

AN ABSTRACT OF THE THESIS OF

Jacob E. Gjesdahl for the degree of Master of Science in Applied Economics presented on June 13, 2014

Title: Improved Viewshed Modelling for Estimating Effects of Neighboring Land Uses on Property Values.

Abstract approved:

Randall S. Rosenberger

The effect of land uses on neighboring property values is of critical interest to urban planners, developers, and citizens. Property values are in part defined by built structure and lot characteristics, locational attributes (e.g., proximity to amenities and disamenities), and other features such as views from a property. The effects of these attributes can be estimated using the hedonic property valuation method. While many of these attributes are easily quantified, views are less tangible. This study attempts to develop a viewshed estimation technique that takes into account the limbing up of trees, a major limitation of viewshed estimates to date. The limbing up effect was estimated using a conditional statement on three viewsheds: ground elevation only, ground elevation plus 8 feet where there are trees, and the elevation of vegetation (LIDAR). The viewsheds areas were intersected with their land use zone (residential, institutional, agricultural, etc) as well as their land use category (pavement, building, short, or tall vegetation) and perspective was controlled for. The viewshed estimates were evaluated within econometric models, yielding some interesting results, including that increased visibility of houses and pavement reduced property values, short vegetation had a positive effect, and views of trees themselves had no significant effect. While the techniques used in this study

have great potential, they were limited in their current application due to computing power. .

Further investigations may help overcome some of the constraints within this study and explore variations on this technique in order to generate robust viewshed estimates.

© Copyright June 13, 2014 by Jacob E. Gjesdahl

All Rights Reserved

Improved Viewshed Modelling for Estimating Effects of Neighboring Land Uses on Property Values

by

Jacob E. Gjesdahl

A THESIS

submitted to

Oregon State University

in partial fulfilment of
the requirements for the
degree of
Master of Science

Presented June 13, 2014

Commencement June 2015

Master of Science thesis of Jacob E. Gjesdahl presented on June 13, 2014

APPROVED:

Major Professor, representing Applied Economics

Head of the Department of Applied Economics

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Jacob E. Gjesdahl, Author

ACKNOWLEDGEMENTS

I'd like to thank first and foremost Jesus Christ, whose teachings of serenity and love have helped me through many difficult parts of my life. Second, I'd like to thank my parents, without whose love and support I would have never been able to accomplish this or much of anything else. My utmost gratitude goes to Randy and Jeff, who provided me with financial support and research advice and to who gave me the invaluable advice that I needed a simple model to compare this to. Thanks also to Leslie who agreed to be my GCR on such short notice. I also thank the Applied Economics department which admitted me and gave me financial aid. Finally, I'd like to thank Aniko, Karl, Bo, Kevin, and Eric for their emotional support and advice.

TABLE OF CONTENTS

	<u>Page</u>
Introduction	1
Literature Review	3
<i>Core Theory of the Hedonic Method</i>	<i>3</i>
<i>Theory Concerning Functional Form</i>	<i>6</i>
<i>Overview of Empirical Studies</i>	<i>8</i>
Method	13
Results	21
Conclusion	33
Works Cited	37

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
Figure 1: Isoprofit curve and 2 bid curves.....	5
Figure 2: Isoprofit curve and 2 producers' offer curves.....	6
Figure 3: Components used to build viewshed estimate	14

LIST OF TABLES

<u>Table</u>	<u>Page</u>
Table 1: Broad Classifications of Land Uses.....	16
Table 2: Means and Standard Deviations of Variables (N=566)	22
Table 3: Log-linear Regression Results (N=566).....	24
Table 4: Log-log Regression Results (N=566)	25

Introduction

When an agent other than the buyer or seller has an impact on the value of a good, the impact is known as an externality. Different land uses intuitively have positive and negative externalities associated with them, such as the effects of forests or factories on property values, respectively. Many of these externalities are borne by people who live close to a land use, particularly those with a view of the land use from their home. People who visit or drive by an amenity but do not pay for it (or do not get paid for having to experience a disamenity) also experience externalities, but this study only estimates values for property owners.

The issue of property value externalities is important for city planning. In addition to laying out cities for traffic flow, parking, or public transit, planners are concerned with the effect of land uses on neighboring property owners. For this reason, industrial buildings are not scattered through residential areas but parks are. Although general principles of which land uses are amenities and which are disamenities can be determined through subjective surveys, quantitative guidelines are needed if particular planning decisions are to be determined by more than the subjective and imprecise whims of political opinion. For example, exactly how much space to devote to parks and whether to allow an apartment complex on a particular block are decisions which would benefit from precise estimates of the impact on neighboring properties. This study is a proof of concept for three methods which may improve the predictive power of viewsheds (what is visible from a house) in order to better estimate externalities.

In fact, with enough precision, the current system of zoning could be replaced with a much more fine-tuned one. For example, if a developer wished to build multi-family residential in a single family zone (i.e. they would like a zoning change for a certain area), they would simply

input the height, footprint, and occupancy of their building into a model which would estimate the impact to neighbors' views and traffic. The developer would have to pay neighbors for their loss or not construct the building. This approach is a pseudo-Coasian rather than "command and control" attitude towards land use planning, with the main difference from a pure Coasian solution in that the neighbors are told what the average person would need for compensation rather than directly negotiating with the developer over their compensation. This is necessary due to the huge transaction costs that would be accrued by the developer negotiating with the owners of the hundreds of properties whose view or traffic would be impacted by the development. Similarly, cities could check every block to see whether a land parcel would add more value to surrounding properties if turned into a park than the value of the current land use.

Hedonic studies have attempted to improve the precision of estimates of the effects of neighboring land-uses on property values for decades. Most studies that have estimated the value of amenities or disamenities conferred by certain land uses to nearby homes have used simple linear distance to measure proximity, largely due to the lack of technology that would enable other methods. In the past 15 years however, more studies have begun to use road distance and viewsheds (what can be seen from a house) to create estimates. These methods appear to offer more theoretically sound measures for amenities, since road distance correlates with the ease of getting to an amenity to recreate, and viewsheds capture the more passive, aesthetic use of the amenity. Linear distance, on the other hand, seems more appropriate for disamenities such as noise or smell, which spread somewhat evenly from their source.

This study seeks to refine these measures of amenity value further by using driving time to amenities instead of road distance, as well as a more accurate viewshed. Previous studies have created viewsheds which either ignore the obstruction of vegetation or treat trees as if they are solid, with no limbing up at the bottom. This model treats shrubs and buildings as if they are solid obstructions but trees as if they are trimmed for visibility from 0 to 8 feet above the ground, giving a more accurate prediction of house prices than a model that treats trees as solid objects. Hopefully, this study will serve as another step on the path of improving the precision of hedonic property modeling, which could eventually lead to much more dynamic and precise city planning that takes into account the actual economic impacts of each proposed project.

Literature Review

Core Theory of the Hedonic Property Method

The following section draws heavily from Sherwin Rosen's seminal article on the Hedonic Property Method (1974). Although other studies had used econometrics to determine implicit prices for amenities using housing prices before Rosen (described below), Rosen was the first to lay out the formal microeconomic rationale for the technique.

A good's price may be seen as a function of that good's attributes: "The class of goods under consideration is described by n objectively measured characteristics. Thus, any location on the plane, is represented by a vector of coordinates $z = (z_1, z_2, \dots, z_n)$, with z_i measuring the amount of the i th characteristic contained in each good." (Rosen, 1974: 35). This theory assumes that there are combinations of z_i that are ideal for every consumer given their utility functions and budget constraint. Since you may want a house with a half-acre lot which is right

next to a park and close to downtown, but the only houses for sale may have small lots next to centrally-located parks or big lots next to remote parks, this may seem an unrealistic assumption. However, with enough money almost any combination of housing attributes can be created. In this case, you could hypothetically buy up neighboring lots and demolish the houses in order to create the house with the half acre lot near downtown.

Rosen writes the utility function as $U(x, z_1, z_2, \dots, z_n)$, where x is all other goods. The utility function is assumed to be concave (a common assumption in microeconomics). The price of x is set to 1, for the sake of simplicity and it is assumed that only one “ z ” is purchased (as is usually the case with a house). Therefore, income, which is labeled y is equal to $x+p(z)$. To maximize utility, this equation must be solved along with the first order condition: $\partial p / \partial z_i = p_i = U_{z_i} / U_x$.

Figure 1 (from Rosen) shows two bid curves for z_i with different levels of income and utility. Within the two curves, utility provided by the house that is purchased and income are constant and therefore $\Theta(z_1, z_2, \dots, z_n, u, y)$. $\Theta = p(z_1, z_2, \dots, z_n)$ is the equilibrium for a given level of income and utility. What this means is that with a given level of income, a person has various bundles of house features that would make them equally satisfied (i.e. 100 extra square feet of house in exchange for 500 sq feet less of lot, gas stoves and granite countertops in exchange for not having a park nearby). The house-buyer would like to get a house that gives them this level of utility for as little money as possible. Producers are only willing to offer bundles at or above the price where the utility curve is tangent to their iso-profit function (explained below). However, the buyer has no desire to pay more for the same level of utility than necessary, and therefore won't buy a house that costs more than the price at the tangent of producers' iso-

profit functions. Therefore, they will buy the house that provides the bundle of house features that is at the tangent point. Spending as little as possible on a house that provides them a given level of utility frees up money for the buyer to spend on other things that make them happy, thus maximizing their total utility.

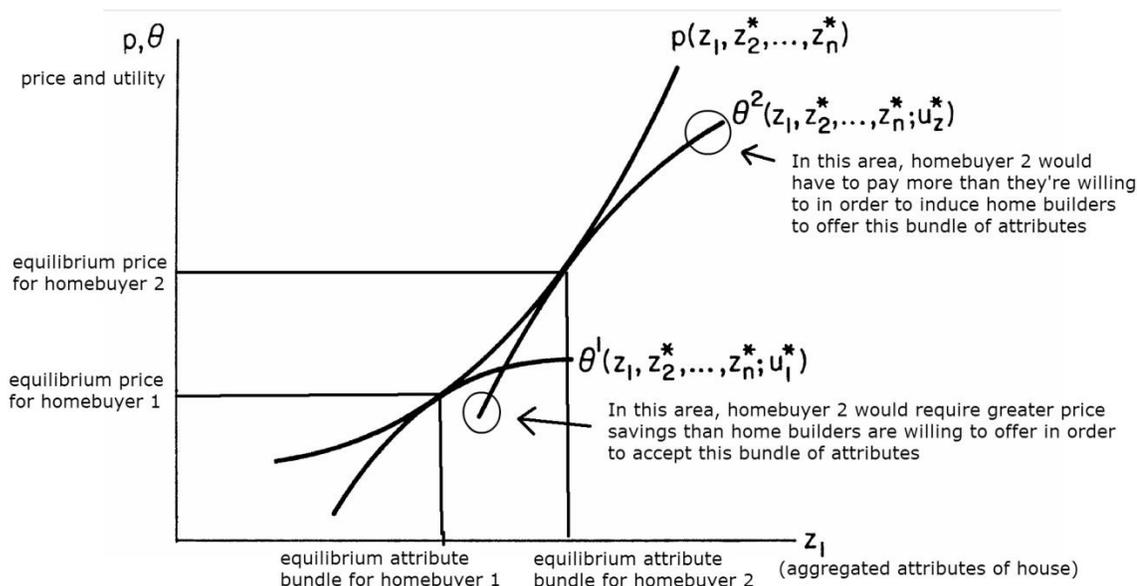


Figure 1: Isoprofit curve and 2 bid curves

Producers of housing (ie developers) face a similar decision except that they are maximizing profit rather than utility. They have offer functions of how much they are willing to accept for a product with a given profit and unit cost: $\Phi(z_1, z_2, \dots, z_n, \pi, \beta)$ where π is profit and β is unit input cost. There are many combinations of housing inputs that would yield the same level of profit, but the producer is constrained by the consumer. Above and below the tangent point, consumers are not willing to pay that price for the given bundle of goods, and therefore producers offer the house with the bundle of features specified by the tangent. Therefore,

where Φ and Θ are tangent, the house is produced and sold, creating an equilibrium price. The combination of all of the equilibrium prices forms the price curve for varying levels of z_i .

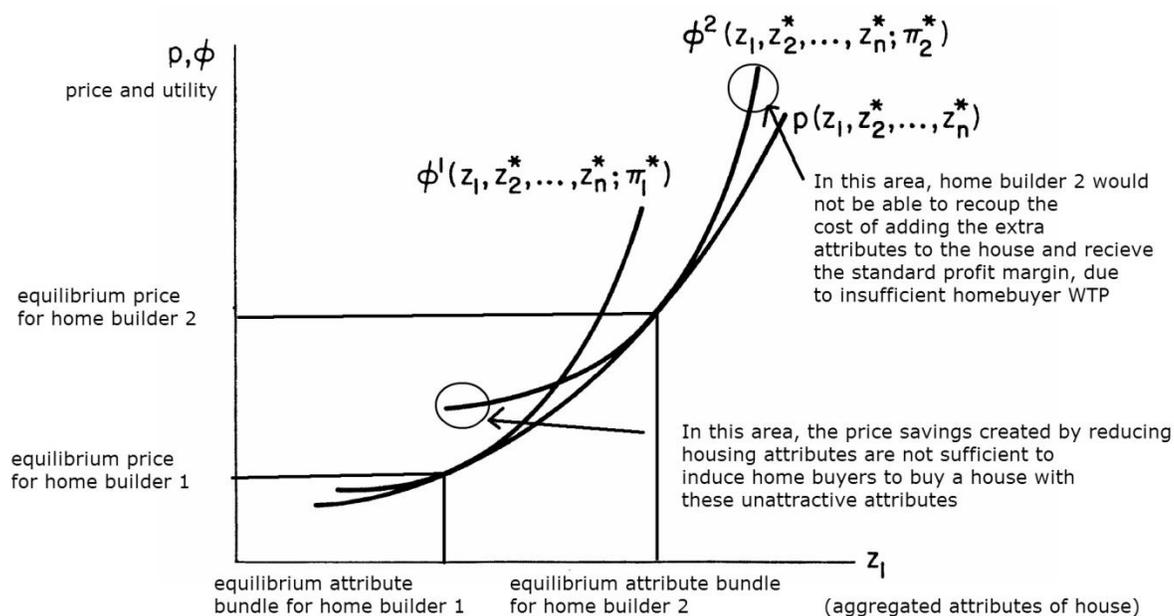


Figure 2: Isoprofit curve and 2 producers' offer curves

Theory Concerning Functional Form

This section draws heavily from Taylor's overview of the hedonic method. In order to estimate the value of individual components of what makes a house valuable (the "z"s), a regression is estimated with house prices as the dependent variable and as many "z"s as are available (assuming they are not strongly correlated with another z and that they have a theoretical basis for influencing property value) as the independent variables. This then provides an estimate of how much each "z" adds to the price of a house. The researcher faces a decision on how to model this function in order to best mimic real life. Most often they face the

decision of how to manipulate the variables within an ordinary least squares regression – the functional form. According to Taylor (2003: 352), “Little theoretical guidance exists for the choice of functional form... because the price schedule is determined in the marketplace by the interactions between many different buyers and sellers.” If the individual attributes of a house could be repackaged without cost (for example the granite countertops from one house transferred to another for free), then the price of a house would be a simple linear function of its attributes. This would mean that each incremental increase in an attribute would cause the same impact on price – meaning the difference in price between houses with 0 and 50 foot of beach frontage and houses with 50 and 100 foot of beach frontage would be the same.

