

Automated Assessment of Potential Cell Tower Signal  
Interference with High Accuracy Surveys In Los Angeles County

by

Philip Henry Griffin

A Thesis Presented to the  
FACULTY OF THE USC DORNSIFE COLLEGE OF LETTERS, ARTS AND SCIENCES  
University of Southern California  
In Partial Fulfillment of the  
Requirements for the Degree  
MASTER OF SCIENCE  
(GEOGRAPHIC INFORMATION SCIENCE AND TECHNOLOGY)

December 2021

To my mother, Ranwa Haddad, who described how useful a tool like SurvInt could be.

## **Acknowledgements**

I'd like to thank my advisor, Dr. Jennifer Swift, whose assistance and guidance was critical to completing this thesis. I would also like to thank Dr. Vanessa Osborne for helping me edit this manuscript and Dr. Andrew Marx for inspiring me to develop this project into a master's thesis. Dr. Robert Vos provided insightful advice at the outset of the project, which helped shape the scope and organization of my thesis, and I appreciate his support. I am grateful to my family and fiancé for their patience and support throughout this process. Finally, I would like to give a special thank you to my father, Ed Griffin. His endless reservoir of enthusiasm helped me persevere and overcome the many challenges I faced in the development SurvInt. Without his encouragement this project would not have been possible.

# Table of Contents

|  |     |
|--|-----|
| Dedication.....  | ii  |
| Acknowledgements.....  | iii |
| List of Tables .....   | vi  |
| List of Figures.....   | vii |
| Abbreviations.....   | ix  |
| Abstract.....  | x   |
| Chapter 1 Introduction .....                                     | 1   |
| 1.1. Background .....  | 1   |
| 1.2. Goals and Motivation.....                                   | 2   |
| 1.3. Overview of GIS Automation and the Development Process..... | 4   |
| 1.4. Structure and Organization of This Thesis.....              | 6   |
| Chapter 2 Related Work.....                                      | 1   |
| 2.1. Related Physics and Engineering Works.....                  | 1   |
| 2.2. Related Interference Studies.....                           | 3   |
| 2.3. Related GIS Methods .....                                   | 7   |
| Chapter 3 Methods.....   | 11  |
| 3.1. Input Data Description .....                                | 11  |
| 3.2. Tool Development Methods.....                               | 15  |
| 3.3. Use Cases and Methods of Operation .....                    | 17  |
| 3.3.1. Use Case 1 - Overview.....                                | 17  |
| 3.3.1.1. UC1 Submodel: Validate County .....                     | 18  |
| 3.3.1.2. UC1 Submodel: Create City GDB.....                      | 21  |
| 3.3.1.3. UC1 Submodel: Clip City Layers.....                     | 22  |
| 3.3.1.4. UC1 Submodel: Polish City GDB.....                      | 24  |

|            |   |    |
|------------|---|----|
| 3.3.1.5.   | UC1 Submodel: Output City Map.....                                      | 24 |
| 3.3.2.     | Use Case 2 - Overview.....  | 27 |
| 3.3.2.1.   | UC2 Submodel: Extract Survey Parcels.....                               | 30 |
| 3.3.2.2.   | UC2 Submodel: Process Survey Interference.....                          | 30 |
| 3.3.2.2.1  | UC2 Submodel: Process Survey Interference – SurvInt Parcel Rasters..... | 31 |
| 3.3.2.2.2  | UC2 Submodel: Process Survey Interference – SurvInt Critical Towers...  | 34 |
| 3.3.2.3.   | UC2 Submodel: Output SurvInt Map.....                                   | 36 |
| 3.3.2.4.   | UC2 Submodel: Output SurvInt Reports.....                               | 38 |
| Chapter 4  | Results.....  | 40 |
| 4.1.       | The SurvInt Toolset.....  | 40 |
| 4.2.       | Use Case Validation and Visualization Results.....                      | 43 |
| 4.2.1.     | Use Case 1 – Validation Results.....                                    | 43 |
| 4.2.2.     | Use Case 2 – Validation Results.....                                    | 48 |
| 4.3.       | Effects Of Terrain On Survey Interference.....                          | 54 |
| Chapter 5  | Discussion and Conclusions.....   | 63 |
| 5.1.       | Development Challenges.....   | 63 |
| 5.1.1.     | Challenge 1: Controlling ModelBuilder execution.....                    | 64 |
| 5.1.2.     | Challenge 2: Raster Automation.....                                     | 65 |
| 5.1.3.     | Challenge 3: ArcPy Updates.....   | 67 |
| 5.2.       | Future Enhancements.....  | 69 |
| 5.2.1.     | Enhancement 1: Physical Models.....                                     | 69 |
| 5.2.2.     | Enhancement 2: Real-World Data and Validation.....                      | 73 |
| 5.2.3.     | Enhancement 3: GIS Automation.....                                      | 75 |
| 5.3.       | Conclusion.....   | 78 |
| References | .....   | 80 |

## List of Tables

|  |    |
|--|----|
| Table 1 SurvInt Spatial Data and Metadata .....  | 14 |
| Table 2 List of Components of the SurvInt Project .....                                    | 40 |
| Table 3 Cell Tower Density and Statistics of Terrain Elevation In Los Angeles County ..... | 55 |
| Table 4 Critical Towers Per Parcel Increased in the Flat Terrain of Compton, CA .....      | 62 |

## List of Figures

|  |    |
|--|----|
| Figure 1 SurvInt Design from Problem Definition to Logical Model, to Build and Test.....       | 4  |
| Figure 2 Impact of Cellular Base Station on High Precision GPS Receiver (USDOT 2018) .....     | 5  |
| Figure 3 High-level SurvInt Geoprocessing Workflow with Two Interconnected Use Cases .....     | 17 |
| Figure 4 Top Level Model for SurvInt UC1 - Create and Populate a City Geodatabase .....        | 19 |
| Figure 5 UC1 Submodel (1 of 5): Validate County GDB .....                                      | 19 |
| Figure 6 Python Code for Validating the County Geodatabase.....                                | 20 |
| Figure 7 UC1 Submodel (2 of 5): Create and Populate a City Geodatabase .....                   | 22 |
| Figure 8 UC1 Submodel (3 of 5): Clip County Data with City Boundary Features.....              | 23 |
| Figure 9 UC1 Submodel (4 of 5): Python Code Snippet of “Polish City GDB” Code .....            | 25 |
| Figure 10 UC1 Submodel (5 of 5): Output a City Map with SurvInt Feature Layers .....           | 25 |
| Figure 11 Python Code Snippet Creates a City Map with SurvInt Layers Required for UC2 .....    | 27 |
| Figure 12 Top Level Model for SurvInt UC2 – Process Survey Parcels.....                        | 29 |
| Figure 13 UC2 Submodel (1 of 4): Process Survey Interference .....                             | 31 |
| Figure 14 Calculation of the SurvInt Metric Involves Both Distance and Visibility Rasters..... | 32 |
| Figure 15 UC2 Submodel (1 of 4): Python Code Snippet Creates SurvInt Parcel Rasters .....      | 35 |
| Figure 16 UC2 Submodel (2 of 4): Python Code Snippet Identifies Critical Towers .....          | 36 |
| Figure 17 UC2 Submodel (3 of 4): Python Code Snippet Outputs a SurvInt Metric Map.....         | 37 |
| Figure 18 UC2 Submodel (4 of 4): Python Code Snippet Generates SurvInt Reports.....            | 39 |
| Figure 19 SurvInt Project in ArcGIS Pro Includes City Boundaries in Los Angeles County .....   | 41 |
| Figure 20 SurvInt Testing Used Rancho Palos Verdes (yellow) as a Study Area.....               | 42 |
| Figure 21 Processing Dialog for SurvInt Use Case 1 (UC1) – Model Parameter Inputs .....        | 44 |
| Figure 22 Visual Result of SurvInt Use Case 1 – Layered City Map of Rancho Palos Verdes ...    | 45 |

|   |    |
|---|----|
| Figure 23. Example Output of SurvInt UC1 - Create and Populate a City Geodatabase .....       | 45 |
| Figure 24 Output Data Result of UC1 – SurvInt Rancho Palos Verdes GDB.....                    | 46 |
| Figure 25 Multiple Runs of UC1 Create City Maps Accumulated on the County Map.....            | 47 |
| Figure 26 Processing Dialog for SurvInt UC2 – Model Parameter Inputs .....                    | 48 |
| Figure 27 Visual Result of SurvInt UC2 –SurvInt Raster and Critical Towers in RPV.....        | 49 |
| Figure 28 Visual Result of SurvInt UC2 –Critical Towers Outside the City Boundary.....        | 50 |
| Figure 29 Both Base Station and RTK Roving Receiver Are Subject to Interference .....         | 51 |
| Figure 30. Example Output of SurvInt UC2 - Assess Interference and Notify Providers.....      | 52 |
| Figure 31 Out Data Result of SurvInt UC2 – RPV GDB With SurvIntMetric Raster .....            | 53 |
| Figure 32 A Network Notification Letter Is Generated for Each Associated Critical Tower ..... | 54 |
| Figure 33 Cell Tower Density Correlates Inversely With Variation in Terrain Elevation.....    | 56 |
| Figure 34 Two SurvInt Metric Rasters in the Flat Terrain of Compton, CA.....                  | 57 |
| Figure 35 Two SurvInt Metric Rasters in the Hilly Terrain of Rancho Palos Verdes, CA .....    | 59 |
| Figure 36 Two SurvInt Metric Rasters Separated by the Hill in Signal Hill, CA .....           | 61 |
| Figure 37 Elevation Beam Pattern Reduces RF Signal Power Near a Cell Tower.....               | 70 |
| Figure 38 Cellular Traffic Is Nearly Constant During Business Hours (Wang et al. 2015).....   | 72 |
| Figure 39 Atmospheric Attenuation is Negligible for Cell Network Signals (Milione 2011).....  | 73 |
| Figure 40 Graphical Selection Could Replace SQL Queries in SurvInt Input Dialogs.....         | 77 |



## Abbreviations

|         |  |
|---------|--|
| GDB     | Esri File Geodatabase                      |
| GIS     | Geographic information system              |
| GISci   | Geographic information science             |
| GNSS    | Global Navigation Satellite System         |
| GPS     | Global Positioning System                  |
| HPP     | High Precision Positioning                 |
| LA      | Los Angeles                                |
| LTE     | Long Term Evolution                        |
| OOP     | Object-Oriented Programming                |
| RF      | Radio Frequency                            |
| RTK     | Real Time Kinematic                        |
| SSI     | Spatial Sciences Institute                 |
| SurvInt | Survey Interference tool                   |
| USDOT   | United States Department of Transportation |
| USGS    | United States Geological Survey            |
| USC     | University of Southern California          |

## **Abstract**

A vast array of systems, including cellular telephones and GPS receivers, use Radio Frequency (RF) signals for communication. These devices transmit and receive RF signals, and when one of them is transmitting, there is the potential for interfering with other nearby receivers. The objective of this work was to build a tool to analyze potential interference between cell telephone transmissions and high accuracy GPS receivers used in land survey equipment.

Many Geographic Information Systems (GIS) applications, remote sensing systems, and specialized land survey receivers depend on highly accurate location services such as the Global Positioning Satellite System (GPS) and cellular telephone networks. GPS location is ubiquitous in GIS applications and allows land surveyors to accurately measure location, distance, and area. Unfortunately, since cell signal frequencies overlap GPS frequency bands and can interfere with GPS signals upon which land survey equipment relies.

This thesis documents the development of an innovative automated toolset called SurvInt, which quantifies how spatially pervasive, severe, and variable the impact of cellphone signal RF interference is in Los Angeles County. Los Angeles is the study area for SurvInt for three reasons. First, the county includes 88 separate cities to test the SurvInt tools. Second, the county's variable terrain ranges from sea level to 3068m to tests critical visibility calculations in SurvInt. Finally, LA County makes most SurvInt data publicly available.

SurvInt provides two direct benefits to land surveyors. First, SurvInt helps visualize the spatial distribution of RF interference. Second, SurvInt identifies the locations and operators of specific interfering cell towers so that interference mitigation efforts can be planned. Both benefits exemplify the power of GIS to help manage complex interactions between spatially distributed cellular telephone networks.

## **Chapter 1 Introduction**

Radio Frequency (RF) signals are used for communication by a vast array of users, radio and television, Computer equipment, cellular telephones, and GPS receivers. These devices transmit and receive RF signals, and whenever one of them is transmitting, there is the potential for interfering with other nearby receivers. This interference is called RF interference or RFI. The fundamental objective of this work was to build a tool that can provide spatial analyses of regions of potential RFI between cell telephone transmissions and high accuracy GPS receivers used in land survey equipment.

Many GIS applications, remote sensing systems, and specialized receivers used in land surveys are enabled by highly accurate location services. Two such location services are the GPS and cellular telephone networks. GPS location is ubiquitous in modern GIS applications and allows land surveyors to accurately measure location, distance, and area. Unfortunately, since cell signal frequencies overlap GPS frequency bands, they can interfere with the GPS signals upon which land survey equipment relies. Developing a method for spatial analysis and visualization of the potential conflict between these networks was the main goal of this thesis project.

### **1.1. Background**

Geographic Information System applications, such as mapping with remote sensing systems, and the equipment used in precision land surveys, depend on accurate location services that must be constantly accessible and globally available. For example, two very familiar location services used extensively in spatial applications like mapping and navigation are GPS and cellular telephone networks. These services are used by most people daily. Land surveyors also use

location services to assist in making accurate spatial measurements of distance and area. Most importantly, along with the many benefits of using these worldwide location systems, lies the challenge of interoperability. In the context of this thesis, interoperability is the ability to use multiple location services systems in conjunction without experiencing inter-system interference.

In terms of interference with location services, the key interoperability issue addressed by SurvInt is RF signal interference. The interoperability issues modeled by SurvInt result from the natural competition between two principal users of the two location services. Although cell phone users often want the highest signal strength possible, other users requiring High Precision Positioning (HPP), like land surveyors, do not. Young et al. (2017) documented that LTE cell phone signals can degrade survey equipment's speed and accuracy. Norhisyam et al. (2013) found that the degradation results from GPS and cell network location services both being based on signal triangulation using overlapping parts of the RF spectrum. Since cell signal frequencies overlap the GPS frequency bands, they can, and do, interfere with or completely disrupt the lower power GPS signals that land survey equipment was designed to use. The GIS tools developed in this thesis project provide a unique new set of tools to assist in understanding the spatial distribution and impact of this kind of interference on land surveys.

## **1.2. Goals and Motivation**

The original motivation for this thesis project was the USDOT (2018) Adjacent Band Compatibility report that assessed the “extent to which such adjacent band transmitters impact GPS/GNSS devices used for ... numerous other civil applications”. The USDOT found potentially severe impacts from the growing “Long Term Evolution” (LTE) cellular network on many high precision GPS receivers, which confirmed the author's own experience over a decade with the negative effects of lost GPS signals on equipment used for geological field surveys. This

work follows Bolstad (2016) and defines a GPS receiver, or land survey receiver, as an electronic device that records data transmitted by each GPS satellite and then processes these data to obtain three-dimensional coordinates. Duffy (2020) notes that high accuracy land surveys (e.g., tax parcel surveys) are one example of a civil application that is of special importance to GIS analysts. The specific focus of the SurvInt visualization tool developed as part of this thesis is to provide spatial representations of the potential for cell signal interference with tax parcel surveys as an easy-to-use GIS-ready tool. This tool is a unique innovation in land surveying not previously invented at the time this work was accomplished, to the best knowledge of the author.

The goal of creating these representations, or visualizations, is to assist city and county governments in the planning and execution of tax parcel surveys. Specifically, SurvInt was designed to support city planning offices in two ways: First, SurvInt provides a quantitative, spatial assessment of the potential for severe cell tower interference with high accuracy surveys of tax parcels located in any city in Los Angeles County. Second, SurvInt's spatial analysis results for a user-selected parcel in a report suitable for notifying cell tower operators and requesting interference mitigation efforts. These critical Use Cases are illustrated in Chapter 4 using example inputs and outputs from several cities in LA County.

To summarize the objectives of this thesis work, the overarching goals of this project include the following:

- Create an automated, Python-based, ArcGIS Pro tool (called SurvInt) that analyzes spatial characteristics of RF interference caused by cellular telephone towers
- Demonstrate that SurvInt quantifies and visualizes the extent and severity of RF interference on tax parcel surveys

- Apply the tool to cities in Los Angeles County to qualitatively assess the effects of varying terrain elevation on the patterns of cell tower interference

The results in Chapter 4 confirm that these objectives have been met.

### 1.3. Overview of GIS Automation and the Development Process

Stone and Stone (2018) describe a standard software development process that was followed in the development of SurvInt. A critical early step in their development process is “*Identifying use cases, actors, and scenarios.*” This simple idea for decomposing the work into user-centric functional operations made the development of this project more tractable and efficient. In particular, building tools for use cases allowed the development of later use cases to benefit from lessons learned while developing earlier ones. Figure 2 compares the general development steps of a typical Object-Oriented Programming (OOP) project with the details of the development steps followed in the design, build and test of SurvInt.

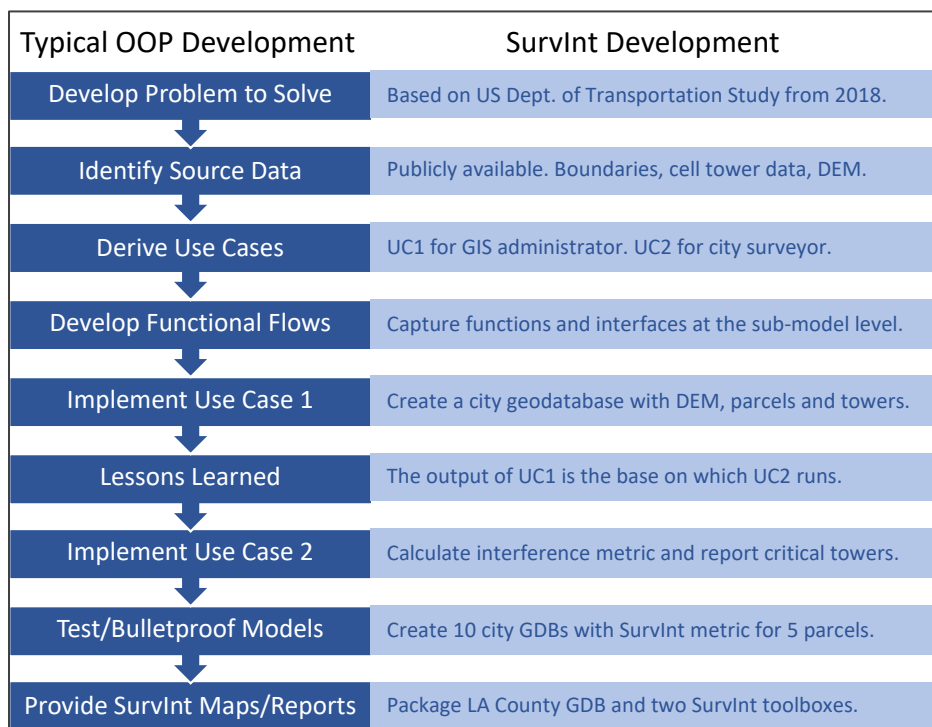


Figure 1 SurvInt Design from Problem Definition to Logical Model, to Build and Test.

At least some form of scripting was utilized early in the development of the SurvInt tools, including the use of spatial analysis tools built into Esri ArcGIS Pro. Project-wide automation using a true programming language was added well after most of the basic GIS capabilities were implemented. Allen (2011) describes an early innovation in ArcGIS Pro automation, which was the use of Esri ModelBuilder. ModelBuilder's visual programming environment makes it easy to use a wide range of standard ArcGIS Pro tools in a repeatable sequence of operations. The fact that it is a visual programming tool that uses diagrams instead of code makes it easily accessible to ArcGIS Pro users. It allows for rapid prototyping of automated functions. ModelBuilder significantly reduces the time needed to execute a series of GIS operations and reduces the rate of human errors that might be introduced into an analysis without this kind of automation. However, the benefits of a simple-to-use automation system comes at a price – limited processing speed. Data processing speed led to the next evolution of GIS automation in this project, which was to incorporate Python programming and a dedicated library that Esri (2013) calls ArcPy.

SurvInt combines ModelBuilder tools with ArcPy-based Python scripts to comprise a pair of ArcGIS toolboxes with tools that automate the process of assessing the potential for cell tower interference with tax parcel surveys in Los Angeles County. The SurvInt tools require two different types of input. First, text and directory paths are input through standard model builder dialog boxes. Specific examples of completed dialog boxes are given in Chapter 4, along with the resulting maps and reports the SurvInt tools produce using the example inputs. In addition, to direct user inputs through ModelBuilder dialog boxes, an example county-level Esri geodatabase was prepared and is required to operate the SurvInt tools. The contents and form of this

geodatabase are detailed in Chapter 3, specifically for the case study area of Los Angeles County.

To support the stated model objectives, two Use Cases formed the basis for the SurvInt design. In each use case, a single ModelBuilder model was built as a top-level structure to organize the main workflow operations. This allowed for quick model design iterations, which helped reveal development challenges early in the process. The top-level models were implemented using sub-modules, several of which provided useful standalone capabilities. These were designed to support spatial data post-processing after the main use cases created their output geodatabases. Esri (2021e) documents the functionality of ModelBuilder's Python script containers with which the author overcame several obstacles in the development of SurvInt. These obstacles included difficulties in debugging large and complex visual models and maintenance of functionality when input data formats changed. The combination of automation tools will also be a key enabler for planned future spatial data processing improvements, the refinement of physical models, and the addition of future SurvInt functionality discussed in Chapter 5.

#### **1.4. Structure and Organization of This Thesis**

The thesis is organized into five chapters beginning with this introduction. Chapter 2 provides a review of related literature in three broad areas, including the basics of the physics behind interference, previous studies of the impacts of interference on existing equipment, and a summary of the built-in geoprocessing tools available for automating analysis in ArcGIS applicable to this study. An explanation of the methods used in the project is provided in Chapter 3, including those used in the development of physics-based functions to compute cell signal interference metrics. Chapter 3 also includes a review of the form and content of the input data



sets that can be processed by SurvInt. Output datasets from example SurvInt sessions are presented and discussed in Chapter 4. The sample outputs in this chapter illustrate the visualizations generated by SurvInt in each of its two design Use Cases and show the functionality of the tool using various input data. Design decisions made during the development of SurvInt imposed some restrictions on its capabilities and indicated several avenues for extending the tool's functionality in the future. These limitations and opportunities are the subject of the fifth and final chapter of the thesis.

## Chapter 2 Related Work

Three distinct categories of existing literature had strong influences on the motivation for and development of SurvInt, as of July 2021: 1) the physics and engineering characteristics of cell phone signals; 2) studies of specific instances and types of interference experienced by existing survey equipment in common use today; and 3) methods of automation in GIS. This chapter provides a brief review, organized into these three categories of relevant published works.

Although the literature on these topics is extensive, based on this review, a parcel-based spatial analysis of cell signal interference and the inclusion of the mitigating effects of terrain blockage on the interfering signals are technical innovations unique to SurvInt, as of July 2021.

### 2.1. Related Physics and Engineering Works

The basic physics of RF propagation and the methods of signal interference have been studied for over seventy-five years. As confirmed by everyday experience in the real world, the fundamental factors that influence cell signal levels are visibility and distance. This is why finding land-based cell service in rural areas is challenging. There are few cell towers within range, and those that are in range can be separated by great distances.

SurvInt represents these concepts in the form of a line of sight, inverse-distance-squared propagation model first introduced by Friis (1946), which is one of the foundational engineering papers on RF (radio frequency) propagation. Friis (1946) describes two fundamental facts about RF signals used in the development of SurvInt. The first fact is that RF signals propagate in straight, unobstructed lines and cannot pass through terrain. This simple propagation law led the design of SurvInt to include the idea of visibility in its model of RFI. Esri (2021g) documents the details of the ArcGIS Pro visibility toolbox used by SurvInt to calculate terrain elevation

obstructions from a Digital Elevation Model (DEM). Friis (1946) includes a second key fact which states that the strength of an RF signal decreases as the square of the distance from its source. As documented by GPS.gov (2020, lower cell signal power means less interference with the weak GPS signals for which a survey receiver is designed to receive. Therefore, the further a cell signal gets from a cell tower, the less it can interfere with a survey receiver. This second fact motivates the use of the Euclidean distance raster in the calculation of the SurvInt interference metric. Publications by Hata (1980), Walrand and Varaiya (2000), the FCC (2011), and others over the last 75 years continue to support the basic propagation facts established in Friis (1946). This ensures that these facts form a solid foundation for SurvInt's interference calculations

The SurvInt propagation model is applied in the context of wireless phone networks that were first designed and deployed at the end of the twentieth century. The basic design parameters such as required power levels, antenna designs, and spatial distribution are described by Walrand and Varaiya (2000). Their explanation of high-performance communications networks provides valuable details about the challenges of designing cell phone network systems. Their work emphasizes that network design, specifically tower location, must take the potential for interference into account both within and external to the network, in terms of both distance and elevation. Walrand and Varaiya (2000) provide detailed descriptions of the potential impact of variable terrain when deciding where to locate cell towers. These descriptions of the impacts of variable terrain motivated the design of SurvInt to incorporate DEM-derived visibility analysis and confirm that the line-of-sight calculations included are aligned with basic RF network design principles.

Recently Mazzola (2020) provided another practical exposition of almost every aspect of cell network design and operation. Mazzola (2020) emphasizes that line-of-sight (straight line)

propagation makes cell tower siting a key consideration in the design of networks. His descriptions of the mechanisms of signal interference and the effects of terrain blockage directly influenced the implementation of terrain blockage computation in SurvInt. For example, Mazzola (2020) states that the power level at the receiving site is the key interference factor and explains that there can also be less significant secondary phenomena that affect total interference. Thus, SurvInt was designed to only utilize the received power level in the calculation of potential interference.

Government regulation and oversight of cell network operations have generated a vast literature on the basic modeling of RF propagation. The FCC (2011) issued a public safety bulletin that was a particularly influential motivation for the development of SurvInt. An important difference between this document and the previous more descriptive design surveys was the inclusion of quantitatively measured signal power levels at different radio frequencies and ranges from transmission towers. The bulletin's measured data help validate SurvInt's use of the Friis (1946) equation in studying the phenomenon of RF signal interference. This bulletin also provides alternative propagation models that include additional criteria or variables in addition to distance and visibility. These alternative equations, also based on measured signal levels, could form the basis of future enhancements of SurvInt.

## **2.2. Related Interference Studies**

This section briefly reviews four published investigations into the nature and severity of potential signal interference between cell phone signals and GPS/GNSS receivers. These projects provided both motivation and context for the development of SurvInt. These studies also offered a comparative type of validation procedure of the predicted severity of the problem. As described in the Results chapter, SurvInt predicts the spatially widespread potential for cell signal

interference. This result is aligned with the findings of other studies outlined in the following paragraphs.

GPS.gov (2020) published the official US Government position paper, which emphasized the importance of the issue of cell signal interference with GPS receivers. As noted in that paper, the U.S. government “works to minimize human sources of GPS interference through regulations interference detection and mitigation efforts.” The original motivation for the development of SurvInt was to assist in identifying and mitigating cell signal interference after the author personally experienced the effects of lost GPS signals during geological field surveys. This position paper also provided a link to the LightSquared (aka Ligado) cases described below. Those use cases are among the most significant examples of cell signal interference with GPS systems known to date which, have the potential to impact every user of GPS-aided navigation.

