellipsoidal model of Earth

While Earth may appear as a perfect sphere when viewed from space, its actual shape has a slight elongation at the equator and slight flattening at the poles. The ellipsoid — a three-dimensional equivalent to the two-dimensional ellipse or oval — provides the simplest geometric and mathematical approximation of this shape. Its simplicity has made it a widely adopted model in georeferencing applications such as satellite positioning and navigation.

An ellipsoid consists of three ellipses that are mathematically defined by three axis values aligned with the three Cartesian coordinate planes. Ellipsoids used in georeferencing can have either two axis values (biaxial ellipsoid) or three axis values (triaxial ellipsoid). A triaxial ellipsoid provides the most accurate ellipsoidal representation of Earth since the planet actually has a slight flattening along the equatorial plane in addition to a flattening at the poles. However, triaxial ellipsoids are more mathematically complex making them more difficult to use and compute — an important consideration for time sensitive applications. As a result, most georeferencing applications use biaxial ellipsoids such as the WGS 84 ellipsoid used in the Global Positioning System (GPS).

Biaxial ellipsoids have just two axis values — a single value for the x- and y-axes on the equatorial plane and another smaller value for the z-axis running along the polar axis. Geometrically, a biaxial ellipsoid only models flattening at the poles while forming a perfect circle at the equator. Other names used in georeferencing for a biaxial ellipsoid include ellipsoid of revolution, spheroid, oblate ellipsoid of revolution, and oblate spheroid. The term spheroid implies an ellipsoid of revolution and the term oblate implies flattening along the axis of rotation.

Units of angular measurement

In georeferencing, angular measurements are commonly expressed in degrees, although grads and radians are also used (Zimmerman 1995; Van Sickle 2010). The conventional unit of degree (°or deg) is based on the sexagesimal system which divides a full circle into increments of 60: a full circle has 360 degrees (360°), one degree has 60 minutes (60′), and one minute has 60 seconds (60″), with fractions of a second expressed in decimal form. This system is often referred to as the degrees, minutes, seconds (DMS) system although other variations exist such as decimal degrees (DD) and degrees decimal minutes (DDM).

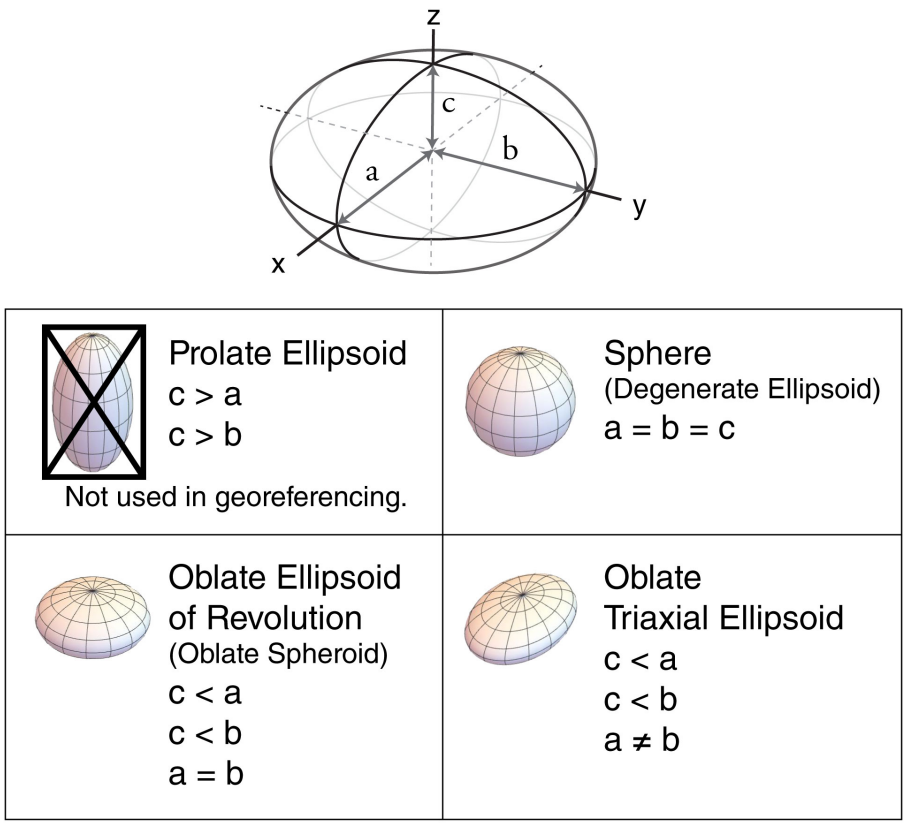


Figure 8.4: Various forms of the ellipsoid used to model Earth

The unit of grad (superscript ^{g} or grad) is based on the metric centesimal system which divides a full circle into increments of 100: a full circle has 400 grads (400^g), one grad has 100 centesimal minutes (100^c), and one centesimal minute has 100 centesimal records (100^{cc}), with decimal fractions afterwards (Zimmerman 1995; Hooijberg 1997; Van Sickle 2010). Note that a right angle (90°) has 100 centesimal minutes. Unlike the sexagesimal system, grads are usually expressed in metric decimal notation which facilitates addition and subtraction; for example, $80^g32^c45.2^{cc}$ can be directly converted to 80.32452^g.

Angular coordinates can also use the radian (rad) as a unit of measure. A radian is defined as the angle formed by dividing the radius of a circle by its circumference, in which a full circle has 2 rad. This property leads to the convenient conversion factors of

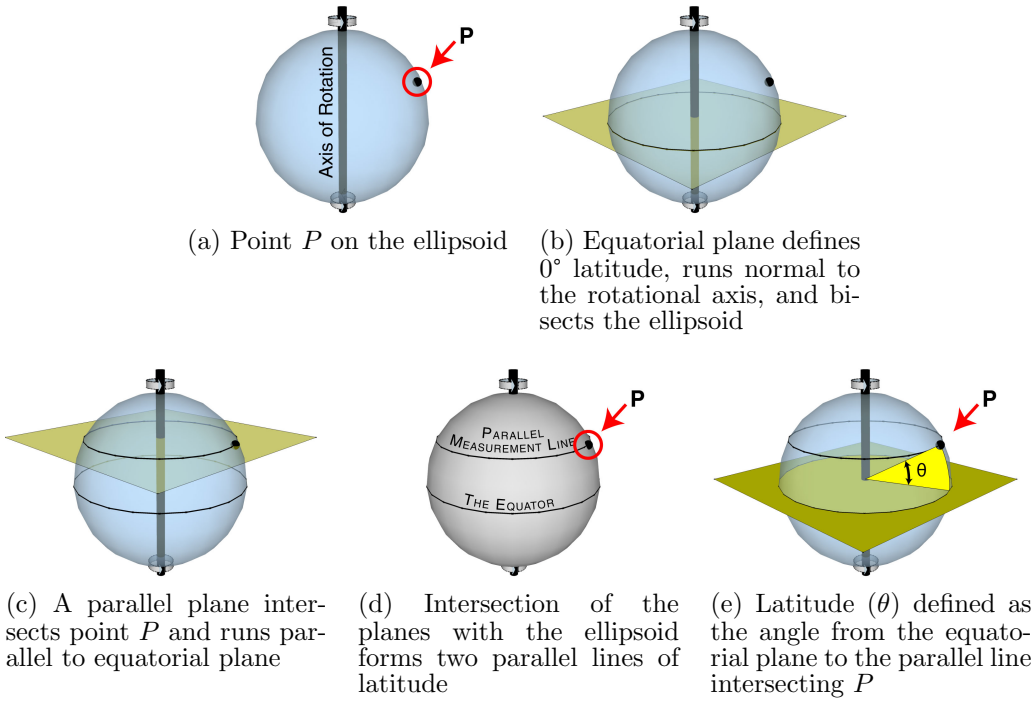
$$1^\circ = \frac{\pi}{180}\,\text{rad} \approx 0.0175\,\text{rad} \tag{8.1}$$

$$1\,\text{rad} = \frac{180}{\pi}\,\text{deg} \approx 57^\circ 17' 44.8'' \tag{8.2}$$

Central angles for defining geographic coordinates

Any location on an ellipsoid model of Earth's surface can be described using a pair of angular *geographic coordinates*, but these coordinates can take three different forms — geodetic, geocentric, or astronomic — depending on the method used to measure their angular values (Van Sickle 2010). Geodetic and geocentric coordinates measure angles from a point at or near the center of Earth, while astronomic coordinates indirectly derive central angles through celestial observation. Variations in ellipsoids, angular measurements, and datum parameters (covered in Section 8.3) means that geographic coordinates have non-unique values; for example, two identical sets of geographic coordinates can refer to different locations if the refer to different ellipsoids.

All geographic coordinates use the conventional form of latitude (θ) and longitude (λ) values, which measure angles from two arbitrary orthogonal planes that bisect the ellipsoid. Latitude describes the angle formed at or near the center of the ellipsoid between the equatorial plane (i.e., x-y axes plane that bisects the ellipsoid) and a curved line of latitude, which is itself formed by intersecting a parallel plane with the ellipsoid at the point of interest (see Figure 8.5). Lines of latitude run parallel to the equatorial plane and to other lines of latitude and are often referred to as *parallels*.



(a) Point P on the ellipsoid

(b) Equatorial plane defines 0° latitude, runs normal to the rotational axis, and bisects the ellipsoid

(c) A parallel plane intersects point P and runs parallel to equatorial plane

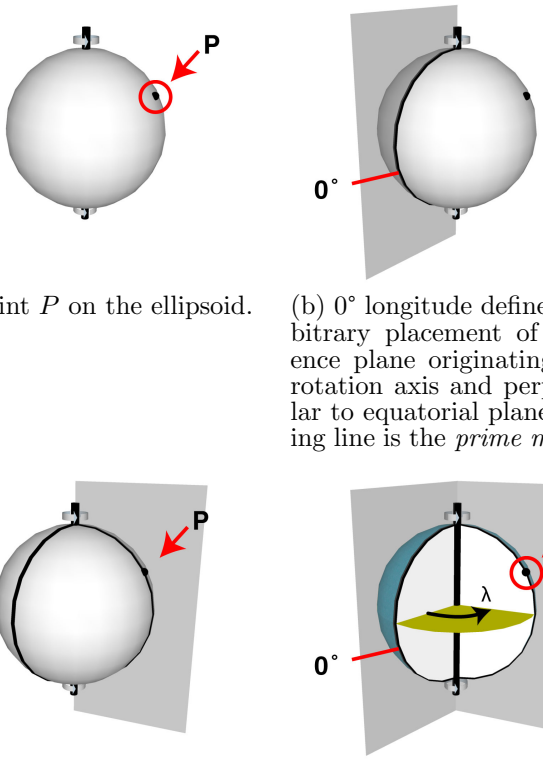
(d) Intersection of the planes with the ellipsoid forms two parallel lines of latitude

(e) Latitude (θ) defined as the angle from the equatorial plane to the parallel line intersecting P

Figure 8.5: Definition of latitude

Longitude describes the angle formed at or near the center of the ellipsoid between two planes that run perpendicular to the equatorial plane and pass through the polar axis (Figure 8.6d); the first plane intersects an arbitrary reference point (Figure 8.6c) and the second plane intersects the point of interest on the surface of the ellipsoid[3] (Figure 8.6b). Lines of longitude form semi-elliptical *meridian* lines that run from pole-to-pole along the surface of the ellipsoid. The reference meridian or prime meridian describes the the single meridian that coincides with the longitudinal reference plane as shown in Figure 8.6b.

[3]The intersection of the reference plane and equatorial plane also defines the direction of the x-axis.



(a) Point P on the ellipsoid.

(b) 0° longitude defined by arbitrary placement of a reference plane originating at the rotation axis and perpendicular to equatorial plane; resulting line is the *prime meridian*.

(c) A second plane originating at the rotation axis and intersecting point P forms a second meridian line.

(d) Longitude (λ) of P defined as the angle between the prime meridian and the meridian intersecting P measured along the equatorial plane.

Figure 8.6: Definition of longitude

Three forms of geographic coordinates

Geographic coordinates can take astronomic, geocentric, or geodetic forms depending on the method used to measure their angles; however, current usage of the term *geographic coordinates* generally implies the prevailing use of geodetic coordinates (Van Sickle 2010).

Astronomic coordinates The astronomic method represents the oldest form of measuring geographic coordinates; it uses Earth-based optical instruments to derive values of latitude and longitude based on the positions of celestial objects, such as the Sun, stars, and distant quasars. For instance, astronomic latitude can be determined by measuring

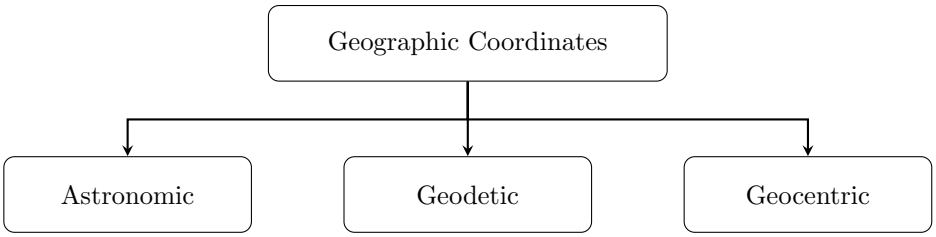
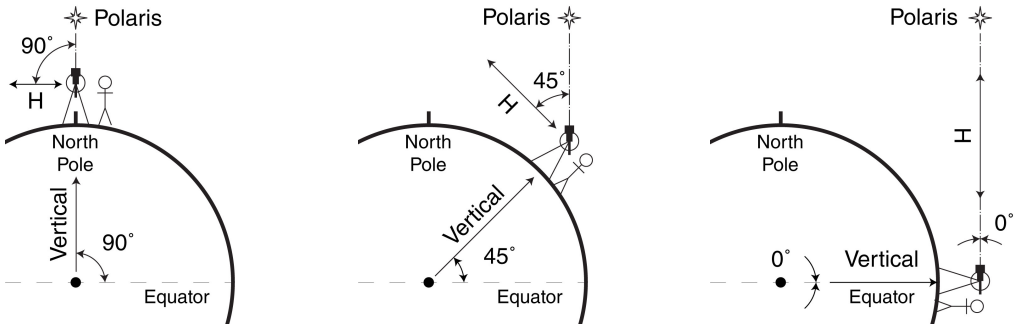


Figure 8.7: Three types of geographic coordinates

the angle between the line-of-sight to Polaris and the local horizontal plane, established through leveling of the sighting instrument. As a very distant star, Polaris will always point in the same northern direction regardless of the instrument's location.

Prior to the age of satellites, measuring longitude was a much more complicated task due to the arbitrary nature of reference meridians and Earth's constant rotation. In fact, the timing of Earth's rotation served as the basis for most measures of longitude since ancient times. The earliest surviving records showed that the ancient Greeks used lunar eclipses to measure the longitude difference between two different locations when coupled with local observation times (Howse 1980). This would remain the dominant method for calculating longitude until the advent of the mechanical clock during the Middle Ages. Invention of the marine chronometer, a high-accuracy clock, in the 1700s would further revolutionize the astronomical measurement of longitude (Howse 1980; Sobel 2007).



(a) At the North Pole (b) Between North Pole and equator (c) At the equator

Figure 8.8: Examples of measurements of astronomic latitude. H represents the horizontal direction, which runs orthogonal to the vertical direction of gravity.

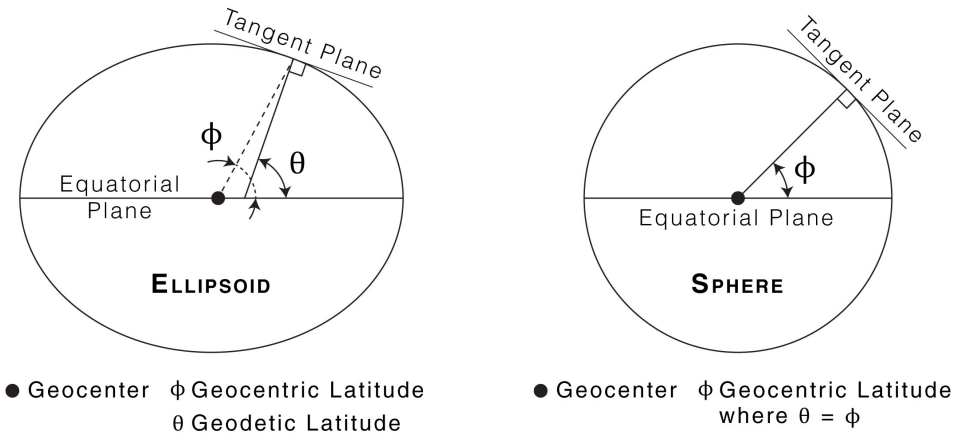
Since gravity determines the direction of the plumb line, and thus the directions of vertical and horizontal, astronomic coordinates relate more closely to the geoid than the ellipsoid. However, these gravity-based measurements had historically been used to approximate ellipsoid-based coordinates since no method existed for measuring the ellipsoid prior to the age of satellites.

Geodetic and geocentric coordinates Unlike astronomic coordinates, geodetic and geocentric coordinates represent true ellipsoid-based angles. For the common oblate spheroid, the commonly used geodetic latitude represents the angle formed between the equatorial plane and a line running normal to a tangent plane on the ellipsoid's surface, as designated by θ in Figure 8.9a. Note that the normal line may not intersect the geocenter, i.e., the center of the ellipsoid, when measuring this angle. This is different from the less-used geocentric latitude (ϕ in Figure 8.9a), which is measured from the geocenter. In the case of a degenerate ellipsoid or sphere (Iliffe and Lott 2008), the spherical shape guarantees that the normal line will always intersect the geocenter resulting in equivalent geocentric and geodetic angles as shown in Figure 8.9b[4].

Since the oblate spheroid and sphere have perfectly circular shapes along the equatorial plane, the normal line will always run along the same longitudinal plane (Figure 8.6c) resulting in equivalent geodetic and geocentric longitude values[5].

[4]Note that geocentric coordinates should not be confused with spherical coordinates used in mathematics, which have three values: polar angle, azimuthal angle, and distance from the center.

[5]In a triaxial ellipsoid, the combination of polar and equatorial flattening results in a complex situation where the normal vector may no longer coincide with the longitudinal plane.



(a) Ellipsoid

(b) Sphere, i.e., degenerate ellipsoid

Figure 8.9: Geodetic and geocentric latitudes

8.2.3 Vertical systems

Vertical measurements have their own separate coordinate system in the context of georeferencing, indicating the importance of height and depth for describing locations on Earth (Open Geospatial Consortium 2010). Height refers to the distance above or below Earth's surface pointing in the direction of gravity, which defines the local vertical direction, with negative values indicating distances below the surface (i.e., depth). Vertical coordinate systems commonly refer to a geoid datum (Section 8.3.2) as the baseline but can also refer to other surfaces determined through tides, barometric pressure, or other methods (Open Geospatial Consortium 2010). However, vertical coordinate systems do not apply to ellipsoids since these theoretical surfaces do not model gravity.

8.2.4 Combined systems

Linear, angular, and vertical coordinate systems can be combined to form combined coordinate systems. For instance, elements of linear and angular systems can be combined to create polar coordinate systems, cylindrical coordinate systems, and spherical coordi-

nate systems (Open Geospatial Consortium 2010). A commonly used combined system involves combining a 2D Cartesian coordinate system for planimetric measurements with a vertical coordinate system for height in the third dimension.

8.3 Datums

Coordinate systems simply define measurement conventions that have no basis in reality. Operationalizing a coordinate system requires setting up a relationship with the physical world by anchoring it to real locations with the use of a *datum* (plural form *datums*), which combine to form a unique *coordinate reference system* (Section 8.4). The term datum literally means "that which is given"; when used in a georeferencing context, a datum sets the initial conditions for taking measurements such as the origin, orientation, and scale of the coordinate system relative to Earth (Junkins and Garrard 1998; Drewes et al. 2007; Iliffe and Lott 2008; Drewes 2009; Becker 2011; Torge and Müller 2012). Open Geospatial Consortium (2010) lists four categories of datums: geodetic, vertical, engineering, and image. Of these, this chapter will only examine the three most commonly used types: geodetic datums, the geoid component of vertical datums, and engineering datums.

8.3.1 Geodetic datums

Section 8.2.2 previously covered the ellipsoid in the context of angular measurements. As a simplified model of Earth's surface, the ellipsoid also serves as a datum for two- and three-dimensional coordinate reference systems of Earth's surface. At a minimum, a geodetic datum must include the ellipsoid parameters and a prime meridian definition (Open Geospatial Consortium 2010), usually Greenwich based on the International Meridian Conference of 1884 (Howse 1980). For 2D horizontal systems that use angles,

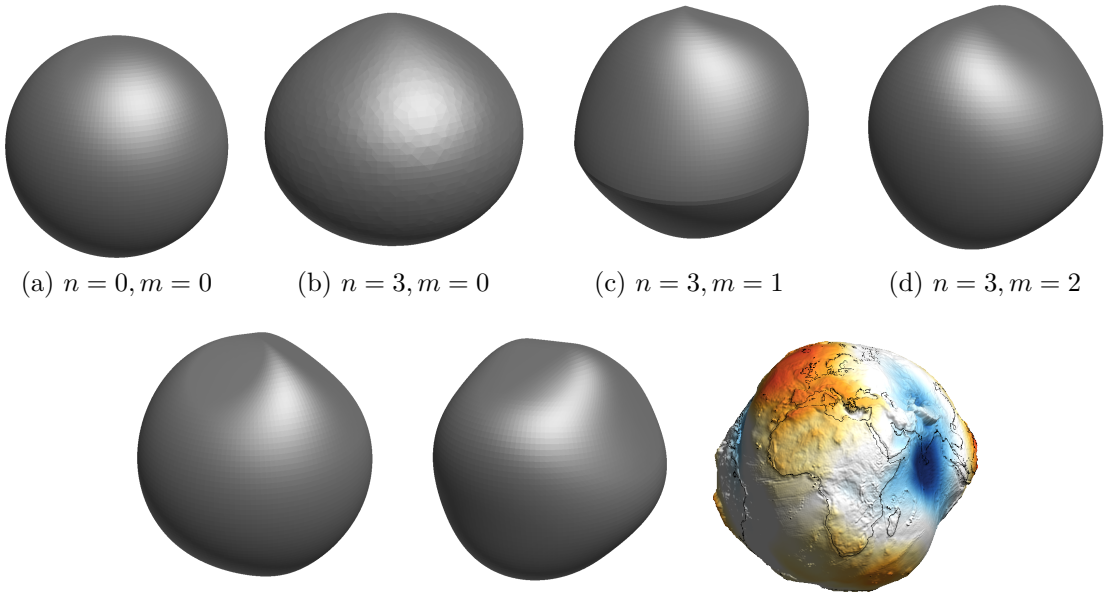
the prime meridian defines 0° longitude. For 3D Cartesian coordinate systems, the intersection of the prime meridian and equatorial plane defines the direction of the x-axis. All major geodetic datums also have definitions for anchor points and orientation relative to Earth, such as Earth's geocenter based on satellite measurements or locations on the surface (Open Geospatial Consortium 2010; Van Sickle 2010).

8.3.2 Geoids

Vertical coordinate systems (Section 8.2.3) require a datum surface from which to measure vertical heights and depth. Of the four types listed by Open Geospatial Consortium (2010), the geoid stands out as the most commonly used for vertical measurements. To understand the geoid, it helps to first understand the phenomena that a geoid attempts to model: water. Water — specifically, sea level — historically served as the most accessible and reliable method for establishing a vertical baseline from which to make elevation and depth measurements due to its natural leveling property, making sea level measurements the default baseline for many national surveys until recent times.

The advent of highly sensitive gravity measurement instruments — first, the gravimeter followed by the gravity gradiometer — revolutionized the vertical datum by augmenting highly varying observations of sea level with more reliable measurements of gravity, specifically the potential energy of gravity or *geopotential* at a predetermined sea level height. The ability to measure gravity anywhere in the world thus made it possible to directly determine sea level (via the geopotential) within continental areas, whereas in the past sea level readings had to be extended through manual surveys from coastal areas. A geoid, therefore, represents a mathematical model of this geopotential surface of Earth and can encompass varying geographic extents from local to global. Figure 8.10g provides an exaggerated version of the EGM2008 geoid for illustrative purposes.

Compared to the perfectly smooth ellipsoid, the undulating geoid represents the next best representation of the true shape of Earth, where the 'true" shape of Earth would be its topography. These undulations represent actual heights measured from an ellipsoid, so every geoid must have a single parent ellipsoid. As for the mathematics, a geoid uses high degrees and orders of spherical harmonics to provide a best fit model for empirical geopotential data as illustrated in Figure 8.10.



(a) $n = 0, m = 0$ (b) $n = 3, m = 0$ (c) $n = 3, m = 1$ (d) $n = 3, m = 2$

(e) $n = 3, m = 3$ (f) $n = 5, m = 2$ (g) EGM2008 Geoid $n = 2,159, m = 2,159$

Figure 8.10: Mathematical development of a geoid model using spherical harmonics with degree n and order m

Use of a geoid as the vertical datum thus presents three different heights: the geoid height, the ellipsoid height, and the orthometric height or elevation as illustrated in Figure 8.11. For a point A, a satellite positioning receiver will typically provide an ellipsoid height (h) converted from Earth centered Earth fixed (ECEF) Cartesian coordinates[6]. Ellipsoid heights have almost no practical value since the ellipsoid is imaginary, measurable

[6]http://www.ngs.noaa.gov/PUBS_LIB/gislis96.html

only from space. The geoid adjusts the ellipsoid height to a more tangible elevation value (H) using the calculated geoid height (N) for point A's location on Earth. Therefore, $H = h - N$.

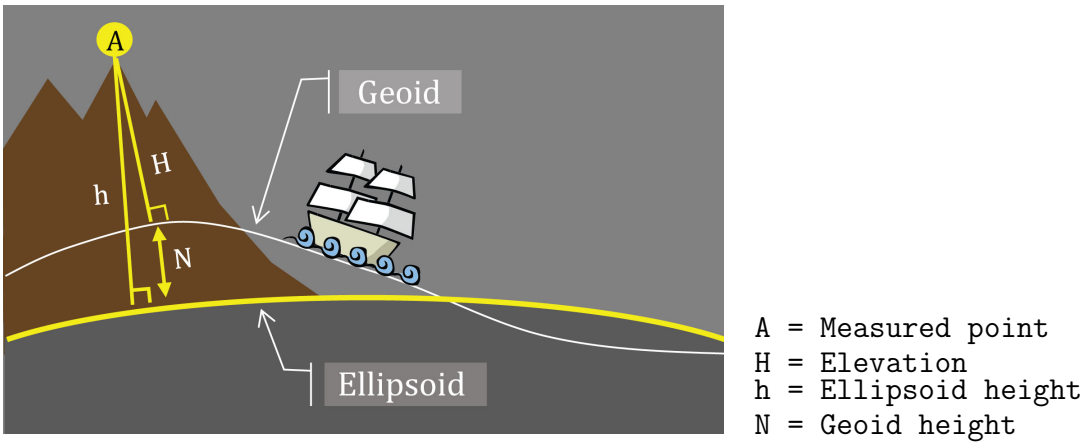


Figure 8.11: Relationship between different height measurements

It should be noted that a geoid represents a static mathematical model, not the actual geopotential surface itself, which is very complex and dynamic. For instance, studies by Rodell et al. (2006), Yeh et al. (2006), and Strassberg, Scanlon, and Rodell (2007) have shown that groundwater levels can cause gravity levels — and the geopotential surface — to change over short time periods.

8.3.3 Engineering datums

An engineering datum provides the initial conditions for an engineering coordinate reference system, also referred to as a local CRS, which may or may not have a direct relationship with locations on Earth (Open Geospatial Consortium 2010). Examples include CRSs for buildings and moving platforms such as ships and aircraft. Coordinate transformation or conversion operations can be used to relate an engineering CRSs to a geodetic or projected CRS, as described in the next section. The indoor mapping work explored in this dissertation uses engineering datums.

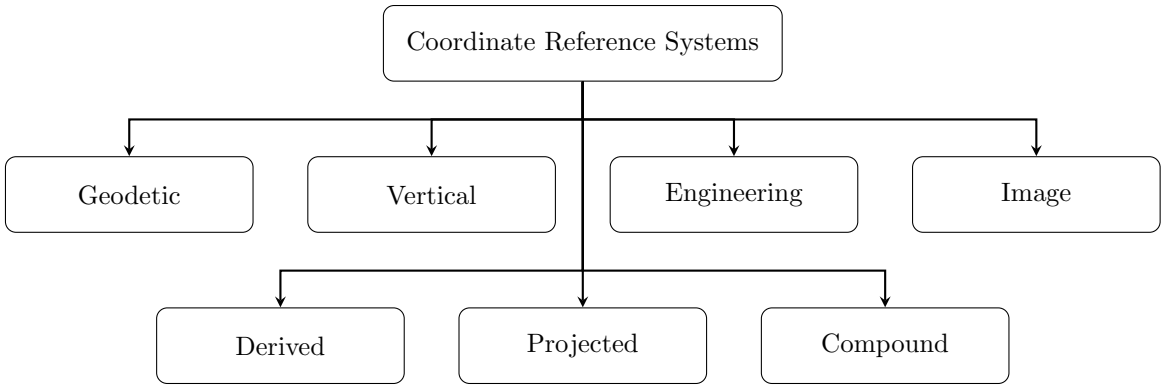


Figure 8.12: Three types of geographic coordinates

8.4 Coordinate reference systems

The pairing of a coordinate system with a datum results in a coordinate reference system (CRS), which often assumes the classification of its parent datum (Open Geospatial Consortium 2010) and provides the detailed *specifications* for taking measurements. For instance, a CRS based on a geodetic datum would be considered a geodetic CRS. In addition to the CRSs corresponding to the four main types of CRSs (i.e., geodetic, vertical, engineering, and image), the OGC specification has also added three other commonly used CRSs: derived, projected, and compound. The following sections provide further discussions on geodetic, engineering, derived, projected, and combined CRSs.

