

University of Redlands

InSPIRe @ Redlands

MS GIS Program Major Individual Projects

Theses, Dissertations, and Honors Projects

12-2011

City of Redlands Safe Routes to Schools Shadow Mapping

Joseph William Crowley
University of Redlands

Follow this and additional works at: https://inspire.redlands.edu/gis_gradproj



Part of the [Geographic Information Sciences Commons](#), and the [Urban Studies Commons](#)

Recommended Citation

Crowley, J. W. (2011). *City of Redlands Safe Routes to Schools Shadow Mapping* (Master's thesis, University of Redlands). Retrieved from https://inspire.redlands.edu/gis_gradproj/46



This work is licensed under a [Creative Commons Attribution 4.0 License](#).

This material may be protected by copyright law (Title 17 U.S. Code).

This Thesis is brought to you for free and open access by the Theses, Dissertations, and Honors Projects at InSPIRe @ Redlands. It has been accepted for inclusion in MS GIS Program Major Individual Projects by an authorized administrator of InSPIRe @ Redlands. For more information, please contact inspire@redlands.edu.

University of Redlands

City of Redlands Safe Routes to Schools Shadow Mapping

A Major Individual Project submitted in partial satisfaction of the requirements
for the degree of Master of Science in Geographic Information Systems

by

Joseph William Crowley

Douglas M. Flewelling, Ph.D., Committee Chair

Ruijin Ma, Ph.D.

December 2011

City of Redlands Safe Routes to School Shadow Mapping

Copyright © 2011

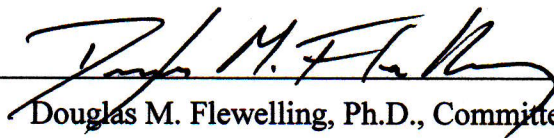
by

Joseph William Crowley

The report of Joseph William Crowley is approved.



Ruijin Ma, Ph.D.



Douglas M. Flewelling, Ph.D., Committee Chair

December 2011

Acknowledgements

Sincere thanks to Douglas M. Flewelling, Ph.D. for his wisdom, and guidance through this process. It was greatly appreciated, as was the healthy dose of good humor. Thank you to Ruijin Ma, Ph.D. for his patience, as well as the many hours he spent developing the script used in this analysis. His generosity in sharing his time and knowledge has left me indebted, and profoundly grateful. A special thanks to Debra Riley for her daily kindness, and encouragement. I would not be here if she had not rescued my application and sent that email several years ago. I am also deeply appreciative of the time and energy Lynn Flewelling contributed to the editing of this document. Thanks also to the other members of the MS GIS Program, especially Feng Ren, Ph.D., and Mark Kumler, Ph.D. I thoroughly enjoyed their classes, and will carry their lessons and enthusiasm for Geography with me.

Thank you to Phil Mielke, Tom Resh, and Matthew Bradbury from the City of Redlands for the opportunity to work on this interesting endeavor.

Thanks to Cohort 19 for adopting me as one of their own.

This project is dedicated with love to my parents, Joseph and Barbara Crowley for everything they have done over the years to help me reach this goal.

A special dedication also, to the memory of David Freda... whose inspiration lives on in those lucky enough to have known him.

“To them, I said, the truth would be literally nothing but the shadows of the images...And if there were a contest, and he had to compete in measuring the shadows ... would he not be ridiculous?”
...Socrates

“A man has made at least a start on discovering the meaning of human life when he plants shade trees under which he knows full well he will never sit.”
...D. Elton Trueblood

Abstract

City of Redlands Safe Routes to School Shadow Mapping

by

Joseph William Crowley

The City of Redlands seeks to improve the tree canopy coverage over key pedestrian zones with the intention of providing more shade to pedestrians in support of the Safe Routes to School Program. An analysis of the current canopy and the shaded sidewalks allowed city planners to distinguish those areas that are both “walkable” and in need of more shade. LIDAR data can generate a detailed and accurate measurement of the city’s canopy index, and was used to determine the total shadow coverage of trees and buildings. Overlaid with priority sidewalks, this map identified pedestrian zones in need of shade. The results allow the City of Redlands to more clearly understand the current canopy near school zones, and determine areas with a deficit of shade coverage.

Table of Contents

Chapter 1 – Introduction	1
1.1 Client.....	1
1.2 Problem Statement	2
1.3 Proposed Solution	3
1.3.1 Goals and Objectives	3
1.3.2 Scope.....	4
1.3.3 Methods.....	4
1.4 Audience	5
1.5 Overview of the Rest of this Report	5
Chapter 2 – Background and Literature Review	9
2.1 Urban Tree Canopy.....	9
2.1.1 Urban Tree Canopy Studies and Measurements	10
2.2 LIDAR	10
2.2.1 LIDAR for Tree Canopy and Urban Structures	11
2.3 Shadows	13
2.4 Summary	13
Chapter 3 – Systems Analysis and Design.....	17
3.1 Problem Statement	17
3.2 Requirements Analysis	17
3.2.1 Functional Requirements	18
3.2.2 Non-Functional Requirements	19
3.3 System Design	19

3.4	Project Plan	21
3.4.1	Original Plan	21
3.4.2	Revised Plan.....	22
3.5	Summary	23
Chapter 4 – Database Design.....		27
4.1	Conceptual Data Model	28
4.2	Logical Data Model	30
4.3	Data Sources	31
4.4	Data Scrubbing and Loading	33
4.5	Summary	34
Chapter 5 – Implementation.....		37
5.1	LIDAR Tile Processing.....	37
5.2	Raster and Vector Processing	38
5.3	Shadow Coverage	46
5.4	Shadow Mapping	51
5.5	3-D Efforts	52
5.6	Summary	54
Chapter 6 – Results and Analysis.....		55
6.1	Vegetation and Shadow Results.....	55
6.2	Discussion of Problematic Issues.....	58
6.3	Discussion of Results and Client's Needs	63

Chapter 7 – Conclusions and Future Work	67
7.1 Conclusions.....	67
7.2 Future Work.....	67
Works Cited	69
Appendix A. Raster Calculator Statements	75
Appendix B. Shadow Casting Script	76

Table of Figures

Figure 1.1:	Redlands and the Safe Routes to School study area.....	2
Figure 2-1:	Conceptual illustration of LIDAR data.....	11
Figure 3-1:	Major analytic components and workflow.....	20
Figure 4-1:	Conceptual model for main data elements.....	29
Figure 4-2:	Conceptual model for shadow casting.....	29
Figure 4-3:	Color infrared image with overlay of sidewalks and buildings showing errors.....	32
Figure 4-4:	Study area selection.....	33
Figure 5-1:	LIDAR data examination.....	37
Figure 5-2:	Raster and vector processing steps.....	38
Figure 5-3:	Multipoint record selection example.....	39
Figure 5-4:	First and last return comparison.....	39
Figure 5-5:	Point to raster interpolation with null values and focal statistics.....	41
Figure 5-6:	Unclassified, bare earth, and vegetation LIDAR point density.....	42
Figure 5-7:	Building mask creation.....	42
Figure 5-8:	Initial tree canopy procedure.....	43
Figure 5-9:	Raster reclassify of vegetation surface.....	44
Figure 5-10:	Final tree canopy procedure.....	45
Figure 5-11:	Editing polygons with aggregation and elimination.....	45
Figure 5-12:	Polygons with coincident vertices.....	46
Figure 5-13:	Overview of the shadow casting process.....	47
Figure 5-14:	Conceptual illustration of shadow casting.....	47
Figure 5-15:	Polygon to ASCII transformation process.....	48

Figure 5-16:	Shadow casting model parameters.....	48
Figure 5-17:	Shadow casting script operation.....	49
Figure 5-18:	Calculation of shadow points.....	50
Figure 5-19:	Shadow points and polygons from aggregation at ten feet.....	51
Figure 5-20:	Creation of overall shadow polygons.....	51
Figure 5-21:	Mapping of unshaded sidewalks.....	52
Figure 5-22:	3D buildings with flat roofs.....	53
Figure 5-23:	LIDAR vegetation points and the extrude between method.....	54
Figure 6-1:	Shadow map of city blocks with unshaded sidewalks.....	58
Figure 6-2:	LIDAR points and the Redlands tree canopy.....	59
Figure 6-3:	LIDAR vegetation points and color infrared image.....	59
Figure 6-4:	Point to raster error.....	60
Figure 6-5:	Tree canopy polygon from raster sampling and point aggregation...	61
Figure 6-6:	Raster to polygon process showing original raster overlaid with new polygon.....	61
Figure 6-7:	Canopy LIDAR points and aggregation distances.....	62

List of Tables

Table 1.	Functional and non-functional requirements.....	18
Table 2.	Original project timeline.....	21
Table 3.	Revised project timeline.....	23
Table 4.	Logical model matrix.....	31
Table 5.	LIDAR to multipoint conversion criteria.....	34
Table 6.	Coverage areas.....	56
Table 7.	Study area index measurements.....	57
Table 8.	Sidewalk index and coverage measurements.....	57
Table 9.	Summary of issues affecting measurements.....	63

List of Acronyms and Definitions

ASCII	American Standard Code for Information Interchange
DEM	Digital Elevation Model
DHM	Digital Height Model
DSM	Digital Surface Model
DTM	Digital Terrain Model
GIS	Geographic Information System
.las	LIDAR file format
LIDAR	Light Detection and Ranging
RS	Remote Sensing
TIN	Triangulated Irregular Network
UTC	Urban Tree Canopy

Chapter 1 – Introduction

Redlands, California is a medium sized city of approximately 70,000 people located 60 miles east of Los Angeles. The climate of Redlands is typical of other Southern California inland valleys, consisting of warm, dry summers and cool, wet winters. From June to October, Redlands averages only .5 inches of rain per month, and has an average maximum monthly temperature of over 80° F. The citizens of Redlands typically endure hot, sunny days where daily temperatures can easily top 100° F.

Because high temperatures are a standard feature of life in Redlands, the heat often prevents people from taking advantage of a pedestrian network that includes school zones, the central business district, the University of Redlands, mass transit hubs, and various parks and footpaths. To address this situation, the city has enacted several measures aimed at enhancing the livability of the city through various “greening” practices. One of these strategies is to maximize the effect of trees in the community through an active arbor program. The city has created a Quality of Life Department, initiated a “Street Tree Committee,” and maintains a database of over 38,000 municipal trees, which includes many city-owned citrus groves. The National Arbor Day Association has named Redlands a Tree City USA. In spite of this, the perception endures that Redlands lacks enough shade to effectively moderate temperatures for pedestrians.

The concern about the lack of shade extends to schoolchildren, who are encouraged to walk to school, through Redlands’ active participation in the Safe Routes to School Program. This program provides information and funding resources with the aim of improving “the health and well-being of children by enabling and encouraging them to walk and bicycle to school” (United States Department of Transportation 2011). In order to address these concerns and to ensure the best possible environment for pedestrians, the city decided to undertake an analysis of the current tree canopy, the existing shade available to pedestrians, and to identify areas of potential canopy and shade enhancement near several schools. The results of the analysis will guide future municipal tree-planting activities with a focus on an improved pedestrian experience for schoolchildren as per the Safe Routes to School guidelines (Fig 1.1).

1.1 Client

The client for this project was Mr. Philip Mielke, representing the City of Redlands. Mr. Mielke is the Administrator of GIS Services for the City of Redlands and, along with Tom Resh (GIS Administrator), acted as the contact for all issues and queries. The GIS Services department provides spatial analysis and GIS products to city government.

The City of Redlands has an collection of GIS data; including commercial data sets, proprietary information, and derived products, that were fully available for use. GIS Services is an active proponent of GIS within city government and has integrated GIS solutions, data acquisition, and spatial planning into a wide range of city projects covering areas as diverse as policing, utilities, event planning, and quality of life. The success of previous projects and the prominent role of GIS services in city government

indicate the likelihood of strong support, and that the results of this project would enhance the Redlands community through targeted arbor practices.

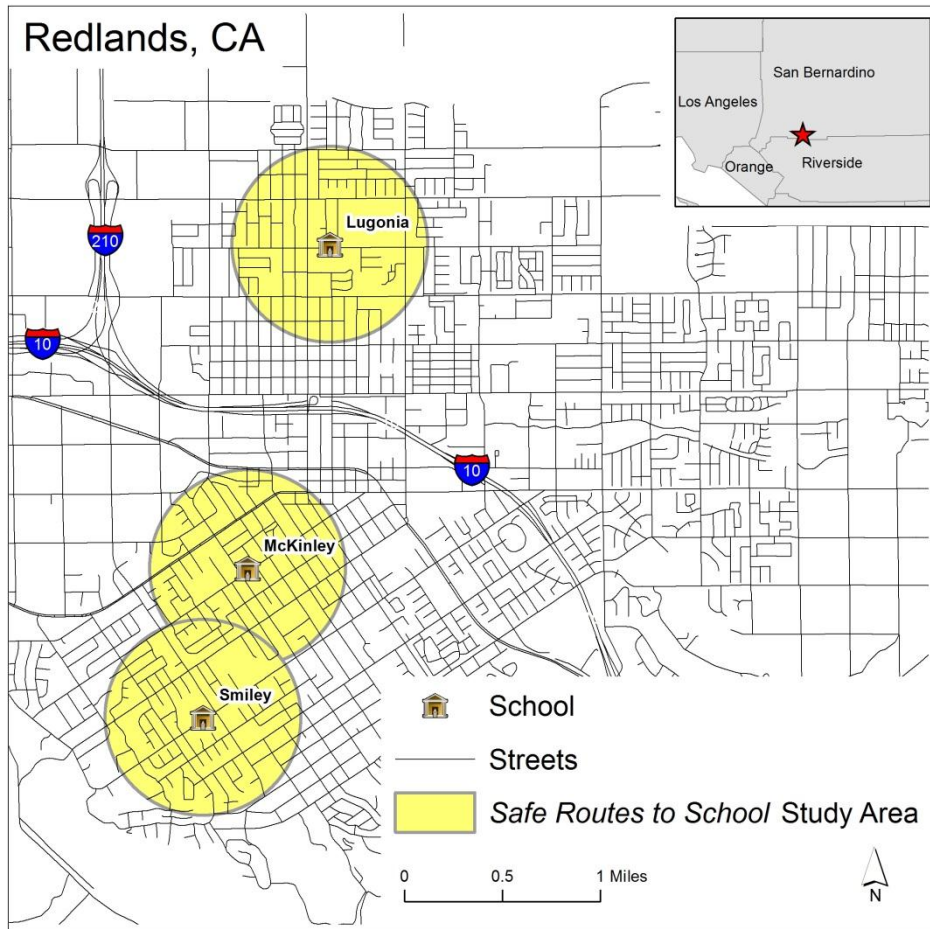


Figure 1.1 Redlands and the Safe Routes to School study area.

1.2 Problem Statement

The City of Redlands is an active participant in Tree City USA and Safe Routes to Schools. Both of these programs encourage the development and maintenance of a robust and healthy urban tree canopy (UTC) that provides adequate shade to pedestrian areas. The city has a proven history of proactive arbor programs, encouraging and subsidizing both public and private tree planting. Previous tree planting programs within the city have been ad-hoc and demand-driven, and have not targeted specific areas that are known to have a shade deficit.

In order to best provide shade in pedestrian areas, specifically around public schools, the city requires an understanding of the extent of the current tree canopy, the overall shade provided by trees and buildings, and the identification of areas of insufficient

shade. This analysis was delivered in a GIS format that was compatible with current systems in use. In addition to GIS files and maps, the analysis and resultant tools were flexible enough to estimate shade at specific dates and times, and were easily used by city staff without advanced programming skills.

1.3 Proposed Solution

The availability of numerous GIS resources within local government, and the unique geographic component of the problem provide ample reasons to address this problem from a spatial perspective. The city was not only interested in developing the richest possible UTC, but also intended to ensure the canopy provides the maximum shade coverage in key pedestrian areas near schools. These aspects of the project were ideally suited to the use of a GIS. A geographic analysis of the tree canopy can generate information about how much shade is currently available compared to recommended national standards. The incorporation of pedestrian network data into the canopy analysis allows a deeper examination of Redlands' situation and provides new information about potential tree-planting zones. Ideally the results of this project will lead to informed, fact-based decision making by city authorities to maximize canopy coverage in the near future.

Urban tree canopy estimates using GIS and remote sensing techniques are a recognized and accepted method of analysis (Poracsky & Lackner, 2004). The client provided Light Detection and Ranging (LIDAR) data for the study area. LIDAR data is composed of dense coverage of high resolution points that record a feature's height among other characteristics. LIDAR data have become the preferred way of analyzing tree canopies because of the relative ease of collection as well as the rich data sets that result (Dwyer, Miller 1999). By combining LIDAR data with existing city data sets such as building footprints and sidewalks, and the use of a custom shadow generation script, a UTC and shade analysis of the study area was possible.

This analysis identified and mapped the following variables:

- The overall UTC location and UTC index of the study area
- The overall shade index, as well as individual shade maps for trees and buildings
- Sidewalks in the study area that are devoid of shade coverage

1.3.1 Goals and Objectives

The goal of this project was to identify potential zones for tree planting within the client's study area to maximize shade coverage for pedestrians. This analysis resulted in the provision of data to city planners to geographically target tree planting in support of the Safe Routes to School program.

A series of technical objectives supported the overall goal. The main technical objectives were: processed LIDAR data, rasterized vegetation map, polygon vegetation map, shadow tool development, building and vegetation shadow projection, and final analysis. These technical objectives and the multiple processes within each step will be described in greater detail in section 1.3.3 and in Chapter 5.

1.3.2 Scope

In consultation with the client, it was decided that the shadow analysis project would map the UTC, determine the canopy index score for the study area, evaluate the current shade potential in key pedestrian zones at a specific date and time, and identify those pedestrian areas most suitable for future tree planting. The project would deliver the results of the analysis to the client as a series of GIS files in digital form, as well as soft copies of all derived products produced during the analysis in GIS compatible formats.

The proposed project was scheduled to run for approximately 10 months. This was judged to provide ample time for all necessary GIS operations, as well as sufficient time for the client to examine preliminary results and approve the final products. Client meetings were conducted on an as-needed basis. It was beneficial for the client to actively participate in the development of the project to avoid problems and miscommunications. These meetings were held at the offices of the GIS Department for the City of Redlands, and served as an update on progress as well as giving the client an opportunity to make suggestions or contribute ideas.

The client was solely responsible for providing all data sets for the project. The use of other data sources was possible, but was not required or requested. Every effort was made to ensure accurate analysis, but it was agreed that any errors in the final analysis or products due to faulty data sets did not change the project scope, as it was initially defined. The client was also responsible for ensuring the implementation of deliverables within their own organizations.

When both parties approved the initial project proposal, no further changes were possible, unless both parties agreed in writing that the changes were minor, feasible, and would not adversely affect the deliverables or schedules.

1.3.3 Methods

The project followed a waterfall method, in which technical objectives were linked in a progression of steps that built on the results of the previous outcomes. This linear approach was necessary due to the dependence on a single LIDAR data set that underwent a series of transformations from raw data into the component GIS features. The transformative process from remote sensing data to raster cells to vector polygons, with pre-developed models and subsequent post-production modeling of derived products, included multiple processing tasks with their own unique sub-methodologies. The steps outlined below constituted the main project objectives in the chronological order in which they were conducted. The different data types and the fairly complex procedures also required a modified spiral approach within each major objective, where an evaluation of results determined whether an acceptable result was achieved, or if a different approach was warranted.

An initial data design was composed to capture all project components and results with the intention of creating a logical data structure that would facilitate functionality and analysis. Separate databases were then created to manage the working files of each major technical objective, where similar data types and results could be stored and accessed. File naming conventions were important to identify a plethora of intermediate results of complex processing procedures. During the course of the project, where some

results were generated and then kept or discarded, the pre-planning of data structures allowed a more precise control and organization of overlapping procedures and results.

Following the study and design of data structures, the raw LIDAR files were analyzed and cataloged by important characteristics, such as point spacing and geographic coverage. They were also converted to a multipoint vector format and classified according to ground based features including vegetation and bare earth. Vectorized LIDAR vegetation data were then converted to raster cells in order to separate vegetation from other features, and then further classified according to a minimum height threshold. Vegetation rasters were then converted back to polygons that provided a more realistic vegetation canopy footprint, and would better facilitate further modeling and analysis.

The process of projecting a shadow from either a 3D shape or from the LIDAR data was tested to evaluate the best method of shadow casting. Trial and error proved that using trigonometric functions within a custom scripting environment provided the best results. The actual process of creating shadows involved the use of the custom script and raw LIDAR data in conjunction with vegetation and building footprints vector files. The resultant building and vegetation shadows were cleaned and merged to create an overall shadow map of the study area. This shadow map was then analyzed with city sidewalks within the study area to identify those pedestrian areas in need of enhanced shade coverage.

1.4 Audience

The primary audience for this study is the client, the GIS Department of the City of Redlands. They will use the results to create maps and data to plan and analyze the implications of an enhanced UTC in school zones and other pedestrian areas. A secondary audience will be the city planners themselves, who would use the maps when designing or directing urban renewal, city maintenance, or arbor activities within the study area. A tertiary audience may include various special interest and civic groups, such as the Climate Action Task Force and the Redlands Conservancy. These groups have demonstrated a strong interest in Redlands' urban greening policies and could use the information and maps to discuss tree-planting strategies, as well as for public information campaigns.

1.5 Overview of the Rest of this Report

The remainder of the report is dedicated to an in-depth explanation of all components of the project. A detailed description of the research, systems, data, and analysis will give the reader a comprehensive understanding of how the results were achieved, while serving as a guide for further exploration on the topic. Chapter 2 outlines previous research on the importance of the UTC and the use of LIDAR in tree canopy studies. Chapter 3 covers the system design and project planning, as well as what modifications were necessary as the project evolved. Chapter 4 is dedicated to data concerns and explains the reasoning behind the structures used, data relations, models, collection, and cleaning. Chapter 5 explores the project implementation and details the procedures of each technical objective, including the various processes of trial and error and why certain aspects of the project were either discarded or reinforced. Chapter 6 contains the

final results and delivers the shade analysis as a series of maps, calculations, and accompanying explanations. Chapter 7 concludes the project with final observations and suggestions for future work.

Chapter 2 – Background and Literature Review

The study of urban shadows in pedestrian zones is an interesting problem that crosses several disciplines. The unique intersection of natural and human geography lends itself to an examination using GIS. Remote sensing data, street and sidewalk maps, patterns of activity, and databases of municipal trees are all singular sources of valuable information that, combined astutely, can produce a reasonable analysis. However, using a GIS to understand dynamic phenomenon rather than discreet attributes of human activity is challenging (Batty, 2003). Research has shown that a GIS is an effective way of examining the urban characteristics of “walkability” (Schollsberg et al., 2007), and actual movement patterns may not be as important if urban classification zones are used, each with their own set of attributes (e.g., density, type of use, temporal patterns, or ownership). These zones help prioritize the analysis by identifying public areas of high activity where the impact of a municipal intervention will have the greatest effect. Because the project will utilize school zones as the study area, the concerns of Batty (2003) are not considered a hindrance.