The USDOT (2018) study mentioned in Chapter 1 was one of many commissioned by U.S. Government agencies over the last twenty years. This USDOT report provided key context as well as quantitative inputs used in the development of SurvInt. As context, the report notes that high accuracy land surveys, the focus of SurvInt, are likely to be impacted by cell signal interference. These civil surveys are important to local governments, which rely on these surveys for the collection of property taxes. The report also quantifies how pervasive the interference problem is throughout the U.S. and how severe it can be for land survey receivers, such as those modeled in SurvInt. Figure ES-3 in the USDOT report (shown in Figure 2) guided the selection of both the numerical range of the SurvInt interference metric and SurvInt’s maximum interference distance limit, which is 3000m. Figure 2 illustrates the potential for complete loss of GPS functionality in a land survey receiver located less than 350m (magenta) from a cellular telephone tower and a loss of some degree of capability out to a range of 3000m (purple).

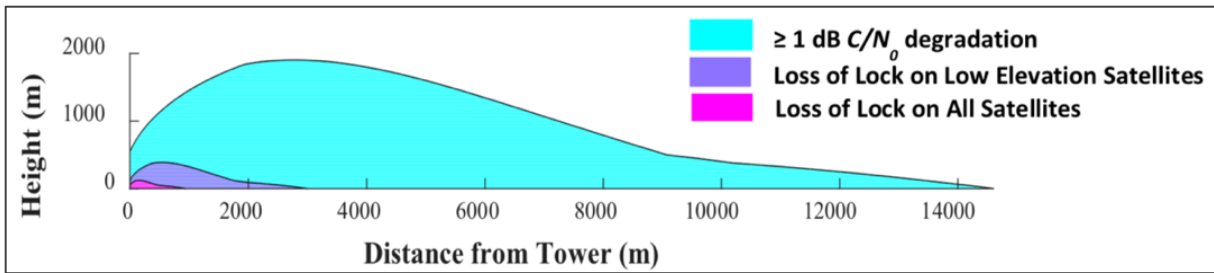


Figure 2 Impact of Cellular Base Station on High Precision GPS Receiver (USDOT 2018)

As illustrated by the USDOT (2018) report, the US Government has focused much of its RF interference characterization work on specific sources and victims of interference. For example, the significant cell signal interference from land-based satellite communication systems like LightSquared (aka Ligado) have been reported as crucial issues before the FCC bulletin was published (“Public Safety Tech Topic #17 – propagation Characterization” 2011; Huerta 2012). The case of LightSquared interference with GPS receivers provided an early example of the severity and spatial reach of this phenomenon, interference due to land-based satellite communication systems. Huerta’s (2012) assessment that interference could be a significant issue for commercial aviation was a specific motivation for developing SurvInt since, according to the FAA (2020), more than *one billion* passengers fly in the U.S. every year.

Studies of cell signal interference with specific GPS receivers like those made by Trimble (2014), a key land survey equipment maker, provide examples of how serious the inference issue can be in certain conditions of use. Trimble is the manufacturer of land survey receivers that are directly impacted by the kind of cell signal interference that SurvInt was designed to map. Trimble quantified the potential effects in a presentation before the DOT Adjacent Band Compatibility 2014 Conference. The Trimble (2014) presentation is a quantitative companion to the LightSquared article by Huerta. Their data corroborate the DOT findings of a serious potential for real-world interference with GPS from cell network signals and provide concrete

examples of high precision position equipment that are impacted by cellular signals operating within or near the GPS frequency bands.

Young et al. (2017) documented how increases in cellular bandwidth have exacerbated the interference problem. Their article points to fourth-generation (4G) cell towers, specifically LTE towers, as having a high potential for interfering with GPS receivers. Young's (2017) work supports an early SurvInt design decision to limit analysis to only those cell towers using 4G (LTE or GSM) signals. Not only did this design decision align with the scientific findings of Young et al. (2017), but it also reduced the total number of cell towers with the potential for interference. This reduction in data size enabled the use of SurvInt on more than one city at a time. The ability to analyze multiple cities allowed for a comparison of the effects of terrain on the interference phenomenon.

Cell signal interference is a worldwide issue that is well represented in international literature. For example, the Canadian company NovaTel (2021) described a tool they developed, called the Interference Tool Kit (ITK), to study the specific mechanisms and sources of interference in urban settings. Novatel created this product to help actively mitigate RF signal interference by locating its source and "canceling" the interfering signal digitally. This work did not include mapping nor land surveying and appeared after SurvInt was developed, yet represents a possible future enhancement of SurvInt, discussed in detail in Chapter 5. In its current implementation, when SurvInt detects the potential for severe interference with a tax parcel survey, it passively generates a letter that notifies an interfering tower's owner of the situation. In the future, SurvInt could be used to actively pre-plan the use of ITK-like digital interference cancellation for tax parcel surveys it finds at high risk for interference.

As stated above, the ITK provides interference location capabilities in addition to digital cancellation. Alves et al. (2018). reviewed the ITK as an example of a tool used for locating sources of interference in the same urban settings where SurvInt might be used. Their article provides a second idea for a future enhancement of SurvInt's capabilities. By using more sophisticated propagation formulas in place of the relatively simple Friis (1946) equation, a future version of SurvInt could describe both terrain-based and non-terrain-based RF signal obstructions in a report format. This would provide a more accurate assessment of the risk of severe interference with land surveys.

The SurvInt use cases testing described in Chapter 4 only validated the general functionality of SurvInt, not the accuracy of its quantitative predictions. Before the tool is deployed publicly to municipal users, validation through ground truth measurements of interference locations and levels should be completed. Wyatt (2018) provided a detailed description of methods used to locate and measure RF interference. This article includes helpful information about the engineering equipment and methods that would be needed for field validation of the accuracy of all results produced by SurvInt.

### **2.3. Related GIS Methods**

The literature on basic RF physics, such as Friis (1946), directly influenced the mathematical implementation of SurvInt, while Trimble (2014), Young et al. (2017), and USDOT (2018) documented the level of cell signal interference experienced by current surveyors using existing equipment and helped underscore the need for such a tool. Given this foundation and motivation, this thesis began as a demonstration of the wide-ranging capabilities of ArcGIS Pro automation applied to this important spatial problem. The development started with Allen's (2011) description of an ArcGIS Pro tool called ModelBuilder. ESRI publications, training courses, and



their technical support websites for Modelbuilder were indispensable in the development. For example, SurvInt’s user interface and operational controls employ standard ModelBuilder elements, including designating key inputs as “Model Parameters” and the organization of the models in a hierarchical structure. Lavery (2019) briefly and clearly explained these basic concepts along with some specifics of ModelBuilder scripting tools that allow for auto-formatting of model diagrams and the creation of sub-models. These concepts and tools were used extensively during the development of SurvInt.

A brief description of the ModelBuilder implementation, no matter how clear, provides only an outline of the development steps of developing a complex tool like SurvInt. Allen (2011) provided a comprehensive Esri workbook with hands-on examples that illustrated how to build and use ModelBuilder functionality. All of the most essential ModelBuilder capabilities and skills used in the development of SurvInt were acquired from this book. A particularly important example in Allen (2011) was the use of an iterator as part of a model. SurvInt uses an iterator to loop over a set of user-selected tax parcels to determine the susceptibility of surveys of those parcels to cell signal interference from nearby towers.

The physics-based, mathematical calculations in SurvInt could have been implemented in almost any programming language with output data in the form of large data files. However, a thorough understanding of the spatial distribution and variation of survey interference required visualization tools and analysis capabilities like those provided by ArcGIS Pro. Bolstad (2016) describes the application of GIS methods and technologies to a wide range of spatial problems. In particular, the spatial connections between the physics model and geographic entities like cities and tax parcels ran the gamut of basic GIS operations described in Bolstad (2016). Bolstad’s

book also explains the uses of key spatial operations such as clip, buffer, join, intersect, and union that are used throughout the models, and thus Python code, that comprise SurvInt.

As explained in Chapter 3, several of the most important and complex functionalities in SurvInt are to process and generate raster data. Keranen and Kolvoord (2014) documented many useful applications of GIS methods and technologies to a wide range of spatial problems involving raster data. Although their work is focused on rasters consisting of remote sensing data, Keranen and Kolvoord (2014) also describe more general GIS workflows in processing raster data and connecting it to vector feature classes. Concepts such as sampling, spatial joins, raster arithmetic and classification were translated directly into valuable functions in the SurvInt models.

Esri's ModelBuilder provided an excellent foundation for ArcGIS Pro automation, but the very large input datasets required for this study generated intensive processing loads that quickly overwhelmed this graphical tool. For processing an entire city's worth of cell tower data and tax parcels, a true programming language like Python was required. Esri (2013) foresaw this situation and integrated Python as a scripting language in ArcGIS 9.0 which was released in 2004. During the development of SurvInt, Esri (2013) was an invaluable user's manual which thoroughly documented all the important Python scripting capabilities in ArcGIS and provided key examples of practical code that were modified for use in SurvInt. For instance, Esri (2013) clearly described the process of integrating Python code into ModelBuilder "script" elements which were vital in handling large data sets in SurvInt.

Esri (2012c) has also created a dedicated, open-source Python module called ArcPy. Almost every Python-based GIS capability used in SurvInt was obtained by importing functions from the ArcPy module. Esri (2021e) is the official website for using Python in ArcGIS Pro and was

useful throughout the development of SurvInt as the source authority on ArcPy capabilities and the syntax of functions implemented in the ArcPy module. In addition to the formal syntax of ArcPy functions, Esri (2012c) also provides examples of practical, ArcPy-based GIS analyses that helped the author formulate coding solutions for specific challenges in creating SurvInt.

Most of the output products of SurvInt are in the form of maps or geodatabases. Slocum (2005) describes many of the basic principles of good cartography, which influenced the design of SurvInt's key visual outputs. In particular, the final visual data produced by the two SurvInt design use cases described in Chapter 1 are automatically generated maps with multiple feature layers. The automation ensures that the output maps have a consistent appearance that facilitates comparisons of results between different parcels and different cities. Slocum (2005) provides practical and wide-ranging advice on developing a "visual hierarchy" in thematic maps that provided guidance in designing the SurvInt output maps, which helped make the results more straightforward and more attractive to end-users. Appropriate choices of scale, color schemes, aggregation, and labeling of maps are specific examples of the influences that this work had on the development of SurvInt.

## Chapter 3 Methods

This thesis project encompasses the development of the SurvInt software as well as the acquisition of publicly available data used as inputs to the tool. Understanding the format and content of the input data was an integral part of the tool design, and thus the availability of data shaped the final product. Detailed descriptions of the required input data, the specific methods used to build the tools, and the methods used to implement the design use cases are all included in this section. These three topics represent the perspectives of the tool builder, and the tool users, at different points in the life cycle of the project.

### 3.1. Input Data Description

The two categories of spatial data inputs into SurvInt are related to two different aspects of quantifying the cell signal interference problem. The spatial data inputs influenced both the design of the SurvInt tool and the presentation of its results in map form. Data in the first category are related to fundamental physics of RF signals and networks. The second category of input data is related to SurvInt's purpose as a tool for mapping the extent of cell signal interference on land surveys, a unique innovation in land surveying strategy not previously invented or published to the best knowledge of the author at the time this work was accomplished. This data organizes the physics-based interference results into governmental units, typical property tax parcels, that are surveyed by cities and counties. Further information on the uses and sources of the input data is provided in subsequent paragraphs.

It is assumed that most readers have everyday experience with the physics of cell signals and can readily understand the purpose of this first category of inputs. RF signals propagate in straight lines, cannot pass through terrain obstructions, and diminish in strength with distance

from the source. Two key factors – visibility and distance –influence the strength (“number of bars”) of a cell signal at a given location. Thus, the input data in this category include both a digital elevation model (DEM) that is used to determine visibility, and a cell tower location database that is used to determine distance to nearby cell towers.

The second category of input spatial data is related to SurvInt’s purpose as a tool for mapping the effects of cell signal interference on commonly used land survey equipment. The design use cases for SurvInt involve all the key stake holders in the processes of commissioning, conducting, and using land surveys of property tax parcels in the state of California. It is assumed that the use cases for other states in the U.S. would be similar because the organization of property tax agencies into administrative and land surveying divisions seems inevitable. California’s municipal or county offices are typically in charge of conducting land surveys for the purpose of assessing and collecting taxes on individual parcels of land. Thus, the tax parcel and the city, county, and state in which it is located form the spatial basis of the property tax administration system. As a result, SurvInt requires polygon feature classes representing state, county, and city boundaries. The boundaries of property tax parcels in the county are also key spatial inputs to the model. The County of Los Angeles - eGIS (2020) was the public source of all the polygon feature classes used in SurvInt.

The Esri (2021b) training course, *Getting Started with the Geodatabase*, explains that the file geodatabase is the “native data storage format for ArcGIS ... for storing and managing your GIS data in one place.” The Los Angeles County Esri geodatabase created as part of SurvInt contains point and polygon feature classes and a DEM raster for the county. The polygons represent the boundaries of Los Angeles County, cities within the county, and tax parcels within the cities. The point feature class represents the cell towers located within Los Angeles County.

Finally, the USGS (2018) provided the county-wide DEM raster used in the visibility calculations, which creates an essential spatial data layer in the output of SurvInt. One example of the extensibility of these tools is that SurvInt can read and process county input geodatabases from *any other county* in California that provides its city and tax parcel boundary data as polygon feature classes in the form described by Esri (2019).

The specific geographic input data used in the development and testing of SurvInt are listed in Table 1. The table is divided into sections, one for each input data set. The data sets are labeled with either “physics” or “government,” reflecting their category as explained above. Each section includes a description of the data and its use in the project, the source of the data, and metadata fields. The metadata include the data model (vector, raster, file), the compilation date and license information. Scale, resolution and accuracy required by SurvInt are provided as applicable. Note that scale for a given data set represents the scale at which the full extent of the data is visible in ArcGIS Pro (i.e., “zoom to extent”). This project was greatly facilitated by the fact that all data are publicly available and were easily acquired by the author.

Table 1 SurvInt Spatial Data and Metadata

| Physics: Digital Elevation Map (DEM) (USGS 2018)       |             |   |            |           |             |
|--|-------------|---|------------|-----------|-------------|
| Description / Use in Project:                          |             | Provides raster values representing ground surface heights in meters above sea level based on a United States Geological Survey (USGS) digital terrain model (DTM). The DEM is used to calculate visibility of a parcel from a cell tower, a fundamental parameter in the interference calculation. |            |           |             |
| Data Source:   |             | <a href="https://www.arcgis.com/home/item.html?id=0383ba18906149e3bd2a0975a0afd8e">https://www.arcgis.com/home/item.html?id=0383ba18906149e3bd2a0975a0afd8e</a>   |            |           |             |
| Data Model:  | Raster      | Date:   | 5/15/2018  | License:  | Public data |
| Scale:   | 1:6,000,000 | Resolution:   | 1-arcsec   | Accuracy: | 30m         |
| Physics: Cell Tower Location and Attribute Table       |             |   |            |           |             |
| Description / Use in Project:                          |             | This .csv datafile contains the locations, types, and identification of cell towers worldwide. The dataset's spatial extent was limited to California to be tractable in the analysis, then converted to a point feature class.   |            |           |             |
| Data Source:   |             | <a href="https://opencellid.org/downloads.php?token=world">https://opencellid.org/downloads.php?token=world</a>   |            |           |             |
| Data Model:  | File (.csv) | Date:   | 10/25/2019 | License:  | Public data |
| Scale:   | 1:6,000,000 | Resolution:   | N/A        | Accuracy: | N/A         |
| Government: California State Boundary                  |             |   |            |           |             |
| Description / Use in Project:                          |             | This shapefile contains a feature class with a single polygon representing the boundary of the state of California. This boundary is used to restrict (clip) the extent of other spatial feature classes to the boundaries of the state to reduce the size of big datasets.                         |            |           |             |
| Data Source:   |             | <a href="https://data.ca.gov/dataset/ca-geographic-boundaries">https://data.ca.gov/dataset/ca-geographic-boundaries</a>   |            |           |             |
| Data Model:  | Vector      | Date:   | 8/9/2019   | License:  | Public data |
| Scale:   | 1:6,000,000 | Resolution:   | N/A        | Accuracy: | Unknown     |
| Government: Los Angeles County Boundary                |             |   |            |           |             |
| Description / Use in Project:                          |             | This shapefile contains a feature class with a single polygon representing the boundary of Los Angeles County. The boundary is used to limit the extent of the county geodatabase that stores the data required of calculations (city boundaries, cell towers, parcels, etc.).                      |            |           |             |
| Data Source:   |             | <a href="http://geohub.lacity.org/datasets/county-boundary">http://geohub.lacity.org/datasets/county-boundary</a>   |            |           |             |
| Data Model:  | Vector      | Date:   | 3/16/2016  | License:  | Public data |
| Scale:   | 1:800,000   | Resolution:   | N/A        | Accuracy: | Unknown     |
| Government: Los Angeles County City Boundaries         |             |   |            |           |             |
| Description / Use in Project:                          |             | This shapefile contains a feature class with a polygon for each incorporated city in Los Angeles County. This is the basic geographic unit of SurvInt.  |            |           |             |
| Data Source:   |             | <a href="https://hub.arcgis.com/datasets/7b0998f4e2ea42bda0068afc8eeaf904_19">https://hub.arcgis.com/datasets/7b0998f4e2ea42bda0068afc8eeaf904_19</a>   |            |           |             |
| Data Model:  | Vector      | Date:   | 5/15/2018  | License:  | Public data |
| Scale:   | 1:800,000   | Resolution:   | N/A        | Accuracy: | Unknown     |
| Government: Los Angeles County Tax Assessor Parcel Map |             |   |            |           |             |
| Description / Use in Project:                          |             | This feature class includes a polygon for each tax parcel in Los Angeles County. These parcels are the basic unit of the land survey process.   |            |           |             |
| Data Source:   |             | <a href="http://geohub.lacity.org/datasets/lacounty::parcels/data">http://geohub.lacity.org/datasets/lacounty::parcels/data</a>   |            |           |             |
| Data Model:  | Vector      | Date:   | 5/15/2018  | License:  | Public data |
| Scale:   | 1:800,000   | Resolution:   | N/A        | Accuracy: | Unknown     |
| Government: Cell Tower Provider Lookup Table           |             |   |            |           |             |
| Description / Use in Project:                          |             | An Excel table that allows a user to determine the cell service provider (network) that operates a tower with the potential to interfere with a survey. This information is used in constructing the notification reports.  |            |           |             |
| Data Source:   |             | <a href="https://www.arcgis.com/home/item.html?id=0383ba18906149e3bd2a0975a0afd8e">https://www.arcgis.com/home/item.html?id=0383ba18906149e3bd2a0975a0afd8e</a>   |            |           |             |
| Data Model:  | File (.xls) | Date:   | 12/3/2019  | License:  | Public data |
| Scale:   | N/A         | Resolution:   | N/A        | Accuracy: | N/A         |

### 3.2. Tool Development Methods

SurvInt was developed in several stages using several ArcGIS Pro automation systems. The initial development of the project relied on automation provided by the ModelBuilder tool in ArcGIS Pro. However, the relevant input datasets (parcel maps and cell tower catalogs) are extremely large, so, as the processing demands on the software and hardware increased, the study area was restricted to cities in Los Angeles County, and the tool development migrated from ModelBuilder only into ModelBuilder combined with the Python scripting language. The combination of built-in ModelBuilder elements with raw Python scripting allowed for efficiencies in development, execution speed, and debugging. Specifically, the calculation of a cell signal interference metric raster was implemented in Python for all three of these reasons. The most important change from the initial model concept to the completed tool was a shift from a “cell-tower-focused” to a “parcel-centric” tool that would be more useful to a surveyor assigned to survey a parcel rather than an in-depth investigation of nearby cell tower infrastructure. Once that conceptual shift occurred, the SurvInt design took shape quickly.

The formation of a raster dataset to represent the potential for survey interference of a cell tower with a given parcel is the computational heart of SurvInt. This key process is listed as step seven in Figure 1. It is implemented in a Use Case 2 submodel using both ModelBuilder controls and dedicated Python script tools. Based on basic RF signal physics, the two key parameters used in this computation are the distance from a cell tower and the elevation of the surrounding terrain. Details of the methods of calculating this crucial survey interference potential raster dataset are provided in the following section as part of the detailed description of Use Case 2.

Use Case 2 consists of three steps. In the first step, SurvInt generates interference metric rasters for each user-selected survey parcel. The rasters provide a clear visualization of both the



severity and the spatial extent of potential interference from nearby cell towers for each selected parcel. In the second step, a set of cell towers are identified as potential sources of interference for each parcel survey. This is done by extracting the interference metric raster value onto the point feature class representing the cell tower locations and then determining which cell towers meet two criteria related to the type of signal it uses and its proximity to the selected parcel. Any tower that meets the criteria is identified as a *critical interference tower* for that parcel. Finally, with the critical towers identified, SurvInt assists the surveyor by generating a report that can be sent to the operator of each identified tower. SurvInt automatically populates the report with the parcel's assessor identification number (AIN) and the predicted interference level from the identified tower. This data informs the operator of the level of interference expected and requests the operator take steps to mitigate it. An example report letter is included in the Use Case 2 results in Chapter 4.

Note, the two criteria that a cell tower must meet to be identified as a critical interference tower arise from two references discussed in section 2.2. First, Young et al. (2017) found that modern (4G) towers had the highest potential to interfere with GPS receivers. So, the Radio field in the cell tower feature class attribute table must be set to either 'LTE' or 'GSM' for SurvInt to identify it as a critical tower. Second, the normalized interference metric extracted at the tower location must be 100 or higher, which corresponds to a distance of less than 320m from one of the corners of the selected parcel. Referring to USDOT (2018), the magenta region shown in Figure 2 corresponds to the distance at which the potential cell tower interference can cause "loss of lock on all (GPS) satellites. In other words, at this distance, a cell tower can cause a complete loss of function in survey equipment that uses a GPS receiver for obtaining a position.

### 3.3. Use Cases and Methods of Operation

As shown in Figure 1, the development of SurvInt followed a standard software development outlined by Stone and Stone (2018). In the case of SurvInt, the step titled “*Identifying use cases, actors, and scenarios*” resulted in the identification of two use cases that eventually formed the foundation for this work. This section includes descriptions of both the development methods used to create the SurvInt models for the two use cases and the geospatial processing methods used by those models.

A high-level overview flowchart of the SurvInt workflow is provided in Figure 3. The workflow steps are explained in detail in this section of Chapter 3. The symbology in Figure 3 is similar to standard ModelBuilder models. Ellipses represent data, blue for major inputs and green for use case outputs. The yellow rectangles represent major workflow steps or processing which are symbolized like process elements in ModelBuilder. One important takeaway in Figure 3 is that the outputs of use case 1 provide the main inputs for use case 2.

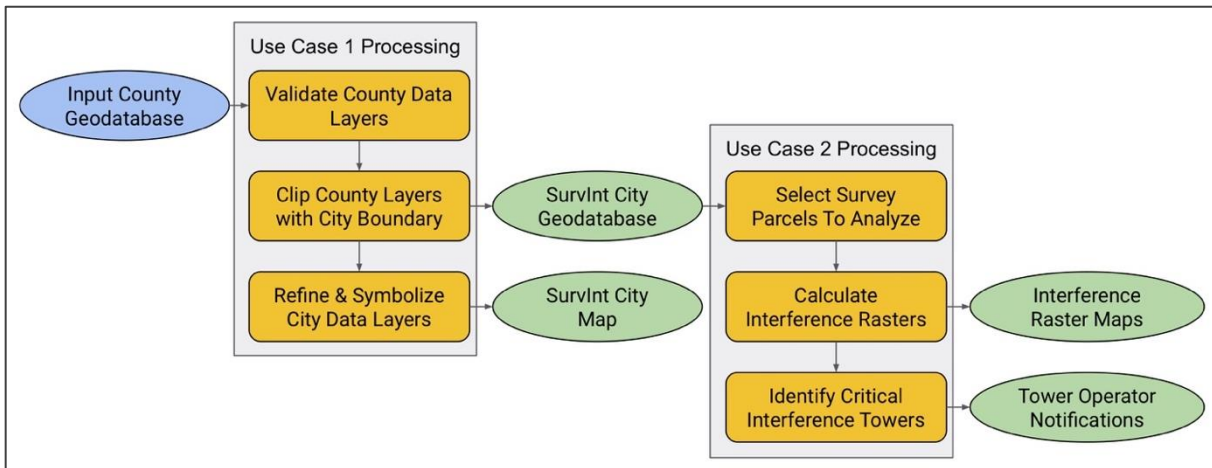


Figure 3 High-level SurvInt Geoprocessing Workflow with Two Interconnected Use Cases

#### 3.3.1. Use Case 1 - Overview

The goal of the first use case is to create and save a geodatabase (GDB) for one selected city that contains all the feature classes needed by a surveyor in making an interference assessment in

Use Case 2. The principal actor in the first use case (UC1) is a GIS administrator who can validate the Los Angeles County GDB and can run the SurvInt tool that creates the city GDB. Figure 4 shows the UC1 model as seen in ArcGIS Pro. The standard ModelBuilder symbology represents *processes* as yellow boxes and *data* as blue or green ellipses. Some data ellipses are marked with a “P” which indicates that those are model parameters that are typically the user inputs to the model. The top-level ModelBuilder processes (yellow boxes) in this use case are: 1) validate the Los Angeles County data, 2) select a city in the county and create a geodatabase for it that will be used by municipal surveyors employed by that city, 3) clip the county geodata using the selected city boundary to generate smaller sized datasets manageable in ArcGIS Pro, 4) refine (“polish”) the feature classes and 5) store the results in a newly created geodatabase. Details of the methods used in each of these processes are provided in the following subsections.

#### 3.3.1.1. UC1 Submodel: Validate County

Figure 5 shows the very simple submodel that is used to validate the county geodatabase (GDB). The UC1 model parameter, which holds a path to the county GDB, is the sole input. The only output is a Boolean flag indicating whether the county GDB has the required feature layers and that they are of the correct type. The main processing of UC1 only executes if the “County\_Is\_Valid” flag is true. The simplicity of this submodel is deceiving since the key functionality is implemented in a Python script. The four validation steps implemented in that code are shown in Figure 6.