This is an unrealistic assumption, so different functional forms are often used. In particular, most researchers assume decreasing returns to square footage in terms of price. This is most often captured by log transforming price but is also accomplished with other transformations, such as double log transformation, quadratic, and box-cox transformations. A major advantage of the log-linear and log-log forms is that they are easier to interpret. Coefficients in a log-linear model represent the percent change in price caused by a unit change of the independent variable, while the coefficients in a log-log model are simply elasticities. Implicit price on log-linear models is $\beta \cdot P$ and in log-log models is $\beta \cdot P/z$, where β is the coefficient. These interpretations are much simpler than in quadratic or Box-Cox models, increasing their popularity (the Box-Cox model is a flexible model, which transforms variables in order to create the best fitting model). Linear models are extremely easy to interpret – the coefficient is simply the implicit price (change in P with a change in z). However, linear models are theoretically flawed as mentioned above, so log-linear and log-log models are the most frequently used.

Cropper, Deck, and McConnell in 1988 simulated datasets based off of known marginal prices and estimated models with differing functional forms on samples of them to test the accuracy of the various models (1988). They found that when all housing attributes that influence price are perfectly observed, quadratic and box-cox models worked best but that when attributes were missing, inaccurate, or represented by proxy, the simpler forms performed better.

A new kind of model was introduced around the turn of the 21st century with improved computing power (Geoghegan et al 1997). This is the spatial model, which uses data points which are geocoded to control for spatial autocorrelation between houses. Spatial autocorrelation is defined as the price of a house influencing prices of nearby houses more than distant houses, which is not an unreasonable assumption. Spatial models, though promising, have lagged behind Ordinary Least Squares in popularity, probably due to the complexity involved in implementing and interpreting them. Additionally, no standard spatial model for hedonic property studies has emerged, with each method used only a few times (often by the same researchers), then abandoned. This makes comparison between results of different studies impossible. For these reasons, this study uses log-linear and log-log models, which are fairly standard, rather than the more complex models, which are harder to interpret and are therefore less useful in spite of their potential increased predictive power.

Overview of Empirical Studies

Primitive hedonic models date back to 1874, when Frederick Law Olmstead used one to show how Central Park in NYC had increased property values (Crompton, 2001). Unfortunately, Olmstead just looked at how much property prices increased nearby, ignoring the fact that the vast majority of that increase was due to the rapid expansion of New York City (In 1850, the

farthest north urbanized part of Manhattan was 30th St and most people lived below 15th - the creation of Central Park which starts at 60th led to the eviction of only 1600 people). People also applied similar methodology to parkways and found them to do wonders for property values – later it was realized that most of this change was probably due to the parkways making public transit and vehicle traffic more effective (Crompton, 2001). The first study that used more sophisticated statistical tools was by Herrick in 1939 (Crompton, 2001). Herrick ran a regression by hand to look at the effects of park acreage and population density on property values in DC for 1911-1937. However, his mathematical models were criticized, and no one else did another study on the effects of parks on property value until 1964 (they simply continued to rely on the flawed studies of the past as well as anecdotal evidence) (Crompton, 2001).

The first modern hedonic property study concerning land use amenities was by Knetsch in 1964. Due to the limitations of computing power at the time, the regression was linear, had 8 variables, and a sample size of only 103. The amenity was a reservoir and the measures of it were adjacency, linear distance, and linear distance squared. The study estimated a positive effect of proximity to the reservoir, a result which would be borne out by almost all subsequent studies.

In 1972, a hedonic property study used logarithmic transformations for the first time (Emerson). It used both log-log and log-linear regressions with 26 independent variables. They introduced the log transformations in order to reflect relative rather than absolute changes. Although log models showed up throughout the 1970s, the linear form was still far more common until the 1980s. Starting in the 2000s, however, logarithmic forms became overwhelmingly dominant (54 log studies versus 14 linear).

Although there have been many hedonic property studies, I will focus on those that use views as variables since they pertain most directly to this study. The first hedonic property study to include views was published by Maser, Riker, and Rosett in 1977. Their research was highly ambitious for the time, using 62 variables including views of 10 different land uses. To determine if a land use could be seen from a property, the researchers actually drove to each of the 398 properties. Unfortunately, this large number of variables yielded few significant results with a sample of this size, with the only view variable of significance being that of the airport. This is an indication of the limits of using views as variables before computing power was such that they could be modeled rather than directly observed. A 1989 study by Kirshner and Moore made a coarse model of view, by predicting that houses in a flat neighborhood only had a view of water if they were adjacent to water or across the street from water but that houses in steep neighborhoods always had a view. Doss and Taff's 1993 study used a dataset that included whether or not a house had a view of the Mississippi River or of a Lake, but such datasets are rare and it still provides only a dummy variable, rather than a gauge of the quality of the view. It also only provided views for a couple of features – no land uses were recorded. Benson et al in 1997 for the first time introduced view variables that took into account how dominant each feature was within the viewshed, but again this was done by driving to each house.

The first study to use GIS (Geographic Information Systems) to model viewsheds was by Lake et al in 1998. It used a DEM (Digital Elevation Model), which is the elevation of the ground, to predict what could be seen from each house. To improve the predictive power of the DEM they added height wherever there were buildings based on the number of stories in the building. From this elevation map they produced a viewshed of what could be seen within 500 meters. They then estimated regressions using the total areas of each land use visible as

variables, as well as applying linear and square distance decays to the areas. Of note, this study was able to use 4000 properties because it was modeling viewsheds rather than directly observing. Two years later Lake et al published two similar studies with different foci.

These studies were undoubtedly groundbreaking, but the technology of the time limited their accuracy. Crucially, they ignore the ability of trees and shrubs to obstruct vision and virtually place the observer on the roof of their house – a vantage point rarely experienced by most. Therefore, they overestimate the area that can be seen.

In 2002, Paterson and Boyle made use of viewsheds but in a less accurate manner than Lake et al, even though they extended the range to 1000 meters. They use as variables the percentage of the viewshed in various land uses as well as the percentage of the area within 1000 meters visible. They lacked a digital file of the footprints of buildings, meaning buildings were not present as obstructions of views, so they overestimated to an even greater extent the area visible. They also didn't weigh closer views more heavily than distant views, even though 10 square meters of trees five meters from the house dominates the view much more than 10 square meters of trees 300 meters from the house.

Joly et al in 2009 created viewsheds that controlled for vegetation for the first time by applying an estimated height to different types of vegetation (based on a satellite landcover map). They used the area visible within two bands (0-70 m, 70-280m) for woodlands and agricultural land as variables as well as the total area of roads within 280 m and the total area of water visible within 40 km. The use of bands for two of the variables roughly controls for the effect of perspective.

Sander and Polasky took a similar approach to in their 2009 study to the Lake et al study, by using DEM along with an estimated building height to create their viewsheds. Their viewsheds were of land uses within 1000 m. They produced several variables, including the total area that could be seen from the house, the “standard deviations of elevation in a viewshed (measure of relief),” percentages of the viewshed composed of forest, low vegetation, and water as well as the diversity of land uses visible (unspecified, but presumably different classes of buildings and impervious ground along with vegetation type). Finally, they included a dummy variable for whether or not downtown Minneapolis was within the viewshed. One year later, they published another study with Haight that focused on tree canopy cover near homes rather than viewsheds but still included a variable for total area within the viewshed (Sander, Polasky and Haight 2010). These studies have many of the flaws discussed in earlier studies such as lack of vegetation as an obstruction and placing the viewer on top of their house. Also, they admit that much of the effect of the view of downtown may simply be due to properties within 1 km of downtown having lower property values *ceteris paribus* than suburbs.

In 2010, Poudyal et al published a study examining the effect of views of a forest within a viewshed. However, their viewshed modeling was less precise than Sander and Polasky’s. They simply used the DEM (without adjusting for buildings). Therefore, their predictions lacked the obstruction of buildings as well as vegetation. Their variable was the area of forest visible, which creates the same problems of perspective mentioned above. One positive point about this study, however, is that they did not constrain their viewsheds by distance, which means they captured views more than a kilometer away.

The most recent study I have found that uses viewsheds for a hedonic property study is a thesis by T. Dudley in 2012. This was the first study to use LIDAR to account for the obstruction of vegetation. LIDAR produces a map of the actual elevations of the tops of vegetation, buildings, etc. called a DSM (Digital Surface Model). Because his DSM data are at a much higher resolution than previous studies, Dudley constrains his viewsheds to 1/8 of a mile for computation purposes. He uses as his viewshed variables total areas of visible buildings, streets, and water, as well as areas of various vegetation heights within different distances from the house. This is a decent way to control for the fact that views of distant areas will tend to have a lesser impact than a view of a nearby area of the same size, but creates the problem of introducing many variables.

Although the use of DSM rather than DEM is a major improvement, this study still has accuracy problems. Again, this study places the virtual viewer on top of the house, which is a seldom-experienced vantage point. Another issue, which would be more serious if the observer were placed at ground level, is that this viewshed model assumes trees are solid obstructions, when in fact typically they are limbed up, meaning that this model predicts obstructions where there are in fact none. I attempt to remedy these issues with this study.

Method

A major methodological contribution of this study is the creation of a viewshed that takes into account the limbing up of trees. To do this I created three viewsheds of what can be seen within half a mile of a house and used a series of conditional statements to create a more accurate viewshed. The vantage point is a random point somewhere along the edge of the

house. This technique was chosen because of the difficulty of automatically locating the front or back of a house with the software.

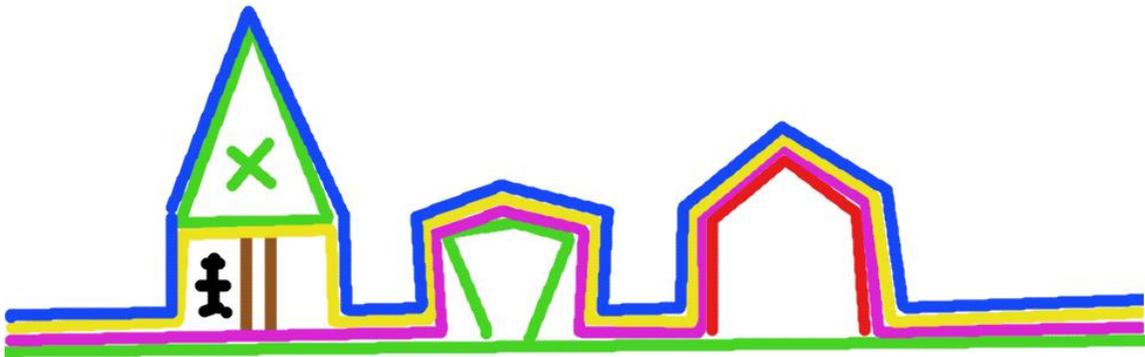


Figure 3: Components used to build viewshed estimate

In the above diagram, the viewer is looking at a house on the right, a shrub in the middle, and a tree with a person standing behind it on the left. The first viewshed is based on the pink line: the DSM everywhere except where there are trees - where a building's footprint is not coded and where the difference between the DSM and DEM (the height of vegetation) is greater than 20 feet. In these areas with trees, the DEM is used instead. This viewshed is of a world where there are buildings and shrubs but not trees.

The second viewshed is based on the yellow line, which is the same as the pink line except that instead of DEM being used where there are trees, $DEM + 8$ feet is used. This value is an estimate of how much trees are limbed up based on city of Corvallis regulations for trees

near sidewalks (Tree & Park Strip Planting). Clearly this is more of an average than a true measurement, since many trees are limbed up to 20 feet and others aren't limbed at all. A more precise way to determine the average limbing height would be to sample the limbing height of trees within the study area. This second viewshed depicts what is visible if shrubs and buildings are present but the limbed up part of trees are an obstruction and the upper part of trees aren't. This viewshed later serves as a form of negative in a conditional statement to determine what is actually visible through the limbed up portion of a tree. The third viewshed is based on the blue line, which is simply the LIDAR elevation. This assumes that everything below a tree's canopy is obstructed.

By combining these viewsheds with the use of conditional statements, we can create a prediction of what can actually be seen from a given lot. If an area is not visible in the first viewshed, then it truly is not visible since it's being obscured by a building, shrub, or the ground. If an area is visible in viewshed 1, it is actually visible under two circumstances: (1) if it is visible in both viewsheds 2 and 3, meaning that it is not behind a tree at all or (2) if it is obstructed in both viewsheds 2 and 3, meaning that it is behind a tree, but is behind the limbed up part of the tree. This is not strictly true, since in areas with steep topography, an area may indeed be behind the limbed up area of a tree, but it's still being obstructed by the upper portion of a different tree. In the case of an uphill perspective this would be the upper portion of a closer tree and looking downhill it would be the upper portion of a farther tree.

This new viewshed is used as a variable in the regressions and the results are compared to those generated using a simple viewshed that only takes the DSM into account. This simple model is much easier to build, and took far less time to run – traits which are very valuable for

ease of use. Similar models have been used in previous studies such as Dudley's 2012 study. If the complex model provides no more insight than the simple model, it is clearly not worth the effort and restricted sample size. The complex model divides the viewshed into city and county zones as listed in Table 1. These zones are intersected with the land use within the zone: buildings (drawn from a shapefile of the footprints of buildings), other impervious surfaces (also from a shapefile), tall vegetation (over 20 feet), and short vegetation. To give an example, the area of tree canopy visible within house zones would be coded under "house, tall vegetation". The simple model uses total building area, other impervious surface area, and total vegetation area visible from a house as variables.

Table 1: Broad Classifications of Land Uses	
Houses (RS 3.5-RS 12)	Right of Way (railroads or roads)
Apartments (RS 20 & MUR)	Conservation (land set aside from logging or development)
Commercial	Forestry (logging land)
Industrial	Farmland
Institutional (government and university buildings)	

Since a given area of shrubs close-by dominates the view more than the same area farther away, I control for perspective in the complicated model, but not in the simple model. To do this, the viewshed is divided into bands of within 100 feet and 100-500 feet. According to Bartie et al (2010), the perceived area of an object is $\text{area}/\text{distance}^2$. Therefore the area of each land use within 100 feet would be divided by $[(100-0)/2]^2$ and within 100-500 feet would be divided by $[(500-100)/2]^2$. This creates an approximation of the perceived area from each zone.

These total perceived areas for each zone are then added together for each land-use category, reducing the total number of variables. These total perceived areas are the primary variables used to predict housing prices. Each of the 10 zones has 4 land uses (except for buildings in a right of way) yielding 39 variables, which have been pared down using correlation tables and variable inflation factors to prevent multicollinearity.

In addition to the viewshed variables, the model includes 8 other spatial variables and 6 house feature variables that predict housing prices. The data set is from the Benton County Assessor's office and includes properties I consider to be in the Corvallis, Oregon housing market: all properties in Benton County sold from 8/9/2011 to 8/8/2012 except for those in North Albany, which I consider a different housing market from the rest of the county. Only arm's length transactions are considered – any transactions of less than \$60,000 were eliminated, since they all seemed to have a large disparity with the Zillow estimate of their price. Vacant land and apartment complexes with more than 7 bedrooms were excluded because vacant land is typically valued for its productivity rather than its views or access to amenities and because data on number of rooms was missing for larger apartment complexes. This yields a sample size of 566.

The Benton County dataset includes information on lot size, house size, year the house was built, number of bathrooms, number of bedrooms, and presence of central air conditioning. These were used as independent variables and are all commonly used in the literature, except for central air conditioning, which intuitively seems like something is correlated with housing price.

