Summer 2022

Shadow-Based Automatic Building Height Estimation From High Spatial Resolution Satellite Imagery

Lonnie Lee Byrnside III
Missouri State University, Byrnside23@live.missouristate.edu

SHADOW-BASED AUTOMATIC BUILDING HEIGHT ESTIMATION FROM HIGH SPATIAL RESOLUTION SATELLITE IMAGERY

A Master's Thesis

Presented to

The Graduate College of

Missouri State University

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science, Geography and Geology

By

Lonnie L. Byrnside III

August 2022

SHADOW-BASED AUTOMATIC BULDING HEIGHT ESTIMATION FROM HIGH SPATIAL RESOLUTION SATELLITE IMAGERY

Geography, Geology and Planning

Missouri State University, August 2022

Master of Science

Lonnie L. Byrnside III

ABSTRACT

Three-dimensional city (3D) models are very useful in supporting natural disaster preparation and response. LiDAR surveying is currently the main method by which 3D city models are created; however, LiDAR data on a local scale is hard to obtain for developing countries. This project sought to test whether or not urban feature height data obtained using the photogrammetric sun-angle shadow method is a viable alternative to LiDAR-derived 3D city models. A core element of this work was the development of a toolset to be shared freely to the public to promote crowdsourcing of 3D building data. Prior works were reviewed and a shadow-overlapping method for estimating building heights was selected and implemented in the Python programming language. Shadow detection methods in the literature were also reviewed and eight were modified into a simple command-line Python tool to batch process shadow detection using multiple algorithms. The Missouri State University, Springfield, MO, campus was selected as the study site for testing the shadow-overlapping process and LiDAR/DEM data was used to create building footprints with ground truth height values. For the shadow detection methods that produced the most accurate results, roughly 60% of building height estimates were within 10 feet (or one floor) of the true height, and buildings whose heights were between 37-50 feet consistently had the lowest margins of error. A systematic finding, however, was that shorter buildings' heights were overestimated and taller buildings were underestimated. A similar pattern was identified with building size/square footage with smaller buildings being overestimated and larger buildings being underestimated. In the end, the results suggested that the shadow-overlapping method is likely not reliable enough to produce height estimations comparable to LiDAR-derived methods and that these height estimates are not suitable for downstream calculations. However, for a simple/generalized 3D cartographic representation of an area, it appeared that this low-cost method could produce adequate results.

KEYWORDS: 3D city models, building height estimation, GIS, photogrammetry, Python, shadows

SHADOW-BASED AUTOMATIC BUILDING HEIGHT ESTIMATION

FROM HIGH SPATIAL RESOLUTION SATELLITE IMAGERY

By

Lonnie L. Byrnside III

A Master's Thesis
Submitted to the Graduate College
Of Missouri State University
In Partial Fulfillment of the Requirements
For the Degree of Master of Science, Geography and Geology

August 2022

Approved:

Xin Miao, Ph.D., Thesis Committee Chair

Toby Dogwiler, Ph.D., Committee Member

Jun Luo, Ph.D., Committee Member

Julie Masterson, Ph.D., Dean of the Graduate College

ACKNOWLEDGEMENTS

First, I would like to thank the phenomenal faculty of the Department of Geography, Geology and Planning for their support and guidance during my time as an undergraduate and graduate student at Missouri State University. I am proud to say I am an MSU alumnus and feel confident as I enter the workforce as a young geospatial professional because of the academic foundation laid by my GGP professors—especially those on my thesis committee. Specifically, I must recognize Dr. Xin Miao, my advisor, for his unwavering support throughout my academic career and the opportunity to serve as an undergraduate researcher on his Arctic sea ice project for a short time. The skills I acquired during this experience were critical in preparing me for the work undertaken in this paper. I also want to extend my gratitude to Dr. Toby Dogwiler and Dr. Douglas Gouzie for having been fantastic resources throughout my time at MSU as I navigated academic requirements and the Graduate College. Lastly, a thank you to Dr. Jun Luo for your coursework in web GIS, which got me interested in GIS programming, and the time you have given to be on my committee.