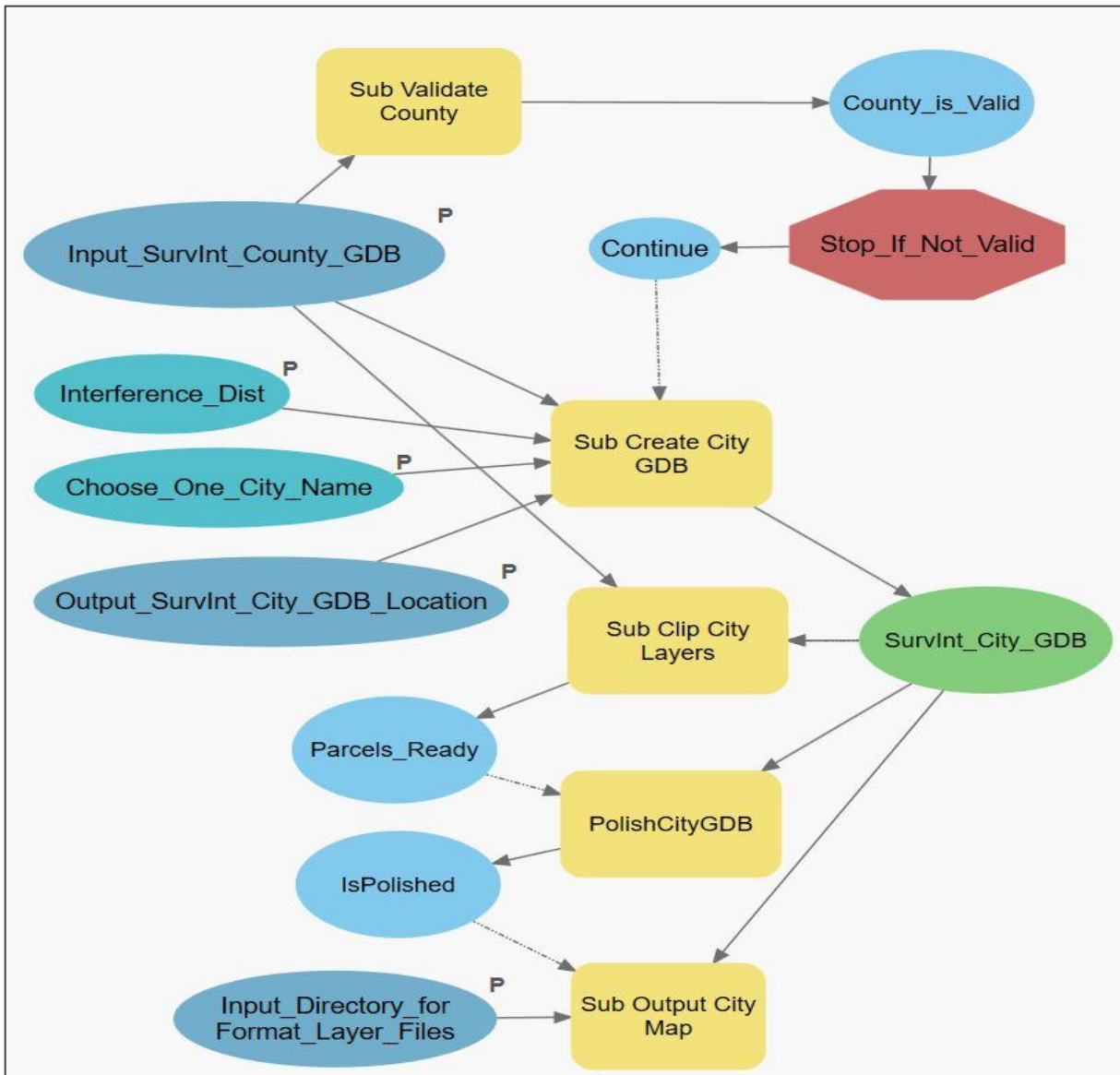


Figure 4 Top Level Model for SurvInt UC1 - Create and Populate a City Geodatabase

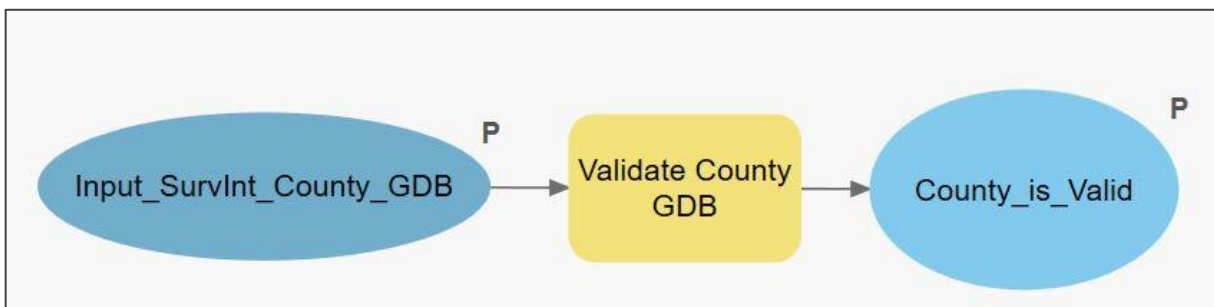


Figure 5 UC1 Submodel (1 of 5): Validate County GDB

```

#-----
# There are four validity checks. CHECK 1:
# First check to see if the .gdb file exists
#-----
if arcpy.Exists(countyGDB) == False:
    isValid = isValid & False
    errNum += 1
    errDict.update( {errNum : '>>>> Input geodatabase does not exist!'} )
    arcpy.AddMessage(errDict[errNum])
#-----
# Since it exists try to set it as the workspace
#-----
arcpy.env.workspace = countyGDB
#-----
# CHECK 2:
# Next check to see if it is a geodatabase
#-----
gdbDesc = arcpy.Describe(countyGDB)
if gdbDesc.extension != '.gdb':
    isValid = isValid & False
    errNum += 1
    errDict.update( {errNum : '>>>> Input is not a geodatabase!'} )
    arcpy.AddMessage(errDict[errNum])
#-----
# CHECK 3:
# Next, loop over the valDict keys and check to see if the required feature
# classes are present and have the correct types
#-----
for fc in valDict.keys():
    if arcpy.Exists(fc) == False:
        isValid = isValid & False
        errNum += 1
        errDict.update( {errNum : '>>>> Feature class "' + fc + '" does not
exist!'} )
        arcpy.AddMessage(errDict[errNum])
    else:
        fcDesc = arcpy.Describe(fc)
        if fcDesc.shapeType != valDict[fc]:
            isValid = isValid & False
            errNum += 1
            errDict.update( {errNum : '>>>> Feature class "' + fc + '" has the
wrong shape type!'} )
            arcpy.AddMessage(errDict[errNum])
#-----
# CHECK 4:
# Finally, check that the county DEM raster is present in the GDB
#-----
rastList = arcpy.ListRasters()
if rastList.count("DEM_Buf") != 1:
    isValid = isValid & False
    errNum += 1
    errDict.update( {errNum : '>>>> Elevation Raster "DEM_Buf" does not exist!'} )
    arcpy.AddMessage(errDict[errNum])

```

Figure 6 Python Code for Validating the County Geodatabase

### 3.3.1.2. UC1 Submodel: Create City GDB

The most complex submodel in SurvInt is illustrated in Figure 7. This submodel uses only built-in ArcGIS Pro geoprocessing tools and standard ModelBuilder construction methods. Model parameter inputs are a validated county GDB, a valid, user-selected city name, the path to a unique directory where the city GDB output is stored, and a critical parameter for interference analysis named “Interference\_Dist”. The USDOT (2018) study illustrated in Figure 2 suggests that the greatest distance at which significant interference can occur is approximately 3000m, which is the value of “Interference\_Dist” used in generating all results provided in Chapter 4.

As depicted in Figure 7, the submodel processing starts at the upper left and flows down and to the right. It begins with extracting the polygon representing the user-selected city from the Cities feature class in the county GDB. Then a GDB is created, given a name without special characters, spaces, dashes, etc., and stored in the directory specified by the input model parameter “Output\_SurvInt\_City\_GDB\_Location”. Once the GDB exists, the polygon layer of the selected city boundary is stored inside it. Finally, a buffered version of this polygon is created and stored in the city GDB as well. The buffer distance is set by the model parameter “Interference\_Dist”. The purpose of the buffered city boundary is to capture all of the cell towers located within the interference distance beyond the city boundary. Since a tax parcel can be located at or near a city boundary, and since RF signals propagate in straight lines if there is no physical obstruction, interference with survey equipment can come from towers located outside of a given city boundary. An example of this phenomenon is provided in Chapter 4. Including this buffer zone in the analysis is a crucial step in creating a comprehensive visualization of the spatial distribution of interference from all relevant cell towers. Without buffering, not all interfering signals will be represented in the results.

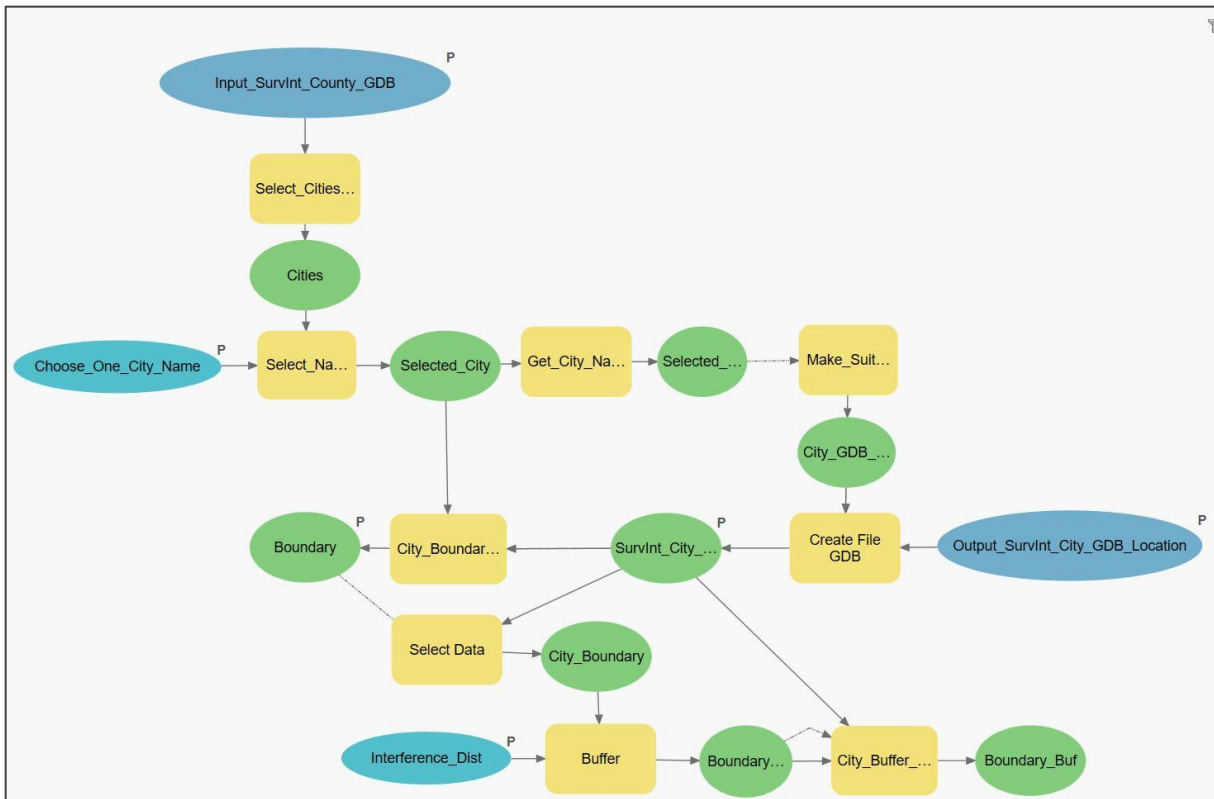


Figure 7 UC1 Submodel (2 of 5): Create and Populate a City Geodatabase

### 3.3.1.3. UC1 Submodel: Clip City Layers

The third step in the UC1 workflow uses the newly created city boundary feature classes to clip the cell tower, the DEM, and the parcel feature classes, included in the county GDB. The resulting city-specific feature classes are stored in the SurvInt city GDB created in the previous step. The submodel that implements this processing is shown in Figure 8.

Both the county and city GDBs are inputs to this model, and the processing proceeds from left to right. The large county cell tower class and the DEM raster are clipped using the buffered city output. The Parcels feature of the county GDB is clipped using the unbuffered city boundary since a municipal surveyor only needs to survey parcels within the city limits. The clipped outputs are stored in the city GDB.

The only output is a Boolean flag indicating that the clipping is complete. This was needed to address one of the most challenging aspects of automation using ModelBuilder, namely, execution control. There is no built-in ModelBuilder construct to cause model processes to execute in a particular order. However, ModelBuilder does provide the ability to use output data as a "precondition" for execution. Referring to Figure 4, the precondition is represented as the dashed line from the "Parcels\_Ready" data ellipse to the "Polish City GDB" process. The use of a "precondition" ensures that the polishing and map output functions do not execute before the feature clipping in this workflow step is complete.

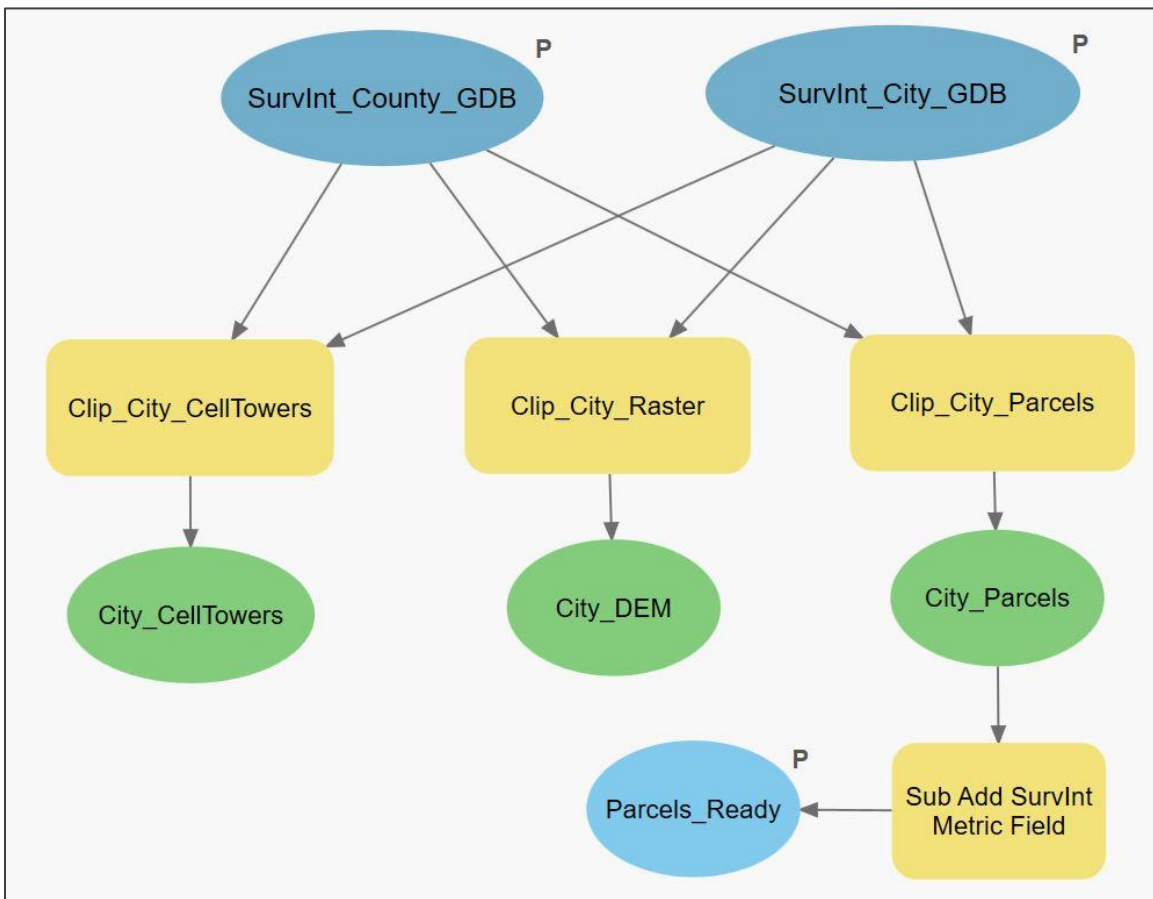


Figure 8 UC1 Submodel (3 of 5): Clip County Data with City Boundary Features



#### 3.3.1.4. UC1 Submodel: Polish City GDB

Testing SurvInt on many cities in Los Angeles County revealed a few cases where a cell tower was located exactly on the boundary of the buffered city-clipped DEM raster. This type of tower location caused errors in some raster processing tools, specifically the Euclidean distance function. The Euclidean distance calculation failed if a tower was located precisely one raster pixel outside of the clipped DEM raster. This error was difficult to find but represents an interesting example of the effects of the discrete nature of rasters when combined with geometric feature classes.

To solve the issue of towers located on the DEM boundary, the “Polish City GDB” process was implemented in a Python script tool which provided a way to work around this error. Figure 9 shows the relevant snippet of that code. The process removes any cell tower that is within one pixel (cellRes) of the buffered city boundary from the clipped cell tower feature class. In practice, this removed less than ten cell towers for the cities used to produce the outputs in Chapter 4. For some cities, no towers were removed. Note this issue cannot be addressed by adjusting the buffer distance. Since cell tower locations are very dense in the county, this pixel-level condition can occur at any buffer distance.

#### 3.3.1.5. UC1 Submodel: Output City Map

The final workflow step in use case 1 adds a new group of layers to an existing map in the SurvInt ArcGIS Pro project file. Figure 10 shows the ModelBuilder diagram for this map processing step. Like the Validate County submodel, the Output City Map submodel diagram is simple because all of the processing is implemented in a Python script tool.

```

#-----
# One other constant is useful. The maximum distance at which a cell tower
# can create significant interference. This is 3000 m throughout SurvInt
#-----
cellRes = arcpy.GetRasterProperties_management(minusR, "CELLSIZEX")
maxIntDist = 3000.0 # meters

#-----
# Set up a distance string that will be used to "inset" the buff by one raster
# cell to keep the cell towers from edge effects in visibility computations.
# Write out some messages for the execution history
#-----
distStr = "{0} meters".format(float(maxIntDist) - float(str(cellRes)))
arcpy.AddMessage("The max distance is: " + str(maxIntDist))
arcpy.AddMessage("The cell size is: " + str(cellRes))
arcpy.AddMessage("The new buffer distance is: " + str(float(maxIntDist) -
float(str(cellRes))))

#-----
# Second, the cell towers must be COMPLETELY contained in the DEM_Buf for
# computations to be guaranteed to work. I found one case where
# a tower was inside the polygon city buffer but one raster cell outside the
# DEM_Buf. This crashed the visibility tool!
#-----
arcpy.env.workspace = cityGDB
arcpy.Buffer_analysis("Boundary", "Boundary_Buf_Inset", distStr,
"FULL", "ROUND", "ALL", "", "PLANAR")
arcpy.Clip_analysis("CellTowers", "Boundary_Buf_Inset", "CellTowers_Inset")

#-----
# One of the key challenges of ModelBuilder is to control execution order.
# The standard way is to create an output and then use that as a 'precondition'
# for further processing. Creating that output is what the next line does
#-----
arcpy.SetParameterAsText(1, isPolished)

```

Figure 9 UC1 Submodel (4 of 5): Python Code Snippet of “Polish City GDB” Code  
(Removes Cell Towers on the Boundary of the Buffered City DEM Raster)

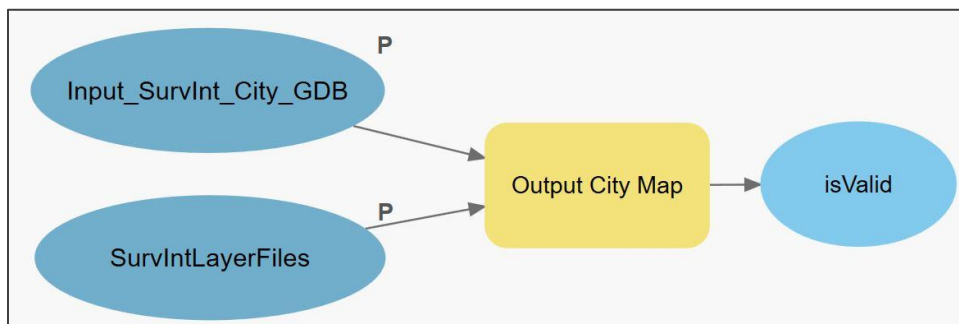


Figure 10 UC1 Submodel (5 of 5): Output a City Map with SurvInt Feature Layers

The ArcPy module associated with ArcGIS Pro version 2.8 differed considerably from the version distributed with ArcGIS Pro version 2.3 that was initially used to develop SurvInt. The changes affected the “Output City Map” Python code more than any other part of the SurvInt tool set. Specifically, a key mapping function, “ApplySymbologyfromLayer,” provided by the original arcpy.mapping library is no longer supported in the new arcpy.mp library (Esri 2021i). This made the application of consistent symbology across multiple automatically generated maps more challenging and was the single most time-consuming problem solved during the development of SurvInt.

The workaround represented by the code snippet in Figure 11 is to add pre-symbolized layers to the map and then to change the feature data associated with those layers to point to the recently created city data. This workaround utilized the new “UpdateConnectionProperties” method of the feature layer class (Esri 2019). The most important line of code using this method was added three lines from the end of this snippet. Although this workaround adequately accomplishes the intended goal, the code creates the visually annoying behavior of briefly causing the map layers to appear on the map for a second or two in a fixed location, then the visible map extent seems to “jump” to a final position. Despite this visual distraction, the output of this final step in UC1 is a correctly layered map in the correct location and an appropriate map view extent. This final map output represents all of the SurvInt layer data needed for a surveyor to execute Use Case 2.

```

#-----
# We start by adding a layer under which the other layers will be grouped. It is
# just a place holder for now. City layers will be added to it later
#-----
lyrPath = lyrDir + "\\\" + "CityMap.lyrx"
arcpy.AddMessage('>>>> dir layer ' + lyrPath)
grpLyr = m.addDataFromPath(lyrPath)

#-----
# Change the group layer name to the name of the city. Remove the SurvInt prefix
# since it is already in the map name and it clutters the display
#-----
grpLyr.name = cityName.replace("SurvInt_", "")
arcpy.AddMessage('>>>> group layer ' + grpLyr.name)

#-----
# The basic idea now is to loop over the list of feature classes in the city GDB
# and add a pre-formatted layer for each one of them. These come from .lyrx files.
# Then we change the data connection for each layer to the newly created feature
# classes in the city GDB.
#-----
for fc in cityLyrList:
    if arcpy.Exists(fc) == True:
        arcpy.AddMessage('>>>> Feature class "' + fc + '" exists!')
        lyrPath = lyrDir + "\\\" + fc + ".lyrx"
        inLyr = m.addDataFromPath(lyrPath)
        m.addLayerToGroup(grpLyr, inLyr, "TOP")
        m.removeLayer(inLyr)
    else:
        #isValid = isValid & False
        arcpy.AddMessage('>>>> Feature class "' + fc + '" does not exist!')

#-----
# Finally, we change the data connection for each new layer to point to the
# feature classes in the city GDB.
#-----
for fc in cityLyrList:
    fcLyr = m.listLayers(fc)[0]
    arcpy.AddMessage('>>>> Layer long name "' + fcLyr.longName)
    conProp = fcLyr.connectionProperties
    if conProp is not None:
        conProp['connection_info']['database'] = cityGDB
        fcLyr.updateConnectionProperties(fcLyr.connectionProperties, conProp)
    else:
        arcpy.AddMessage('>>>> Feature class "' + fc + '" does not exist!')

```

Figure 11 Python Code Snippet Creates a City Map with SurvInt Layers Required for UC2

### 3.3.2. Use Case 2 - Overview

The goal of the second use case is to calculate rasters whose values quantify the potential for cell tower interference, to use those rasters to identify the critical interference towers, and to

generate interference reports that can be sent to the cell tower operators. A city surveyor, trained in the use of ArcGIS Pro and the SurvInt tool, is the principal actor in the second SurvInt use case. As inputs, the surveyor uses an Assessor Identification Number (AIN) for a parcel and the city geodatabase created in the first use case. As illustrated in Figure 12, Use Case 2 (UC2) includes the following four process steps: First, the user selects a parcel within the city limits. Second, the SurvInt raster processing code calculates the survey interference metric raster for the selected parcels using distance and visibility and then extracts the values of that raster at nearby cell tower locations to identify those with the highest potential for severe interference with a survey of the selected parcels. Third, the model outputs the inference raster around the towers identified as creating critical interference onto the city map created in UC1. Last, SurvInt generates a report listing those critical towers which may require action to mitigate the most significant interference. Possible mitigations include adjusting the sensitivity of the precision surveying equipment or notifying network operators, and requesting that they take action to mitigate the interference. Both the developmental and spatial methods used in the first level submodels of UC2 are described in the following paragraphs.

The model diagramming conventions used in Figure 12 are the ModelBuilder standards, with blue and green ellipses indicating data stores and yellow rectangles representing processing. The model parameter inputs include a city GDB constructed using UC1, a query indicating which city tax parcels are to be surveyed, and a look-up table that associates a given tower with the tower ID and network operator. Additional inputs include a path to a directory of layer files (.lyrx) used to symbolize the output maps and a path to a folder used to store all SurvInt interference reports. Note, allowing the user to specify the location of the layer files as a

ModelBuilder model parameter input enables any city to substitute their layer file directory and apply any unique symbology standard required by their charter to symbolize maps.

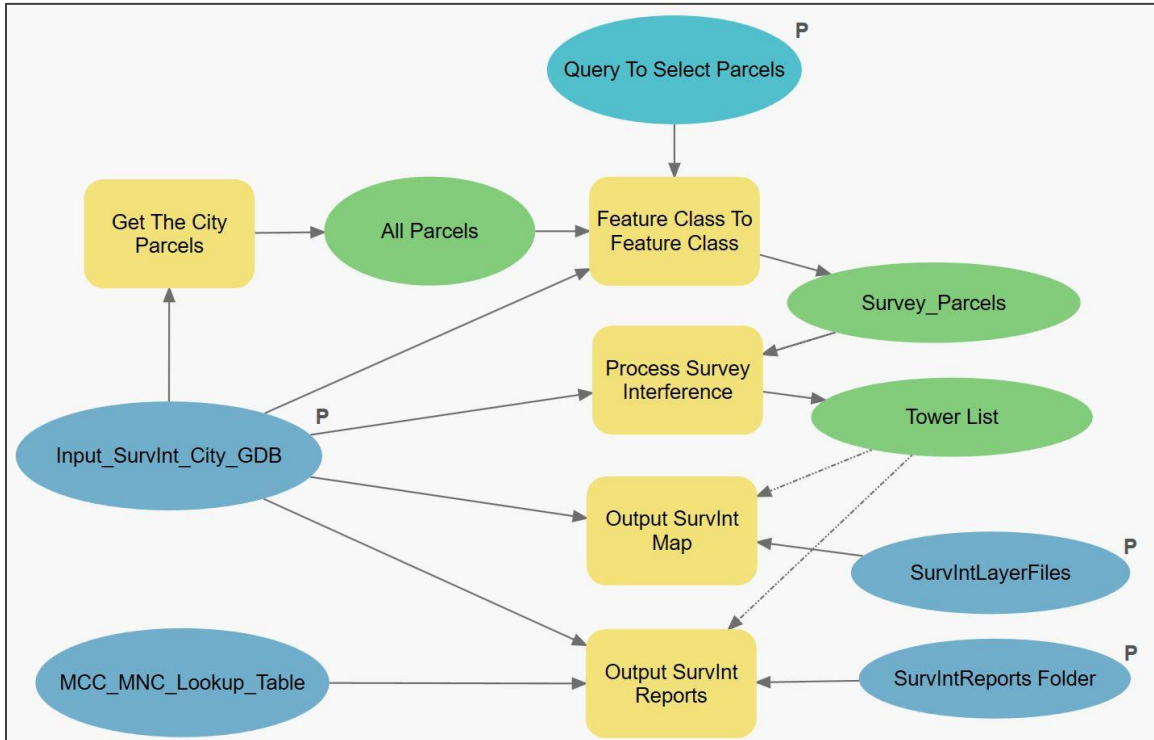


Figure 12 Top Level Model for SurvInt UC2 – Process Survey Parcels

Although they are superficially similar, the models for UC1 and UC2 differ in one crucial way. UC2 relies much more heavily on Python script tools than UC1. This design shift was the result of increased familiarity with ArcPy functionality by the author, increasing dataset size, and continuing technical challenges with controlling execution in standard ModelBuilder models. The latter is apparent on the right edge of the model diagram in Figure 12. The green Tower List data ellipse is attached to two of the processing blocks using dotted lines that indicate a “precondition.” Without creating the “Tower List” as an output from the Process Survey Interference block and using it to hold off execution, there was no way to prevent producing a blank map and empty report outputs generated before the interference metric data is created.