As people of European and Asian descent have a higher than average income in the US, I controlled for the percentage of the people in a block that are non-Euro, non-Asian American. This could indicate that blocks with large non-Euro, non-Asian populations have homes that are of lower quality in a way not captured by the data, which reflects these ethnic groups' lower incomes. The ethnic data comes from census.gov. I assign the appropriate school district to each taxlot, based on data from the Oregon Explorer (<http://spatialdata.oregonexplorer.info/GPT9/catalog/main/home.page>). School districts serve as better proxies for communities within Benton County than city limits, since they include unincorporated rural areas with the community they interact with most, and for that reason I assign each property the school district it falls within. From the FEMA website, I downloaded a shapefile of 100 and 500 year floodplains and assigned these to lots, due to the literature suggesting this could have a negative impact on property values.

In addition to these spatial variables, I created several others using the travel time to various features. Travel time is a better measure of disamenity than Euclidean distance, since most of the effects of features were related to the disamenity of nearby traffic or how long it takes to get somewhere. To do this I created a raster from a polyline of roads and assigned each road type an amount of time to travel a given distance. This was the cost raster used to calculate a raster of the minimum number of minutes to get to various geographic features. I checked the accuracy of these maps using Google maps from random taxlots in the sample. I calculated driving time to the following features: the center of Corvallis, commercial zones, busy roads, large parks (over 50 acres), and small parks (less than 50 acres). Proximity to the center of Corvallis is expected to increase values due to ease of getting to trendy areas and the University, while being close to commercial zones and arterial roads (typically with a speed limit

of 35 MPH or above) is expected to reduce values, since they are associated with noise and traffic. Both kinds of parks are expected to have positive effects, but I separated them since smaller parks are very different experiences from big parks, typically with manicured landscapes, sporting facilities, or children's play structures rather than lightly managed "natural" vegetation.

An outline of the steps taken follow below:

1. Download shapefiles for property sales, land use zones, building and pavement areas, school districts, floodplains, racial data, etc, from city, county, state, and federal websites and integrate data from different data sources. Obtain LIDAR data from US forest service.
2. Assign road velocities to different types of road convert to rasters and use the cost distance tool to create a raster of travel times to different features. Check against google maps for rough accuracy, convert back to vector, and join vector values to sales data.
3. Join other locational data such as racial data and school districts to sales data.
4. Create a raster of building and vegetation height which is DSM (LIDAR) minus DEM (elevation).
5. Select areas with vegetation or buildings over 20 feet tall and exclude areas defined as buildings by the building shapefile. These areas are defined as areas with trees
6. Create a new raster which is identical to the DEM except in areas with trees, where the raster is equal to DEM + 8 feet.
7. Iterate the ID numbers for the properties in the sample.

8. For each iterated property, create a random point within the lot.
9. Snap the random point to the nearest edge of a building shape
10. Create a 500 foot buffer around the point
11. Clip the three elevation rasters (DEM, DEM + 8 ft on tree footprints, DSM) to the buffer area.
12. Create viewsheds for each of these three clipped elevation rasters
13. Run the raster calculator with the following conditional statement:


```
"Con(\ "%noview_value%" == 1 & ((\ "%beview_value%" == 0) & (\ "%hhview_value%" == 0)) | ((\ "%beview_value%" == 1) & (\ "%hhview_value%" == 1))), 1, 0)"
```

Where noview is the DEM + 8, beview is DEM, and hhview is DSM.
14. Create a 100 foot buffer shape around the point
15. Erase the 100 foot buffer from the 500 foot buffer
16. Intersect the viewshed twice: once with the 100 foot buffer, and once with the 500 foot “doughnut”
17. Add a field for the view type
18. Calculate the field to be equal to the various land use categories by selecting rows based on various attributes. For example, "\ "GRIDCODE\" = 1 AND \ "broad_zone\" = 'houses' AND \ "FID_imperv\" = -1 AND \ "FID_buildi\" = -1" selects residential short vegetation. Repeat this for every land use type.
19. Dissolve both viewsheds so that every view type is one shape rather than many separate shapes and calculate the area of each view type.
20. Add a field for perceived area and calculate it as the viewed area divided by the midpoint of the distance band squared.

21. Join the perceived area of one of the viewsheds to the other and add them together for total perceived area of each view type.
22. For every iteration after the first, append the resulting viewshed to the first viewshed iterated
23. Pivot the resulting table such that there are columns of each view type and a row for each property sold
24. Join these view type values to the sales data and run descriptive statistics and OLS regressions

In the appendix can be found a python version of the model builder file for the iterated part of the GIS work. Note that this is not python, but model builder code and will not work as is.

Results

In order to determine if this alternative approach to estimating viewsheds actually improves the predictive power of hedonic models, I compare the results from using it to a regression using a viewshed based on the LIDAR elevations (i.e. it assumes trees are not limbed up). The log-linear model is specified as follows:

$$\text{Ln}(\text{Price}_j) = \beta_0 + \sum_{i=0}^n \beta_i X_{ij} + \varepsilon_j$$

Where X_i is a vector of all housing features. The log-log model is specified as follows:

$$\text{Ln}(\text{Price}_j) = \beta_0 + \sum_{i=0}^n \beta_i \ln X_{ij} + \sum_{k=0}^m \beta_k X_{kj} + \varepsilon_j$$

Where X_i is the vector of all housing variables that aren't dummy variables. X_k is the vector of dummy variables, which include living in Monroe or Philomath rather than Corvallis and living in a 100 or 500 year floodplain instead of not in a floodplain.

Table 2: Means and Standard Deviations of Variables (N=566)				
<i>Variable</i>	<i>Mean</i>	<i>Standard Deviation</i>	<i>Minimum</i>	<i>Maximum</i>
Property sale price	286,765	161,617	60,000	2,150,000
Year Built	1978.403	25.35917	1860	2011
Finished Area (Square Feet)	1754.713	861.087	400	19280
Bedrooms	3.246006	2.303257	0	40
Bathrooms	2.21246	1.215052	1	20
Acres	0.5347604	2.043513	0.02	33.5
Minutes of driving time to the center of Corvallis	7.092	4.6262	0	31
Minutes to a park over 50 acres	2.978	2.121	1	18
Minutes to a park under 50 acres	2.04	3.213	1	27
Minutes to Commercial Zone	1.552	1.904	1	17
Minutes to an arterial road	1.913	1.196	1	8
Percentage of block not of Euro or Asian descent (1 = 100%)	0.0548	.0586	0	0.330
Square feet of area visible from the property taking into account limbing up	27639.97	45722.98	110.3688	261133.4
Square feet of area visible from the property using LIDAR only	30510.88	61460.89	7.525146	403280.1
Houses in a 100 year floodplain (dummy variable)	0.0106007	0.1025031	0	1
Houses in a 500 year floodplain (dummy variable)	.024735	.1554537	0	1
Houses in Monroe (dummy variable)	0.0106007	0.1025031	0	1
Houses in Philomath (dummy variable)	0.1246006	0.3307944	0	1
Non-house	0.0351438	0.184438	0	1

From Table Two, we can see a result that goes against what I predicted. Rather than the viewshed with trees limbed up expanding the area of the viewshed, on average, it decreased it. I believe that this is due to the fact that in the more complex model, for an area to be considered visible, it must be visible in a world without trees. When using LIDAR alone, however, if there is a tree adjacent to a house at the randomly selected point, then the viewshed will be made as if the viewer is standing on top of the tree. So my method of predicting viewsheds improved the viewshed in a way that I had not envisioned.

Due to multicollinearity issues, some variables were dropped or combined. Any variables which had a correlation coefficient over .5 were either combined or one of the variables was dropped. Number of bedrooms and baths were dropped for this reason, since finished area controlled for these variables already. Driving time to small parks and commercial areas were dropped, as these features are primarily clustered near the center of Corvallis. In addition, views of various land uses of the zones which are less commonly visible were often combined. All of the land uses in apartment zones were combined, as were land uses in conservation areas, forestry land, industrial zones, and right of ways. Commercial land uses, except for short vegetation, were combined and institutional (University or municipal) buildings and impervious areas were combined. Farmland categories were simplified to vegetation and nonvegetation.

Tables 3 and 4 show the regression results

	<i>Simple Viewshed</i>			<i>Complex Viewshed</i>		
<i>R-Squared</i>	0.7251			0.7377		
<i>Adj. R-Squared</i>	0.7186			0.7251		
<i>F</i>	111.99			58.31		
<i>variable</i>	<i>Coefficient</i>	<i>T-Value</i>	<i>P-value</i>	<i>Coefficient</i>	<i>T-Value</i>	<i>P-value</i>
Year Built	.00086	2.07	0.039	0.00111	2.70	0.007
Finished Area (Square Feet)	.0003332	29.22	0.000	0.00033	28.53	0.000
Acres	.0743145	5.04	0.000	0.06687	4.38	0.000
Minutes of driving time to the center of Corvallis	.0004381	0.11	0.913	-0.00229	-0.56	0.574
Minutes to a park over 50 acres	.00018	0.03	0.979	0.00179	0.26	0.796
Minutes to an arterial road	.016496	1.73	0.084	0.01784	1.82	0.069
Percentage of block not of Euro or Asian descent (1 = 100%)	-1.0640	-7.30	0.000	-1.03095	-6.99	0.000
Houses in a floodplain (dummy variable)	-.08462	-1.88	0.060	-0.12185	-2.54	0.011
Houses in Monroe (dummy variable)	-.60276	-5.13	0.000	-0.55245	-4.63	0.000
Houses in Philomath (dummy variable)	-.16652	-3.32	0.001	-0.17111	-3.27	0.001
Perceived area of buildings	2.10e-06	2.11	0.036	NA	NA	NA
Perceived area of pavement	4.52e-07	0.42	0.678	NA	NA	NA
Perceived area of vegetation	-3.05e-07	-2.54	0.011	NA	NA	NA
Perceived area of apartments	NA	NA	NA	-0.00999	-0.01	0.989
Perceived area of commercial short vegetation	NA	NA	NA	-0.72890	-0.59	0.559
Perceived area of other commercial	NA	NA	NA	0.30602	0.44	0.660
Perceived area of conservation areas	NA	NA	NA	-0.54910	-1.13	0.260
Perceived area of industrial areas	NA	NA	NA	-0.16029	-0.34	0.731

Perceived area of institutional short vegetation	NA	NA	NA	0.44793	0.32	0.747
Perceived area of institutional tall vegetation	NA	NA	NA	-1.08384	-0.06	0.950
Perceived area of institutional pavement and buildings	NA	NA	NA	15.16488	1.70	0.090
Perceived area of residential buildings	NA	NA	NA	-0.09229	-0.38	0.707
Perceived area of residential pavement	NA	NA	NA	-0.72780	-2.88	0.004
Perceived area of residential short vegetation	NA	NA	NA	0.06052	0.61	0.543
Perceived area of residential tall vegetation	NA	NA	NA	0.22647	1.28	0.199
Perceived area of right of ways	NA	NA	NA	-0.06221	-0.50	0.620
Perceived area of farm buildings and pavement	NA	NA	NA	-0.35614	-0.46	0.648
Perceived area of farm vegetation	NA	NA	NA	12.51956	4.48	0.000
Perceived area of non-park forests	NA	NA	NA	0.02282	0.06	0.955
Constant	10.13107	12.58	0.000	9.65750	12.07	0.000

Bold numbers are significant at $p < .1$

	<i>Simple Viewshed</i>			<i>Complex Viewshed</i>		
<i>R-Squared</i>	0.7170			0.7311		
<i>Adj. R-Squared</i>	0.7097			0.7197		
<i>F</i>	98.59			64.07		
<i>variable</i>	<i>Coefficient</i>	<i>T-Value</i>	<i>P-value</i>	<i>Coefficient</i>	<i>T-Value</i>	<i>P-Value</i>
Year Built	.80877	1.02	0.310	1.68783	2.13	0.033
Finished Area (Square Feet)	.66894	27.35	0.000	0.68075	28.41	0.000
Acres	.0280946	6.14	0.000	0.02757	6.43	0.000
Minutes of driving time to the center of Corvallis	-.00421	-1.68	0.093	-0.00444	-1.69	0.091

Minutes to a park over 50 acres	.00061	0.48	0.632	0.00049	0.39	.698
Minutes to an arterial road	.0019669	1.65	0.099	0.00311	2.68	0.008
Percentage of block not of Euro or Asian descent (1 = 100%)	-.00327	-2.74	0.006	-0.00171	-1.74	0.083
Houses in a floodplain (dummy variable)	-.13553	-3.08	0.002	-0.15559	-3.35	0.001
Houses in Monroe (dummy variable)	Omitted in this regression	Omitted in this regression	Omitted in this regression	-0.57367	-6.90	0.000
Houses in Philomath (dummy variable)	-.1547812	-2.16	0.031	-0.11678	-2.89	0.004
Perceived area of buildings	-.0040636	-0.48	0.635	NA	NA	NA
Perceived area of pavement	-.0187502	-2.67	0.008	NA	NA	NA
Perceived area of vegetation	.0258653	2.78	0.006	NA	NA	NA
Perceived area of apartments	NA	NA	NA	-0.00006	-0.02	0.983
Perceived area of commercial short vegetation	NA	NA	NA	-0.01394	-2.28	0.023
Perceived area of other commercial	NA	NA	NA	0.01140	1.91	0.057
Perceived area of conservation areas	NA	NA	NA	0.00020	0.09	0.927
Perceived area of industrial areas	NA	NA	NA	0.00137	0.26	0.791
Perceived area of institutional short vegetation	NA	NA	NA	0.00599	1.19	0.235
Perceived area of institutional tall vegetation	NA	NA	NA	-0.01430	-2.60	0.010
Perceived area of institutional pavement and buildings	NA	NA	NA	0.00513	0.83	0.409
Perceived area of residential buildings	NA	NA	NA	-0.01047	-2.63	0.009
Perceived area of residential pavement	NA	NA	NA	-0.00760	-2.94	0.003

Perceived area of residential short vegetation	NA	NA	NA	0.00984	2.11	0.035
Perceived area of residential tall vegetation	NA	NA	NA	-0.00081	-0.54	0.589
Perceived area of right of ways	NA	NA	NA	0.00153	0.78	0.435
Perceived area of farm buildings and pavement	NA	NA	NA	0.00984	1.20	0.229
Perceived area of farm vegetation	NA	NA	NA	0.03179	2.94	0.003
Perceived area of non-park forests	NA	NA	NA	-0.00742	-1.02	0.306
Constant	1.31804	0.22	0.826	-5.49590	-0.92	0.356

Bold numbers are significant at $p < .1$

In all of the regressions, most of the non-viewshed variables were significant. Every regression except the simple log-log indicated that newer houses cost more than older houses, which is appropriate, since old houses can require expensive renovations. Also, bigger houses and houses on bigger lots tended to cost more ceteris paribus in all regressions, as would be predicted by the consumer theory of strong monotonicity, or “more is better, all else equal”. In the complex log-log regression, distance to downtown Corvallis as well as the dummy variables for Philomath and Monroe areas showed significant negative effects. In the simple log-log regression, the same was found except that the Monroe variable was dropped due to collinearity. However, in the log linear regressions, only the dummy variables were significant, perhaps due to multicollinearity. Regardless, it seems clear that proximity to the center of Corvallis has a positive effect on housing prices, likely due to the reduction in commute times and the accessibility of stores, bars, restaurants, etc. Dudley (2012) found the same in his study (the study to date that’s most similar to this one in viewshed complexity) with regard to distance to downtown. Houses near downtown and campus (which is itself near downtown) may also be

rented out to multiple families, which could be increasing prices. Conversely, housing prices tend to increase with distance from arterial roads in all regressions, indicating that people dislike the traffic, noise, presence of strangers near their house, or some combination of the above, which comes from close proximity to major roads. This seems to indicate that home buyers like to be close to busy areas so they are easy to access, but not too close, which anecdotally seems to be the case. This result supports what many previous studies have found. Li and Saphores in their 2011 study found in all of their regressions that a house within 250 m of an arterial road tended to have a lower price.