The Safe Routes to Schools program identifies urban trees as one of the important components of a positive pedestrian environment for children. The program not only encourages tree planting, but rewards effective programs with grants to plant more trees. In addition, trees and shrubs planted on parking and median areas are recognized as an important safety feature. (National Center for Safe Routes to School, 2011). There is evidence that the greening of school zones has a positive effect on the academic performance of children, as well as encouraging positive social behavior (American Forests, 2009). There is also evidence that Attention Deficit Disorder can be significantly mitigated by the presence of mature green trees in a child’s immediate surroundings (Kuo and Taylor, 2004).

2.1 Urban Tree Canopy

There has been a significant amount of research on the benefits of a robust and healthy tree canopy in the urban environment. Major metropolitan areas acknowledge the general importance of the urban tree canopy (UTC) and have commissioned studies using GIS and remote sensing (RS) to provide recommendations on tree planting (McPherson et al, 2008). The City of Redlands has recognized for some time the importance of the UTC, and has also authorized reports to study the feasibility of increasing the number of trees in the city (Arnold, 2002).

While most recognize the value of a green city, exactly how “green” the urban landscape must be is a matter of less certainty, and depends upon the specific objectives to be achieved. There are numerous benefits to maintaining urban forests, including cleaner air, improved animal habitat, wind reduction, cooler urban temperatures, run-off and watershed maintenance, a better “sense of place,” and even increased property values (Carver et al, 2004; Ebenreck, 1989; Nowak, et al., 2010). Urban planners and city managers have long recognized that effective arboriculture contributes to a better and healthier environment for citizens (USDA and USFS, 2010). In the realm of ozone mitigation, increasing the urban tree canopy is a successful factor in the reduction of

ozone (Nowak et al., 2000), and GIS studies of current urban tree canopies include recommendations for future canopy enhancement with the goal of ozone mitigation (USDA and USFS, 2006).

2.1.1 Urban Tree Canopy Studies and Measurements

If it is generally agreed that a robust tree canopy enhances the urban landscape, any proposal to increase the canopy must be accompanied by an understanding of the current state of the canopy, as well as ways of making recommendations and measuring improvements. Researchers have used an urban canopy index score to identify the area of canopy on a number of different scales. Studies of coverage per person, per urban area, and per county all yield useful index scores, but are not necessarily the best approach for Redlands. Therefore it is important to initially define the terms “tree canopy” and “tree canopy index.” The UTC is the layer of leaves, branches, and stems of trees that cover the ground when viewed from above (USDA and USFA, 2009). For the purpose of this study the definition of a tree canopy index is “the ratio of the area of the city covered by tree canopy” (American Forests, 2009).

There has also been some experimentation with a variety of data sources and methods for tree canopy measurements (Azizi et al., 2008), each with its own apparent advantages and characteristics (Maco and McPherson, 2002; National Association of State Forests, 2009). These methods include Synthetic Aperture Radar (Indumathi et al., 2010), as well as standard remote sensing techniques, ground surveys, aerial photography, and national level data sets, (Patterson and Mowrer, 2003). Poracsky and Lackner (2004) researched various combinations of all of these.

In addition to estimating the canopy area, this analysis considered the amount of canopy covering sidewalks, streets, and pedestrian areas. These estimates will yield the true effect of a canopy’s coverage and a higher score will indicate a healthier canopy more capable of mitigating the urban heat island effect for pedestrians.

2.2 LIDAR

Because of its high resolution, wide coverage, and relative ease of acquisition, Light Detection and Ranging (LIDAR) data is used for a variety of purposes. Aerial LIDAR surveys involve gathering data from an aircraft where thousands of pulses per second are emitted in a beam up to 90 cm wide. The data is recorded as points that contain information on a feature’s height and intensity characteristics (Fig. 2-1). Interesting and diverse applications such as route visibility (Bartie and Kumler, 2010) and meteorological forecasting (Dupont et al., 2004) are possible with LIDAR. The number of applications continues to grow with this emerging technology, as does its exposure to the general public through mainstream media (Lemonick, 2011). The best tools for canopy studies are debated in the literature but utilization often depends upon fiscal and technical issues. However, experts recognize that the complexity of the UTC requires a flexible and adaptive approach. LIDAR has become the preferred method of canopy analysis, and in many cases has proven to yield superior results (Walton et al., 2008).

2.2.1 LIDAR for Tree Canopy and Urban Structures

LIDAR is useful because the same data set can be used to facilitate the identification of new tree planting areas, the differentiation of various types of vegetation and non-ground features, and the mapping and analysis of pedestrian zones affected by canopy cover (Dwyer and Miller, 1999), all considerations of this canopy analysis. LIDAR data has the advantage of being very high resolution, thereby allowing a more precise measurement of the canopy. It has proven accurate enough that canopy analysis can be conducted using leaf-off data (gathered during winter months after deciduous trees lose their leaves) (Brandtberg et al., 2003). However, the accuracy of the analysis depends on the quality of the data. Inaccurate or poorly classified data can result in errors of tree canopy estimations of over 30% (Arnold, et al. 2009).

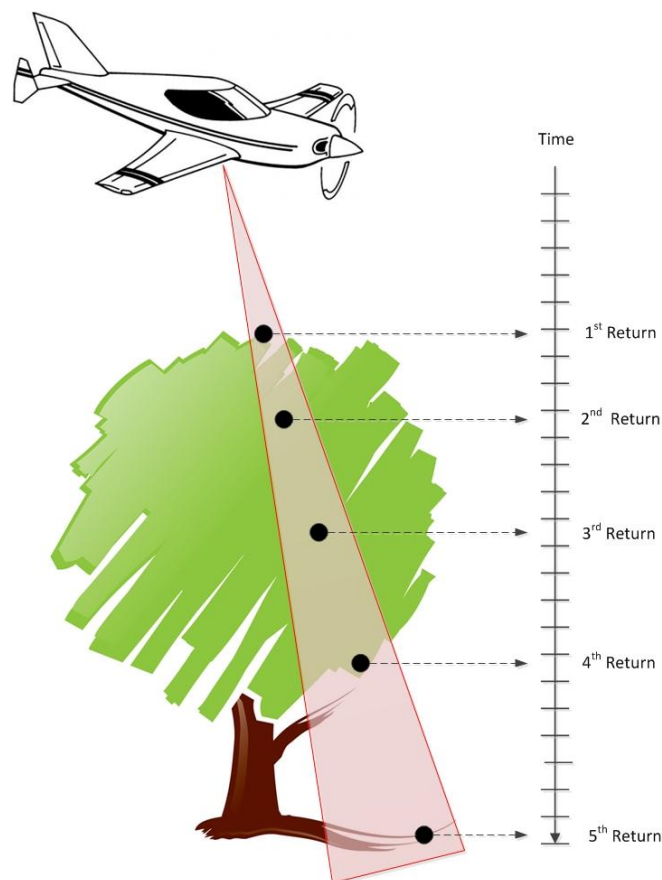


Figure 2-1: Conceptual illustration of LIDAR data.

LIDAR methodologies for canopy analysis involve the conversion of LIDAR data into raster cells, the classification of raster cells into unique groups, and further sorting into sub-categories based on attributes. This classification first attempts to identify the tree canopy areas as separate from other urban surfaces and vegetation. High-resolution data

can prove challenging when an individual feature (a tree's canopy) is represented by many pixels, but canopy extraction parameters have been well-defined and constitute a fairly straightforward exercise (Staub et al., 2009).

Two important indicators of a canopy's shade potential are area and height. However, some results show that the desired levels of accuracy are not always optimum (Popescu et al., 2002) due to the complex layering of tree canopies. Determining canopy height is a fairly common LIDAR exercise, but it can become complex and uncertainties should be thoroughly understood (Smith, 2003). Distinguishing individual tree crowns that are shorter than their neighbors or those that grow at odd angles is problematic. The identification of these features may be less accurate when using standard processing algorithms. Accurate calibration of data is often best achieved when using a combination of image processing generalization and ground-truthing with field visits or secondary data (Staub et al., 2009).

The use of ground-derived vector data to accompany the canopy analysis is a topic less explored and one that lends itself to a variety of approaches. At the most basic level, street and parcel data is useful for shade mapping. Vector data prioritized according to defined variables (such as pedestrian density) can identify future tree planting areas, as well temporal projections of estimates of shade values. Determining canopy characteristics using standard GIS software and data in conjunction with LIDAR data is well documented (Esri, 2009). Methodologies using GIS, remote sensing, and areal classification have proven successful in identifying tree-planting sites, as well (Wu, et al., 2008). The application of these methodologies to similar analysis may prove fruitful, especially if LIDAR data increases the accuracy of estimates and measurements. However, a LIDAR tree canopy analysis incorporating basic ground sampling will provide a further level of accuracy and understanding. Other researchers have documented consistent errors in using LIDAR to calculate vegetation height, and have suggested applied methodologies for correcting these errors (Hopkinson et al., 2005).

Others have used LIDAR to study tree canopies for fire fuel estimates, but noted that LIDAR data returns information for leaves, stems, and branches without the ability to distinguish between different types of tree features (Anderson et al., 2005). This is an important factor for this study since different tree features return different shade characteristics.

One advantage of using LIDAR data for canopy mapping in an urban environment is that LIDAR detects trees covered by building shadows that would otherwise be missed using traditional imagery techniques (O'Neil-Dunne, 2011), as well as being able to identify individual tree species (Kim et al., 2008). The difficulty of separating trees from buildings in sparsely sampled urban environments is also recognized and has been approached by using segmentation and classification algorithms within the context of 3D modeling (Secord and Zakhor, 2007). Problems can also arise due to inferior data sets, complex building shapes, or when vegetation and the built environment are in close proximity (Ma, 2005). Likewise, research into the types of LIDAR beam, and methodologies for improving accuracy of tree measurements is ongoing (Anderson et al., 2006).

These topics are interesting advances in LIDAR research and reveal the depth of issues facing GIS practitioners when using such data. Other cities have successfully used LIDAR to map their urban tree canopy (Morrow et al., 2001). The availability of LIDAR

data sets for the City of Redlands, as well as its universal recognition as a superior technology for tree canopy study, indicate that while problems exist, it is the best tool to undertake the analysis.

2.3 Shadows

A dearth of research on using LIDAR data to generate shadows was a cause for concern. One exception (Capuana, 2010) used the Esri Hillshade Tool to create a shade model. However, errors were noted due to the inability of the process to distinguish areas of partial shade within a raster format. This study also encountered the possible over-estimation of shadow areas caused by the transformation of LIDAR into 3D images. Similar image distortion can be seen in other work (Bartie and Kumler, 2010), although it was not recognized to have an impact on the results of the research.

The new suite of shadow tools in ArcGIS 10.0 offered a promising way forward. Upon preliminary investigation it seemed suitable in producing shadows in a 3D environment that can be exported and mapped in a more traditional planar fashion (Esri, 2010). These tools rely upon a series of trigonometric functions with feature data supplied by the user, and are intended to be used with the built environment only. Given the data at hand, it seemed possible to create adequate 3D representations of building structures for shadow casting. However, the creation of vegetation features and canopies in this environment was unknown, and was investigated in the course of this project.

Other commercial software applications offer methods of generating shadows, but are insufficient for rigorous shadow analysis in a GIS environment. One of the most popular is Google SketchUp and its accompanying suite of third party add-ons. Although it is capable of rendering shadows in a 3D environment, there is no way to export the shadows in either 3D or vector formats for use in a GIS. Other commercial and proprietary applications have been developed for stream monitoring (Laurie and Reichert, 2010) and landscape design scenarios (Kellern, 2010). These programs are unsuitable for modeling large urban areas due to limitations such as the reduction of tree canopies to generic shapes, or numeric results that give a shade index, without specifying the ground area of the shade.

2.4 Summary

There is ample evidence in the literature of the multiple benefits of an enhanced tree canopy. The City of Redlands will gain both measurable and intangible advantages with any canopy enhancement program. Although shaded streets for schoolchildren are the most obvious result examined in this study, additional benefits will certainly add weight to any arbor program.

There are a variety of methods for measuring tree canopies, each of them having advantages and disadvantages. Choosing a method often depends upon the data at hand, budgets, skills, time, or the actual problem to be analyzed. LIDAR has proven useful to accurately measure virtually any surface feature, both natural and man-made, although limitations are recognized. The availability of data for Redlands is a key factor in the decision to base this canopy analysis on LIDAR.

Although LIDAR data is readily available, the analysis encompasses areas of GIS that are relatively unexplored and have had limited successful results. Producing shadows from LIDAR data is a nascent area of GIS and is not clearly documented or understood. Previous efforts have encountered limitations due to methods, data, or

technological restrictions, with many practitioners having to invent their own methodology. These methods, while appropriate for some applications, may not be suitable for a detailed study of a complex canopy over a large urban area. This study attempted to apply some of the previous efforts as well as exploring new ways of shadow mapping with the intention of using the best method to create the most accurate map.

Chapter 3 – Systems Analysis and Design

Planning and design was an important aspect of the project that created an overall structure for monitoring activities and measuring achievement. The client requested a mapping file that was the result of an analysis, and did not need the implementation of an organizational system or the creation of GIS tools. Many of the complex planning exercises, such as technology seminars, information product descriptions, user systems requirements, and cost-benefit analysis, were unnecessary with a single-purpose project such as this. Therefore, the planning of the project was fairly conventional and followed standard practices associated with delivering information for decision makers without consideration for enterprise implementation or long-term support (Tomlinson, 2007).

3.1 Problem Statement

The City of Redlands seeks to create a pleasant environment for local students; therefore it is concerned with the amount of shade on key pedestrian walkways. Well-shaded sidewalks support the City's participation in the Safe Routes to School initiative. In addition, a healthy urban tree canopy (UTC) has been shown to have numerous positive benefits for communities including such diverse advantages as reduced ozone levels, and enhanced real estate prices. While recognizing the benefits of a robust UTC, it is essential that the city pursues arbor enhancement with a clear plan that will provide shade to areas in need.

For a better understanding of these needy areas, the City required a study that identified the amount and location of shade in pedestrian zones at a specific date and time. To achieve this analysis there was a comprehensive mapping of the current UTC, a measurement of the overall shade provided by trees and buildings, and the identification of areas of insufficient shade. These components allowed the city to understand shade deficiency around schools and to target tree planting specifically to remedy this situation.

3.2 Requirements Analysis

The requirements analysis supported a primary goal of creating and delivering a shade map to the client. The final product consisted only of Esri-compatible files, therefore most of the project activities revolved around developing methodologies for creating those files. These requirements evolved after several client consultations in which the client's needs and intentions were discussed and documented. The requirement analysis does not contain any reference to how the client intends to use or display the results, only to the process involved in their creation. The client's main requirements were the utilization of LIDAR data that resulted in a shade map. They had no further demands based on methodology, post-analysis use of the data, or the development of other tools and systems for manipulating data (such as a web interface or custom scripted tools).

The requirements analysis consists of both functional and non-functional requirements (Table 1). Functional requirements were defined as what the system must do. In this case, it was the required data and procedures to produce the shade analysis.

Non-functional requirements were related to how the procedures were done. These were mainly the software packages chosen for practical reasons.

Table 1. Functional and non-functional requirements.

Requirement	Functional / Non-Functional	Description
LIDAR Analyst (ArcGIS 10.0 extension)	NF	LIDAR software chosen based on compatibility with data, suitability, and availability
ArcGIS 10.0 and Spatial Analyst extension	NF	GIS software chosen based on compatibility with client, suitability, and availability
LIDAR file processing	F	Input, conversion, selection by class, point file
Raster processing	F	Focal statistics, point to raster, polygon to raster, natural neighbor, raster calculator, raster to polygon,
Geo-processing	F	Aggregate, buffer, calculate, clip, edit, eliminate, erase, interpolate, merge, select
Coverage processing	F	Build, feature to coverage, ungenerate
Shadow casting	F	Create shadows from processed LIDAR data
Data	F	LIDAR points, building footprints, sidewalks, schools, solar position
Data	NF	Remote sensing image, streets
Output in Esri format	F	Tree shadows, building shadows, unshaded sidewalks

3.2.1 Functional Requirements

The shade analysis allows the client to visualize those areas of the city that are shaded during a specific time, and to subsequently identify areas that are in need of tree planting based on defined priorities (in this case, proximity to schools). Specifically, the client was presented with maps and data representing: percentage of the study area under tree canopy, areas shaded by trees and buildings at a specific date and time, and sidewalks that are not shaded during that time. These will allow the client to advocate for further tree planting within the city based on areas of defined need.

The analysis was conducted entirely within a stand-alone desktop environment. The results were delivered to the client in the form of Esri mapping files (.shp), which the city may then use to raise awareness among appropriate government and civic groups. The client requested no other products, nor did they require access to the project itself other than information regarding the project methodologies. The file formats requested by the client are industry standards, and fully compatible with the project's development environment, so no data exchange issues existed.

The client did not require training, but the transition consisted of a detailed handover of data and a debriefing of project results and recommendations. The handover ensured the client is fully aware of all analytic procedures, data sets, and resultant products. They also requested suggestions for follow-up actions and possibilities for future studies. Data updates were problematic and were unnecessary before the city acts on the current analysis. Updates would require new LIDAR flights as well as a new analysis. Because the LIDAR data is relatively new and because the tree canopy changes slowly the need for updates is minimal and would not significantly affect the results of the current analysis.

Procedural details and specific considerations are examined below and are described in detail in Chapter 5.

3.2.2 Non-Functional Requirements

The non-functional requirements encompassed the software used for the analysis, as well as some ancillary data for orientation, comparison, and display. These were not a part of the analysis itself. Software was chosen based on availability and familiarity, but other packages could have been used. There was no functioning software or systems delivered to the client, thereby negating the need for many of the standard requirements of a technical review. The mapping files will reside on the client's network in file folders and database environments and no other special considerations are required.

3.3 System Design

As with the requirements described above, the system design refers to the work undertaken to develop a methodology and tool to create shadows from LIDAR data according to the client's criteria. The design included no consideration of how the data were used or where it resided after it was delivered to the client. As per the client's request, the final product was a series of GIS files that were fully compatible with the client's systems that could be used and manipulated as required.

The system design described below (Figure 3.1) identifies the main components of the analysis and shows the processing and transformation of each component into a useable product for the next segment. The component processing (downward vertical arrows) is simplified here into major tasks, but will be detailed further in Chapter 5. The system workflow follows a distinct iterative approach where each step must be fully completed before the next component is addressed. This approach was most appropriate due to the reliance of each component on a previously completed process, as well as the heavy dependence on LIDAR data for the major elements of the project. The files for each major component were stored in separate databases, which will be detailed in Chapter 4.

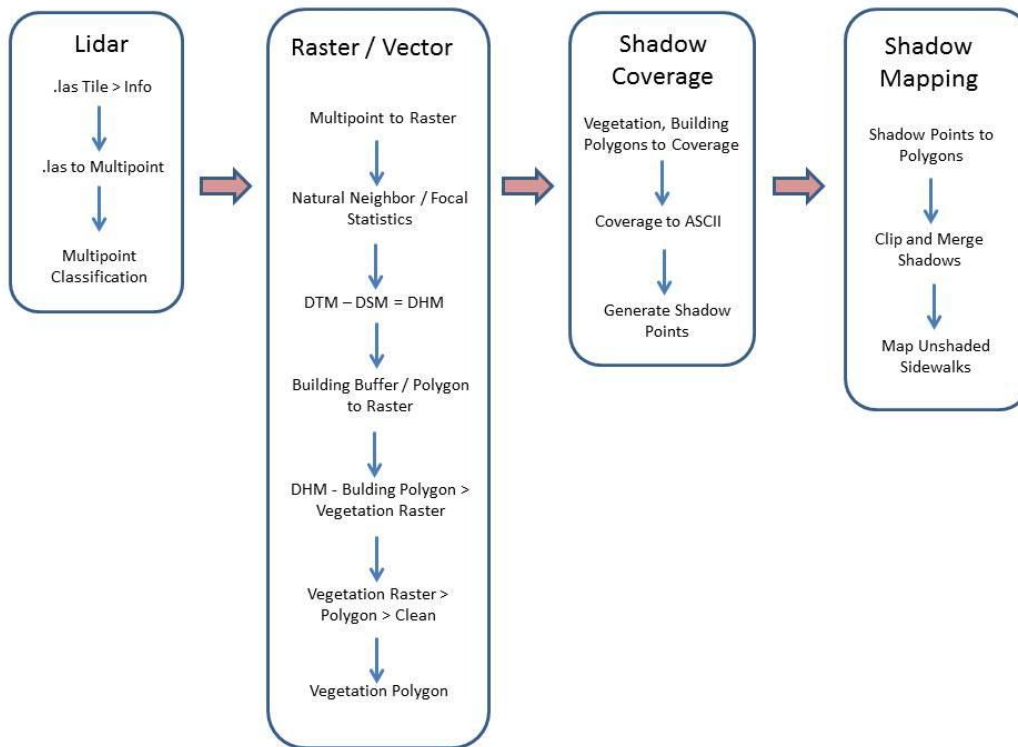


Figure 3.1 Major analytic components and workflow.

LIDAR data measuring tree and building heights were the primary elements of conversion and underwent a series of transformations as the analysis progressed. The LIDAR data were received pre-processed by the vendor in the “.las” format and converted to the Esri “multipoint” format. These point data were further processed into raster cells containing elevation data. The raster cells are used to extract tree features that are then converted into polygons. These polygons, overlaid with city features, give the total area of the city currently under canopy.

The processed polygons were also used to clip out areas of raw LIDAR data, giving detailed elevation information for each polygon, either trees or buildings. This elevation data was then processed through a custom script that identified shadow areas on the ground. The conversion and processing of these shadow areas to either tree or building shadows represented the final step of shadow casting. Shadows were then mapped with conventional city features (sidewalks), and simple geo-processing allowed the identification of non-shaded pedestrian routes.

The main system design challenge centered on the actual methods of shadow casting. Although the design illustrated above represents the final version, other more conventional methods were tried and discarded after yielding unsatisfactory results.

These other approaches are mentioned below in reference to the planning phase, as well as in Chapter 5.

3.4 Project Plan

The project plan was developed using the waterfall method. The chronologic dependence of processed data meant that parallel workflows were not possible. The early availability of data and the front-loading of processing tasks assumed there would be enough flexibility at the end of the project to ensure a timely delivery. As with most project timelines, the activities were planned to fit the deadlines, and not necessarily the actual amount of work required for a satisfactory completion. In this case considerable deadline extensions required to create a final product. These extensions were implemented after serious problems were encountered in some of the basic planning assumptions, especially involving the functionality of off-the-shelf software and the methods required to produce shadows.