### 3.3.2.1. UC2 Submodel: Extract Survey Parcels

The five ModelBuilder elements (“Get The City Parcels”, “All Parcels”, “Query to Select Parcels”, “Feature Class to Feature Class”, and “Survey\_Parcels”) shown along the top of Figure 12 were implemented using built-in geospatial methods to extract the tax parcels specified by the surveyor/user in UC2 workflow step 1. The process “Get The City Parcels” uses a simple feature class selection tool to extract the Parcels feature class from the city GDB. Then an SQL query entered by the surveyor is processed by the built-in tool “Feature Class to Feature Class” to extract just the subset of parcels to be processed. Note the standard ArcGIS Pro user interface for queries allows the user to write SQL if they prefer or use the built-in data-driven query constructor that requires no prior experience in coding SQL.

### 3.3.2.2. UC2 Submodel: Process Survey Interference

The second workflow step in UC2 includes the most complex and time-consuming calculations in the SurvInt workflow. It also required the use of several forms of raster processing implemented in ArcPy-based Python code. Figure 13 shows the submodel diagram for this crucial process, which is implemented as an iterator over the selected parcels identified by their unique Assessor ID Number (AIN). During each iteration, a SurvInt metric raster is created, and the towers responsible for the most significant interference in that raster are identified as *critical interference towers*. These steps are accomplished in the Python script tool processes labeled “Create Parcel SurvInt Rasters” and “Find Parcel SurvInt Towers,” respectively. At the end of each iteration, both the interference metric rasters and the critical tower feature class are stored in the selected city GDB. At the end of the iterator, a built-in tool called “Collect Values” is used to create a list of all of the identified critical towers. As noted

earlier, this was done to control the execution of later steps in the UC2 workflow to ensure that the final outputs were generated after the raster and tower data were created.

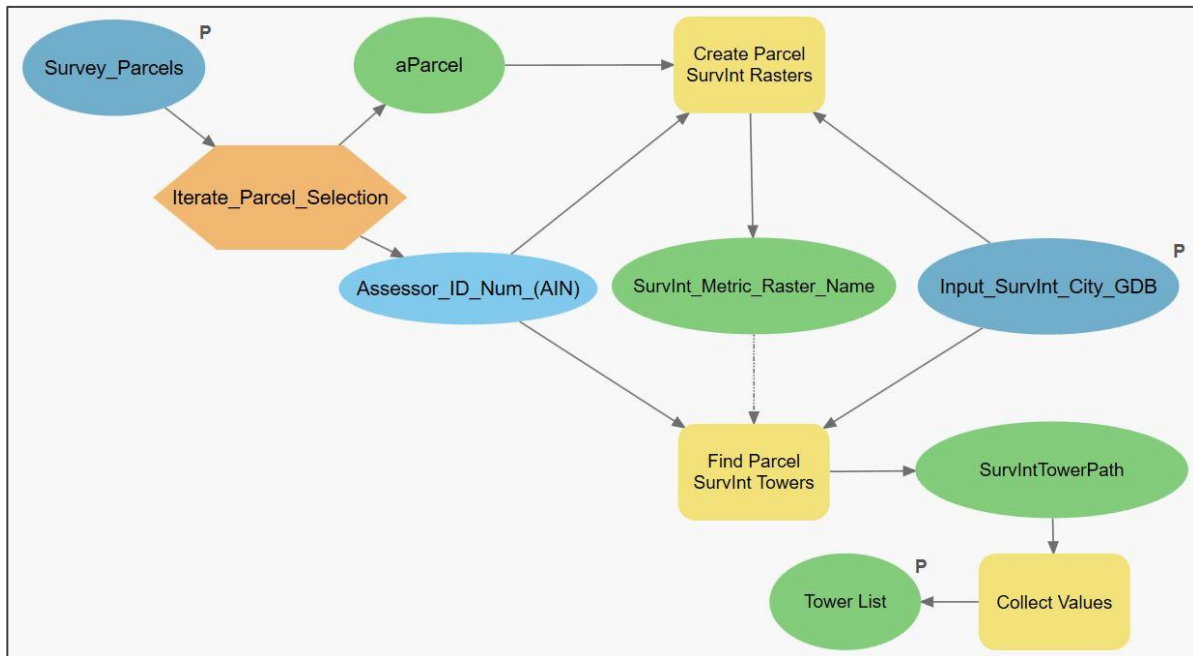


Figure 13 UC2 Submodel (1 of 4): Process Survey Interference

### 3.3.2.2.1 UC2 Submodel: Process Survey Interference – SurvInt Parcel Rasters

Figure 14 graphically illustrates the results of the numerous steps in the time-consuming raster calculations. The key methodological challenge throughout this processing was to align all the rasters at the pixel level and to ensure that they have the same extent and resolution so that ArcGIS Pro raster arithmetic can be used to combine them. The key to doing this was to set the clipped city DEM raster as the “Snap Raster” in the ArcGIS Pro working environment. The outputs of all other raster processing were then automatically aligned with the pixels in the DEM.

The raster processing shown in Figure 14 proceeds from left to right and top to bottom. Given a user-selected tax parcel, the top row of Figure 3 shows the three steps used to create a distance raster from each DEM cell to the nearest corner of the selected parcel’s polygon. The DEM is first converted to a constant raster with the value -1. The vertices of the survey parcel



(green polygon) are converted to a point feature class then used to create a raster that counts the number of corner points in that cell. Next, a mosaic raster is created from the point count raster and the DEM. This creates a raster with the spatial extent of the DEM but with positive values only at the parcel corner locations. The -1 values are converted to “NULL” and the EucDist function creates the raster, named “distR”, shown at the end of the first row.

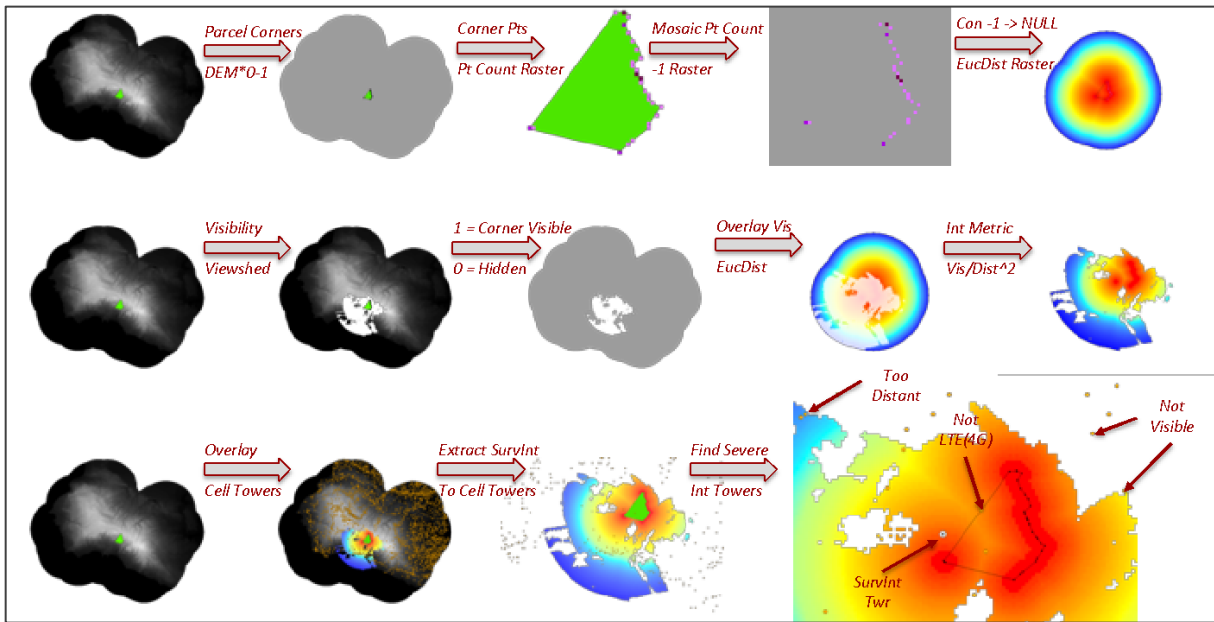


Figure 14 Calculation of the SurvInt Metric Involves Both Distance and Visibility Rasters

In the second row of Figure 14, the ArcGIS Pro spatial analyst function “Visibility” is used to create a “viewshed” raster from the parcel corners and the terrain model provided by the DEM. At each cell, the value of this raster is equal to the number of the parcel corners that are visible from that cell. Then each positive Surv cell value is set equal to 1 in this raster, and all other cells are set equal to zero. As shown in the third position in the second row, this raster (called “visR” in the code) forms a “visibility mask” that is used in the final computation of the SurvInt metric raster.

Before continuing with the description of the code for this submodel, it is worth highlighting some design decisions made during the development of this raster calculation. First, the maximum distance value used in the distR raster is limited to the model parameter “Interference\_Dist”, whose default value is 3km. It is true that the RF signal power from a cell tower does not abruptly drop to zero at this range and that the distance raster could have been extended to an arbitrarily long range. However, the 3km range used in the SurvInt distance raster is derived from the data underlying Figure 2. According to USDOT (2018), the 3km limit in distR encompasses the most severe cell tower interference with GNSS-enabled receivers according to the USDOT study. The visibility raster visR is used as a “mask” on the distance raster, as illustrated in Figure 14. This means that each raster cell has a binary value, 1 or 0 that represents if the selected parcel is either “visible” or “not visible” from a location. Of course, it is possible to imagine a probabilistic, rather than a binary, visibility raster, with values ranging between zero and one. The visibility probability might be influenced by factors such as level of urban development, the certainty of terrain elevation, atmospheric conditions, etc. However, the derivation of a probability density for a visibility raster was beyond the scope of this project.

With the two rasters, “distR” and “visR”, in hand, the model computes the interference raster. Friis (1946) states that the interference power radiated from a given tower decreases proportionally to the distance from the tower squared. So, the rasters are combined and scaled to provide the interference metric:

$$\text{SurvIntMetric} = \text{visR} * 10.0 * (1000.0 / \text{distR}) ** 2.0 \quad (1)$$

This metric is *normalized* to range from 1 to 1000 and is scaled to equal 10 at 1000m. The normalization procedure allows an easy-to-understand interference metric value of 100 to represent the potential for severe interference with the most sensitive survey equipment.

The processing represented in Figure 14 is implemented using ArcPy raster arithmetic functions in a single Python script tool. The main processing steps of that tool are shown in the Python code snippet provided in Figure 15. The first four lines create the parcel corner count raster. The next four lines create a Euclidean distance raster, “distR”, as illustrated in the top row of Figure 14. The next three lines create the viewshed raster “visR”. Finally, the formula for the SurvIntMetric given above is implemented using raster arithmetic provided in arcpy.sa. For example, multiplying by “visR” effectively masks the Euclidean distance raster, so the results have non-NULL values only for locations where a cell tower has a straight propagation line to at least one corner of the selected parcel. This embodies the principle in the Friis (1946) model that RF signals travel in straight lines through unobstructed space.

#### 3.3.2.2.2 UC2 Submodel: Process Survey Interference – SurvInt Critical Towers

The second raster-based calculation included in this submodel uses the parcel interference raster just created to identify the cell towers that contribute the most significant interference at the corners of the selected parcel. Figure 16 provides a snippet of the Python code that implements this calculation. First, the interference raster values are “extracted” to the point feature class representing the cell towers located within the buffered city boundary. The interference values are stored with each tower location in a field called RASTERVALU. Then a call to “Feature Class to Feature Class” applies the filter that defines a critical tower. As discussed in Chapter 2, the most significant interference comes from modern LTE and GSM cell towers. Also, the normalization of the SurvInt metric is such that a value above 100 indicates the potential for severe interference. So, the query used in the “Feature Class to Feature Class” method call in the fifth line of the code snippet in Figure 16 provides the definition of a critical tower. Finally, a field is added to the critical tower feature class to hold the AIN of the parcel with which that tower is interfering.

```

#-----
# The value of each cell in the Euclidean distance raster is the minimum distance
# to one of the surveyed "corner points" of the
# parcel. To start, construct a Point feature class with the vertices of the
# parcel polygon.
#-----
arcpy.FeatureVerticesToPoints_management(aParcel, "Corners", "ALL")
#-----
# Now convert this point class to a raster. Each cell has a value equal to the
# number of "corner" points it contains.
#-----
arcpy.PointToRaster_conversion('Corners', 'AIN', 'ptRaster',
                              "COUNT", "NONE", cellRes)
ptR = Raster('ptRaster')
#-----
# Use the Mosaic function to create a new raster that embeds the ptRaster in
# the raster of all -1. Setting the snap raster makes this work
# WGS_1984_Web_Mercator_Auxiliary_Sphere
#-----
arcpy.MosaicToNewRaster_management("ptRaster;minusRaster",cityGDB,"pt_min_Raster",
                                   "", "32_BIT_FLOAT",cellRes, 1, "MAXIMUM")
#-----
# Now set the -1 values to NODATA. This is necessary to use EucDistance
# The Con function replaces values > -1 with +1 and -1 with ""
#-----
seedRaster = Con("pt_min_Raster", 1.0, "", "VALUE > -1.0")
#-----
# With the seedRaster set up we calculate a EucDistance raster. Each cell has
# the distance to the nearest parcel corner out to a max distance of maxIntDist
#-----
distR = arcpy.sa.EucDistance(seedRaster, maxIntDist, cellRes)
distR = Con(distR, 100.0, distR, "VALUE < 100.0")
distR.save('distRaster') # Just for a map. Will be overwritten
#-----
# The other piece of calculating potential cell tower interference is to take
# into account whether there is a "line of sight" between the cell tower and
# a point on the parcel. We use the Visibility function to create a viewshed
# for the current parcel.
#-----
visR = arcpy.sa.Visibility(demBufPath, "Corners", "", "FREQUENCY", "", "",
                          "CURVED_EARTH", 0.13, 25.0, "", "", "", maxIntDist)
visR = Con(visR, 1.0, "", "VALUE > 0")
visR.save("visRaster") # Just for a map. Will be overwritten
#-----
# FINALLY, we compute the raster that shows potential interference for this parcel
# Since radio waves "spread out" as they propagate, the potential interference
# decreases as the square of the distance. The visibility raster in the numerator
# makes the potential interference 0 if the parcel is not visible from the tower.
# The SurvIntMetric is scaled to lie between 1 and 1000 with a value of 10 at a
# distance of 1000m. Any value greater than 10 represents severe interference.
#-----
SurvIntMetric = visR * 10.0 * (1000.0 / distR) ** 2.0
SurvIntMetric.save('SurvIntMetric_' + AIN)

```

Figure 15 UC2 Submodel (1 of 4): Python Code Snippet Creates SurvInt Parcel Rasters

```

#-----
# This script depends on two elements in the cityGDB that must have already
# been created. The SurvIntMetric raster and the Cell_towers point feature class.
# The values of the interference raster are extracted at the points corresponding
# to the tower locations. This assigns an interference level to each tower with
# respect to the input parcel's corners.
#-----
simRaster = arcpy.sa.Raster(cityGDB + '\\\' + 'SurvIntMetric_' + AIN)
arcpy.sa.ExtractValuesToPoints(cityGDB + "\\\" + 'CellTowers', simRaster,
                               cityGDB + "\\\" + 'SurvIntTowers',
                               'INTERPOLATE', 'ALL')

#-----
# Now that the towers have been equipped with the interference metric values
# a new feature class can be extracted from them. The selected subset has
# radio = LTE or GSM (i.e., 4G towers) and RASTERVALU (SurvIntMetric) >= 100.
#
# Use Feature to Feature with this query to create an output class with just
# critical towers
# NOTE: It is possible to have zero critical towers!
#-----
outFC = cityGDB + "\\\" + 'SurvIntTowers_' + AIN
arcpy.AddMessage("\n<<<<< " + "output feature class: " + outFC + " >>>>>\n")
arcpy.FeatureClassToFeatureClass_conversion('SurvIntTowers', cityGDB,
                                             'SurvIntTowers_' + AIN,
                                             "((radio = 'LTE') OR (radio = 'GSM')) AND (RASTERVALU >= 100.0)")

#-----
# Add a field into the new table to hold the AIN of the input parcel
#-----
arcpy.AddField_management(outFC, "AIN", 'string')
arcpy.CalculateField_management(outFC, "AIN", AIN)

```

Figure 16 UC2 Submodel (2 of 4): Python Code Snippet Identifies Critical Towers

### 3.3.2.3. UC2 Submodel: Output SurvInt Map

The Output SurvInt Map submodel is very similar to the Output City Map submodel of UC1. However, unlike in UC1, this map output function is strictly a Python script tool. It is not embedded in a small model, as shown in Figure 10. The fact that the script tool parameters are NOT inside a ModelBuilder model was a boon in this case. In ArcGIS Pro 2.8, script tool parameters can be directly associated with a layer file (.lyrx) for symbology purposes. The pure Python implementation of this submodel, as shown in Figure 17, meant that the workaround for

the absence of the “ApplySymbologyFromLayer” that was required in UC1 was NOT required in UC2. A comparison of Figure 11 and Figure 17 confirms that, in the latter case, there is no call to the “updateConnectionProperties method.”

```
#-----  
# We start by installing the ccurrent city map. It will be a group layer that  
# holds the specific data layers. Remove the "SurvInt_" prefix and just use the  
# city name for the group layer.  
#-----  
cityL = m.listLayers(cityName.replace("SurvInt_", ""))  
if len(cityL) > 0:  
    grpLyr = cityL[0]  
#-----  
# If the city map does not already exist then we should call the citymap  
# function. For now we just bail out (exit)  
#-----  
else:  
    arcpy.AddMessage('>>>> No City Map found!!!!')  
    arcpy.AddMessage('>>>> Try to run Use Case 1 first.')  
    sys.exit()  
#-----  
# Change the group layer name to just the name of the city. Remove the  
# SurvInt prefix since it is already in the map name and it clutters the  
# Contents display  
#-----  
grpLyr.name = cityName.replace("SurvInt_", "")  
arcpy.AddMessage('>>>> group layer ' + grpLyr.name)  
#-----  
# >>>><<<< Update due to arcpy.mp - Changed layer and grouping  
#-----  
#-----  
# Now add the SurvInt metric rasters to the new group layer  
#-----  
rasList = arcpy.ListRasters('SurvIntMetric_*')  
for ras in rasList:  
    rasPath = cityGDB + "\\\" + ras  
    arcpy.AddMessage('>>>> ras layer ' + rasPath)  
    inLyr = m.addDataFromPath(rasPath)  
    m.addLayerToGroup(grpLyr, inLyr, "TOP")  
    m.removeLayer(inLyr)  
#-----  
# Finally map the critical interference tower point feature classes  
#-----  
fcList = arcpy.ListFeatureClasses('SurvIntTowers_*', 'Point')  
for fc in fcList:  
    fcPath = cityGDB + "\\\" + fc  
    inLyr = m.addDataFromPath(fcPath)  
    m.addLayerToGroup(grpLyr, inLyr, "TOP")  
    m.removeLayer(inLyr)
```

Figure 17 UC2 Submodel (3 of 4): Python Code Snippet Outputs a SurvInt Metric Map

## Which Includes Interference Rasters and Critical Towers

### 3.3.2.4. UC2 Submodel: Output SurvInt Reports

SurvInt produces non-spatial output in the form of automatically generated notification reports intended for the network operators of cell towers that are identified as critical interference towers. As shown in Figure 18, this submodel uses standard Python text file output functions to combine the spatial and interference data computed in the preceding submodels with a form letter that can be sent to the network operator of each interfering tower. This submodel uses the model parameter “MCC\_MNC\_Lookup\_Table” to associate the tower identification number with the network that operates that tower. This lookup function is implemented in the first three lines of code in the snippet provided in Figure 18. In the SurvInt Map output submodel, each critical tower generates a point feature class that is stored in the city GDB with the name SurvInt + AIN. The code loops over each such feature class and extracts all the data needed to fill out the report that serves as a notification of interference to the tower operator. Once the data elements have been collected, a call to the SurvIntReport function (eight lines above the end of the snippet) creates the string, which is written to a text file. One report file is created for each critical tower identified in the raster process for that parcel.

Note that the UC2 model is parcel-driven. Each parcel in the user-selected list is processed in turn, as shown in Figure 13. That means that a given cell tower might be identified as a critical tower for more than one selected parcel. In that case, the SurvInt tool issues one notification file per parcel, even if that results in more than one letter going to the operator of a single tower.

```

#-----
# Loop over each critical tower FC found earlier in the SurvInt Use Case 2 model.
# For each critical tower generate a letter to the network operator.
#-----
fcList = arcpy.ListFeatureClasses('SurvIntTowers_*', 'Point')
#-----
# Next loop over each of the critical tower FCs and open a file to receive the
# reports. Open for writing and create the file if need be
# We then need five pieces of information to write the report:
# 1) The AIN
# 2) The city name (we already have this)
# 3) The cell tower network provider
# 4) The tower ID
# 5) The interference level from that tower.
# These will be inputs to the function SurvIntReport defined above
#-----
for fc in fcList:
#-----
# First extract the AIN from the FC name which has the form SurvIntTower_AIN
#-----
fcDesc = arcpy.Describe(fc)
AIN = fcDesc.name.split("_")[-1]
#-----
# Send message to the execution history at start of each FC
#-----
arcpy.AddMessage("\n<<<<< Reports for Parcel {0} >>>>>\n".format(AIN))
#-----
# Open (create if necessary) a text to store the notification letter
#-----
f = open(reportDir + "\\\" + 'SurvIntReports_' + cityName + "_" + str(AIN) +
".txt", "w+")
#-----
# The remaining three fields are extracted from a python cursor applied to the
# Look up table. Two cell tower point feature fields are named differently
# mcc = MCC in the lookup table and net = MNC in the look up table
#-----
sCur = arcpy.da.SearchCursor(fc, ['mcc', 'net', 'cell', 'RASTERVALU'])
if int(arcpy.GetCount_management(fc).getOutput(0)) > 0:
    for row in sCur:
        # First concatenate the mcc and net and use the lookup dictionary
        curNet = lookUpDict[ConCatKeys(row[0], row[1])]
        curTwr = row[2]
        curInt = row[3]
        # Now compute the form letter using SurvIntReport and write it to
        # the file
        rptStr = SurvIntReport(AIN, curNet, curTwr, curInt, cityNameSp)
        f.write(rptStr)
        # Also write the report to the execution history
        arcpy.AddMessage(rptStr)
else:
    f.write("<<<<< No interfering towers identified for this parcel. >>>>>")

```

Figure 18 UC2 Submodel (4 of 4): Python Code Snippet Generates SurvInt Reports  
(One for Each Identified Critical Tower)



## Chapter 4 Results

The results documented in this chapter confirm that the broad objectives of the project, listed in section 1.2, have been met. The SurvInt tool itself meets the first objective, which was to develop an ArcGIS tool to analyze spatial characteristics of RF interference. The details of SurvInt’s implementation are the subject of section 4.1. The second broad objective of the project was to demonstrate SurvInt’s ability to quantify and visualize RF Interference. Section 4.2 uses two design use cases to confirm the successful completion of this objective through a detailed review of the map and geodatabase outputs from each use case. Finally, SurvInt’s tools are applied to three cities in Los Angeles County with very different terrain elevation variability. The results of this application of SurvInt, captured in Section 4.3, indicate that terrain elevation may have a significant effect on the likelihood of cell tower interference when conducting tax parcel surveys.

### 4.1. The SurvInt Toolset

This project focused on the development of a set of ArcGIS Pro processing tools and geodatabases that could quantify and visualize potential cell tower interference with land survey equipment. The first and most important result was the successful creation and demonstration of SurvInt. This tool is composed of a set of files and directories that are described in Table 2.

Table 2 List of Components of the SurvInt Project

| Name                         | Contents  |
|------------------------------|---|
| SurvInt.aprx                 | ArcGIS Pro project file with one predefined map             |
| City_GDBs                    | A directory of city geodatabases created in UC1             |
| SurvIntLayerFiles            | Set of .lyrx files used to symbolize map outputs of SurvInt |
| SurvIntPython                | All of the Python scripts used in the SurvInt script tools  |
| SurvIntQueries               | Saved survey parcel queries that can be used in UC2         |
| SurvIntReports               | Directory for SurvInt generated critical tower reports      |
| SurvInt_Los_Angeles_County   | File geodatabase with county feature class and DEM data     |
| Griffin_SSCI_Masters_Project | File geodatabase with UC1 and UC2 toolboxes                 |

An overview of the ArcGIS Pro SurvInt project is shown in Figure 19. On the left-hand side of the figure, the contents pane indicates that there is only one layer in the map., the Los Angeles City boundaries. The map is required to be named SurvInt\_Cities so that the Python code can uniquely identify it. On the right-hand side of the figure, the Catalog pane lists the two key geodatabases (GDBs) and their contents, including the two toolboxes that implement UC1 and UC2. The Los Angeles County GDB only contains six feature classes and a DEM raster. The SSCI Masters Project GDB contains the use case toolboxes. The submodels and Python script tools detailed in Chapter 3 are listed under the toolboxes in which they are stored.

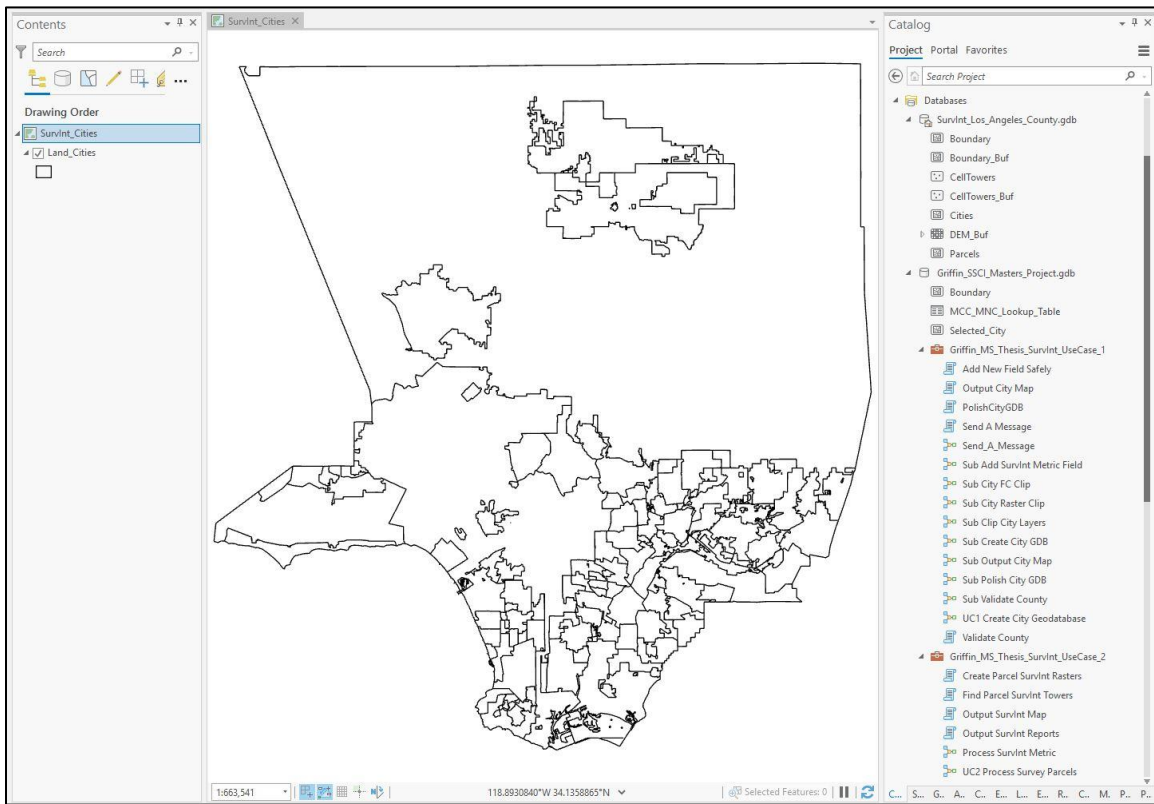


Figure 19 SurvInt Project in ArcGIS Pro Includes City Boundaries in Los Angeles County

Although the number of components in the SurvInt toolset is relatively small (only 37 objects in the Catalog pane), without any of the city GDBs it generates, this number is still large in terms of required storage space. For example, the Los Angeles County GDB occupies 1.24 GBytes of

storage space. The size of storage needed is mainly driven by the large number of cell towers and tax parcels in the county. There are over 184,000 cell towers and more than 2.39 million tax parcels in the county. The sizes of these two feature classes directly contributed to the decision to implement the UC2 model mainly in the form of Python script tools, as mentioned in previous chapters. Geoprocessing using Python is much faster than using Esri's ModelBuilder functionality, and the code written for this project was designed to process very large datasets.