As expected, blocks which have a higher percentage of non-Euro, non-Asian residents were found to correlate with lower housing prices in all regressions. This negative effect from an increased presence of non-euro-Americans has been indicated by studies many times, most recently by Strippelhoff in 2011. A house being located in a floodplain had a negative effect in all regressions, as would be expected due to the risk of owning a house in a floodplain. This result also has a substantial body of literature supporting it, including Cho et al in 2010.

The only non-viewshed variable which was consistently nonsignificant was distance to parks over 50 acres in size. This may be because Corvallis has such evenly distributed large natural areas. Other studies such as Karkoski, 2009 have found that distance from a large nature area did have a negative effect. However, these studies are often oriented towards one dominant park, unlike this study which has large parks dispersed throughout the study area.

With the viewshed variables, the log-log regression showed the most significance, perhaps reflecting the decreasing marginal effects of good and bad views as they become more dominant in the visible landscape. Intuitively, in the complex models, more houses and

residential pavement in view tended to decrease housing prices *ceteris paribus* (residential pavement was significant in both complex regressions, whereas buildings only showed a significant effect in the log-log model). This represents a significant success, since Dudley's (2012) model, which did not take into account limbing up, found no significant effect from pavement or buildings. Given the negative impact of views of buildings and pavement, it is understandable that people plant trees and shrubs to screen the view of their neighbors. Interestingly in the simple models, although pavement correlated negatively with price in the log-log regression and was nonsignificant in the log-linear model, there was no significant effect of buildings in the log-log regression and a *positive* effect in the log-linear model. This may reflect the positive effect of being close to a commercial area or to campus described below outweighing the effect which is seen when restricted to residential areas.

In the log-log model, views of short residential vegetation was correlated with increased property values, but the lack of significance in tree variables indicates that people do not actually value trees *per se* as much as they value the privacy they provide. This stands in direct contrast to Dudley's (2012) results which found a positive effect of views of tall vegetation and a negative effect of views of short vegetation. His difference in results concerning the value of trees could be a reflection of the part of the country our studies are located in. In the Southeast, heat in the summer can be oppressive and shade trees are highly valued for that reason. In the Northwest, sun is a scarce resource except in the summer, when the heat is not nearly as intense as in the Southeast. Therefore, the dislike of shade most of the year may be providing a negative effect from trees, which when combined with the positive screening effect yield nonsignificant results. In the simple log-log regression, views of vegetation were positively correlated with price, but in the simple log-linear regression they were negatively correlated. I

know of no instance of this being theoretically explained in the literature, so this may be related to houses which are close to commercial areas having higher property values. However, given the fact that the two regressions yielded opposite results, it seems likely that this is simply random noise from the regression. This may indicate that the less sophisticated viewshed is a less effective variable than the viewshed that took into account limbing up, perspective, and zoning.

Somewhat counterintuitively, in the log-log regression, visibility of short vegetation in a commercial zone decreased property values, whereas a view of anything else in a commercial zone increased property values. This could be an effect of desire to be close to commercial areas for convenience but not too close due to the disamenities of noise, traffic, having strangers around, etc. A house which can see little short vegetation in a commercial zone but much of other things in a commercial zone is probably seeing trees and buildings only, because they are more distant. This supports Lipscomb's (2009) findings that distance from commercial areas increased property prices in some neighborhoods, but decreased them in others, depending on the neighborhood's distance from the commercial area.

The inverse could be at play when it comes to institutional zones (university and municipal). In the log-log model, views of institutional trees were correlated with lower property values but views of other things in institutional zones had no significant effect and in the log-linear model, views of institutional buildings and pavement had a positive correlation with price, but the other variables were nonsignificant. In this case it could be that people who aren't students generally dislike being close to campus, but students want to be as close as possible. Therefore, areas that are close enough to campus to see the trees, but not close

enough to see much of anything else may have the disamenities of being close to campus (loud, boisterous students) without the amenities (the ability to quickly walk to campus).

In both regressions, farmland vegetation had a positive correlation with house prices and farm buildings and pavement had no effect. This makes sense since people tend to have a positive attitude towards bucolic scenes but are ambivalent towards loud, dusty, sometimes smelly farm buildings. This stands in stark contrast to the nonsignificant effect of conservation areas described below. This decisive positive effect of farmland stands among a literature which has not shown agriculture to have a decisively positive or negative effect, perhaps due to variances in what farmland is like in different parts of North America. Cotteleer et al's (2008) study in British Columbia found a negative effect for farms that had animals but a nonsignificant effect for farms without animals. In the area of that study, the dominant agricultural industry is dairy, which nowadays typically means huge cow factories with associated manure lagoons. It is understandable that this is considered a disamenity rather than a bucolic ideal. However, the nonsignificant effect of non-animal farms is surprising, given the positive results found in this study. Most non-grazed farmland in British Columbia is in hay, which is aesthetically quite similar to the grass fields which surround Corvallis. I can think of no theoretical reason for this difference, and therefore assume that one of our studies is providing spurious results.

No models found a significant effect of visibility of industrial areas. The industrial area in Corvallis which is most visible from residential zones is on Research Way, and it is not heavy industry. It is mostly large parking lots, generic blocky office buildings, lawns, and young trees. Although the view is probably not inspiring to most people, it does not elicit the negative response heavy industry anecdotally seems to and has a side benefit of the house potentially

being close to one's place of work. Other studies have found that in the case of heavy industry, there is a negative effect. Strippelhof (2011), for instance found that a greater number of manufacturing jobs in the block group was correlated with lower housing prices.

Surprisingly, views of apartments, conservation areas, forestry land, and right of ways had no significant impact in any model. The lack of significant effect of right of ways is perhaps less surprising than the other two. Every property has a road going to it, and these roads are typically visible from the house. People may not see views of right of ways as a disamenity because they're largely seen as unavoidable. The lack of significance of views of apartments is highly surprising, since there is anecdotally much resistance to their construction, and since houses have a negative impact. This may reflect several things. Firstly, it could be simply that there are not enough data points with views of apartments to capture the effect. There are few studies which examine the effect of views of or proximity to multifamily housing, perhaps for this reason. Secondly, houses which have views of apartments would be very valuable if they were also zoned such that they could be redeveloped into apartments. This may be enough to offset the negative effect in houses which are not zoned high density but which have views of apartments but not enough to create a positive effect overall.

The lack of effect of forestry land is surprising but can potentially be explained by the fact that forestry land views are an amenity when the trees are tall, but a disamenity when logged, since people don't like the look of recently clear cut forest. These two opposite effects may account for the nonsignificant response. Kim's (1999) dissertation indicated that clear cut and even age forestry land had a negative impact on property values, but that mixed age stands had a positive effect.

The lack of significant effect of conservation areas is fairly disappointing, since determining the effect of conservation areas on property values has been a major thrust in the environmental economic hedonic literature. I believe that the results in this study were confounded by the fact that the two main areas where houses border conservation areas in Corvallis are different in important ways which were not controlled for. There are houses with views of natural areas in the hills to the North and West of Corvallis as well as in South Town. I believe uncontrolled-for factors which make South Town a less pleasant place to live than other parts of Corvallis are being attributed to being in eyeshot of the park, which negates the positive effect of parks in other parts of Corvallis. These negative factors include being separated from downtown by a freeway, having a five lane freeway and strip mall running through the area (I controlled for arterial roads, but a three lane road was considered the same as a five lane road, with the latter potentially being substantially more disruptive), being near the only heavy industrial areas in town, and having the large park be primarily sports fields rather than natural vegetation. Previous studies have indicated that these factors can be disamenities rather than amenities. For instance, Cotteleer et al (2008) found that housing prices decreased with distance from parks over 12 acres in size while they increased with distance from parks less than half an acre in size. If Willamette park is functioning more like a small park than a large one, then this could partially explain the unexpected result. This study could be improved by having a second category of arterial road, distinguished between light and heavy industry, and between sports and nature parks.

Conclusion

Overall this new viewshed model yielded many nonsignificant results, probably due to the small sample size caused by the complexity of the viewshed variables and the restriction of

viewsheds to within 500 feet of houses. Nonetheless, the study provides many interesting findings, particularly concerning views of residential areas which are more common and therefore had a greater number of predictive data points. The study found that views of residential buildings and pavement were negatively correlated and short vegetation positively correlated with price. The lack of significance on views of residential trees may indicate that people don't actually value trees per se as much as their screening attributes. The study also indicated that being as close to campus as possible was desirable but that it's better to be close but not too close to commercial areas. Also, views of farm vegetation had a significant positive correlation with property values, which is surprising, as views of conservation areas, industrial areas, and apartments did not.

Building the viewshed model took a significant amount of time. If its predictive power is no greater than that of a simple model which can be built and run quickly, then it is not worth the effort. Overall, the poor results of the simple model make the complex model seem worthwhile. The simple model, by not dividing viewsheds into land-use zones gives much coarser variables. These simple variables gave results that were not particularly enlightening and which sometimes contradicted themselves depending on functional form. Although the log-log regression estimated a negative correlation with views of pavement and the log-linear regression gave a nonsignificant correlation, there was a nonsignificant and *positive* correlation respectively for buildings. Given the lack of specificity in the variable, we do not know if this is merely a fluke or a reflection of proximity to more densely built areas having a positive impact on property prices. Short vegetation showed a positive correlation in the log-log regression but a negative correlation in the log-linear. This seems to indicate that these results are simply spurious correlations. Unfortunately, without running the model controlling for everything

except for one of the three added complexities (perspective, land use zone, and limbing up), it is unknown the absence of which is causing the inferior results in the simple model.

A natural next step for this line of research is to determine which of the three viewshed improvements adds the most improvement to the variable estimates. The precision of accounting for limbing up in estimating a viewshed may not be worth the effort given the statistical noise created by the imperfection of the dependent variable and the omission of distant views. However, with access to greater computing power, this may be a moot point, with the ability to easily adopt all three changes and even add greater complexity as well as increase sample size. For example, qualitative differences within land use zones could be identified to create new categories. In the county zoning file, heavy industry and office parks are undifferentiated, yet this could be easily remedied and was simply not something I thought about when designing the study. I did think of the impact of different types of parks (sports vs nature) but decided it would be too difficult to go through every park in the county and code it based on its recreation type. Therefore, I distinguished between small and large parks for the purposes of road distance variables but not for viewsheds since I was worried about the number of viewshed variables. It would probably be worth trying the model with a separate category of viewshed variables for sports parks which would include all parks under 50 acres as well as Willamette Park (the large park south of downtown which I believe may have confounded the conservation viewshed variable results).

Another improvement on this study would be to select a random subsample of the properties being evaluated and actually measure the height the trees are limbed up to in order to get an average that is based on more than simply looking at a subsample on google

streetview and not systematically estimating the heights. Also, a larger sample size may reduce the correlation between distance to downtown and distance to commercial areas, meaning distance to commercial areas could be added back to the regressions. This might reduce some of the ambiguity in the results concerning the impacts of visibility of commercial areas. Adding distance to institutional and apartment zones would also clarify in this regard. This clarification would be most useful if the increase in processing power added not only to sample size but to distance from the house the viewshed is cut off. I originally had the viewsheds cut off at half a mile, but the model would have taken too long to run – on the order of months, given on the computing power I had available to me. I also originally had 4000 data points with similar downsides. Unfortunately, it will probably be a long time before processing power makes this complicated model quick and easy to use. If processing power doubles every three years, it will take about 20 years before this model can be run on an average personal computer in a week with 4000 data points and a half mile viewshed. This is a considerable period. Since access to a supercomputer 100 times more powerful than the average personal computer is hard to get, this leaves the options of ESRI overhauling their program to make more efficient calculations (unlikely given their lack of competition and their tendency to patch rather than overhaul) or firms such as Google or Microsoft expanding their mapping software into the type of advanced spatial analytics currently monopolized by ESRI.

There are several potential options available for increasing sample size and viewshed distance even given current computing power. It may be possible to link up many lab computers to run the viewshed model faster. I tried to do this and failed, but it may be possible. Another option is to reduce the resolution of the DSM (LIDAR imagery) in order to reduce processing time. Also, the viewshed could be estimated for only one house per block with the viewshed

variables from that one house applied to all other houses in the block. Finally, it may be that a totally different method may be more effective for estimating viewsheds. Images from Google Streetview could be analyzed by a program based on color in order to estimate the area of each land use type visible. I considered this option, but I do not have experience with this type of color recognition analysis. These simplifications are worth testing to see whether the increased sample size and view distance accomplished by them more than compensate for the reduced precision.

Overall, I believe these types of advanced models could eventually (or currently with access to a supercomputer or linked computers) be extremely powerful in providing fine-tuned policy recommendations. However, given the technology used in the current study, the above results should be taken with a grain of salt given the small sample size and shortcomings of not controlling for certain features as described above.

Works Cited

- Bartie, P., F. Reitsma, S. Kingham, and S. Mills. 2010. Advancing visibility modelling algorithms for urban environments. *Computers, Environment and Urban Systems* 34 (6): 518–531.
- Benson, E.D., J.L. Hansen, A.L. Schwartz, Jr., and G.T. Smersh. 1997. The influence of Canadian investment on U.S. residential property values. *Journal of Real Estate Research* 13: 231-249.
- Cho, S.H., D.M. Lambert, R.K. Roberts, and S.G. Kim. 2010. Demand for open space and urban sprawl: the case of Knox County, Tennessee. In *Progress in Spatial Analysis* (pp. 171-193). Springer Berlin Heidelberg.
- Cotteleer, G., T. Stobbe, and G.C. van Kooten. 2008, February. Expert Opinion Versus Transaction Evidence: Using the Reilly Index to Measure Open Space Premiums in the Urban-Rural Fringe. *Modelling of Agricultural and Rural Development Policies*. Paper prepared for the 107th EAAE Seminar, Sevilla, Spain.