3.4.1 Original Plan

The original plan was organized around the progression of project planning courses and the major milestones. Although there were a number of unknown elements to the project, standard methodologies for processing raster data and determining canopy coverage allowed for initial rapid progress. The objectives were distinctive and easily lent themselves to established milestones within the project. Due to the compounded nature of the analysis, by necessity each milestone was finished before the next operation started. This approach ensured full completion of each milestone and allowed for a better understanding of progress through the course of the project. Related thematic operations are defined by similar objectives and shared similar inputs and outputs. These thematic operations are color-coded by row in the table below (Table 2).

Table 2. Original project timeline.

Major Task	Milestone	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Client Discussions		x	x	x	x	x	x	x	x		
Receive Data			x								
Self-Training	Workshops, Online	x	x	x	x	x	x	x			
Project Proposal	Initiate		x								
	Scoping		x	x	x						
	literature Review		x	x	x						
	Develop methods			x	x	x					
	Refine Proposal					x	x				
	Finalize Work plan						x				
Data Review	Cleaning		x	x	x						
Data Design/Model	UML/Analysis Model				x	x	x	x			
LIDAR Processing	Point Cloud						x	x			
Vector Analysis	Priority Areas						x	x			
Prototype Modeling	Shade Analysis							x	x	x	
Final Modeling	Shade Analysis									x	x

An examination of the original plan shows that although a waterfall methodology was used, many tasks were performed concurrently, resulting in a modified waterfall approach. While this would seem problematic or contradictory to the traditional waterfall approach, it was actually necessary. This overlapping approach is a viable alternative to the pure waterfall, and well known within the realm of software development as the “Sashimi Model” (McConnell, 1996). Overlap exists both within and between major thematic tasks. An examination of the March schedule shows five different thematic operations occurring together. Additionally, the project was planned to use one sample tile of data as a pilot for testing. Work on the pilot tile would continue separately, while processing the main body of data followed behind. This was necessary as methods were tested on the sample and either discarded or accepted, depending on the quality of results achieved. Thus in March, prototype modeling could occur on the sample, while LIDAR processing was conducted on the main study area. In the meantime, the client was informed of progress and resources were reviewed for similar practices or problems.

Client discussions were conducted on a regular basis in the first half of the project, either remotely or on-site. This was beneficial in establishing a professional rapport as well as ensuring a clear understanding of the client’s vision, and resources.

Because of untried methods and unknown problems associated with such analysis, nearly half the total project time was devoted to data processing and modeling. In anticipation of the production and testing of several different shadow prototypes, considerable time was allocated to shadow modeling. The time allocated for shadow creation was disproportionately large because of the alarm at the scarcity of evidence in the literature regarding shadow creation, as well a lack of ideas from practitioners who were questioned on these methods. Eventually it became clear that these fears were well-founded and that even more time was needed for shadow work in the revised plan.

3.4.2 Revised Plan

As the project progressed it became apparent that timeline revisions were necessary (Table 3). Revised and new tasking schedules are indicated with a grey and red X, respectively. The scope of the study area was reduced after consultations with the client in July. The original study area was to include a half mile buffer around all Redlands schools. This was reduced to three schools after it became apparent that the processing time using the new scripted approach was exceedingly long and labor intensive. Although the same modified waterfall approach was still used, there were changes to the tasking as the schedule was extended.

Table 3. Revised project timeline.

Major Task	Milestone	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Client Discussions		x	x	x	x	x	x	x	x			x		x	x
Receive Data			x												
Self-Training	Workshops, Online	x	x	x	x	x	x	x							
Project Proposal	Initiate		x												
	Scoping		x	x	x							x			
	literature Review		x	x	x										
	Develop methods			x	x	x	x	x	x	x	x	x	x		
	Refine Proposal					x	x								
	Finalize Work plan						x				x			x	
Data Review	Cleaning		x	x	x						x	x	x		
Data Design/Model	UML/Analysis Model				x	x	x	x							
LIDAR Processing	Point Cloud						x	x							
Vector Analysis	Priority Areas						x	x							x
Prototype Modeling	Shade Analysis							x	x	x	x	x	x	x	x
Final Modeling	Shade Analysis									x	x				x

x = Task accomplished as scheduled. x = Task rescheduled. x = Task performed as per revised schedule.

The initial proposal and research stages were completed as scheduled, although method development was extended as several variations were tested without satisfactory results. Data preparation re-started in June with the introduction of a new script that required extensive data cleaning in a variety of formats. Data cleaning continued as the script was modified and perfected. When the scope of the project was reduced, vector analysis was simplified to a query of unshaded sidewalks (rather than a more elaborate pedestrian model including feature such as all sidewalks, business areas, and bus stops). This was accomplished quickly after the shadow casting was perfected.

Creating a suitable method of creating shadows proved to be the most difficult portion of the project (see Chapter 5). This is clearly reflected in the revised schedule, as method development and shade analysis consumed more than twice as much time as originally intended. Technical limitation with the software, as well as mistaken assumptions about its functionality, forced the development of a solution outside of the immediate Esri environment. The scripted solution went through several iterations before a final product was ready in early September. This had a knock-on effect as the post-analysis activities were also delayed by several months, and subsequently rescheduled.

3.5 Summary

An examination of the project’s systems and designs covered several related areas crucial to the project’s success. These areas included functional and non-functional requirements, as well as the system design and overall project plan. These three distinct but related elements were an integral part of the project. They were carefully developed to ensure the client’s needs were met in a timely fashion, and the project was successful.

The client’s requirements were identified through consultations and a comprehensive study of the problem, tools, and techniques available. These requirements formed the guide that allowed all subsequent project activities to effectively target specific

deliverables. The design of the analytic process positioned the available data with the appropriate process to ensure a logical process of data development would result in the proper outcome. This process was constructed within the framework of the requirement analysis, and was measured by a project timetable. The timetable placed all the project elements in a matrix of activities and milestones to create a platform for the measurement of progress. Problems encountered during the project were then easily identified not only by their data relations, but also by their association to the client's requirements and the project's systems. Thus, an issue with any of the projects components, requirements, or processes could be re-measured, planned and calibrated within the overall project framework.

Chapter 4 – Database Design

Before implementing any actual GIS work, it was necessary to conduct a detailed examination of the data. This included not only a review of data sources and standards, but also the analysis and design of data structures and relations. A thorough design ensured the analysis model was an actual representation of the real world, accurately describing features and behaviors. Because the creation of shadows within a GIS mimics a common phenomena, it is important to ensure the data elements in the model also occur in the natural world. Although these GIS elements are by necessity a compromise or a proxy for a much more complex system, a good design attempts to recreate them as accurately as possible without placing undue restrictions on functionality. The data review also confirmed only essential data for the analysis was used, and eliminated all other peripheral entities.

The review relied upon basic database principles to strengthen the model and to help focus on the most streamlined approach from raw data to a finished product. While these principles were not utilized in the strictest sense of database construction or normalization, they were used more like guiding principles to assist with the design. The domain closure axiom makes the assumption that all the elements in the model were named, or that the items in the database are the only ones of interest. This was a useful way of ensuring that extraneous information did not find its way into the system, and that everything not directly used to produce a result was excluded. Much of the land-use data, parcels, streets, and the city's tree database were eliminated at this point as "nice to have," and not a "must have" data set. While the domain closure helped to eliminate unnecessary items, the closed world assumption was used to examine completeness. This assumption supposes that all true facts are present and what is not present is therefore false. This concept was helpful in reinforcing the choices made with domain closure, as well as acting as a checklist for crucial datasets.

While these principles helped to ensure the database contained only essential elements for analysis, additional measures were necessary to further explore the data and the data relations between entities. Two modeling techniques were employed; the conceptual and logical data models. These models were visual representations of the data and data relations and were useful in understanding how the elements were analyzed and where they resided within the system. The models served as a roadmap to structure and classify data, and to guide the process towards a finished product. Although these models didn't guarantee an accurate result they did recognize that the data entities were present and in the proper sequence to produce that result. Minor changes were made to these charts as the analysis progressed and certain methods were discarded.

This phase of the project also examined other, more standard characteristics of the data, such as source, scale, projection, precision, accuracy, and other metadata elements. A thorough understanding of the nature and quality of the data was critical before proceeding with the GIS analysis. Since the results of the analysis ultimately depended on the initial quality of the data, any problems or issues needed to be recognized before proceeding. Although there were serious deficiencies in the metadata, the data seemed to be of a reasonable standard. Furthermore, the client who was the sole source of the data, was currently engaged in a number of projects using the same data sets, and there was no

possibility of obtaining better data. Therefore the lack of some information regarding the data sets did not inhibit progress.

4.1 Conceptual Data Model

The conceptual model was constructed to examine the interaction of real world elements in an abstract form. This theoretical consideration of the main entities is recommend for a variety of data development tasks, including a review of information products produced, the identification of key thematic layers based on information requirements, and the proper grouping of data sets (Arctur and Zeiler, 2004). Organizing the conceptual model allowed for a better understanding of the main data entities and their characteristics. It was also an opportunity to examine traits such as scale, resolution, projection, error tolerance, and temporal considerations in greater detail. The linkages in the conceptual model represent the interoperability of entities and are an element of the model that is as important as the data itself. These behaviors show not only how systems function in the real world, but confirm how the data must also be analyzed within the system.

The conceptual model of the shadow analysis may be read from the bottom up and shows LIDAR points and trees as the initial primary elements (Figure 4-1). The points make up a LIDAR cloud and constitute the main data source for further analysis. Trees are presented more as a conceptual element that are associated with the canopy, and in this case were not necessarily analyzed on an individual basis. The canopy and cloud are both composed of many trees and points that were treated as a whole entity.

The association between buildings and the LIDAR cloud was similar to the canopy, although the building data were composed of digitized footprints provided by the client. However, both elements relied upon the LIDAR cloud to interpret height information. Footprint information was crucial to the analysis in order to identify elements on the ground, and the canopy footprint was derived solely from the LIDAR cloud. The multiplicity of relations here was interpreted by the division of the study area into tiles. Although a tree canopy and a LIDAR cloud could be defined as one element, in this case they were classified as composed of many similar entities. Therefore a tree canopy could be referred to, and analyzed as, all the trees of one particular street, tile, or city. The same held true for the LIDAR cloud and buildings, which may be treated as individual items, or a class of entities.

One of the more interesting aspects of the conceptual model was the relation between the solar position and the development of a shadow. This element closely resembled the way the sun interacts with the Earth in the real world and therefore must be defined as such in the model. The solar position needed for a shadow calculation consists of the azimuth (or compass angle at which the sun appears in the sky) and its elevation above the horizon at a particular time. The end name informs, and its accompanying arrow signifies the position of the sun and its influence both in the real world and in the analysis. The dependency association and multiplicity indicators also signify that a specific solar position will produce varying types and amount of shadows.

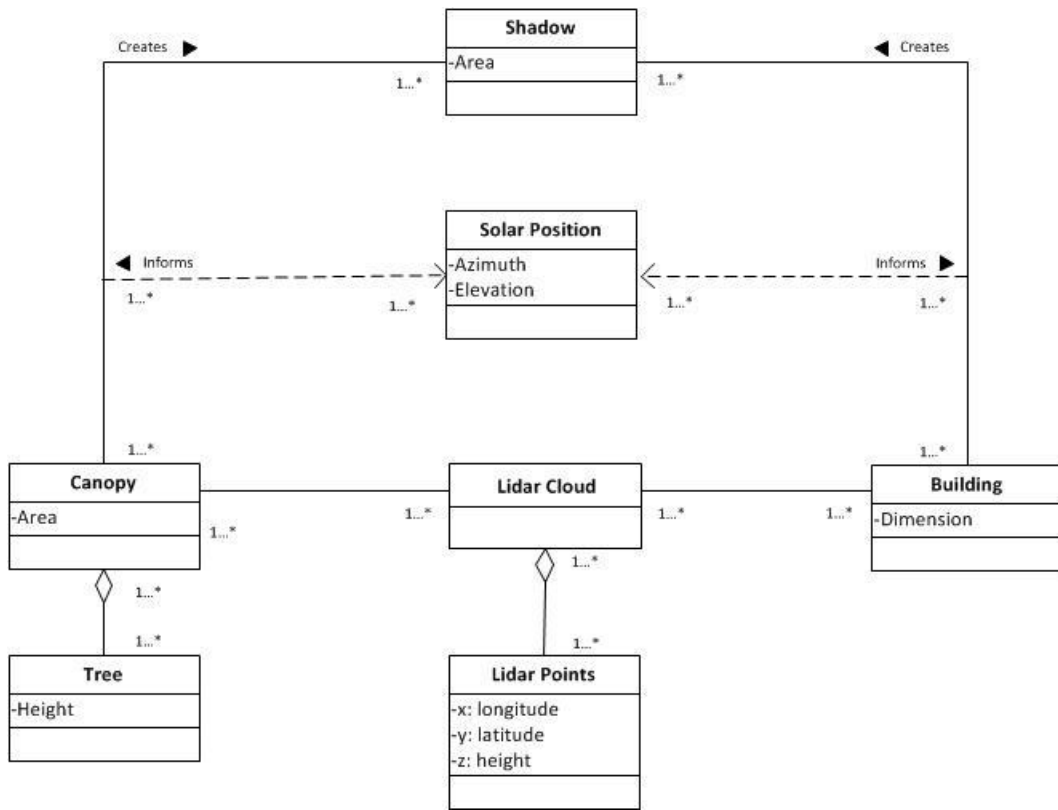


Figure 4-1: Conceptual model for main data elements.

The shadow class in Figure 4-1, the culmination of shadow development, is the result of the combined building and canopy shadows. The merging of building and canopy shadows, produced using the same solar inputs, yielded a representation of all shadows in the study area. As with the other elements, the same multiplicity rules apply. The combined shadow entity overlaid with sidewalks results in a map of all shaded sidewalks (Figure 4-2). The inverse representation is a map of unshaded sidewalks leading to the study area schools. The identification of the unshaded areas is the primary objective of the analysis, and Figure 4-2 illustrates how shaded sidewalks must be identified first and were the key to the analysis.

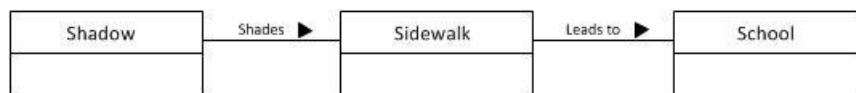


Figure 4-2: Conceptual model for shadow casting.

4.2 Logical Data Model

While the conceptual model was useful in confirming the relationship between project data elements and real world environments, a logical model was necessary to define the organization of tabular structures, and where the data resided in the greater scheme of specific databases. The elements of the logical model are much more specific and indicate actual files and their location.

This project utilized a simplified object-relational data model, where tables related to similar places in the study area but were not actually linked by a key field. Likewise, data elements represented real world entities, such as a tree canopy or a building with behaviors or attributes. These elements in turn influenced the behavior of other objects (shadows). This particular model can be considered simplified because the relations, behaviors, and attributes were elementary and in most cases consisted of a single linear function with defined neighbors. While a building in the real world may have many attributes (name, address, owner, renters, histories, structural information) in this analysis the footprint location and structure height were the only concerns. In addition, these elements were derived from analysis and were not a part of the original building attribute table. The object building was bounded by LIDAR inputs and shadow casting outputs, thereby restricting any other interaction with real world elements.

Considering this analysis was very much like a case study, utilizing this model was an easy and practical way of structuring the data. Defining logical linkages was not applicable due to lack of attributes (and therefore key fields). This “flat” structure also allows for future modifications, access and organization, and facilitated quicker performance. One disadvantage was that it is a limited version of the real world. However, this has no impact on the shadow analysis and could be easily remedied by the client.

The logical model matrix (Table 4) shows how the data were organized. The Phase column represents the part of the analysis in the data were used. All project data were organized into Esri Geodatabases. As previously mentioned, there was no need for the construction of a unique relational database since there was very little attribute management, joining functions, or normalization. The geodatabase allowed for a simplified and transparent storage function that also facilitated the organization of a multitude of derived files in a fairly complex analysis. Although there were no functionality constraints that led to this approach, the reality of managing the results of multiple different methodologies during the analysis dictated a structure whereby specific data sets could be quickly identified. This was also true for the naming conventions chosen for the geodatabases and files. Geodatabases were named for the results of function performed on its contents. Feature classes (in this case individual files) were named according to the element being analyzed, and the process performed on it. Thus a feature class named “Tiles_shadow_polygon_bldg” residing in the database “Shadow Point to Polygon Bldg” is easily identified as all the building polygons in a particular tile that have just undergone the shadow point-to-polygon process. Feature classes are further identified by data type (raster or vector) in the logical matrix. Finally, the Data Entity Supported column indicates the link between the logical and conceptual models.

Table 4. Logical model matrix.

Phase	Geodatabase	Feature Class (raster/vector)	Data Entity Supported
Analysis	Study Area	Sidewalks shaded (v) Sidewalks un-shaded (v) Final shadow veg + bldg (v)	Shadow
	Shadow Points to Polygon Veg	Tile shadowpoint veg (v) Tile shadowpolygon veg (v)	Shadow
	Shadow Point to Polygon Bldg	Tiles shadowpoint bldg. (v) Tiles shadowpolygon bldg. (v)	Shadow
Processing	Veg polygon Tiles Processing	Tile clip (v) Tile aggregate (v)	Canopy
	Veg Polygon Clean	Tile veg polygon clean (v) Tile veg polygon eliminate (v)	Canopy
	Raster Calculator	Study area DSM (r)	Canopy
	Natural Neighbor DEM	Study area (r)	Canopy
	Digital Surface Model	Digital Surface Model (r)	Canopy
Digital Height Model	Digital Height Model (r)	Canopy	
Preparation	LIDAR Points Processed	Tiles veg (v) Tiles building (v)	LIDAR
	Multipoints	Tile veg 1 st , 2 nd , 4 th returns (v) Tile bare earth 1 st return (v)	LIDAR
	Point Density	Study area (v)	LIDAR
	Study Area	All LIDAR point info (v) Study area LIDAR point info (v)	LIDAR
	Buildings	Schools (v)	School
	Buildings	Building footprints (v)	Building
	Study Area	Sidewalks (v)	Sidewalk

4.3 Data Sources

All project data, including the three primary data sets (LIDAR, building footprints, and sidewalks) were provided by the client. Subsequent tree canopy footprints and all canopy and building height information were derived from the LIDAR. Unfortunately, the LIDAR data came with no metadata, only anecdotal information from the client about its origin. According to the client, the LIDAR data were gathered in 2009 using the NAD 1983 datum and the State Plane California Zone V Feet coordinate system, and this information can be verified from the LIDAR file headers. The density of LIDAR points in the study area ranges from 1.833 to 2.415 per square meter, and the average number of points per tile for the study area is 1,552,074.6 points within a size of 2000 x3000 feet. LIDAR data

is known to be highly accurate, with a typical commercial dataset having a standard vertical accuracy of 50 cm or better (Arnoff, 2005).

The building and sidewalk data provided by the city consisted of building footprints and sidewalk lines digitized in an Esri vector format from another source (possibly a high-resolution aerial image). Conversations with local GIS professionals indicated the data might be a combination of efforts by University of Redlands students and City of Redlands interns from 2006-2009. No further information as to the source or the scale at which it was digitized is available. A comparison of the data with available color infra-red images shows a relative degree of accuracy. However, locational errors do appear (Figure 4-3), especially when observing sidewalks and buildings. This may be due to the scale, source, or method from which the data were generated. However, it should be noted that the comparison of data sets with imagery is not exact, as the image may have positional errors also.



Figure 4-3: Color infrared image with overlay of sidewalks and buildings showing errors.

The client made no request for any particular standard of accuracy or error tolerance, nor did they indicate at what scale the resultant analysis data would be used, or if there were any specific cartographic or analytic requirements. A cursory data survey revealed positional errors of up to 10 feet. According to the National Map Accuracy Standards, this data set could be accurate at a scale of approximately 1:4,800 if a formal survey of 10% of the sample points were corroborated. However, no such audit was undertaken. It is perhaps more important to consider how the accuracy of the data affects the types of errors in the final analysis. The shadow volumes are likely to be more accurate than their actual locations since the digitized elements seem to be more correct in size and shape than position. Although it would have been possible to extract building footprints from LIDAR, the client preferred to use the original building footprint file to ensure consistency and integration with other projects.

The project was conducted using the NAD 1983 State Plane California V FIPS 0405 coordinate system, which is based on a Lambert Conformal Conic Projection. The building and sidewalk data were received from the client in the State Plane Coordinate System. Common practice dictates that areal analysis use equal area projections, and that remote

sensing data be used in their original projection. While the analysis was conducted in the original projection of the LIDAR data, the transformation of the LIDAR data to an equal area projection was not done. Conversations with GIS professionals supported the notion that the study area was small enough that the conversion would have a minimal impact on the final results.

4.4 Data Scrubbing and Loading

The data preparations for the GIS were straightforward and consisted of an examination, cataloging, and preparation of LIDAR data, and the creation of the initial geodatabases. The vector files required no pre-processing or cleaning. Both the LIDAR and vector data sets were reduced from city-wide coverage to the study area only, consisting of a half-mile buffer around Lugonia, McKinley, and Smiley elementary schools. LIDAR tiles were selected by a union with the school buffer, thereby retaining complete LIDAR tiles regardless of the amount of study area inside each tile. All vector data were then clipped to the LIDAR tiles (Figure 4-4). This method facilitated the LIDAR work by maintaining the tile structure and metadata. The final analysis considered only features inside the buffer zones, and not the entire LIDAR tile.

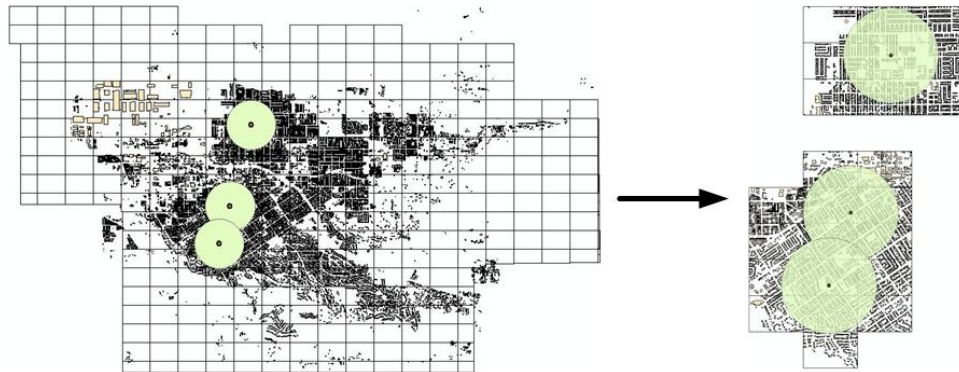


Figure 4-4: Study area selection

The LIDAR data were received from the client in 3,000 x 2,000 ft tiles. There was no significance to the tile size except consideration for processing speeds and manageable sizes. The tiles were labeled and the point spacing noted for each tile using the LIDAR Analyst extension in ArcGIS. The point spacing information was necessary for several subsequent procedures. Next, the raw LIDAR files were converted to the Esri multipoint format using the LAS to Multipoint tool. This proprietary data type allows the compression and rapid use of large point files. Since a single LIDAR tile may contain over a million points, they would be extremely cumbersome to use in a traditional database table format. Multipoints allow the grouping of several thousand points into a single database record, although the individual point data are still retained. Multipoints were processed only for the unclassified, bare earth, and vegetation point classes, since other LIDAR point classes (such as water) were not needed. Table 5 illustrates the

processing parameters from .las to multipoint. Last returns were used for unclassified, trees and buildings to eliminate power lines and other object that were not part of the tree canopy. Empty Esri geodatabases were created to store multipoint data that were named according to their tile of origin LIDAR class and return value.