Lastly, I would like to thank my parents, Lonnie Byrnside Jr., and Kristine Byrnside, as well as my twin sister, Abigail Byrnside, for their love, support, and sacrifices made throughout the years. Without you all, I would have never had many of the opportunities and experiences that shaped me into the person I am today. *Thank you.*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Geospatial data is playing an ever-increasing role in natural disaster prevention and response [1]. Today, geospatial analysis can help answer questions such as “Who or what infrastructure is most vulnerable?”, “Where is the best evacuation route?”, and “Where should relief resources be positioned to best support those in need?” These questions form the key information requirements during a crisis event. Three-dimensional city (3D) city models, in particular, allow emergency responders to interact with and visualize the environment, infrastructure, and people using a more realistic world view than 2D maps, giving them a greater ability to find hidden relationships and subtle patterns [2]. For example, 3D building models can be used to estimate building occupancy or storage capacity—valuable information that can assist planners of relief efforts. Perhaps even more importantly, they also allow emergency responders to visualize the operating environment prior to a crisis response operation. One mainstream example of this was the successful raid on Osama bin Laden’s compound in 2011 by US Navy SEALs who, after the event, remarked that 3D geospatial visualizations helped them “feel like they had been there before” [3]. Lastly, using 3D data, scientists and engineers are also better equipped to model and simulate the impacts of natural disasters, such as floods and earthquakes, under a variety of conditions before they strike [4]—a potential lifesaver.

3D Building Data Sources

Light Detection and Ranging (LiDAR) aerial laser scan technology is currently one of the most widely used remote sensing methods of collecting the data used to create 3D city models [5]. However, in developing nations, which are already disproportionately affected by natural

disasters, LiDAR and other GIS data on a local scale is not always readily available, let alone data that is current and accurate [6]. The lack of available 3D data for such regions hinders the geospatial community's ability to support disaster analysis and response.

With that being said, LiDAR is not the only method of obtaining 3D data of man-made features, as object heights can also be calculated using the photogrammetric technique of measuring heights on single aerial photos using the sun-angle shadow method [7]. By utilizing the solar elevation angle at the time of image capture (a known value stored in the image's metadata) and by measuring the length of an object's cast shadow, its height can be calculated using the trigonometric formula from [7] shown in Fig. 1**.** Then, to create simple 3D city models using this information, building footprint polygons can be extruded vertically to the estimated height values, as seen in Fig. 2.

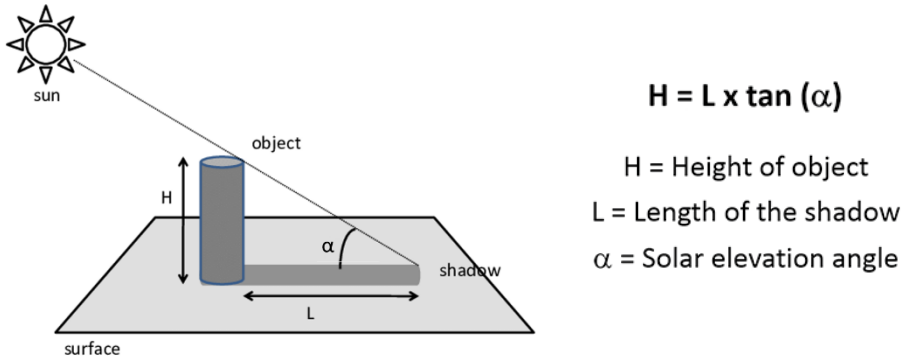


Fig. 1. Formula to calculate object height using shadow-based photogrammetry method.

Objective

While the potential for deriving 3D building information from shadow-based photogrammetry methods has been described in the literature, there does not currently exist a robust, commercially available tool for geospatial professionals to create this data in any easily

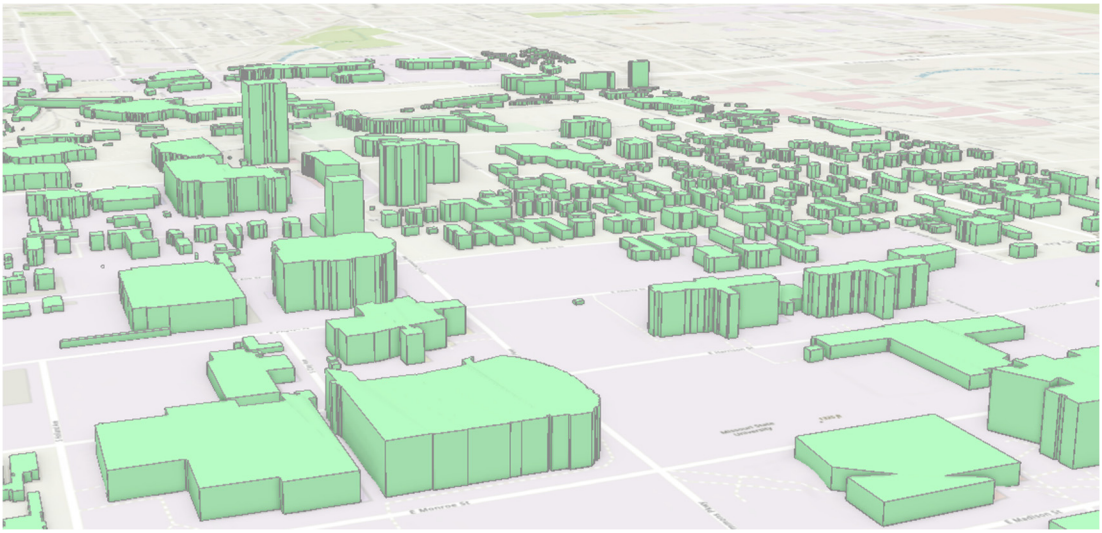


Fig. 2. Rudimentary 3D model of Springfield, MO, created by vertically extruding footprints.