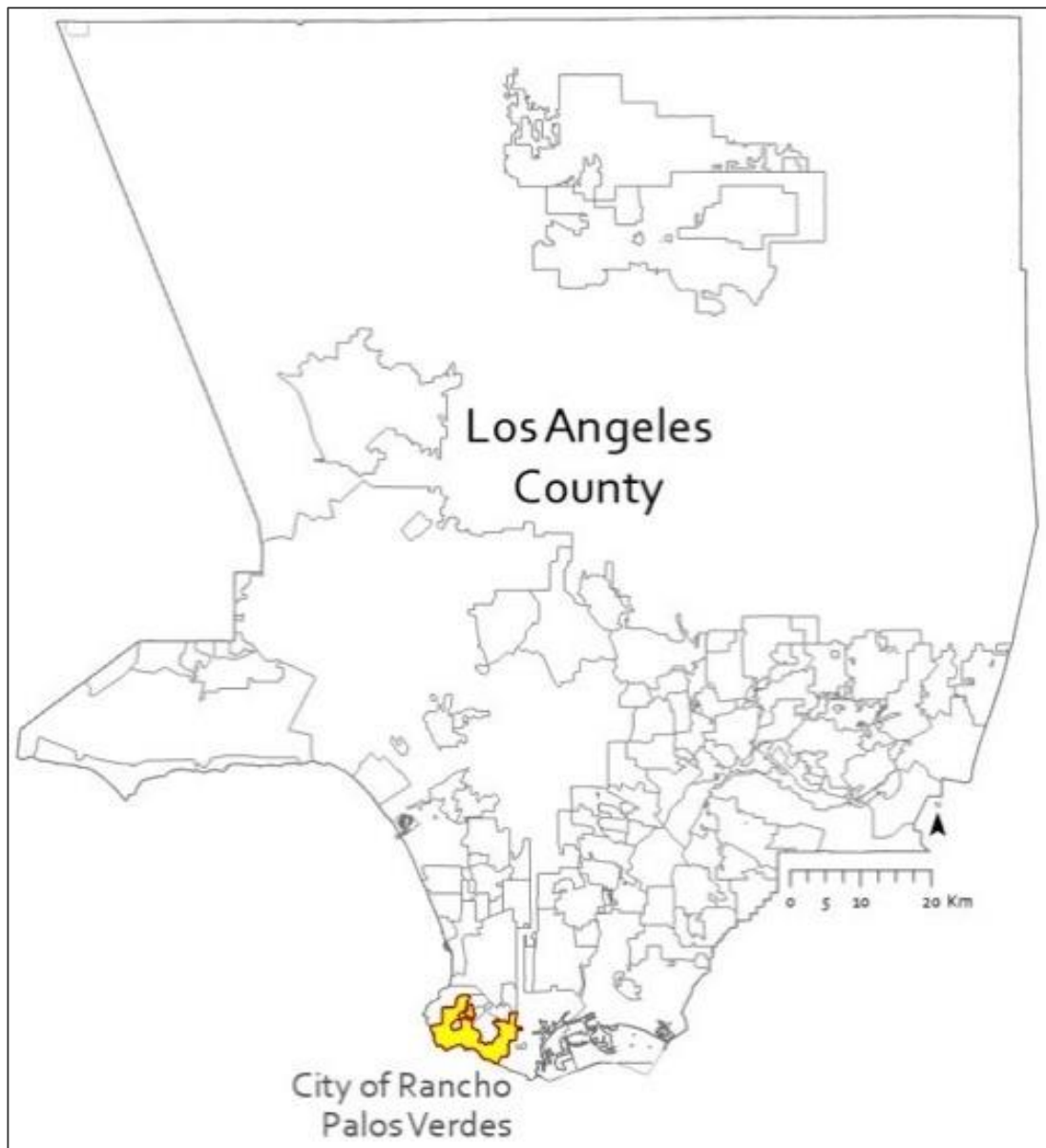


Figure 20 SurvInt Testing Used Rancho Palos Verdes (yellow) as a Study Area

## 4.2. Use Case Validation and Visualization Results

The output data results are the product of testing of the SurvInt methods described in Chapter 3. Here the emphasis is on how the results validate the specific functionality of SurvInt.

Although the tools have been tested on many cities in Los Angeles County, much of the development and testing used the city of Rancho Palos Verdes (RPV) as a study area. As shown in Figure 20, RPV is located on the Pacific coast in the southwestern corner of the county.

### 4.2.1. Use Case 1 – Validation Results

The model for SurvInt UC1 operates like any other geospatial processing tool in ArcGIS Pro. Double-clicking the tool in the Catalog pane brings up the input dialog shown in Figure 21. The dialog allows user selection of each of the model parameters described in Figure 4. The Input\_SurvInt\_County\_GDB parameter holds the path to the SurvInt\_Los\_Angeles\_County\_GDB shown in Table 2. A simple SQL query selects the city named Rancho Palos Verdes. The Output\_SurvInt\_City\_GDB\_Location is a path to the directory in which the city GDB is stored. The Input\_Directory\_for\_Format\_Layer\_Files is the path to the lyrx files used to symbolize the output maps. Lastly, based on the USDOT (2018) interference study, the critical physical parameter Interference\_Dist is set to 3km (3000m) in accordance with Figure 2.

The layered map shown in Figure 22 is the result of executing the SurvInt UC1 tool with the input parameters shown in Figure 21. Note that the majority of cell phone towers within the interference distance are located *outside* the city limits. This underscores the need to use the buffered city limits to clip the cell towers feature class and the DEM. If the city boundary was used for this clip process, most of the potential interference sources would be removed from the analysis and would lead to the undesirable result of underestimating the potential for interference.

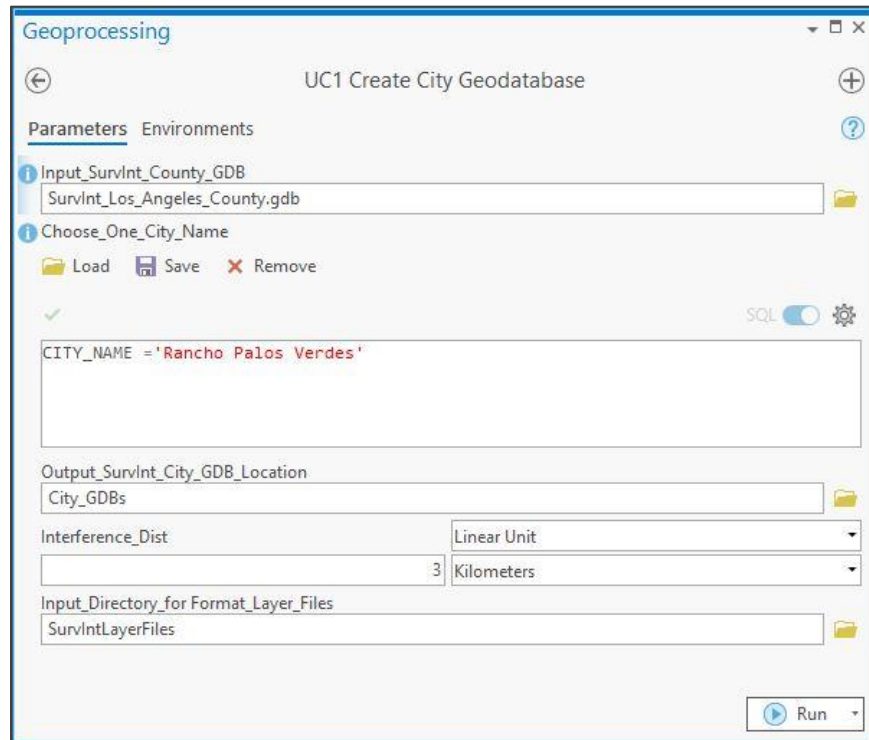


Figure 21 Processing Dialog for SurvInt Use Case 1 (UC1) – Model Parameter Inputs

The geospatial data processing provided by the SurvInt model *UC1 Create City Geodatabase* includes the five workflow steps described in section 3.3.1. The implementation of these steps in the use case model is shown in Figure 4 and is reproduced in the left panel of Figure 23. The results of this data processing at the use case level are a city GDB and a layered map of the city that includes the city boundary, a DEM clipped to the buffered city boundary, the tax parcels located in the city, and all the cell towers that are close enough to interfere with surveys of those parcels. The thumbnails shown on the right side of Figure 23 represent the step-by-step process of assembling the city map outputs of UC1 using the city of Rancho Palos Verdes as an example. The map on the right side of the figure is the visual output of running the UC1 model in the SurvInt project in ArcGIS Pro.

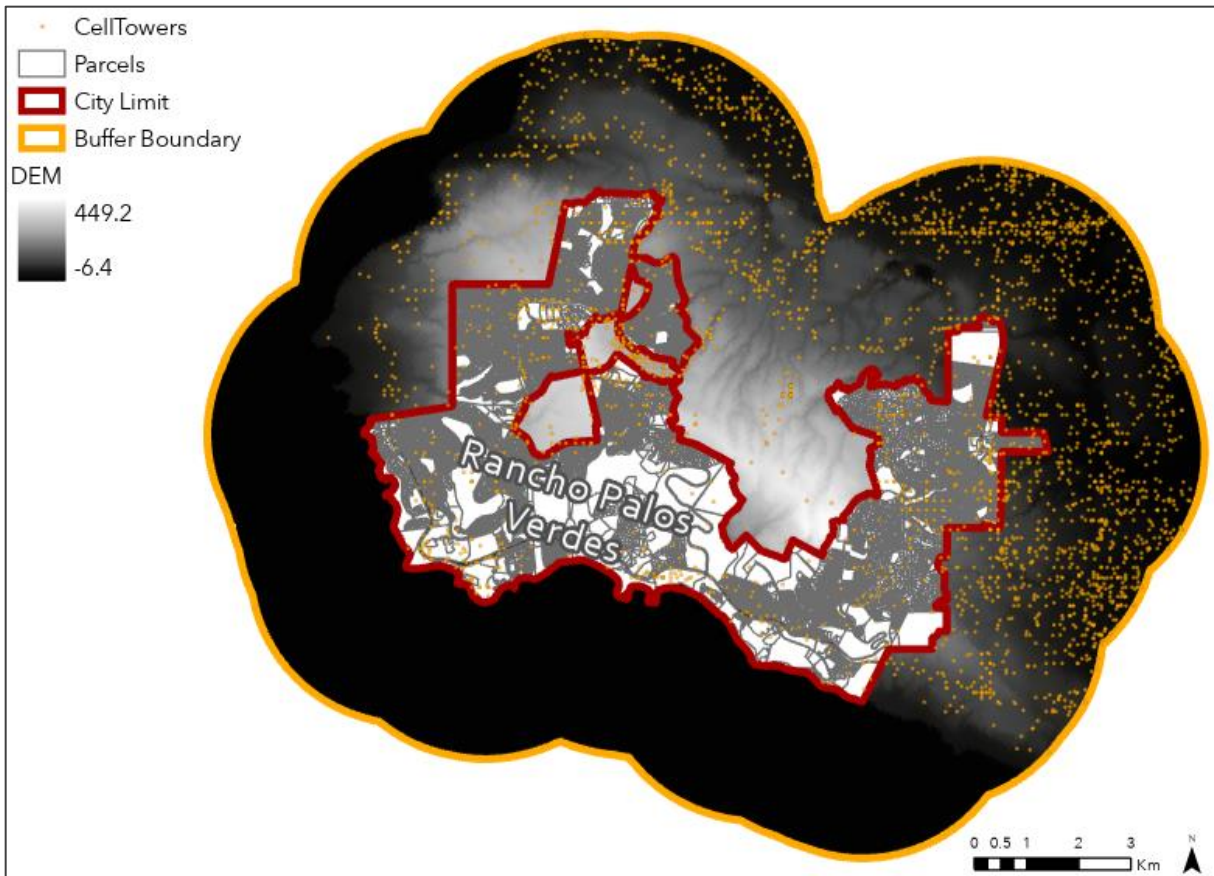


Figure 22 Visual Result of SurvInt Use Case 1 – Layered City Map of Rancho Palos Verdes

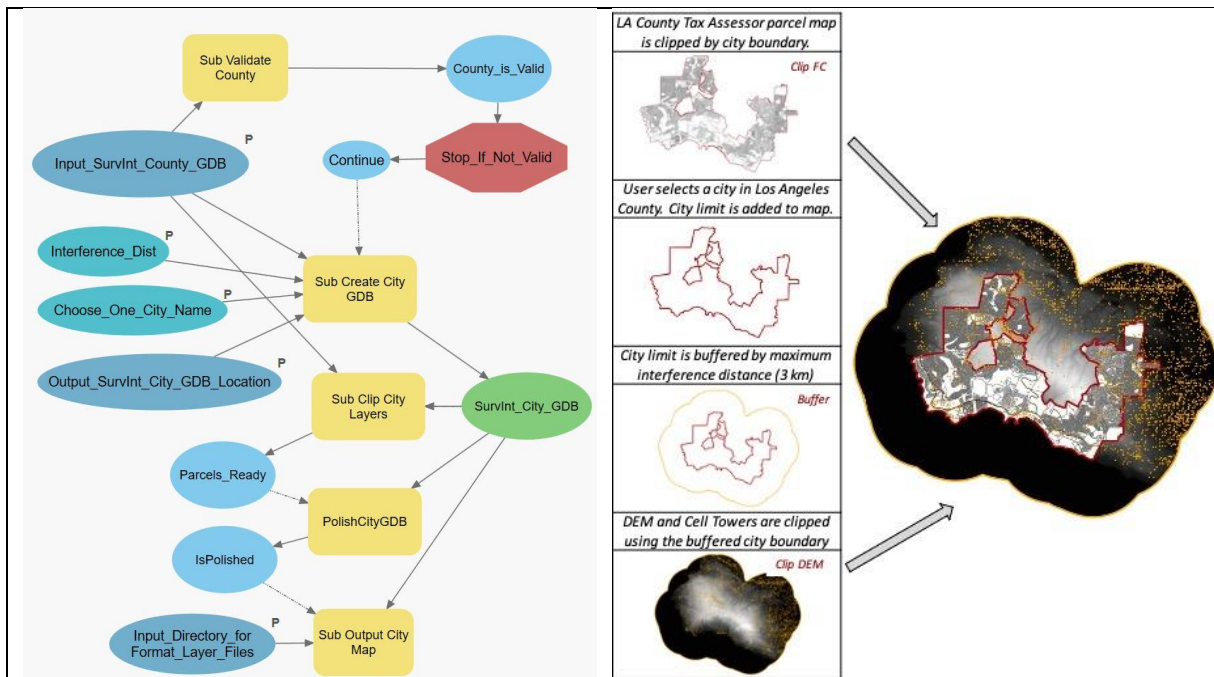


Figure 23. Example Output of SurvInt UC1 - Create and Populate a City Geodatabase

Although the map output of UC1 is an essential first step in visualizing the potential for cell signal interference in a city, the most important result of UC1 is the creation of the city GDB, which is a required input to the processing in UC2. This crucial data output is represented in Figure 24. The highlighted geodatabase `SurvInt_Rancho_Palos_Verdes.gdb` was created and stored in the `City_GDBs` during the execution of UC1. The critical feature classes are the city boundary, buffered boundary, buffered DEM, cell towers in the buffered boundary, and the city parcels class. The feature class `CellTowers_Inset` is the product of the processing (described in section 3.3.1.4) in the “Polish City GDB” submodel, which removes cell towers within one DEM pixel of the buffered boundary to avoid an error in the Visibility spatial analyst function.

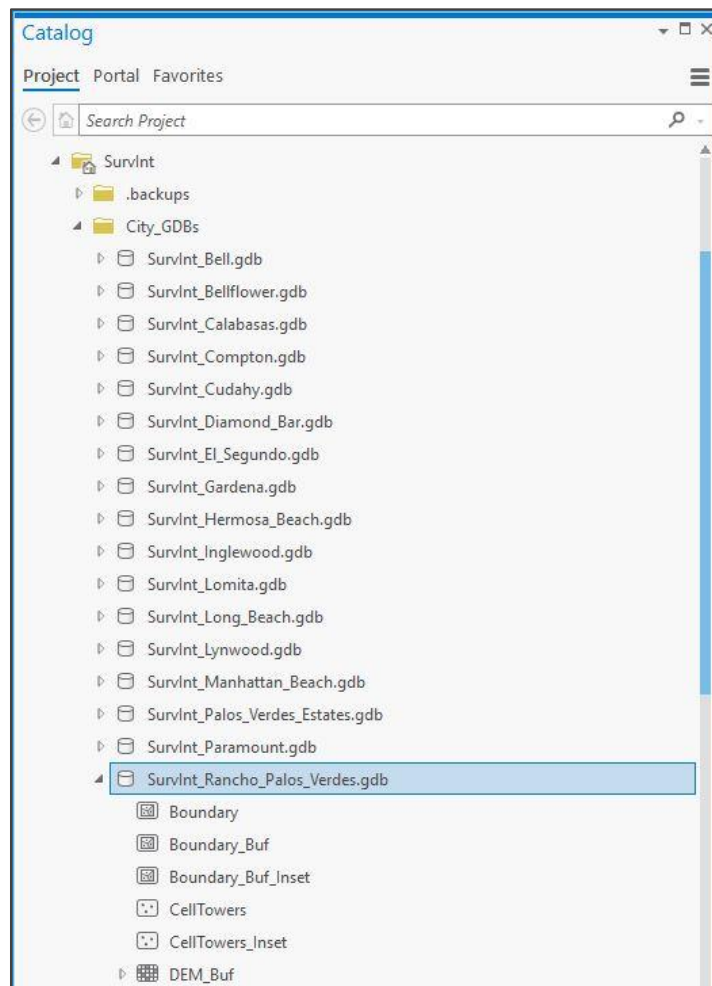


Figure 24 Output Data Result of UC1 – SurvInt Rancho Palos Verdes GDB

Note that the workflow steps of UC1 generate geodatabases and layered city map outputs that are specific to each selected city and are independent of each other. Although not envisioned in the original concept of UC1, this set of steps allows a GIS administrator to create the geodatabases and maps for more than one city at a time. The geodatabase catalog shown in Figure 24 confirms this iterative feature of SurvInt and shows the GDB results after testing UC1 on 17 different cities in Los Angeles County. Figure 25 illustrates the visual results of UC1 accumulated for three cities, Rancho Palos Verdes, Compton, and Signal Hill, located in the southern part of the county.

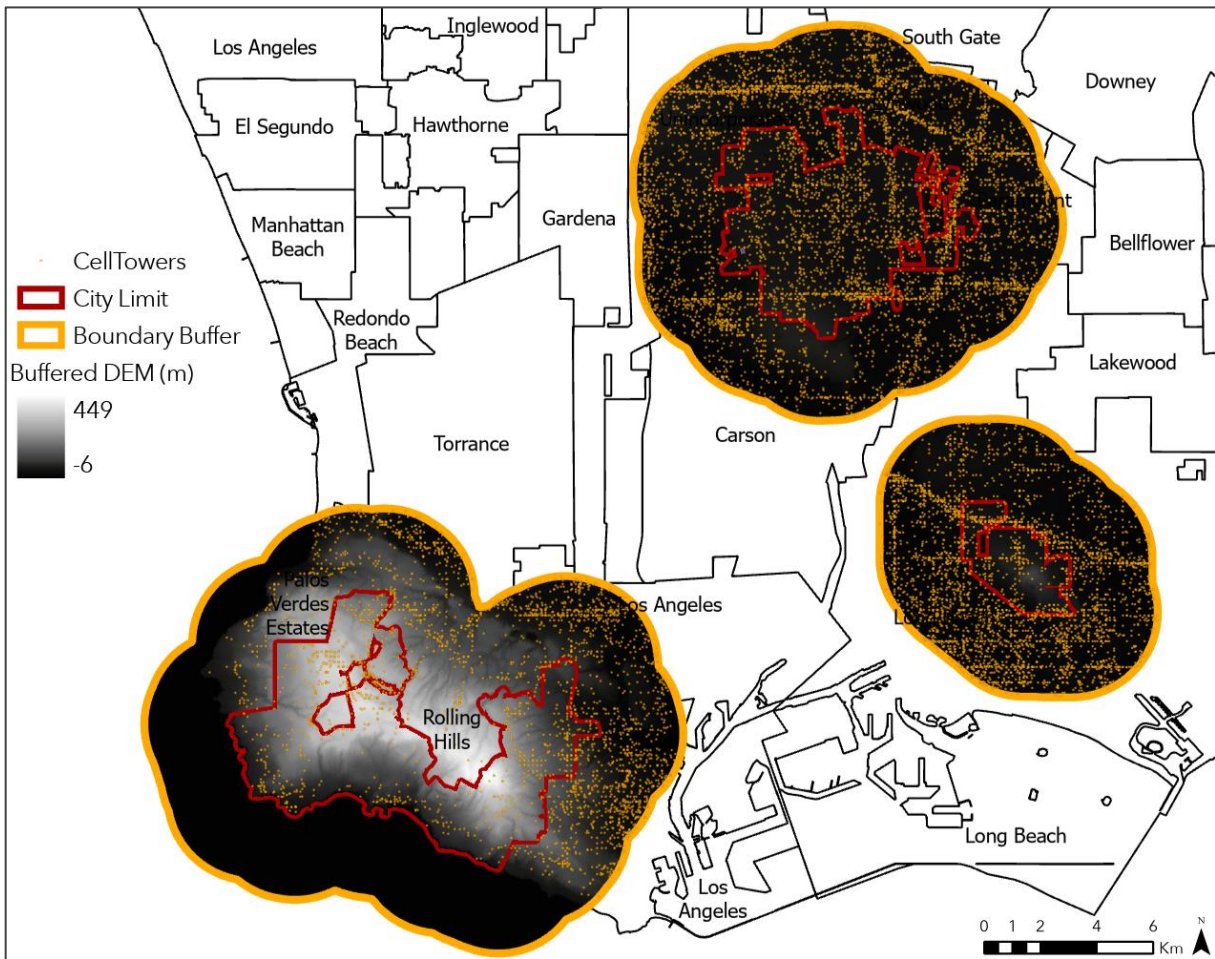


Figure 25 Multiple Runs of UC1 Create City Maps Accumulated on the County Map



#### 4.2.2. Use Case 2 – Validation Results

To run the ArcGIS Pro model for SurvInt Use Case 2, the user activates the geoprocessing input dialog by double-clicking the tool in the Catalog pane. The UC2 input dialog is shown in Figure 26. The dialog allows user selection of each of the model parameters described in Figure 12. Figure 26 illustrates the parameters that were used in the development and early testing of SurvInt UC2. The Input\_SurvInt\_City\_GDB parameter holds the path to the SurvInt\_Rancho\_Palos\_Verdes.gdb created by running UC1 with the parameters shown in Figure 21. An SQL query is used to select a subset of the parcels in Rancho Palos Verdes. In this test case, only one parcel with the AIN 7586026041 was selected. Parcel selection is the single piece of information that will be added by a surveyor assigned to survey this parcel. The parameter SurvIntLayerFiles stores the path to the .lyrx files used to symbolize the interference metric rasters and the associated critical towers. Lastly, the SurvIntReportsFolder parameter provides the path to the folder where all the SurvInt notification letters are stored.

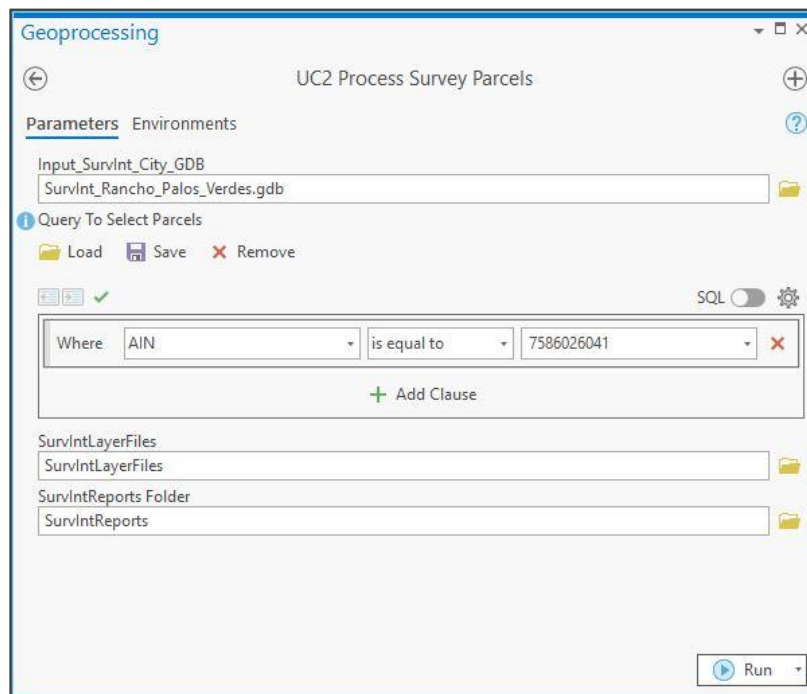


Figure 26 Processing Dialog for SurvInt UC2 – Model Parameter Inputs

Figure 27 shows the layered map output that resulted from executing UC2 with the parameters in Figure 26. This map was built from the UC1 map in Figure 22 with the addition of a SurvIntMetric raster and the four critical interference towers that have the potential to create severe interference with a survey of the parcel 7586026041.