Table 5. LIDAR to multipoint conversion criteria.

LIDAR Class	Point Spacing	Return	Coordinate Sys.	Z-value
Bare earth (2)	As per metatdata	1 st	Cal. S.P. V	.001
Unclassified, Trees and Buildings (1,2,4)	As per metatdata	Last	Cal. S.P. V	.001

4.5 Summary

The data design phase of the project was conducted to ensure a comprehensive understanding of the individual data sets, the relations between them, and their function in the analysis. Visual conceptual representations of the data and their relations allowed a succinct overview of the main components, ensuring the completeness of interactions and dependencies both in the analysis and in the real world. A logical model matrix cataloged data structures and their relation to the main data entities in order to examine more detailed dependencies, legacies and outcomes.

This project is slightly unusual in that much of the data used for the final analysis were derived from a single source (LIDAR). In addition, the client’s requirements included only a single final result, so there was no need to construct complex data systems supporting periphery post-analysis data integration and management. The design phase also included the preparation of data for integration into the analysis, consisting mainly of a review of the metadata, study area selection of data sets, and initial processing of LIDAR data into Esri formats.

Several important decisions and considerations marked this phase of the project that impacted on following action and analysis. Among these decisions were the acceptance of legacy data errors, projections and transformations, file naming conventions, database criteria, and fairly complex choices regarding LIDAR processing. Considerable time was spent ensuring a comprehensive understanding of these concepts and procedures to guarantee there were no negative results. The data design was revisited during the course of the project, as some unforeseen circumstances and procedures necessitated adjustments to the concepts and procedures. However, the models illustrated here represent the final version.

Chapter 5 – Implementation

The use of LIDAR data to create shadows is a unique proposition and a fairly unexamined area of GIS analysis. Determining the best practices and procedures for obtaining accurate results was a process of experimentation. Although some of the techniques used here were not new, their impact on shadow creation and accuracy was previously unknown. Implementation consisted of the development of products and a methodology to deliver the best possible building and vegetation shadows identifying shaded and unshaded sidewalks in the study area.

5.1 LIDAR Tile Processing

The first step was an examination of the LIDAR data. This consisted of inspecting the tiles for metadata and relevant attributes such as point count and point spacing, followed by the assembly of individual tiles into a study area grid (Figure 5-1).

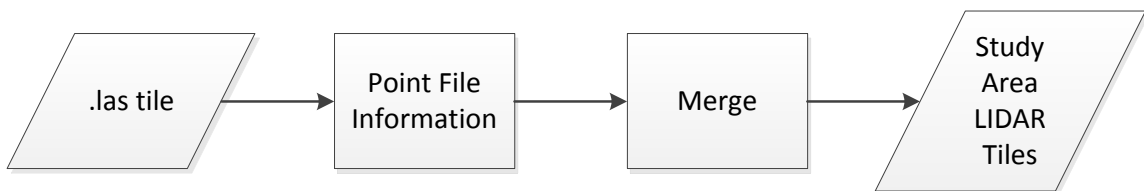


Figure 5-1: LIDAR data examination.

A thorough understanding and documentation of the raw LIDAR data enabled a better understanding of how its particular characteristics contributed to the final analysis. For example, an unusually low point count, or high point spacing measurements may signal problems with vendor processing or the actual flight parameters. Point spacing refers to the horizontal distance that separates points, while point count is the number of points in a particular tile. In this case, the point spacing ranged from 1.833 to 2.415 meters, and the point density was 1,028,467 to 1,783,994 points per tile (3,000 x 2,000 ft.). This data were generated using the Point File Information Tool, which created individual tile shapes with LIDAR attributes. Compared to standard commercial-grade LIDAR data, the files were acceptable but only of average quality. It was also useful to examine the number of LIDAR returns for each data class (unclassified, bare earth, medium vegetation, noise, contour keypoints, water, and overlap). Any unusually low percentages here may indicate processing errors; however, one must also consider the characteristics of the terrain surveyed. In addition to enabling a better understanding of the data, this information was necessary to perform the analysis.

Data area delineation using a grid allowed for a quick and convenient reference. It facilitated queries of LIDAR attributes for individual tiles or summaries for the entire study area, and a well-structured LIDAR guide enabled examinations of specific geographic areas. A grid proved useful later in the analysis for comparing results using

different methods, and was also needed for various processes that required either micro or macro area analysis. The grid was created by using the Merge tool to group tiles generated with the Point File Information Tool.

5.2 Raster and Vector Processing

Following an examination of the data, it was necessary to transform the LIDAR into a format conducive for analysis that was also compatible with the Esri software suite. This transformation used the raw LIDAR data and the building footprints to derive a tree canopy for the study area. While some of these processes are typical for the development of elevation models from LIDAR data, others were unique to this project and required some experimentation (Figure 5-2). The results described below are the final efforts that proved most suitable.

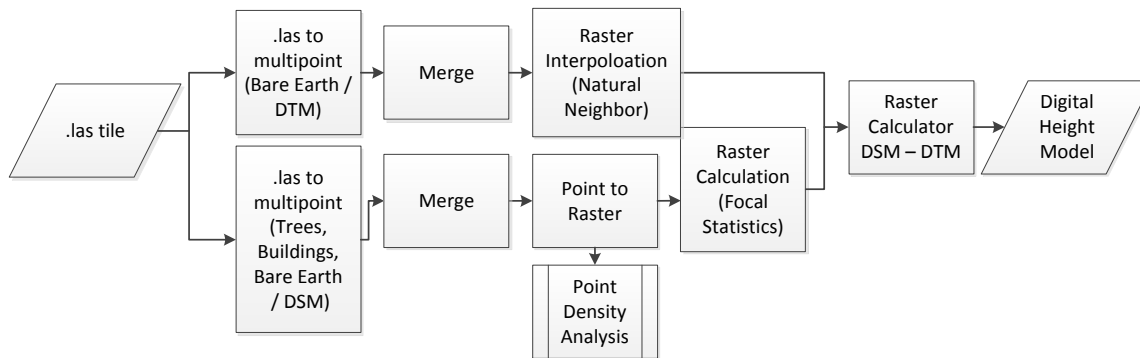


Figure 5-2: Raster and vector processing steps.

The conversion of LIDAR points to the Esri multipoint data type facilitates the manipulation of LIDAR points in a GIS. Because a single LIDAR tile in the study area may contain over a million points, it can be quite cumbersome to manage in a database due to slower processing speeds and great storage space requirements. A multipoint feature simplifies the data by occupying only one record in the database, while representing many points. Figure 5-3 illustrates this concept. Although only one record is selected in the table, there are several thousand points selected in the data frame. The data are “compressed” into a single record, but the elevation data are retained for each point.

In this case, multipoint data were processed in two batches, one for the bare earth (first return), and one combined batch using unclassified, bare earth, and vegetation (last returns). Bare earth first return data are considered the most accurate for creating Digital Terrain Models (Esri, 2009b). The combined processing of unclassified, bare and vegetation classes using last return, although not conventional, was the result of testing various configurations for best results.

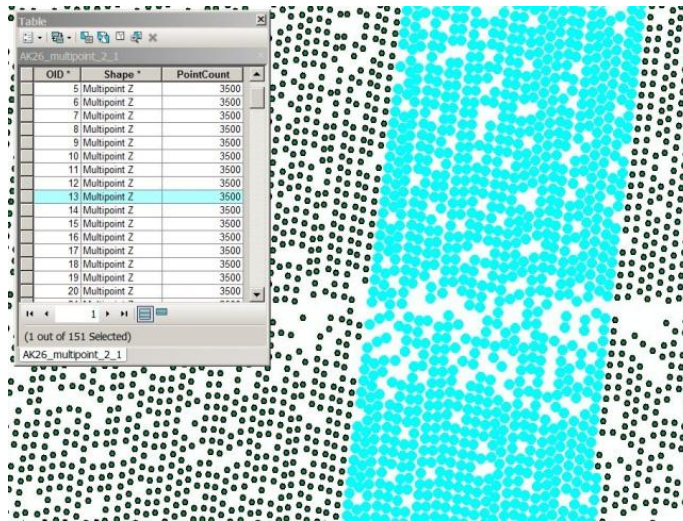


Figure 5-3: Multipoint record selection example.

First return data for tree canopy analysis is the usual choice because it produces a denser canopy with more height information. However, typical LIDAR canopy analysis usually involves large stands of unbroken vegetation, rather than individual trees, in an urban situation. In an urban environment, the presence of many objects of equal or greater heights than the canopy adds clutter to the canopy area that requires later editing. In this case, power transmission lines as well as other anomalies, appeared as tree canopy. Editing out these errors without supporting geographic data was problematic, and necessitated using other returns. Comparing different methods showed that using last return data for the tree canopy, although less dense, did not significantly affect the resultant canopy area (Figure 5-4).

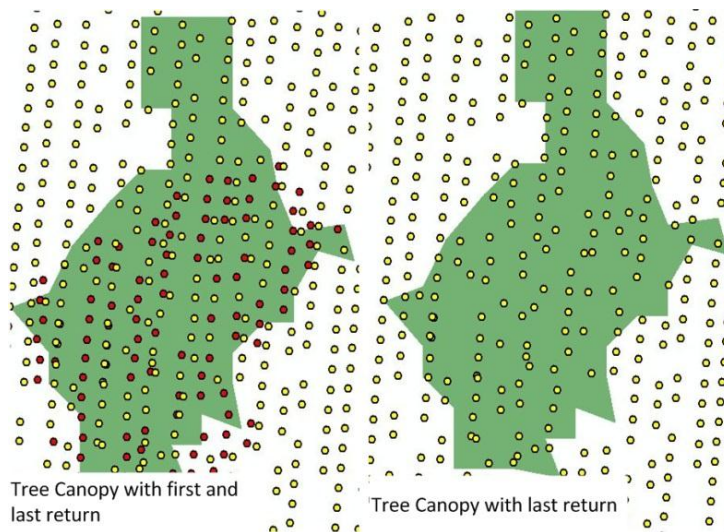


Figure 5-4: First and last return comparison.

Therefore, the last return data provided acceptable canopy information without containing extraneous objects with respect to the tree canopy, such as power lines. The combination of three LIDAR classes into one multipoint file ensured that the subsequent raster processing would capture not only the required vegetation data, but also buildings and bare earth points to create a surface model. Each set of multipoint tiles were processed and merged to create two complete multipoint study area files, one for each feature type.

At this point, the process diverged slightly to conduct different transformations on the bare earth and combined point files. Both data sets required raster transformation to produce elevation models that would allow the identification and selection of the tree canopy. There are varieties of techniques available for creating continuous surfaces from point data, but some form of interpolation was necessary because the bare earth LIDAR data contained many “holes” that would have affected the quality of the derived surface if a standard point to raster operation was used. The Natural Neighbor model is a deterministic method that uses weighted average values of nearby points in a triangulated method. This method, used on the bare earth data, preserves original values, and is more suitable for large point data sets, such as LIDAR. The raster used a cell size of 4 x 4 ft, and the value field was z. The result of the interpolation was a smooth surface of ground elevation data in raster format, as confirmed when viewed as a hillshaded surface.

Unlike the ground data, the combined unclassified, bare earth, and vegetation points demanded a slightly different method of interpolation that would also generate a continuous elevation surface. The initial step required a simple point-to-raster interpolation. This tool is appropriate for the combined data set because the coverage is far denser and the resultant number of rasters with null values are fewer. The tool parameters entered were a value field of z, a cell assignment of mean and a cell size of 4 feet. The tool functions by assigning the same height value to the cell as the points it contains. If there are several points in a cell, there are priority options available, but this project used the mean values. The cell size is approximately double the point spacing to enable the best point capture, the avoidance of empty cells, and to provide a reasonable resolution. Regardless, the resulting surface raster will still contain some empty cells with null values, as represented by the white cells in Figure 5-5a.

The null values represent raster cells where no LIDAR points were found. This is corrected by applying a Focal Statistic function to the surface that will fill the holes. Focal Statistics can use a wide array of operatives to calculate values, but this particular operation used the neighbor circle function statement (Equation 1).

$$\text{Con(IsNull("Digital Surface Model"),FocalStatistics("Digital Surface Model ",NbrCircle(2),"Mean"),"Digital Surface Model")} \quad \text{Eq. 1}$$

This statement says: when the condition of a raster cell is null in the digital surface model, apply Focal Statistics to the digital surface model using the Neighborhood Circle Method with a radius of two cells, containing the mean values of those cells, and apply the changes to the same digital surface model.

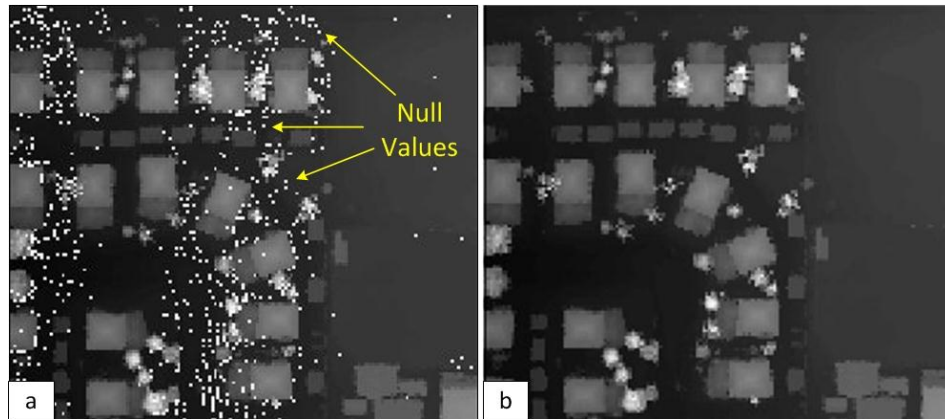


Figure 5-5: Point to raster interpolation with null values (a) and focal statistics (b).

The mean circle method calculates the averages of all cells whose centers fall within a circle of a given radius. If any null values are in the circle, ignore them. This method assumed that nearby cells would have similar values, and the small neighborhood circle would restrict the sampling size to nearby cells. The result of the process, is illustrated in Figure 5-5b, where the null values were filled and a continuous surface created.

Although not strictly required for the analysis, a point density raster was constructed. The point density surface allows an evaluation of the critical LIDAR data, in this case the layer of points containing vegetation data. This evaluation facilitates a better understanding of the LIDAR coverage and may be useful when interpreting the results. The density image (Figure 5-6) used a cell size of 8 x 8 ft and reveals an irregular patterning to the coverage represented by the darker cells in the prominent horizontal and vertical bands. These bands may be overlap boundaries from the flight path. There are also areas of contrasting point density in close proximity, notably in the northwestern quadrant of the tile. Irregular patterns like this may have an impact on the analysis.

The final step in this segment was the creation of a digital height model. The height model is the result of the difference between the combined surface models and the bare earth elevation model. Because both models contain bare earth elevation, a simple subtraction process will eliminate the bare earth and the remainder will be a raster file representing features above the ground. This step used the Raster Calculator with the expression show in Equation 2. The digital height model (DHM) is the main element of further transformation and the key component in constructing the tree canopy.

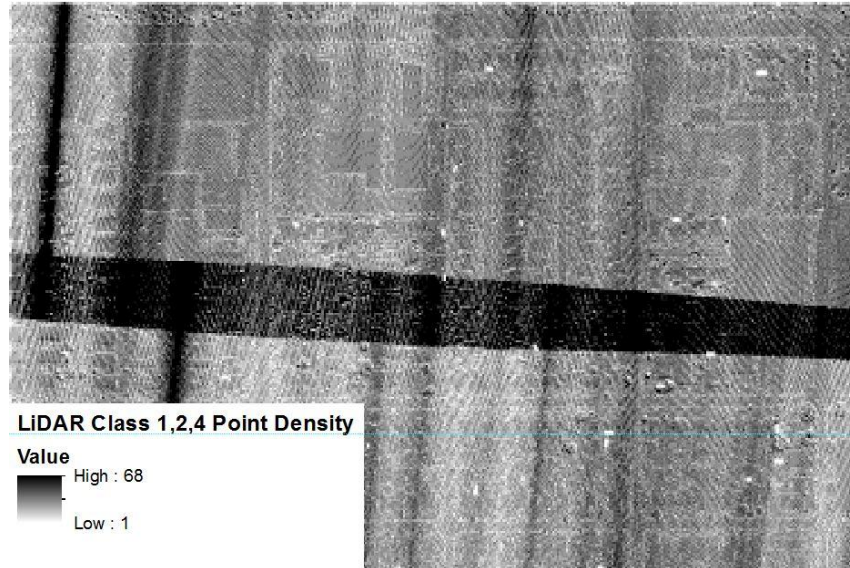


Figure 5-6: Unclassified, bare earth, and vegetation LIDAR point density.

$$\text{Digital Surface Model} - \text{Digital Elevation Model} = \text{Digital Height Model} \quad \text{Eq. 2}$$

The digital height model represents all above-ground features, both buildings and trees. Although shadow modeling includes trees and buildings, the canopy analysis is only concerned with trees. Therefore the separation of trees and buildings in the height model was required so both elements could be modeled separately and later combined. The first part of this process (Figure 5-7) is the creation of a building mask to eliminate buildings from the height model. The client’s building footprint data were used to create an 8 foot buffer around all buildings. The buffer eliminates LIDAR returns from the walls of buildings. The building footprint is a representation of the area of ground covered when viewed from directly above, so data from the sides of buildings were misleading and not needed. The buffer also removed some vegetation data in close proximity to the structures.

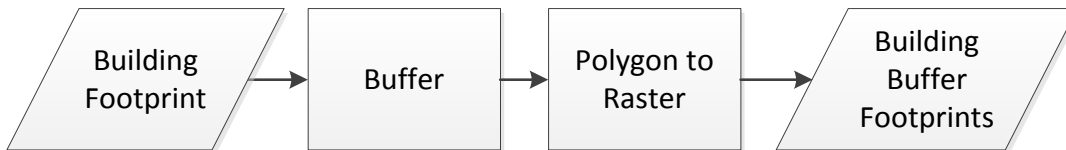


Figure 5-7: Building mask creation.

A process of experimentation with different buffer sizes showed that eight feet removed the maximum number of sidewall returns and an acceptable of vegetation. The process continued on the assumption that the vegetation data that were removed would not be detrimental to the analysis because of the blended shadows provided by both buildings and trees, that vegetation shading buildings were not part of the analysis, and

the fact that there would still be a significant canopy captured beyond the eight foot buffer.

Following the buffer process, the new building buffer file was converted to a raster format to allow for further integration. This conversion used the Polygon to Raster tool with a value field of “input date” (zero value), a cell size of four feet, the Cell Center raster assignment type, and no priority field. The input date field was used because there were no values in that field, resulting in a raster cell also with no value. The Cell Center method creates a raster for every portion of the polygon that intersect the center of the cell. Comparisons between different raster creation methods (maximum area, maximum combined area) showed that the cell center method created rasters that best matched the original buffer.

A raster file of the building buffers allowed the next set of procedures; the identification and creation of the initial tree canopy (Figure 5-8). This segment consisted of the elimination of buildings from the height model, the classification of vegetation rasters, and the transformation of the raster surface to a polygon shape. As previously mentioned, the building buffer was utilized as a mask to eliminate the building data from the height model. This operation used the Raster Calculator with a conditional statement (Equation 3) to create the vegetation raster. The conditional statement reads: when the condition of the building buffer raster file is null, replace this with a “0” value in the digital height model. Because all the raster cells of the building buffer file are null, this allows the isolation of all above-ground elements that are not buildings, and by default leaves only the vegetation rasters.

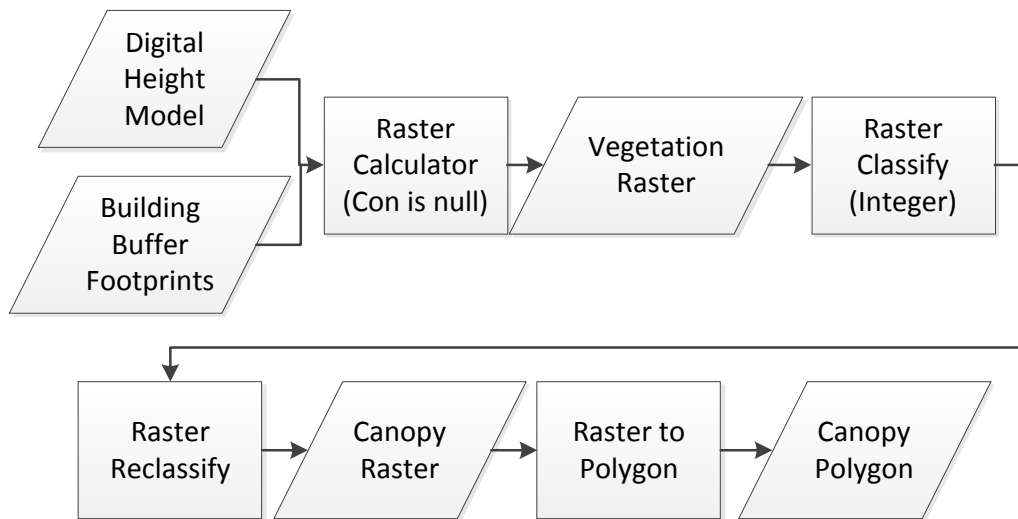


Figure 5-8: Initial tree canopy procedure.

$$Con(isNull)("building\ buffer\ raster\ file"), "digital\ height\ model", 0) \quad \text{Eq. 3}$$

Although the vegetation has been separated from the buildings, the overall data set still contains heights for low objects and ground elevations. Vegetation objects (or any object) under six feet tall provide minimal amounts of shade to humans, and can therefore be eliminated from the data set. Eliminating these data, as well as all the “0” ground values, will leave a raster data surface of only vegetation objects above six feet. Before this elimination takes place, the raster data must be converted from float to integer format using the Integer tool, since the integer attributes will complicate the upcoming transfer to polygons. Following the Integer process, the rasters are ready to be reclassified through the use of the Reclassify tool (Figure 5-9), where all values under six feet are reclassified as “NoData”. This process results in a raster data set cleaned of all values except those representing the tree canopy.