repeatable way or at scale. As such, a major goal was to implement algorithms presented in prior works and combine them into one open-source Python toolset to be shared freely in the public domain so future researchers can quickly compare results between different shadow methods as well as fine tune parameters without having to "reinvent the wheel". Lastly, it must be acknowledged that the process outlined in this paper assumes vector building footprints are available for the study area and does not attempt any automated bulding footprint extraction. The goal in this research is only to test the feasibility of extracting and applying height metadata to existing footprints. Even so, it should be noted that GIS building footprint data without height attribution is widely and freely available to the public through crowdourced projects like OpenStreetMap [8] and through Microsoft's AI Assisted Mapping project [9].

Hypothesis

While shadow-based methods are not likely able to match the granularity of 3D meshes derived from LiDAR point clouds, the hypothesis explored here was that shadow-based building

height estimation from high spatial resolution (HSR) satellite imagery could provide comparable results to LiDAR-derived techniques in terms of the basic height attribution necessary to create rudimentary 3D city models.

LITERATURE REVIEW

The literature review for this study consists of two parts—an overview of different automated methods for building height estimation from satellite imagery and an overview of different shadow detection algorithms.

A Brief Note on Imagery

In satellite imagery, a common finding for the remote sensing professional is that shadow regions are less illuminated than their surroundings, and as a result, have lower values for red, green, and blue (RGB), as well as graylevel intensity, standard deviation, variance, local maximum, and brightness [10]. However, since other dark colored objects in the imagery usually have similar characteristics, it can often be difficult to discriminate between the two. This is especially the case in panchromatic imagery, a single-band image type that depicts features solely in shades of black and white. Generally, the more spectral bands, the more information is gathered by the sensor. As such, true color RGB imagery and imagery with a near infrared (NIR) band can provide more data which can be used to detect shadow regions. In fact, a NIR band can be used in index calculations such as NDVI (Normalized Difference Vegetation Index) to assist in filtering out vegetation from candidate shadow regions—reducing the likelihood of misclassification as shadow pixels [11]. While true color RGB imagery does not have this capability, it does have the greatest availablilty, and in alignment with the goal to promote crowdsourcing of global building height data, developing a method using RGB imagery seemed most prudent in this study.

Building Height Estimation Methods

Many prior scholars have studied the automation of building height estimation from HSR satellite imagery. For this paper, four categories of algorithms have been established—measuring shadow length, overlapping artificial shadows, stereo-pair photogrammetry, and depth estimation in computer vision—with the first two being shadow methods and the second two being non-shadow methods.

Measuring Shadow Length. The first method of estimating building height is to simply measure the length of building shadows within a pre-defined azimuth angle range followed by height calculation using the shadow formula. In one method, shadow objects were segmented using supervised classification of the panchromatic and multispectral bands of IKONOS images [12]. The authors visually observed that the shadows from taller objects (high-rise buildings) in the image were more compact in shape than those from shorter objects (low-rise buildings and other non-building features in the image scene). As such, they used a compactness index of perimeter-area ratio on detected shadow features to filter out those not likely to be buildings (taller buildings/compact shadows having lower perimeter-area ratios). However, since smaller image objects also tend to have higher perimeter-area ratios than large image objects and the authors did not want to filter out important shadows that may have smaller pixel areas depending on the image, the index was sampled for 20 high-rise buildings and the highest perimeter-area ratio was then set as the threshold value to separate shadow objects of taller/larger buildings from smaller/shorter objects in the image scene [12]. Next, candidate shadow lengths were found by measuring the length of each shadow object at intervals between the pre-defined azimuth angle range (e.g., 315-345°). Then, the highest length within the angle range was selected. In Fig. 3, the selected building shadow length would be around 9 pixels long. A major advantage of this

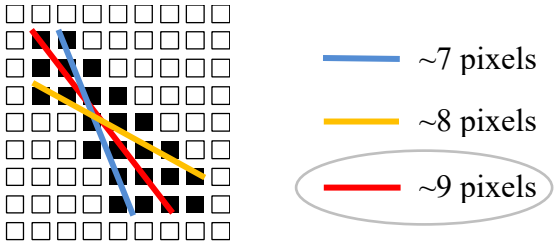


Fig. 3. Candidate shadow lengths within pre-defined azimuth angle range.