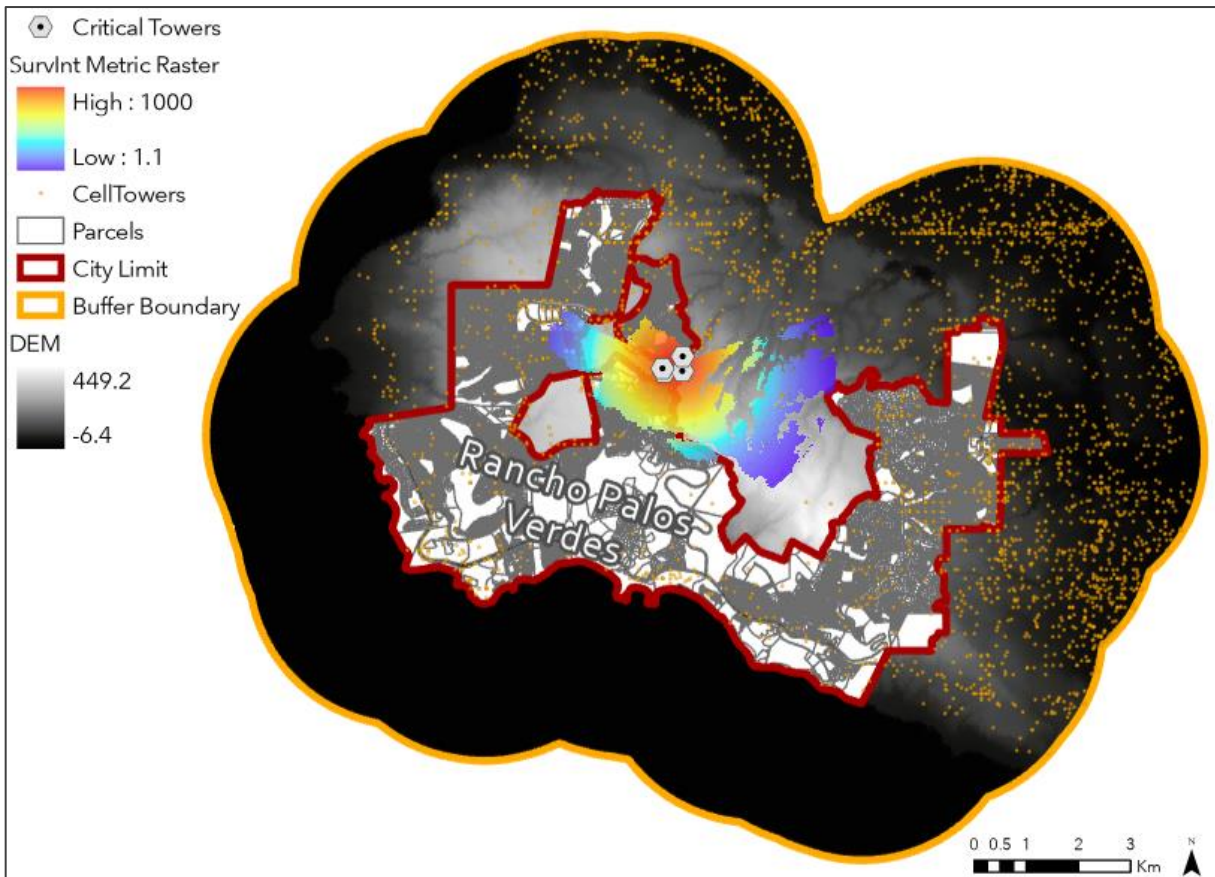


Figure 27 Visual Result of SurvInt UC2 –SurvInt Raster and Critical Towers in RPV

Increasing the scale of the map in Figure 27 provides a clear example of the importance of including cell towers outside the city limits in the interference analysis. Their priority was only suggested by the distribution of cell tower locations shown in Figure 22. The small-scale map in Figure 28 clearly illustrates that a majority of the critical interference towers near parcel 7586026041 are located outside the boundary of RPV. In Figure 28, the selected parcel is symbolized in purple. The four towers identified by SurvInt as the critical interference towers are

shown as gray hexagons. The critical towers are labeled with their normalized interference metric, which ranges from 103.8 to 741.1. Three out of four of the critical towers lie outside the boundary of the city, which is indicated with a dark red line. While it is true that the boundary of RPV is very complex in this neighborhood, this result still underscores the importance of including all potential interfering towers in the SurvInt city GDB.

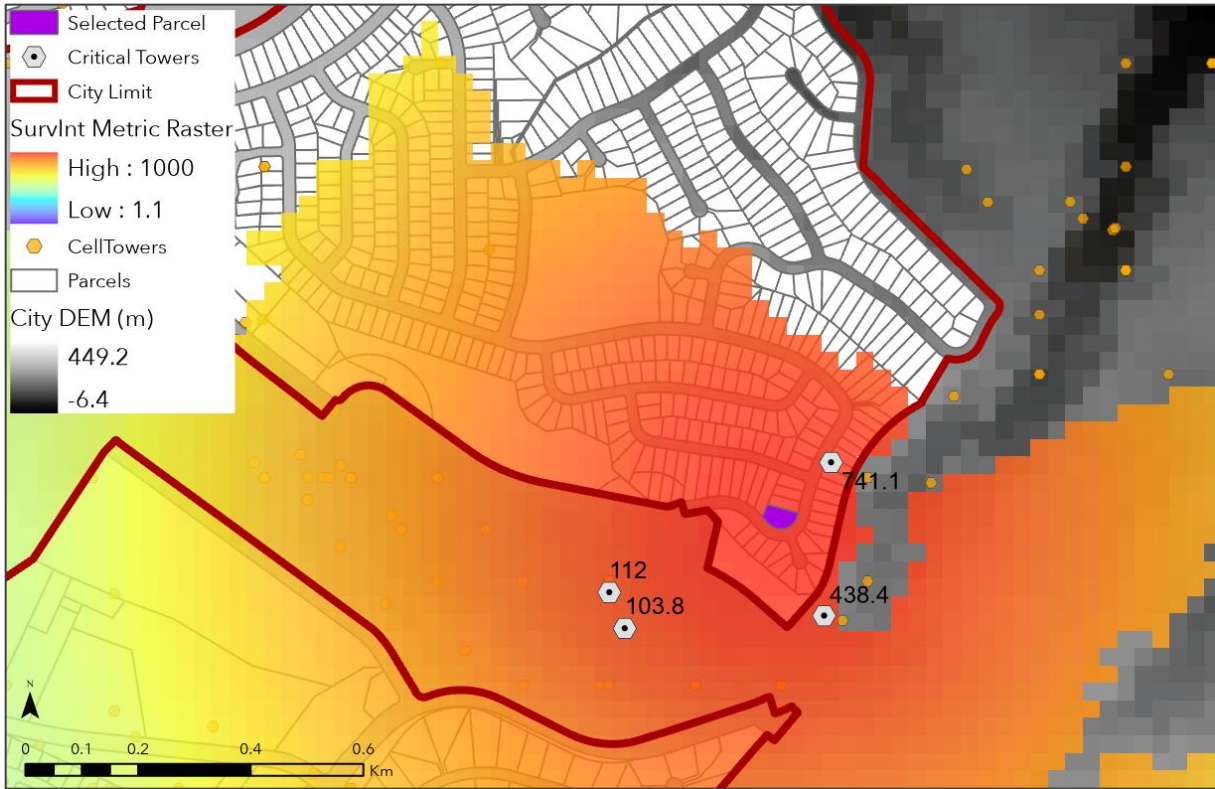


Figure 28 Visual Result of SurvInt UC2 –Critical Towers Outside the City Boundary

Figure 28 also illustrates how densely distributed cell towers are and how spatially widespread the potential for interference is. Unfortunately, these two spatial features work together to limit the utility of Real Time Kinematic (RTK) augmentation as a mitigation for cell signal interference. RTK is a form of differential correction, and Bolstad (2016) explains that “differential correction requires the base station and roving receivers to collect data from ... satellites.” The RTK concept is reproduced in Figure 29. Since both the base station and the

roving receiver must receive GNSS signals, the pervasive interference shown in Figure 28 means that the roving RTK receiver, regardless of position, will likely be subject to interference either from an identified critical tower for the selected tower or from another of the numerous nearby towers.

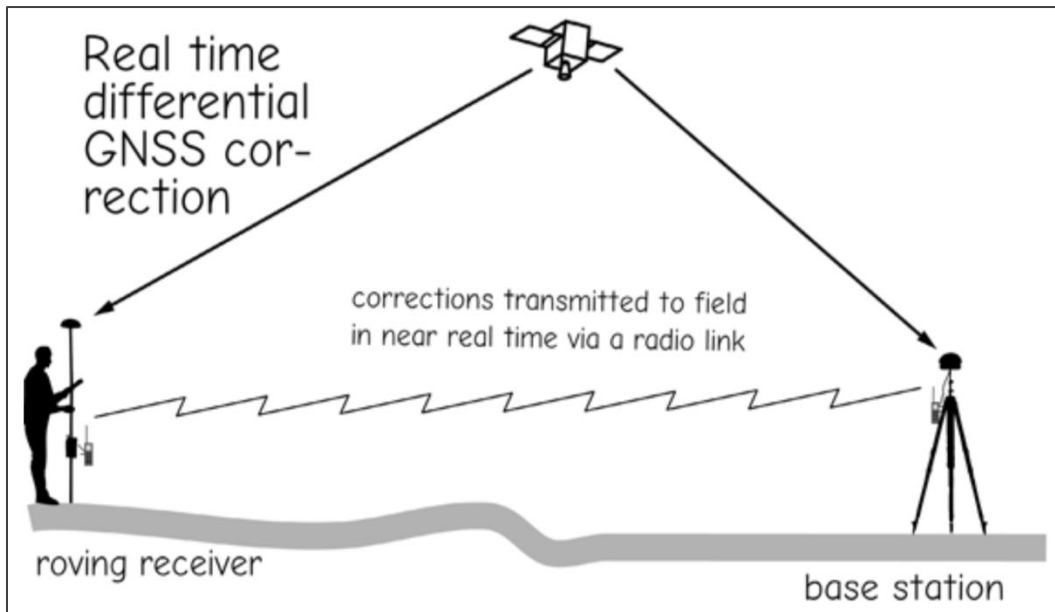


Figure 29 Both Base Station and RTK Roving Receiver Are Subject to Interference

The SurvInt model “UC2 Process Survey Parcels” includes the four geoprocessing steps detailed in section 3.3.2. The ModelBuilder implementation of the UC2 steps shown in Figure 12 is reproduced in the left panel of Figure 30. The results of UC2 are updates to both the city map and the city GDB created in UC1. The updated map includes one or more SurvInt interference metric rasters and a point feature class representing the critical interference towers for each parcel. The column of thumbnail maps in Figure 30 represents the step-by-step process of creating the visibility raster (visR) and the Euclidean distance raster (distR) that are used to compute the interference metric raster for each parcel. The visual output of running the SurvInt UC2 tool in ArcGIS Pro is shown on the right side of the figure

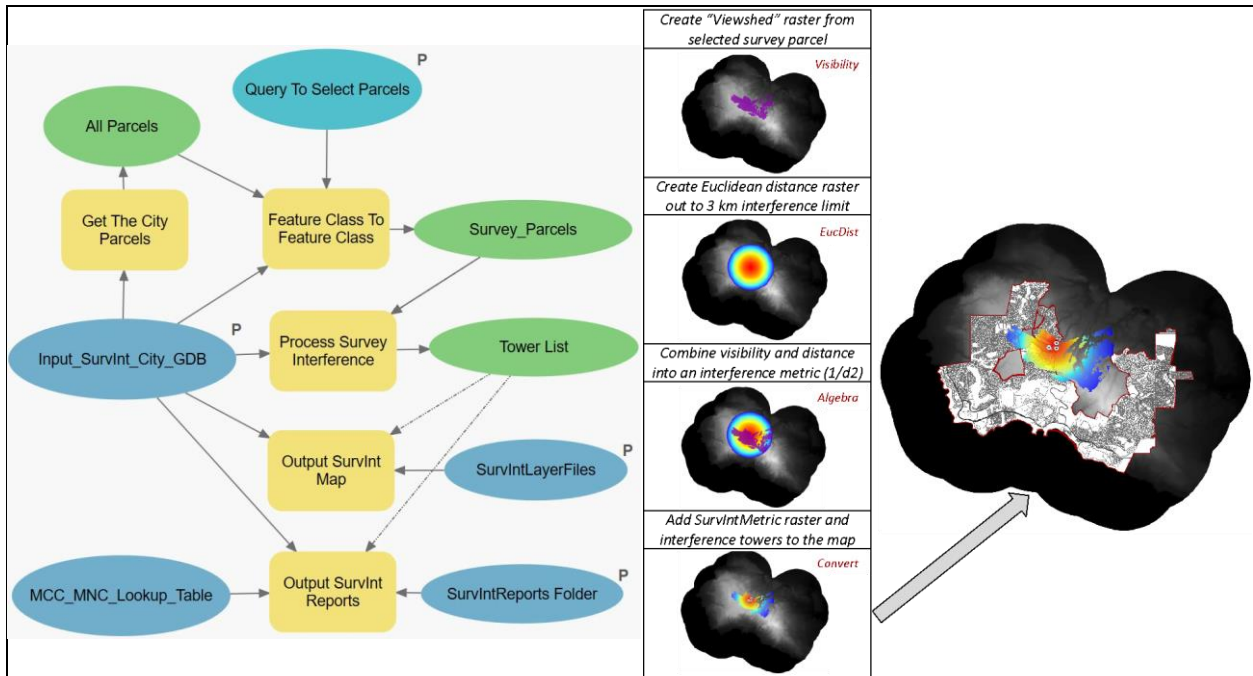


Figure 30. Example Output of SurVInt UC2 - Assess Interference and Notify Providers

As in the case for UC1, the UC2 data updates to the city geodatabase are just as important as the visual updates. Each parcel processed by UC2 generates a SurVIntMetric raster element and a point feature class representing critical interference towers for that parcel. The added data elements in the SurVInt\_Rancho\_Palos\_Verdes.gdb are highlighted in Figure 31. Figure 31 shows that, in addition to the raster and point feature class that are named with the parcel AIN and added to the output city map, the rasters distR and visR used in the interference metric calculations described in section 3.3.2.2.1 are also stored in the GDB. It is important to note that distR and visR are temporary, intermediate data elements. These rasters represent the results for the last parcel processed in UC2, and are overwritten on subsequent runs of the tool.

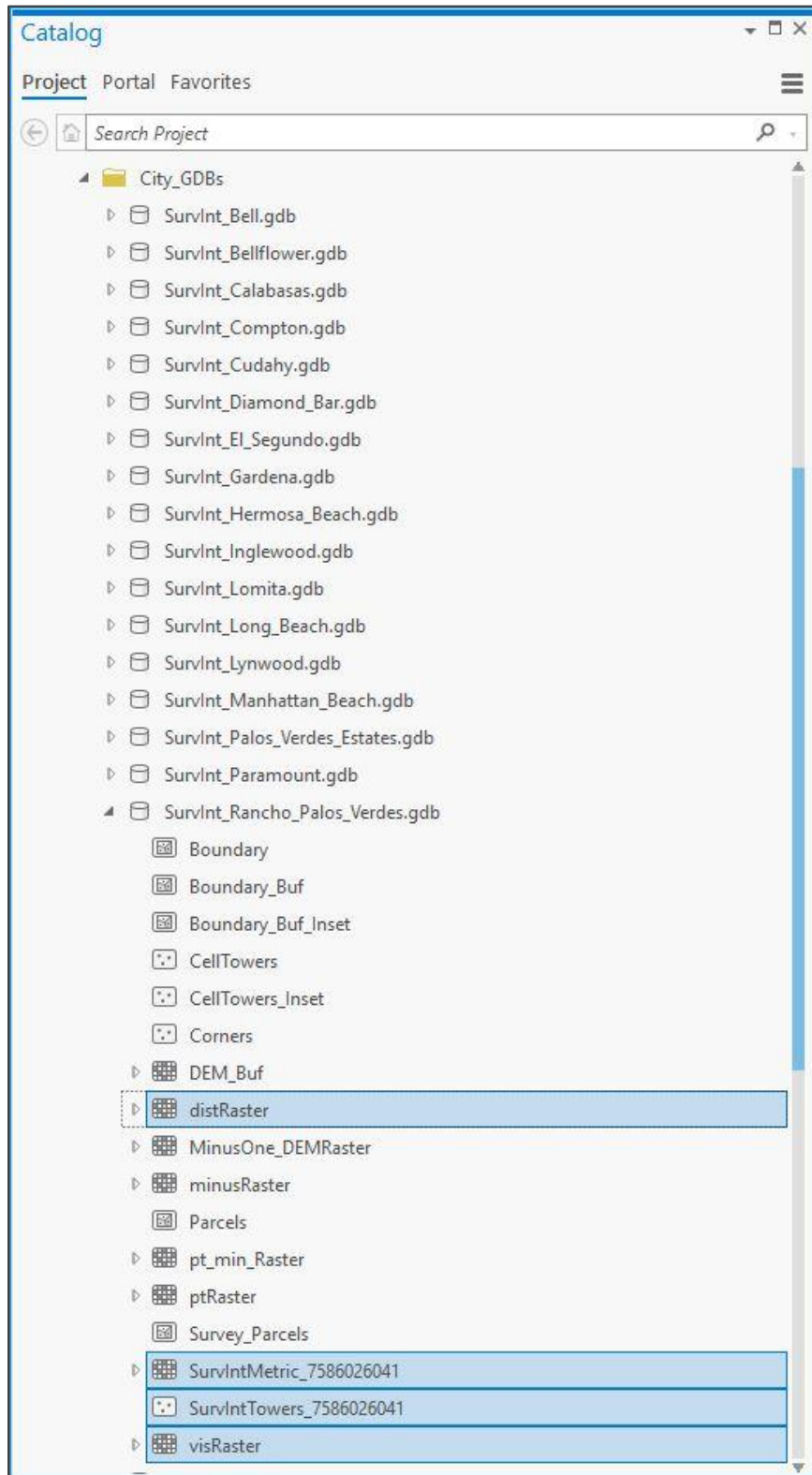


Figure 31 Out Data Result of SurvInt UC2 – RPV GDB With SurvIntMetric Raster

SurvInt also produces non-spatial output in the form of automatically generated notification letters. The tool uses standard Python text file output functions to combine the spatial and interference data computed in UC2 with a form letter that can be sent to the network operator of each interfering tower. The left side of Figure 32 shows the final map output for a combined survey task covering multiple parcels. The right panel in Figure 32 shows a sample notification report that was automatically generated for the single critical towers identified for the parcel on the far eastern edge of the city. Its normalized interference metric is 255.6, indicating a potential for a very severe impact on the parcel survey. An individual letter like the example shown in Figure 32 is generated for each of the interfering towers found during the execution of UC2.

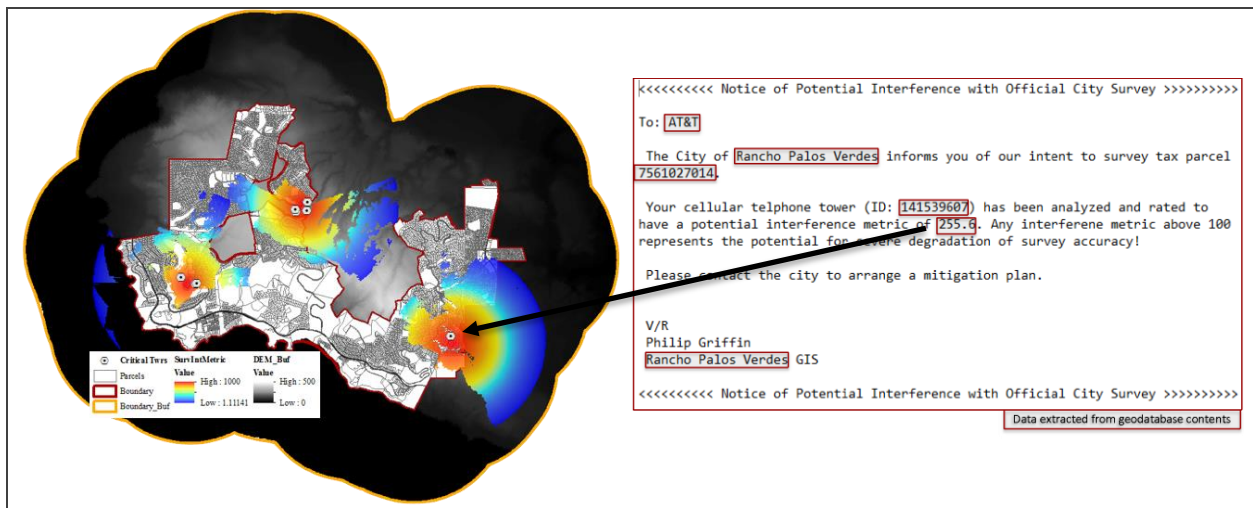


Figure 32 A Network Notification Letter Is Generated for Each Associated Critical Tower

#### 4.3. Effects Of Terrain On Survey Interference

The Friis (1946) model for the propagation of RF signals used in SurvInt includes the constraint that the signals travel in straight, unobstructed lines. This physics fact drove the design innovation to include the effects of obstructions in the SurvInt tool using the Esri Spatial Analyst Visibility function. In SurvInt, the source of RF propagation obstructions is variation in terrain

elevation, which was retrieved as a DEM from the USGS (2018) with a resolution of 10m. Since Los Angeles County includes areas with significant elevation changes, extensive testing of the SurvInt UC2 tool revealed many examples of the effects of terrain on potential cell signal interference.

Figure 26 provides an anecdotal, visual insight into the distribution of cell towers in three cities in LA County. These cities were selected because they vary considerably in the density of cell tower locations. Rancho Palos Verdes has the lowest density, and Signal Hill has the highest. Table 3 quantifies this observation. The table lists the min, max, and standard deviation of terrain elevation within each buffered city boundary. Table 3 also lists the area, the count of cell tower locations within the buffered city boundary, and the resulting cell tower density. The fourth and seventh columns in the table suggest that lower standard deviations in elevation (flatter terrain) correspond to higher cell tower densities and vice-versa.

Table 3 Cell Tower Density and Statistics of Terrain Elevation In Los Angeles County

| City                | Min  | Max   | St. Dev | Area  | Tower Count | Tower Density |
|---------------------|------|-------|---------|-------|-------------|---------------|
| Compton             | 0.2  | 63.3  | 9.6     | 111.6 | 6496.0      | 58.2          |
| Signal Hill         | -0.1 | 110.0 | 11.6    | 53.2  | 3428.0      | 64.4          |
| Rancho Palos Verdes | -6.4 | 449.2 | 112.7   | 147.6 | 3231.0      | 21.9          |

Figure 33 is the result of a simple linear regression that demonstrates a strong ( $R^2 = 0.977$ ) inverse correlation between the tower density and the variation of terrain elevations in each city. Although a sample of three cities is too small to produce a reliable conclusion, the same visual correlation between terrain and tower density was observed in many of the cities with geodatabases listed in Figure 24.



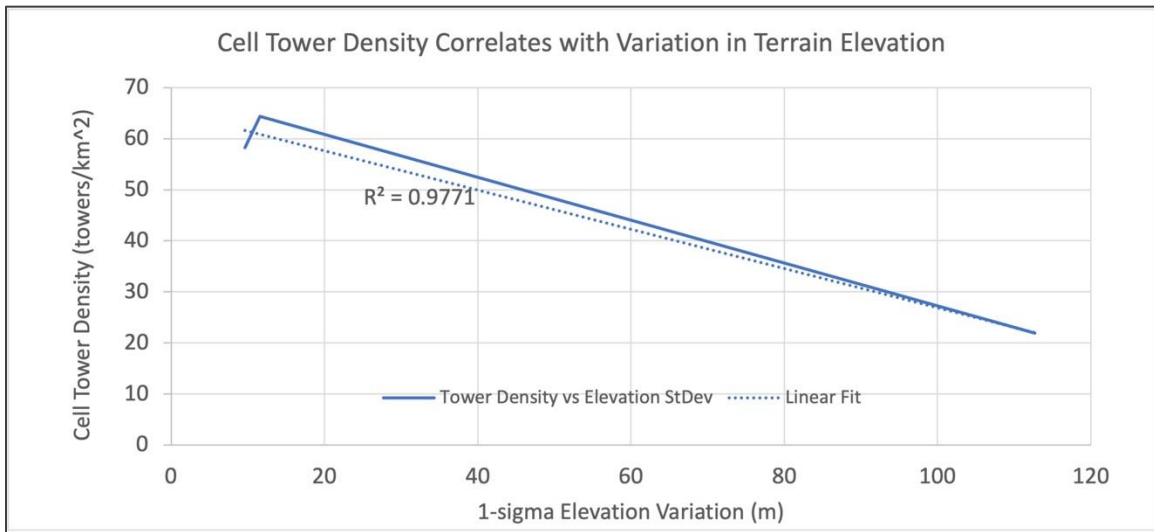


Figure 33 Cell Tower Density Correlates Inversely With Variation in Terrain Elevation

SurvInt UC2 provides another way to visualize the impact of terrain elevation on the potential for cell signal interference with land surveys. The design of this use case allows a surveyor to calculate and map interference rasters for multiple parcels at the same time. This design supports a visual, “side-by-side” assessment of how terrain elevation, as represented by the buffered DEM, affects the shape of multiple SurvInt metric rasters. Two tax parcels, selected in each of the three cities mapped in Figure 25, illustrate the range of terrain impacts.

The terrain of the city of Compton is typical of the Los Angeles County basin, where elevations rarely vary by more than 50m. The flat black in the upper panel of Figure 34 indicates that there is no visible terrain variation in Compton when visualized on a stretched color ramp that ranges from 0 to 400m in elevation. As a result, the Visibility function had no impact on the SurvInt interference metric rasters in this city. To illustrate this, UC2 was run on two test parcels located in the northeast and western edges of the city of Compton. The test parcels are highlighted in purple in the upper panel.

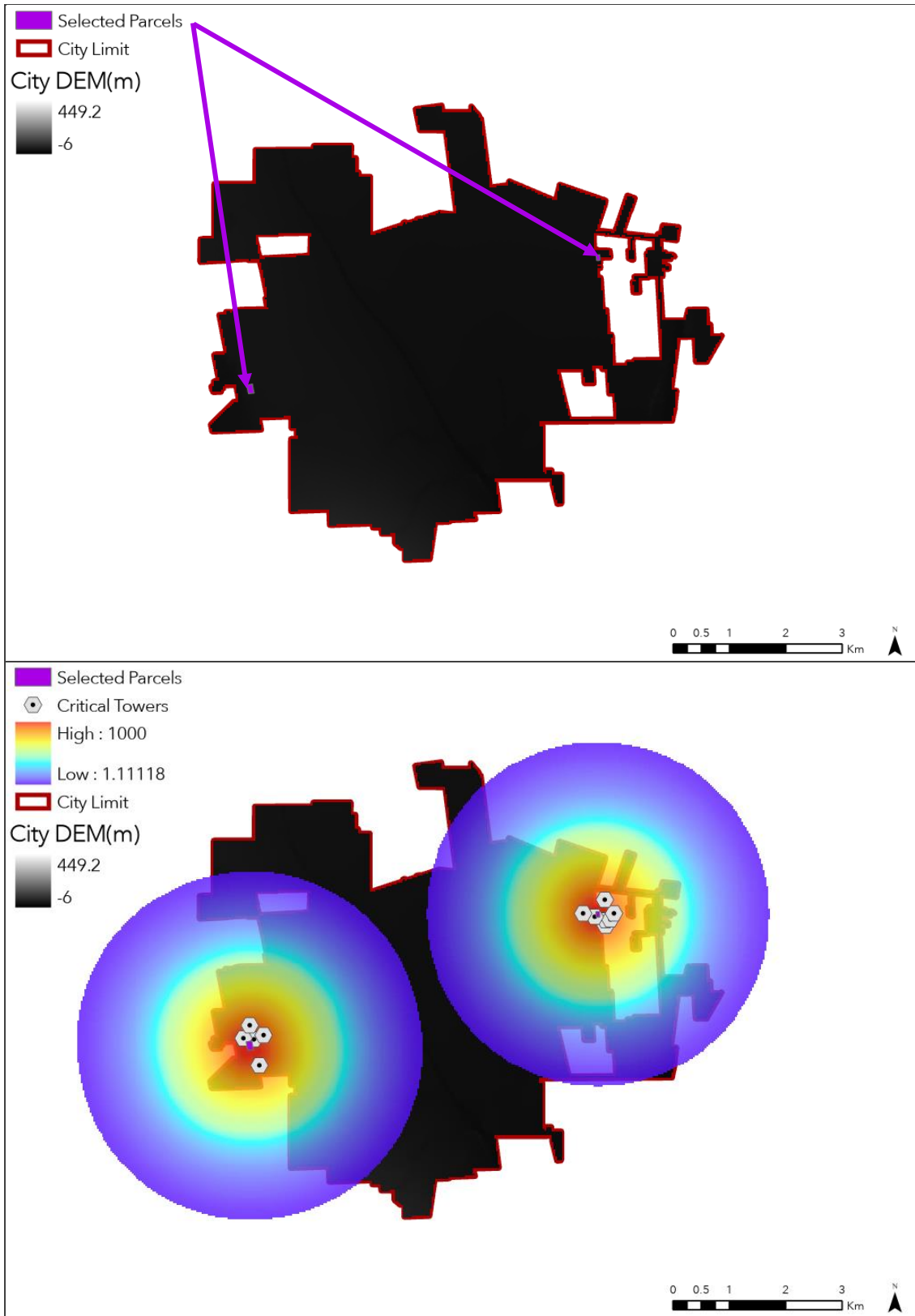


Figure 34 Two SurvInt Metric Rasters in the Flat Terrain of Compton, CA

The lower panel of Figure 34 shows the resulting SurvInt metric rasters and the critical interference towers for these two parcels. Two aspects of these data are worth noting. First, the shapes of the rasters are nearly circular, which indicates that Euclidean distance, not visibility, was the dominant raster parameter in this test case. In terms of the processes detailed in Figure 14, the viewshed raster, rVis, created no masking effect on the distance raster rDist.