- Crompton, J.L. 2001. The impact of parks on property values: A review of the empirical evidence. *Journal of Leisure Research* 33 (1):1-31
- Cropper, M.L., L. Deck, and K.E. McConnell. 1988. On the choice of functional form for hedonic price functions. *Review of Economics and Statistics*. 70(4):668-675.
- Doss, C.R. and S.J. Taff. 1993. The relationship of property values and wetlands proximity in Ramsey County, Minnesota. Economic Report 93-4, Department of Agricultural and Applied Economics, University of Minnesota, St. Paul, MN.
- Dudley, T.E. 2012. The Value of Landscape Amenities: Contrasting Feature Proximity and View. Master's Thesis, North Carolina State University.
- Emerson, F.C. 1972. Valuation of residential amenities: an econometric approach. *The Appraisal Journal* 40 (April):268-278.
- Geoghegan, J., L. A. Wainger and N. E. Bockstael. 1997. Spatial Landscape Indices in a Hedonic Framework: An Ecological Economics Analysis Using GIS. *Ecological Economics*, 23: 251-264.
- Joly, D., T. Brossard, J. Cavailhes, M. Hilal, F.P. Tourneux, C. Tritz, and P. Wavresky. 2009. A Quantitative Approach to the Visual Evaluation of Landscape. *Annals of the Association of American Geographers*, 99(2): 292–308.
- Karkoski, J. 2009. A Hedonic Pricing Model of the Effect of the American River Parkway on Home Prices in Sacramento County, California, USA. Master's Thesis, California State University, Sacramento.
- Kim, Y.S. 1999. Measuring the economic impact of forests on neighboring properties. Doctoral Dissertation, Oregon State University, Corvallis.
- Kirshner, D. and D. Moore. 1989. The effect of San Francisco Bay water quality on adjacent property values. *Journal of Environmental Management* 27: 263-274.
- Knetsch, J.L. 1964. The influence of reservoir projects on land values. *Journal of Land Economics* 46 (1):231-243.
- Lake, I.R., A.A. Lovett, I.J. Bateman, and B.H. Day. 2000. Improving land compensation procedures via GIS and hedonic pricing. *Environment and Planning* 18: 681-696.
- Lake, I.R., A.A. Lovett, I.J. Bateman, and B.H. Day. 2000. Using GIS and large-scale digital data to implement hedonic pricing studies. *International Journal of Geographic Information Science* 14 (6):521-541.

- Lake, I.R., A.A. Lovett, I.J. Bateman, and I.H. Langford. 1998. Modelling environmental influences on property prices in an urban environment. *Computers, Environment, and Urban Systems* 22 (Mar.): 121-136.
- Li, W. and J. Saphores. 2011. A Spatial Hedonic Analysis of the Value of Urban Land Cover in the Multifamily Housing Market in Los Angeles, CA. *Urban Studies* 49 (12):2597-2615.
- Lipscomb, C.A. 2009. An Alternative Spatial Hedonic Estimation Approach. *Journal of Housing Research* 15 (2):143-160.
- Maser, S., W. Riker, and R. Rosett. 1977. The Effects of Zoning and Externalities on the Price of Land: An Empirical Analysis of Monroe County, New York. *Journal of Law and Economics* 20 (Spring):111-132.
- Paterson, R.W. and K.J. Boyle. 2002. Out of sight, out of mind? Using GIS to incorporate visibility in hedonic property value models. *Land Economics* 78(3):417-425.
- Poudyal N.C., D.G. Hodges, J. Fenderson, and W. Tarkington. 2010. Realizing the Economic Value of a Forested Landscape in a Viewshed. *Southern Journal of Applied Forestry* 34 (2): 72-78
- Rosen, S. 1974. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of Political Economy* 82(January/February):34-55.
- Sander, H.A. and S. Polasky. 2009. The value of views and open space: Estimates from a hedonic pricing model for Ramsey County, Minnesota, USA. *Land Use Policy* 26: 837-845.
- Sander, H., S. Polasky, and R.G. Haight. 2010. The value of urban tree cover: A hedonic property price model in Ramsey and Dakota Counties, Minnesota, USA. *Ecological Economics* 69: 1646–1656
- Strippelhoff, C.C. 2011. The Influence of Parks and Greenspace on the Value of Commercial Real Estate. Master's Thesis, Georgia Institute of Technology.
- Taylor, L.O. 2003. The Hedonic Method. In *A Primer on Nonmarket Valuation* Edited by P.A. Champ, K.J. Boyle, and T.C. Brown Dordrecht: Kluwer: 331-393.
- Tree & Park Strip Planting. Corvallis, OR. Ordinance 2.19.130.

Appendix

Python export of model builder code:

```

# -*- coding: utf-8 -*-
# -----
-----
# viewshed model.py
# Created on: 2014-06-11 14:40:45.00000
# (generated by ArcGIS/ModelBuilder)
# Usage: viewshed model <intersect_100__value__shp__2_>
<taxlots_2011_2> <intersect_500__value__shp__4_>
# Description:
# -----
-----

# Set the necessary product code
# import arcinfo

# Import arcpy module
import arcpy

# Check out any necessary licenses
arcpy.CheckOutExtension("spatial")

# Load required toolboxes
arcpy.ImportToolbox("Model Functions")

# Script arguments
intersect_100__value__shp__2_ = arcpy.GetParameterAsText(0)
if intersect_100__value__shp__2_ == '#' or not
intersect_100__value__shp__2_:
    intersect_100__value__shp__2_ =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
intersect_100_%value%.shp" # provide a default value if unspecified

taxlots_2011_2 = arcpy.GetParameterAsText(1)
if taxlots_2011_2 == '#' or not taxlots_2011_2:
    taxlots_2011_2 = "taxlots_2011_2" # provide a default value if
unspecified

intersect_500__value__shp__4_ = arcpy.GetParameterAsText(2)
if intersect_500__value__shp__4_ == '#' or not
intersect_500__value__shp__4_:
    intersect_500__value__shp__4_ =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
intersect_500_%value%.shp" # provide a default value if unspecified

# Local variables:
notree_UTM = "notree_UTM"
hh_utm = "hh_utm"
five_tree_50_utm =
"C:\\Users\\Jacob\\Documents\\ArcGIS\\Default.gdb\\five_tree_50_utm"

```

```

viewshed2_mdb =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
viewshed2.mdb"
building_Merge_utm =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\points.mdb\\
\\building_Merge_utm"
be8_utm2__2_ =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
viewshed.mdb\\be8_utm2"
Layer_half_value__4_ =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
Layer_half_%value%"
taxlots_2011 = "taxlots_2011"
points =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\points.mdb\\
\\points"
dissolve_100_value_shp = intersect_100_value_shp_2_
v100_dissolve_value_shp_2_ = dissolve_100_value_shp
v100ft_dissolve2__8_ = v100_dissolve_value_shp_2_
dissolve_100_value_shp_2_ = v100ft_dissolve2__8_
dissolve_100_value_shp_3_ = dissolve_100_value_shp_2_
dissolve_500_value__5_ = dissolve_100_value_shp_3_
dissolve_500_value_shp_2_ = dissolve_500_value__5_
dissolve_500_value_shp_3_ = dissolve_500_value_shp_2_
two_dissolve_value_shp_8_ = dissolve_500_value_shp_3_
two_dissolve_value_shp_4_ = two_dissolve_value_shp_8_
dissolve_half_value__5_ = two_dissolve_value_shp_4_
dissolve_half_value_shp_8_ = dissolve_half_value__5_
dissolve_500_8_shp_2_ = dissolve_half_value_shp_8_
dissolve_half_8 = "dissolve_half_8"
intersect_500_value_shp_3_ =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
intersect_500_%value%.shp"
Value = taxlots_2011_2
point2_value_ = Value
point2_value__3_ = point2_value_
hhview_value = point2_value__3_
comb_value_ = hhview_value
value_vector_shp = comb_value_
select7_value_shp = value_vector_shp
select7_value_shp_3_ = select7_value_shp
select7_value_shp_4_ = select7_value_shp_3_
intersect_100_value_shp = select7_value_shp_4_
v2_value_100_shp_2_ = intersect_100_value_shp
Layer_100_value_ = v2_value_100_shp_2_
Layer_100_value__3_ = Layer_100_value_
viewshed_intersect100_4_60_ = Layer_100_value__3_
Layer_100_value__4_ = viewshed_intersect100_4_60_
viewshed_intersect100_4_52_ = Layer_100_value__4_
Layer_100_value__5_ = viewshed_intersect100_4_52_
viewshed_intersect100_4_53_ = Layer_100_value__5_
Layer_100_value__6_ = viewshed_intersect100_4_53_

```

viewshed_intersect100_4_54_ = Layer_100_value__6_
Layer_100_value__7_ = viewshed_intersect100_4_54_
viewshed_intersect100_4_59_ = Layer_100_value__7_
Layer_100_value__8_ = viewshed_intersect100_4_59_
viewshed_intersect100_4_55_ = Layer_100_value__8_
Layer_100_value__9_ = viewshed_intersect100_4_55_
viewshed_intersect100_4_56_ = Layer_100_value__9_
Layer_100_value__10_ = viewshed_intersect100_4_56_
viewshed_intersect100_4_57_ = Layer_100_value__10_
Layer_100_value__11_ = viewshed_intersect100_4_57_
viewshed_intersect_3_3_97_ = Layer_100_value__11_
Layer_100_value__12_ = viewshed_intersect_3_3_97_
viewshed_intersect_half4_119_ = Layer_100_value__12_
Layer_100_value__13_ = viewshed_intersect_half4_119_
viewshed_intersect_half4_123_ = Layer_100_value__13_
Layer_100_value__14_ = viewshed_intersect_half4_123_
viewshed_intersect_half4_124_ = Layer_100_value__14_
Layer_100_value__15_ = viewshed_intersect_half4_124_
viewshed_intersect_half4_125_ = Layer_100_value__15_
Layer_100_value__16_ = viewshed_intersect_half4_125_
viewshed_intersect_half4_127_ = Layer_100_value__16_
Layer_100_value__17_ = viewshed_intersect_half4_127_
viewshed_intersect_half4_128_ = Layer_100_value__17_
Layer_100_value__18_ = viewshed_intersect_half4_128_
viewshed_intersect_half4_133_ = Layer_100_value__18_
Layer_100_value__19_ = viewshed_intersect_half4_133_
viewshed_intersect_half4_134_ = Layer_100_value__19_
Layer_100_value__20_ = viewshed_intersect_half4_134_
viewshed_intersect_half4_135_ = Layer_100_value__20_
Layer_100_value__21_ = viewshed_intersect_half4_135_
viewshed_intersect_half4_137_ = Layer_100_value__21_
Layer_100_value__22_ = viewshed_intersect_half4_137_
viewshed_intersect_half4_138_ = Layer_100_value__22_
Layer_100_value__23_ = viewshed_intersect_half4_138_
viewshed_intersect_half4_143_ = Layer_100_value__23_
Layer_100_value__24_ = viewshed_intersect_half4_143_
viewshed_intersect_half4_144_ = Layer_100_value__24_
Layer_100_value__25_ = viewshed_intersect_half4_144_
viewshed_intersect_half4_145_ = Layer_100_value__25_
Layer_100_value__26_ = viewshed_intersect_half4_145_
viewshed_intersect_half4_147_ = Layer_100_value__26_
Layer_100_value__27_ = viewshed_intersect_half4_147_
viewshed_intersect_half4_148_ = Layer_100_value__27_
Layer_100_value__28_ = viewshed_intersect_half4_148_
viewshed_intersect_half4_153_ = Layer_100_value__28_
Layer_100_value__29_ = viewshed_intersect_half4_153_
viewshed_intersect_half4_154_ = Layer_100_value__29_
Layer_100_value__76_ = viewshed_intersect_half4_154_
viewshed_intersect_half4_155_ = Layer_100_value__76_
Layer_100_value__31_ = viewshed_intersect_half4_155_
viewshed_intersect_half4_157_ = Layer_100_value__31_
Layer_100_value__32_ = viewshed_intersect_half4_157_

```

viewshed_intersect_3_3_76_ = Layer_100_value_32_
Layer_100_value_33_ = viewshed_intersect_3_3_76_
viewshed_intersect_3_3_81_ = Layer_100_value_33_
Layer_100_value_34_ = viewshed_intersect_3_3_81_
viewshed_intersect_3_3_82_ = Layer_100_value_34_
Layer_100_value_35_ = viewshed_intersect_3_3_82_
viewshed_intersect_3_3_83_ = Layer_100_value_35_
Layer_100_value_36_ = viewshed_intersect_3_3_83_
viewshed_intersect_3_3_85_ = Layer_100_value_36_
Layer_100_value_37_ = viewshed_intersect_3_3_85_
viewshed_intersect_3_3_86_ = Layer_100_value_37_
Layer_100_value_38_ = viewshed_intersect_3_3_86_
viewshed_intersect_3_3_91_ = Layer_100_value_38_
Layer_100_value_39_ = viewshed_intersect_3_3_91_
viewshed_intersect_3_3_92_ = Layer_100_value_39_
Layer_100_value_40_ = viewshed_intersect_3_3_92_
viewshed_intersect_3_3_93_ = Layer_100_value_40_
Layer_100_value_41_ = viewshed_intersect_3_3_93_
Layer_100_value_51_ = Layer_100_value_41_
Layer_100_value_47_ = Layer_100_value_51_
Layer_100_value_43_ = Layer_100_value_47_
Layer_100_value_50_ = Layer_100_value_43_
Layer_100_value_44_ = Layer_100_value_50_
Layer_100_value_49_ = Layer_100_value_44_
Layer_100_value_48_ = Layer_100_value_49_
Layer_100_value_45_ = Layer_100_value_48_
Layer_100_value_42_ = Layer_100_value_45_
Layer_100_value_2_ = Layer_100_value_42_
intersect_500_value_shp = select7_value_shp_4_
intersect_500_value_shp_2_ = intersect_500_value_shp_
Layer_500_value_ = intersect_500_value_shp_2_
Layer_100_value_140_ = Layer_500_value_
viewshed_intersect100_4_47_ = Layer_100_value_140_
Layer_100_value_100_ = viewshed_intersect100_4_47_
viewshed_intersect100_4_13_ = Layer_100_value_100_
Layer_100_value_101_ = viewshed_intersect100_4_13_
viewshed_intersect100_4_14_ = Layer_100_value_101_
Layer_100_value_102_ = viewshed_intersect100_4_14_
viewshed_intersect100_4_17_ = Layer_100_value_102_
Layer_100_value_106_ = viewshed_intersect100_4_17_
viewshed_intersect100_4_46_ = Layer_100_value_106_
Layer_100_value_103_ = viewshed_intersect100_4_46_
viewshed_intersect100_4_18_ = Layer_100_value_103_
Layer_100_value_104_ = viewshed_intersect100_4_18_
viewshed_intersect100_4_19_ = Layer_100_value_104_
Layer_100_value_105_ = viewshed_intersect100_4_19_
viewshed_intersect100_4_20_ = Layer_100_value_105_
Layer_100_value_138_ = viewshed_intersect100_4_20_
viewshed_intersect_3_3_21_ = Layer_100_value_138_
Layer_100_value_107_ = viewshed_intersect_3_3_21_
viewshed_intersect_half4_23_ = Layer_100_value_107_
Layer_100_value_108_ = viewshed_intersect_half4_23_