8.4.1 Geodetic CRSs

A geodetic CRS provides a two- or three-dimensional locational framework for Earth based on an ellipsoid included within a datum. The 2D form provides horizontal locations based on angular measurements of latitude and longitude, while the 3D form adds the height or depth relative to the ellipsoid. The two methods for anchoring a datum (Section 8.3.1) lead to an important distinction between two types of datums and, consequently, two types of geodetic CRSs: regional and global (Van Sickle 2010).

Regional

Prior to the age of satellites and gravimetry, the best way to determine ellipsoid parameters involved using extensive field measurements manually surveyed at a handful of locations on Earth. This had the effect of optimizing the resulting ellipsoids for specific geographic regions, but they were often adopted for other areas out of convenience. Establishing the datum involved anchoring the ellipsoid to a fixed arbitrary location on Earth's surface called the principal point and orienting it to another fixed arbitrary location using astronomic surveys. The principal point represented the initial point from which an entire network of survey control points could be built using the coordinate reference system. For instance, English geodesist Alexander R. Clarke developed the Clarke 1866 ellipsoid based on land surveys in England, France, South Africa, Peru, Lapland (northern Finland), Russia, and India, leaving out vast parts of Earth including North America and most of Asia. However, the U.S. government used it as the basis for the North American Datum of 1927 (NAD 27) CRS by anchoring it at a surveying point at Meades Ranch, Kansas, and orienting it towards another point in Waldo, Kansas. Since a regional CRS optimizes a datum for a certain geographic area, its use in other areas may result in significant errors; for example, the NAD 27 CRS should not be used for mapping Afghanistan.

Global

The advent of satellites and gravimetry made it possible to define geodetic datums at a global level through direct measurements of Earth's size, shape, gravity field, and rotation. The anchor for a global datum exists at Earth's geocenter with its orientation determined by an arbitrary axis of rotation and precise measurements for the prime

meridian. As a result, global geodetic datums — therefore, global geodetic CRSs — are optimized for global use.

8.4.2 Engineering CRSs

An engineering CRS provides a simplified approach to measuring locations within a small or local geographic region, where either the effects of Earth's curvature are considered negligible (Chapter 11) or have already been accounted for (e.g., projected CRSs). Open Geospatial Consortium (2010) divides engineering CRSs into Earth-fixed and moving platforms categories. An Earth-fixed engineering CRS uses a flat-Earth, projected CRS (Section 8.4.4) to define its datum and provides a flexible way to relate local measurements to Earth. An engineering CRS for a moving platform exists independent of Earth but can be related to Earth-based CRSs through time-based transformations; for this reason, OGC characterizes this as an intermediate CRS. This dissertation uses engineering CRSs to define indoor measurements as discussed in greater detail in Chapters 9 and 10.

8.4.3 Derived CRSs

As numerical measurements, coordinates can be manipulated to change from one CRS to another. Open Geospatial Consortium (2010) calls this process a coordinate operation, which has two variations: transformation and conversion. Coordinate transformation[7] involves the use of empirically derived data and a change in datum; the stochastic nature of the empirical data means that different versions of the same transformation can exist.

[7]This should not be confused with geometric transformation, which involves the actual mathematical operations.

In contrast, conversion involves either no parameters or explicitly given parameters that result in consistent changes. Derived CRSs involve the latter operation.

Derived CRSs are new CRSs created from the conversion of a parent or *base* CRS. A simple example is the use of unit conversions. For example, the North American Datum of 1983 (NAD 83) CRS uses units of meter. However, the imperial measurement system still prevails in many parts of the country, requiring NAD 83 (meters) to be converted to units of feet for many applications resulting in NAD 83 (feet).

8.4.4 Projected CRSs

Coordinate conversion can also convert geodetic CRSs from angular to linear units, such as latitude/longitude to units of meters in 2D Cartesian coordinates. Frequent use of this type of conversion led Open Geospatial Consortium (2010) to give this type of derived CRS its own classification: projected CRSs. Map projections is a widely covered topic and will not be discussed in this dissertation; further information can be found in numerous publications including Snyder (1987), Maling (1992), Slocum et al. (2010), and Clarke (2014).

Two widely used projected CRSs are the Universal Transverse Mercator (UTM) CRS and the family of State Plane Coordinate System (SPCS) CRSs used in the United States. As a projected CRS, UTM applies the Transverse Mercator map projection to 60 equal meridinal subdivisions of one of at least three *base* geodetic CRSs: WGS 84 (World Geodetic System of 1984), NAD 83, and NAD 27. Since each base geodetic CRS has a different datum, the different versions of UTM will show different coordinate values for the same location on Earth. As a result, UTM (WGS 84), UTM (NAD 83), and UTM (NAD 27) represent three different CRSs, though in common usage the designation of

the base CRS is often omitted, which can unintentionally result in conflation of the three different systems.

Another widely used projected CRS is the United States State Plane Coordinate System (SPCS), which actually represents a family of projected CRSs rather than a single CRS. The U.S. Coast and Geodetic Survey — now called the U.S. National Geodetic Survey (NGS) — established SPCS in the 1930s to provide surveyors and engineers with easy-to-use plane maps that would eliminate the need for complicated angular to linear conversions from the "lat/lon" of geodetic CRSs. Each state manages its own implementation of SPCS using a pre-defined national geodetic CRS and an error limit $\frac{1}{10000}$. California's implementation of SPCS — called the California Coordinate System (CCS) — divides the state into six zones, each with its own projected CRS. All six zones use the Lambert Conformal Conic projection, with each zone having different specifications for the central meridian and standard parallels thus resulting in six different CRSs for one base geodetic CRS. CCS can use NAD 27 or NAD 83 as the base geodetic CRS, resulting in CCS27 and CCS83 with 13[8] different projected CRSs for the state.

8.4.5 Compound CRSs

Combining different CRSs provides another way of forming a new CRS, most often used to form a 3D CRS from traditionally separated 2D horizontal and 1D vertical CRSs. However, compound CRSs can not be nested, i.e., used to form other compound CRSs. Compound CRSs that combine projected horizontal CRSs with vertical CRSs play an important role in indoor mapping since they allow the direct conversion of 3D indoor coordinates to 3D outdoor coordinates using the Cartesian coordinate system. Chapter 10 provides an example of this type of conversion.

[8]CCS27 had a separate zone for Los Angeles that later merged with Zone 5 in CCS83.

8.5 Coordinate reference frames

While the CRS provides theoretical specifications for measurements, the coordinate reference frame (CRF) realizes the CRS by associating key points in the CRS to actual physical markers in the real world. These markers can range from carefully placed survey markers to elaborate calibration stations that measure their positions with space-based observations. In the past, the system and the frame existed separately whereby the reference system defined the theoretical model while the reference frame represented its physical manifestation. Modern dynamic coordinate reference systems — such as the International Terrestrial Reference System (ITRS) — blur that distinction with their use of continuously monitored data from reference frame locations — such as the International Terrestrial Reference Frame (ITRF) — where the reference frame becomes an inherent part of the reference system (National Research Council 2010).

8.6 Alternative terminologies

This chapter reviewed one framework for spatial referencing as outlined by two international bodies — ISO and OGC. The widely referenced EPSG database of CRSs produced by the International Association of Oil and Gas Producers (IOGP) (2017) largely follows the conventions outlined by these two bodies. However, other organizations define the ISO/OGC terminology differently, which can lead to a significant amount of confusion.

Esri — maker of ArcGIS, the most widely-used family of geographic information systems (GIS) software — approaches spatial referencing in a different manner. It retains the difference between referencing by identifiers and referencing by coordinates, calling the former *geocoding* and the latter *spatial referencing* (as opposed to the ISO/OGC

definition). Geocoding generally follows the ISO/OGC concept but its implementation of spatial referencing (Esri definition) has a more restrictive definition as shown in Figure 8.13. First, Esri refers to the ISO/OGC coordinate *reference* system as a coordinate system and it omits classifying (ISO/OGC) coordinate system specifications, only listing the unit of measurement as a required property of an (Esri) coordinate system. Second, it subsumes ellipsoid heights under the (Esri) vertical coordinate system and treats the prime meridian as a property separate from the GCS's datum. Finally, Esri's GCS represents a subset of an ISO/OGC geodetic CRS; namely, it only includes the horizontal component (moving ellipsoid height to the vertical coordinate system) and it only covers geographic coordinates (i.e., latitude and longitude) but not Cartesian coordinates. In summary, the main differences between Esri and ISO/OGC include use of the term "coordinate system" and "spatial referencing" to represent the ISO/OGC concept of a CRS and a slightly different definition of the datum. However, Esri's approach to defining the datum is not unusual.

8.7 Conclusion

This chapter reviewed the multitude of ways for referencing locations in space either through natural language identifiers or measured coordinates. On the one hand, well-established conventions for coordinate referencing makes it possible to unambiguously describe nearly any location on Earth. Yet on the other hand, this flexibility has resulted in a dizzying array of identifiers and coordinate reference systems, with the widely-referenced EPSG database identifying over 5,700 CRSs in 2017. Additionally, the ISO/OGC standards are not universally followed, which can lead to identical terminology having subtly different meanings such as with Esri's georeferencing framework. An understanding of the international standards coupled with awareness of how different

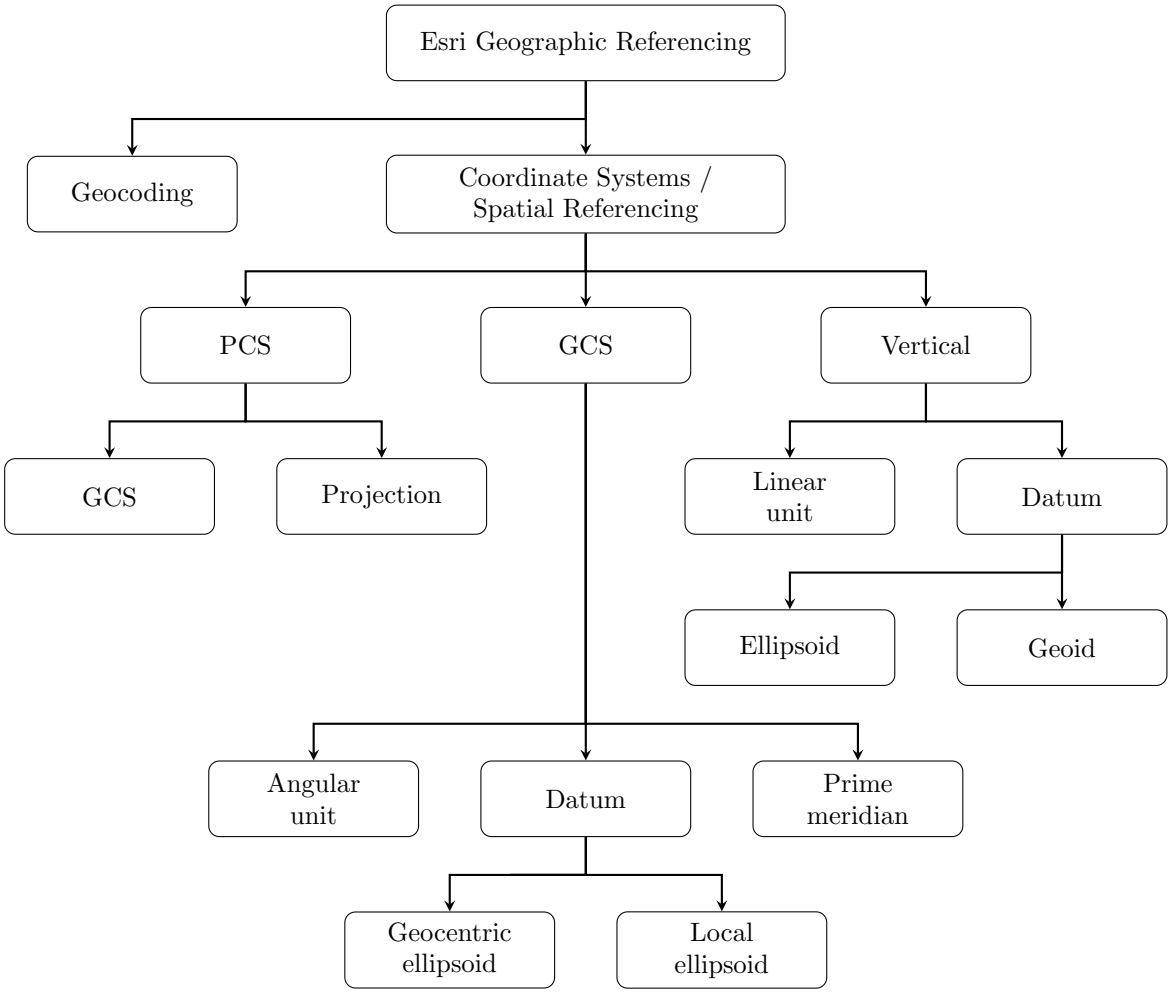


Figure 8.13: Esri's approach to geographic referencing

implementations can deviate from these standards can help prevent misunderstandings and ensure the effective communication of spatial locations.

Chapter 9

Grid-Based Indoor Coordinate Referencing

An abbreviated version of this chapter will be published in the Location Based Services (LBS) 2018 conference proceedings (Chen 2018).

9.1 Introduction

Cartographer E. H. Thompson (1973) once remarked, "Grids are simply sets of squares and... a square is a square is a square." Yet the simple grid has made map locations easier to read, use, and communicate for a wide range of applications that include shopping malls, mass transit systems, air spaces, military operations, and amusement parks. The growing adoption of 3D indoor maps and indoor navigation technology presents an opportunity for extending the benefits of map grids indoors, beyond the simple 2D grids commonly found in shopping malls (Zlatanova et al. 2013). While some may argue that computerized mapping has made map grids and the ideas they represent obsolete, over-reliance on automated methods may obscure the benefits of grids and

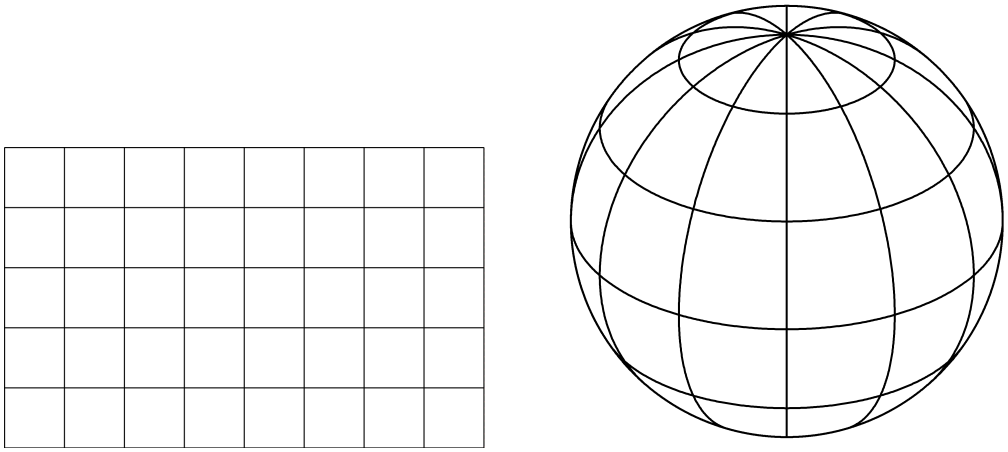
under-utilize the inherent spatial reasoning abilities of map users (National Research Council 2006). Map grids of indoor spaces do not compete with computerized maps; instead, they can serve as a complementary technique for enhancing their readability as well as providing an extra layer of robustness. This chapter presents a general framework for developing a coordinates-based grid reference system for indoor spaces that can also serve as a starting point for other types of indoor gridding.

9.2 Background

9.2.1 Defining the Map Grid

In its most basic form, a map grid represents a visual aid consisting of a set of orthogonal lines overlaid on a map–usually a projected map–to partition it at regular intervals. In a cartographic context, the term *map grid* implies use of straight orthogonal lines in Euclidean space delineated using linear measurements (e.g., meters, feet, etc.); in contrast, the term *graticule* represents a special case of a grid. . . one formed by three-dimensional arcs circumscribing a round Earth with *meridian* lines converging at the poles and *parallel* lines running locally-orthogonal to meridians and parallel with the equator, both partitioned at regular intervals of angular units (e.g., degrees of latitude and longitude) (Snyder 1987). This chapter addresses the former, the conventional map grid, applied to indoor spaces.

Grid references generally follow the two broad spatial referencing conventions of *referencing by identifiers* (i.e., geocoding) or *referencing by coordinates* and can even use a combination of both; its design limited only by the imagination and sound communication practice (International Organization for Standardization 2002, 2003).



(a) Grid (b) Graticule

Figure 9.1: Grid and graticule

9.2.2 Historical Perspective

Grids have existed since ancient times as an aid to map reading. The ancient Greeks were known to have used gridded maps, with Dicaearchus credited with developing the first at around the third century B.C., followed by Eratosthenes and Hipparchus five decades later, and Ptolemy in the second century A.D. (Royster 2002). However, these grids represented primitive graticules since they were based on astronomical observations. Cartographer Norman J. W. Thrower speculated that Chinese cartographers may have also employed grids no later than the third century B.C. through interpretation of historical written records, but more concrete evidence points to the time of Pei Hsiu and Zhang Heng, Chinese cartographers who formally introduced a system of grids at around the time of Ptolemy in second century A.D. (Cheng-siang 1978; Yee 1994b; Thrower 2008). As with the Greek grids, these Chinese grid systems also appeared to represent early graticules, some of which may have combined astronomical observations with linear terrestrial measurements (Yee 1994a).

The earliest known gridded map from China comes from the *Yuji Tu* (map of the tracks of Yu), a stone-carved map from c.1136 A.D., which had a square grid system which "seems to have been superimposed arbitrarily on a given area of interest" (Yee 1994b). However, the introduction of gridded systems for the planning of ancient cities[1] points to the use of graticules and even actual map grids long before the earliest discovered map grid artifact (McIntosh 2008; Fairman 1949).

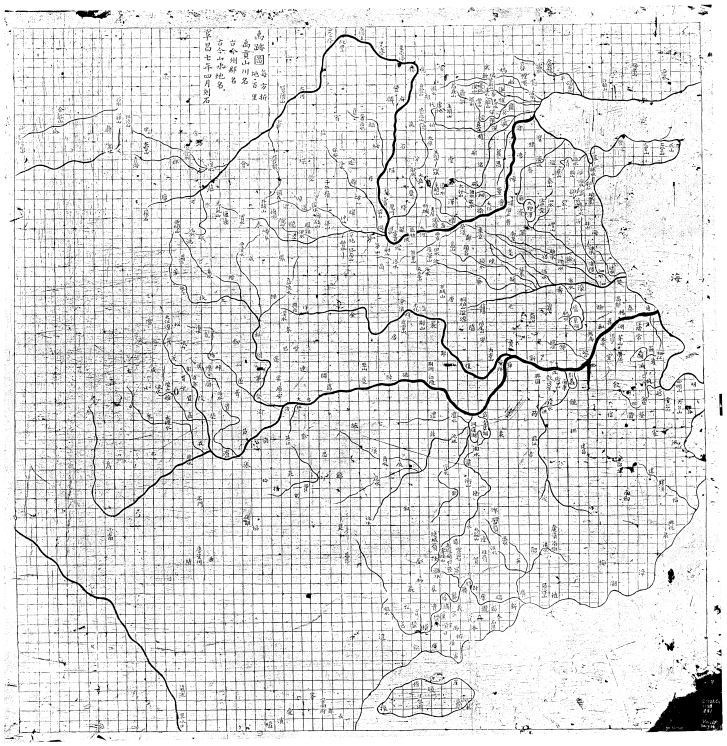


Figure 9.2: Negative image of an oil rubbing of the Yuji Tu[2]
from China, c.1136 A.D.

Map grids saw increasing use beginning in 18th century Europe with notable applications in military operations, and they featured prominently during World Wars I and II (Liebenberg, Demhardt, and Vervust 2016). Two grid systems that developed in

[1]For example, in the Indus Valley (26th century B.C.), Egypt (25th century B.C.), Babylon (17th century B.C.), China (15th century B.C.), Greece (5th century B.C.), and Rome (4th century B.C.)

[2]Library of Congress (1136)

the United States at the time of the two world wars include the Universal Transverse Mercator (UTM) system and the Military Grid Reference System (MGRS). These two systems, with formalized grid specifications, are still in use today and serve as models for the indoor grid referencing framework presented in this chapter.

9.2.3 Properties of UTM and MGRS

Three notable properties of UTM and MGRS grids–as well as most cartographic grids–include the use of *eastings* and *northings* to describe coordinates, the use of a *false origin* to ensure that all grid coordinates have positive values, and the designation of a central meridian for each grid where it attaches to the projected map at the locations of least distortion (Snyder 1987; Buckner 1995; U.S. Geological Survey 2001). Eastings and northings follow the Cartesian coordinate convention of measuring from left-to-right along the x-axis and bottom-to-top along the y-axis, respectively. The false origin is established by translating the intersection of the coordinate axis from the true origin (Cartesian coordinates $(0, 0)$) to a point located on or outside of the grid boundary in the west and south directions; this results in all grid coordinates having positive values (Maling 1992).

The UTM system divides the surface of Earth into 60 projected map *zones* oriented from south-to-north–each with a width of 6° longitude, bounded by latitudes 80°S and 84°N, and numbered from 1 to 60, moving eastward from 180°W. Over each projected zone is overlaid an orthogonal grid centered on its central meridian (at 3° from the boundary) with a grid datum shift of 500,000 m to the west measured from the central meridian, resulting in a false easting coordinate of 500,000 m at the central meridian. Locations in the northern hemisphere have a false northing of 0 m coinciding with the equator, while southern hemisphere locations have a false northing value of 10,000,000 m at the equator.

In 2014, the U.S. National Geospatial-Intelligence Agency (NGA) formally incorporated MGRS zone bands (described in the next paragraph) into the UTM grid system (National Geospatial-Intelligence Agency 2014b).

The MGRS system adapts the UTM system for military use by further dividing each UTM zone into 20 bands labeled C through X (from south to north) at 8° intervals of latitude, with the last band (X) having a 12° interval. This subdivision results in *grid zones*, each with a unique grid zone designator (GZD) consisting of the zone number and band, e.g., 34S (Ackeret et al. 1990). (Bands A/B and Y/Z are reserved for the southern and northern polar regions, respectively, which use a separate polar map projection.) Grid zones represent part of the projected map–i.e., they are not map grids–over which a series of 100,000 meter *grid squares* are overlaid, each with a unique *100,000 m square identification*, and no false origin. Grid squares are sequentially stacked longitudinally from the equator and placed side-by-side starting from the central meridian–with one square on each side of the meridian–so that they effectively subdivide UTM grids into grid squares.

9.2.4 Expressing UTM and MGRS Grid Coordinates

Grids for UTM and MGRS use both identifiers and measured coordinates for referencing. UTM uses the following form for expressing coordinates

```
<GZD> <easting>mE <northing>mN
```

while MGRS uses this form

```
<GZD><square identification><MGRS easting><MGRS northing>
```

where an MGRS coordinate tuple can include as little as the GZD or as much as 5 digits each for the easting and northing to accommodate different precision requirements. For instance, UTM 11S 238488mE 3811890mN is equivalent to MGRS 11SKU3848811890 at 1-meter precision. The MGRS coordinates can also be stated as 11SKU (100,000 meter precision), 11SKU31 taken from **11SKU3**8488**1**1890 (10,000 meter precision), etc.

Prior to standardization of UTM and MGRS expressions by NGA in 2014, UTM coordinates could also be expressed using hemisphere designations instead of bands. This often led to confusion between N (northern) and S (southern) hemispheres with MGRS bands N and S. For example, the legacy expression of 34S 650000mE 6500000mN refered to a location in the southern hemisphere but the "S" designation could also have been interpreted as MGRS grid zone S, which exists in the northern hemisphere. While NGA's 2014 standardization addressed this problem, legacy UTM coordinates still exist and some coordinate conversion software continue to use hemisphere designations; these should be used with caution.

9.2.5 Principal Digits

Principal digits can improve the readability of large UTM coordinate values by graphically emphasizing the digit or digits associated with the published grid spacing interval (Buckner 1995; National Geospatial-Intelligence Agency 2014b). On printed maps, full easting and northing values of grid lines are printed at least once per map, with all other grid lines omitting trailing zeros. For instance, a grid with a 1,000 or 10,000 meter interval would have principal digits at the fourth and fifth digits, e.g., $^{3}50^{000}$mE for an easting of 350,000 mE. Subsequent grid labels would be expressed as $^{3}51$mE for 351,000 mE at 1,000 m intervals or $^{3}6$mE for 360,000mE at 10,000 m intervals.

9.2.6 Reference System versus Reference Frame

Reference systems describe formal conventions used for describing locations in space and largely consist of *referencing by identifiers* and *referencing by coordinates*, as codified by the International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC) (International Organization for Standardization 2002, 2003; Open Geospatial Consortium 2010).

Referencing by identifier, also called *geocoding*, uses labels or codes for describing location–e.g., street addresses, place names, or building number–which can act as surrogates for coordinates. In contrast, referencing by coordinates uses systems of measurements to describe location; these systems, called *coordinate reference systems (CRSs)*, consist of a starting point parameter (datum) and measurement conventions (coordinate system). ISO and OGC identify three types of CRSs based on the datum type: geodetic, vertical, and engineering. Maps of indoor spaces nearly always use engineering CRSs with the Cartesian coordinate system, but may require use of all three types for integrating with outdoor coordinates.

Since a coordinate reference system exists only in theory, its realization in the physical world takes the form of a *reference frame*. A reference frame consists of a network of carefully placed fiducial or survey markers that conform to CRS specifications, which then serve as the basis for other physical measurements (Junkins and Garrard 1998).

9.3 Indoor Grid Referencing

Indoor spaces and their buildings provide ideal environments for the application of grids due to their native use of the Cartesian coordinate system. Deliberate use of a gridding convention for buildings and indoor spaces has several advantages:

- improved consistency & predictability in the use of indoor locational data;
- improved usability due to predictable measurement patterns among different buildings;
- robustness of measurements through the use of all-positive coordinate values and the ability to accommodate future expansion of the mapped area;
- ability to directly use machine-readable coordinate values in easy-to-read form for published maps and navigation aids (e.g., landmarks);
- improved robustness of mapping systems by providing a human-readable backup device during system failures, e.g., loss of indoor localization signal or catastrophic failure of the electronic map; and
- provides a common template for building other reference systems.

One advantage of grid referencing over using arbitrary unshifted coordinate values is the ability to use the same numerical values for machine processing and human use. While machines can accurately read and process numbers of any type, humans are susceptible to making errors in reading numbers and performing calculations. For instance, unshifted coordinates based on the true origin $(0, 0, 0)$ can results in negative values that human users can inadvertently misinterpret as positive. Land surveyors have long recognized this shortcoming, which explains why surveying maps–such as those based on the U.S. State Plane Coordinate Systems–use all-positive false coordinates.

Rather than attempting to change human behavior, changing the design of the coordinate environment can lead to fewer errors and more accurate results. This is an important feature of robustness–a design philosophy that seeks to minimize the impact of environmental variation without eliminating the variations themselves (Taguchi and Clausing 1990). Since machines can read and process numbers of any type, why not just use the same format that humans use?

9.3.1 Defining the Indoor Coordinate Reference System

The indoor grid uses a coordinate reference system (CRS) that includes a base CRS and a coordinate conversion to implement the false origin.

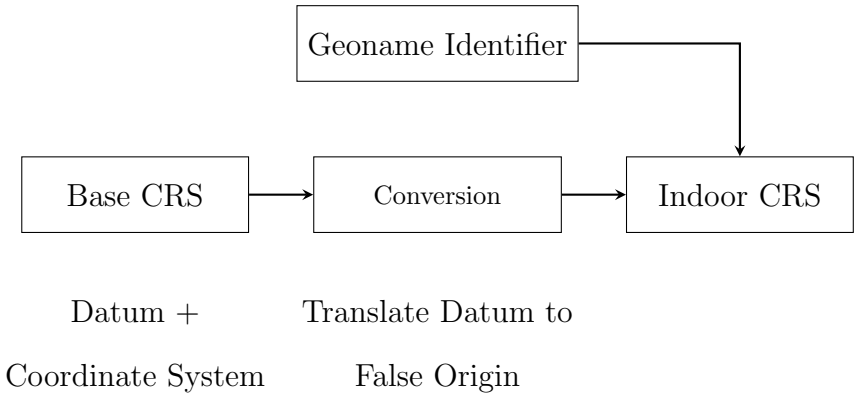


Figure 9.3: Process for establishing an indoor grid coordinate reference system

Datum

The indoor grid natively uses an engineering datum with a building-level scope and two anchor points. The scope of an indoor space should be confined to the boundaries of its surrounding structure in order to provide a predictable partitioning of space from one structure to the next and to facilitate transmission of CRS data for automated positioning applications. The datum anchor, though optional in the ISO and OGC standards, is required for indoor grid referencing and consists of two points, assuming that a building model exists: the principal point ($\mathbf{b}_1$) and the orientation point ($\mathbf{b}_2$). A line from $\mathbf{b}_1$ to $\mathbf{b}_2$ establishes the direction of local north from which the local east and up directions can be derived using the building model. Assuming that the indoor space and the building are perfectly level (i.e., the vertical direction runs orthogonal to ground surfaces), these points can be located anywhere in a building and do not have to be on the same horizontal plane since that information can be derived from the building model;

however, they should coincide with salient physical building features, e.g., easy-to-access fiducial markers, physical survey points, or corner points.