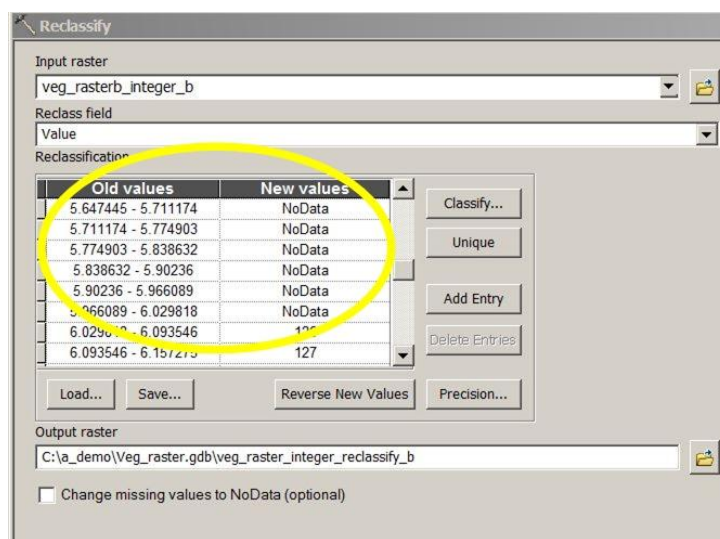


Figure 5-9: Raster reclassify of vegetation surface.

Finally, the raster surface is ready for conversion to a polygon layer using the Raster to Polygon tool. The “simplify” option was chosen for this process to eliminate the right angles of some outlying or isolated raster cells in an attempt to restore more natural contours to the polygons. At this point the polygon layer can be reclassified into vegetation heights for visual inspection, or for comparison with other data sources such as a high resolution aerial image. It is then ready for the cleaning process.

Cleaning the polygons (Figure 5-10) is necessary to prepare the data for the subsequent coverage conversion process. This will eliminate common problems such as overlapping shapes and donut polygons, and will aggregate multiple adjacent polygons into single features, all of which would otherwise cause errors when using the coverage tools. The polygons must first be clipped back to the individual LIDAR tiles from which they were originally derived. Unfortunately the Esri software will crash when attempting to clean a moderately large area with the tools used here, so the division of the tree canopy polygons into smaller areas was a necessity. In addition, the subsequent coverage

conversion and shadow casting procedures were designed to run on individual LIDAR tiles, so clipping the vegetation polygons at this point is required.

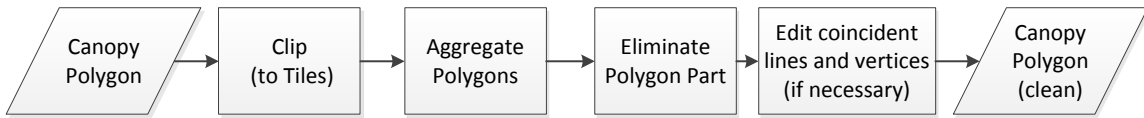


Figure 5-10: Final tree canopy procedure.

The result of the raster to polygon process creates vegetation polygons that are complex and unsuited for further analysis (Figure 5-11a). A variety of problems exist, including donut polygons, nested and overlaid polygons, and coincident lines and vertices. Each of these issues required a separate editing process. Figure 5-11b shows the result of using the Aggregate Polygon tool, which combines polygons within a specific distance of each other. A distance of eight feet was chosen as this is twice the size of the raster cell and would be sufficient to group most polygons without compromising the overall canopy shape. In addition, the option to preserve orthogonal shape was not used. This ensured that the resultant canopy would retain its natural dimensions, as recommended by Esri. Figure 5-11b also shows that this procedure eliminated the native donut polygons, but created new ones when bridging gaps left by the raster file. The Eliminate Polygon Part tool was used to correct all donut polygon problems, both native and induced. The presence of a variety of enclosed donuts dictated that the parameters be set to 90% of contained parts only. This would enforce the elimination of all interior donuts but kept smaller isolated shapes intact (Figure 5-11c).

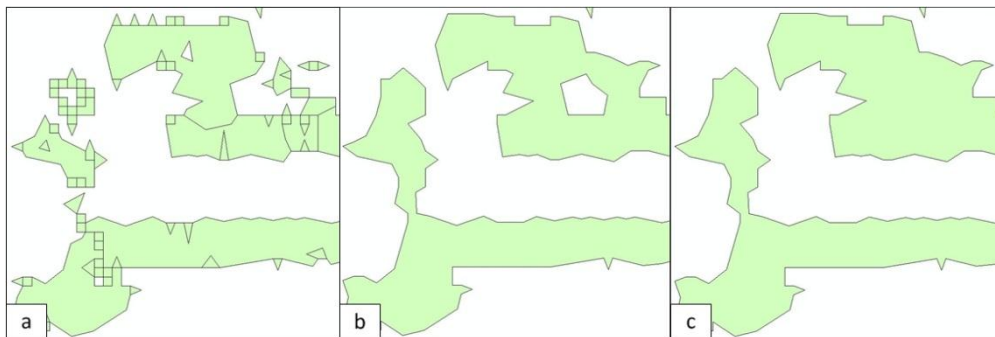


Figure 5-11: Editing raster polygons (a) with aggregation (b) and elimination (c).

Although these procedures were effective for most polygon problems, one interesting issue remained; that of polygons with coincident vertices. While they initially appear to be donut polygons, they are actually two polygons that share two or more vertices (Figure 5-12). This unusual situation is a facet of the raster to polygon

transformation and cannot be corrected by the methods described above. These polygons must be edited by hand and the vertices separated slightly to allow a small gap to occur between the formerly connected points. Although this introduces a small error into the original shapes, careful editing can create separations of less than one inch on the ground. There may also be smaller polygons concealed “underneath” larger polygons that should be deleted.

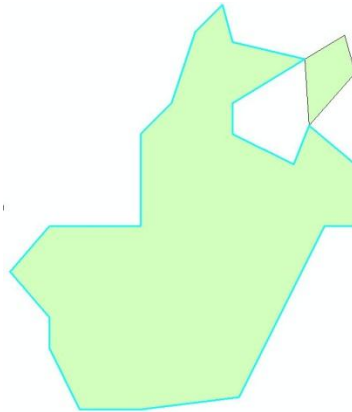


Figure 5-12: Polygons with coincident vertices.

These editing procedures were repeated for every tile in the study area. Upon completion, the polygons were suitable for the following coverage and shadow casting processes.

5.3 Shadow Coverage

The process of creating shadows was quite complex and necessitated the development of solutions outside of the Esri environment. The procedures detailed here are specifically designed to be used with the custom shadow script and also require the direct manipulation of .las format files. The premise of the procedure (Figure 5-13) is that the script produces shadow points on the ground based on a trigonometric calculation of the intersection of the polygon’s LIDAR heights, the azimuth and height of the sun, and the elevation of the ground. These points can then be aggregated into polygons that will simulate shadows on the ground (Figure 5-14).

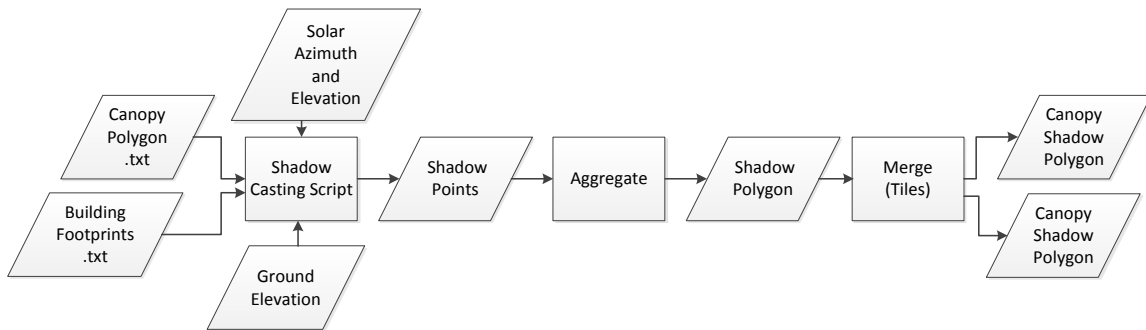


Figure 5-13: Overview of the shadow casting process.

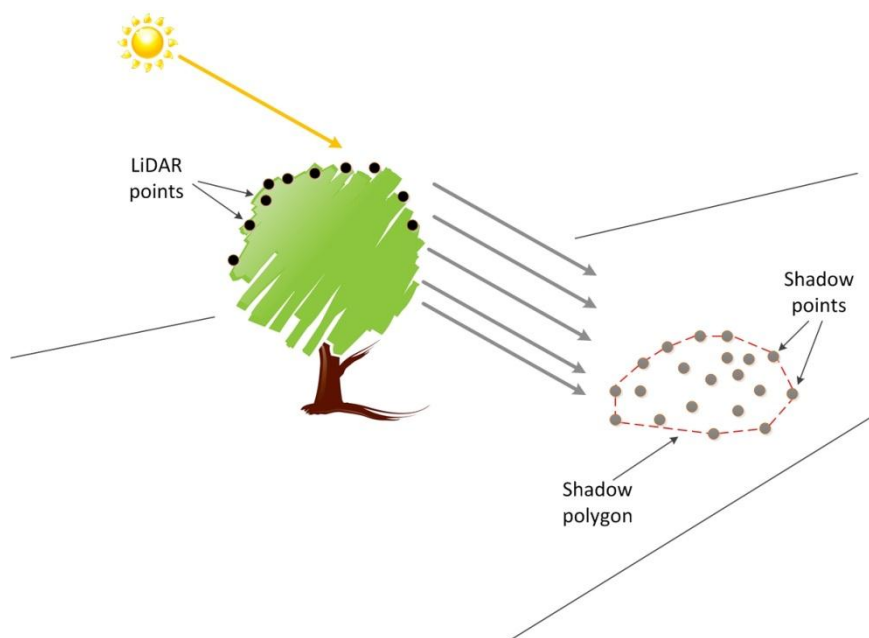


Figure 5-14: Conceptual illustration of shadow casting.

The first requirement of the process is the transformation of the previous building footprints and canopy polygons into an ASCII text file (Figure 5-15). The canopy and building files are processed separately, but follow identical procedures. The transformation of polygons to text files is necessary to facilitate compatible operations with the shadow casting script. The Feature Class to Coverage tool allows for the deconstruction of polygons into an associated collection of points, lines, polygons, and text. The tool ran with the standard default parameters. The resulting coverage was then processed through the Build tool which created polygons and topology for the coverage. Selection of polygon as the feature type ignored the other facets of the coverage, but the polygon topology ensured that the shadow tool would be able to read each polygon completely but separately and generate individual shadows. Next, the Ungenerate tool

transformed the coverage polygons to a text file consisting of map coordinates for the arcs. The parameters were set to “Poly” and the numeric format was fixed to avoid exponential representation of float-precision numbers. The resulting canopy and building files were then ready for use with the shadow casting script.

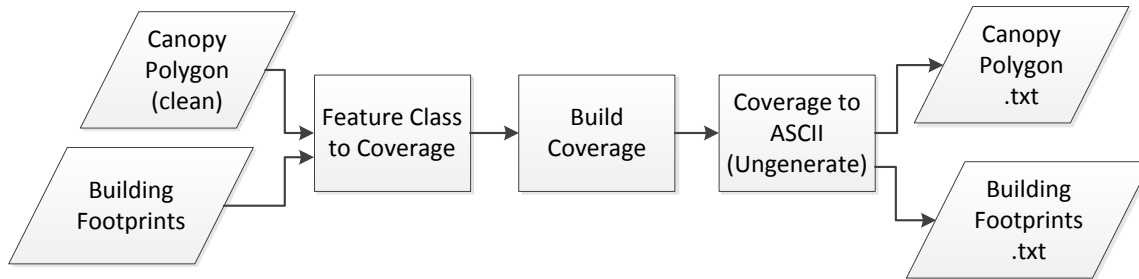


Figure 5-15: Polygon to ASCII transformation process.

The script developed to create the shadow points is composed of several different sets of internal operations, but the entire process is executed using a single ArcGIS Toolbox script. As with the polygon to ASCII operation, identical procedures were repeated on both building and canopy files. To prepare for this operation the LIDAR tile vegetation, building footprint, and ground data were extracted into three separate .las format files. This was done using a custom script that directly accessed the .las format file. The parameters of the tool could then be entered through the familiar Esri interface developed for this process (Figure 5-16).

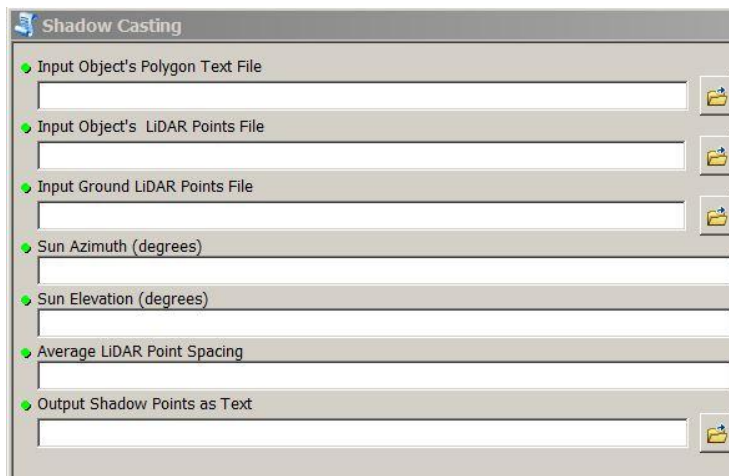


Figure 5-16: Shadow casting model parameters.

The Object Text Files are those files created in the last step during the ASCII conversion process. Either the building or canopy file may be entered. The Object's LIDAR Points file, as well as the Ground LIDAR points file, were extracted and saved in the preliminary steps of this operation, as mentioned above. The Sun Azimuth and Elevation data were specific to the date, time, and location of the shadow location and are entered in decimal degrees. The azimuth and elevation information depends on the latitude and longitude of the object. Providing coordinates for every shadow object was not practical, so the Mean Center tool was used to determine the center point and coordinates of the study area. Average point spacing was recorded in an earlier operation and was taken from the LIDAR header file. The last parameter is the location of the output text file.

The actual operations of the script deserve scrutiny, as this is one of the more crucial elements in creating the shadows. The process of running the script combines several different geographic and mathematic operations (Figure 5-17) the result of which is a table of map coordinates of shadow points. The tool initially examines the polygon file and selects the appropriate class of LIDAR points that are contained within each polygon. After extracting these, it examines the LIDAR ground points contained within the polygon. From these it calculates the average ground elevation. If there are no ground points within the polygon, it examines the entire LIDAR tile for its mean elevation. These extracted points and elevations are then used to calculate shadow locations. This requires some trigonometric operations that uses both LIDAR data, and the information just extracted from the files.

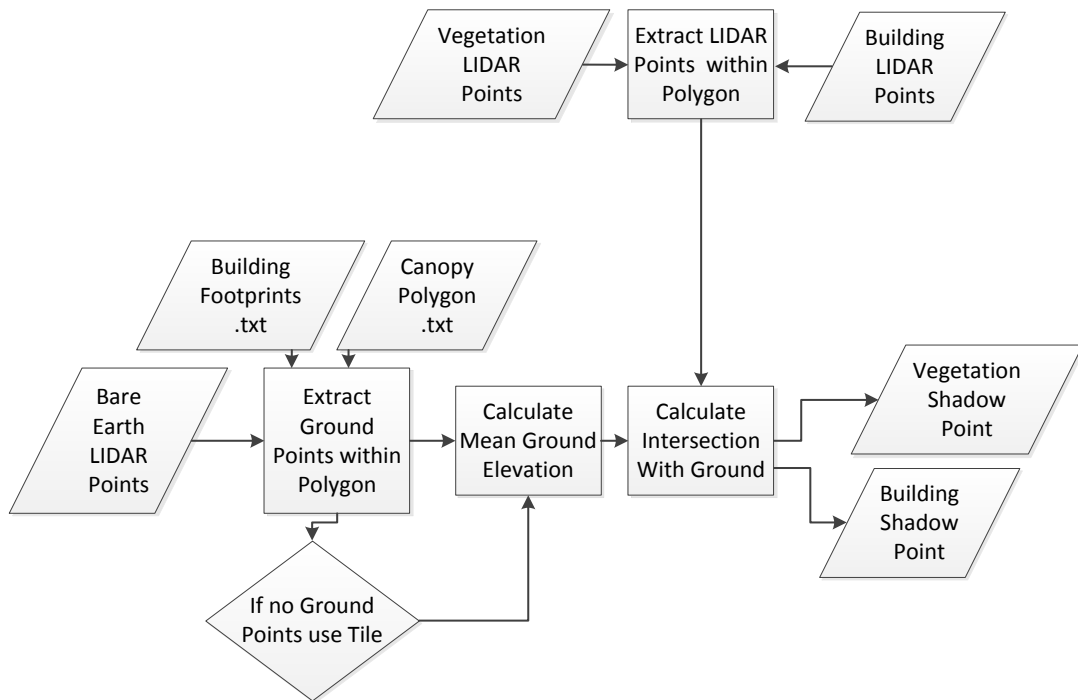


Figure 5-17: Shadow casting script operation.

The basic tenants of the Pythagorean Theorem were used to locate the shadow points based on the elevation of the LIDAR points, and the angle of the sun (Figure 5-18). Here, a right triangle was created from the LIDAR point to the ground, with side c as the hypotenuse, and b as the opposite side. The length of side a is calculated as the difference between the LIDAR point and the mean ground elevation (assuming there is no slope). Angle A is the height of the sun in degrees. The properties of sines and tangents were used to calculate the lengths of the other sides. Thus, the shadow point is located at the intersection of sides c and b . This position was noted and recorded as map coordinates in a text file. This process was repeated for every LIDAR point in every polygon in the tile and all coordinates of shadow points are written to the same text file. A typical LIDAR tile containing 3000 vegetation polygons produced approximately 40,000 - 50,000 coordinate points, depending on the size of the polygons.

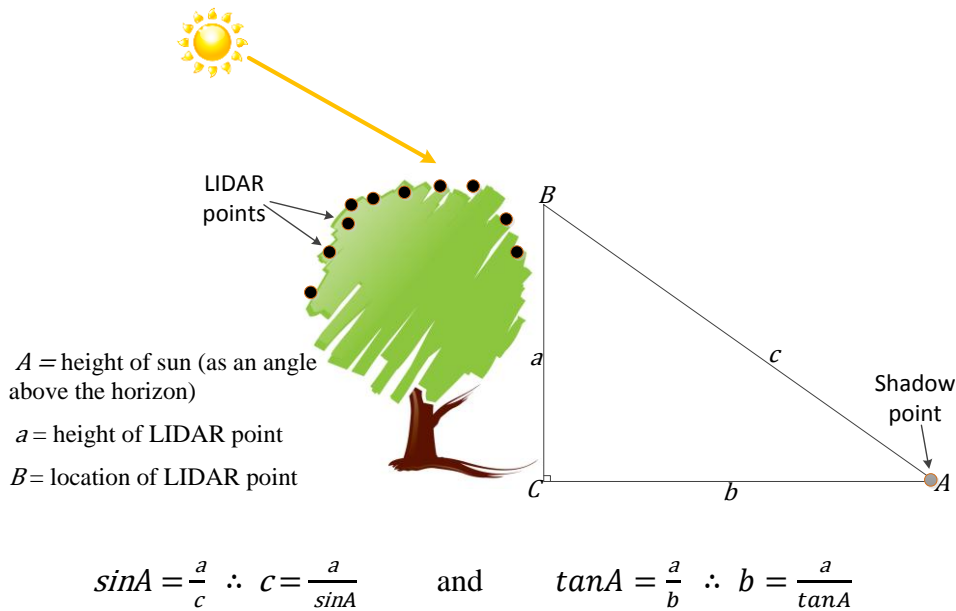


Figure 5-18: Calculation of shadow points.

When all the shadow points were processed they were then grouped into individual polygons with an aggregation distance of 10 feet using the Aggregate Point tool from ArcGIS. This distance (like the previous polygon aggregation distance of 10 feet) was the result of experimentation with different parameters. A distance of 10 feet seemed to capture the most number of proximate shadow points without creating too many artificial bridges between point clusters (Figure 5-19). This distance was determined by visually testing different parameters in conjunction with a color infrared image of the study area to derive the most accurate canopy shapes that collected proximate points.

Once the separate shadow polygons for building and vegetation were produced, all the tiles were merged by shadow type. This resulted in two files, each with a complete

shadow coverage of the study area for either building or vegetation. At this point, the data were ready for the final stages in the creation of a shadow map.

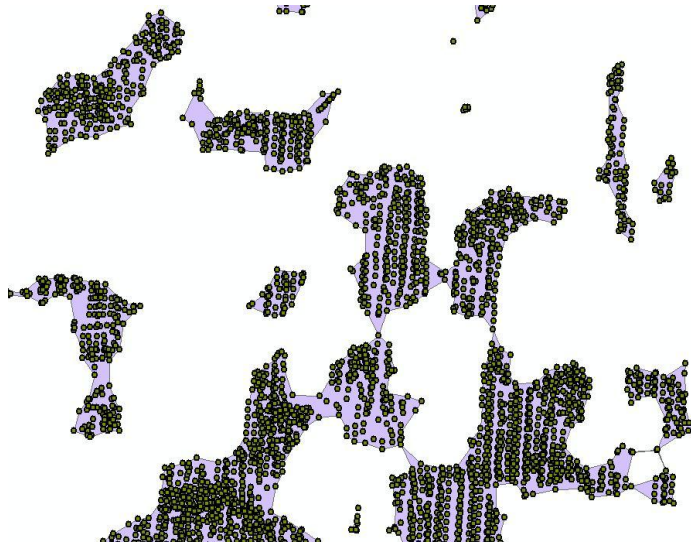


Figure 5-19: Shadow points and polygons with aggregation at ten feet.

5.4 Shadow Mapping

Both building and tree canopy shadow shape files were merged together into a single file covering the entire study area (Figure 5-20) in preparation for the next process.

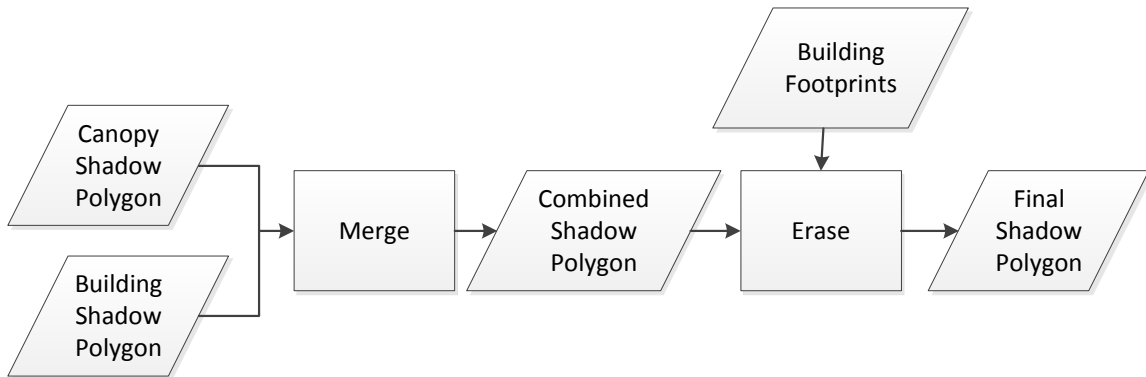


Figure 5-20: Creation of overall shadow polygons.