```

```
viewshed_intersect_half4_24_ = Layer_100_value_108_
Layer_100_value_109_ = viewshed_intersect_half4_24_
viewshed_intersect_half4_25_ = Layer_100_value_109_
Layer_100_value_110_ = viewshed_intersect_half4_25_
viewshed_intersect_half4_26_ = Layer_100_value_110_
Layer_100_value_111_ = viewshed_intersect_half4_26_
viewshed_intersect_half4_27_ = Layer_100_value_111_
Layer_100_value_112_ = viewshed_intersect_half4_27_
viewshed_intersect_half4_28_ = Layer_100_value_112_
Layer_100_value_113_ = viewshed_intersect_half4_28_
viewshed_intersect_half4_29_ = Layer_100_value_113_
Layer_100_value_114_ = viewshed_intersect_half4_29_
viewshed_intersect_half4_30_ = Layer_100_value_114_
Layer_100_value_115_ = viewshed_intersect_half4_30_
viewshed_intersect_half4_31_ = Layer_100_value_115_
Layer_100_value_116_ = viewshed_intersect_half4_31_
viewshed_intersect_half4_32_ = Layer_100_value_116_
Layer_100_value_117_ = viewshed_intersect_half4_32_
viewshed_intersect_half4_33_ = Layer_100_value_117_
Layer_100_value_118_ = viewshed_intersect_half4_33_
viewshed_intersect_half4_34_ = Layer_100_value_118_
Layer_100_value_119_ = viewshed_intersect_half4_34_
viewshed_intersect_half4_35_ = Layer_100_value_119_
Layer_100_value_120_ = viewshed_intersect_half4_35_
viewshed_intersect_half4_36_ = Layer_100_value_120_
Layer_100_value_121_ = viewshed_intersect_half4_36_
viewshed_intersect_half4_37_ = Layer_100_value_121_
Layer_100_value_122_ = viewshed_intersect_half4_37_
viewshed_intersect_half4_38_ = Layer_100_value_122_
Layer_100_value_123_ = viewshed_intersect_half4_38_
viewshed_intersect_half4_39_ = Layer_100_value_123_
Layer_100_value_124_ = viewshed_intersect_half4_39_
viewshed_intersect_half4_40_ = Layer_100_value_124_
Layer_500_value_3_ = viewshed_intersect_half4_40_
viewshed_intersect_half4_41_ = Layer_500_value_3_
Layer_100_value_126_ = viewshed_intersect_half4_41_
viewshed_intersect_half4_120_ = Layer_100_value_126_
Layer_100_value_127_ = viewshed_intersect_half4_120_
viewshed_intersect_3_3_12_ = Layer_100_value_127_
Layer_100_value_128_ = viewshed_intersect_3_3_12_
viewshed_intersect_3_3_13_ = Layer_100_value_128_
Layer_100_value_129_ = viewshed_intersect_3_3_13_
viewshed_intersect_3_3_14_ = Layer_100_value_129_
Layer_100_value_130_ = viewshed_intersect_3_3_14_
viewshed_intersect_3_3_15_ = Layer_100_value_130_
Layer_100_value_131_ = viewshed_intersect_3_3_15_
viewshed_intersect_3_3_16_ = Layer_100_value_131_
Layer_100_value_132_ = viewshed_intersect_3_3_16_
viewshed_intersect_3_3_17_ = Layer_100_value_132_
Layer_100_value_133_ = viewshed_intersect_3_3_17_
viewshed_intersect_3_3_18_ = Layer_100_value_133_
Layer_100_value_134_ = viewshed_intersect_3_3_18_
```

```

viewshed_intersect_3_3_19_ = Layer_100_value__134_
Layer_100_value__135_ = viewshed_intersect_3_3_19_
viewshed_intersect_3_3_20_ = Layer_100_value__135_
Layer_100_value__136_ = viewshed_intersect_3_3_20_
Layer_100_value__137_ = Layer_100_value__136_
Layer_100_value__141_ = Layer_100_value__137_
Layer_100_value__144_ = Layer_100_value__141_
Layer_100_value__142_ = Layer_100_value__144_
Layer_100_value__145_ = Layer_100_value__142_
Layer_100_value__143_ = Layer_100_value__145_
Layer_100_value__146_ = Layer_100_value__143_
Layer_100_value__147_ = Layer_100_value__146_
Layer_100_value__148_ = Layer_100_value__147_
Layer_500_value__2_ = Layer_100_value__148_
noview_value = point2_value__3_
beview_value = point2_value__3_
hundred_value_shp = point2_value__3_
doughnuts_five_value_shp = hundred_value_shp
five_value_shp = point2_value__3_
no3_value_ = five_value_shp
be_value_ = five_value_shp
hh2_value_ = five_value_shp
taxlots_2011_lyr_2_ = Value
dissolve_500_8 =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
viewshed2.mdb\\dissolves\\dissolve_500_8"
dissolve_500_8_shp =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
dissolve_2\\dissolve_500_8.shp"
intersect_500_8_shp_3_ = "intersect_500_8.shp (3)"
dissolve_500_value_shp = intersect_500_value_shp_4_
v100_dissolve_value_shp_4_ = dissolve_500_value_shp
v100ft_dissolve2_4_ = v100_dissolve_value_shp_4_
dissolve_500_value__4_ = v100ft_dissolve2_4_
dissolve_500_value__2_ = dissolve_500_value__4_
taxlots_2011_lyr =
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
taxlots_2011_lyr"

# Process: Make Feature Layer (5)
arcpy.MakeFeatureLayer_management(taxlots_2011, taxlots_2011_lyr,
"", viewshed2_mdb, "FID OBJECTID_12 VISIBLE NONE;Shape Shape VISIBLE
NONE;FID_1 FID_1 VISIBLE NONE;FID_1_1 FID_1_1 VISIBLE NONE;FID_1_1_1
FID_1_1_1 VISIBLE NONE;OBJECTID_1 OBJECTID_1 VISIBLE NONE;OBJECTID_2
OBJECTID_2 VISIBLE NONE;OBJECTID_3 OBJECTID_3 VISIBLE
NONE;FID_1_1__1 FID_1_1__1 VISIBLE NONE;FID_1_1__2 FID_1_1__2
VISIBLE NONE;FID_1_1__3 FID_1_1__3 VISIBLE NONE;MapNumber MapNumber
VISIBLE NONE;Taxlot Taxlot VISIBLE NONE;SERIAL SERIAL VISIBLE
NONE;PIN PIN VISIBLE NONE;LEVY LEVY VISIBLE NONE;SITUS SITUS VISIBLE
NONE;AGENT AGENT VISIBLE NONE;OWNER OWNER VISIBLE NONE;MAILING
MAILING VISIBLE NONE;MAILCITY MAILCITY VISIBLE NONE;MAILSTATE
MAILSTATE VISIBLE NONE;MAILZIP MAILZIP VISIBLE NONE;SUBNAME SUBNAME

```

VISIBLE NONE;BLOCK BLOCK VISIBLE NONE;LOT LOT VISIBLE NONE;LANDVAL
 LANDVAL VISIBLE NONE;IMPVAL IMPVAL VISIBLE NONE;ASSESSED_V
 ASSESSED_V VISIBLE NONE;MARKET_VAL MARKET_VAL VISIBLE NONE;YEARBUILT
 YEARBUILT VISIBLE NONE;FIN_AREA FIN_AREA VISIBLE NONE;BEDROOMS
 BEDROOMS VISIBLE NONE;BATHS BATHS VISIBLE NONE;STORY STORY VISIBLE
 NONE;SALE_PRICE SALE_PRICE VISIBLE NONE;SALES_DATE SALES_DATE
 VISIBLE NONE;RECORDING_RECORDING VISIBLE NONE;DEEDREF DEEDREF
 VISIBLE NONE;PROPSTATUS PROPSTATUS VISIBLE NONE;PROPCLASS PROPCLASS
 VISIBLE NONE;ACRES ACRES VISIBLE NONE;NMA NMA VISIBLE
 NONE;SitusStree SitusStree VISIBLE NONE;Situsst Situsst VISIBLE
 NONE;FID_2 FID_2 VISIBLE NONE;ARCID ARCID VISIBLE NONE;GRID_CODE
 GRID_CODE VISIBLE NONE;broad_zone broad_zone VISIBLE NONE;Uniq_ID
 Uniq_ID VISIBLE NONE;Entity Entity VISIBLE NONE;Handle Handle
 VISIBLE NONE;LyrFrzn LyrFrzn VISIBLE NONE;LyrLock LyrLock VISIBLE
 NONE;LyrOn LyrOn VISIBLE NONE;LyrVPFrzn LyrVPFrzn VISIBLE NONE;Color
 Color VISIBLE NONE;EntColor EntColor VISIBLE NONE;LyrColor LyrColor
 VISIBLE NONE;BlkColor BlkColor VISIBLE NONE;Linetype Linetype
 VISIBLE NONE;EntLinety EntLinety VISIBLE NONE;LyrLnType LyrLnType
 VISIBLE NONE;BlkLinety BlkLinety VISIBLE NONE;Elevation Elevation
 VISIBLE NONE;Thickness Thickness VISIBLE NONE;LineWt LineWt VISIBLE
 NONE;EntLineWt EntLineWt VISIBLE NONE;LyrLineWt LyrLineWt VISIBLE
 NONE;BlkLineWt BlkLineWt VISIBLE NONE;RefName RefName VISIBLE
 NONE;LTScale LTScale VISIBLE NONE;ExtX ExtX VISIBLE NONE;ExtY ExtY
 VISIBLE NONE;ExtZ ExtZ VISIBLE NONE;DocName DocName VISIBLE
 NONE;DocPath DocPath VISIBLE NONE;DocType DocType VISIBLE
 NONE;DocVer DocVer VISIBLE NONE;MHNUMBER MHNUMBER VISIBLE
 NONE;RIMELEV RIMELEV VISIBLE NONE;YRCONSTR YRCONSTR VISIBLE
 NONE;INSPECTED INSPECTED VISIBLE NONE;CONDITION CONDITION VISIBLE
 NONE;MAINT_DATE MAINT_DATE VISIBLE NONE;MAINT_CODE MAINT_CODE
 VISIBLE NONE;DRAWING_SE DRAWING_SE VISIBLE NONE;MISC MISC VISIBLE
 NONE;Shape_Le_1 Shape_Le_1 VISIBLE NONE;ZONING_ZONING VISIBLE
 NONE;ZONING_ID ZONING_ID VISIBLE NONE;COUNTY COUNTY VISIBLE
 NONE;ZONING_LAB ZONING_LAB VISIBLE NONE;GENERAL_ZO GENERAL_ZO
 VISIBLE NONE;LAND_USE LAND_USE VISIBLE NONE;DESCRIPTIO DESCRIPTIO
 VISIBLE NONE;SYMBOL SYMBOL VISIBLE NONE;EDITED EDITED VISIBLE
 NONE;OBJECTID_4 OBJECTID_4 VISIBLE NONE;OBJECTID OBJECTID VISIBLE
 NONE;DESIGNATIO DESIGNATIO VISIBLE NONE;DESIG_TTL DESIG_TTL VISIBLE
 NONE;COMPPLAN_COMPPLAN VISIBLE NONE;COMPPLAN_I COMPPLAN_I VISIBLE
 NONE;X_COORD X_COORD VISIBLE NONE;Y_COORD Y_COORD VISIBLE
 NONE;CITYNAME CITYNAME VISIBLE NONE;FID_4 FID_4 VISIBLE NONE;NAME
 NAME VISIBLE NONE;AREA_1 AREA_1 VISIBLE NONE;PERIMETE_1 PERIMETE_1
 VISIBLE NONE;DIST_DIST VISIBLE NONE;CNTY CNTY VISIBLE NONE;INST_ID
 INST_ID VISIBLE NONE;DIST_ID DIST_ID VISIBLE NONE;apartbuild
 apartbuild VISIBLE NONE;apartimper apartimper VISIBLE
 NONE;apartshort apartshort VISIBLE NONE;aparttall aparttall VISIBLE
 NONE;commbuild commbuild VISIBLE NONE;commimper commimper VISIBLE
 NONE;commshort commshort VISIBLE NONE;commtall commtall VISIBLE
 NONE;conserbuil conserbuil VISIBLE NONE;conserimpe conserimpe
 VISIBLE NONE;conershor conershor VISIBLE NONE;consertall
 consertall VISIBLE NONE;forestbuil forestbuil VISIBLE
 NONE;forestimpe forestimpe VISIBLE NONE;forestshor forestshor
 VISIBLE NONE;foresttall foresttall VISIBLE NONE;indusbuild

```

indusbuild VISIBLE NONE;indusimper indusimper VISIBLE
NONE;indusshort indusshort VISIBLE NONE;industall industall VISIBLE
NONE;instibuild instibuild VISIBLE NONE;instishort instishort
VISIBLE NONE;institall institall VISIBLE NONE;resbuild resbuild
VISIBLE NONE;resimperv resimperv VISIBLE NONE;resshort resshort
VISIBLE NONE;restall restall VISIBLE NONE;ROWbuild ROWbuild VISIBLE
NONE;ROWimperv ROWimperv VISIBLE NONE;ROWshort ROWshort VISIBLE
NONE;ROWtall ROWtall VISIBLE NONE;instiimper instiimper VISIBLE
NONE;FID_3 FID_3 VISIBLE NONE;FLD_AR_ID FLD_AR_ID VISIBLE
NONE;FLD_ZONE FLD_ZONE VISIBLE NONE;FLOODWAY FLOODWAY VISIBLE
NONE;SFHA_TF SFHA_TF VISIBLE NONE;STATIC_BFE STATIC_BFE VISIBLE
NONE;V_DATUM V_DATUM VISIBLE NONE;DEPTH DEPTH VISIBLE NONE;LEN_UNIT
LEN_UNIT VISIBLE NONE;VELOCITY VELOCITY VISIBLE NONE;VEL_UNIT
VEL_UNIT VISIBLE NONE;AR_REVERT AR_REVERT VISIBLE NONE;BFE_REVERT
BFE_REVERT VISIBLE NONE;DEP_REVERT DEP_REVERT VISIBLE
NONE;SOURCE_CIT SOURCE_CIT VISIBLE NONE;Floodplain Floodplain
VISIBLE NONE;FID_5 FID_5 VISIBLE NONE;STATEFP10 STATEFP10 VISIBLE
NONE;COUNTYFP10 COUNTYFP10 VISIBLE NONE;TRACTCE10 TRACTCE10 VISIBLE
NONE;BLOCKCE10 BLOCKCE10 VISIBLE NONE;GEOID10 GEOID10 VISIBLE
NONE;NAME10 NAME10 VISIBLE NONE;MTFCC10 MTFCC10 VISIBLE NONE;UR10
UR10 VISIBLE NONE;UACE10 UACE10 VISIBLE NONE;UATYP10 UATYP10 VISIBLE
NONE;FUNCSTAT10 FUNCSTAT10 VISIBLE NONE;ALAND10 ALAND10 VISIBLE
NONE;AWATER10 AWATER10 VISIBLE NONE;INTPTLAT10 INTPTLAT10 VISIBLE
NONE;INTPTLON10 INTPTLON10 VISIBLE NONE;OID_OID_ VISIBLE NONE;Id1
Id1 VISIBLE NONE;Id2 Id2 VISIBLE NONE;Id3 Id3 VISIBLE NONE;percent_
percent_ VISIBLE NONE;IDT IDT VISIBLE NONE;Shape_Leng Shape_Leng
VISIBLE NONE;Shape_Area Shape_Area VISIBLE NONE;FID_6 FID_6 VISIBLE
NONE;ARCID_1 ARCID_1 VISIBLE NONE;GRID_COD_1 GRID_COD_1 VISIBLE
NONE;FROM_NODE FROM_NODE VISIBLE NONE;TO_NODE TO_NODE VISIBLE
NONE;Distance Distance VISIBLE NONE;FID_7 FID_7 VISIBLE NONE;ARCID_2
ARCID_2 VISIBLE NONE;GRID_COD_2 GRID_COD_2 VISIBLE NONE;FROM_NOD_1
FROM_NOD_1 VISIBLE NONE;TO_NODE_1 TO_NODE_1 VISIBLE NONE;Distance_1
Distance_1 VISIBLE NONE;FID_8 FID_8 VISIBLE NONE;ARCID_3 ARCID_3
VISIBLE NONE;GRID_COD_3 GRID_COD_3 VISIBLE NONE;FROM_NOD_2
FROM_NOD_2 VISIBLE NONE;TO_NODE_2 TO_NODE_2 VISIBLE NONE;Distance_2
Distance_2 VISIBLE NONE;FID_9 FID_9 VISIBLE NONE;ARCID_4 ARCID_4
VISIBLE NONE;GRID_COD_4 GRID_COD_4 VISIBLE NONE;FROM_NOD_3
FROM_NOD_3 VISIBLE NONE;TO_NODE_3 TO_NODE_3 VISIBLE NONE;Distance_3
Distance_3 VISIBLE NONE")