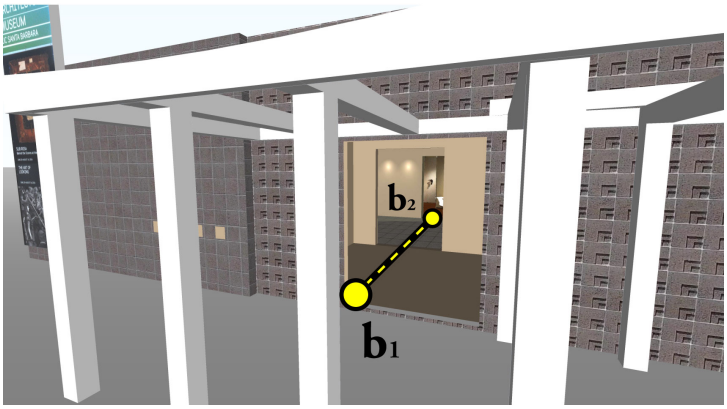


Figure 9.4: Example of points $\mathbf{b}_1$ and $\mathbf{b}_2$

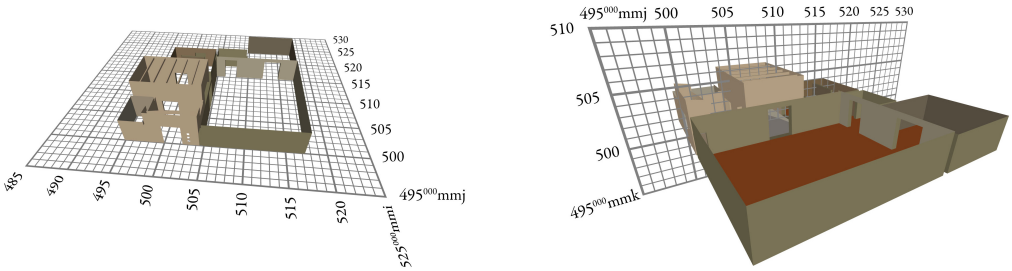
Coordinate System

The coordinate system for indoor grids consists of three orthogonal Cartesian coordinate axes corresponding to the three dimensions of indoor space. Here, the axes i, j, and k correspond to the local east, local north, and up directions, respectively, using any standard unit of measure for all three axes. The recommended unit of measure is the metric millimeter due to its simple divisibility by powers of ten.

Coordinate Conversion

Matching the datum with the coordinate system forms the grid's *base CRS* with an origin at $(0, 0, 0)$ corresponding to the principal anchor point $\mathbf{b}_1$. Translating $\mathbf{b}_1$ to a *false origin* ensures that all coordinates have positive values. This is done through coordinate conversion with a datum shift in the local west, local south, and down directions. The west and south datum shift should have identical values for consistency, while the shift downward can have a different value.

While the grid's datum can be shifted by any set of values to produce the false origin, using increments of 100,000 mm can help with readability of displayed values. A datum shift of 500,000 mm to the west and south should suffice for most indoor spaces that have a maximum horizontal dimension of between 500 m to 1 km, depending on the location of $\mathbf{b}_1$.



(a) Horizontal grid (b) Vertical grid

Figure 9.5: Indoor grid

Global Context

An indoor grid is first and foremost associated with a building structure instead of a single geodetic point, which has ramifications for indoor-outdoor continuity, e.g., navigation and location-based services. For instance, a geodetic reference point provides insufficient information for identifying the boundaries of an underground corridor or shopping mall with a span of one kilometer–information that can trigger the switching of localization technologies as well as quickly establishing a position within the indoor grid.

One solution involves using standardized geographic names, or *geonames*, to identify and associate a building with an indoor grid. The geoname can thus be a part of the indoor grid's specification, which itself can be related to a building's geoname. This allows a navigation system to compare its current location relative to a building's outer

boundary (accessed through an online database of building models tied to geonames) and to switch to indoor navigation when indoors, i.e., indoor localization using the building's indoor grid and indoor map.

Several nations already have national-level databases of geographic names, or *gazetteers*, and the United Nations Group of Experts on Geographical Names (UNGEGN) continues to work with U.N. member nations to develop and coordinate the standardization of geographic names at a worldwide level (United Nations Group of Experts on Geographical Names 2017; Kerfoot 2009). The United States Board on Geographic Names under the United States Geological Survey maintains geographic names for the United States.

Coordinate Reference System

The coordinate reference system for an indoor grid thus consists of the base CRS translated to the false easting, northing, and elevation (Figure 9.3) and associated with a building identifier. The operations used for establishing the indoor CRS can also be used to transform the model from its native model space to grid space as described in section 9.3.2, a useful process when using the model for computational or cartographic purposes. Expression of a grid coordinate triple takes the form

```
<building ID> <local easting>mmi <local northing>mmj <local elevation>mmk
```

9.3.2 Aligning the Model with the Grid

Alignment of the building model with the grid involves translating and rotating the model to match the grid's CRS. This involves translating the model to the origin $(0, 0, 0)$; rotating it to match the grid's orientation; and translating it to the false easting, northing,

and elevation coordinates of the grid. Let $\mathbf{b}_1$ and $\mathbf{b}_2$ represent the anchor points on the model.

$$\mathbf{b}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \mathbf{b}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \tag{9.1}$$

$$\mathbf{v}_{H12} = \mathbf{b}_{H2} - \mathbf{b}_{H1} = \begin{bmatrix} x_2 \\ y_2 \\ 0 \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \\ 0 \end{bmatrix} \tag{9.2}$$

$$\hat{\mathbf{v}}_{H12} = \frac{\mathbf{v}_{H12}}{\|\mathbf{v}_{H12}\|} = \begin{bmatrix} \hat{v}_x \\ \hat{v}_y \\ 0 \end{bmatrix} \tag{9.3}$$

where H denotes the projection of the orientation vector onto the horizontal plane and $\hat{\mathbf{v}}_{H12}$ denotes the unit vector of that projection. Transformations at the origin consist of a translation ($\mathbf{T}_1$) and rotation ($\mathbf{R}_z$)

$$\mathbf{t}_1 = -\mathbf{b}_1 = \begin{bmatrix} -x_1 \\ -y_1 \\ -z_1 \end{bmatrix} \text{ (or) } \mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.4}$$

$$\mathbf{r}_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{v}_y & -\hat{v}_x & 0 \\ \hat{v}_x & \hat{v}_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ (or) } \mathbf{R}_z = \begin{bmatrix} \hat{v}_y & -\hat{v}_x & 0 & 0 \\ \hat{v}_x & \hat{v}_y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.5}$$

$$\mathbf{M}_1 = \mathbf{R}_z \cdot \mathbf{T}_1 \tag{9.6}$$

where $\mathbf{t}_1$ and $\mathbf{r}_z$ represent the 3x3 transformation matrices, $\mathbf{T}_1$ and $\mathbf{R}_z$ the homogeneous 4x4 matrices, and $\mathbf{M}_1$ the master homogeneous matrix that can be applied to the model to match the model to the grid at the origin (see Figure 10.3).

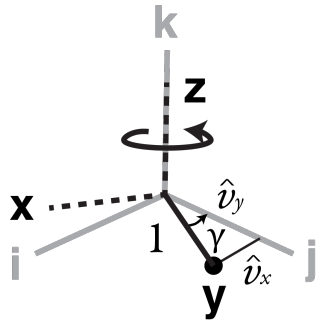


Figure 9.6: Rotation about vertical axis at origin (see Equation 9.6)

Once aligned at the origin, the model can be translated to the grid's location based on the false origin using $\mathbf{T}_2$.

$$\mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 & \langle \text{false easting of } \mathbf{b}_1 \rangle \\ 0 & 1 & 0 & \langle \text{false northing of } \mathbf{b}_1 \rangle \\ 0 & 0 & 1 & \langle \text{false elevation of } \mathbf{b}_1 \rangle \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.7}$$

Assembling these equations results in a master 4x4 transformation matrix ($\mathbf{M}_2$) that can be applied directly to the model.

$$\mathbf{M}_2 = \mathbf{T}_2 \cdot \mathbf{R}_z \cdot \mathbf{T}_1 \tag{9.8}$$

As with the datum, this simplified approach assumes that the model and the actual building are perfectly level, i.e., the vertical direction runs orthogonal to the building's ground surfaces.

9.3.3 Establishing the Reference Frame

Having a detailed building model can make establishing an indoor reference frame much easier than establishing one for a geodetic CRS. The principal and orientation anchor points serve as the most accurate and reliable points for building a network of other reference points. In adverse situations that prevent reliable access to the anchor points (such as over multiple floors), intermediate points can be established based on their locations on the model; future adjustments can be made and propagated to the rest of the network.

When the indoor grid reference frame serves as the basis for location beacons, indoor positions will correspond to false origin grid positions that can be further tied to visual

grid-based landmarks. This unified approach to localization–i.e., using landmarks to augment electronic maps–can enhance the effectiveness of indoor positioning and navigation systems.

9.3.4 Indoor Grid Specifications

Using standard formats for presenting indoor grid specifications has major two advantages. First, it institutes a rigorous convention for capturing details critical to building a CRS and makes it uniform across all indoor spaces. Uniformity supports predictability and can improve usability and levels of adoption. Second, a standard format allows specifications to be machine-readable–an important feature for automated indoor mapping, positioning, and navigation systems.

Three related CRS standards include ISO Standard 19111 *Geographic information–Spatial referencing by coordinates*, OGC Abstract Specification 08-015r2 *Topic 2: Spatial referencing by coordinates*, and EPSG Publication 373-7-1 *Using the EPSG Geodetic Parameter Dataset*, with the latter two standards conforming to the ISO standard. OGC has relevance since it guides development of geospatial information across a wide cross section of practices that include "companies, NGOs, research organizations, agencies and universities." OGC oversees standards such as CityGML, IndoorGML, and KML. EPSG has relevance since it maintains a publicly accessible database containing thousands of CRSs that are incorporated into geospatial software, such as Esri ArcGIS, for coordinate transformations. This initial indoor grid proposal was developed with standardization in mind and largely conforms to the baseline ISO standard. Table 9.1 in the example below provides the ISO parameters for a sample indoor grid CRS.

9.4 Applications of the Indoor Grid

9.4.1 Expressing Indoor Grid Coordinates

While the use of all-positive numbers eliminates one potential source of human errors (i.e., misinterpreting positive and negative values), the high number of digits in indoor grid coordinates can result in values being misread by factors of 10, i.e., accidentally dropping trailing zeros, such as misreading 500,000 as 50,000. Applying the concept of principal digits to indoor grids can mitigate this problem.

For instance, indoor grid lines with false northings of 500000mmj and 50000mmj can be shown on a map or physical landmark display as 500^{000} and 50^{000} using principal digits in the fourth to sixth digits.

9.4.2 Indoor-Outdoor Transformations

One advantage of using ISO-based standards is the availability of established processes for coordinate conversions and transformations. Transforming between indoor coordinates and geodetic coordinates has many applications, such as rescue operations for long underground structures and military operations. The ISO-based standards has provisions for detailing these conversions and transformations; this is a topic of on-going research related to material presented in this chapter.

9.4.3 Reference System Extensions

A grid-based indoor CRS can serve as a foundation for developing other types of reference systems. Examples include a geocoded system for supporting topological mapping, such as for IndoorGML; hybrid systems that use floor identifiers (instead of nominal elevations) to identify discrete stories while using coordinates within each floor; one-

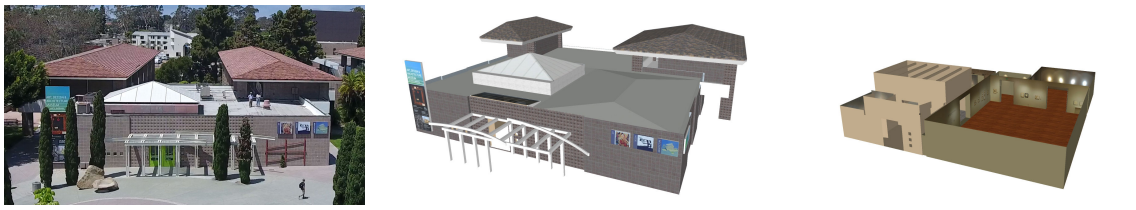
Table 9.1: Coordinate Reference System Specifications for the UCSB AD&A Museum

Element Name	Entry	Comment
Coordinate reference system type code	1	Single coordinate reference system
Coordinate reference system identifier	IGRS274278B534	IGRS signifies use of indoor grid CRS UCSB (U.S. GNIS Feature ID 274278) UCSB AD&A Museum (Bldg. 534)
Datum identifier	Engineering	
Datum anchor point	Principal and Orientation Points	Markers located on the 1st Floor, north wing
Coordinate system axis name	j	
Coordinate system axis direction	local north	Aligns with vector from principal to orientation point.
Coordinate system axis unit identifier	millimeter	
Coordinate system axis name	i	
Coordinate system axis direction	local east	Derived from building model.
Coordinate system axis unit identifier	millimeter	
Coordinate system axis name	k	
Coordinate system axis direction	up	Derived from building model.
Coordinate system axis unit identifier	millimeter	
Coordinate operation parameter name	false easting	
Coordinate operation parameter value	500,000 mm	500,000 mm
Coordinate operation parameter name	false northing	
Coordinate operation parameter value	500,000 mm	500,000 mm
Coordinate operation parameter name	false elevation	
Coordinate operation parameter value	100,000 mm	100,000 mm

dimensional linear systems for long corridors, such as those used in highways; and conversions or transformations to geodetic coordinates. The baseline grid can serve as the common "language" for tying all these systems together.

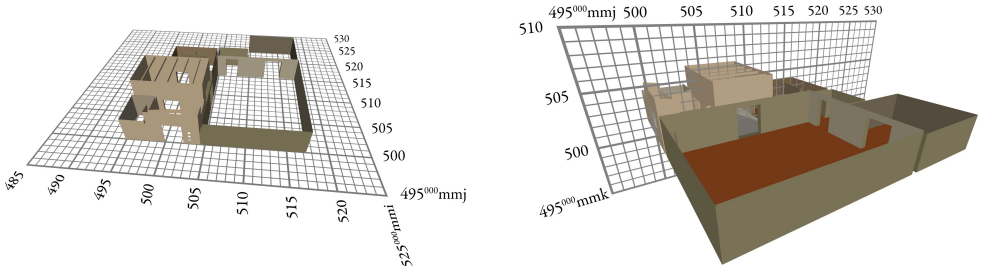
9.5 Example

Table 9.1 and Figures 9.7 to 9.8 provide an example of an indoor grid reference system for a museum located on the campus of the University of California, Santa Barbara.



(a) Picture of museum (b) Building model (exterior) (c) Building model (interior)

Figure 9.7: UCSB Art, Design, and Architecture Museum



(a) Horizontal indoor grid (b) Vertical indoor grid

Figure 9.8: Indoor Grid Reference System for UCSB AD&A Museum

9.6 Conclusion

The concepts and techniques presented in this chapter provide a general approach to developing a grid-based coordinate reference system for indoor spaces. Once established, an IGRS can serve as the basis for developing other types of indoor reference systems. Development of a standard convention for indoor grid referencing can also foster the adoption of indoor maps and location-based services by making indoor measurements more consistent, which can lead to greater predictability across different indoor spaces.

Chapter 10

Georegistering building models to point clouds using ICP

10.1 Introduction

One current challenge for indoor mapping involves the seamless integration of indoor and outdoor maps, especially in three dimensions. Finding a convenient and accessible solution to this problem can help realize the goal of ubiquitous mapping and navigation by removing the metaphorical wall separating the two spaces. An ideal solution would allow local indoor coordinates and outdoor geodetic coordinates to be transformed on-the-fly, permitting the presentation of indoor and outdoor spaces using any choice of coordinate reference system. However, two factors complicate this task: wide variations exist in coordinate reference systems — i.e., the language of location — and building models and measurements of Earth can have unknown levels of error and uncertainty. This chapter provides a general approach to overcoming these difficulties by using building models, georeferenced point clouds, and existing geodetic transformations. Given only three pairs of coordinate triples corresponding to three common anchor points, this

approach translates and scales the building model (containing the indoor space) to match its corresponding points in the georeferenced point cloud then uses an iterative closest point (ICP) routine to refine the alignment. Error minimization in ICP minimizes imperfections in the building model, the point cloud, and the locations of the anchor points. This *registration* process results in two 4x4 homogeneous transformation matrices — one for transforming from building to geodetic coordinates and the other for the reverse transformation.

10.2 Background

10.2.1 Spatial referencing

The International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC) list two ways to describe spatial locations: *referencing by identifiers* and *referencing by coordinates* (International Organization for Standardization 2002, 2003; Bernhardsen 2002). Referencing by identifiers involves using a place name (e.g., Eifel Tower) to describe a location associated with a geographic region, while referencing by coordinates uses coordinate tuples corresponding to some type of *coordinate reference system (CRS)*, which must, at a minimum, consist of a coordinate system and a datum, i.e., measurement starting point.[1] Nearly all coordinates-based maps of indoor spaces use the Cartesian coordinate system paired with arbitrary datums, while outdoor maps can use a wide range of linear, angular, or vertical coordinate systems paired with various (local) engineering, geodetic, or vertical datums. The International Association of Oil and Gas Producers (2017) listed 5,786 different CRSs in 2017 in its widely referenced

[1]However, some organizations such as Esri use a slightly different approach to classifying reference systems and use the term coordinate system to describe a CRS.

EPSG CRS database. While the term *geodetic* has a specific meaning in the ISO/OGC context, this chapter will use the term in a more generic sense to differentiate Earth-based coordinates from building-based coordinates for indoor spaces.

Since nearly all buildings use the Cartesian coordinate system, the simplest approach to indoor-outdoor integration involves using Cartesian coordinate-based geodetic CRSs for the outdoor data, which means using either 2D projected systems or 3D Earth-centered Earth-fixed (ECEF) systems. While ECEF would provide global context and make the transformation results amenable to global navigation satellite systems, use of a 2D projection coupled with a vertical system makes more intuitive sense and can provide a greater level of precision within a small geographic area. Orthogonal grid systems used in 2D projected CRSs provide the horizontal coordinates (x, y) while the vertical CRS provides the vertical z elevation coordinate to form a 3D (x, y, z) coordinate triple.[2] Once the indoor-outdoor transformation has been solved, transforming the projected coordinates into other systems can be done using transformation algorithms implemented in a number of widely available transformation software packages, which this chapter does not cover.

10.2.2 Affine transformation

When both indoor and outdoor CRSs use the Cartesian coordinate system, converting coordinates from one system to the other involves applying an affine transformation (i.e., translation, rotation, and scaling) (Clarke 1995). Use of a constant scaling factor for all three axes results in a special case of an affine transformation called the seven-parameter Helmert transformation (Watson 2006; Maling 1992). For geodetic applications, the co-

[2]In the strictest sense, the vertical axis runs orthogonal to a geoid surface which may slightly vary from the spheroid-based projection of the x, y coordinate plane. However, that is assumed to be negligible for this transformation approach.

ordinate axes follow the right-hand rule with the z-axis pointing in the vertical direction, unlike in computer graphics, which sometimes uses the left-hand rule with a vertical y-axis.

10.2.3 Homogeneous coordinates

Homogeneous coordinates can simplify transformation calculations by allowing multiple transformation operations to be combined into a single matrix (Hearn, Baker, and Carithers 2010). A coordinate triple (x, y, z) for a given point p_i is converted to homogeneous coordinates by appending a value of 1 after the z-coordinate value as shown in Equation 10.1.

$$p_i = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{10.1}$$

Translation, rotation, and scaling matrices then take the form:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \mathbf{S} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10.2}$$

Individual elements of $\mathbf{R}$ represent the product of multiple rotational transformations and may not contain explicit geometric information. The order-dependent nature of transformations means that matrix multiplication occurs from right-to-left.

10.2.4 Coarse and fine registration

The Helmert transformation can provide error-free registration in a single step for perfect data and perfectly corresponding anchor points, but perfect data seldom exists in the real world. Therefore, most registration techniques employ a two-phase approach consisting of an initial coarse registration followed by fine registration using some type of error-minimization technique. In the first phase, coarse registration brings two sets of point clouds into rough alignment using one of three general approaches: matching anchor points, matching scanned targets, or matching features extracted from the point clouds. Once roughly aligned, *fine registration* brings the two data sets closer together using various techniques, with the *iterative closest point (ICP)* technique standing out as the most prominent and widely used. This chapter loosely uses the term ICP to describe both the original ICP approach proposed by Besl and McKay (1992) as well as its variants (Rusinkiewicz and Levoy 2001).

Iterative closest point technique

The iterative closest point (ICP) technique uses matching pairs of nearest points or surface vertices for aligning two geometric data sets, and it can work with various types of geometries, e.g., points, lines, surfaces, etc. (Besl and McKay 1992). Conceptually, ICP involves calculating the distances between nearest point-pairs (i.e., the residuals) and finding a 6-DOF (*degree-of-freedom*) rigid transformation that minimizes a given residual cost function. The transformation process will alter the point pair correspondences, so the ICP algorithm repeats the point-matching and transformations until reaching some terminating criteria, e.g., convergence of residuals, a residual threshold, or maximum number of iterations.

ICP has four important characteristics (Rusinkiewicz 2011; Ju 2012). First, the concept of nearest neighbor can be implemented in a variety of ways, e.g., simple Euclidean distance, shortest normal distance to a tangent line, projection, etc. Second, no prior knowledge exists on point correspondences aside from a vague notion of their relationships established during coarse registration. This means that point pairings will constantly change with each iteration and that a point can potentially have more than one nearest neighbor. It also means that the ICP process may also fail for various reasons, e.g., spacing between the two data sets or insufficient overlap. Third, various efficient methods exist for implementing the 6-DOF rigid transformations that optimize a cost function, two of which include the Levenberg-Marquardt algorithm for non-linear minimization and singular value decomposition (SVD) for linear minimization. Finally, ICP optimization based on mean values — such as with least squares — is very sensitive to outliers and may not provide a robust solution in itself.

This chapter uses Euclidean distances between the point cloud and nearest points on the exterior of the building model, SVD for finding the residual-minimizing transformation matrix, and outlier filtering for mitigating the effects of outliers on the mean.

Singular value decomposition

Singular value decomposition, or SVD, provides an efficient and concise matrix-based approach for aligning two sets of points (e.g., model vertices, point clouds, etc.) using linear least squares optimization (Umeyama 1991; Marschner 2010). Given two sets of matching points, i.e., source points $\mathbf{P}$ and target points $\mathbf{Q}$, the goal involves finding the 3x3 rotation matrix, $\mathbf{r}$, that will best align $\mathbf{P}$ with $\mathbf{Q}$ at their centroids. (Some texts

use the capital $\mathbf{R}$ for the 3x3 rotation matrix, but it is used here to represent the 4x4 homogeneous $\mathbf{R}$ matrix.)

$$\mathbf{P} = \left[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\right]; \mathbf{Q} = \left[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\right] \text{ where } \mathbf{p}_i = \begin{bmatrix} x_{pi} \\ y_{pi} \\ z_{pi} \end{bmatrix}; \mathbf{q}_i = \begin{bmatrix} x_{qi} \\ y_{qi} \\ z_{qi} \end{bmatrix} \tag{10.3}$$

The first step involves calculating the centroids for $\mathbf{P}$ and $\mathbf{Q}$ and then calculating the NxN cross-covariance matrix, $\mathbf{A}$, using the equations:

$$\mathbf{c}_P = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i; \mathbf{c}_Q = \frac{1}{N}\sum_{i=1}^{N}\mathbf{q}_i; \mathbf{A} = \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{p}_i - \mathbf{c}_P)(\mathbf{q}_i - \mathbf{c}_Q)^\top] \tag{10.4}$$

$$\mathbf{A} = \mathbf{Q}^\top \cdot \mathbf{P} \tag{10.5}$$

The next step involves finding the orthonormal matrix, $\mathbf{r}$, that maximizes the trace of $\mathbf{r} \cdot \mathbf{A}$ and also happens to represent the optimal $\mathbf{P}$-to-$\mathbf{Q}$ rotation matrix. $\mathbf{A}$ decomposes into the following form, which can be calculated with math processing software:

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^\top \tag{10.6}$$

The equation for $\mathbf{r}$ that maximizes $\mathbf{A}$ takes the form

$$\mathbf{r} = \mathbf{U} \cdot \mathbf{S}_{SVD} \cdot \mathbf{V}^{\top} \tag{10.7}$$

$$\mathbf{S}_{SVD} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} \tag{10.8}$$

$$d = \det(\mathbf{U}) \cdot \det(\mathbf{V})$$

Transforming from $\mathbf{P}$ to $\mathbf{Q}$ involves translating the centroid of $\mathbf{P}$ to the origin, rotating $\mathbf{P}$ to match the orientation of $\mathbf{Q}$ at the origin, and translating the centroid of $\mathbf{P}$ to match the centroid of $\mathbf{Q}$. For each point $\mathbf{p}_i$, the transformed location is represented by $\mathbf{p}'_i$ in the following equation:

$$\mathbf{p}'_i = \mathbf{r} \cdot (\mathbf{p}_1 - \mathbf{c}_P) + \mathbf{c}_Q \tag{10.9}$$

Converting $\mathbf{c}_P$, $\mathbf{r}$, $\mathbf{c}_Q$, and point set $\mathbf{P}$ into homogeneous form results in

$$\mathbf{T}_P = \left[\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 0 & -\mathbf{c}_P \\ 0 & 0 & 1 & \\ \hline 0 & 0 & 0 & 1 \end{array}\right]; \mathbf{R} = \left[\begin{array}{ccc|c} & \mathbf{r} & & 0 \\ & & & 0 \\ & & & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]; \mathbf{T}_Q = \left[\begin{array}{ccc|c} 1 & 0 & 0 & \\ 0 & 1 & 0 & \mathbf{c}_Q \\ 0 & 0 & 1 & \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{10.10}$$

so that equation 10.9 for a single point $\mathbf{p}_i$ becomes

$$\mathbf{P}' = \mathbf{M} \cdot \mathbf{P} \text{ where } \mathbf{M} = \mathbf{T}_Q \cdot \mathbf{R} \cdot \mathbf{T}_P \tag{10.11}$$

for the entire point set $\mathbf{P}$, where $\mathbf{P}'$ represents the best least-squares fit with $\mathbf{Q}$.

Filtering outliers

SVD implicitly uses mean values for least squares optimization (Equation 10.4), but the mean is highly sensitive to outliers with its finite sample breakdown point of $1/n$ meaning that just one out of n observations can result in arbitrarily large or small results. Wilcox (2010) reviewed several robust statistical approaches for mitigating this shortcoming, which included trimming the two tails of the distribution of residuals resulting in least trimmed squares (LTS) values. For univariate residuals, he recommended trimming 20% of the upper and lower tails but noted that the very act of trimming negated the assumption of homoscedasticity used in calculating the variance.

For point cloud data with extraneous features (e.g., trees, parked vehicles, and sidewalks), trimming may not necessarily negate homoscedasticity early in the ICP process since extreme outliers would correspond to irrelevant non-building features. However, it can present a problem as the building and point cloud converge since trimming would remove desirable leverage points. Chetverikov et al. (2002) implemented an LTS ICP algorithm, which they called Trimmed ICP or TrICP, that a fixed proportion of residuals in 2D data sets. Their study demonstrated the robustness of TrICP compared to using the full set of residuals but it also showed a progressive growth in errors beyond a certain trimming value.

Among approaches to multivariate residuals, Wilcox mentioned the relplot and minimum volume ellipsoid estimator as two robust approaches. The relplot represents a 2D equivalent of the boxplot while the minimum volume ellipsoid represents a special case of a minimum bounding volume, each aligned with the principal directions of the data points and sized to contain a certain quantile of data.

10.3 Methodology

Given a building model, a georeferenced point cloud of the building's exterior, and three anchor points between the building model and point cloud, we wish to find the 4x4 transformation matrices that will convert building coordinates into geodetic coordinates and vice versa using pure geometry (i.e., omitting other variables such as color). This process involves five phases: coarse registration at the origin, fine registration at the origin, translation to the geodetic location, calculating the Helmert transformation for the building's anchor points between the original and final locations, and calculating the reverse transformation.

10.3.1 Coarse registration at origin

Coarse registration involves approximating the alignment of the building model with the georeferenced point cloud *at the origin*. This phase requires an *a priori* unit of measurement conversion factor and the manual selection of three common anchor points in both the point cloud and building model, which are used in a Helmert transformation to translate, rotate, and scale the model to approximately fit the point cloud. With perfect data, the Helmert transformation would result in perfect registration. Empirical data, however, will result in an imperfect fit. Figure 10.1 illustrates the coarse registration process.