This combined shadow polygon file represents all building and tree canopy shadows in the study area. The shadows in the model were cast on the ground without

consideration as to the objects that were being shaded. In many cases, buildings were shaded by trees or by other buildings. Because this project only sought to analyze shaded areas for pedestrians, there must be some process of eliminating those areas that are unnecessarily shaded. In this case, it was considered that shade cast on buildings might have some benefits, but not to pedestrians walking to school. Therefore, the building footprints were used to erase shadows that were cast within those footprints. The remaining shadows would only be those that were on the ground. It was noted that some building might shade trees, or that trees will shade other trees. These shadows are beneficial to the model.

As in the real world, shadows from various objects will mingle and create extended or denser ground shadows. As long as these combined shadows are accessible to pedestrians they can remain untouched. It is important to note that if a shadow index is calculated (area covered by shadow) then the overlapping polygons must be aggregated, or the index will be too high. This occurs because two polygons covering the same area of ground will both be counted in the area calculation.

The final part of the analysis was initiated once all the shadows were created, erased, and aggregated (Figure 5-21). Because the client was interested in identifying those sidewalks that are unshaded, all shaded sidewalk areas were erased from the sidewalk file. The sidewalks remaining were those with no shadows and therefore sunny. The finished product was clipped to the ½-mile buffer around the study area schools and mapped.

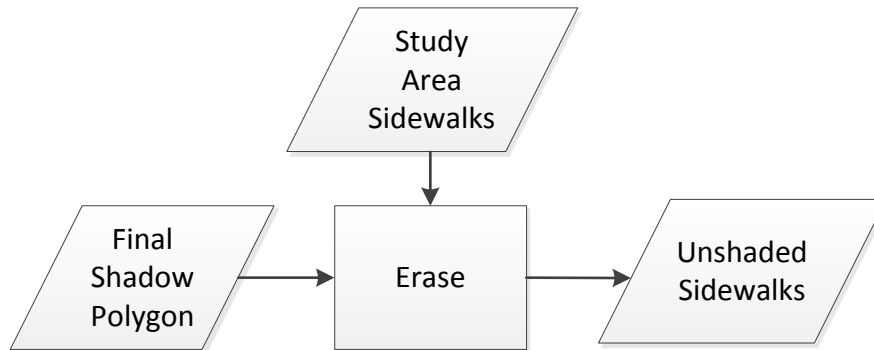


Figure 5-21: Mapping of unshaded sidewalks.

5.5 3-D Efforts

Initially the analysis planned to utilize the Shadow Tools native to ArcGIS 10. These tools are able to project shadows and create shadow footprints on the ground, which are exported to other GIS environments for further evaluation. Although the tools do provide this basic functionality, many problems were encountered during their use, and some of the key issues are outlined here. The two main problems were integration of LIDAR data into a 3D model, and the 3D Intersect operations.

ArcScene, Esri’s 3D platform for GIS, uses a proprietary shape format called “multipatch.” These are objects with 3D-like properties that represent real-world

geometric features. The standard way of integrating multipatches into a GIS is to import them from other software platforms, but they can also be created from surface data such as rasters or triangulated irregular networks (TIN). In the case of LIDAR data, it is possible to create a 3D feature, but this is best applied to bare earth surfaces that do not represent the built environment or other features. One method for interpreting LIDAR points to multipatch format is the Extrude Between tool. This tool fills the space between the LIDAR point and the ground surface with a solid shape, thereby adding substance to the difference between the point and the ground. This is the method often recommended for above-ground objects. Another method is the Random Point-Surface Information process, which samples elevation surfaces with random points before using the Extrude Between tools. This method is recommended for buildings.

While these methods initially seemed suitable, certain problems were soon evident. In the case of buildings, the application of random sampled points on the building's roof was followed by an averaging algorithm that creates a mean surface height for the top of every feature. This creates a flat roof on every building, thereby drastically altering the shape of any shadow that would be cast from it (Figure 5-22).

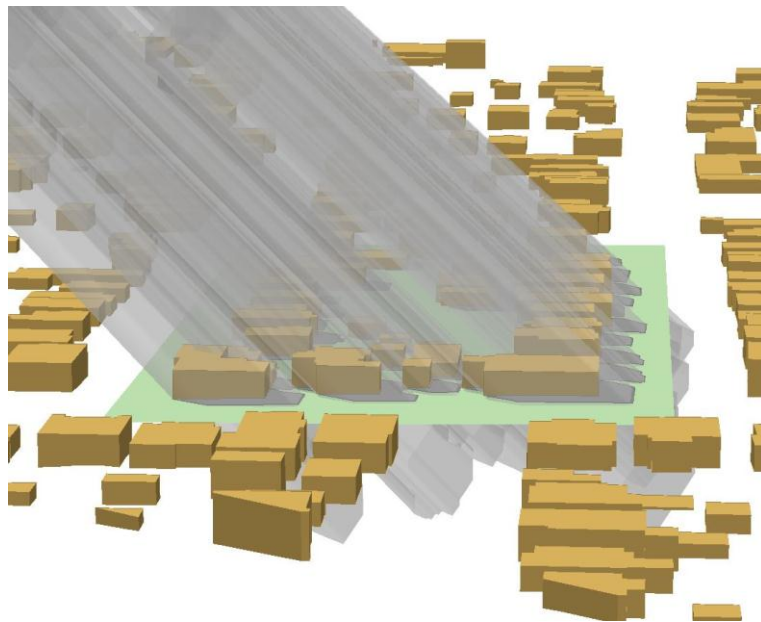


Figure 5-22: 3D buildings with flat roofs and shadows.

The extrude between procedure also is problematic when attempting to model natural above-ground features such as vegetation. With trees in particular, the extrusion of the solid ignores the negative space underneath the tree canopy and creates a solid prism that bears little resemblance to a tree (Figure 5-23). The LIDAR data only represents elevation points in space and the horizontal connection of these points, or the modeling of shapes based on these points is not quite possible with standard GIS applications. This is not necessarily a shortcoming of the software, but evidence of attempting to create complex shapes with unsuitable tools. As with the building roofs,

the subsequent 3D model in no way resembles a tree, and would create an inaccurate shadow when projected.

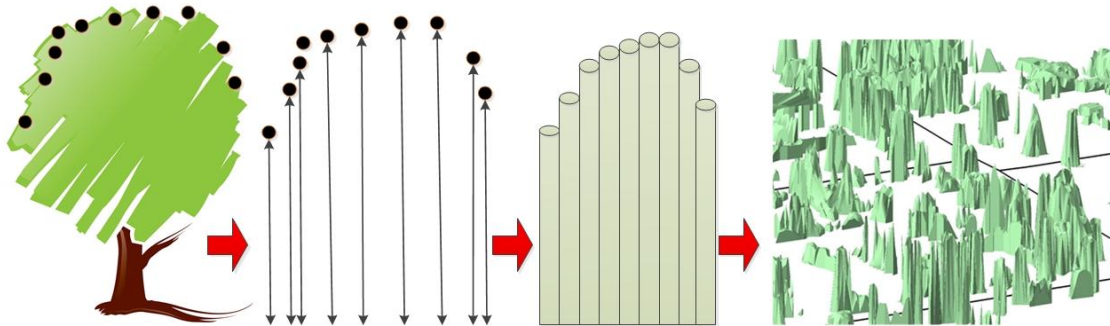


Figure 5-23: LIDAR vegetation points and the extrude between method.

The other main challenge to attempting a shadow project in ArcScene was a software problem that drastically limited the functionality needed for a large-scale analysis. In order to create shadow footprints, the intersection of the shadow and the ground must be captured and saved as a standard 2D feature using the 3D Intersect tool. These features can then be merged with other shadow footprints to create an overall shadow map. Although this is possible, it can only be done when extracting shadow footprints over a very small area. In this study, any attempt to capture shadows over an area larger than a city block resulted in the software crashing. This problem was reported to Esri and logged as a new bug (Bug NIM067835). Even if the other problems of 3D representation did not exist, performing a shadow capture operation on every city block in the study area would have been a laborious activity necessitating other solutions.

5.6 Summary

The implementation of the project consisted of the development of a series of specific operations and techniques to create and map shadows. Numerous decisions were made regarding data procedures and processing parameters that attempted to retain the accuracy of the original data, while incorporating the flexibility needed for modeling and transformations. Nearly every procedure described here underwent many variations and iterations to determine the best methods, or to ensure that standard practices were in fact producing the best results. Some of these exercises included an exploration of the limits of off-the-shelf software and resulted in the creation of custom tools and procedures.

Chapter 6 – Results and Analysis

The project was successful in determining the location and amount of shadow within the study area, and identifying unshaded sidewalks for the Safe Routes to School program. However, the measurements and maps derived from the analysis were recognized as a product of the unique methodology used, and reflect inconsistencies in the data as well as the multiple transformations performed on features. The difficulty of measuring shadows in the real world mirrors the complexity of the analysis and the problems inherent in quantifying the results or applying standards of accuracy. Although the project delivered a methodology and maps to the client to be used for future analysis, certain caveats exist when interpreting the data. The results and those caveats are detailed in the following chapter.

6.1 Vegetation and Shadow Results

The project results were divided into three main areas: the overall coverage areas, a coverage index, and an analysis of sidewalks. Each of these result areas contain important information derived from the analysis that the client may use when addressing shade-related issues. Although the client only requested an analysis for the school buffer zone, the necessary division of LIDAR data into tiles facilitated a simple expansion of the analysis to all LIDAR tiles intersecting the study area. This allowed for a quick comparison between school zones and the larger immediate area. It is important to note that all analysis areas included adjusted calculations of shadows. This indicates that the shadows were cleaned of all overlaps, and the building footprints were subtracted from the shadow areas. As explained in Chapter 5, shadows cast on buildings were assumed to be of no benefit to pedestrians, and therefore eliminated.

The table of coverage areas (Table 6) gives a broad view of the city's important features and shade characteristics. These measurements were derived from the GIS analysis previously described in Chapter 5. These findings enable a basic understanding of the study area through a comparison between the two areas. An initial survey of the LIDAR areas and the school zone showed that the LIDAR study zone is slightly more than twice as large as the school buffer zone. A comparison of other measurements shows that in general, the area covered by UTC, buildings, and shadows reflects this same ratio. It is possible to conclude that both areas have similar natural and built characteristics in terms of area coverage and that they exhibit similar shade characteristics.

Table 6. Coverage areas.

LIDAR Study Area Feature	Square Feet	School Buffer Zone	Square Feet
Study Area	143,996,194.04	School Study Area	65,686,932.00
Tree Canopy Area	33,931,621.79	Tree Canopy Area	14,062,156.19
Canopy Shadow Area	14,346,059.61	Canopy Shadow Area	6,185,900.25
Canopy Shadow Area (adjusted)	14,027,421.53	Canopy Shadow Area (adjusted)	6,079,440.63
Buildings Area	24,903,588.10	Buildings Area	11,870,776.23
Buildings Shadow Area	22,277,117.57	Buildings Shadow Area	10,639,041.68
Buildings Shadow Area (adjusted)	2,318,492.94	Buildings Shadow Area (adjusted)	1,110,495.46
Total Shadow Area	36,623,177.20	Total Shadow Area	16,824,941.47
Total Shadow Area (adjusted)	15,891,315.16	Total Shadow Area (adjusted)	6,910,249.70

It is also interesting to note that the area of land covered by tree canopy is significantly larger than that covered by buildings. However, the area of shadow provided by trees is almost seven times larger than the area shaded by buildings. While this adjusted figure is slightly misleading because of the preference given to pedestrian shade, the sidewalk shadow results discussed below further support this observation.

These measurement were then used to create a set of indexes (Table 7) allowing further analysis of the data. The index measurements were defined as a standard function of the percentage of area covered by an object. This showed that the city’s general canopy index was slightly lower than the recommended coverage of 25% for a city with a similar climate, geography, and land use (American Forests, 2009). One disturbing index comparison is that between the vegetation shadow index and the vegetation canopy. The large disparity shows that there may be issues with the methodology used for either the canopy or shadow development. Because the shadows were created with a modeled time of 2 pm on August 8, 2010, there is an expectation that the resultant shadows were not too distended and would therefore more closely match the size of the canopy. The supposition that canopy methodology problems exist was confirmed by a comparison of the building and building shadow index. These are more closely matched and meet expectations regarding the relation between objects and the size of their shadows. This apparent discrepancy may be due to data transformation and representation issues and is discussed below in Section 6.2.

Table 7. Study area index measurements.

LIDAR Study Area Index	% Coverage	School Buffer Zone	% Coverage
Vegetation Canopy Index	23.56	Vegetation Canopy Index	21.41
Vegetation Shadow Index	9.96	Vegetation Shadow Index	9.42
Building Index	17.29	Building Index	18.07
Buildings Shadow Index	15.47	Buildings Shadow Index	16.20
Total Shadow Index	25.43	Total Shadow Index	25.61
Total Shadow Index (adjusted)	11.04	Total Shadow Index (adjusted)	10.52

The sidewalk index measurements (Table 8) and mapping files are the key project deliverables and indicate the amount shade available to pedestrians after school. The coverage for sidewalks is measure by length in feet. The client provided the sidewalk data as a line file. Because sidewalk widths in Redlands usually vary from four to seven feet, creating exact sidewalk area files was beyond the scope of the project. The project worked instead with the premise that a shaded sidewalk is completely covered by shadow. The discrepancy between canopy and shadow coverage is also evident here, but to a lesser extent. The analysis shows that about 32% of the sidewalks are covered by canopy and about 17% of the sidewalks are covered by shadow at 2 pm. It is interesting to note that the canopy coverage over sidewalks exceeds the recommended minimum, whereas the general canopy coverage was below the minimum. This may be because more trees are in proximity to sidewalks, thereby increasing the canopy coverage to a as might be expected. These are predominantly residential areas where the building setback prevents any shade influence. Using the data to produce shadow maps clearly identifies the location of shadows, and the portion of sidewalks that receive no shade (Figure 6.1).

Table 8. Sidewalk index and coverage measurements.

LIDAR Study Area Feature	Length (ft)	School Buffer Zone	Length (ft)
Sidewalk	899,347.37	Sidewalk	426,956.67

LIDAR Study Area Index	% Coverage	School Buffer Zone	% Coverage
Sidewalk Vegetation Canopy Index	31.71	Sidewalk Vegetation Canopy Index	32.19
Sidewalk Vegetation Shadow Index	16.75	Sidewalk Vegetation Shadow Index	16.85
Sidewalk Buildings Shadow Index (adjusted)	.071	Sidewalk Buildings Shadow Index (adjusted)	.15
Total Sidewalk Shadow Index (adjusted)	16.99	Total Sidewalk Shadow Index (adjusted)	17.07



Figure 6.1: Shadow map of city blocks with unshaded sidewalks.

6.2 Discussion of problematic issues

The project successfully produced a methodology for creating shadows and identifying unshaded pedestrian areas, as well as measuring and mapping the amount of tree canopy and shadows. However, the results of the analysis and the methods deserve a closer inspection to better understand the products.

The issue of the difference in size between canopy and shadows is one of the obvious areas in question. The exploration of this discrepancy starts with the drastically different methods used to produce related objects. Specifically, shadows were largely generated from LIDAR points, but canopy polygons were derived from a process of LIDAR, vector, and raster manipulations. While an investigation into the accuracy of each method was beyond the scope of the project, it is useful to mention several areas where errors may have been introduced and compounded.

Modeling a UTC is a challenging task, complicated by the presence of many proximate but dissimilar objects. Trees and buildings of similar heights situated next to each other are often difficult to separate as distinct features. In addition, a UTC composed of different tree species of varying heights also adds complexity to the analysis. Figure 6.2 shows a typical Redlands canopy profile and its accompanying LIDAR data. The variety of deciduous, evergreen, and palm trees shows that in addition to its complexity, there is a strong vertical component to the canopy that isn't always captured with LIDAR.

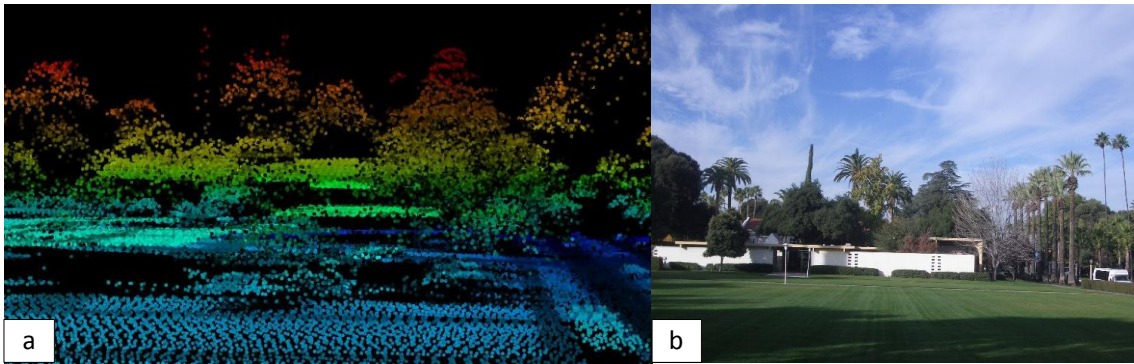


Figure 6.2: LIDAR points (a) and the Redlands tree canopy (b).

As a complex tree canopy will add to the difficulty of a LIDAR solution, the quality of the LIDAR data may also lead to estimation errors. Although the general point spacing for this data set was deemed acceptable, the classification of points as objects on the ground also deserves scrutiny. The density of classified points and the consistency of their object assignment are two important qualities when examining the canopy data. Figure 6.3 shows the points classified as vegetation by the vendor.

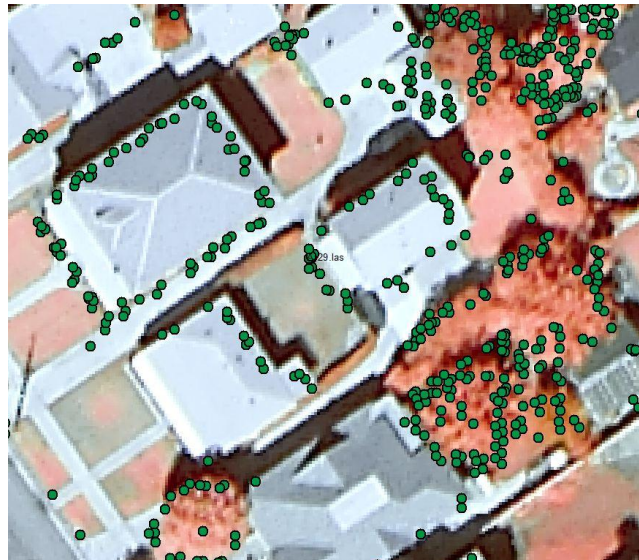


Figure 6.3: LIDAR vegetation points and color infrared image.

In this example it is clear that LIDAR points on some buildings were mistakenly classified as vegetation, and that the point density within some vegetation features was low. The methodology described in Chapter 5 attempted to address this problem by using a combination of point classifications and the masking of building footprints.

However, it is likely the analysis would be more accurate if there was a better initial classification of points.

Another potential problem with the vegetation calculations was the transformation of LIDAR points to raster cells, and then to polygons. Each of these steps will introduce errors regardless of how accurate the data are. Figure 6.4 shows the process of point to raster and how the addition of a raster grid can add area to a vegetation polygon. The original vegetation area consisted of several LIDAR points that were then sampled to a raster grid. The resultant vegetation shape is considerably larger than the actual true area. One alternative method is avoiding the raster process and creating vegetation polygons directly from the LIDAR points. An examination of this method revealed that the patchy LIDAR coverage for vegetation excluded large areas of tree canopy, resulting in errors of underestimation (Figure 6.5). In this example it is clear that the raster method tends to overestimate at the edges of the canopy, but captures canopy area toward the center. The simple aggregation method underestimates at the center and at the edges because of the “holes” in the LIDAR coverage.

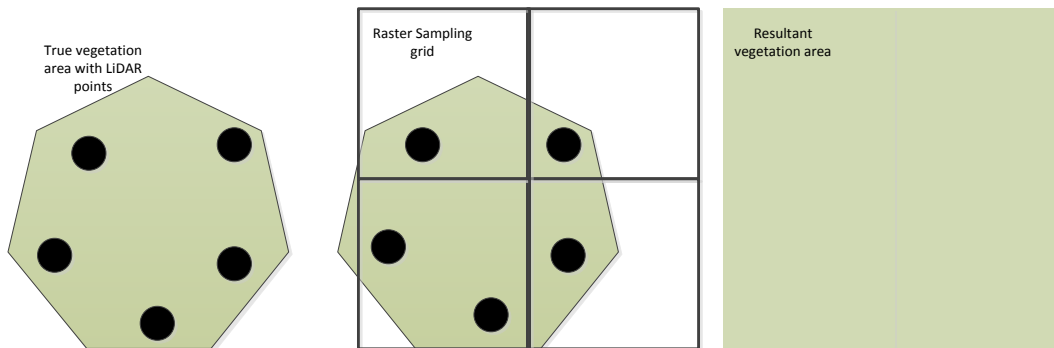


Figure 6.4: Point to raster error.

This comparison reveals one of the key problems noted earlier; the difference between the canopy area and the shadow area. Because the shadows are created directly from the LIDAR vegetation points, there is probably a tendency to underestimate the shadow size due to the inconsistent coverage of vegetation points. Although the raster-based canopy polygons may contain some degree of overestimation, the shadow polygons certainly contain a degree of underestimation. Because the raster cell size was four feet, every extra cell had the potential to increase the canopy estimation by 16 square feet.

Following the point-to-raster process there was another transformation from raster to polygon. This procedure helped reduce the general overestimation of canopy area but retained some errors. The transformation used the “simplify” method, which adjusted the raster pixels to more natural shapes, but an examination of Figure 6.6 shows the necessary compromise that re-shaped the area.