```

```

# Process: Iterate Field Values
arcpy.IterateFieldValues_mb(taxlots_2011_2, "FID_1", "String",
"true", "false", "0")

```

```

# Process: Select Layer By Attribute
tempEnvironment0 = arcpy.env.configKeyword
arcpy.env.configKeyword = "1"
arcpy.SelectLayerByAttribute_management(taxlots_2011_lyr,
"NEW_SELECTION", "\"FID_1\" = %value%")
arcpy.env.configKeyword = tempEnvironment0

```

```

# Process: Create Random Points

```

```

tempEnvironment0 = arcpy.env.configKeyword
arcpy.env.configKeyword = "1"
arcpy.CreateRandomPoints_management(points, "point2_%value%",
taxlots_2011_lyr_2_, "0 0 250 250", "1", "0 Meters", "POINT", "0")
arcpy.env.configKeyword = tempEnvironment0

# Process: Snap
arcpy.Snap_edit(point2__value_,
"C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\points.mdb\\
\\building_Merge_utm EDGE '1000 Meters'")

# Process: Buffer (4)
arcpy.Buffer_analysis(point2__value___3_, hundred__value__shp, "100
Feet", "FULL", "ROUND", "NONE", "")

# Process: Buffer (5)
arcpy.Buffer_analysis(point2__value___3_, five__value__shp, "500
Feet", "FULL", "ROUND", "NONE", "")

# Process: Clip
arcpy.Clip_management(notree_UTM, "482867.7435 4947339.9222
482867.7435 4947339.9222", no3__value_, five__value__shp, "-
3.402823e+038", "NONE", "NO_MAINTAIN_EXTENT")

# Process: Viewshed (2)
arcpy.gp.Viewshed_sa(no3__value_, point2__value___3_, noview_value,
"1", "FLAT_EARTH", "0.13", "")

# Process: Clip (2)
arcpy.Clip_management(be8_utm2__2_, "482867.7435 4947339.9222
482867.7435 4947339.9222", be__value_, five__value__shp, "-
3.402823e+038", "NONE", "NO_MAINTAIN_EXTENT")

# Process: Viewshed (3)
arcpy.gp.Viewshed_sa(be__value_, point2__value___3_, beview_value,
"1", "FLAT_EARTH", "0.13", "")

# Process: Clip (3)
arcpy.Clip_management(hh_utm, "482867.7435 4947339.9222 482867.7435
4947339.9222", hh2__value_, five__value__shp, "-3.402823e+038",
"NONE", "NO_MAINTAIN_EXTENT")

# Process: Viewshed
arcpy.gp.Viewshed_sa(hh2__value_, point2__value___3_, hhview_value,
"1", "FLAT_EARTH", "0.13", "")

# Process: Raster Calculator
arcpy.gp.RasterCalculator_sa("Con(\\%noview_value\\" == 1 &
((\\%beview_value\\" == 0) & (\\%hhview_value\\" == 0)) |
(\\%beview_value\\" == 1) & (\\%hhview_value\\" == 1)), 1,
0)", comb__value_)

```

```

# Process: Raster to Polygon
arcpy.RasterToPolygon_conversion(comb__value_, value_vector_shp,
"NO_SIMPLIFY", "VALUE")

# Process: Select
arcpy.Select_analysis(value_vector_shp, select7__value__shp,
 "\"GRIDCODE\" = 1")

# Process: Add Field (15)
arcpy.AddField_management(select7__value__shp, "GRIDCODE_1",
"SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (167)
arcpy.CalculateField_management(select7__value__shp__3_,
"GRIDCODE_1", "1", "VB", "")

# Process: Intersect (4)
arcpy.Intersect_analysis("C:\\Users\\Jacob\\Documents\\ArcGIS\\Defau
lt.gdb\\five_tree_50_utm
#;C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
\\buffers\\hundred_%value%.shp
#;C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
\\select7_%value%.shp #", intersect_100__value__shp, "ALL", "",
"INPUT")

# Process: Add Field (4)
arcpy.AddField_management(intersect_100__value__shp, "view_type2",
"TEXT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Make Feature Layer (4)
arcpy.MakeFeatureLayer_management(v2__value__100_shp__2_,
Layer_100__value_, "", "", "FID FID VISIBLE NONE;FID_five_t
FID_five_t VISIBLE NONE;Shape Shape VISIBLE NONE;GRIDCODE GRIDCODE
VISIBLE NONE;broad_zone broad_zone VISIBLE NONE;FID_imperv
FID_imperv VISIBLE NONE;FID_buildi FID_buildi VISIBLE
NONE;Shape_Leng Shape_Leng VISIBLE NONE;Shape_Area Shape_Area
VISIBLE NONE;FID_hundre FID_hundre VISIBLE NONE;CID CID VISIBLE
NONE;BUFF_DIST BUFF_DIST VISIBLE NONE;FID_select FID_select VISIBLE
NONE;ID ID VISIBLE NONE;GRIDCODE_1 GRIDCODE_1 VISIBLE
NONE;GRIDCODE_2 GRIDCODE_2 VISIBLE NONE;view_type2 view_type2
VISIBLE NONE")

# Process: Select Layer By Attribute (161)
arcpy.SelectLayerByAttribute_management(Layer_100__value_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses' AND
\"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (156)
arcpy.CalculateField_management(Layer_100__value__3_, "view_type2",
 "\"res_short\"", "VB", "")

# Process: Select Layer By Attribute (122)

```

```

arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_60_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'houses'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (118)
arcpy.CalculateField_management(Layer_100__value__4_, "view_type2",
 "\"res_tall\"", "VB", "")

# Process: Select Layer By Attribute (123)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_52_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (119)
arcpy.CalculateField_management(Layer_100__value__5_, "view_type2",
 "\"res_imperv\"", "VB", "")

# Process: Select Layer By Attribute (124)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_53_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'houses'
AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (120)
arcpy.CalculateField_management(Layer_100__value__6_, "view_type2",
 "\"res_build\"", "VB", "")

# Process: Select Layer By Attribute (129)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_54_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")

# Process: Calculate Field (124)
arcpy.CalculateField_management(Layer_100__value__7_, "view_type2",
 "\"res_build\"", "VB", "")

# Process: Select Layer By Attribute (125)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_59_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (121)
arcpy.CalculateField_management(Layer_100__value__8_, "view_type2",
 "\"ROW_short\"", "VB", "")

# Process: Select Layer By Attribute (126)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_55_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'Right of
Way' AND \"FID_buildi\" = -1")

# Process: Calculate Field (122)
arcpy.CalculateField_management(Layer_100__value__9_, "view_type2",
 "\"ROW_tall\"", "VB", "")

```

```

# Process: Select Layer By Attribute (127)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_56_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (123)
arcpy.CalculateField_management(Layer_100__value__10_,
"view_type2", "\"ROW_imperv\"", "VB", "")

# Process: Select Layer By Attribute (160)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4_57_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND NOT \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (155)
arcpy.CalculateField_management(Layer_100__value__11_,
"view_type2", "\"ROW_build\"", "VB", "")

# Process: Select Layer By Attribute (130)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_97_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'apartments'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (125)
arcpy.CalculateField_management(Layer_100__value__12_,
"view_type2", "\"apart_short\"", "VB", "")

# Process: Select Layer By Attribute (131)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__11
9_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'apartments' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (126)
arcpy.CalculateField_management(Layer_100__value__13_,
"view_type2", "\"apart_tall\"", "VB", "")

# Process: Select Layer By Attribute (132)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
3_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'apartments' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (127)
arcpy.CalculateField_management(Layer_100__value__14_,
"view_type2", "\"apart_imperv\"", "VB", "")

# Process: Select Layer By Attribute (133)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
4_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'apartments' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (128)

```

```
arcpy.CalculateField_management(Layer_100__value__15_,
"view_type2", "\"apart_build\"", "VB", "")

# Process: Select Layer By Attribute (134)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
5_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'apartments' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (129)
arcpy.CalculateField_management(Layer_100__value__16_,
"view_type2", "\"apart_build\"", "VB", "")

# Process: Select Layer By Attribute (135)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
7_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'commercial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (130)
arcpy.CalculateField_management(Layer_100__value__17_,
"view_type2", "\"comm_short\"", "VB", "")

# Process: Select Layer By Attribute (136)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
8_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'commercial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (131)
arcpy.CalculateField_management(Layer_100__value__18_,
"view_type2", "\"comm_tall\"", "VB", "")

# Process: Select Layer By Attribute (137)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__13
3_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'commercial' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (132)
arcpy.CalculateField_management(Layer_100__value__19_,
"view_type2", "\"comm_imperv\"", "VB", "")

# Process: Select Layer By Attribute (138)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__13
4_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'commercial' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (133)
arcpy.CalculateField_management(Layer_100__value__20_,
"view_type2", "\"comm_build\"", "VB", "")

# Process: Select Layer By Attribute (139)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__13
5_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
```

```

'commercial' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (134)
arcpy.CalculateField_management(Layer_100__value__21_,
"view_type2", "\"comm_build\"", "VB", "")

# Process: Select Layer By Attribute (140)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__13
7_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (135)
arcpy.CalculateField_management(Layer_100__value__22_,
"view_type2", "\"conser_short\"", "VB", "")

# Process: Select Layer By Attribute (141)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__13
8_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'conservation' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (136)
arcpy.CalculateField_management(Layer_100__value__23_,
"view_type2", "\"conser_tall\"", "VB", "")

# Process: Select Layer By Attribute (142)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__14
3_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (137)
arcpy.CalculateField_management(Layer_100__value__24_,
"view_type2", "\"conser_imperv\"", "VB", "")

# Process: Select Layer By Attribute (143)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__14
4_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'conservation' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (138)
arcpy.CalculateField_management(Layer_100__value__25_,
"view_type2", "\"conser_build\"", "VB", "")

# Process: Select Layer By Attribute (144)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__14
5_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" =
-1")

# Process: Calculate Field (139)
arcpy.CalculateField_management(Layer_100__value__26_,
"view_type2", "\"conser_build\"", "VB", "")

```

```
# Process: Select Layer By Attribute (145)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__14
7_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (140)
arcpy.CalculateField_management(Layer_100__value__27_,
"view_type2", "\"insti_short\"", "VB", "")

# Process: Select Layer By Attribute (146)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__14
8_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'institutional' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (141)
arcpy.CalculateField_management(Layer_100__value__28_,
"view_type2", "\"insti_tall\"", "VB", "")

# Process: Select Layer By Attribute (147)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__15
3_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -
1")

# Process: Calculate Field (142)
arcpy.CalculateField_management(Layer_100__value__29_,
"view_type2", "\"insti_imperv\"", "VB", "")

# Process: Select Layer By Attribute (148)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__15
4_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'institutional' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (143)
arcpy.CalculateField_management(Layer_100__value__76_,
"view_type2", "\"insti_build\"", "VB", "")

# Process: Select Layer By Attribute (149)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__15
5_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" =
-1")

# Process: Calculate Field (144)
arcpy.CalculateField_management(Layer_100__value__31_,
"view_type2", "\"insti_build\"", "VB", "")

# Process: Select Layer By Attribute (150)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__15
7_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'forestry' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")
```

```
# Process: Calculate Field (145)
arcpy.CalculateField_management(Layer_100__value__32_,
"view_type2", "\"forest_short\"", "VB", "")

# Process: Select Layer By Attribute (151)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_76_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'forestry'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (146)
arcpy.CalculateField_management(Layer_100__value__33_,
"view_type2", "\"forest_tall\"", "VB", "")

# Process: Select Layer By Attribute (152)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_81_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'forestry'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (147)
arcpy.CalculateField_management(Layer_100__value__34_,
"view_type2", "\"forest_imperv\"", "VB", "")

# Process: Select Layer By Attribute (153)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_82_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'forestry'
AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (148)
arcpy.CalculateField_management(Layer_100__value__35_,
"view_type2", "\"forest_build\"", "VB", "")

# Process: Select Layer By Attribute (154)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_83_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'forestry'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")

# Process: Calculate Field (149)
arcpy.CalculateField_management(Layer_100__value__36_,
"view_type2", "\"forest_build\"", "VB", "")

# Process: Select Layer By Attribute (155)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_85_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'industrial'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (150)
arcpy.CalculateField_management(Layer_100__value__37_,
"view_type2", "\"indus_short\"", "VB", "")

# Process: Select Layer By Attribute (156)
```

```

arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__86_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (151)
arcpy.CalculateField_management(Layer_100__value__38_,
"view_type2", "\"indus_tall\"", "VB", "")

# Process: Select Layer By Attribute (157)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__91_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'industrial' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (152)
arcpy.CalculateField_management(Layer_100__value__39_,
"view_type2", "\"indus_imperv\"", "VB", "")

# Process: Select Layer By Attribute (158)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__92_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (153)
arcpy.CalculateField_management(Layer_100__value__40_,
"view_type2", "\"indus_build\"", "VB", "")

# Process: Select Layer By Attribute (159)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__93_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'industrial' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (154)
arcpy.CalculateField_management(Layer_100__value__41_,
"view_type2", "\"indus_build\"", "VB", "")

# Process: Select Layer By Attribute (2)
arcpy.SelectLayerByAttribute_management(Layer_100__value__51_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (2)
arcpy.CalculateField_management(Layer_100__value__47_,
"view_type2", "\"farm_tall\"", "VB", "")

# Process: Select Layer By Attribute (3)
arcpy.SelectLayerByAttribute_management(Layer_100__value__43_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'farmland'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (3)

```

```

arcpy.CalculateField_management(Layer_100__value__50_,
"view_type2", "\"farm_imperv\"", "VB", "")

# Process: Select Layer By Attribute (4)
arcpy.SelectLayerByAttribute_management(Layer_100__value__44_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'farmland'
AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (4)
arcpy.CalculateField_management(Layer_100__value__49_,
"view_type2", "\"farm_build\"", "VB", "")

# Process: Select Layer By Attribute (5)
arcpy.SelectLayerByAttribute_management(Layer_100__value__48_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'farmland'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")

# Process: Calculate Field (5)
arcpy.CalculateField_management(Layer_100__value__45_,
"view_type2", "\"farm_build\"", "VB", "")

# Process: Select Layer By Attribute (128)
arcpy.SelectLayerByAttribute_management(Layer_100__value__42_,
"CLEAR_SELECTION", "")

# Process: Erase
arcpy.Erase_analysis(five__value__shp, hundred__value__shp,
doughnuts_five__value__shp, "")

# Process: Intersect (5)
arcpy.Intersect_analysis("C:\\Users\\Jacob\\Documents\\ArcGIS\\Defau
lt.gdb\\five_tree_50_utm
#;C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
\doughnuts_five_%value%.shp
#;C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\viewsheds\\
\\select7_%value%.shp #", intersect_500__value__shp, "ALL", "",
"INPUT")

# Process: Add Field (7)
arcpy.AddField_management(intersect_500__value__shp, "view_type2",
"TEXT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Make Feature Layer (7)
arcpy.MakeFeatureLayer_management(intersect_500__value__shp_2_,
Layer_500__value__, "", "", "FID FID_VISIBLE NONE;FID_five_t
FID_five_t_VISIBLE NONE;Shape Shape_VISIBLE NONE;GRIDCODE GRIDCODE
VISIBLE NONE;broad_zone broad_zone_VISIBLE NONE;FID_imperv
FID_imperv_VISIBLE NONE;FID_buildi FID_buildi_VISIBLE
NONE;Shape_Leng Shape_Leng_VISIBLE NONE;Shape_Area Shape_Area
VISIBLE NONE;FID_doughn FID_doughn_VISIBLE NONE;CID CID_VISIBLE
NONE;BUFF_DIST BUFF_DIST_VISIBLE NONE;FID_select FID_select_VISIBLE
NONE;ID ID_VISIBLE NONE;GRIDCODE_1 GRIDCODE_1_VISIBLE