Preliminaries

Let $\mathbf{b}_n$ and $\mathbf{e}_n$ denote coordinate triples for the three anchor points in the engineering CRS and geodetic CRS, respectively, where $n = 1, 2, 3$ denotes the point number and $\mathbf{B}$ and $\mathbf{E}$ denote the coordinate sets for $\mathbf{b}_n$ and $\mathbf{e}_n$. Points $\mathbf{b}_1$, $\mathbf{b}_2$, and $\mathbf{b}_3$ and $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ thus represent the three pairs of corresponding anchor points for a total of six

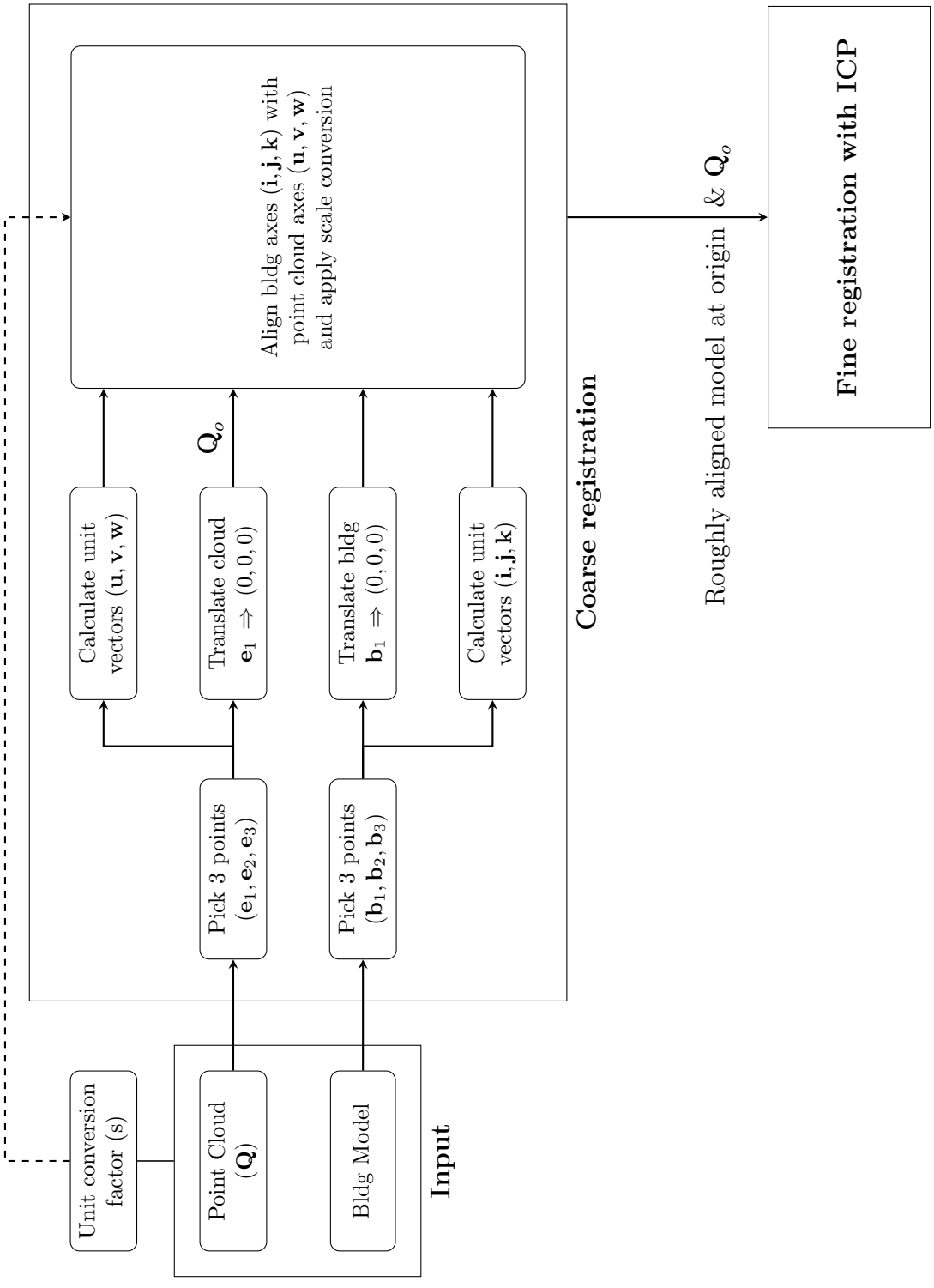


Figure 10.1: Coarse registration

coordinate triples. A unitless intermediate CRS is introduced with axes labels (x, y, z). The intermediate CRS has its origin at $(0, 0, 0)$ and, by definition, contains both the building model (engineering CRS) and point cloud (geodetic CRS) stripped of their units of measurement as shown in Figure 10.2. Here, **i**, **j**, and **k** and **u**, **v**, and **w** define the unit vectors of the building and point cloud in terms of (x, y, z) values.

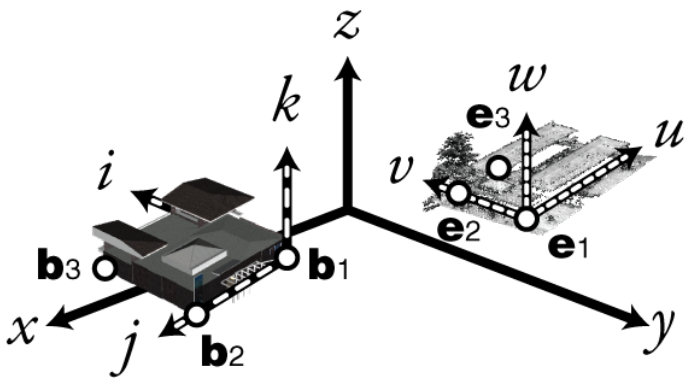


Figure 10.2: Coarse registration of building model and point cloud

$$\mathbf{i} = \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix} ; \mathbf{j} = \begin{bmatrix} j_x \\ j_y \\ j_z \end{bmatrix} ; \mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} ; \mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} ; \mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} ; \mathbf{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \tag{10.12}$$

Find the unit vectors

Define the unit vectors **i**, **j**, and **k** as follows and summarized in Tables 10.1 and 10.2; apply the same process for finding **u**, **v**, and **w**:

- $\mathbf{b}_1$ defines the origin
- vector $\mathbf{V}_j$ from $\mathbf{b}_1$ to $\mathbf{b}_2$ points along the secondary axis (j)
- vector $\mathbf{V}_{Lp}$ from $\mathbf{b}_1$ to $\mathbf{b}_3$ points in a direction along the $i - j$ plane

Table 10.1: Raw vectors

	Engineering CRS	Geodetic CRS
Origin	$\mathbf{b}_1$	$\mathbf{e}_1$
First Axis	$\mathbf{V}_i = \mathbf{V}_j \times \mathbf{V}_k$	$\mathbf{V}_u = \mathbf{V}_v \times \mathbf{V}_w$
Second Axis	$\mathbf{V}_j = \mathbf{b}_2 - \mathbf{b}_1$	$\mathbf{V}_v = \mathbf{e}_2 - \mathbf{e}_1$
Third Axis	$\mathbf{V}_k = \mathbf{V}_{Lp} \times \mathbf{V}_j$	$\mathbf{V}_w = \mathbf{V}_{Ep} \times \mathbf{V}_v$
Planar Point	$\mathbf{V}_{Lp} = \mathbf{b}_3 - \mathbf{b}_1$	$\mathbf{V}_{Ep} = \mathbf{e}_3 - \mathbf{e}_1$

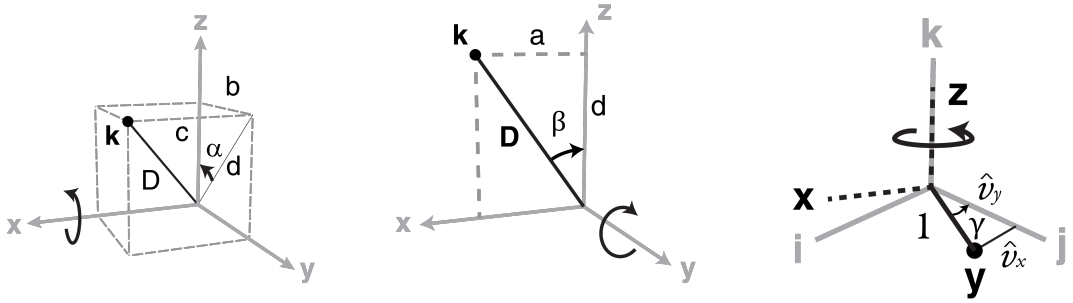
Table 10.2: Unit vectors

	Engineering CRS	Geodetic CRS
Origin	$\mathbf{b}_1$	$\mathbf{e}_1$
First Axis	$\mathbf{i} = \mathbf{V}_i / Norm(\mathbf{V}_i)$	$\mathbf{u} = \mathbf{V}_u / Norm(\mathbf{V}_u)$
Second Axis	$\mathbf{j} = \mathbf{V}_j / Norm(\mathbf{V}_j)$	$\mathbf{v} = \mathbf{V}_v / Norm(\mathbf{V}_v)$
Third Axis	$\mathbf{k} = \mathbf{V}_k / Norm(\mathbf{V}_k)$	$\mathbf{w} = \mathbf{V}_w / Norm(\mathbf{V}_w)$

- vector $\mathbf{V}_k$ emanates from $\mathbf{b}_1$ and runs orthogonal to the $i - j$ plane
- vector $\mathbf{V}_i$ runs orthogonal to $\mathbf{V}_j$ & $\mathbf{V}_k$ using the right-hand rule from j to k
- normalize $\mathbf{V}_i$, $\mathbf{V}_j$, and $\mathbf{V}_k$ to get unit vectors $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$

Calculate transformations to the (x, y, z) origin

Using the unit vectors $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ and $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, use Equations 10.13 to 10.15 to find the translations and rotations to match them with the (x, y, z) axes; rotations at the origin are illustrated in Figure 10.3. Subscripts B2O, E2O, and O2E denote building-to-origin, geodetic-to-origin, and origin-to-geodetic transformations, respectively, while lower-case matrix variables represent the 3x3 form and upper-case the 4x4 homogeneous form.



(a) Rotate α about $x-$axis (b) Rotate β about $y-$axis (c) Rotate γ about $z-$axis

Figure 10.3: Rotations in the order a-b-c to align the ijk axes with the xyz axes

$$\mathbf{t}_{B2O} = \mathbf{a}_1 - \mathbf{b}_1; \mathbf{t}_{E2O} = \mathbf{a}_1 - \mathbf{e}_1 \tag{10.13}$$

$$\mathbf{r}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c/d & -b/d \\ 0 & b/d & c/d \end{bmatrix}; \mathbf{r}_y = \begin{bmatrix} d & 0 & -a \\ 0 & 1 & 0 \\ a & 0 & d \end{bmatrix}; \mathbf{r}_z = \begin{bmatrix} -e & f & 0 \\ -f & -e & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10.14}$$

Engineering CRS ($\mathbf{r}_{B2O}$): $a = k_x, b = k_y, c = k_z, d = \sqrt{b^2 + c^2} = \sqrt{k_y^2 + k_z^2}$

$$e = i_{2x}, f = i_{2y} \text{where } \mathbf{i}_2 = \begin{bmatrix} i_{2x} \\ i_{2y} \\ i_{2z} \end{bmatrix} = \mathbf{r}_y \cdot \mathbf{r}_x \cdot \mathbf{i}$$

$$\text{Geodetic CRS } (\mathbf{r}_{E2O}):\ a = w_x, b = w_y, c = w_z, d = \sqrt{b^2 + c^2} = \sqrt{w_y^2 + w_z^2}$$

$$e = u_{2x}, f = u_{2y} \text{where } \mathbf{u}_2 = \begin{bmatrix} u_{2x} \\ u_{2y} \\ u_{2z} \end{bmatrix} = \mathbf{r}_y \cdot \mathbf{r}_x \cdot \mathbf{u}$$

$$\mathbf{r}_{B2O} = \mathbf{r}_{zB2O} \cdot \mathbf{r}_{yB2O} \cdot \mathbf{r}_{xB2O};\ \mathbf{r}_{O2E} = \mathbf{r}_{xE2O}^T \cdot \mathbf{r}_{yE2O}^T \cdot \mathbf{r}_{zE2O}^T \tag{10.15}$$

Equation 10.16 shows the scaling matrix using the *a priori* unit conversion factor s from building to point cloud units.

$$\mathbf{s}_{B2E} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \tag{10.16}$$

Assembling and applying the transformation matrices

The assembled transformation matrices in Equations 10.17 and 10.18 will align the building model with the point cloud at the origin, completing the coarse alignment. Note that the building model is rotated a second time using $\mathbf{R}_{O2E}$ to match the unrotated point cloud so that the vertical direction always points up, which can be useful for intepreting ICP results.

$$\mathbf{M}_{B,coarse} = \mathbf{R}_{O2E} \cdot \mathbf{S}_{B2E} \cdot \mathbf{R}_{B2O} \cdot \mathbf{T}_{B2O} \tag{10.17}$$

$$\mathbf{M}_{E,coarse} = \mathbf{T}_{E2O} \tag{10.18}$$

10.3.2 Fine registration at the origin

Point cloud filtering

This chapter takes a naive approach to point cloud filtering and assumes an unclassified point cloud that has not been processed with a morphological filter to remove vegetation, vehicles, and other extraneous features. Therefore, removal of undesirable elements in the point cloud takes place with two filters. A preliminary filter — applied prior to ICP — removes extraneous points beyond a certain distance from the building, while an ICP filter works with each ICP iteration to progressively remove irrelevant points. The preliminary filter preserves all points within an expanded convex hull of the building and discards all other points; it also removes all ground points surrounding the building.

The main ICP filter uses simple trimming to remove points corresponding to the upper 40% of nearest neighbor distances. Three simple volumetric filters that retain 60% of points (i.e., 40% trimming) are also explored (cuboid, octahedron, and ball) with the following alignments: the cuboid faces run orthogonal to the three principal axes (eigenvectors) of the residuals while the octahedron's vertices align with the principal axes.

Iterative closest point process

The ICP process takes the following form: calculate the nearest neighbor (NN), filter the point cloud based on NN Euclidean distances, calculate the mean NN distance of the filtered points, evaluate termination criteria, use SVD to find the optimum rotation and translation of the building model, and iterate, as illustrated in Figure 10.4.

Iteration ceases upon reaching one of three termination criteria: convergence of residual values, reaching a mean residual threshold, or reaching a maximum number of itera-

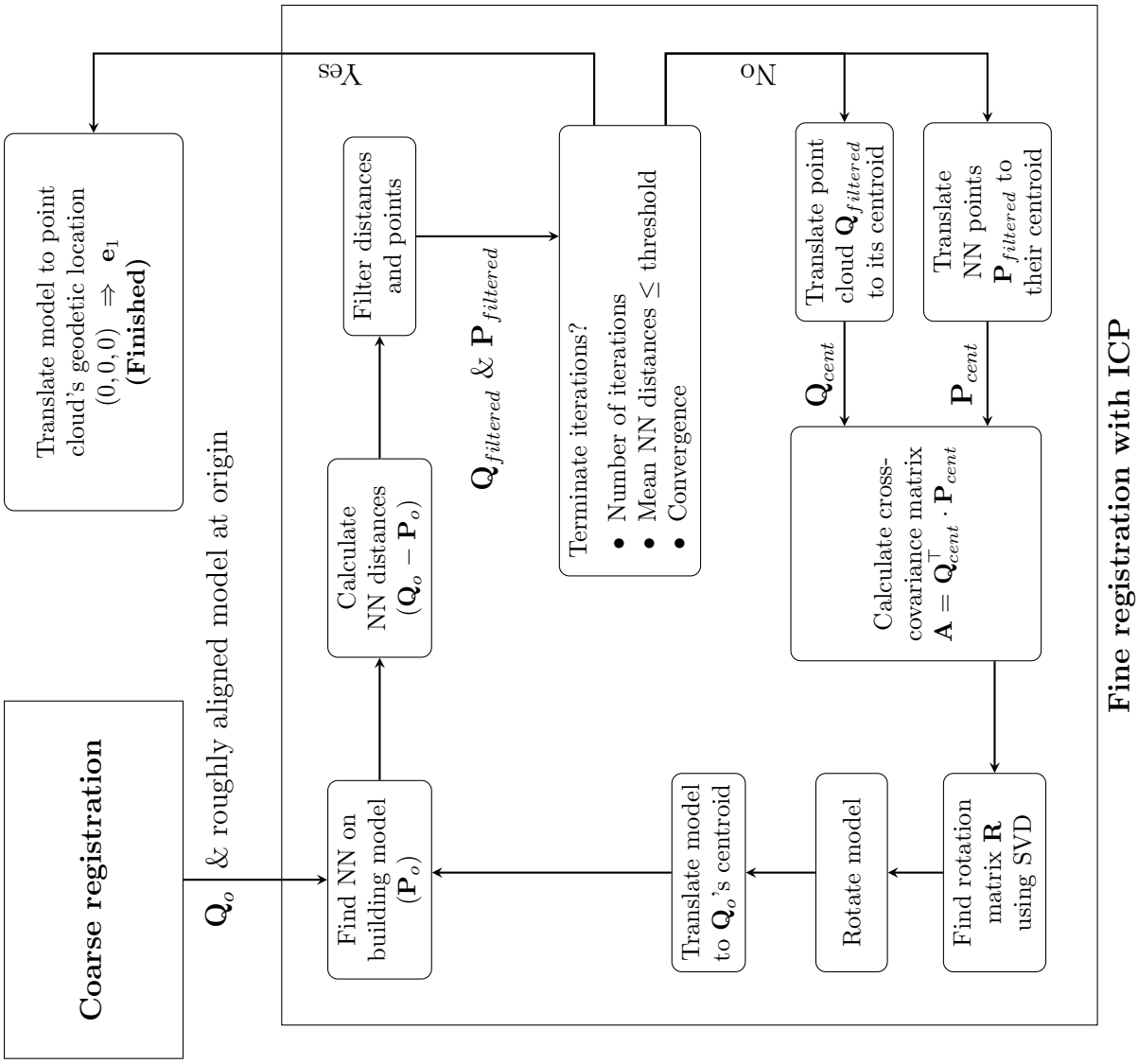


Figure 10.4: Fine registration with ICP

tions. Ten ICP iterations are used to evaluate convergence and the residual threshold in order to prevent premature termination from occasional erratic movements of the mean residual values. Convergence involves ten consecutive iterations where the mean residuals deviate no more than a certain threshold, such as 1 millimeter; upon convergence, the first of the ten iterations represents the convergence point.

10.3.3 Translation to final geodetic location

Upon completion of ICP registration, the model is translated to its geodetic location using the inverse of the translation for moving the point cloud to the origin as shown in Equation 10.19.

$$\mathbf{T}_{O2E} = \mathbf{T}_{E2O}^{-1} \tag{10.19}$$

10.3.4 Final transformation matrices

Completion of the transformation process results in one unknown matrix, $\mathbf{M}_{unk}$, in the following chain of transformations from building to geodetic coordinates:

$$\mathbf{M}'_{B2E,final} = \mathbf{T}_{O2E} \cdot \boxed{\mathbf{M}_{unk}} \cdot \mathbf{S}_{B2E} \cdot \mathbf{R}_{B2O} \cdot \mathbf{T}_{B2O} \tag{10.20}$$

While $\mathbf{M}_{unk}$ can be found by accumulating the translations and rotations for each ICP iteration, a more convenient solution involves applying the Helmert transformation to the building's three anchor points between their original locations ($\mathbf{B}$) and final locations ($\mathbf{B}'$) as described in Equations 10.21 and 10.22.

$$\mathbf{M}_{B2E,final} = \text{Helmert transformation from } \mathbf{B} \text{ to } \mathbf{B}' \tag{10.21}$$

$$\mathbf{M}_{E2B,final} = \text{Helmert transformation from } \mathbf{B}' \text{ to } \mathbf{B} \tag{10.22}$$

10.4 Case studies

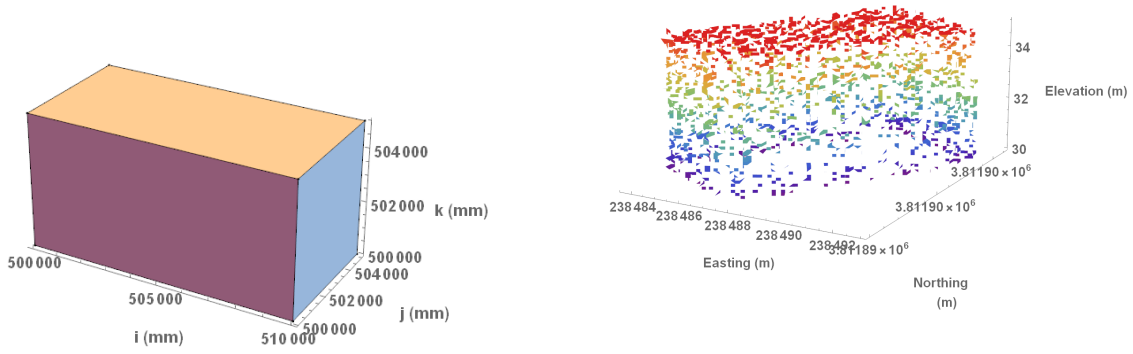
Three case studies demonstrate the effectiveness of this registration approach. The first study uses a computer-generated building (i.e., rectangular box) and a point cloud that was created by randomly sampling points from the building's surface. The second study adds a ground surface, trees, and random noise to the point cloud. For these two synthetic data sets, eight *key points* were manually inserted into the point cloud at locations corresponding to the buildings's vertices; these points made it possible to evaluate the building model's fit based on *a priori* locations of the corner points — impossible with real data. Finally, the last study uses an actual model of a building and a point cloud from aerial LiDAR. All three case studies used 40% trimming for the ICP filter, corresponding to retention of the 60% closest points.

10.4.1 Case Study 1: Perfect point cloud

Case Study 1 demonstrates the basic operation of ICP and shows the problems associated with over-trimming. The goal of Case Study 1 is to find the transformation matrix for fitting the building model in local coordinates (Figure 10.5a) to the point cloud in UTM (WGS84) coordinates (Figure 10.5b) using the three anchor points shown in Table 10.3. Termination criteria for ICP included convergence at the fourth decimal place, a residual threshold of 0.1 mm, and a maximum iteration count of 100.

Table 10.3: Anchor points

	Building (Engineering CRS) millimeters	LiDAR (Geodetic CRS) meters (intentionally misaligned)
First Point	$\mathbf{b}_1 = \{500000, 500000, 505000\}$	$\mathbf{e}_1 = \{238488.500, 3811890.000, 35.5\}$
Second Point	$\mathbf{b}_2 = \{510000, 505000, 505000\}$	$\mathbf{e}_2 = \{238488.170, 3811901.660, 34.5\}$
Third Point	$\mathbf{b}_3 = \{510000, 500000, 505000\}$	$\mathbf{e}_3 = \{238493.500, 3811898.160, 35.0\}$

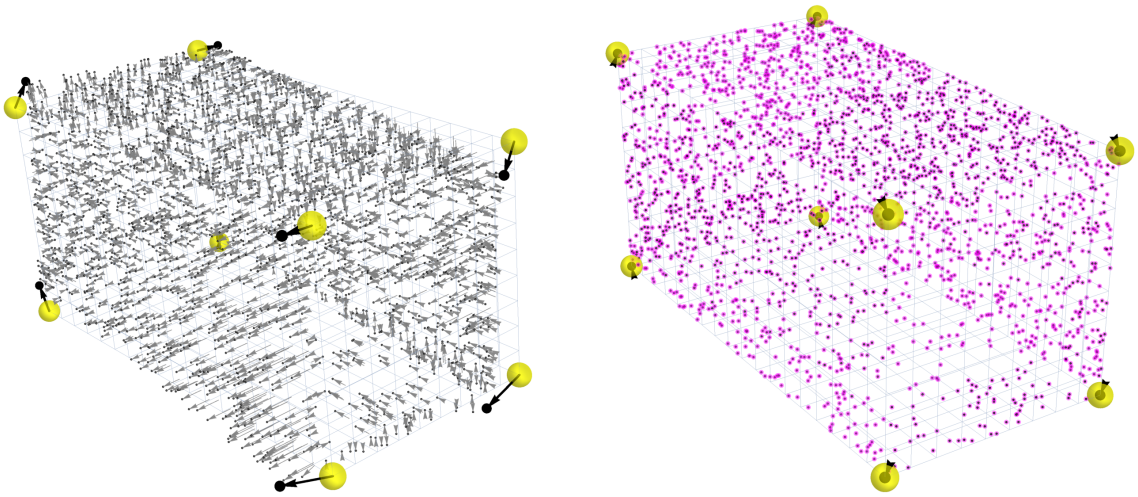


(a) Building model, local coordinates

(b) Point cloud, UTM (WGS84) coordinates

Figure 10.5: Building model and perfectly simulated point cloud

Figure 10.6 shows conditions before and after ICP. The gridded box in Figure 10.6a shows the location of the building before ICP registration, with the grey arrows showing the NN vectors pointing from the building to the point cloud and the black arrows showing the actual (non-NN) vectors for the eight key points. Figure 10.6b shows the result of running ICP using the entire unfiltered point cloud.



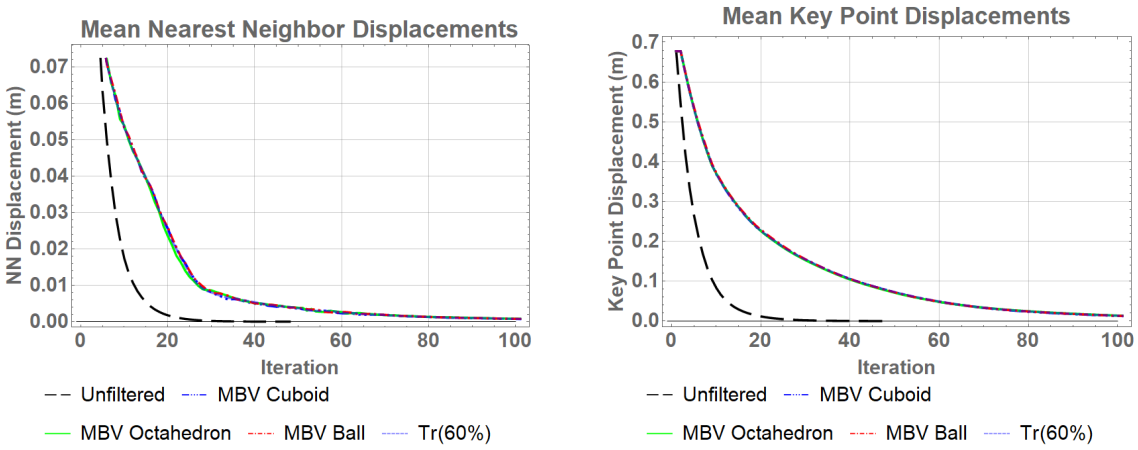
(a) Before ICP (b) After ICP

Figure 10.6: Nearest neighbor displacements before and after ICP

Unsurprisingly, unfiltered ICP provided the best performance in terms of least number of iterations, precision (measured by the NN displacements), and accuracy (measured by the key point displacements), as shown in Figure 10.7 and Table 10.4. Use of the key points provides an additional insight not possible with actual data; with perfect data, NN underestimated the mean error by around a factor of ten for the filtered points and a factor of five for unfiltered. This implies that NN precision metrics should be used with caution when interpreted as a measure of accuracy.

Table 10.4: Case Study 1 results

Filtering Method	I	T	C	Time (sec)	Iteration Count	Mean NN Disp (m).	Mean KP Disp. (m)	NN & KP Diff. (m)
Unfiltered			•	1.2	37	0.000105	0.000541	0.000435
MBV Cuboid (60%)	•			16.2	100	0.001361	0.012904	0.011543
MBV Octahedron (60%)	•			15.1	100	0.001432	0.013643	0.012211
MBV Ball (60%)	•			440.0	100	0.001391	0.013152	0.011761
Trimming (60%)	•			4.1	100	0.001390	0.013410	0.012020

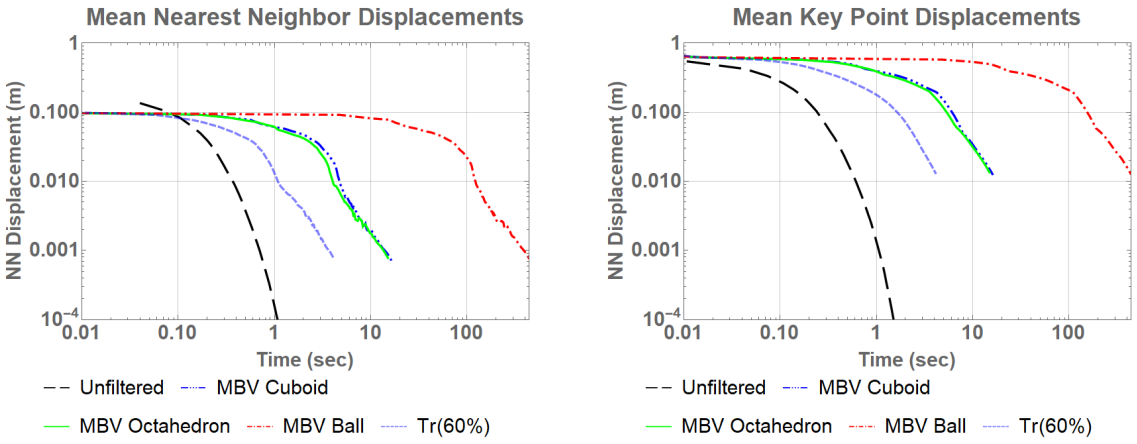


(a) Mean NN Displacements (m)

(b) Mean Key Point Displacements (m)

Figure 10.7: Mean NN and key point displacements versus number of iterations

The choice of filtering method had a significant influence on processing time as shown in Figure 10.8 and Table 10.4. Without the need for pre-processing, the unfiltered data converged in the shortest amount of time, as expected with perfect data. Simpler filters resulted in faster processing times with similar results: trimming approached convergence in the shortest time followed by volumetric filtering with increasing complexity (cuboid, octahedron, and ball), which required iterative volume searches. Notably, the ball filter had a similar shape to trimming but took over 100 times the time to process without reaching convergence.