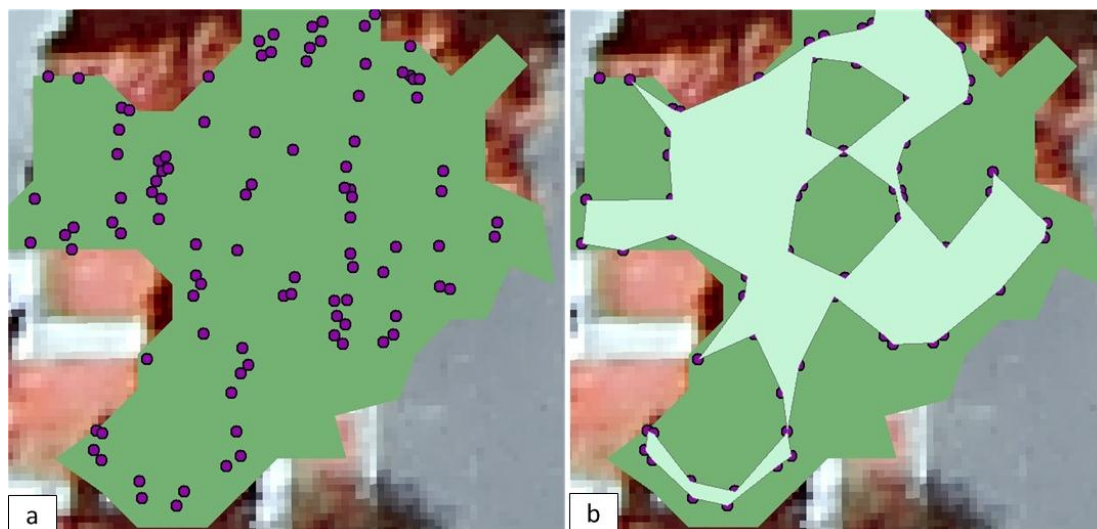


Figure 6.5: Tree canopy polygon from raster sampling (a) and point aggregation (b).

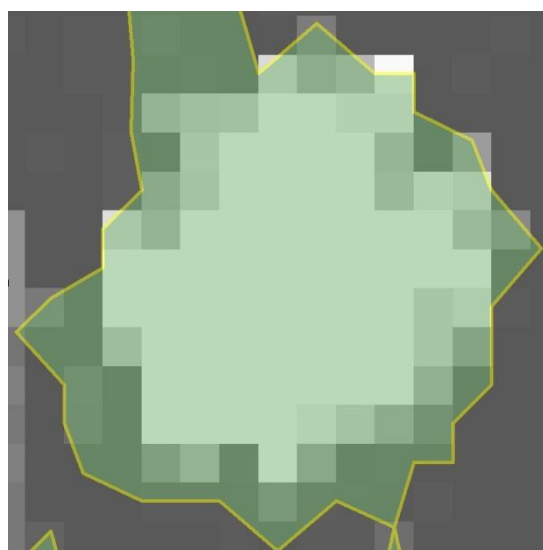


Figure 6.6: Raster to polygon process showing original raster overlaid with new polygon.

There are a few other areas where data issues, or transformations may have led to estimation errors. Among the lesser of these problems is the use of a study area mean center for solar azimuth and elevation, the possible inclusion of non-building objects over six feet into canopy measurements, and the requirement that the shadow script casts shadows on a flat surface. However, shadow point aggregation has had a greater impact on the final measurements. The methodology behind shadow point aggregation, as well as the decision to use a 10-foot aggregation distance, was explained in Chapter 5 (Figure 5-22). The creation of shadow points also involved data loss. The aggregation of points requires three points within 10 feet of each other to create and fill a convex hull. If there are only two points, or if there are several points outside the range,

there will be no aggregation, and therefore no canopy polygon created (Figure 6-7). Consequently, the shadow area is reduced because of the inability to capture areas where LIDAR coverage was less dense. Attempts to increase the aggregation distance adversely affected the study area by the creation of artificial “bridges” between independent elements of the shadows, thus creating false shadows. Naturally, this problem exacerbated a situation where there was already low vegetation classification.

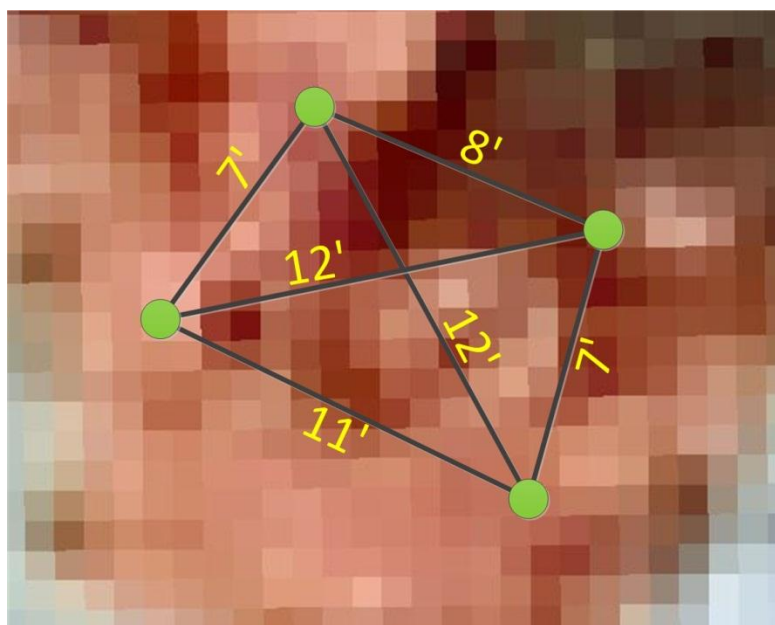


Figure 6-7: Canopy LIDAR points and aggregation distances.

Although the strengths and weaknesses of the analysis were noted and discussed, an examination of the accuracy of these methods, or a discussion of the merits of different methodologies is beyond the scope of this project. Researchers have noted that a UTC analysis using LIDAR is plagued by many small problems, and that these small problems multiplied many times over the study area can have a negative impact on the analysis (O'Neil-Dunne, 2011). A summary of the problems encountered in this analysis appear below in Table 9. However, this project met the criteria of the client's request and provided a proof-of-concept for a method of creating shadows using LIDAR data.

As previously noted in Figure 4-3, there appears to be some measure of error in the sidewalk files provided by the client. Although not uniform, there is a horizontal displacement of up to 10 feet. If these errors were corrected they may have a positive impact on the sidewalk shade index of 17%. A test of some displaced sidewalks showed an increase in shade coverage of up to 5% after correcting the displacement. However, this figure is dependent upon the accuracy of the processed tree canopy and shade polygons, whose intrinsic errors were noted above. Therefore, although calculated sidewalk shade may increase with features that are more accurate, the initial shaded areas may also contain errors that affect the overall percentage of shade.

Table 9. Summary of issues affecting measurements.

Issue	Impact on Results	Consequence
Point Classification	Negative (over estimation)	Objects will appear larger or smaller than reality.
Point to Raster	Negative (over estimation)	Irregular shaped objects will increase in size.
Point Aggregation	Negative (over estimation) / Positive (retain accuracy) Negative (over estimation)	It will hinder results that are already distorted by other processes, but can be manipulated to correct inherent errors. Sparse point coverage will inhibit aggregation due to distance parameters.
Raster to Polygon	Negative (over estimation) / Positive (mitigate estimation)	Tends to reduce the area of the raster, but this may help reduce raster errors.

6.3 Discussion of Results and the Client’s Needs

The results of the project provide the client with numerous possible avenues of exploration and action. If the premise is accepted that 17% of sidewalks are shaded after school, and that this amount is too small as per the Safe Routes to School program, the client has several options. Among these are the analysis of where to plant trees, what type of trees to plant, the expansion of the study area, and the replication of the methodology.

There are several aspects of tree planting related to the analysis that the client may consider. The results of the shade mapping exercise have delineated shade-deficit sidewalks, but there are other characteristics of place to consider. The type of tree is certainly important to the characteristic of shade. A study of the canopy and shade map show that overhanging canopy provides the best coverage and that the time of day is an important consideration when examining the interaction of canopy and shade. This study did not attempt to analyze the type of shade provided by different trees, nor the quality of the shade in terms of solid or broken coverage. Redlands has a profusion of Mexican and California Fan Palms that provide little shade but contribute to the overall sense of place. Some have pointed out that blanket arbor policies or the enhancement of the UTC are not necessarily a positive development when communities would prefer to exhibit the more natural xeriscape characteristics of the area (Groniger et.al, 2002). Therefore, the client will want to consider the types of trees. Likewise, the study of where to plant trees in itself requires many more considerations than just shade deficit (O’Neil-Dunne, 2009; Wu et. al. 2008) if the program were to encompass a city-wide effort. The client may

also want to formulate strategies based on geography, such as determining if north-south streets have less shade than east-west streets, or whether similar neighborhoods have similar shadow characteristics or needs.

The client was also interested in the feasibility of reproducing the shadow study in-house, using different study areas, dates, and times. In its current state, the results can be duplicated and the model can produce shadows with different dates and times for the study area only. However it would require dedicated resources to produce a flexible package that can quickly generate shadows for any area at any time. The main activities would be the preparation of all the data for shadow operations. This would include: LIDAR data into city-wide DTM and DSM layers, selection of LIDAR classifications from the .las tiles using custom scripts (already developed), and the preparation of all canopy and building features into polygons and ASCII files. Once this was accomplished the data would be ready, but the individual analysis would still require input into the shadow model, and any post analysis processing (such as sidewalk coverage). Some of these procedures could be aided with additional scripts and tools, but the client should be aware that a shadow analysis would take time and preparation to produce.

Chapter 7 – Conclusions and Future Work

7.1 Conclusions

The City of Redlands shade analysis identified, measured, and mapped sidewalks with insufficient shade in support of the Safe Routes to Schools program. The results were delivered to the client and enabled them to include this facet of the pedestrian environment while designing programs to enhance school routes in Redlands. The project utilized high-resolution LIDAR data that proved useful in modeling a variety of elevation surfaces and in representing features with sub-meter accuracy. In addition to the shadow mapping, the project delivered a methodology and tools for creating additional shadows, as well as information regarding the city's tree canopy and building shadows.

The analysis showed that the study area's UTC is slightly below the national recommendation. The canopy measurements involve a series of transformations and calculations using remote sensing data, and digitized vector files. The procedures were modeled on standard practices and were tested using various parameters to ensure accuracy. However, errors in the original data, as well as errors introduced during geoprocessing, did create some variance in the results. The extent of this variance and the accuracy of the final measurements are currently unknown. The methods used in the analysis were documented for the client so that they may continue to conduct shadow analysis, albeit with the knowledge that it may consume considerable time and resources.

One of the key methods for deriving shadows from LIDAR data was a script that interacted directly with the .las tiles to retrieve classification heights, trace the shadow angles using trigonometry, and create a web of points on the ground that represented shadow areas. Ray tracing has been used in other applications, including GIS, but there is no other known example of it being used in conjunction with LIDAR data to produce shadow features. Although the method is sound, the results again depend upon the integrity of the initial data sets, especially the density of the LIDAR point cloud and the correct classification of points.

7.2 Future Work

This project touched upon several areas of possible future investigation and research that would benefit shadow analysis with LIDAR data. Most of these recommendations deal with conceptual issues and methodology, as those areas presented the most problems during this analysis.

Measuring shadows is actually a common task performed as an exercise in trigonometry, as well as a method of determining a tree's height. The first accurate calculation of the Earth's circumference by Eratosthenes in 240 B.C. was determined by measuring shadows. Although hill-shading is a prevalent facet of remote sensing and GIS data, GIS platforms have only recently begun implementing dynamic shadow modeling. Technical fields such as architecture, urban planning, landscape design, and various site suitability methods all use solar exposure and shadows as an important feature of design. Some of these fields have created proprietary software that utilizes

shadow placement, but without the ability to engage in cross-cutting analysis. Perhaps with the emergence of geo-design, the development of methods for modeling shadow placement, movement, and density will enable a more realistic way of studying the interaction of shadow and the environment. If so, accompanying shadow tools that can perform a cluster analysis of shade, measure the amount of shade per feature, or the effect of intermittent shade would all be helpful to users.

The study of tree canopies usually involves either a top down (remote sensing) or a bottom up (ground survey) approach. There has been some research examining the benefits of a various fusion methods (Baller, 2008), where elements of high-resolution satellite imagery and LIDAR were used in tandem to improve tree canopy measurements. It would also be interesting to examine the combination of these techniques with ground surveying to determine how the two methods could best complement each other in other ways than the traditional ground-truthing.

Although LIDAR data are usually very dense and very precise, there are still problems in developing derived data and using them in conjunction with other GIS data, as was the case with this project. While the use of LIDAR is growing rapidly, and new methods are being developed for capturing and using LIDAR, it would also be helpful to examine ways in which to limit errors in data transformation and classification. Better classification algorithms and the ability to more accurately transform LIDAR to 3D and vector shapes would have been a benefit to this analysis.

Shadow modeling, while still a nascent element of GIS, has great potential in terms of its ability more accurately describe the interaction of features in the real world. Shadows are usually only noticed when extremes of temperature or sunlight demand their recognition. However, the ability to create these elements in a GIS would greatly enhance many aspects of geographic analysis and would certainly open the door to a host of new methods and applications.

Works Cited

American Forests. (2009). *Assessing Urban Forest Canopy Cover: A Primer for State Foresters*. Retrieved from <http://www.americanforests.org/resources/urbanforests>.

Arctur, A., & Zeiler, M. (2004). *Designing Geodatabases*. Redlands: Esri Press.

Andersen, H., McGaughey, R.J., & Reutebuch, S.E. (2005). Estimating Forest Canopy Fuel Parameters Using LIDAR Data. *Remote Sensing of Environment*, 94, 441–449.

Andersen, H., Reutebuch, S.E., & McGaughey, R.J. (2006). A Rigorous Assessment of Tree Height Measurements Obtained Using Airborne LIDAR and Conventional Field Methods. *Canadian Journal of Remote Sensing*, 32(5), 355-366. Retrieved from <http://www.treearch.fs.fed.us/pubs/25922>

Arnoff, S. (2005). *Remote Sensing for GIS Managers*, Redlands, USA: Esri Press.

Arnold, S. (2002). Redlands Street Tree Project. Unpublished manuscript, Urban and Regional Planning Department, College of Environmental Design, California State Polytechnic University, Pomona, CA, USA.

Arnold, T.F., Parker, R.C., & Evans, D.L. (2009). Precision of LIDAR Tree Measurements on Orthogonal Flight Lines. Proceedings of the SilviLaser 2009 Conference on LIDAR Remote Sensing of Vegetation Studies, College Station, TX. Retrieved from ftp://astro.rit.edu/...ForestInventory3IndividualTree/Arnold_56.pdf

Azizi, A., Najafia, A., & Sohrabia, H. (2008). Forest Canopy Density Estimates Using Satellite Images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(B8), 1127-1130.

Bartie, P., & Kumler, M.P. (2010). Route Ahead Visibility Mapping: A Method to Model How Far Ahead a Motorist May View a Designated Route. *Journal of Maps*, 2010, (84-95). Retrieved from <http://www.journalofmaps.com/crossrefMap.php?mid=1107>.

Batty, M. (2003). Agent-Based Pedestrian Modeling. In Longley, P., & Batty, M. (Eds.), *Advanced Spatial Analysis*, Redlands, USA: Esri Press.

Brandtberg, T., Warner, T., Landenberger, R., & McGraw, J. (2003). Detection and Analysis of Individual Leaf-off Tree Crowns in Small Footprint, High Sampling Density LIDAR Data from the Eastern Deciduous Forest in North America. In *Remote Sensing of Environment*, 85, 290-303.

Capuana, E., (2010). Stream Shade Patterns and Canopy Models Derived from LIDAR Data For the South Fork Nooksack River Study Area. Unpublished Manuscript, Department of Environmental Science, Western Washington University, Bellingham, WA, USA, Retrieved from

http://deptweb.wvu.edu/huxley/huxweb/gis/EGEO452/08_projects/capuana/spring_project/writeup_files/writeup.pdf

Carver, A.D., Unger, D.R., & Parks, C.L., (2004). Modeling Energy Savings from Urban Shade Trees: An Assessment of the CITYgreen Energy Conservation Module. *Environmental Management*, 34(5), 650–655. doi: 10.1007/s00267-002-7003-y

Danson, F., & Morsdorf, F., & Koetz, B. (2009). Airborne and Terrestrial Laser Scanning for Measuring Vegetation Canopy Structure. In Heritage, G., & Large, A. (Eds.) *Laser Scanning for the Environment* (pp. 201-219). Chichester, UK: Wiley-Blackwell.

Dupont, S., Ching, J., & Burian, S. (2004). Introduction Of Urban Canopy Parameterization Into MM5 To Simulate Urban Meteorology At Neighborhood Scale. Paper presented at the Symposium on Planning, Nowcasting, and Forecasting in the Urban Zone, Seattle, WA. Retrieved from <http://ams.confex.com/ams/pdfpapers/74072.pdf>.

Dwyer, M., Miller, R. (1999). Using GIS to Assess Urban Tree Canopy Benefits and Surrounding Greenspace Distributions. *Journal of Arboriculture* 28(6), 102-107.

Ebenreck, S. (1989). The Value of Trees. In Moll, G., & Ebenreck, S. (Eds.), *Shading Our Cities: A Resource Guide for Urban and Community Forests*. Washington D.C.: Island Press.

Esri. (2009a). Estimating Forest Density and Height [Web log post]. Retrieved from http://blogs.Esri.com/Dev/blogs/geoprocessing/archive/2009/03/17/LIDAR-Solutions-in-ArcGIS_5F00_part4_3A00_-Estimating-Forest-Density-and-Height.aspx

Esri. (2009b). Managing LIDAR Data in ArcGIS. Retrieved from <http://training.Esri.com/Courses/LIDARManage/index.cfm?c=262>

Esri. (2010). How Skyline Barrier Works. Retrieved from <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/00q900000014000000.htm>

Groninger, J.W., Close, D.D., Basman C.M. (2002). Can Small, Rural Communities Practice Urban Forestry? *Journal of Forestry*. 100(1), 23-28.

Hopkinson, C., Chasmerb, L.E., Zsigovicsc, G., Creedc, I.F., Sitard, M., Treitzb, P., & Mahera, R.V. (2005). Errors in LIDAR Ground Elevation and Wetland Vegetation Height Estimates. *Canadian Journal of Remote Sensing*. 31(2), 191–206.

- Indumathi, J., Burian, S.J., & Stetson, J. (2010). Estimating Urban Canopy Parameters Using Synthetic Aperture Radar Data. *The Journal of Applied Meteorology, Climatol.*, 49, 732–747. doi: 10.1175/2009JAMC2075.1
- Kim, S., Schreuder, G., McGaughey, R.J., & Andersen, H. (2008). Individual Tree Species Identification Using LIDAR Intensity Data. *Proceedings ASPRS 2008 Annual Conference, Portland, OR*. Retrieved from www.asprs.org/a/publications/proceedings/portland08/0043.pdf
- Kellern, C. (2010). Arbor Shadow Manual. Retrieved from http://www.chrisskellern.co.uk/arborshadow_site/arborshadow_main.htm 2010
- Kuo, F.E., Taylor, A.F. (2004). A Potential Natural Treatment for Attention-Deficit/Hyperactivity Disorder: Evidence From a National Study. *American Journal of Public Health*. 94(9), 1580–1586.
- Laurie, G., Reichert, M. (2010). Stream Shade Monitoring on the Klamath National Forest 2010, Retrieved from http://www.fs.usda.gov/Internet/FSE_DOCUMENTS/stelprdb5312674.pdf
- Lemonick, M. (2011). The Great Tree Survey. *National Geographic*, May 2011, 30-35.
- Ma, Ruijin. (2005). DEM Generation and Building Detection from LIDAR Data. *Photogrammetric Engineering & Remote Sensing*, 71(7), 847–854.
- Maco, S., & McPherson, G. (2002). Assessing Canopy Cover Over Streets and Sidewalks in Street Tree Populations. *Journal of Arboriculture*, 28(6), 270-276.
- McConnell, S. (1996). *Rapid Development-Taming Wild Software Schedules*. Redmond: Microsoft Press.
- McPherson, G.E., Simpson, J.R., Xiao, Qi., & Chunxia, W. (2008). Los Angeles 1-Million Tree Canopy Cover Assessment. *Gen. Tech. Rep. PSW-GTR-207 USDA USFS Pacific Southwest Research Station*. Retrieved from <http://www.treesearch.fs.fed.us/pubs/29402>
- Morrow, V., Young, J., & Roberts, C. (2001). Mapping Tree Canopy in Broward County Florida. *Proceedings of the 2001 Esri User Conference, San Diego, CA, USA*. Retrieved from <http://proceedings.Esri.com/library/userconf/proc01/professional/papers/pap273/p273.htm>
- National Association of State Foresters. (2009). *Briefing Paper: Assessing Urban Forest Canopy Cover: A Primer for State Foresters*. Retrieved from http://www.stateforesters.org/urban_forest_canopy_cover_primer

National Center for Safe Routes to School. (2011). What Kind of Buffers Can be Used to Separate Sidewalks and Pedestrians from Motor Vehicle Traffic? Retrieved from <http://guide.saferoutesinfo.org/engineering/sidewalks.cfm>

Nowak, D.J., Civerolo, K.L., Trivikrama Rao, S.T., Sistla, G., Luley, C.J., & Crane, D.E. (2000). A Modeling Study of the Impact of Urban Trees on Ozone. *Atmospheric Environment*, 34, 1601-1613. Retrieved at www.treesearch.fs.fed.us/pubs/15519

Nowak, D. J., Stein, S.M., Randler, P.B., Greenfield, E.J., Comas, S.J., Carr, M.A., & Alig, R.J. (2010). Sustaining America's Urban Trees and Forests: A Forest on the Edge Report. *General Technical Report NRS-62 USDA USFS Northern Research Station*. Retrieved from www.fs.fed.us/.../fote/reports/nrs-62_sustaining_americas_urban.pdf

O'Neil-Dunne, J. (2009). A Report on the City of Baltimore's Existing and Possible Urban Tree Canopy. Retrieved from http://www.fs.fed.us/nrs/utc/reports/UTC_Report_BACI_2007.pdf

O'Neil-Dunne, J. *Letters from the Spatial Analysis Laboratory (SAL)*. Retrieved October 3, 2011, from <http://letters-sal.blogspot.com/search/label/UTC>

O'Neil-Dunne, J. (2011). Mapping the Green Infrastructure. *LIDAR News*, Retrieved from <http://www.LIDARnews.com/content/view/8252/199/>

Patterson, P.L., Mowrer, H.T. (2003). Comparison of Ground Sampling Methods for Estimating Canopy Cover. *Forest Science*, 49(2), 235-246.

Poracsky, J., Lackner, M. (2004). Urban Forest Canopy Cover in Portland Oregon 1972-2002. Retrieved from <http://web.pdx.edu/~poracskj/Cart%20Center/psucc200404-047.pdf>

Popescu, S., Wynne, R., & Nelson, R. (2002). Estimating Plot-Level Tree Heights with LIDAR: Local Filtering with a Canopy-Based Variable Window Size. In *Computers and Electronics in Agriculture*, 37, 71-95.

Raber, B., Cannistra, J. (2005). *LIDAR Guidebook: Concepts, Project Design, and Practical Applications*. Park Ridge: Urban and Regional Information Systems Association.

Schlossberg, M., Agrawal, A.W., & Irvin, K. (2007). An Assessment of GIS-Enabled Walkability Audits. *URISA Journal*, 9(2). Retrieved from www.urisa.org/files/Journal_Vol19No2.pdf

Secord, J., Zakhor, A. (2007). Tree Detection in Aerial LIDAR and Image Data. *Geoscience and Remote Sensing Letters, IEEE*, 4(2).