```

```
NONE;GRIDCODE_2 GRIDCODE_2 VISIBLE NONE;view_type2 view_type2
VISIBLE NONE")
```

```
# Process: Select Layer By Attribute (207)
arcpy.SelectLayerByAttribute_management(Layer_500__value_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses' AND
\"FID_imperv\" = -1 AND \"FID_buildi\" = -1")
```

```
# Process: Calculate Field (213)
arcpy.CalculateField_management(Layer_100__value___140_,
"view_type2", "\"res_short\"", "VB", "")
```

```
# Process: Select Layer By Attribute (169)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__47_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'houses'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")
```

```
# Process: Calculate Field (175)
arcpy.CalculateField_management(Layer_100__value___100_,
"view_type2", "\"res_tall\"", "VB", "")
```

```
# Process: Select Layer By Attribute (170)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__13_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")
```

```
# Process: Calculate Field (176)
arcpy.CalculateField_management(Layer_100__value___101_,
"view_type2", "\"res_imperv\"", "VB", "")
```

```
# Process: Select Layer By Attribute (171)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__14_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'houses'
AND NOT \"FID_imperv\" = -1 ")
```

```
# Process: Calculate Field (177)
arcpy.CalculateField_management(Layer_100__value___102_,
"view_type2", "\"res_build\"", "VB", "")
```

```
# Process: Select Layer By Attribute (175)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__17_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'houses'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")
```

```
# Process: Calculate Field (181)
arcpy.CalculateField_management(Layer_100__value___106_,
"view_type2", "\"res_build\"", "VB", "")
```

```
# Process: Select Layer By Attribute (172)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__46_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")
```

```
# Process: Calculate Field (178)
arcpy.CalculateField_management(Layer_100__value__103_,
"view_type2", "\"ROW_short\"", "VB", "")

# Process: Select Layer By Attribute (173)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__18_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'Right of
Way' AND \"FID_buildi\" = -1")

# Process: Calculate Field (179)
arcpy.CalculateField_management(Layer_100__value__104_,
"view_type2", "\"ROW_tall\"", "VB", "")

# Process: Select Layer By Attribute (174)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__19_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (180)
arcpy.CalculateField_management(Layer_100__value__105_,
"view_type2", "\"ROW_imperv\"", "VB", "")

# Process: Select Layer By Attribute (206)
arcpy.SelectLayerByAttribute_management(viewshed_intersect100_4__20_
, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'Right of
Way' AND NOT \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (212)
arcpy.CalculateField_management(Layer_100__value__138_,
"view_type2", "\"ROW_build\"", "VB", "")

# Process: Select Layer By Attribute (176)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__21_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'apartments'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (182)
arcpy.CalculateField_management(Layer_100__value__107_,
"view_type2", "\"apart_short\"", "VB", "")

# Process: Select Layer By Attribute (177)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__23_
, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'apartments' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (183)
arcpy.CalculateField_management(Layer_100__value__108_,
"view_type2", "\"apart_tall\"", "VB", "")

# Process: Select Layer By Attribute (178)
```

```
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__24
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'apartments' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (184)
arcpy.CalculateField_management(Layer_100__value__109_,
"view_type2", "\"apart_imperv\"", "VB", "")

# Process: Select Layer By Attribute (179)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__25
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'apartments' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (185)
arcpy.CalculateField_management(Layer_100__value__110_,
"view_type2", "\"apart_build\"", "VB", "")

# Process: Select Layer By Attribute (180)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__26
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'apartments' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (186)
arcpy.CalculateField_management(Layer_100__value__111_,
"view_type2", "\"apart_build\"", "VB", "")

# Process: Select Layer By Attribute (181)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__27
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'commercial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (187)
arcpy.CalculateField_management(Layer_100__value__112_,
"view_type2", "\"comm_short\"", "VB", "")

# Process: Select Layer By Attribute (182)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__28
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'commercial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (188)
arcpy.CalculateField_management(Layer_100__value__113_,
"view_type2", "\"comm_tall\"", "VB", "")

# Process: Select Layer By Attribute (183)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__29
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'commercial' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (189)
```

```
arcpy.CalculateField_management(Layer_100__value__114_,
"view_type2", "\"comm_imperv\"", "VB", "")

# Process: Select Layer By Attribute (184)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__30
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'commercial' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (190)
arcpy.CalculateField_management(Layer_100__value__115_,
"view_type2", "\"comm_build\"", "VB", "")

# Process: Select Layer By Attribute (185)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__31
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'commercial' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (191)
arcpy.CalculateField_management(Layer_100__value__116_,
"view_type2", "\"comm_build\"", "VB", "")

# Process: Select Layer By Attribute (186)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__32
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (192)
arcpy.CalculateField_management(Layer_100__value__117_,
"view_type2", "\"conser_short\"", "VB", "")

# Process: Select Layer By Attribute (187)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__33
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'conservation' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (193)
arcpy.CalculateField_management(Layer_100__value__118_,
"view_type2", "\"conser_tall\"", "VB", "")

# Process: Select Layer By Attribute (188)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__34
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (194)
arcpy.CalculateField_management(Layer_100__value__119_,
"view_type2", "\"conser_imperv\"", "VB", "")

# Process: Select Layer By Attribute (189)
```

```

arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__35
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'conservation' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (195)
arcpy.CalculateField_management(Layer_100__value__120_,
"view_type2", "\"conser_build\"", "VB", "")

# Process: Select Layer By Attribute (190)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__36
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'conservation' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" =
-1")

# Process: Calculate Field (196)
arcpy.CalculateField_management(Layer_100__value__121_,
"view_type2", "\"conser_build\"", "VB", "")

# Process: Select Layer By Attribute (191)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__37
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (197)
arcpy.CalculateField_management(Layer_100__value__122_,
"view_type2", "\"insti_short\"", "VB", "")

# Process: Select Layer By Attribute (192)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__38
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'institutional' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (198)
arcpy.CalculateField_management(Layer_100__value__123_,
"view_type2", "\"insti_tall\"", "VB", "")

# Process: Select Layer By Attribute (193)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__39
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -
1")

# Process: Calculate Field (199)
arcpy.CalculateField_management(Layer_100__value__124_,
"view_type2", "\"insti_imperv\"", "VB", "")

# Process: Select Layer By Attribute (194)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__40
_, "NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'institutional' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (200)

```

```
arcpy.CalculateField_management(Layer_500__value__3_, "view_type2",
"\insti_build\\"", "VB", "")

# Process: Select Layer By Attribute (195)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__41
_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'institutional' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" =
-1")

# Process: Calculate Field (201)
arcpy.CalculateField_management(Layer_100__value__126_,
"view_type2", "\"insti_build\\"", "VB", "")

# Process: Select Layer By Attribute (196)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_half4__12
0_, "NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'forestry' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (202)
arcpy.CalculateField_management(Layer_100__value__127_,
"view_type2", "\"forest_short\\"", "VB", "")

# Process: Select Layer By Attribute (197)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__12_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'forestry'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (203)
arcpy.CalculateField_management(Layer_100__value__128_,
"view_type2", "\"forest_tall\\"", "VB", "")

# Process: Select Layer By Attribute (198)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__13_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'forestry'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (204)
arcpy.CalculateField_management(Layer_100__value__129_,
"view_type2", "\"forest_imperv\\"", "VB", "")

# Process: Select Layer By Attribute (199)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3__14_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'forestry'
AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (205)
arcpy.CalculateField_management(Layer_100__value__130_,
"view_type2", "\"forest_build\\"", "VB", "")

# Process: Select Layer By Attribute (200)
```

```

arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_15_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'forestry'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")

# Process: Calculate Field (206)
arcpy.CalculateField_management(Layer_100__value__131_,
"view_type2", "\"forest_build\"", "VB", "")

# Process: Select Layer By Attribute (201)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_16_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'industrial'
AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (207)
arcpy.CalculateField_management(Layer_100__value__132_,
"view_type2", "\"indus_short\"", "VB", "")

# Process: Select Layer By Attribute (202)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_17_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (208)
arcpy.CalculateField_management(Layer_100__value__133_,
"view_type2", "\"indus_tall\"", "VB", "")

# Process: Select Layer By Attribute (203)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_18_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'industrial' AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (209)
arcpy.CalculateField_management(Layer_100__value__134_,
"view_type2", "\"indus_imperv\"", "VB", "")

# Process: Select Layer By Attribute (204)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_19_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (210)
arcpy.CalculateField_management(Layer_100__value__135_,
"view_type2", "\"indus_build\"", "VB", "")

# Process: Select Layer By Attribute (205)
arcpy.SelectLayerByAttribute_management(viewshed_intersect_3_3_20_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" =
'industrial' AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -
1")

# Process: Calculate Field (211)

```

```

arcpy.CalculateField_management(Layer_100__value__136_,
"view_type2", "\"indus_build\"", "VB", "")

# Process: Select Layer By Attribute (208)
arcpy.SelectLayerByAttribute_management(Layer_100__value__137_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" =
'industrial' AND \"FID_imperv\" = -1 AND \"FID_buildi\" = -1")

# Process: Calculate Field (214)
arcpy.CalculateField_management(Layer_100__value__141_,
"view_type2", "\"farm_tall\"", "VB", "")

# Process: Select Layer By Attribute (209)
arcpy.SelectLayerByAttribute_management(Layer_100__value__144_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'farmland'
AND \"FID_buildi\" = -1 AND NOT \"FID_imperv\" = -1")

# Process: Calculate Field (215)
arcpy.CalculateField_management(Layer_100__value__142_,
"view_type2", "\"farm_imperv\"", "VB", "")

# Process: Select Layer By Attribute (210)
arcpy.SelectLayerByAttribute_management(Layer_100__value__145_,
"NEW_SELECTION", "\"GRIDCODE\" = 2 AND \"broad_zone\" = 'farmland'
AND NOT \"FID_imperv\" = -1 ")

# Process: Calculate Field (216)
arcpy.CalculateField_management(Layer_100__value__143_,
"view_type2", "\"farm_build\"", "VB", "")

# Process: Select Layer By Attribute (211)
arcpy.SelectLayerByAttribute_management(Layer_100__value__146_,
"NEW_SELECTION", "\"GRIDCODE\" = 1 AND \"broad_zone\" = 'farmland'
AND NOT \"FID_imperv\" = -1 AND NOT \"FID_buildi\" = -1")

# Process: Calculate Field (217)
arcpy.CalculateField_management(Layer_100__value__147_,
"view_type2", "\"farm_build\"", "VB", "")

# Process: Select Layer By Attribute (43)
arcpy.SelectLayerByAttribute_management(Layer_100__value__148_,
"CLEAR_SELECTION", "")

# Process: Dissolve (3)
arcpy.Dissolve_management(intersect_500__value__shp__4_,
dissolve_500__value__shp, "view_type2", "", "MULTI_PART",
"DISSOLVE_LINES")

# Process: Add Field (8)
arcpy.AddField_management(dissolve_500__value__shp, "area",
"DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

```

```
# Process: Calculate Field (42)
arcpy.CalculateField_management(v100_dissolve__value__shp__4_,
"area", "!Shape.Area!", "PYTHON_9.3", "")

# Process: Add Field (9)
arcpy.AddField_management(v100ft_dissolve2__4_, "perc_2", "FLOAT",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (43)
arcpy.CalculateField_management(dissolve_500__value__4_, "perc_2",
"!area! / 90000", "PYTHON_9.3", "")

# Process: Dissolve (4)
arcpy.Dissolve_management(intersect_100__value__shp__2_,
dissolve_100__value__shp, "view_type2", "", "MULTI_PART",
"DISSOLVE_LINES")

# Process: Add Field (12)
arcpy.AddField_management(dissolve_100__value__shp, "area",
"DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (164)
arcpy.CalculateField_management(v100_dissolve__value__shp__2_,
"area", "!Shape.Area!", "PYTHON_9.3", "")

# Process: Add Field (13)
arcpy.AddField_management(v100ft_dissolve2__8_, "perc_1", "FLOAT",
"", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (165)
arcpy.CalculateField_management(dissolve_100__value__shp__2_,
"perc_1", "!area! / 2500", "PYTHON_9.3", "")

# Process: Join Field (2)
arcpy.JoinField_management(dissolve_500__value__2_, "view_type2",
dissolve_100__value__shp__3_, "view_type2", "perc_1")

# Process: Add Field (5)
arcpy.AddField_management(dissolve_500__value__5_, "perc_total",
"DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field (157)
arcpy.CalculateField_management(dissolve_500__value__shp__2_,
"perc_total", "[perc_1] + [perc_2]", "VB", "")

# Process: Add Field (14)
arcpy.AddField_management(dissolve_500__value__shp__3_,
"identifier", "DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED",
"")

# Process: Calculate Field (166)
```

```

arcpy.CalculateField_management(two_dissolve__value__shp__8_,
"identifrier", "%value%", "VB", "")

# Process: Add Field
arcpy.AddField_management(two_dissolve__value__shp__4_, "close_",
"DOUBLE", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

# Process: Calculate Field
arcpy.CalculateField_management(dissolve_half__value__5_, "close_",
"sum(!perc_2!, !perc_1!)", "PYTHON_9.3", "")

# Process: Append
arcpy.Append_management("C:\\Users\\Jacob\\Documents\\seagate\\docs\\
\\thesis_GIS\\viewsheds\\dissolve_2\\dissolve_500_%value%.shp",
dissolve_500_8_shp, "NO_TEST", "view_type2 \\\"view_type2\\\" true false
false 254 Text 0 0
,First,#,C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\vie
wsheds\\dissolve_2\\dissolve_500_%value%.shp,view_type2,-1,-
1;Shape_Leng \\\"Shape_Leng\\\" true false false 19 Double 0 0
,First,#;Shape_Area \\\"Shape_Area\\\" true false false 19 Double 0 0
,First,#;area \\\"area\\\" true false false 19 Double 0 0
,First,#,C:\\Users\\Jacob\\Documents\\seagate\\docs\\thesis_GIS\\vie
wsheds\\dissolve_2\\dissolve_500_%value%.shp,area,-1,-1;percieved
\\\"percieved\\\" true false false 13 Float 0 0 ,First,#;percieved_
\\\"percieved_\\\" true false false 13 Float 0 0 ,First,#;percieved1
\\\"percieved1\\\" true false false 13 Float 0 0 ,First,#;percieve_1
\\\"percieve_1\\\" true false false 13 Float 0 0 ,First,#", "")

```