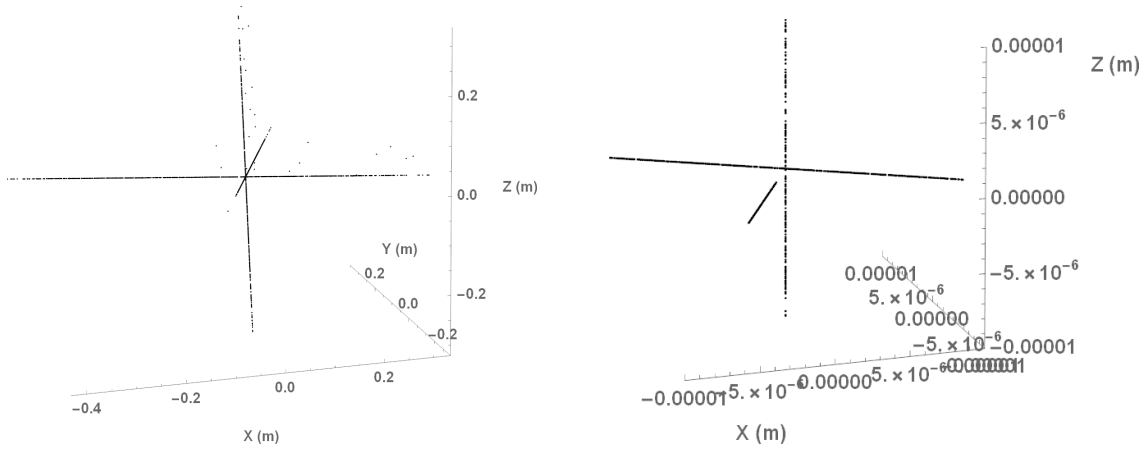


(a) Mean NN Displacements (m)

(b) Mean Key Point Displacements (m)

Figure 10.8: Mean NN and key point displacements versus time

A plot of the NN displacements (Figure 10.9) showed a unique spatial pattern related to the shape of the building. Since coarse registration placed the building model into rough alignment with the point cloud, most of the displacement vectors ran normal to one of the model surfaces, resulting in vectors having one of six orientations (corresponding to each of the six surfaces of the rectangular box) but with different magnitudes. Vertices would have NN points that could point in any direction, resulting in the "speckles" in Figure 10.9a. As the model converges with the point cloud, a greater number of displacement vectors will emanate from a surface resulting in fewer speckles and greater alignment with the surface's normal vectors (Figure 10.9b).

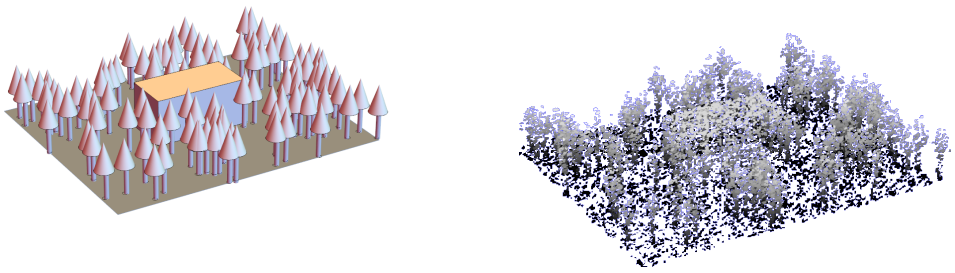


(a) Before ICP (b) After ICP

Figure 10.9: Plot of residuals for all NN points

10.4.2 Case Study 2: Point cloud with outliers

Case Study 2 introduces noise and imperfections to Case Study 1's point cloud in the form of a ground surface, trees, and random noise as shown in Figure 10.10. The three anchor points remain the same (Table 10.6).

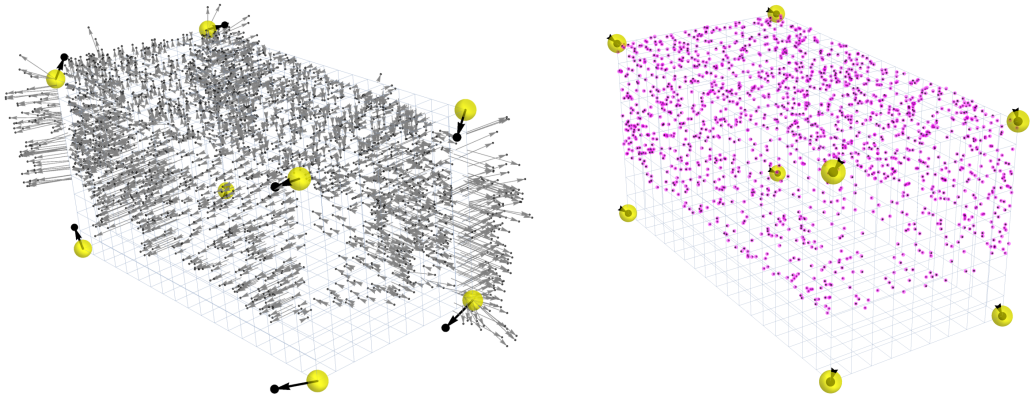


(a) Computer generated environment (b) Point cloud, UTM (WGS84)

Figure 10.10: Building model with environmental noise and simulated point cloud

Figure 10.11 shows conditions before and after ICP for Case Study 2. Preliminary filtering removed most of the points for ground and trees from the point cloud, although a few of the closest points remained (Figure 10.11a). ICP with filtering removed nearly

all of the non-building points allowing for a close fit between model and point cloud as shown in Figure 10.11b using Trimmed ICP.



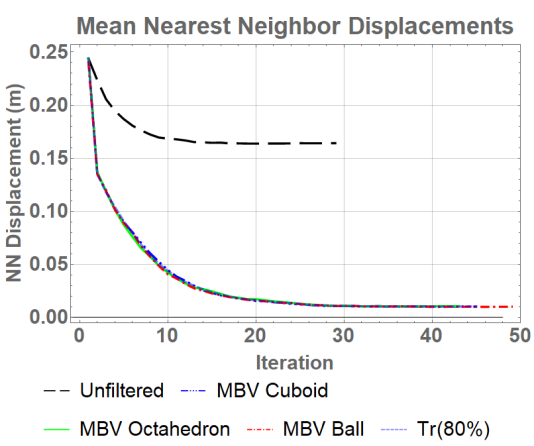
(a) Before ICP (b) After ICP

Figure 10.11: Nearest neighbor displacements before and after ICP

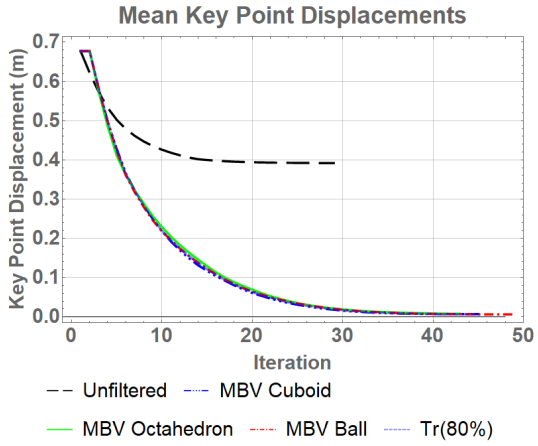
Mean displacements for all five ICP methods converged at the fourth decimal place although unfiltered ICP produced poorer results compared to the filtered methods as shown in Figure 10.12. Mean NN measures more closely mirrored the mean error of the key points with differences ranging from 0.6% (Ball) to 29% (octahedron)... far less than differences over 300% observed in Case Study 1. However, greater correlation did not imply greater accuracy; the cuboid (11.5% difference) had the most accurate fit at 0.9 mm mean KP error followed by trimming (6.5% difference) at 1.0 mm error. Unfiltered ICP underestimated the actual error by 16.8%, producing a NN mean of 33.7 cm when the actual mean error was 40.5 cm.

Table 10.5: Case Study 2 results

Filtering Method	I	T	C	Time (sec)	Iteration Count	Mean NN Disp (m).	Mean KP Disp. (m)	NN & KP Diff. (m)
Unfiltered			•	0.7	18	0.337205	0.405401	0.068196
MBV Cuboid (60%)			•	9.0	34	0.010217	0.009042	-0.001176
MBV Octahedron (60%)			•	7.3	32	0.010545	0.013652	0.003107
MBV Ball (60%)			•	335.6	34	0.010426	0.010486	0.000060
Trimming (40%)			•	1.6	34	0.010416	0.009736	-0.000679



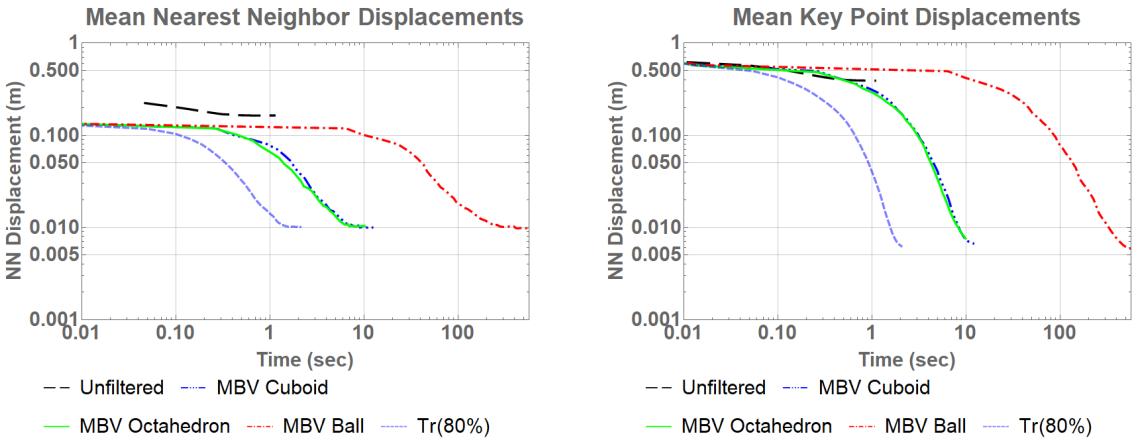
(a) Mean NN displacements (m)



(b) Mean key point displacements (m)

Figure 10.12: Mean NN and key point displacements versus number of iterations

Similar to Case Study 1, processing times generally correlated to the complexity of the filter, although the octahedron converged in fewer iterations compared to the cuboid resulting in a faster processing time (Figure 10.13).

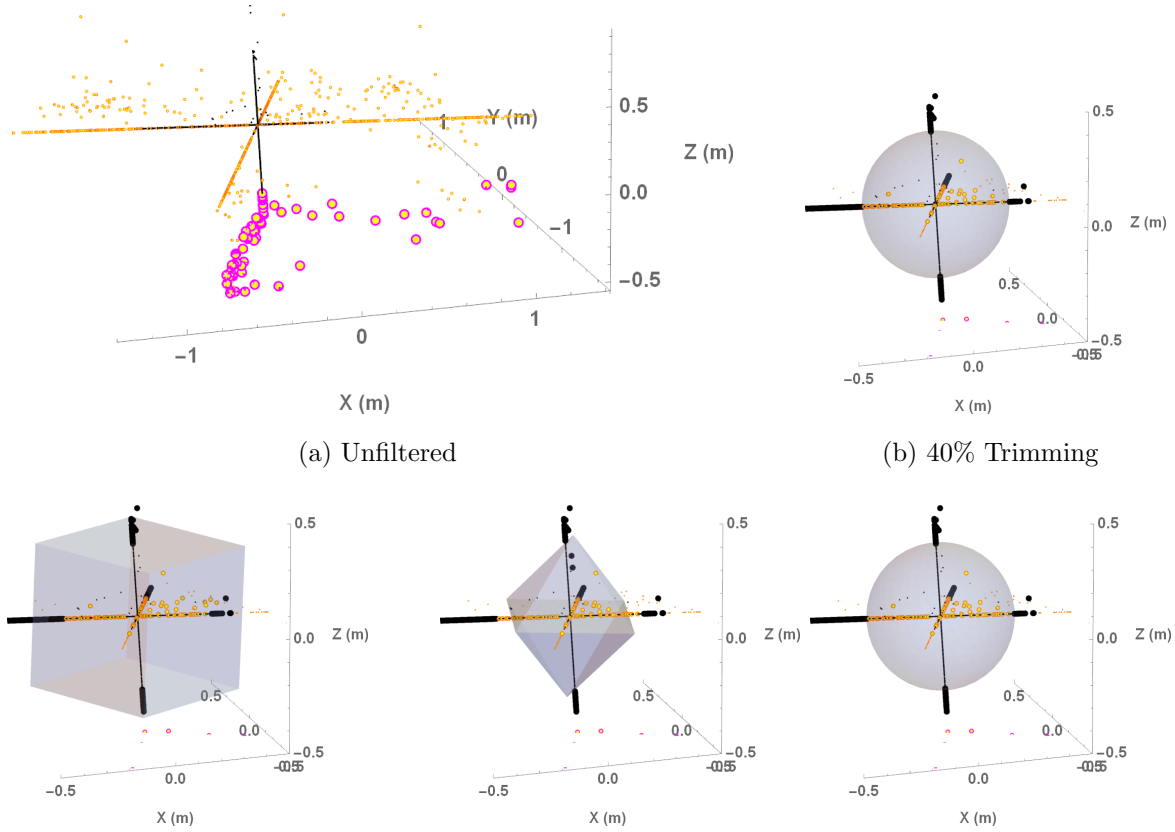


(a) Mean NN displacements (m)

(b) Mean key point displacements (m)

Figure 10.13: Mean NN and key point displacements versus time

Figure 10.14 provides visualizations of the ICP filters at the first iteration. Black dots represent correspondences with point cloud points for the building: small dots for those captured as inliers and large dots for those mistakenly labeled as outliers. Orange dots represent correspondences with tree points: small dots for those correctly filtered as outliers and large dots as those mistakenly labeled as inliers. Yellow dots with a magenta outline represent correspondences with the ground; filtering effectively removed all ground points.

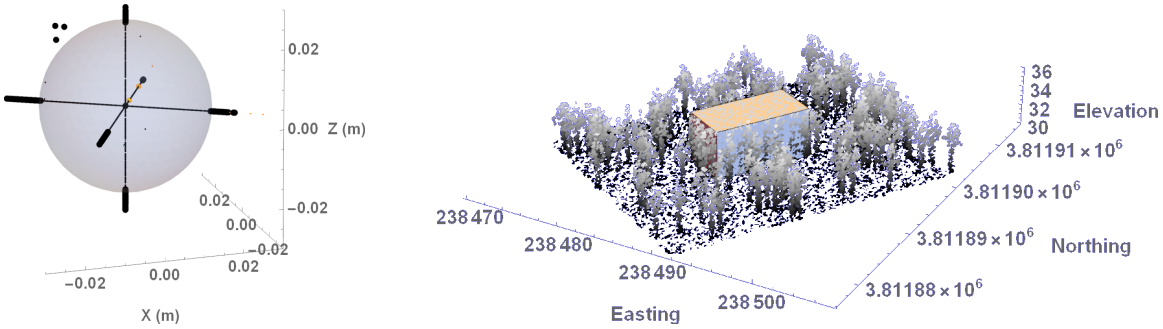


(a) Unfiltered (b) 40% Trimming

(c) Cuboid (d) Octahedron (e) Ball

Figure 10.14: Effects of various filters during first ICP iteration

All four filters effectively removed outliers and produced transformations resulting in fits at around 1 cm accuracy for the 10 m x 5 m x 5 m building. Figure 10.15 shows results for 40% trimming.



(a) Displacement vectors at convergence

(b) Building transformed and placed in point cloud

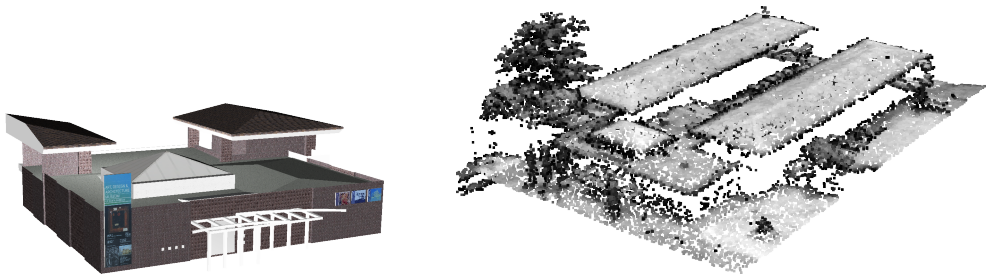
Figure 10.15: Results of trimmed ICP at 40% trimming

10.4.3 Case Study 3: Point cloud with outliers

Case Study 3 applies the registration approach to a building model and point cloud of the Art, Design, and Architecture (AD&A) Museum at the University of California, Santa Barbara. The building model included the interior and exterior of the building and was manually created in Trimble Sketchup using terrestrial laser scans of the building's interior and exterior and drone-based photogrammetry of the exterior (Figure 10.16a). The building model used a building CRS with units of millimeters and a false origin displacement of 500,000 mm in all three directions. The point cloud came from a high-resolution Geiger mode aerial LiDAR collection in 2016 and used the State Plane Coordinate System, California Zone 5, with units of feet (Figure 10.16b).

Table 10.6: Anchor points

	Building (Engineering CRS) millimeters	LiDAR (Geodetic CRS) feet
First Point	$\mathbf{b}_1 = \{517979, 499586, 499900\}$	$\mathbf{e}_1 = \{6004107.469, 1977506.566, 42.807\}$
Second Point	$\mathbf{b}_2 = \{490272, 499586, 499900\}$	$\mathbf{e}_2 = \{6004106.031, 1977414.188, 42.866\}$
Third Point	$\mathbf{b}_3 = \{517989, 511684, 499900\}$	$\mathbf{e}_3 = \{6004068.369, 1977506.600, 42.982\}$

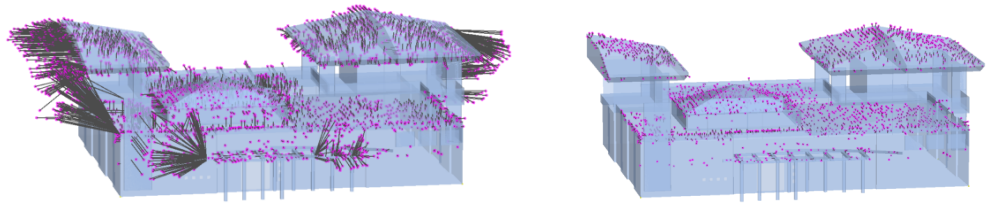


(a) Building model, local coordinates

(b) Point cloud, State Plane

Figure 10.16: UCSB AD&A Museum building model and point cloud

Figure 10.17a shows the building model and point cloud at the origin after coarse registration but before ICP, with lines showing displacements from the NN on the model to the point cloud. Figure 10.17b shows the final result after ICP using a trimming filter at 40% trimming.



(a) Before ICP

(b) After ICP

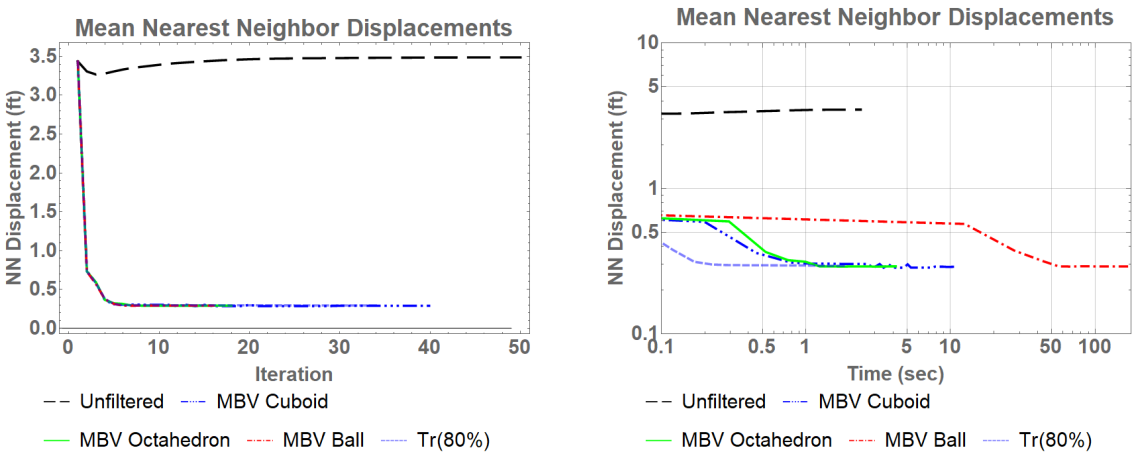
Figure 10.17: Nearest neighbor displacements before and after ICP

Unfiltered ICP resulted in no meaningful improvement to the coarse registration, with the process converging at 3.5 feet or 42 inches mean NN precision. However, all four filtering methods produced significant improvements with final mean NN precisions at about

Table 10.7: Case Study 3 results

Filtering Method	I	T	C	Time (sec)	Iteration Count	Mean NN Disp (ft)
Unfiltered			•	1.9	39	3.486180
MBV Cuboid (60%)			•	7.8	29	0.289166
MBV Octahedron (60%)			•	1.5	7	0.290259
MBV Ball (60%)			•	56.0	5	0.290543
Trimming (40%)			•	1.2	23	0.293741

3.5 inches (Table 10.7). As with Case Study 2, time to convergence for these methods depended on the complexity of the filter; trimmed ICP worked the fastest followed by the octahedron, cuboid, and ball. While the cuboid has a simpler geometry, the octahedron took fewer iterations to converge (7 versus 29) resulting in a lower convergence time (1.5 sec verus 7.8 sec).

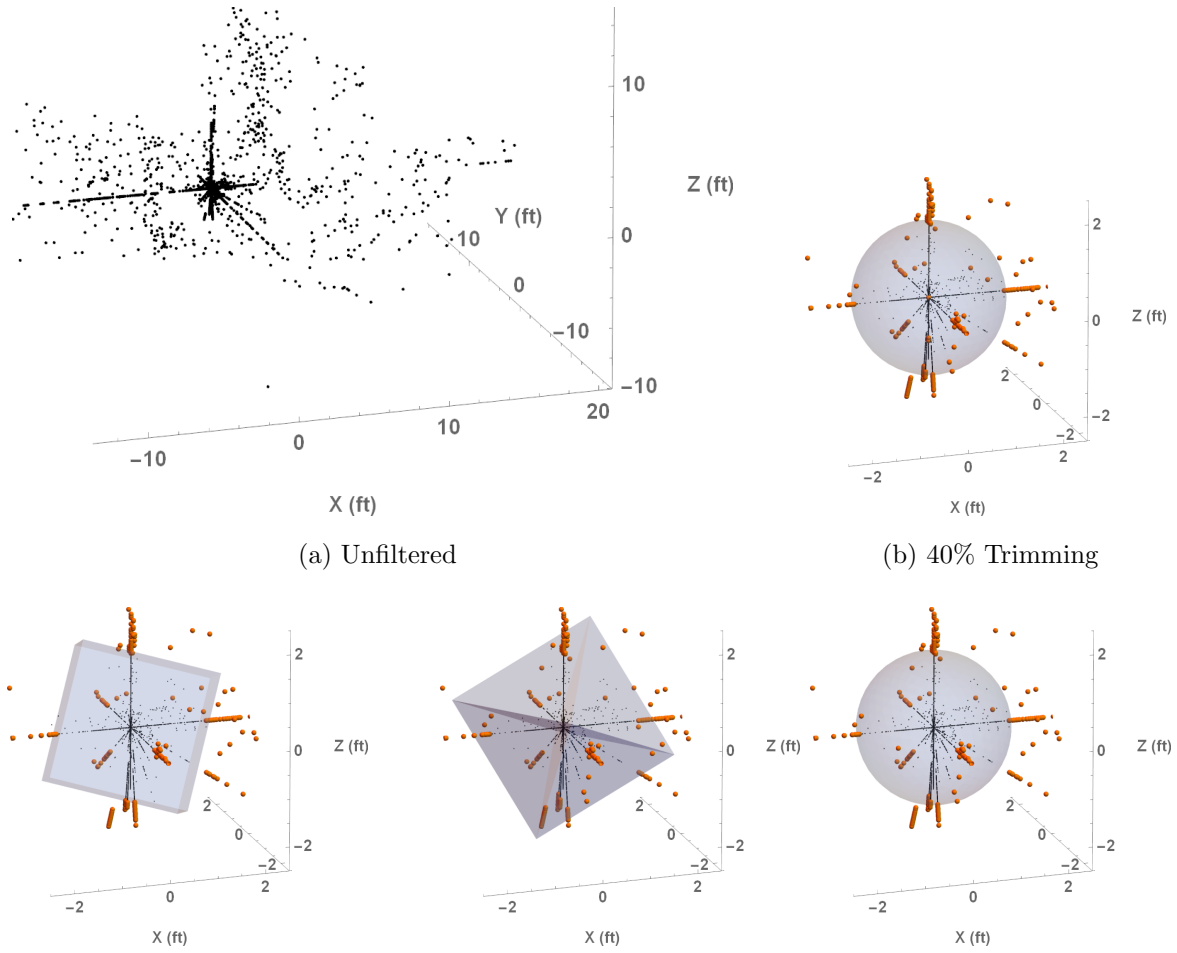


(a) By iteration

(b) By time

Figure 10.18: Displacements by iteration and time

Figure 10.19 shows the unfiltered data and various filters at the first ICP iteration, with black dots representing inliers and orange dots outliers. Without a priori point classification, it was impossible to distinguish building from non-building points.



(a) Unfiltered (b) 40% Trimming

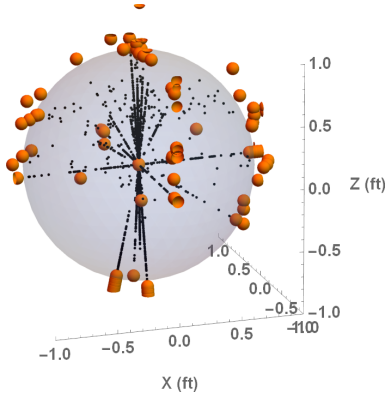
(c) Cuboid (d) Octahedron (e) Ball

Figure 10.19: Effects of various filters during first ICP iteration

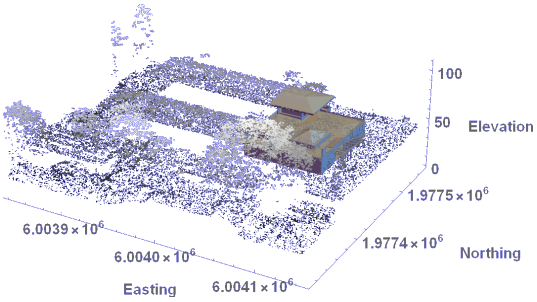
Figure 10.20 shows the final results using ICP with 40% trimming. The forward and reverse 4x4 transformation matrices (Equations 10.23 and 10.24) make the process of transforming CRSs simple and efficient.

$$\mathbf{M}_{B2E} = \left[\begin{array}{rrr|r} 0.000051 & -0.003280 & -0.000052 & 6,005,745.94 \\ 0.003280 & 0.000050 & 0.000011 & 1,975,775.89 \\ -0.000010 & -0.000052 & 0.003280 & -1,565.88 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{10.23}$$

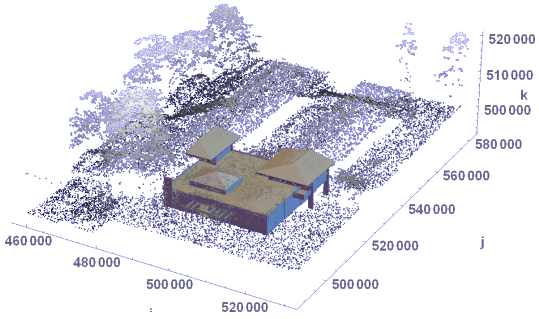
$$\mathbf{M}_{E2B} = \left[\begin{array}{rrr|r} 4.734816 & 304.765314 & -0.992131 & -630,585,613.63 \\ -304.729262 & 4.718605 & -4.807423 & 1,820,796,091.11 \\ -4.791459 & 1.066568 & 304.764177 & 27,146,212.39 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{10.24}$$



(a) Displacement vector filter at convergence



(b) Building model in SPCS coordinates (feet)



(c) Point cloud in building coordinates (millimeters)

Figure 10.20: Results of trimmed ICP at 40% trimming

10.5 Conclusion

This chapter demonstrated the effective integration of indoor and outdoor coordinate reference systems using a two-phase semi-automated approach with a building model and a georeferenced outdoor point cloud. The first phase performed coarse registration using three manually-selected anchor points and the second phase performed automated fine registration using an iterative closest point approach with filtering of residuals. The three case studies highlighted three important considerations. First, the selected filtering level should account for the estimated level of *noise* — i.e., non-building features — in

the point cloud. Severe over-filtering can result degraded performance as both datasets reach convergence and excessive amounts of "good" data get removed, as demonstrated with all four filters in Case Study 1. On the other hand, insufficient filtering can preserve undesirable non-building points resulting in slower convergence and possibly reduced accuracy. Two improvements to this approach can include a method for estimating filtering levels prior to performing ICP and an adaptive filtering method that reduces filtering levels as the model and point cloud converge. Second, the nearest neighbor metric should be used with caution when used as an estimate of actual errors. In the case of the perfect data set (Case Study 1), the mean NN displacements over-estimated the actual accuracy (or under-estimated the actual errors) by 400% for the unfiltered data and over 800% for the filtered data, with the high values partially due to the high level of precision in the numbers. However, these differences fell to values below 30% for the more realistic data set containing environmental noise... partially accounted for due to the relatively lower precision of the numbers. Finally, possible improvements to this approach can take advantage of the distinct geometric patterns in the residual plots. Knowing the building's geometry makes it possible to estimate the orientations of the residual vectors; these vectors can be used to establish thresholds where, as the data sets approach convergence, points with residual vectors that do not line up get discarded.