By comparison, the city of Rancho Palos Verdes has highly variable terrain. As a result, the effects of creating a viewshed had a significant impact on the interference metric rasters. Figure 35 follows the same visual pattern as Figure 34. The large terrain elevation variations in RPV are apparent in the upper panel of this figure, which used the same stretched color ramp from -6 to 449m. Once again, the two test parcels used in the run of UC2 in RPV were selected near the northern and eastern limits of the city and were highlighted in purple.

The interference rasters for the two RPV test parcels appear in the lower panel of Figure 35. These rasters are very different compared to the circular rasters produced in the flat terrain in Compton, as shown in Figure 34. The northern raster (also used in Figure 28) was significantly masked by the viewshed and is almost wedge-shaped instead of circular. The rising terrain on the southern edge of this test parcel blocked the line of sight from the critical towers and prevented those towers from creating interference with any parcels there. The second raster is almost circular for 120 degrees on the east side of the city. However, it is significantly truncated on its west side. The terrain in this area slopes strongly to the east allowing for signal interference in that direction yet not in the opposite direction.

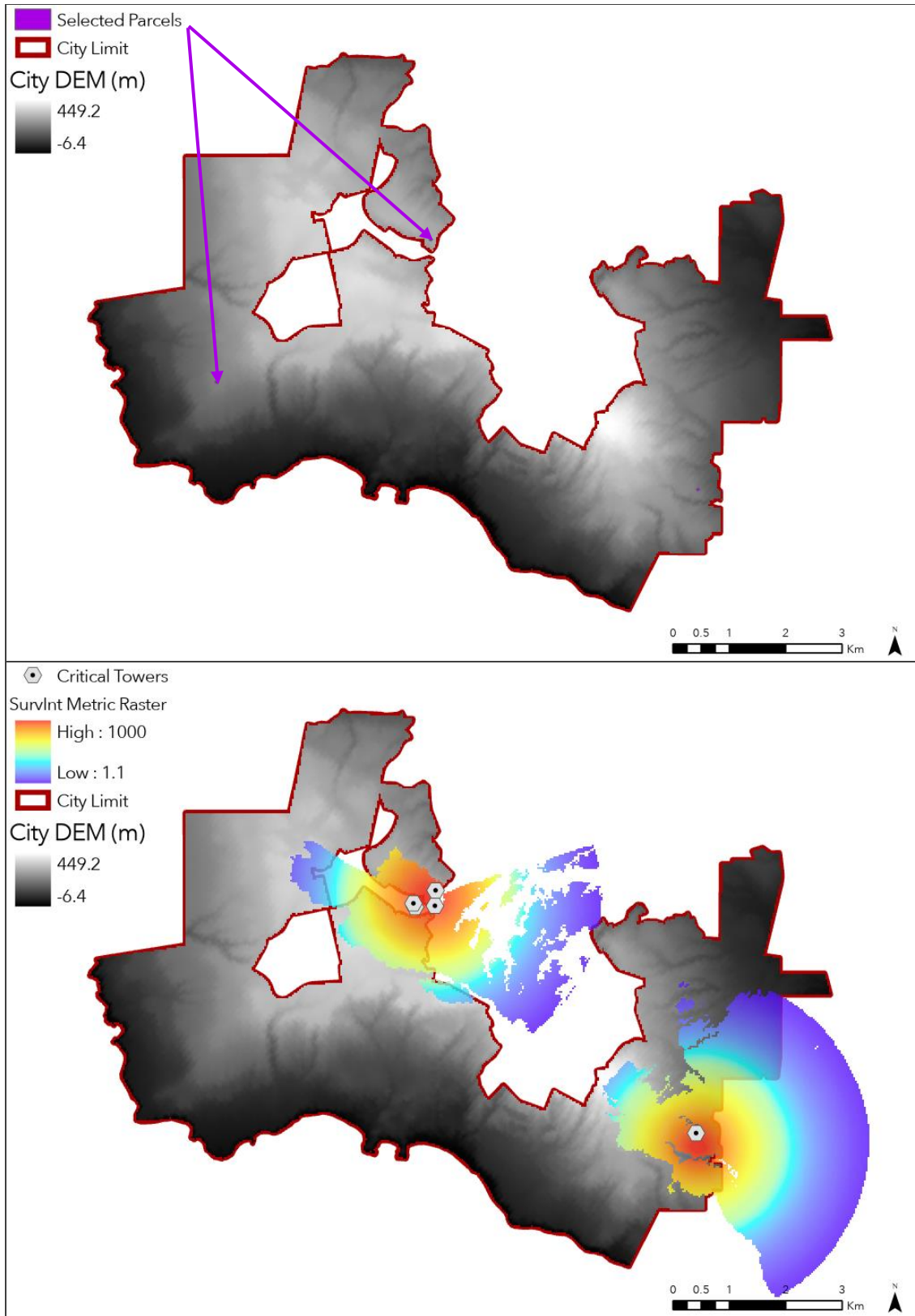


Figure 35 Two SurvInt Metric Rasters in the Hilly Terrain of Rancho Palos Verdes, CA

An interesting intermediate example came from testing SurvInt in the city of Signal Hill. According to the City of Signal Hill (2021), the city is named after a hill in the city center that has been used as an elevated signaling platform for hundreds of years. The elevation of the hill is only about 45m, but it is visible (highlighted by the white ring) in the DEM map in the upper panel of Figure 36. The hill is southeast of the center of the city. In this case, two test parcels were chosen on opposite sides of the hill, with one on the north flank and the other south of the hill. As in Figure 34, the test parcels are highlighted in purple.

The SurvInt metric rasters that resulted from this test are shown in the lower panel of Figure 36. They are both fan-shaped. The portions of the rasters facing away from the hill appear semi-circular, while the portions facing the hill are significantly truncated by that obstruction. Because of the simple geometry in this test case, it is easy to understand that the potential interference from the two critical towers in the south raster does not “reach” the northern test parcel due to RF blockage by Signal Hill.

A comparison of the lower panels in Figure 34, Figure 35, and Figure 36 provided a final anecdotal observation about the relationship between terrain and the risk of survey interference from cell towers. The number of critical towers per parcel in these examples also appears, based simply on visual interpretation of the resulting maps, to be related to terrain variation. Table 4 shows the counts of the critical interference towers derived by SurvInt for each of the city test cases. The density of critical interference towers per parcel was much higher for the flat terrain in Compton than in the other two cities. However, this might simply reflect the general tower density correlation with elevation shown in Table 3. The statistical robustness of this observation could not be evaluated with only six parcels worth of data.

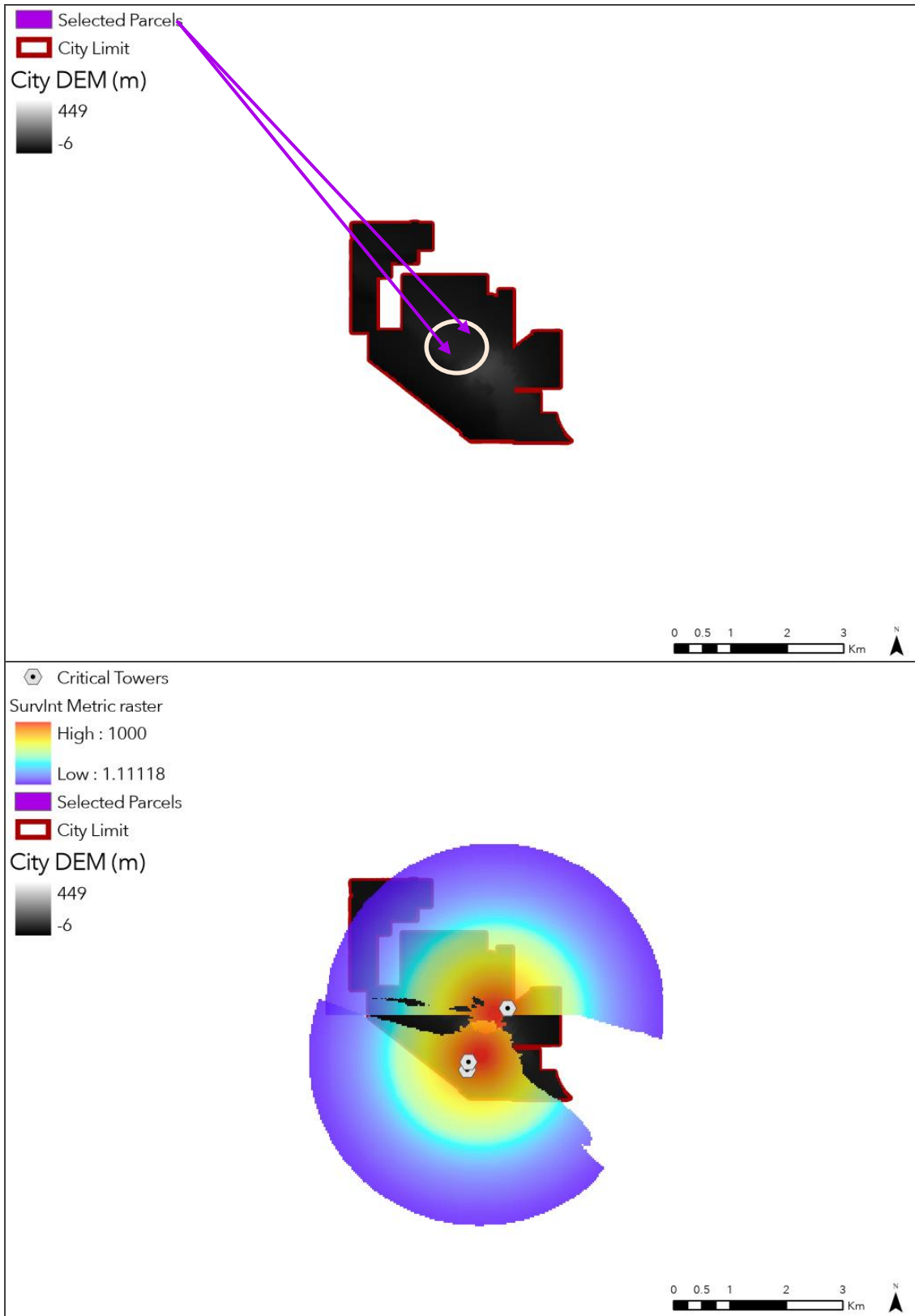


Figure 36 Two SurvInt Metric Rasters Separated by the Hill in Signal Hill, CA

Table 4 Critical Towers Per Parcel Increased in the Flat Terrain of Compton, CA

| City                | Parcel AIN | Critical Tower Count | Critical Towers/Parcel |
|---------------------|------------|----------------------|------------------------|
| Compton             | 6140023902 | 5                    | 6                      |
|                     | 6184007065 | 7                    |                        |
| Signal Hill         | 7215016020 | 2                    | 1.5                    |
|                     | 7214014113 | 1                    |                        |
| Rancho Palos Verdes | 7586026041 | 4                    | 2.5                    |
|                     | 7561027014 | 1                    |                        |

The results presented in Chapter 4 confirm that SurvInt met its primary design objective, which was to provide an automated, ArcGIS Pro-based tool set to visualize and quantify the impact of cellphone signal interference on land surveys in Los Angeles County. All of the necessary input data, geoprocessing, and output visualizations have been developed and demonstrated in many cities in the county. However, the development and testing processes revealed several improvements in functionality and accuracy that could be implemented in future versions of SurvInt. These are the subject of Chapter 5.

## **Chapter 5 Discussion and Conclusions**

As in any complex software development process, many implementation challenges had to be overcome, and some remain unresolved. More importantly, numerous opportunities for extending SurvInt's capabilities appeared during the development and testing of the project. Chapter 5 begins with descriptions of the SurvInt development challenges that have been overcome. For each challenge, a synopsis of the solution was provided along with the principal lesson learned, which will help guide future efforts. The second half of this chapter presents some of the foremost capability enhancements that are possible in future evolutions of SurvInt.

### **5.1. Development Challenges**

The development of an automated tool that includes geospatial processing, database management, and complex visualizations included a number of challenges. Each step in the implementation can be viewed as an individual challenge. There were three key hurdles that were overcome in the development of SurvInt, which stand out from the rest of the challenges in terms of difficulty and the potential to serve as lessons learned for this kind of project in the future. These challenges spanned the SurvInt effort from building models to calculating interference metric rasters to symbolizing the SurvInt map outputs. The first challenge was controlling the flow of ModelBuilder execution. Creating, aligning, and modifying the rasters used in the computation of interference metrics represented the second key challenge. The third and final challenge appeared late in the project. Adapting the models and scripts to significant changes in the latest update of ArcGIS Pro (2.8) was the third key development challenge. These ArcGIS Pro changes included an updated arcpy.mp python library and the migration to Python 3.0. Both required significant changes to the Python code initially developed for SurvInt.



Chapter 3 contains descriptions of these challenges and their resolutions in the development of SurvInt. A brief synopsis of each challenge and resolution is provided in the following subsections. The goal of these summaries is to provide additional helpful guidance and lessons learned for developers of similar GIS automation toolsets in the future.

#### *5.1.1. Challenge 1: Controlling ModelBuilder execution*

It is apparent in the ModelBuilder models for SurvInt UC1 and UC2 that the order in which the submodels are executed is crucial to obtaining correct results from the use case operations. In particular, each use case follows a design pattern in which data is generated and then added to the SurvInt map. In UC1 (see Figure 4), a SurvInt City GDB must be created before the polishing and map output submodels can execute. Likewise, in UC2 (see Figure 12), the SurvInt metric raster maps and the notification reports to tower operators can be created, the raster computations and tower identifications must be complete.

At certain times during this development, both use case models failed because of execution order issues. One method recommended by Allen (2011) to address the order of execution challenge is the use of preconditions. Submodels are designed to produce outputs that must exist before another process in the model will execute. Graphically these are represented in ModelBuilder diagrams as dashed arrows like those in Figure 4 and Figure 12.

A more subtle challenge in controlling execution occurs when a model uses an iterator. UC2 depends on this construct to process a list of survey parcels, as shown in Figure 13. In the early versions of SurvInt, the iteration to compute the rasters as a separate submodel from the iterator that identified the critical interference towers. Ensuring that all the rasters were calculated before the towers were processed was very challenging. The challenge was that each time the iterator ran, it generated the output used as a precondition. So, when the raster for the very first parcel

was complete, the tower processing would start. Since the tower processing was much quicker than the raster processing, it soon got ahead of the raster processing, and the operation failed. Since they both looped over the same list of parcels, these two iterators were eventually combined in a single sub-model as in Figure 13. Then the internal precondition (dashed line) on the interference raster controlled the processing of the towers.

The final instance of this challenge was the most difficult to address. While responding to the Python 3.0 changes in ArcGIS Pro 2.8, some of the script tool submodels of UC2 needed to be replaced. After they were replaced, their execution order changed. Surprisingly, Hibma (2018) identified that the submodels “will be executed in the order you added them to the model”. Odd behavior like this was one of the major reasons that as the development of SurvInt progressed, more and more functionality was implemented in Python scripts instead of ModelBuilder elements.

These three instances were not the only execution order issues encountered during the development, but they form the basis of the key lessons learned from this challenge area:

Lessons Learned:

1. Provide submodel outputs used as preconditions for later submodels
2. Combine iterator submodels that loop over the same inputs
3. Create model components in order of execution

*5.1.2. Challenge 2: Raster Automation*

As detailed in section 3.3.2.2.1, the most complex geospatial processing in SurvInt is the calculation of the interference metric rasters. The most challenging aspect of this calculation was to ensure that rasters derived from a point feature class (parcel corners) were aligned at the pixel level with the two calculated rasters, distR and visR, which represent the distance from a corner

and the visibility of that corner from a given cell tower respectively. Esri (2021d) provided the solution in this case which was to use the USGS DEM raster as the “snap” raster in ArcGIS Pro terminology. Esri (2021d) describes the snap raster as an ArcGIS Pro environment setting that controls the extent of rasters to ensure “when the output cell size is the same as the snap raster cell size, the cells in the output raster are aligned with the cells of the snap raster”. Without it, the parcel corner raster could not have been aligned with the DEM, which would have made the viewshed results of Visibility function meaningless.

The `arcpy.sa` library provides raster arithmetic expressions that, according to Esri (2013), allow the creation of “Raster Calculator-style expressions directly in Python”. The Raster Calculator is an ArcGIS Pro tool that treats rasters as arithmetic entities that can be summed, multiplied, scaled, etc. The most fundamental equation in this project is the `SurvInt` metric formula shown in equation (1) in Chapter 3. As written there, the equation looks like typical scalar arithmetic. `Arcpy.sa` allowed it to be implemented for rasters using almost the identical form as shown in the next to last line in the Python code in Figure 15.

Finally, as noted in section 3.3.1.4, a subtle error occurred in the early version of UC2 when a cell tower was located just outside the clipped `DEM_Buf` raster. To resolve this challenge, a specialized submodel was added to the UC1 model. The Polish City GDB function makes a one-pixel inset of the buffered DEM raster and then clips the cell tower feature class to remove a small number of cell towers at the very edge of the DEM. Although the fix was simple, this error required considerable effort to track down and understand. The lesson learned in this case was to be especially careful about any processing at the boundary of a clipped raster since unexpected results are possible. For example, simply increasing the buffer distance did not solve this

problem because the density of tower locations is very high, and moving the boundary just put other towers in the very narrow raster boundary zone.

Raster calculations in ArcGIS Pro are very powerful but require care to use correctly. The need to implement them in SurvInt led to three more lessons learned.

Lessons Learned:

1. Use a snap raster to align unrelated rasters at pixel resolution
2. Use raster arithmetic to implement formulas across multiple rasters
3. Apply extra scrutiny to calculations at the boundary of rasters

*5.1.3. Challenge 3: ArcPy Updates*

The last major development challenge was not the result of the design or implementation of SurvInt's geoprocessing models. Towards the end of the development phase, Esri released a major update to ArcGIS Pro that included two very impactful changes. The first was a complete rewrite of the `arcpy.mapping` library that was used to automate mapping in earlier versions of ArcGIS Pro. This not only changed functionality but shifted the fundamental unit of the mapping system from a data frame to a map. The second impactful change in ArcGIS Pro 2.8 was a shift from Python version 2 to version 3.

The most difficult challenge caused by the rewrite of the `arcpy.mapping` library, `arcpy.mp`, was that the ability to apply formatting from a layer (.lyrx) file to a feature layer in a map no longer works when called from within a Python script. The resolution of this challenge was more of a workaround than a true solution. As described in section 3.3.1.5, the workaround required a paradigm shift. Instead of creating a feature layer and applying symbology from a layer file, SurvInt now adds a symbolized layer first and then connects it to the correct feature data in the GDB.

The second impactful change in ArcGIS Pro 2.8, the shift from Python version 2 to version 3, had less impact on SurvInt than the change to `arcpy.mp`. However, it did expose some Python code in the generation of the critical interference tower notifications. The original code used old file handling mechanisms that no longer worked with Python 3.0. This challenge was met by rewriting the code to conform with Python 3.0. This rewrite could have been avoided if a more standard approach had been taken in the earlier version of the “Generate SurvInt Reports” submodel.

The adoption of `arcpy.mp` presented a meta-challenge in terms of how Esri reviews the purpose of automation in ArcGIS Pro. In earlier versions of ArcGIS Pro, Esri seemed to be moving in the direction of allowing users to automate any and all functionality using Python and ModelBuilder. The response from Esri (2021i) to the “ApplySymbologyFromLayer” bug report indicates a change in their direction. Esri (2021i) suggests that the user create a new parameter output from the code and then edit the applicable layer field using ModelBuilder. Although this approach is valuable and does appear to work around the issue, it requires manual intervention by the author of the automation. This is emphasized in the Esri (2021c) documentation for the new `arcpy.mp` library, which states, “*arcpy.mp module was designed so it can be used to modify existing elements ... for the purpose of map or data automation. It can't be used to create ... maps, layouts, and legend elements*”. The lack of support for creating output maps impacted both of the SurvInt use cases and required considerable rework of the Python code at their map output stages.

The three key lessons learned with respect to using Python automation in ArcGIS Pro all relate to major changes in the ArcPy library provided by Esri.

### Lessons Learned:

1. Modify data connections instead of applying symbology from layer files
2. Write version independent Python code
3. Use automation for repetitive tasks, not for the creation of geospatial visualizations

## **5.2. Future Enhancements**

Chapter 2 related this project to existing literature and geospatial processing methods organized into three distinct categories of influence on the development of SurvInt. Each category played an important role in shaping SurvInt's design and capabilities. The three categories each play a similar role in organizing ideas for future enhancements of the tool set. Over the course of SurvInt development, many opportunities for improvement in the model presented themselves. First are enhancements in SurvInt's physical modeling of cell towers and RF signals. Testing in the real world and validation of the tool with ground truth data are the second opportunity for enhancement. Finally, improvements in the implementation, from the user interface to data capacity, make up the third category of potential enhancements. The subsequent sections describe two possible enhancements in each of the three categories of related works described in Chapter 2.

### *5.2.1. Enhancement 1: Physical Models*

The physics model of cell towers used in SurvInt is based on the Friis (1946) propagation model. This means that the power of an interfering signal RF signal at the corner of a tax parcel is determined by the distance of that point from an interfering cell tower. The closer the parcel is to the tower, the more RF signal power it receives. If distance were the only parameter, then the interference rasters would look circular like the ones in Compton, CA, shown in Figure 34. A key innovation in SurvInt is the inclusion of a terrain visibility mask which limits the potential

for interference by a cell tower. This element of the physics model significantly reduces the area of potential interference, as shown in Rancho Palos Verdes in Figure 35.

The elevation antenna pattern is a physical characteristic of all real-world antennas. This pattern focuses the transmitted energy in specific directions so that the power on the ground is not solely a function of the distance. The top panel in Figure 37 illustrates an antenna elevation pattern used to model cell signal power by the Australian Government (ARPANSA 2017). The graph in the lower panel shows that, unlike in the current SurvInt model, the highest power level is NOT closest to the tower. Including the elevation antenna pattern reduces the likelihood of interference near cell towers and has the potential to significantly improve the fidelity of the SurvInt results.

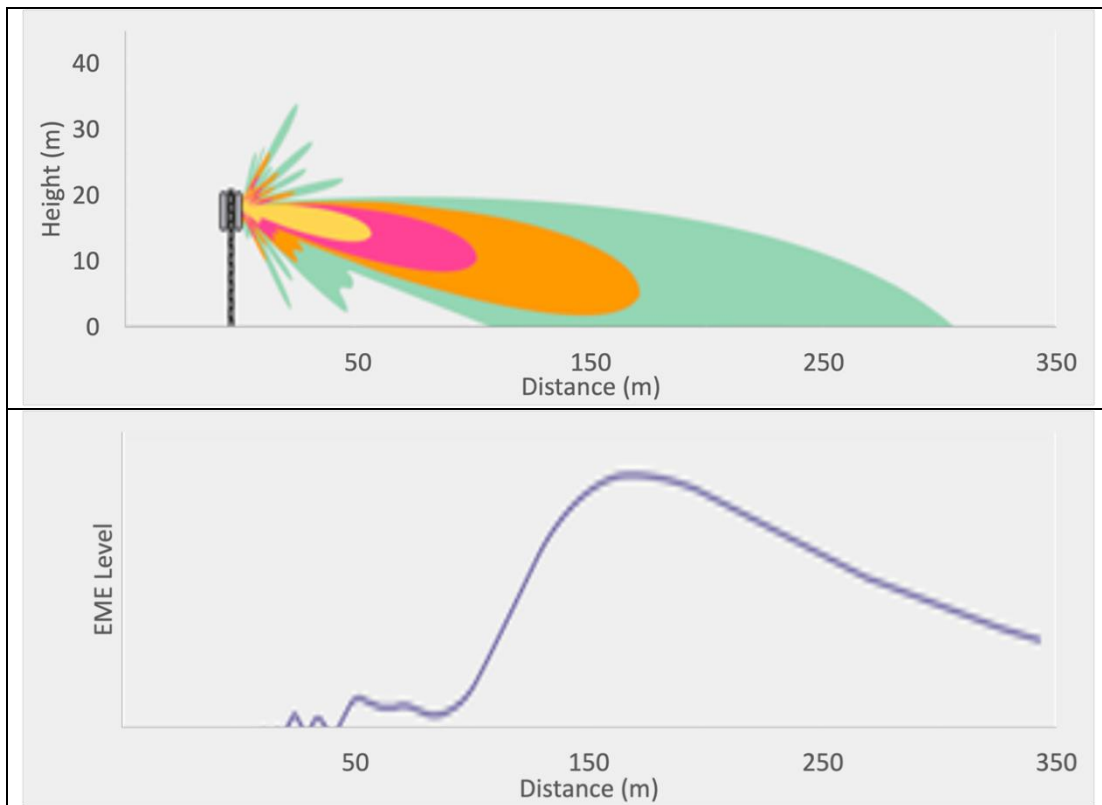


Figure 37 Elevation Beam Pattern Reduces RF Signal Power Near a Cell Tower

All GPS-enabled receivers include an antenna to receive the satellite signals. So, a further enhancement of the SurvInt physics model would be to include the shape of the receiver antenna pattern, which would have a shape similar to the one illustrated in Figure 37. This enhancement will be a more challenging than others since there are many types and models of GPS receivers that each have a unique antenna design. For example, the USDOT (2018) study “tested 80 GPS/GNSS receivers ... along with the associated antennas”. The tested receivers exhibited a wide range of susceptibility to interference. In a future version of SurvInt, the effects of receiver antennas could be included as a “susceptibility factor” attribute included in the Cell Tower feature class that could be used as an overall multiplier in the interference metric raster calculations.

Another straightforward enhancement of the physics model would be to include the transmit power of each cell tower in the cell tower feature class. If the power data were added to SurvInt, then the normalization of the interference metric could be modified by the cell tower power. Towers with more power would increase the standard interference range of 3km, while lower power towers would reduce the range of potential interference. Including transmit power as an additional variable would require a simple change in the Process Survey Interference Python code. The challenge will be finding the data in a publicly available form. For example, Unwired Labs (2021) provides the OpenCellID cell tower data described in Table 1. This data does include a field called *Power*, but all of the values are zero. For competitive reasons, the network operators may consider this data to be proprietary and keep it private, which would make the implementation of this enhancement very difficult.