Smith, S. (2003). Urban Remote Sensing: The Use of LIDAR in the Creation of Physical Urban Models. In Longley, P., & Batty, M. (Eds.), *Advanced Spatial Analysis* (Ch.9). Redlands: Esri Press.

Straub, C., & Wang, Y., & Iercan, O. (2009). Airborne Laser Scanning: Methods for Processing and Automatic Feature Extraction for Natural Artificial Objects. In Heritage, G., & Large, A. (Eds.) *Laser Scanning for the Environment* (pp. 115-122). Chichester, UK: Wiley-Blackwell.

Tomlinson, R. (2007). *Thinking About GIS: Geographic Information System Planning for Managers*. Redlands: Esri Press.

USDA, USFS. (2006). A Report on New York City's Present and Possible Urban Tree Canopy Retrieved from http://nrs.fs.fed.us/nyc/local-resources/downloads/Grove_UTC_NYC_FINAL.pdf

USDA, USFS. (2009). About the Urban Tree Canopy Assessment. Retrieved from <http://www.nrs.fs.fed.us/urban/utc/about/>

USDA, USFS. (2010). Benefits of Urban Tree Growth. Retrieved from <http://www.fs.fed.us/psw/programs/uesd/uep>

Walton, J., Nowak, D., & Greenfield, E. (2008). Assessing Urban Forest Canopy Cover Using Airborne or Satellite Imagery. *Arboriculture & Urban Forester*, 34(6), 334–340.

Weitkamp, C. (2005). LIDAR: Introduction. In Fujii, T., & Fukuchi, T. (Eds.), *Laser Remote Sensing* (pp. 1-36). Boca Raton: Taylor and Francis.

Wu, C., & Xiao, Q., & McPherson, E. (2008). A Method for Locating Potential Tree-Planting Sites In Urban Areas: A Case Study of Los Angeles, USA. [Abstract] *Urban Forestry & Urban Greening*, 7(2), 65-76.

Xu, Y., Prather, J., Hampton, H., Aumack, E., Dickson, B., & Sisk, T. (2006). Advanced Exploratory Data Analysis for Mapping Regional Canopy Cover. In *Photogrammetric Engineering & Remote Sensing*, 72(1), pp. 31-38.

Appendix A. Raster Calculator Statements

Equation 1.	Focal statistic statement.....	40
Equation 2.	Digital height model statement.....	42
Equation 3.	Building mask statement.....	43

Appendix B. Shadow Casting Script

This script was created by Ruijin Ma, Phd. using the C++ programming language.
University of Redlands, MSGIS Program
September, 2011

```
// LIDARShadow.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <fstream>

#include <iostream>

#include "dblPolygon.h"

#include <string>

#include <sstream>

using namespace std;

#define PI 3.14159265

#define max_tree_points 100000

int cal_convex(POINT2D *inPoints, int numPts, POINT2D *hullVertex);

int cal_boundary(POINT2D *inPoints, int numPts, double stepLen, POINT2D *hullVertex);

int _tmain(int argc, _TCHAR* argv[])

{

    if(argc!=8)

        return 0;
```

```

fstream treePolyF;

fstream vegLIDARF;

fstream grdLIDARF;

fstream convHF;

fstream treePtsF;

fstream shadowF;

double sunAzimuth, sunElevation;

double pointSpace;

//string argv2=std::string(argv[2]);

std::stringstream(argv[4])>>sunAzimuth;

std::stringstream(argv[5])>>sunElevation;

std::stringstream(argv[6])>>pointSpace;

//cout<<"Sun Azimuth: "<<sunAzimuth<<endl;

//cout<<"Sun Elevation: "<<sunElevation<<endl;

//cout<<"Average point spacing: "<<pointSpace<<endl;

stringstream txtLine(stringstream::in | stringstream::out);

string strLine;

int polyType, numPolys;

int numPolyPoints;

numPolys=0;

numPolyPoints=0;           //Number of vertices on a polygon

double x, y;           //To consume the label point x and y

```



```

double zDiff, xyDiff;

/*cout<<"Sun Azimuth:"<<endl;
cin>>sunAzimuth;
cout<<"Sun Elevation(not 90 degrees):"<<endl;
cin>>sunElevation;
cout<<"Average point spacing:"<<endl;
cin>>pointSpace;*/

sunAzimuth=sunAzimuth*PI/180.0;      //Radians
sunElevation=sunElevation*PI/180.0;

bool endPoly, firstLine;

endPoly=false;  //Test if the current line is the end of an polygon
firstLine=true;  //Test if the current line is the first line of a polygon, which is used to test if a
polygon has a ring

POINT2D *polyPoints;  //Store polygon points
POINT2D *shadowPts, *convexHull;  //For shadow points and convex hull of shadow
points

shadowPts=new POINT2D[max_tree_points];
convexHull=new POINT2D[max_tree_points];
polyPoints=new POINT2D[max_tree_points];

//string s1(argv[1]);
//string s2;

```

```

//s2=string(argv[1]).substr(0,s1.rfind("\\")+1)+"output1.txt";

//treePolyF.open("C:\\LIDARShadow\\Input\\polygon_text.txt",fstream::in);    //Text input
treePolyF.open(argv[1],fstream::in);    //Text input
if(treePolyF.fail())
    return 0;

//vegLIDARF.open("C:\\LIDARShadow\\Input\\points_for_shadow.las",fstream::in|fstream::binary);
y);

vegLIDARF.open(argv[2],fstream::in|fstream::binary);
if(vegLIDARF.fail())
    return 0;

//grdLIDARF.open("C:\\LIDARShadow\\Input\\ground_points.las",fstream::in|fstream::binary);
grdLIDARF.open(argv[3],fstream::in|fstream::binary);
if(grdLIDARF.fail())
    return 0;

string s1, outputfiles;

s1=string(argv[7]);
outputfiles=s1.substr(0,s1.rfind("."));

//convHF.open("C:\\LIDARShadow\\Output\\shadow_outline.txt",fstream::out);
convHF.open(argv[7],fstream::out);
if(convHF.fail())
    return 0;

//treePtsF.open("C:\\LIDARShadow\\Output\\LIDAR_points.txt",fstream::out);
treePtsF.open(outputfiles+"_LIDAR_points.txt",fstream::out);
if(treePtsF.fail())
    return 0;

```

```

//shadowF.open("C:\\LIDARShadow\\Output\\shadow_points.txt",fstream::out);

shadowF.open(outputfiles+"_shadow_points.txt",fstream::out);

if(shadowF.fail())
    return 0;

treePtsF<<"X, Y"<<endl;
treePtsF.setf(fstream::fixed);

shadowF<<"X, Y"<<endl;
shadowF.setf(fstream::fixed);

//Read LIDAR data for the ground and the points to be processed
//Read the ground points first

lasHeader12 fileHead_g, fileHead_v;
recFormat1 * lasPoints;
POINT2D testPt;

char *groundPtr, *vegPtr, *tempPtr;    //Used to handle data access
double ground_ele;
int numGroundPts;
int numTreePts;
int i;    //Indexing

numTreePts=0;
numGroundPts=0;
ground_ele=0;

```

```

grdLIDARF.read(fileHead_g.fileSig,sizeof(char)*4);
grdLIDARF.read((char*)&(fileHead_g.fileSID),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.gEncoding),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.GUID1),sizeof(unsigned long));
grdLIDARF.read((char*)&(fileHead_g.GUID2),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.GUID3),sizeof(unsigned short));
grdLIDARF.read((char*)(fileHead_g.GUID4),sizeof(unsigned char)*8);
grdLIDARF.read((char*)&(fileHead_g.vMajor),sizeof(unsigned char));
grdLIDARF.read((char*)&(fileHead_g.vMinor),sizeof(unsigned char));
grdLIDARF.read((fileHead_g.sysID),sizeof(unsigned char)*32);
grdLIDARF.read((fileHead_g.genSoft),sizeof(unsigned char)*32);
grdLIDARF.read((char*)&(fileHead_g.fcDay),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.fcYear),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.headerSize),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.dataOffset),sizeof(unsigned long));
grdLIDARF.read((char*)&(fileHead_g.numVLRecords),sizeof(unsigned long));
grdLIDARF.read((char*)&(fileHead_g.formatID),sizeof(unsigned char));
grdLIDARF.read((char*)&(fileHead_g.recordLength),sizeof(unsigned short));
grdLIDARF.read((char*)&(fileHead_g.numPRecords),sizeof(unsigned long));
grdLIDARF.read((char*)(fileHead_g.numPByReturns),sizeof(unsigned long)*5);
grdLIDARF.read((char*)&(fileHead_g.xScale),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.yScale),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.zScale),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.xOffset),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.yOffset),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.zOffset),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.xMax),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.xMin),sizeof(double));

```

```
grdLIDARF.read((char*)&(fileHead_g.yMax),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.yMin),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.zMax),sizeof(double));
grdLIDARF.read((char*)&(fileHead_g.zMin),sizeof(double));
```

```
grdLIDARF.seekg(fileHead_g.dataOffset,ios::beg);
```

```
groundPtr=new char[fileHead_g.numPRecords*fileHead_g.recordLength]; //It is only used
once for now. But it is required when multiple polygons are processed
```

```
grdLIDARF.read(groundPtr,fileHead_g.numPRecords*fileHead_g.recordLength);
```

```
//Now read the points seeking for shadow
```

```
vegLIDARF.read(fileHead_v.fileSig,sizeof(char)*4);
vegLIDARF.read((char*)&(fileHead_v.fileSID),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.gEncoding),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.GUID1),sizeof(unsigned long));
vegLIDARF.read((char*)&(fileHead_v.GUID2),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.GUID3),sizeof(unsigned short));
vegLIDARF.read((char*)(fileHead_v.GUID4),sizeof(unsigned char)*8);
vegLIDARF.read((char*)&(fileHead_v.vMajor),sizeof(unsigned char));
vegLIDARF.read((char*)&(fileHead_v.vMinor),sizeof(unsigned char));
vegLIDARF.read((fileHead_v.sysID),sizeof(unsigned char)*32);
vegLIDARF.read((fileHead_v.genSoft),sizeof(unsigned char)*32);
vegLIDARF.read((char*)&(fileHead_v.fcDay),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.fcYear),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.headerSize),sizeof(unsigned short));
```

```

vegLIDARF.read((char*)&(fileHead_v.dataOffset),sizeof(unsigned long));
vegLIDARF.read((char*)&(fileHead_v.numVLRecords),sizeof(unsigned long));
vegLIDARF.read((char*)&(fileHead_v.formatID),sizeof(unsigned char));
vegLIDARF.read((char*)&(fileHead_v.recordLength),sizeof(unsigned short));
vegLIDARF.read((char*)&(fileHead_v.numPRecords),sizeof(unsigned long));
vegLIDARF.read((char*)(fileHead_v.numPByReturns),sizeof(unsigned long)*5);
vegLIDARF.read((char*)&(fileHead_v.xScale),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.yScale),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.zScale),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.xOffset),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.yOffset),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.zOffset),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.xMax),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.xMin),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.yMax),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.yMin),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.zMax),sizeof(double));
vegLIDARF.read((char*)&(fileHead_v.zMin),sizeof(double));

vegLIDARF.seekg(fileHead_v.dataOffset,ios::beg);

vegPtr=new char[fileHead_v.numPRecords*fileHead_v.recordLength];

vegLIDARF.read(vegPtr,fileHead_v.numPRecords*fileHead_v.recordLength);

//Ready to process the data

while(!treePolyF.eof())

```

```

{
    getline(treePolyF, strLine);

    if(strLine.find("END")==string::npos)    //Not find an "END"
    {
        txtLine.clear();
        txtLine<<strLine;
        //txtLine.clear(); //The eof bit is set so it should be reset so that the txtLine can
be processed

        //int testN;
        //testN=txtLine.gcount();
        //txtLine.seekg(-testN, ios_base::end);
        if(firstLine)    //This is the first line with polygon type and label point
        {
            firstLine=false;
            numPolyPoints=0;    //This line is not a polygon vertex
            txtLine>>polyType;    //>>x>>y;    //EOF is set and it need to
be reset. Otherwise, it will not output new values since it thinks it is the end of the stream.

            if(polyType!=0) //A ring polygon and need to skip it
            {
                getline(treePolyF, strLine);
                while(strLine.find("END")==string::npos) //Not the end of
the polygon to be skipped

                {
                    getline(treePolyF, strLine);
                }
            }
        }
    }
}

```

```

//Now there is the first "END" and we skipped the ring
polygon

    endPoly=true;

    firstLine=true;

}

else //Output the remaining center point x, y
{

    txtLine>>x>>y;

}

}

else //A vertex of a polygon and need to build into a polygon
{

txtLine>>polyPoints[numPolyPoints].x>>polyPoints[numPolyPoints].y;

    numPolyPoints++;

    endPoly=false;

}

}

else //Found "END" and need to test if this is the first or second one
{

    if(endPoly) //The second "END" and the end of the polygons
    {

        break;

    }

    else //The first "END" after a series of vertices meaning the end of a
polygon

```



```

//Need to process the LIDAR points
{
    dblPolygon treePolygon(numPolyPoints,polyPoints); //Construct the
polygon from a list of vertices. It will be re-constructed everytime a new polygon is processed
    numTreePts=0;
    numGroundPts=0;
    ground_ele=0;

    //Find ground elevation first
    tempPtr=groundPtr;
    lasPoints=(recFormat1 *) tempPtr;

    for(i=0;i<(fileHead_g.numPRecords);i++)
    {
        testPt.x=(lasPoints-
>X)*(fileHead_g.xScale)+fileHead_g.xOffset;
        testPt.y=(lasPoints-
>Y)*(fileHead_g.yScale)+fileHead_g.yOffset;
        if(treePolygon.pt_in_envelop5(testPt,pointSpace))
            //The point is inside the polygon's extended envelop
            //if(treePolygon.pt_in_poly(testPt))
            //The point is inside the polygon
            { //Calculate the ground elevation
                ground_ele=((ground_ele*numGroundPts)+((lasPoints-
>Z)*fileHead_g.zScale))/(numGroundPts+1); //Z offset is not considered since only the z difference
is needed
                numGroundPts++;
            }
    }
}

```

```

    }

    tempPtr+=fileHead_g.recordLength;
    lasPoints=(recFormat1 *)tempPtr;
}

if(numGroundPts==0) //No ground points
    ground_ele=(fileHead_g.zMin+fileHead_g.zMax)/2; //Use an
average elevation

//Extract LIDAR points and calculate shadows
tempPtr=vegPtr;
lasPoints=(recFormat1 *) tempPtr;

for(i=0;i<(fileHead_v.numPRecords);i++)
{
    testPt.x=(lasPoints-
>X)*(fileHead_v.xScale)+fileHead_v.xOffset;
    testPt.y=(lasPoints-
>Y)*(fileHead_v.yScale)+fileHead_v.yOffset;
    if(treePolygon.pt_in_poly(testPt)) //The point is inside the
polygon

    { //Calculate the shadow point

        //Write out LIDAR points
        treePtsF<<testPt.x<<"", "<<testPt.y<<endl;

```

```

        zDiff=ground_ele-(lasPoints->Z)*fileHead_v.zScale;

//The offset is not add because the ground elevation was dealt in the same way.

        xyDiff=zDiff/tan(sunElevation);

shadowPts[numTreePts].x=testPt.x+xyDiff*sin(sunAzimuth);

shadowPts[numTreePts].y=testPt.y+xyDiff*cos(sunAzimuth);

        shadowF<<shadowPts[numTreePts].x<<"",
" <<shadowPts[numTreePts].y<<endl;

        numTreePts++;

    }

    tempPtr+=fileHead_v.recordLength;
    lasPoints=(recFormat1 *)tempPtr;

}

//Calculate the convex hull of the shadow points
int numVertex;
if(numTreePts==0)
{
    endPoly=true;
    firstLine=true;
    numPolys++;

    //cout<<"Polygon: "<<numPolys<<"  Number of LIDAR
points: "<<numTreePts<<endl;

    continue;

```

```

    }
else //Find and write out the convex hull points
{
    numVertex=cal_convex(shadowPts,numTreePts, convexHull);
    //numVertex=cal_boundary(shadowPts,numTreePts,
pointSpace*3, convexHull);

    convHF.setf(fstream::fixed);
    convHF<<"0 " <<convexHull[0].x<<"
"<<convexHull[0].y<<endl;

    for(i=0;i<numVertex;i++)
        convHF<<convexHull[i].x<<"
"<<convexHull[i].y<<endl;

    convHF<<convexHull[0].x<<" "<<convexHull[0].y<<endl;

    //Close the polygon

    convHF<<"END"<<endl;
}

endPoly=true;
firstLine=true;
numPolys++;

//cout<<"Polygon: " <<numPolys<<"   Number of LIDAR points:
"<<numTreePts<<endl;

}
}

```

```

    }

    convHF<<"END"<<endl; //Write out the last "END";

    delete[] polyPoints;
    delete[] vegPtr;
    delete[] groundPtr;
    delete[] shadowPts;
    delete[]convexHull;

    vegLIDARF.close();
    convHF.close();
    grdLIDARF.close();
    treePolyF.close();
    treePtsF.close();
    shadowF.close();

    return 0;
}

int cal_convex(POINT2D *inPoints, int numPts, POINT2D *hullVertex)
{
    if(numPts<3)
        return 0;
    if(numPts==3)
        return 3; //The hull is not neseccary clockwise
}

```

```

int numHullPts, ptsIndex;

numHullPts=0;

double baseX, baseY, minX;
double vectX, vectY, vectX1, vectY1;
double minAng, testAng;
double innerPro, dist1, dist2;
int *checkIndex=new int[numPts];

minX=inPoints->x;
ptsIndex=0;
checkIndex[0]=0;
minAng=180;

//Find the first most-left point and it will be one of the convex hull points
int i, j;
for(i=1;i<numPts;i++)
{
    if(minX<inPoints[i].x)
    {
        minX=inPoints[i].x;
        ptsIndex=i;
    }

    checkIndex[i]=0;//Set up the check index
}

```

```

hullVertex[0].x=inPoints[ptsIndex].x;
hullVertex[0].y=inPoints[ptsIndex].y;
checkIndex[ptsIndex]=1;
numHullPts=1;

//Find the other vertices

baseX=inPoints[ptsIndex].x;
baseY=inPoints[ptsIndex].y;

vectX=0; vectY=1.0; //Represents a starting vertical line

for(i=0;i<numPts;i++)
{
    minAng=180;    //Used to find the minmum angle
    dist1=sqrt(vectX*vectX+vectY*vectY);

    for(j=0;j<numPts;j++)
    {
        vectX1=inPoints[j].x-baseX;
        vectY1=inPoints[j].y-baseY;

        dist2=sqrt(vectX1*vectX1+vectY1*vectY1);

        if(dist2==0)    //The same point as the previous convex hull vertex
            continue;
    }
}

```

```

innerPro=vectX*vectX1+vectY*vectY1;

testAng=acos(innerPro/(dist1*dist2));    //Value is between 0 to 180 using
acos

testAng=testAng*180.0/PI;

//This also assumes there is no identical points in the point data. This can be
cleaned when time permitting

if(minAng>testAng)
{
    minAng=testAng;
    ptsIndex=j;
}

}

if(checkIndex[ptsIndex]==0)    //A new vertex detected
{
    hullVertex[numHullPts].x=inPoints[ptsIndex].x;
    hullVertex[numHullPts].y=inPoints[ptsIndex].y;

    vectX=inPoints[ptsIndex].x-baseX;
    vectY=inPoints[ptsIndex].y-baseY;

    baseX=inPoints[ptsIndex].x;
    baseY=inPoints[ptsIndex].y;

    checkIndex[ptsIndex]=1;

    numHullPts++;
}

```



```

        else //Stop checking, a complete convex hull is found
        {
            break;
        }
    }

    return numHullPts;
}

int cal_boundary(POINT2D *inPoints, int numPts, double stepLen, POINT2D *hullVertex)
{
    if(numPts<3)
        return 0;
    if(numPts==3)
        return 3; //The hull is not necessary clockwise

    int numHullPts, ptsIndex;

    numHullPts=0;

    double baseX, baseY, minX;
    double vectX, vectY, vectX1, vectY1;
    double minAng, testAng;
    double innerPro, dist1, dist2;
    int *checkIndex=new int[numPts];

```

```

minX=inPoints->x;

ptsIndex=0;

checkIndex[0]=0;

minAng=360;

//Find the first most-left point and it will be one of the convex hull points

int i, j;

for(i=1;i<numPts;i++)

{

    if(minX<inPoints[i].x)

    {

        minX=inPoints[i].x;

        ptsIndex=i;

    }

    checkIndex[i]=0;//Set up the check index

}

hullVertex[0].x=inPoints[ptsIndex].x;

hullVertex[0].y=inPoints[ptsIndex].y;

checkIndex[ptsIndex]=1;

numHullPts=1;

//Find the other vertices

baseX=inPoints[ptsIndex].x;

baseY=inPoints[ptsIndex].y;

```

```

vectX=0; vectY=1.0; //Represents a starting vertical line

for(i=0;i<numPts;i++)
{
    minAng=360;    //Used to find the minmum angle
    dist1=sqrt(vectX*vectX+vectY*vectY);

    for(j=0;j<numPts;j++)
    {

        vectX1=inPoints[j].x-baseX;
        vectY1=inPoints[j].y-baseY;

        if(abs(vectX1)>stepLen) //Outside the searching scope
            continue;
        if(abs(vectY1)>stepLen) //Outside the searching scope
            continue;

        dist2=sqrt(vectX1*vectX1+vectY1*vectY1);

        if(dist2>stepLen) //Outside the searching scope
            continue;

        if(dist2==0)    //The same point as the previous convex hull vertex (current
base point)
            continue;

```

```

//innerPro=vectX*vectX1+vectY*vectY1;

//testAng=acos(innerPro/(dist1*dist2)); //Value is between 0 to 180 using
acos

testAng=atan2(vectX1,vectY1)-atan2(vectX,vectY);
testAng=testAng*180.0/PI;

//This also assumes there is no identical points in the point data. This can be
cleaned when time permitting

if(minAng>testAng)
{
    minAng=testAng;
    ptsIndex=j;
}

}

if(checkIndex[ptsIndex]==0) //A new vertex detected
{
    hullVertex[numHullPts].x=inPoints[ptsIndex].x;
    hullVertex[numHullPts].y=inPoints[ptsIndex].y;

//The search is different from convex hull search. When a new point is found,
the base vector is reversed to be the new point to the base point instead of from base point to the new point
in the convex hull search.

vectX=baseX-inPoints[ptsIndex].x;
vectY=baseY-inPoints[ptsIndex].y;

```

```
        baseX=inPoints[ptsIndex].x;
        baseY=inPoints[ptsIndex].y;

        checkIndex[ptsIndex]=1;
        numHullPts++;
    }
    else //Stop checking, a complete convex hull is found
    {
        break;
    }
}

return numHullPts;
}
```