Chapter 11

When flat Earth meets round Earth

11.1 Introduction

Virtually all building models assume a flat Earth when they use the Cartesian coordinate system for coordinate referencing, but Earth itself has a round shape. While Earth's curvature has negligible effects on small interior spaces, it has far greater implications for very long or wide spaces such as extremely long buildings or underground cities. In those environments, the curvature of Earth can introduce significant vertical deviations from the building model. This chapter examines these curvature effects by analyzing the geometry of the ellipsoid to provide insights into the geographic limits of the flat Earth assumption. While introductory texts on surveying give simple one-line formulas for estimating vertical deviation, this analysis provides the full mathematical treatment that includes insights into how vertical and horizontal deviations vary with one another.

11.2 Simplified estimates of error

11.2.1 Vertical errors

Introductory surveying textbooks often provide simple equations for estimating the effect of Earth's curvature on vertical measurements. These formulas work perfectly fine and show negligible deviations from the more rigorous approach used in this study. However, their use of a round Earth fails to address the relevance (or irrelevance) of the varying geometry of the ellipsoid, and they do not provide insight into horizontal and vertical deviations vary together.

In his surveying manual for civil engineers, Cole (2017) provided Equation 11.1[1], which is a simplified version of Equation 11.2 provided by Allan (2008).

$$C = 0.0785K^2 \tag{11.1}$$

where C = vertical deviation (m) and K = horizontal distance (KM)

$$x = \frac{S^2}{2R} \tag{11.2}$$

where x = vertical deviation (m), S = horizontal distance (m), R = Earth's radius (m)

Allan derived Equation 11.2 by using geometric relationships of the triangle shown in Figure 11.1. To arrive at Equation 11.2, he assumed the straight line S, defined by AB', had the same distance as the arc AB and made $x^2 = 0$ in Equation 11.5. Using the mean radius of Earth ($R = 6{,}335{,}439$ m) for R results in Cole's equation.

[1]Cole actually showed the equation as $C = 0.785K^2$, which is incorrect by a factor of 10.

$$B'O^2 = AO^2 + S^2 \tag{11.3}$$

$$(R + x)^2 = R^2 + S^2 \tag{11.4}$$

$$R^2 + 2Rx + x^2 = R^2 + S^2 \tag{11.5}$$

$$x = \frac{S^2}{R} \tag{11.6}$$

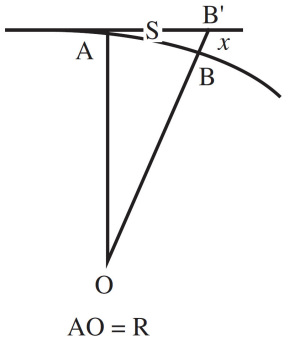


Figure 11.1: Figure used by Allan (2008)

11.2.2 Horizontal errors

The Universal Transverse Mercator (UTM) and State Plane Coordinate System (SPCS) are two widely used grid-based map projections in the United States. UTM has a maximum error limit of $\frac{1}{1000}$ for each 100 km x 100 km grid square and the SPCS for California has a limit of $\frac{1}{10000}$ for any part of its state plane maps (Snyder 1987). These specifications limit the amount of allowable horizontal error in a map due to projection distortions over wide areas. However, these technical specifications provide no insight into the magnitude of horizontal errors at a local geographic scale.

11.3 Ellipsoid-based approach

11.3.1 General approach

This study uses the World Geodetic System 1984 (WGS 84) ellipsoid — a biaxial oblate spheroid used in the Global Positioning System and many internet mapping applications — which has a semi-major axis (a) value of 6,378,137.0 m and a semi-minor axis (b) value of 6,356,752.314245 m (National Geospatial-Intelligence Agency 2014a).

Curvature describes the degree of bending of a line in a 2D plane, so the location where the ellipsoid surface deviates the most from a tangent plane will occur around the point of maximum curvature. For the oblate spheroid, this occurs along a meridian line at the end point of the major axis, a, at the equator. Since an oblate spheroid is simply a 2D ellipse rotated about the polar axis, the solution to this problem simply involves determining the radius of curvature at that point and using geometry to calculate the relevant deviations. The following calculations use the cross-section of the ellipsoid at $y = 0$ as shown in Figure 11.2.

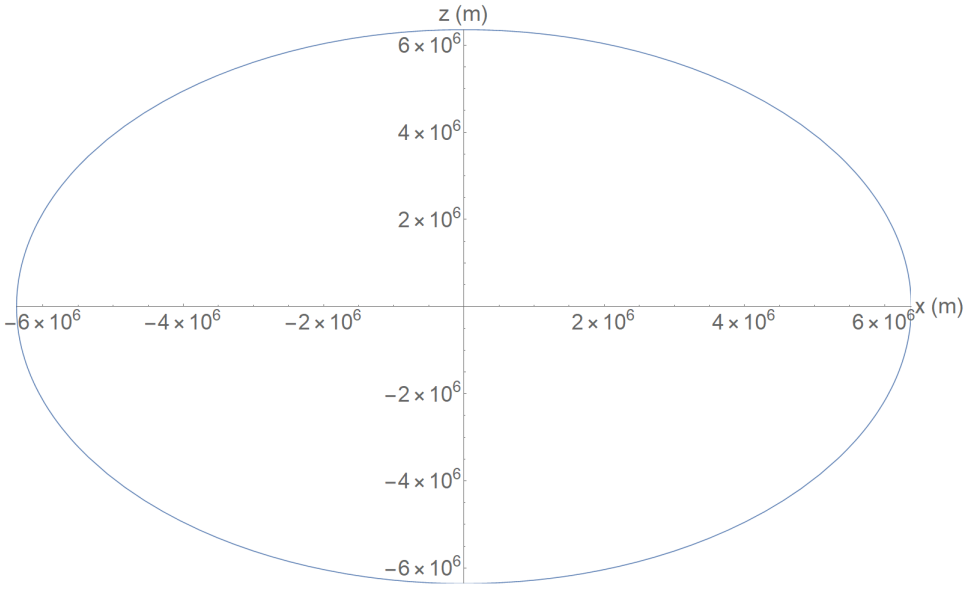


Figure 11.2: Cross-section of the WGS 84 ellipsoid at $y = 0$

11.3.2 Modifying the ellipse equation

Since the point of maximum curvature occurs at the equator, the horizontal direction will run along the z-axis and the vertical along the x-axis. Re-assigning variables for the general equation of an ellipse (Equation 11.7) and observing that the ellipse is centered at the origin $(0, 0, 0)$ results in Equation 11.8, with the x- and z-axes paired with the semi-major and semi-minor axes.

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \tag{11.7}$$

$$\frac{x^2}{a^2} + \frac{z^2}{b^2} = 1 \tag{11.8}$$

Rearranging Equation 11.8 to make x a function of z results in Equation 11.9.

$$x = a\sqrt{1 - \frac{z^2}{b^2}} \tag{11.9}$$

11.3.3 Finding the maximum curvature

Even though the first derivative of x produces the slope of the tangent line at (z, x), the second derivative does not produce the curvature. Curvature, κ, requires use of the curvature equation (Equation 11.10), modified here to reflect x as a function of z.

$$\kappa = \frac{|x''|}{\left[1 + (x')^2\right]^{3/2}} \tag{11.10}$$

Maximum curvature occurs at the point where the value of κ reaches a maximum. A plot of κ for an ellipse based on WGS 84 parameters results in Figure 11.3, which shows that maximum curvature occurs at the equator where $z = 0$.

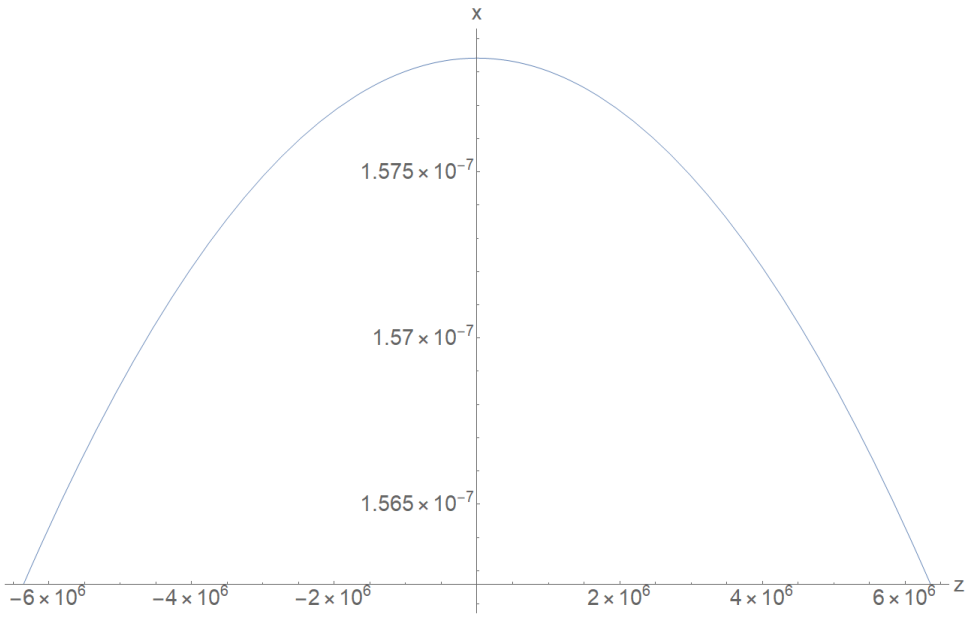


Figure 11.3: Plot of WGS 84 curvature at $y = 0$ from pole to pole

Maximum curvature has the value of $1.578523 \times 10^{-7} m^{-1}$ as shown in Equation 11.11.

$$\kappa_{max} = \kappa_{z=0} = 1.578523 \times 10^{-7} m^{-1} \tag{11.11}$$

11.3.4 Finding the radius of curvature

Using the radius of curvature (ρ) simplifies the geometric analysis of the ellipse. The radius of curvature represents the equivalent radius of a circle that has the same curvature for any point on a curve as shown in Figure 11.4 and is expressed in mathematical form as Equation 11.12. This method is equivalent to the methods used by Allan (2008) and Cole (2017) and produces more conservative results, i.e., larger vertical deviations, compared to direct analysis of an ellipse at $z = 0$ and $x = a$ or $x = -a$.

$$\rho = \frac{1}{\kappa} = \frac{\left[1 + (x')^2\right]^{3/2}}{|x''|} \tag{11.12}$$

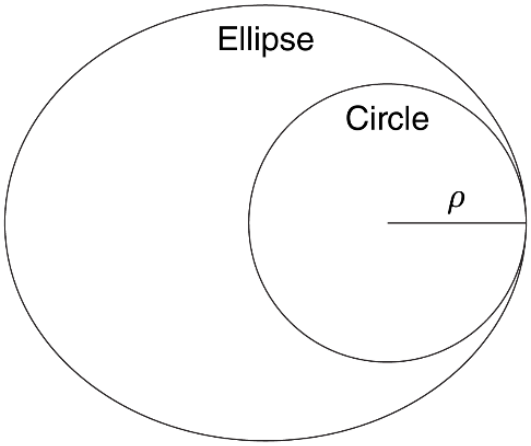


Figure 11.4: Radius of curvature, ρ, for a point on an ellipse

11.3.5 Minimum radius of curvature

Since curvature, κ, and radius of curvature, ρ, have an inverse relationship, the maximum curvature will result in a circle with the smallest radius. The minimum radius of curvature occurs at the equator, $z = 0$, producing a minimum ρ value of 6,335,439 m as shown in Equation 11.13.

$$\rho_{min} = \rho_{z=0} = 6{,}335{,}439\,\text{m} \tag{11.13}$$

This translates to the minimum circle with a radius (R_{min}) having the same value as ρ_{min} as shown in Equation 11.14.

$$R_{min} = \rho_{min} \tag{11.14}$$

Assessing the worst-case effects of Earth's curvature involves finding the deviation of a tangent line from this equivalent circle.

11.4 Using flat Earth as a baseline

The first approach to quantifying the effects of Earth's curvature involves placing an observer on the flat tangent line and measuring deviations of the circular arc below as shown in Figure 11.5. The goal of this analysis involves finding deviations of the circular arc from the tangent line — Δz and Δx, respectively, keeping in mind that z represents the horizontal direction since the observer is standing at the equator.

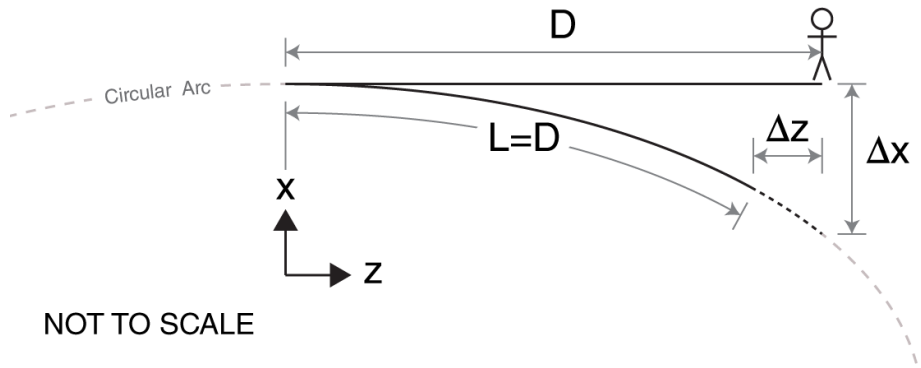


Figure 11.5: Measuring deviations relative to a position on the flat Earth

This analysis uses a (z, x) axis convention for clarity where the z-axis runs in the horizontal direction and x in the vertical. The center of the circle has coordinates (0, 0), the tangent line intersects the circle at coordinates (0, R), and the observer stands at coordinates (D, R) corresponding to a distance D along the tangent line.

11.4.1 Equation for the vertical offset, Δx

The vertical offset, Δx, estimates how far the curve will deviate vertically from the tangent plane at the observation point; that is, how far the curve would drop below the tangent line if the observer looked straight down.

Find the equation for z as a function of x

Equation 11.15 shows the general equation of a circle using the xz coordinate convention used in this study.

$$(x-h)^2 + (z-k)^2 = R^2 \tag{11.15}$$

Centering the circle at the origin ($h = 0, k = 0$) and rearranging Equation 11.15 to express x in terms of z results in Equation 11.16.

$$x = \sqrt{R^2 - z^2} \tag{11.16}$$

Find the equation for Δx at various z values

The vertical deviation of the curve is simply the difference between R (i.e., the x-coordinate at the tangent point) and the x-coordinate of the curve for any distance D along the tangent line, calculated using Equation 11.22.

$$\Delta x = R - x = R - \sqrt{R^2 - x^2} \tag{11.17}$$

$$\Delta x = R - \sqrt{R^2 - D^2} \quad \text{at} \quad z = D \tag{11.18}$$

11.4.2 Equation for the horizontal offset, Δz

The horizontal offset, z, estimates the horizontal difference between a distance D along the curve and the same distance along the tangent plane.

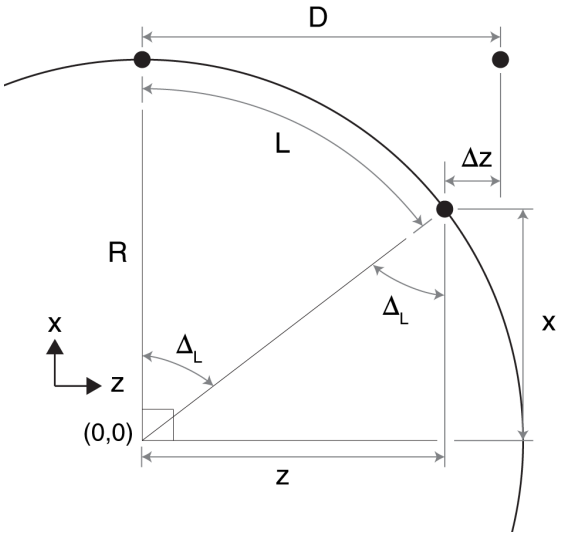


Figure 11.6: Geometric relationships for finding Δz

Find the central angle, Δ_D, for an arc L with length D

Use the arc length, L, to find its equivalent central angle, L, using Equation 11.19, recalling that the central angle for a full circle is simply 2π radians which is found by dividing the circumference by the radius. For any distance D along the curve, Equation 11.19 becomes Equation 11.27.

$$\Delta_L = \frac{L}{R} \quad \text{(in radians)} \tag{11.19}$$

$$\Delta_D = \frac{D}{R} \quad \text{(in radians)} \tag{11.20}$$

Find the equation for z in terms of Δ_D

The coordinate z can be solved using basic trigonometry.

$$z = R \sin \Delta_D \tag{11.21}$$

Find the equation for Δz

The horizontal offset, Δz, is the difference between the distance D along the tangent line and the z-coordinate for an arc of length D. For this analysis, Δz will always have a positive value reflecting the extra horizontal distance introduced by the tangent line compared to the curve.

$$\Delta z = D - z \tag{11.22}$$

11.4.3 Equation for arc length offset, ΔD

The arc length offset, ΔD, describes the distance that needs to be added to arc length D to reach a point on the curve located directly below tangent distance D as shown in Figure 11.7. Figure 11.8 provides the geometric relationships for finding ΔD.

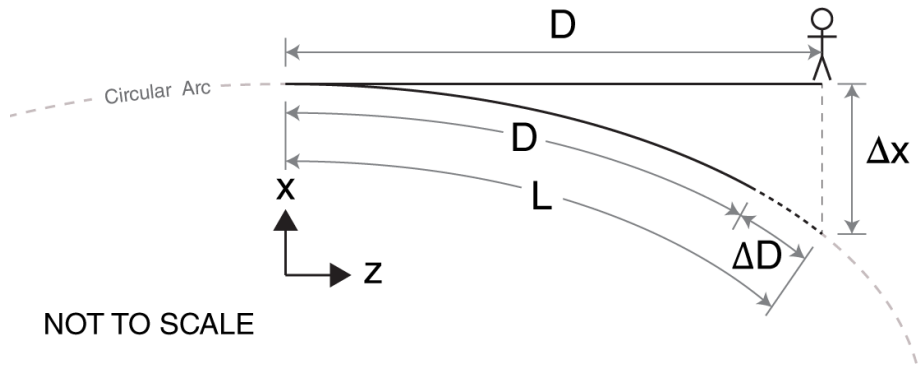


Figure 11.7: Overview of distance D along flat and round Earths

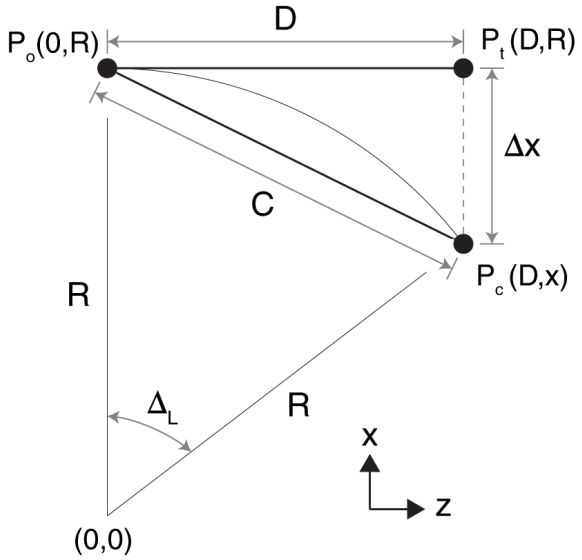


Figure 11.8: Geometry of chord, C

Find the equation for the chord, C

Chord C extends from the tangent point to the point on the curve with arc length D. Equation 11.23 provides the equation for C as a function of Δx from Equation 11.22 using the Pythagorean theorem.

$$C = \sqrt{D^2 + \Delta x^2} \tag{11.23}$$

Find the central angle, ΔL

The central angle (ΔL) can be found using the law of cosines for an isosceles triangle.

$$\cos \Delta_L = \frac{2R^2 - C^2}{2R^2} \tag{11.24}$$

$$\Delta_L = \arccos\left(\frac{2R^2 - C^2}{2R^2}\right) \quad \text{(in radians)} \tag{11.25}$$

Find the extended arc length, L, and excess distance, ΔD

The central angle and radius provide all the required information for finding arc length L whose end point has a z-coordinate of D as shown in Equation 11.26. Subtracting the equivalent curved distance of D from L provides the excess distance (ΔD) as shown in Equation 11.27.

$$L = \Delta_L R \tag{11.26}$$

$$\Delta_D = L - D \tag{11.27}$$

11.4.4 Summary of deviations from the flat Earth

Table 11.1 shows the results of the flat Earth analysis for distances ranging from 100 m to 10 km. Allan's equation provides vertical deviation values nearly identical to ones produced using this study's more rigorous approach. This was due to the ability to use small angle approximations made possible by the relatively large value of R compared to D. Cole's equation also provides a relatively accurate estimate, although with fewer significant digits. Therefore, as a rule-of-thumb, a model with an acceptable vertical error of 2 cm (or less than one inch) can span approximately 1 km, i.e., spanning 500 m in two opposite directions... dimensions within which most indoor spaces fall. Increasing the acceptable vertical error to 8 cm (or slightly over three inches) results in a maximum horizontal span of 2 km. Horizontal deviations remarkably remain negligible until horizontal distances exceed 5 km, with the deviation remaining below 1 cm at a distance of 10 km. This last observation perhaps explains why grid maps such as UTM and state plane maps can span such great horizontal distances; however, that same assumption can not be extended to the vertical component.

Table 11.1: round Earth deviations from flat Earth reference at different distances along the flat surface

	This Study			Allan (2008)		Cole (2017)
D (m)	Horizontal (m)	Curve (m)	Vertical (m)	Vertical (m,R_{min})	Vertical (m, R_{mean})	Vertical (m, R_{mean})
100	0.00000000	0.00000395	0.00078921	0.00078921	0.00078467	0.000785
500	0.00000052	0.00001986	0.01973028	0.01973028	0.01961660	0.019625
1,000	0.00000415	0.00004050	0.07892113	0.07892113	0.07846650	0.078500
1,500	0.00001401	0.00006269	0.17757254	0.17757254	0.17655000	0.176625
2,000	0.00003322	0.00008723	0.31568452	0.31568452	0.31386600	0.314000
5,000	0.00051905	0.00032706	1.97302854	1.97302823	1.96166000	1.962500
10,000	0.00415236	0.00143270	7.89211784	7.89211292	7.84665000	7.850000

11.5 Using round Earth as a baseline

An alternative approach to assessing the effects of Earth's curvature involves placing the observer on the curve (i.e., ellipsoid) and observing deviations along the tangent plane from that position as illustrated in Figure 11.9. In this case, distance D would be measured along the arc resulting in Cartesian coordinate discrepancies of $\Delta z'$ and $\Delta x'$ as shown in Figure 11.10. The prime ($'$) notation denotes the measurement of D along the arc, not the tangent line, where D_T denotes the distance D along the tangent line. Since horizontal and vertical deviations are defined relative to the ellipsoid's surface, $\Delta z'$ and $\Delta x'$ in the xyz coordinate system are transformed into local ijk coordinates to reflect the observer's perspective.

11.5.1 Find equations for offsets $\Delta x'$ and $\Delta z'$

Finding the xz coordinate system offsets involves finding the central angle (ΔD) using Equation 11.28 and using the value of R with the Pythagorean theorem as shown in Equations 11.29 to find x' and z'. $\Delta z'$ is the difference between D_T and the z coordinate

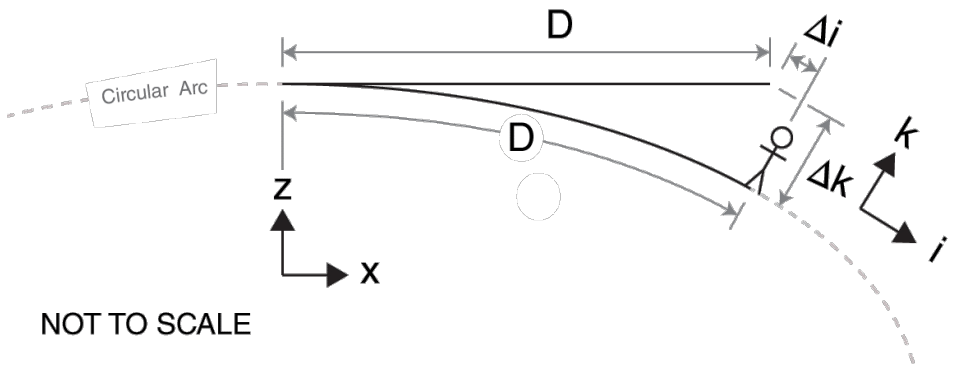


Figure 11.9: Measuring deviations relative to a position on the round Earth

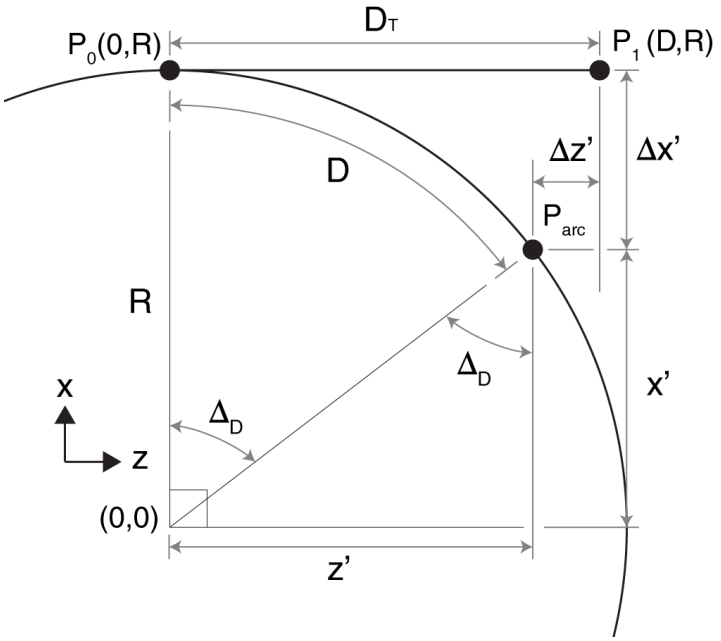


Figure 11.10: round Earth offsets, $\Delta x'$ and $\Delta z'$

for the arc's end point and $\Delta x'$ between the radius (R) and x coordinate for the arc's end point as shown in Equation 11.30, where $D_T = D$.

$$\Delta_D = \frac{D}{R} \quad \text{in radians} \tag{11.28}$$

$$x' = R\cos\Delta D; z' = R\sin\Delta D \tag{11.29}$$

$$\Delta x' = R - x'; \Delta z' = D - z' \tag{11.30}$$

11.5.2 Transforming xz offsets into local ik coordinates

Transforming $\Delta x'$ and $\Delta z'$ into local ik coordinate values will show displacements from the observer's perspective on the curve. This can be done by assembling the xz offsets in the form of a vector $\mathbf{F}$ (Equation 11.33) from P_{arc} to P_1 (Figure 11.10) and projecting this vector onto ik unit vectors $\mathbf{i}$ and $\mathbf{k}$ (Equation 11.32). Since z represents the horizontal direction, it will map to $\mathbf{i}$.

$$\mathbf{F} = \begin{bmatrix} \Delta z' \\ \Delta x' \end{bmatrix} \tag{11.31}$$

$$\mathbf{i} = \begin{bmatrix} i_z \\ i_x \end{bmatrix} = \begin{bmatrix} \sin\Delta D \\ \cos\Delta D \end{bmatrix}; \quad \mathbf{k} = \begin{bmatrix} j_z \\ j_x \end{bmatrix} = \begin{bmatrix} -\cos\Delta D \\ \sin\Delta D \end{bmatrix} \tag{11.32}$$

Assembling $\mathbf{i}$ and $\mathbf{k}$ into transformation matrix $\mathbf{M}$ and multiplying $\mathbf{F}$ by $\mathbf{M}$ produces the deviation vector $\mathbf{G}$ in local ij coordinates.

Table 11.2: Flat Earth deviations from round Earth reference at different distances along the curved surface

	This Study		Allan (2008)		Cole (2017)
D (m)	Δi (m)	Δk (m)	Vertical (m,R_{min})	Vertical (m, R_{mean})	Vertical (m, R_{mean})
100	-0.00000001	0.00078921	0.00078921	0.00078467	0.000785
500	-0.00000104	0.01973028	0.01973028	0.01961660	0.019625
1,000	-0.00000830	0.07892113	0.07892113	0.07846650	0.078500
1,500	-0.00002803	0.17757254	0.17757254	0.17655000	0.176625
2,000	-0.00006644	0.31568451	0.31568452	0.31386600	0.314000
5,000	-0.00103809	1.97302792	1.97302823	1.96166000	1.962500
10,000	-0.00830472	7.89210801	7.89211292	7.84665000	7.850000

$$\mathbf{M} = \begin{bmatrix} i_z & i_x \\ j_z & j_x \end{bmatrix} \tag{11.33}$$

$$\mathbf{G} = \begin{bmatrix} \Delta_i \\ \Delta_k \end{bmatrix} = \mathbf{M} \cdot \mathbf{F} \tag{11.34}$$

11.5.3 Summary of deviations from the round Earth

Table 11.2 shows deviations when using the round Earth as a baseline. Note that vertical deviations remain nearly identical to values found with a flat Earth baseline; however, horizontal deviations have twice the value though still negligible. Both Allan and Cole's equations closely match the more rigorously derived values, as expected. Between the flat Earth baseline and round Earth baseline, Allan's analysis more closely resembles the round Earth analysis since his vertical deviation runs orthogonal to the surface of Earth.