Currently, SurvInt’s physics model includes no temporal variability in the potential for interference. However, environmental factors (external to the towers and receivers modeled in



SurvInt) often vary as a function of the time of day. Two examples of external factors are variation in cell phone tower usage and atmospheric effects on RF propagation. Both of these factors do vary as a function of time of day. Fortunately, they are likely to have a limited impact on the results produced by SurvInt. First, as shown in Figure 38, Wang et al. (2015) determined that signal tower usage does vary considerably over the course of a day. The more towers are used, the higher the chance that they may interfere with nearby land survey receivers. However, the figure shows that during “business hours” (8am to 5pm), when land surveyors are likely to work, the usage is relatively constant and high. So, adding this temporal variation would have a relatively small impact on the utility of SurvInt.

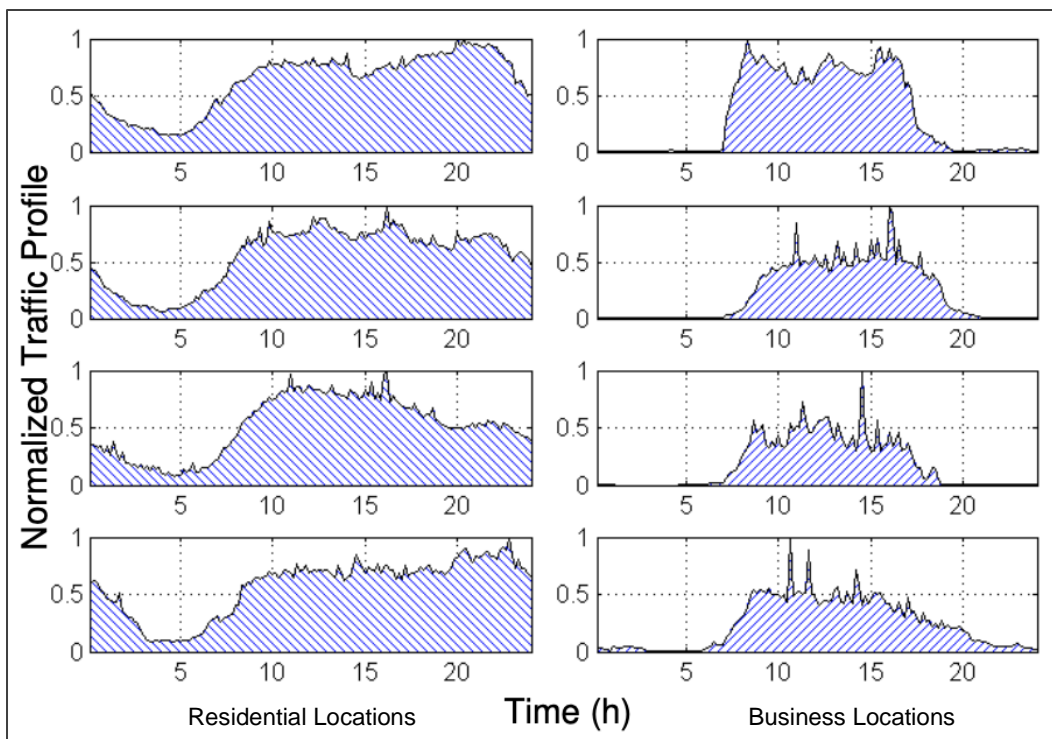


Figure 38 Cellular Traffic Is Nearly Constant During Business Hours (Wang et al. 2015).

An example of an atmospheric effect on RF signals that could be added to SurvInt is the attenuation of signal power. Attenuation reduces the potential for interference. However, according to Milione (2011), at “frequencies below 1 GHz, the effect of atmospheric attenuation

is negligible” Figure 39 quantifies the negligible impact of attenuation and shows that 99.9% of the signals in the GPS/LTE frequency range “get through the atmosphere” without issue.

Neither of these external factors seems likely to be as important to enhancing the SurvInt model as including antenna patterns and transmit power.

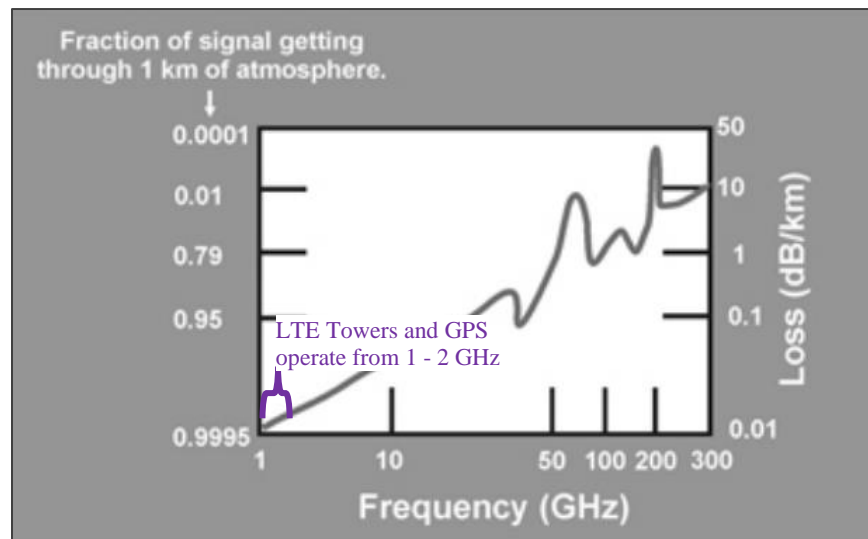


Figure 39 Atmospheric Attenuation is Negligible for Cell Network Signals (Milione 2011).

### 5.2.2. Enhancement 2: Real-World Data and Validation

Three enhancements seem likely to be mandatory if SurvInt is going to be a tool officially used by municipal agencies. First, the usability of SurvInt to surveyors in the field must be verified. Second, the adoption of SurvInt by land surveyors will require the accuracy of its interference estimates to be validated in the field with data collected by real receivers. Finally, its applicability to urban settings like Los Angeles County must be established. Potential approaches for each of these three validation efforts are described in this section. Each of these enhancements will require significant effort and involve other people and organizations. So, the benefits of these validation enhancements must be carefully weighed against investment costs and the likelihood of whether SurvInt will transition from an academic project to commercial software.

Interaction with and feedback from a targeted user community is a well-established process in successful software development. In fact, two of the five steps for evaluating commercial viability given in Ceniza-Levine (2018) are aimed at user interaction. Ceniza-Levine recommends talking to people currently in the business and attending related professional events. In the case of SurvInt, this would mean engaging with working surveyors and attending conferences like the upcoming 2022 Joint Surveying Conference of the California Land Surveyors Association (CLSA 2021). Although engaging with users in these activities is certain to improve the usability and viability of a tool like SurvInt, it was not possible to do so during the initial development phase. It seems likely that user feedback will be even more helpful once some of the other enhancements described in this section are implemented. Although field measurements will be required eventually, one method of analytically validating the accuracy of the SurvInt interference metric is to use the measured, receiver-specific data in the USDOT (2018) adjacent band interference study, which tested Government and industry receiver equipment across a diverse range of applications, including “surveying, precision agriculture, weather forecasting, earthquake monitoring, and emergency response.” They tested a total of 80 receivers and determined a power level that caused severe performance degradation in each of them. Figure 2, which was used in this development to determine the maximum interference distance, is the result of testing with only one receiver. SurvInt’s metric calculation could be enhanced to use all the data in this report to determine a range of possible interference values at a tax parcel corner point instead of a single value. This enhancement would provide something like a confidence interval for interference. If the majority of the measured receiver curves (like those in Figure 2) generated low interference metrics in SurvInt then a surveyor could be much more confident that his equipment would work for that survey.

In urban environments, man-made structures are sources of RF signal path obstruction that are not included in SurvInt. Adding these structures into the terrain and visibility modeling in SurvInt would be an essential enhancement for municipal users. A model of the effects of buildings on potential interference with GPS-based receivers was included in the emergency response portion of the USDOT (2018) study. Similarly, as explained in Alves et al. (2018), because Novatel’s Interference Tool Kit uses actual measurements of RF signal levels, building obstructions are automatically included in its results.

Since field measurements are probably in SurvInt’s distant future, an analytical enhancement would base RF obstructions on Lidar-derived Digital Surface Models or (DSMs) instead of a DEM. The USGS (2018) provides extensive, free Lidar data coverage of most urban areas in the U.S., including southern California, via the National Map project. Esri (2021a) documents the ArcGIS Pro capability to construct Lidar DSMs using built-in geoprocessing tools, which could be included in a future enhanced version of the SurvInt automation. The resulting DSM would then replace the DEM raster in the visibility calculations of the interference metric to capture the effects of buildings and vegetation on potential cell signal interference with sensitive receivers.

### *5.2.3. Enhancement 3: GIS Automation*

As shown in Figure 21 and Figure 26, the user interfaces for running the SurvInt Use Case tools are quite simple. After the locations of the city GDBs, layer, and reports files have been established, a user does not need to change them. Once these entries are set, two entries in the tool dialog boxes control the Use Cases. For UC1, the user must choose a city in Los Angeles County. UC2 then requires the selection of a set of parcels (using their AIN) within the city chosen in UC1. Each of these dialog box entries were implemented in the form of an SQL query. Although the queries are very simple and use either an equality or “LIKE” operator, a graphical

means of selection would be a useful enhancement to SurvInt. The section on *Interactive Feature Selection* in Allen (2011) inspired this idea for a future enhancement.

In UC1, a single city would be selected on a map of Los Angeles County. For UC2, the tool should allow the selection of multiple parcels for interference analysis. The Use Case tools would then operate on the selection set. Figure 40 illustrates how these graphical selections would appear in ArcGIS Pro. The top panel shows the city of Compton selected as the input UC1. The lower panel of the figure illustrates the selection of the two parcels used in the raster illustration in Figure 34. Feature labeling at appropriate scales would be needed to select the survey parcels with an AIN. This labeling is included in the .lyrx file used to symbolize the Parcels feature class in SurvInt.

A second potential enhancement of SurvInt would be much more challenging to implement but more beneficial to the utility of the tool in the real world. Most of the data layers used in SurvInt are relatively constant. These layers include the city and county boundaries and the USGS DEM. On the other hand, both the numbers and locations of cell towers in the county are highly dynamic. Odendahl (2019) finds that since 2015, Los Angeles County's Department of Regional Planning has reported a 300% increase in the number of applications for new cell towers. This high rate of change supports the idea of enhancing SurvInt using dynamic updates of the cell tower feature class used in UC2. Esri (2021j) describes ArcGIS Pro support for serving both dynamic and cached maps, which can be used for this purpose. However, this enhancement would require major changes to the current ArcGIS Pro desktop implementation of SurvInt. Alternatively, the cell tower feature class found in Unwired Labs (2021) could be created periodically from publicly available data tables from *OpenCellID*. So, the benefit of implementing a map service to keep one layer up to date may not be worth the effort involved.

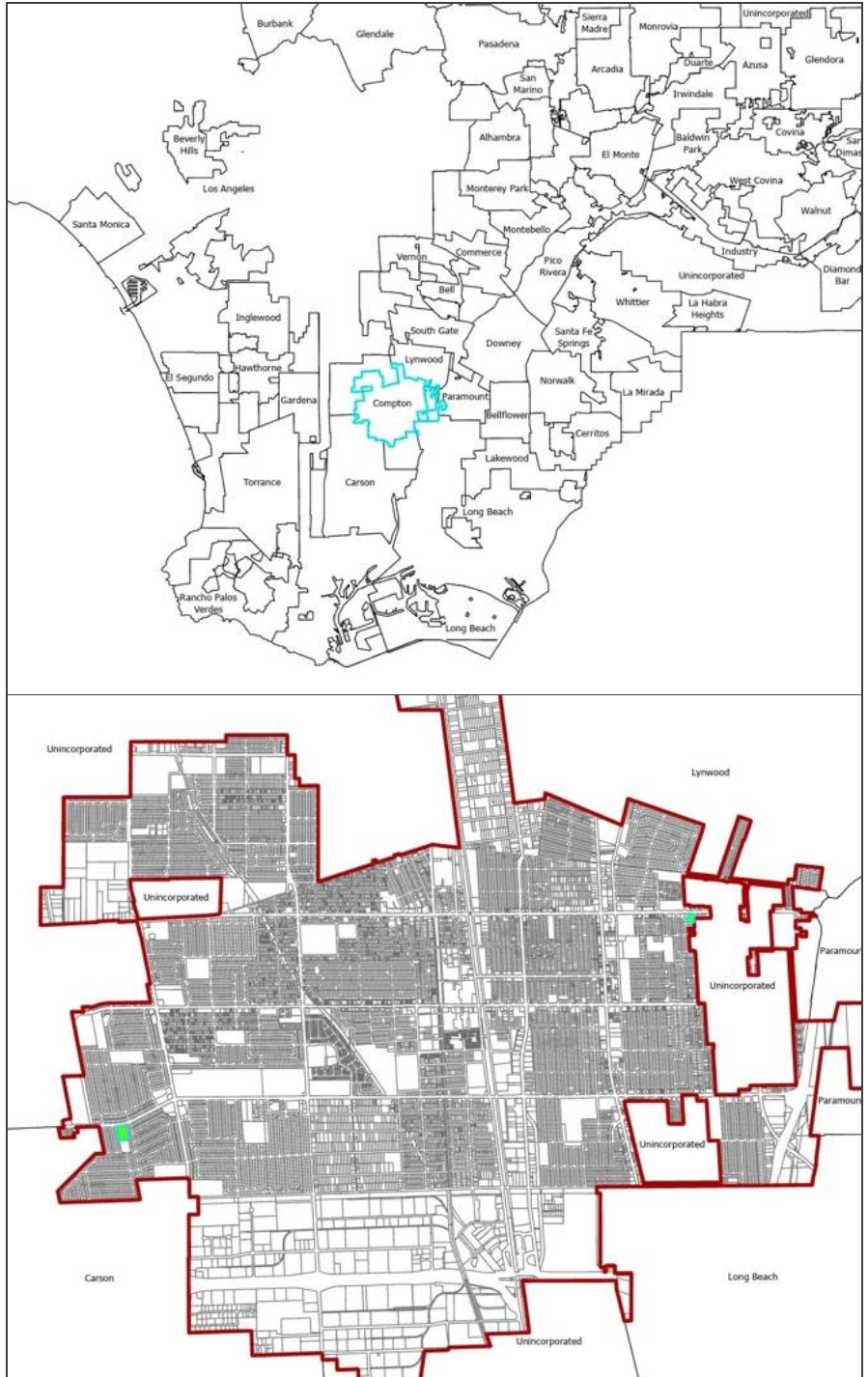


Figure 40 Graphical Selection Could Replace SQL Queries in SurvInt Input Dialogs

In the current implementation, SurvInt is an ArcGIS Pro project that must be accompanied by a large geodatabase for Los Angeles County, along with a small set of Python files, layer files, and two geoprocessing toolboxes. Two challenges significantly limit the ease with which the

project can be distributed and used by others. The first is the size of the data, which exceeds 1.3 GBytes. Second, are the numerous data connections that relate the feature layers to the computed data and the geoprocessing tools with the ModelBuilder models and Python scripts. Esri provides the “project package” (.ppkx) mechanism for sharing a complex project like SurvInt. A project package ensures that other users have all the data and connections they need to use the project. Creating a project package for SurvInt would be a simple enhancement that would improve the portability of the SurvInt tool and thus could increase the user audience significantly.

At the beginning of the project, GitHub (and other similar project-sharing websites) was considered as a platform for sharing the SurvInt tool. GitHub provides excellent functionality for collaborating on the development of *code-based* projects. Unfortunately, SurvInt requires a very large geodatabase in its distribution, and GitHub (2021) recommends “repositories remain small, ideally less than 1 GB”. Although the tool itself is not suitable for collaboration using GitHub, the Python code for all the script tools used in SurvInt is available in the author’s GitHub SurvInt repository (<https://github.com/phgriffi/SurvInt>). Sharing the Python code will allow future GIS automation developers to access (and collaborate on) the latest code used in SurvInt.

### **5.3. Conclusion**

The SurvInt project demonstrated the far-reaching potential of automation in complex GIS applications. The final form of the project met its design objectives and use cases. Anderson (2020) suggests that an essential principle of map design is to understand the intended audience and how they will use the information provided by your spatial products. The development of this project focused on a clearly defined audience of users that include municipal GIS administrators and city or county land surveyors. Helping to mitigate the challenges that this

audience faces from cell signal interference with their equipment was the guiding design goal of SurvInt.

SurvInt is an innovative GIS application that provides a special type of decision support to the community of professional land surveyors. The development was focused on this group of users with an emphasis on providing what Roth, Ross, and MacEachren (2015) call “useful data in a usable form”. The decision support offered by SurvInt does not operate at the policy level, like some GIS spatial analyses, but rather at the day-to-day operations level. Using this tool allows a surveyor to decide if a survey can be conducted without interference from nearby cell towers or if mitigation steps are necessary before the survey is conducted.

The SurvInt tool also has widespread geographical capabilities. SurvInt can be used in any county that provides its municipal and tax parcel boundaries as polygon feature classes. In addition, all the other data inputs to the model are publicly available, as described in Table 1. Once validated, as described in section 5.2.2, the SurvInt tool can support surveyors and cities across the country with accurate assessments of the potential for cell tower interference. This availability should improve survey efficiency by reducing the number of surveys whose results are corrupted by RF interference.



## References

- Allen, David W. 2011. *Getting to Know ArcGIS : ModelBuilder*. Redlands, CA: ESRI Press.
- Alves, Paul, Carmen Wong, Matthew Clampitt, Eric Davis, and Eunju Kwak. 2018. "Tracking down Interference with Mapping." *GPS World*. May 30, 2018. <https://www.gpsworld.com/innovation-tracking-down-interference-with-likelihood-mapping/>.
- Anderson, Cary. 2020. "Before You Map: Audience, Medium, and Purpose." *GEOG 486: Cartography and Visualization*. 2020. <https://www.e-education.psu.edu/geog486/node/642>.
- ARPANSA. 2017. "ARPANSA Environmental EME Reports." Text. ARPANSA. ARPANSA. April 26, 2017. <https://www.arpansa.gov.au/research/surveys/environmental-electromagnetic-energy-reports>.
- Bolstad, Paul. 2016. *GIS Fundamentals: A First Text on Geographic Information Systems, Fifth Edition*. 5th edition. Acton, MA White Bear Lake, Minnesota: XanEdu Publishing Inc.
- Ceniza-Levine, Caroline. 2018. "Five Ways To Confirm If A Business Idea Is Viable Before You Launch." *Forbes*. December 23, 2018. <https://www.forbes.com/sites/carolinecenizalevine/2018/12/23/five-ways-to-confirm-if-a-business-idea-is-viable-before-you-launch/>.
- City of Signal Hill. 2021. "History of Signal Hill." 2021. <https://www.cityofsignalhill.org/218/History-of-Signal-Hill>.
- CLSA. 2021. "2022 Joint Surveying Conference." California Land Surveyors Association. 2021. <https://www.californiasurveyors.org/conference.aspx>.
- County of Los Angeles - eGIS. 2020. "County of Los Angeles Enterprise Geographic Information Systems." 2020. <https://egis-lacounty.hub.arcgis.com/>.
- Duffy, Linda. 2020. "Fundamentals of Surveying: Land Surveying, GIS and Geomatics." Point of Beginning (POB). August 19, 2020. <https://www.pobonline.com/articles/102233-fundamentals-of-surveying-land-surveying-gis-and-geomatics>.
- Esri. 2013. *Python Scripting for ArcGIS*. Redlands: ESRI Press.
- . 2019. "Feature Class Basics—ArcGIS Help | ArcGIS Desktop." ArcGIS Help. Feature Class Basics. 2019. <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/geodatabases/feature-class-basics.htm>.

- . 2021a. “Creating Raster DEMs and DSMs from Large Lidar Point Collections.” Help | ArcGIS Desktop. 2021. <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/lidar-solutions-creating-raster-dems-and-dsms-from-large-lidar-point-collections.htm>.
- . 2021b. “Getting Started with the Geodatabase.” ESRI Training Course. Getting Started with the Geodatabase. 2021. <https://www.esri.com/training/catalog/57630436851d31e02a43f19e/getting-started-with-the-geodatabase/#!/catalog/search>.
- . 2021c. “Guidelines for Arcpy.Mp.” ArcGIS Pro | Documentation. 2021. <https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/guidelines-for-arcpy-mapping.htm>.
- . 2021d. “How the Snap Raster Environment Setting Works.” ArcGIS Pro | Documentation. 2021. <https://pro.arcgis.com/en/pro-app/latest/tool-reference/environment-settings/how-snap-raster-environment-works.htm>.
- . 2021e. “Python in ArcGIS Pro.” 2021. <https://pro.arcgis.com/en/pro-app/latest/arcpy/get-started/installing-python-for-arcgis-pro.htm>.
- . 2021f. “Updating and Fixing Data Sources.” ArcGIS Pro | Documentation. 2021. <https://pro.arcgis.com/en/pro-app/latest/arcpy/mapping/Updatingandfixingdatasources.htm>.
- . 2021g. “Visibility (Spatial Analyst).” ArcGIS Pro | Documentation. 2021. <https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/visibility.htm>.
- . 2021h. “A Quick Tour of Creating Tools with Python—ArcGIS Pro | Documentation.” ArcGIS Pro Documentation. A Quick Tour of Creating Tools with Python. April 20, 2021. [https://pro.arcgis.com/en/pro-app/latest/arcpy/geoprocessing\\_and\\_python/a-quick-tour-of-creating-tools-in-python.htm](https://pro.arcgis.com/en/pro-app/latest/arcpy/geoprocessing_and_python/a-quick-tour-of-creating-tools-in-python.htm).
- . 2021i. “BUG-000119907: The Apply Symbology From Layer Tool Does Not Work.” Support.Esri.Com. July 9, 2021. <https://support.esri.com/en/bugs/nimbus/QIVHLTAwMDExOTkwNw==>.
- . 2021j. “Common Reasons for Using Map Services.” ArcGIS Server | Documentation for ArcGIS Enterprise. 2021. <https://enterprise.arcgis.com/en/server/latest/publish-services/windows/common-reasons-for-using-map-services.htm>.
- FCC. 2011. “Public Safety Tech Topic #17 - Propagation Characterization.” Federal Communications Commission. August 11, 2011. <https://www.fcc.gov/help/public-safety-tech-topic-17-propagation-characterization>.
- Federal Aviation Administration. 2020. “Air Traffic By The Numbers.” Federal Aviation Administration. September 21, 2020. [https://www.faa.gov/air\\_traffic/by\\_the\\_numbers/](https://www.faa.gov/air_traffic/by_the_numbers/).

- Friis, Harald T. 1946. "A Note on a Simple Transmission Formula." *Proceedings of the IRE* 34 (5): 254–56. <https://doi.org/10.1109/JRPROC.1946.234568>.
- GitHub. 2021. "What Is My Disk Quota? - GitHub Docs." GitHub Documentation. 2021. <https://docs.github.com/en/github/managing-large-files/working-with-large-files/what-is-my-disk-quota>.
- GPS.gov. 2020. "GPS.Gov: GPS Spectrum and Interference Issues." Government Information. GPS.Gov. July 23, 2020. <https://www.gps.gov/spectrum/>.
- Hata, Masaharu. 1980. "Empirical Formula for Propagation Loss in Land Mobile Radio Services." *IEEE Transactions on Vehicular Technology* 29 (3): 317–25. <https://doi.org/10.1109/T-VT.1980.23859>.
- Hibma, Kevin. 2018. "Controlling Sub-Model Processing Order within Master ModelBuilder Model?" Geographic Information Systems Stack Exchange. December 20, 2018. <https://gis.stackexchange.com/questions/306592/controlling-sub-model-processing-order-within-master-modelbuilder-model>.
- Keranen, Kathryn., and Robert. Kolvoord. 2014. *Making Spatial Decisions Using GIS and Remote Sensing : A Workbook*. Redlands, Calif.: ESRI Press.
- Lavery, Diana. 2019. "My First Impressions of ModelBuilder in ArcGIS Pro." *ArcGIS Blog* (blog). March 13, 2019. <https://www.esri.com/arcgis-blog/products/arcgis-pro/analytics/my-first-impressions-of-modelbuilder-in-arcgis-pro/>.
- Milione, Ron. 2011. "Fundamentals of RF Propagation in Electronic Warfare," December, 13.
- Norhisyam, Ahmad, A.M. Suldi, Juazer Abdul Hamid, and Dinesh Sathyamoorthy. 2013. *Effect of Radio Frequency Interference (RFI) on the Global Positioning System (GPS) Signals*. <https://doi.org/10.1109/CSPA.2013.6530041>.
- Odendahl, Liz. 2019. "Supervisor Hahn Seeks Cell Tower Ordinance." Supervisor Janice Hahn. March 5, 2019. [https://hahn.lacounty.gov/blog\\_supervisor\\_hahn\\_seeks\\_cell\\_tower\\_ordinance](https://hahn.lacounty.gov/blog_supervisor_hahn_seeks_cell_tower_ordinance).
- Roth, Robert, Kevin Ross, and Alan MacEachren. 2015. "User-Centered Design for Interactive Maps: A Case Study in Crime Analysis." *International Journal of Geo-Information* 4 (March): 262–301. <https://doi.org/10.3390/ijgi4010262>.
- Slocum, Terry A. 2005. *Thematic Cartography and Visualization*. Upper Saddle River, N.J.: Prentice Hall.
- Stone, Barron, and Olivia Stone. 2018. *Learn Object-Oriented Design Principles*. Online. LinkedIn Learning. <https://www.linkedin.com/learning/programming-foundations-object-oriented-design-3/learn-object-oriented-design-principles>.

- Trimble, ed. 2014. “Trimble Use Case Information : A Presentation to DOT Adjacent Band Compatibility Workshop #2,” December. <https://rosap.ntl.bts.gov/view/dot/34724>.
- Unwired Labs. 2021. “OpenCellID - Largest Open Database of Cell Towers & Geolocation - by Unwired Labs.” OpenCellID. 2021. <https://opencellid.org/#zoom=16&lat=37.77889&lon=-122.41942>.
- USDOT. 2018. “DOT GPS Adjacent Band Compatibility Assessment: Final Report | US Department of Transportation.” April 2018. <https://www.transportation.gov/pnt/dot-gps-adjacent-band-compatibility-assessment-final-report>.
- USGS. 2018. “Ground Surface Elevation - 30m - Overview.” May 15, 2018. <https://www.arcgis.com/home/item.html?id=0383ba18906149e3bd2a0975a0afdb8e>.
- USGS-TNM. 2018. “The National Map - Advanced Viewer.” October 22, 2018. <https://apps.nationalmap.gov/viewer/>.
- Walrand, Jean, and Pravin Varaiya. 2000. “CHAPTER 7 - Wireless Networks.” In *High-Performance Communication Networks (Second Edition)*, edited by Jean Walrand and Pravin Varaiya, 305–61. San Francisco: Morgan Kaufmann. <https://doi.org/10.1016/B978-0-08-050803-0.50012-5>.
- Wang, Huandong, Fengli Xu, Yong Li, Pengyu Zhang, and Depeng Jin. 2015. “Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment.” In *Proceedings of the 2015 Internet Measurement Conference*, 225–38. Tokyo Japan: ACM. <https://doi.org/10.1145/2815675.2815680>.
- Wyatt, Kenneth. 2018. “Identifying and Locating Radio Frequency Interference (RFI).” *Interference Technology* (blog). September 7, 2018. <https://interferencetechnology.com/identifying-and-locating-radio-frequency-interference-rfi/>.
- Young, William F., Ari D. Feldman, Sheryl M. Genco, Azizollah Kord, Daniel G. Kuester, John M. Ladbury, Duncan A. McGillivray, et al. 2017. “LTE Impacts on GPS,” February. <https://www.nist.gov/publications/lte-impacts-gps>.