11.6 Conclusion

The two exercises in this chapter showed that Earth's curvature has a significantly greater impact on vertical measurements than on horizontal measurements. While simplified formulas for finding vertical deviations can be found in introductory surveying textbooks, they often go no further in discussing the geometry involved nor do they relate vertical to horizontal deviations. Additionally, while maximum error limits for mapping products such as UTM and SPCS maps provide some insights into the maximum *horizontal* deviations, they do so for projections of large areas... not smaller local areas, such as those encountered with indoor spaces.

This study provided a simplified in-depth assessment of these deviations with a special emphasis on comparing horizontal and vertical deviations over distance traveled along the flat or curved surface. The vertical deviations validated the simplified formulas provided by Allan (2008) and Cole (2017), showing that a more rigorous analysis provided marginal — if not negligible — improvement. However, it showed a stark contrast between vertical and horizontal deviations, with vertical deviations reaching the decimeter level between one and two kilometers while horizontal deviations remained sub-millimeter. This single observation can serve as a rule-of-thumb on the geographic limits of indoor maps before the flat Earth (Cartesian coordinate) assumption begins to break down.

Aside from providing some conceptual insights, this study has limited practical value for three reasons. First, the model of a tangent plane on a sphere — a form of orthographic projection — is seldom used in practical mapping. The extremely large vertical deviations at the upper limits would be mitigated by map projections while the near-zero horizontal deviations would grow. Second, long sight distances in the real world seldom go in a straight line; they curve slightly earthward due to diffraction. Finally, gravity's impact on surveying equipment will introduce gradual deviations during both the construction

of the indoor space and follow-on surveys, moving some of the theoretical errors into the realm of uncertainties.

Part V

Conclusion

Chapter 12

Concluding remarks

12.1 State of 3D indoor mapping

This dissertation reviewed the current state of research for generating 3D indoor maps and models from point cloud data. Areas explored included the strengths and weaknesses of SfM-MVS photogrammetry; the gridding of indoor space for the long-term sustainability of maps; the automatic segmentation of floors, ceilings, and walls from point clouds of rectangular rooms; indoor-outdoor CRS integration using a point cloud and a building model; and an investigation on the limits of the flat Earth assumption for indoor spaces. The review of existing research in conjunction with studies for this dissertation indicated that 3D indoor mapping still remains at an early stage of development. This is especially true from a *cartographic* stand point that would examine issues such as the symbolic representation of indoor spaces, multiple indoor levels-of-detail, and truly integrated indoor-outdoor spatial referencing

12.2 Limits of current research

A review of current methods indicated that roughly half of the studies on indoor mapping and modeling stopped at point cloud segmentation, indicating the difficulty posed by heterogeneous indoor environments and hard-to-capture objects such as windows and mirrors. Of the studies that moved beyond point cloud segmentation, many generated 2.5D geometric models by extruding footprints of projected wall segments on the floor. However, at least one study attempted to derive greater fidelity by using an adaptive approach. Additionally, few studies on indoor mapping and modeling provided formal treatment of coordinate referencing.

12.3 Future opportunities in 3D indoor mapping

Many areas of opportunity exist for future research on 3D indoor mapping and modeling. Point cloud segmentation and geometric modeling from point clouds will continue to occupy an important area of research interest, especially with the use of artificial intelligence. These newer AI-enabled methods will prove beneficial as the availability of 3D indoor point cloud data continues to grow, fueled by a proliferation of inexpensive and miniaturized 3D LiDAR scanners — some of which may soon fit on a smart phone. The cartography of indoor space represents another research area that will likely have far-reaching impacts on the usability of future indoor maps, notably with studies on indoor LOD and indoor-outdoor coordinate integration. Nearly all research to date on 3D indoor maps have assumed a single level of cartographic detail, as reflected by LOD 4 in CityGML. Research on indoor-outdoor CRS integration, another cartographic issue, will also continue to provide opportunities for finding practical ways to seamlessly integrate indoor and outdoor systems. Finally, the growing availability of 3D maps of indoor

spaces will drive research on uses for those maps in such areas as smart cities, emergency management, commerce, and augmented and virtual reality.

Bibliography

Ackeret, James, et al. 1990. *Handbook for Transformation of Datums, Projections, Grids, and Common Coordinate Systems*. DTIC Document. http://www.dtic.mil/dtic/tr/fulltext/u2/a478730.pdf.

Adams, Timothy. 2005. "Using SVG and XSLT to Display Visually Geo-Referenced XML". In *Visualizing Information Using SVG and X3D*, 256–265. London: Springer London. ISBN: 978-1-85233-790-2. doi:10.1007/1-84628-084-2_12.

Adán, Antonio, and Daniel Huber. 2011. "3D Reconstruction of Interior Wall Surfaces under Occlusion and Clutter". In *Visualization and Transmission 2011 International Conference on 3D Imaging, Modeling, Processing*, 275–281. doi:10.1109/3DIMPVT.2011.42.

— . 2010. *Reconstruction of Wall Surfaces Under Occlusion and Clutter in 3D Indoor Environments* CMU-RI-TR-10-12.

Adán, Antonio, and Enrique Valero. 2014. "Furnished Interiors Modeling with Laser Scanners and RFID". In *2014 2nd International Conference on 3D Vision*, 2:95–102. doi:10.1109/3DV.2014.50.

Agisoft LLC. 2013. "Agisoft PhotoScan Tips and Tricks". Visited on 12/26/2013. http://www.agisoft.com/support/tips-tricks/.

— . 2016. "Agisoft PhotoScan User Manual, Professional Edition, Version 1.2." Visited on 09/23/2017. http://www.agisoft.com/pdf/photoscan-pro_1_2_en.pdf.

Akin, Omer. 2009. "CAD/GIS Integration". In *CAD and GIS Integration*, 51–71. Auerbach Publications. ISBN: 978-1-4200-6805-4.

Allan, A. L. 2008. *Principles of Geospatial Surveying*. Dunbeath: Whittles Publishing. ISBN: 9781420073461.

Armeni, Iro, et al. 2016. "3D Semantic Parsing of Large-Scale Indoor Spaces". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1534–1543. doi:10.1109/CVPR.2016.170.

Autodesk. 2011. *AutoCAD 2012 DXF Reference*. Visited on 01/27/2017. http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_enu.pdf.

Baghmisheh, Behnam B. 2017. *Chip-Scale Lidar*. Technical Report UCB/EECS-2017-4. University of California at Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-4.html.

Barnes, Mark, and Ellen L. Finch. 2008. *COLLADA – Digital Asset Schema Release 1.5.0 Specification*. The Khronos Group Inc., Sony Computer Entertainement Inc. Visited on 01/28/2017. https://www.khronos.org/files/collada_spec_1_5.pdf.

Barnes, Peter, and Nigel Davies. 2015. *BIM in Principle and in Practice*. Second edition. London: ICE Publishing. ISBN: 978-0-7277-6092-0.

Becker, Matthias. 2011. "Geodesy". In *Springer Handbook of Geographic Information*, ed. by Wolfgang Kresse and David M. Danko, 95–117. Berlin: Springer. ISBN: 978-3-540-72678-4. doi:10.1007/978-3-540-72680-7_8.

Bedrick, Jim. 2013. "A Level of Development Specification for BIM Processes: AECbytes Viewpoint". Visited on 12/24/2015. http://www.aecbytes.com/viewpoint/2013/issue_68.html.

Bentley Systems. 2016. "MicroStation Wiki". Visited on 03/10/2017. https://communities.bentley.com/products/microstation/w/microstation__wiki.

Bernhardsen, Tor. 2002. *Geographic Information Systems: An Introduction, 3rd Edition*. John Wiley & Sons. ISBN: 978-0471419686.

Bertin, Jacques. 2010. *Semiology of Graphics: Diagrams, Networks, Maps*. 1st ed. Trans. by William J. Berg. Redlands: ESRI Press. ISBN: 978-1-58948-261-6.

Besl, Paul J., and Neil D. McKay. 1992. "Method for Registration of 3-D Shapes". In *Proceedings SPIE 1611, Sensor Fusion IV: Control Paradigms and Data Structures*, 1611:586–606. Boston, MA. doi:10.1117/12.57955.

Bhatla, Ankit, et al. 2012. "Evaluation of Accuracy of As-Built 3D Modeling from Photos Taken by Handheld Digital Cameras". *Automation in construction* 28:116–127. ISSN: 0926-5805.

Billen, Roland, et al. 2014. *3D City Models and Urban Information: Current Issues and Perspectives*. EDP Sciences. ISBN: 978-2-7598-1153-3. doi:10.1051/TU0801/201400001.

Board, Christopher. 1967. "Maps as Models". In *Models in Geography*, 671–725. London: Methuen.

Bolles, Robert C., and Martin A. Fischler. 1981. "A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data". In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, 637–643. IJCAI'81. San Francisco: Morgan Kaufmann Publishers Inc.

Botsch, Mario, ed. 2010. *Polygon Mesh Processing*. Natick: A K Peters/CRC Press. ISBN: 978-1-56881-426-1.

British Museum. N.d.(a). *"L'Architecte Au Plan" from 2120 BC*.

— . N.d.(b). *The Map of the World. c. 6th Century BC.*

Brou, Philippe. 1984. "Using the Gaussian Image to Find the Orientation of Objects". *The International Journal of Robotics Research* 3 (4): 89–125. ISSN: 0278-3649. doi:10.1177/027836498400300406.

Brüggemann, Heinz, and Anke Liebig. 2000. "Geographic Data Infrastructure for North-Rhine Westphalia (GDI NRW)". In *Proceedings of The Spatial Information Society: Shaping the Future, 6th EC-GIS Workshop, Lyon, France, June 28*, vol. 30.

Brutzman, Don. 2012. "X3D Graphics Support for Computer Aided Design (CAD)". Naval Postgraduate School. Visited on 01/28/2017. http://x3dgraphics.com/slidesets/X3dForAdvancedModeling/ComputerAidedDesignInterchangeProfile.pdf.

Buckner, R. B. 1995. "Plane Coordinate Systems". In *The Surveying Handbook*, 454–485. New York: Springer US. ISBN: 978-1-4613-5858-9.

Budiharto, Widodo, Achmad Jazidie, and Djoko Purwanto. 2010. "Indoor Navigation Using Adaptive Neuro Fuzzy Controller for Servant Robot", 1:582–586. IEEE.

Budroni, Angela. 2013. "Automatic Model Reconstruction of Indoor Manhattan-World Scenes from Dense Laser Range Data". PhD thesis, Institute for Photogrammetry, University of Stuttgart.

Budroni, Angela, and Jan Boehm. 2009. "Toward Automatic Reconstruction of Interiors from Laser Data". In *ISPRS Archives Volume XXXVIII 5 W1, 2009*. Trento, Italy. http://www.isprs.org/proceedings/XXXVIII/5-W1/.

Bugayevskiy, L. M., and John Snyder. 1995. *Map Projections: A Reference Manual*. Boca Raton: CRC Press. ISBN: 978-0-7484-0304-2.

buildingSMART. 2017a. "BCF Intro". Visited on 01/27/2017. http://www.buildingsmart-tech.org/specifications/bcf-releases/bcf-intro.

— . 2016. "History". Visited on 01/27/2017. http://buildingsmart.org/about/about-buildingsmart/history/.

— . 2015. "IFC4 - Addendum 1 [Final Standard]". Visited on 01/27/2017. http://www.buildingsmart-tech.org/ifc/IFC4/Add1/html/.

— . 2017b. "Model View Definition Summary". Visited on 01/27/2017. http://www.buildingsmart-tech.org/specifications/ifc-view-definition/summary.

— . 2017c. "Related Specifications Summary". Visited on 01/27/2017. http://www.buildingsmart-tech.org/specifications/related-specifications.

Bureau, R. 1946. "Altimétrie Des Nuages Par Impulsions Lumineuses". *Météorologie*: 292–301.

Burggraf, David. 2015. "OGC KML 2.3". Visited on 01/28/2017. http://docs.opengeospatial.org/is/12-007r2/12-007r2.html.

Buyukaslih, I., U. Isikdag, and S. Zlatanova. 2013. "Exploring the Processes of Generating LOD (0-2) CityGML Models in Greater Municipality of Istanbul". *8th 3DGeoInfo Conference & WG II/2 Workshop, Istanbul, Turkey, 27–29 November 2013, ISPRS Archives Volume II-2/W1*.

Chang, Yi-Hong, Tyng-Ruey Chuang, and Hao-Chuan Wang. 2004. "Adaptive Level-of-Detail in SVG". In *3rd Annual Conference on Scalable Vector Graphics*. Tokyo, Japan.

Chen, Jorge. 2006. *Carlsbad Caverns*.

— . 2018. "Grid Referencing of Buildings". In *Proceedings of the 14th Conference on Location Based Services*. Zurich, Switzerland.

— . 2011. *Kyoto Imperial Palace Courtyard*.

Chen, Jorge, and Keith C. Clarke. 2017. "Modeling Standards and File Formats for Indoor Mapping". In *Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1*, 268–275. Porto, Portugal. ISBN: 978-989-758-252-3. doi:10.5220/0006364202680275.

— . 2016. "Rapid 3D Modeling Using Photogrammetry Applied to Google Earth". In *The 21st International Research Symposium on Computer-Based Cartography and GIScience*, 14–27. Albuquerque, New Mexico.

Cheng-siang, Chen. 1978. "The Historical Development of Cartography in China". *Progress in Geography* 2 (1): 101–120. ISSN: 0556-1892. doi:10.1177/030913257800200106.

Chetverikov, Dmitry, et al. 2002. "The Trimmed Iterative Closest Point Algorithm". In *Object Recognition Supported by User Interaction for Service Robots*, 3:545–548. doi:10.1109/ICPR.2002.1047997.

Choi, Changhyun, Alexander J. B. Trevor, and Henrik I. Christensen. 2013. "RGB-D Edge Detection and Edge-Based Registration". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1568–1575. doi:10.1109/IROS.2013.6696558.

Clarke, Keith C. 1995. *Analytical and Computer Cartography*. 2nd ed. Prentice Hall series in geographic information science. Englewood Cliffs: Prentice Hall. ISBN: 978-0-13-033481-7.

— . 2014. *Maps & Web Mapping*. Pearson. ISBN: 978-0-321-89682-7.

— . 2013. "What Is the World's Oldest Map?" *The Cartographic Journal* 50 (2): 136–143. ISSN: 0008-7041. doi:10.1179/0008704113Z.00000000079.

Clarke, Keith C., and John G. Cloud. 2000. "On the Origins of Analytical Cartography". *Cartography and Geographic Information Science* 27 (3): 195–204. doi:10.1559/152304000783547821.

Cole, George M. 2017. *California Civil Surveying Reference Manual*. Professional Publications, Inc. ISBN: 978-1-59126-494-1.

Davidson, Ronald A. 2008. "Recalcitrant Space: Modeling Variation in Humanistic Geography". *Journal of Cultural Geography* 25 (2): 161–180. ISSN: 0887-3631. doi:10.1080/08873630802214172.

Díaz-Vilariño, Lucía, et al. 2015. "3D Modeling of Building Indoor Spaces and Closed Doors from Imagery and Point Clouds". *Sensors* 15 (2): 3491–3512. doi:10.3390/s150203491.

Dokmanić, Ivan, et al. 2013. "Acoustic Echoes Reveal Room Shape". *Proceedings of the National Academy of Sciences* 110 (30): 12186–12191. doi:10.1073/pnas.1221464110.

Drewes, Hermann. 2009. "Reference Systems, Reference Frames, and the Geodetic Datum". In *Observing Our Changing Earth*, ed. by Michael G. Sideris, 3–9. Berlin: Springer. ISBN: 978-3-540-85425-8.

Drewes, Hermann, et al. 2007. "On the ITRS Datum Specifications". Paris, France. Visited on 11/14/2015. http://www.iers.org/IERS/EN/Organization/Workshops/Workshop2007ParisProgramme.html.

Duh, Jiunn-Der. 2012. "Light Detection and Ranging". Portland State University. http://web.pdx.edu/~jduh/courses/geog493f12/Week04.pdf.

Ertiö, Titiana-Petra, and Akshay Bhagwatwar. 2017. "Citizens as Planners: Harnessing Information and Values from the Bottom-Up". *International Journal of Information Management* 37 (3): 111–113. ISSN: 0268-4012. doi:10.1016/j.ijinfomgt.2017.01.001.

Esri. 2008. *The Multipatch Geometry Type*. Visited on 01/27/2017. https://www.esri.com/library/whitepapers/pdfs/multipatch-geometry-type.pdf.

— . 2016. "Types of Geodatabases". Visited on 01/28/2017. http://desktop.arcgis.com/en/arcmap/10.3/manage-data/geodatabases/types-of-geodatabases.htm.

Fairman, H. W. 1949. "Town Planning in Pharaonic Egypt". *The Town Planning Review* 20 (1): 32–51. ISSN: 0041-0020. JSTOR: 40101924.

Falkingham, Peter L., Karl T. Bates, and James O. Farlow. 2014. "Historical Photogrammetry: Bird's Paluxy River Dinosaur Chase Sequence Digitally Reconstructed as It Was Prior to Excavation 70 Years Ago". *PLOS ONE* 9 (4). doi:10.1371/journal.pone.0093247.

Fischler, Martin A., and Robert C. Bolles. 1981. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Communications of the ACM* 24, no. 6 (): 381–395. ISSN: 0001-0782. doi:10.1145/358669.358692.

Florida Center for Instructional Technology. N.d. *Houses of Parliament, Westminster; Plan of Principal Floor*. http://etc.usf.edu/clipart/74200/74281/74281_parli.htm.

Ganovelli, Fabio, et al. 2015. *Introduction to Computer Graphics: A Practical Learning Approach*. Boca Raton: CRC Press. ISBN: 978-1439852798.

Garfield, Leanna. 2015. "The CIA's EarthViewer Was Basically the Original Google Earth". Visited on 01/28/2017. http://www.businessinsider.com/the-cias-earthviewer-was-the-original-google-earth-2015-11.

GDI-DE. 2016. "SIG 3D Der GDI-DE Geodateninfrastruktur Deutschland". https://www.sig3d.org/index.php?catid=2&themaid=1058292099&language=en.

Geodatastyrelsen. 2014. "Geodatastyrelsen Giver de Unge Hele Danmark i 3D". Visited on 01/28/2017. http://gst.dk/nyheder/nyhedsarkiv/2014/maj/danmark-downloadet-220000-gange-i-3d/.

Geroimenko, Vladimir, and Larissa Geroimenko. 2006. "SVG and X3D: New XML Technologies for 2D and 3D Visualization". In *Visualizing the Semantic Web*, ed. by Vladimir Geroimenko DSc MSc and Chaomei Chen BSc MSc, 124–133. London: Springer London. ISBN: 978-1-85233-976-0. doi:10.1007/1-84628-290-X_7.

Giudice, Nicholas A., Lisa A. Walton, and Michael Worboys. 2010. "The Informatics of Indoor and Outdoor Space: A Research Agenda". In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, 47–53. ISA '10. New York: ACM. ISBN: 978-1-4503-0433-7. doi:10.1145/1865885.1865897.

Gonzalez, Rafael C., and Richard E. Woods. 2002. *Digital Image Processing*. 2nd ed. Upper Saddle River: Prentice Hall. ISBN: 978-0-201-18075-6.

Goodchild, Michael F. 2011. "Cartographic Futures on a Digital Earth". In *The Map Reader*, ed. by Martin Dodge, Rob Kitchin, and Chris Perkins, 141–146. John Wiley & Sons. ISBN: 978-0-470-97958-7.

— . 1992. "Geographical Information Science". *International Journal of Geographical Information Systems* 6 (1): 31–45. ISSN: 0269-3798. doi:10.1080/02693799208901893.

— . 2000. "Part 1 Spatial Analysts and GIS Practitioners". *Journal of Geographical Systems* 2 (1): 5–10. ISSN: 1435-5930. http://link.springer.com/article/10.1007/s101090050022.

Goodchild, Michael F., et al. 2012. "Next-Generation Digital Earth". *Proceedings of the National Academy of Sciences* 109 (28): 11088–11094. doi:10.1073/pnas.1202383109.

Google. 2015. *Indoor Map of Westfield Culver City Shopping Mall*.

Harley, J. B., and David Woodward, eds. 1987. *Cartography in Prehistoric, Ancient, and Medieval Europe and the Mediterranean*. The History of Cartography v. 1. Chicago: University of Chicago Press. ISBN: 978-0-226-31633-8.

Hearn, Donald D., M. Pauline Baker, and Warren Carithers. 2010. *Computer Graphics with Open GL*. 4th ed. Upper Saddle River: Prentice Hall. ISBN: 978-0-13-605358-3.

Héno, Raphaële, and Laure Chandelier. 2014. *3D Modeling of Buildings: Outstanding Sites*. London: ISTE Ltd. ISBN: 978-1-84821-536-8.

Hewitt, Mark. 1985. "Representational Forms and Modes of Conception: An Approach to the History of Architectural Drawing". *Journal of Architectural Education (1984-)* 39, no. 2 (): 2–9. ISSN: 1046-4883. doi:10.2307/1424961. JSTOR: 1424961.

Higgins, Sean. 2017. "A $350 DIY Lidar Kit That Could Fuel Development of 3D Tech". Visited on 10/12/2017. https://www.spar3d.com/news/lidar/350-diy-lidar-kit-fuel-development-3d-tech/.

— . 2015. "A LiDAR Unit for Under $100". Visited on 10/12/2017. https://www.spar3d.com/blogs/the-other-dimension/vol13no27-lidar-unit-your-kids-can-afford/.

— . 2016. "Garmin Resurrects the Dirt-Cheap LiDAR: LiDAR-Lite V3 Will Cost $150". Visited on 10/12/2017. https://www.spar3d.com/news/garmin-resurrects-dirt-cheap-lidar-lidar-lite-3-will-cost-150/.

Hondius II, Henricus. 1631. *Nova Totius Terrarum Orbis Geographica Ac Hydrographica Tabula*. https://commons.wikimedia.org/wiki/File:Nova_totius_Terrarum_Orbis_geographica_ac_hydrographica_tabula_(Hendrik_Hondius)_balanced.jpg.

Hong, Sungchul, et al. 2015. "Semi-Automated Approach to Indoor Mapping for 3D as-Built Building Information Modeling". *Computers, Environment and Urban Systems* 51:34–46. ISSN: 0198-9715. doi:10.1016/j.compenvurbsys.2015.01.005.

Hooijberg, Maarten. 1997. *Practical Geodesy: Using Computers*. Berlin: Springer. ISBN: 978-0720123418.

Horn, B. K. P. 1984. "Extended Gaussian Images". *Proceedings of the IEEE* 72 (12): 1671–1686. ISSN: 0018-9219. doi:10.1109/PROC.1984.13073.

Howse, Derek. 1980. *Greenwich Time and the Discovery of the Longitude*. Oxford: Oxford University Press. ISBN: 978-0-19-215948-9.

Huber, Daniel, et al. 2010. "Using Laser Scanners for Modeling and Analysis in Architecture, Engineering, and Construction". In *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, 1–6. doi:10.1109/CISS.2010.5464818.

Ibrahim, Magdy, and Robert Krawczyk. 2003. "The Level of Knowledge of CAD Objects within the Building Information Model". Visited on 01/16/2017. http://papers.cumincad.org/cgi-bin/works/Show?acadia03_023.

Ikeuchi, Katsushi. 1981. "Recognition of 3-d Objects Using the Extended Gaussian Image". In *IJCAI Conference*, 595–600.

Iliffe, Jonathan, and Roger Lott. 2008. *Datums and Map Projections for Remote Sensing, GIS and Surveying*. 2nd ed. Dunbeath: Whittles Publishing. ISBN: 978-1-904445-47-0.

International Association of Oil & Gas Producers. 2017. *EPSG Dataset, Version 9.1*. Visited on 08/26/2017. http://www.epsg.org/EPSGDataset/DownloadDataset.aspx.

International Organization for Standardization. 2002. *Geographic Information — Spatial Referencing by Coordinates (ISO 19111)* ISO/FDIS 19111:2002(E).

— . 2003. *Geographic Information — Spatial Referencing by Geographic Identifiers (ISO 19112)* ISO/FDIS 19112:2003(E).

Isikdag, Umit, and Sisi Zlatanova. 2010. "Interactive Modelling of Buildings in Google Earth: A 3D Tool for Urban Planning". In *Developments in 3D Geo-Information Sciences*, ed. by Tijs Neutens and Philippe Maeyer, 52–70. Berlin: Springer. ISBN: 978-3-642-04790-9. doi:10.1007/978-3-642-04791-6_4.

Jetstar Airways. 2013. *Boeing's Everett Factory*. https://commons.wikimedia.org/wiki/File:At_Boeing%27s_Everett_factory_near_Seattle_(9130160595).jpg.

Johnson, A. E., et al. 1997. "A System for Semi-Automatic Modeling of Complex Environments". In *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, 213–220. doi:10.1109/IM.1997.603868.

Johnson, Brian R. 2015. "One BIM to Rule Them All: Future Reality or Myth?" In *Building Information Modeling*, 173–185. John Wiley & Sons. ISBN: 978-1-119-17475-2.

Ju, Tao. 2012. "CSE 554, Lecture 7: Alignment". Washington University. Visited on 08/26/2017. http://www.cse.wustl.edu/~taoju/cse554/lectures/lect07_Alignment.pdf.

Jung, Jaehoon, Cyrill Stachniss, and Changjae Kim. 2017. "Automatic Room Segmentation of 3D Laser Data Using Morphological Processing". *ISPRS International Journal of Geo-Information* 6 (7): 206. doi:10.3390/ijgi6070206.

Jung, Jaehoon, et al. 2016. "Automated 3D Wireframe Modeling of Indoor Structures from Point Clouds Using Constrained Least-Squares Adjustment for As-Built BIM". *Journal of Computing in Civil Engineering* 30 (4): 04015074. ISSN: 0887-3801. doi:10.1061/(ASCE)CP.1943-5487.0000556.

Jung, Jaehoon, et al. 2014. "Productive Modeling for Development of As-Built BIM of Existing Indoor Structures". *Automation in Construction* 42:68–77. ISSN: 0926-5805. doi:10.1016/j.autcon.2014.02.021.

Junkins, D., and G. Garrard. 1998. "Demystifying Reference Systems : A Chronicle of Spatial Reference Systems in Canada". In *Geomatica*, 52:468–473. Canadian Institute of Geomatics.

Karimi, Hassan A., and Burcu Akinci. 2010. *CAD and GIS Integration*. Boca Raton: CRC Press. ISBN: 978-1-4200-6806-1.

Karlshøj, Jan. 2011. "Information Delivery Manuals". Visited on 01/27/2017. http://iug.buildingsmart.org/idms/.

Kazhdan, Michael, Matthew Bolitho, and Hugues Hoppe. 2006. "Poisson Surface Reconstruction". In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 61–70. Aire-la-Ville, Switzerland: Eurographics Association.

Kerfoot, Helen. 2009. "Geographical Names - Maps, Identity and the United Nations". Santiago, Chile. Visited on 07/22/2007. https://unstats.un.org/unsd/geoinfo/UNGEGN/otherMaterial.html.

Kersten, Thomas P., and Maren Lindstaedt. 2012. "Image-Based Low-Cost Systems for Automatic 3D Recording and Modelling of Archaeological Finds and Objects". In *Progress in Cultural Heritage Preservation*, 1–10. Berlin: Springer. ISBN: 978-3-642-34233-2. doi:10.1007/978-3-642-34234-9_1.

Khoshelham, Kourosh, and Lucía Díaz-Vilariño. 2014. "3D Modelling of Interior Spaces: Learning the Language of Indoor Architecture". In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5:321–326. Copernicus GmbH. doi:10.5194/isprsarchives-XL-5-321-2014.

Khoshelham, Kourosh, and Sander Oude Elberink. 2012. "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications". *Sensors* 12 (2): 1437–1454. doi:10.3390/s120201437.

Khronos Group. 2017. "COLLADA - 3D Asset Exchange Schema". Visited on 01/28/2017. https://www.khronos.org/api/collada/.

Kim, Changmin, Hyojoo Son, and Changwan Kim. 2013. "Automated Construction Progress Measurement Using a 4D Building Information Model and 3D Data". *Automation in Construction* 31:75–82. ISSN: 0926-5805. doi:10.1016/j.autcon.2012.11.041.

Kim, Joon-Seok, Sung-Jae Yoo, and Ki-Joune Li. 2014. "Integrating IndoorGML and CityGML for Indoor Space". In *Web and Wireless Geographical Information Systems*, ed. by Dieter Pfoser and Ki-Joune Li, 184–196. Berlin: Springer. ISBN: 978-3-642-55333-2.

Kim, Yunji, Hyeyoung Kang, and Jiyeong Lee. 2014. "Developing CityGML Indoor ADE to Manage Indoor Facilities". In *Innovations in 3D Geo-Information Sciences*, ed. by Umit Isikdag, 243–265. Cham: Springer International Publishing. ISBN: 978-3-319-00514-0. doi:10.1007/978-3-319-00515-7_15.

King, Bruce. 2012. "A Webcam-Based Photogrammetric System for the Measurement of Facial Landmarks on Newborns". *Photogrammetric Engineering & Remote Sensing* 78 (12): 1285–1294. ISSN: 0099-1112.

Kitchin, Rob, Martin Dodge, and Chris Perkins. 2011. "Introductory Essay: Conceptualising Mapping". In *The Map Reader*, ed. by rtin Dodge, Rob Kitchin, and Chris Perkins, 1–7. John Wiley & Sons. ISBN: 978-0-470-97958-7.

Klawonn, Frank. 2012. *Introduction to Computer Graphics*. London: Springer London. ISBN: 978-1-4471-2732-1.

Klein, Laura, Nan Li, and Burcin Becerik-Gerber. 2012. "Imaged-Based Verification of as-Built Documentation of Operational Buildings". *Automation in Construction* 21:161–171. ISSN: 0926-5805.

Klepeis, Neil E., et al. 2001. "The National Human Activity Pattern Survey (NHAPS): A Resource for Assessing Exposure to Environmental Pollutants". *Journal of Exposure Analysis and Environmental Epidemiology* 11 (3): 231–252. doi:10.1038/sj.jea.7500165.

Kolbe, Thomas H. 2009. "Representing and Exchanging 3D City Models with CityGML". In *3D Geo-Information Sciences*, ed. by Jiyeong Lee and Sisi Zlatanova, 15–31. Berlin: Springer. ISBN: 978-3-540-87394-5. doi:10.1007/978-3-540-87395-2_2.

Kolbe, Thomas H., Gerhard Gröger, and Lutz Plümer. 2005. "CityGML: Interoperable Access to 3D City Models". In *Geo-Information for Disaster Management*, 883–899. Berlin: Springer. ISBN: 978-3-540-24988-7. doi:10.1007/3-540-27468-5_63.

Kotler, Philip, and Kevin Lane Keller. 2006. *Marketing Management*. Twelfth ed. Upper Saddle River: Pearson Prentice Hall. ISBN: 978-0-13-145757-7.

Kuhn, Werner. 2012. "Core Concepts of Spatial Information for Transdisciplinary Research". *International Journal of Geographical Information Science* 26 (12): 2267–2276. ISSN: 1365-8816. doi:10.1080/13658816.2012.722637.

Kumar, Vineet. 2014. "Making "Freemium" Work". *Harvard Business Review*.

Lanciani, Rudolfo. 1893. *Floorplan of the Baths of Diocletian from the Forma Urbis Romae*. Visited on 12/04/2017. https://commons.wikimedia.org/wiki/File:Baths_Diocletian-Lanciani.png.

Leong, Philip, and Simon Carlile. 1998. "Methods for Spherical Data Analysis and Visualization". *Journal of Neuroscience Methods* 80 (2): 191–200. ISSN: 0165-0270. doi:10.1016/S0165-0270(97)00201-X.

Li, Ki-Joune. 2008. "Indoor Space: A New Notion of Space". In *Web and Wireless Geographical Information Systems*, ed. by Michela Bertolotto, Cyril Ray, and Xiang Li, 1–3. Berlin: Springer. ISBN: 978-3-540-89902-0.

— . 2016. "What Is IndoorGML". Visited on 10/24/2017. http://indoorgml.net/.

Library of Congress. 2016. "DWG (AutoCAD Drawing) Format Family". Visited on 03/07/2017. http://www.digitalpreservation.gov/formats/fdd/fdd000445.shtml.

— . 2011a. "ESRI Arc Geodatabase". Visited on 01/28/2017. http://www.digitalpreservation.gov/formats/fdd/fdd000293.shtml.

— . 2011b. "ESRI Shapefile". Visited on 01/28/2017. http://www.digitalpreservation.gov/formats/fdd/fdd000280.shtml.

— . 1136. *Negative Image of the Yu Ji Tu*. https://commons.wikimedia.org/wiki/File:Yuji_tu_-_enhanced_contrast.png.

Liebenberg, Elri, Imre Josef Demhardt, and Soetkin Vervust, eds. 2016. *History of Military Cartography*. Lecture Notes in Geoinformation and Cartography. Cham: Springer International Publishing. ISBN: 978-3-319-25242-1. doi:10.1007/978-3-319-25244-5.

Luhmann, Thomas, et al. 2011. *Close Range Photogrammetry: Principles, Techniques and Applications*. Dunbeath: Whittles Publishing. ISBN: 978-1-84995-057-2.

— . 2014. *Close-Range Photogrammetry and 3D Imaging*. 2nd ed. Berlin: De Gruyter. ISBN: 978-3-11-030269-1.

Mabel, Joe. 2008. *Alaskan Way Viaduct and Dowtown Seattle, Washington*. https://commons.wikimedia.org/wiki/File:Seattle_-_Alaskan_Way_Viaduct_01.jpg.

Maling, D. H. 1992. *Coordinate Systems and Map Projections*. 2nd ed. Oxford: Pergamon Press. ISBN: 978-0-08-037234-1.

Mansley, Chris, et al. 2010. "Geometric Aspects of Robotic Mapping and Monitoring of Data Centers".

Marschner, Steve. 2010. "CS3220 Lecture Notes: Singular Value Decomposition and Applications". Lecture Notes, Cornell University. Visited on 08/26/2017. http://www.cs.cornell.edu/courses/cs3220/2010sp/notes/svd.pdf.

Marton, Z. C., R. B. Rusu, and M. Beetz. 2009. "On Fast Surface Reconstruction Methods for Large and Noisy Point Clouds". In *2009 IEEE International Conference on Robotics and Automation*, 3218–3223. doi:10.1109/ROBOT.2009.5152628.

McClendon, Brian. 2012. "The Never-Ending Quest for the Perfect Map". Visited on 09/23/2017. https://googleblog.blogspot.com/2012/06/never-ending-quest-for-perfect-map.html.

McGlone, J. Chris, ed. 2013. *Manual of Photogrammetry*. Bethesda: American Society for Photogrammetry and Remote Sensing. ISBN: 978-1-57083-099-0.

McIntosh, Jane. 2008. *The Ancient Indus Valley: New Perspectives*. ABC-CLIO. ISBN: 978-1-57607-907-2.

Metcalfe, Howard H. N.d. *Thomas Bros. Map of Downtown Los Angeles in 1946*. http://lanopalera.org/LAHistory/LAMap1946.gif.

Mikhail, Edward M. 2001. *Introduction to Modern Photogrammetry*. In collab. with James S. Bethel and J. Chris McGlone. New Delhi: Wiley. ISBN: 978-81-265-3998-7.

Minecraft Wiki. 2016. "Minecraft Development Resources". Visited on 01/28/2017. http://minecraft.gamepedia.com/Development_resources.

Mobojiang. 2009. *Qin State Map Fragment from Tomb 5 of the Fangmatan Site*. https://commons.wikimedia.org/wiki/File:Fangmatan_paper_map.jpg.

Moore, Lawrence, and David Smith. 2002. "Parametric Surfaces, Part 1". Visited on 10/17/2017. https://services.math.duke.edu/education/ccp/materials/mvcalc/parasurfs/para1.html.

Morrison, Joel L. 1978. "Towards a Functional Definition of the Science of Cartography with Emphasis on Map Reading". *The American Cartographer* 5 (2): 97–110. ISSN: 0094-1689. doi:10.1559/152304078784022845.

Mura, Claudio, et al. 2014. "Automatic Room Detection and Reconstruction in Cluttered Indoor Environments with Complex Room Layouts". *Computers & Graphics* 44:20–32. ISSN: 0097-8493. doi:10.1016/j.cag.2014.07.005.

Musée du Louvre. N.d. *Plans of a Six-Room Building from Late 3rd Millenium BC in Ancient Girsu*. https://commons.wikimedia.org/wiki/File:Building_plans_Louvre_AO338.jpg.

Nagel, Claus, et al. 2010. *Requirements and Space-Event Modeling for Indoor Navigation*. Tech. Rep. OGC 10-191r1, Open Geospatial Consortium Discussion Paper.

National Geospatial-Intelligence Agency. 2014a. *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems*.

— . 2014b. *Universal Grids and Grid Reference Systems* NGA.STND.0037_2.0.0_GRIDS.

National Intelligence Council. 2012. *Global Trends 2030: Alternative Worlds : A Publication of the National Intelligence Council*. December 2012: Office of the Director of National Intelligence. ISBN: 978-1-929667-21-5.

National Research Council. 2006. *Learning to Think Spatially*. Washington, D.C: National Academies Press. ISBN: 978-0-309-09208-1.

Precise Geodetic Infrastructure: National Requirements for a Shared Resource. 2010. Washington, D.C.: National Academies Press. ISBN: 978-0-309-15811-4.

NBS. 2017. "What Is BIM?" Visited on 01/27/2017. https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim.

Nguyen, Anh, and Bac Le. 2013. "3D Point Cloud Segmentation: A Survey". In *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 225–230. doi:10.1109/RAM.2013.6758588.

Nyerges, Timothy L. 1991. "Analytical Map Use". *Cartography and Geographic Information Systems* 18 (1): 11–22. ISSN: 1050-9844. doi:10.1559/152304091783805635.

Ochmann, Sebastian, et al. 2014. "Automatic Generation of Structural Building Descriptions from 3D Point Cloud Scans". In *2014 International Conference on Computer Graphics Theory and Applications (GRAPP)*, 1–8.

Ochmann, Sebastian, et al. 2016. "Automatic Reconstruction of Parametric Building Models from Indoor Point Clouds". *Computers & Graphics* 54:94–103. ISSN: 0097-8493. doi:10.1016/j.cag.2015.07.008.

Oesau, Sven. 2015. "Geometric Modeling of Indoor Scenes from Acquired Point Data". PhD thesis, Université Nice Sophia Antipolis. Visited on 09/10/2017. https://tel.archives-ouvertes.fr/tel-01176721/document.

Okorn, Brian, et al. 2010. "Toward Automated Modeling of Floor Plans". In *Proceedings of the Fifth International Symposium on 3D Data Processing, Visualization and Transmission*. Paris, France.

El-Omari, Samir, and Osama Moselhi. 2008. "Integrating 3D Laser Scanning and Photogrammetry for Progress Measurement of Construction Work". *Automation in Construction* 18 (1): 1–9. ISSN: 0926-5805. doi:10.1016/j.autcon.2008.05.006.

Open Geospatial Consortium. 2012. *OGC City Geography Markup Language (CityGML) Encoding Standard* OGC 12-019, version 2.0.0.

— . 2010. *The OGC Abstract Specification, Topic 2: Spatial Referencing by Coordinates* OGC 08-015r2.

PDF3D. 2015. "About PRC Format". Visited on 01/28/2017. https://www.pdf3d.com/wp-content/cache/supercache/www.pdf3d.com/about-prc/index.html.gz.

Peckham, Matt. 2016. "Minecraft Is Now the Second Best-Selling Game of All Time". *Time*.

Petrie, Richard. 2016. "buildingSMART Data Dictionary Strategic Overview". Visited on 01/27/2017. http://buildingsmart.org/wp-content/uploads/2016/03/16-03-21-bSDD-Strategic-Overview-Webinar.pps.

Pham, H., A. Ruas, and T. Libourel. 2015. "Representing Urban Phenomena in Their Context and at Different LoD: From Raw Data to Appropriate LoD". In *Proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation*, 31–36. Aire-la-Ville, Switzerland: Eurographics Association. ISBN: 978-3-905674-80-4. doi:10.2312/udmv.20151346.

Plag, Hans-Peter. 2011. "Lecture 2: Geodetic Reference Systems and Frames". Visited on 11/18/2015. http://www.hpplag.com/environmental_geodesy/lecture02/.

PulsedLight. 2014. "Update #9". Visited on 10/12/2017. https://www.dragoninnovation.com/projects/32-lidar-lite-by-pulsedlight.

In-Q-Tel. 2003. "In-Q-Tel Announces Strategic Investment In Keyhole". Mountain View, CA. Visited on 01/28/2017. https://www.iqt.org/in-q-tel-announces-strategic-investment-in-keyhole/.

Reinhardt, Jan, and Jim Bedrick, eds. 2016. *2016 Level of Development Specification*. Visited on 01/27/2017. http://bimforum.org/wp-content/uploads/2016/10/LOD_Spec_2016_Part_I_2016-10-19.zip.

Remondino, Fabio. 2003. "From Point Cloud to Surface: The Modeling and Visualization Problem". In *Proceedings of the ISPRS Working Group V/6*, vol. 23. Tarasp-Vulpera, Engadin, Switzerland.

Renslow, Michael S., ed. 2012. *Manual of Airborne Topographic Lidar*. Bethesda: American Society for Photogrammetry Remote Sensing. ISBN: 978-1-57083-097-6.

Rich, Stu, and Jennifer Smith. 2014. "Introducing the Facilities Information Spatial Data Model at the 2014 Esri User Conference". Visited on 01/28/2017. http://www.prweb.com/releases/2014/07/prweb12009858.htm.

Roberts, Tristan. 2016. "We Spend 90% of Our Time Indoors. Says Who?" Visited on 12/14/2017. https://www.buildinggreen.com/blog/we-spend-90-our-time-indoors-says-who.

Robinson, Arthur H. 1952. *The Look of Maps: An Examination of Cartographic Design*. University of Wisconsin Press. ISBN: 978-0-299-00950-2.

Robinson, Arthur H., and Barbara Bartz Petchenik. 1975. "The Map as a Communication System". *The Cartographic Journal* 12 (1): 7–15. ISSN: 0008-7041. doi:10.1179/caj.1975.12.1.7.

Rodell, Matthew, et al. 2006. "Estimating Groundwater Storage Changes in the Mississippi River Basin (USA) Using GRACE". *Hydrogeology Journal* 15 (1): 159–166. Visited on 04/19/2016. doi:10.1007/s10040-006-0103-7. http://link.springer.com/article/10.1007/s10040-006-0103-7.

Royster, David. 2002. "Mathematics and Maps". Charlotte, North Carolina. http://www.msc.uky.edu/droyster/talks/NCCTM_2002/Mapping.pdf.

Rüppel, Uwe, Udo Meissner, and Bernd Möller. 1993. "Object-Oriented Data Exchange for the Integration of Design Processes in Structural Engineering", 94–101. ASCE.

Rusinkiewicz, Szymon. 2004. "Estimating Curvatures and Their Derivatives on Triangle Meshes". In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004*, 486–493. doi:10.1109/TDPVT.2004.1335277.

— . 2011. "Rigid ICP". Llandudno, North Wales, UK. Visited on 08/26/2017. http://resources.mpi-inf.mpg.de/deformableShapeMatching/EG2011_Tutorial/slides/2.1%20Rigid%20ICP.pdf.

Rusinkiewicz, Szymon, and Marc Levoy. 2001. "Efficient Variants of the ICP Algorithm". In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145–152. doi:10.1109/IM.2001.924423.

Saint-Pol, Bibi. 2006. *Reproduction of Anaximander's World Map.* https://commons.wikimedia.org/wiki/File:Anaximander_world_map-en.svg.

Sanchez, Victor, and Avideh Zakhor. 2012. "Planar 3D Modeling of Building Interiors from Point Cloud Data". In *2012 19th IEEE International Conference on Image Processing*, 1777–1780. doi:10.1109/ICIP.2012.6467225.

Schnabel, R., R. Wahl, and R. Klein. 2007. "Efficient RANSAC for Point-Cloud Shape Detection". *Computer Graphics Forum* 26 (2): 214–226. ISSN: 1467-8659. doi:10.1111/j.1467-8659.2007.01016.x.

Schnabel, Ruwen, et al. 2007. *Shape Recognition in 3D Point-Clouds*, Computer Graphics Technical Reports CG-2007/1. Universität Bonn.

Shan, Jie, and Charles K. Toth. 2009. *Topographic Laser Ranging and Scanning: Principles and Processing*. Boca Raton: CRC Press. ISBN: 978-1-4200-5143-8.

Sinton, David. 1978. "The Inherent Structure of Information as a Constraint to Analysis: Mapped Thematic Data as a Case Study". *Harvard papers on geographic information systems*.

Slocum, Terry A., et al. 2010. *Thematic Cartography and Geovisualization*. Ed. by Keith C. Clarke. Upper Saddle River: Pearson Prentice Hall. ISBN: 978-0-13-801006-5.

Smith, M. W., J. L. Carrivick, and D. J. Quincey. 2016. "Structure from Motion Photogrammetry in Physical Geography". *Progress in Physical Geography* 40 (2): 247–275. ISSN: 0309-1333. doi:10.1177/0309133315615805.

Smullin, Louis D., and Giorgio Fiocco. 1962. "Optical Echoes from the Moon". *Nature* 194 (4835): 1267–1267. ISSN: 0028-0836, visited on 10/12/2017. doi:10.1038/1941267a0. http://www.nature.com/nature/journal/v194/n4835/abs/1941267a0.html?foxtrotcallback=true.

Snyder, John P. 1987. *Map Projections: A Working Manual*, Professional Paper 1395. U.S. Geological Survey.

Sobel, Dava. 2007. *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*. Bloomsbury USA. ISBN: 978-0007790166.

Solberg, Anne. 2016. "Region and Edge Based Segmentation". University in Oslo. http://www.uio.no/studier/emner/matnat/ifi/INF4300/h16/undervisningsmateriale/inf4300-2016-f04-segmentation.pdf.

Stambler, A., and D. Huber. 2014. "Building Modeling through Enclosure Reasoning". In *2014 2nd International Conference on 3D Vision*, 2:118–125. doi:10.1109/3DV.2014.65.

Stones, Alison. 2006a. *Floorplan of the Abbey of Saint-Denis c.1700*. http://www.medart.pitt.edu/image/France/St-denis/plans/plan-felebien-s.jpg.

— . 2006b. *Map of the Town of St. Denis in 1706*. http://www.medart.pitt.edu/image/France/St-denis/plans/citymaps.jpg.

Strassberg, Gil, Bridget R. Scanlon, and Matthew Rodell. 2007. "Comparison of Seasonal Terrestrial Water Storage Variations from GRACE with Groundwater-Level Measurements from the High Plains Aquifer (USA)". *Geophysical Research Letters* 34 (14). ISSN: 1944-8007. doi:10.1029/2007GL030139.

Stroustrup, Bjarne. 2014. *Programming: Principles and Practice Using C++, Second Edition*. 2nd ed. Addison-Wesley Professional. ISBN: 978-0-321-99278-9.

Succar, Bilal. 2009. "Building Information Modelling Framework: A Research and Delivery Foundation for Industry Stakeholders". *Automation in Construction* 18 (3): 357–375. ISSN: 0926-5805. doi:10.1016/j.autcon.2008.10.003.

Suermann, Patrick C. 2009. "Evaluating The Impact Of Building Information Modeling (BIM) On Construction". PhD thesis, University of Florida.

Synge, E. H. 1930. "A Method of Investigating the Higher Atmosphere". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 9 (60): 1014–1020. ISSN: 1941-5982.

Taguchi, Genichi, and Don Clausing. 1990. "Robust Quality". *Harvard Business Review*.

Talbert, Richard J. A. 2014. *Ancient Perspectives: Maps and Their Place in Mesopotamia, Egypt, Greece, and Rome*. Chicago: University of Chicago Press. ISBN: 978-0-226-78940-8.

Tang, Pingbo, et al. 2010. "Automatic Reconstruction of As-Built Building Information Models from Laser-Scanned Point Clouds: A Review of Related Techniques". *Automation in Construction* 19 (7): 829–843. ISSN: 0926-5805. doi:10.1016/j.autcon.2010.06.007.

Taub, Liba. 1993. "The Historical Function of the "Forma Urbis Romae"". *Imago Mundi* 45:9–19. ISSN: 0308-5694. JSTOR: 1151156.

Taylor, Trevor, Wageeh W. Boles, and Shlomo Geva. 2007. "Map Building Using Cheap Digital Cameras", 130–137. IEEE.

Ten Questions for Esri: 3D Expert, Nathan Shepard. 2015. Visited on 01/28/2017. https://geonet.esri.com/thread/168231-ten-questions-for-esri-3d-expert-nathan-shepard.

Thompson, E. H. 1973. "Review of Coordinate Systems and Map Projections, by D. H. Maling". *The Photogrammetric Record* 7 (42): 755–758. ISSN: 1477-9730. doi:10.1111/j.1477-9730.1973.tb01216.x.

Thomson, Charles, and Jan Boehm. 2015. "Automatic Geometry Generation from Point Clouds for BIM". *Remote Sensing* 7 (9): 11753–11775. doi:10.3390/rs70911753.

Thrower, Norman J. W. 2008. *Maps and Civilization: Cartography in Culture and Society, Third Edition*. Chicago: University of Chicago Press. ISBN: 978-0-226-79975-9.

Tobler, Waldo R. 1976. "Analytical Cartography". *The American Cartographer* 3 (1): 21–31. ISSN: 0094-1689. doi:10.1559/152304076784080230.

— . 1959. "Automation and Cartography". *Geographical Review* 49 (4): 526–534. ISSN: 0016-7428. doi:10.2307/212211. JSTOR: 212211.

Tolmer, Charles-Edouard, et al. 2013. "CityGML and IFC: Going Further than LOD". doi:10.1109/DigitalHeritage.2013.6743808.

Torge, Wolfgang, and Jürgen Müller. 2012. *Geodesy*. 4th ed. Berlin: De Gruyter. ISBN: 978-3-11-020718-7.

Turner, Eric L. 2015. "3D Modeling of Interior Building Environments and Objects from Noisy Sensor Suites". Ph.D., University of California, Berkeley. https://search.proquest.com/docview/1725205460/abstract/B195B96ECFE64EF2PQ/1.

Turner, Eric L., Peter Cheng, and Avideh Zakhor. 2015. "Fast, Automated, Scalable Generation of Textured 3D Models of Indoor Environments". *IEEE Journal of Selected Topics in Signal Processing* 9 (3): 409–421. ISSN: 1932-4553. doi:10.1109/JSTSP.2014.2381153.

Turner, Eric L., and Avideh Zakhor. 2012. "Watertight As-Built Architectural Floor Plans Generated from Laser Range Data". In *Visualization Transmission 2012 Second International Conference on 3D Imaging, Modeling, Processing*, 316–323. doi:10.1109/3DIMPVT.2012.80.

Tuve, M. A., E. A. Johnson, and O. R. Wulf. 1935. "A New Experimental Method for Study of the Upper Atmosphere". *Terrestrial Magnetism and Atmospheric Electricity* 40 (4): 452–454. ISSN: 0096-8013. doi:10.1029/TE040i004p00452.

Umeyama, Shinji. 1991. "Least-Squares Estimation of Transformation Parameters between Two Point Patterns". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13 (4): 376–380. ISSN: 0162-8828. doi:10.1109/34.88573.

United Nations. 2017. "Household Size and Composition around the World". *Population Facts*, no. 2. Visited on 12/14/2017. http://www.un.org/en/development/desa/population/publications/pdf/popfacts/PopFacts_2017-2.pdf.

— . 2014. *World Urbanization Prospects 2014: Highlights.* United Nations Publications. ISBN: 978-92-1-151517-6.

United Nations Group of Experts on Geographical Names. 2017. "Overview of the United Nations Group of Experts on Geographical Names (UNGEGN)". Visited on 07/22/2017. https://unstats.un.org/unsd/geoinfo/UNGEGN/default.html.

United Nations Population Fund. 2014. *Population Dynamics in the Post-2015 Development Agenda: Report of the Global Thematic Consultation on Population Dynamics.* UNFPA. ISBN: 978-0-89714-020-1.

U.S. Geological Survey. 2001. *The Universal Transverse Mercator (UTM) Grid.* Fact Sheet 077-01.

Utrilla, P., et al. 2009. "A Palaeolithic Map from 13,660 calBP: Engraved Stone Blocks from the Late Magdalenian in Abauntz Cave (Navarra, Spain)". *Journal of Human Evolution* 57 (2): 99–111. ISSN: 0047-2484. doi:10.1016/j.jhevol.2009.05.005.

Valero, Enrique, Antonio Adán, and Frédéric Bosché. 2016. "Semantic 3D Reconstruction of Furnished Interiors Using Laser Scanning and RFID Technology". *Journal of Computing in Civil Engineering* 30 (4): 04015053. ISSN: 0887-3801. doi:10.1061/(ASCE)CP.1943-5487.0000525.

Valero, Enrique, Antonio Adán, and Carlos Cerrada. 2012. "Automatic Construction of 3D Basic-Semantic Models of Inhabited Interiors Using Laser Scanners and RFID Sensors". *Sensors* 12 (5): 5705–5724. doi:10.3390/s120505705.

Van Sickle, Jan. 2010. *Basic GIS Coordinates.* 2nd ed. Boca Raton: CRC Press. ISBN: 978-1-4200-9231-8.

Verhoeven, Geert. 2011. "Taking Computer Vision Aloft – Archaeological Three-Dimensional Reconstructions from Aerial Photographs with Photoscan". *Archaeological Prospection* 18 (1): 67–73. ISSN: 1099-0763. doi:10.1002/arp.399.

Walton, Lisa, and Michael Worboys. 2010. "Indoor Spatial Theory". Technical report presented at the ISA project meeting held at the 2010 International Workshop on Indoor Spatial Awareness, Taipei, Taiwan.

Watson, G. A. 2006. "Computing Helmert Transformations". *Journal of Computational and Applied Mathematics* 197 (2): 387–394. ISSN: 0377-0427. doi:10.1016/j.cam.2005.06.047.

Web3D Consortium. 2008. *Extensible 3D (X3D), ISO/IEC 19775-1:2008*. Visited on 01/28/2017. http://www.web3d.org/documents/specifications/19775-1/V3.2/Part01/components/group.html.

— . 2017. "What Is X3D Graphics". Visited on 01/28/2017. http://www.web3d.org/what-x3d-graphics.

Westoby, M. J., et al. 2012. "'Structure-from-Motion' Photogrammetry: A Low-Cost, Effective Tool for Geoscience Applications". *Geomorphology* 179 (Supplement C): 300–314. ISSN: 0169-555X. doi:10.1016/j.geomorph.2012.08.021.

Wilcox, Rand R. 2010. *Fundamentals of Modern Statistical Methods*. New York: Springer. ISBN: 978-1-4419-5524-1. doi:10.1007/978-1-4419-5525-8_3.

Wingfield, Nick. 2014. "Microsoft Sees More Than Fun in a Game Like Minecraft". *The New York Times*. ISSN: 0362-4331.

Wong, S. 2012. "A Fast Webcam Photogrammetric System to Support Optical Imaging of Brain Activity". Doctoral, UCL (University College London). Visited on 03/04/2014. http://discovery.ucl.ac.uk/1344106/.

Xiong, Xuehan, and Daniel Huber. 2010. "Using Context to Create Semantic 3D Models of Indoor Environments". British Machine Vision Association. ISBN: 978-1-901725-40-7. doi:10.5244/C.24.45.

Xiong, Xuehan, et al. 2013. "Automatic Creation of Semantically Rich 3D Building Models from Laser Scanner Data". *Automation in Construction* 31:325–337. ISSN: 0926-5805. doi:10.1016/j.autcon.2012.10.006.

Yastikli, Naci. 2007. "Documentation of Cultural Heritage Using Digital Photogrammetry and Laser Scanning". *Journal of Cultural Heritage* 8 (4): 423–427. ISSN: 1296-2074.

Yee, Cordell D. K. 1994a. "Chapter 5: Taking the World's Measure: Chinese Maps between Observation and Text". In *Cartography in the Traditional East and Southeast Asian Societies*, ed. by J. B. Harley and David Woodward, vol. 2. Chicago: University of Chicago Press. ISBN: 978-0-226-31637-6.

— . 1994b. “Chapter 7: Traditional Chinese Cartography and the Myth of Westernization”. In *Cartography in the Traditional East and Southeast Asian Societies*, ed. by J. B. Harley and David Woodward, vol. 2. Chicago: University of Chicago Press. ISBN: 978-0-226-31637-6.

Yeh, Pat J.-F., et al. 2006. “Remote Sensing of Groundwater Storage Changes in Illinois Using the Gravity Recovery and Climate Experiment (GRACE)”. *Water Resources Research* 42 (12). ISSN: 1944-7973. doi:10.1029/2006WR005374.

Yoon, Sanghyun, Jaehoon Jung, and Joon Heo. 2015. “Practical Implementation of Semi-Automated As-Built Bim Creation for Complex Indoor Environments”. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4:143–146. doi:10.5194/isprsarchives-XL-4-W5-143-2015.

Zheng, Jesse. 2005. *Optical Frequency-Modulated Continuous-Wave (FMCW) Interferometry*. Vol. 107. Springer Series in Optical Sciences. New York: Springer US. ISBN: 978-0-387-23009-2. doi:10.1007/b100384.

Zimmerman, Edward G. 1995. “Angle Measurement: Transits and Theodolites”. In *The Surveying Handbook*, 80–112. New York: Springer US. ISBN: 978-1-4613-5858-9.

Zlatanova, Sisi, Jan Stoter, and Umits Isikdag. 2012. “Standards for Exchange and Storage of 3D Information: Challenges and Opportunities for Emergency Response”. *Proceedings of the 4th International Conference on Cartography & GIS* 2.

Zlatanova, Sisi, et al. 2013. “Problems in Indoor Mapping and Modelling”. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-4/W4, 2013*. Cape Town: ISPRS.