method is its simplicity—low computational complexity results in very fast processing times. However, a problem specific to this algorithm is that it frequently fails to be able to calculate heights for low-rise buildings and the authors report that it systematically overestimates the remaining buildings [12]. It is also concerning that this method assumes the sun's illumination is perpendicular to the side of a building (resulting in a single parallelogram-shaped shadow along one side of a building). Additional testing would need to occur to verify if perimeter-area index assumptions would also hold true in the cases where the sun's illumination falls onto the corners of buildings (resulting in connected shadowing from two or more sides of a building forming an "r" or "L" shape). In the latter case, programmatically choosing the start location for measuring the shadow object's length in the direction of the azimuth angle might also be more difficult so as to not measure the building's length in addition to shadow length. Another limitation of this algorithm that the authors conceded was that it assumes the shadows fall onto a flat surface, which can result in invalid results when the shadows of high-rise buildings fall onto other high-rise buildings [12]. Further, while not acknowledged by the original authors, it is assessed that, instead of directly using the known azimuth angle, a sun azimuth angle range was likely used in order to account for terrain variation, which could have made shadows appear to deviate slightly from the azimuth angle.

Overlapping Artificial Shadows. Another shadow-based method for estimating building heights involves generating artificial (or simulated) shadows for buildings at various candidate heights and comparing their similarity to the actual shadow pixels. In one paper, shadow regions and building footprints are first extracted from HSR satellite imagery [13]. Then, for each footprint, an artificial shadow is generated given the solar elevation angle from the image's metadata and a pre-defined initial building height. Next, the artificial shadow is compared to the actual shadow pixels using the Jaccard similarity index (intersection over union metric) [14]. The statistic ranges in value from 0 to 1, with more accurate candidate heights returning a score closer to 1. Using an exhaustive search process, additional artificial shadows are generated for the footprint with the candidate height incremented on each iteration until the Jaccard index is maximized. Once that occurs, the process stops and the candidate height associated with the highest Jaccard value is set as the building's final estimated height. Fig. 4 illustrates this process with one building—the process would then continue for any other building footprints in the image.

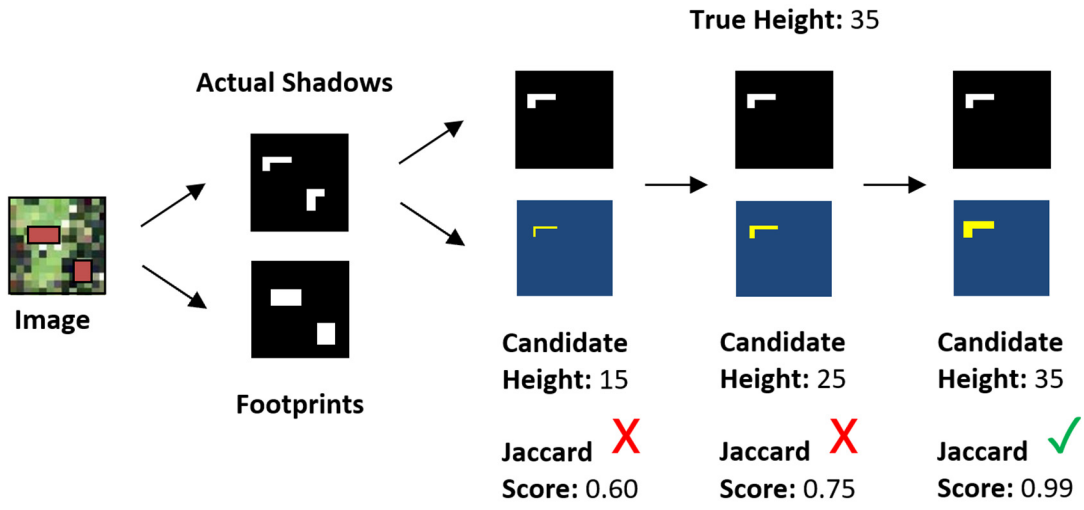


Fig. 4. Process flow diagram of the overlapping artificial shadows algorithm.

The authors reported encouraging results with this method, but it must be noted that an exhaustive search for candidate building heights significantly increases computation time. Instead of an exhaustive search, other scholars have used optimization algorithms to select candidate height values more efficiently. In [15], fuzzy reasoning and a genetic algorithm was used for this step. Genetic algorithms rely on operators inspired by evolution such as mutation, crossover, and selection, to generate solutions to optimization and search problems [16]. In [14], the authors used a gradient descent method, a first-order iterative optimization algorithm based on a convex function and adjusts its parameters iteratively to minimize a given function to its local minimum [17]. As is the case with the other shadow method, the robustness of this algorithm is also limited by the fact it assumes building shadows fall onto flat surfaces and not falling onto nearby or adjacent buildings. (Additional limitations of the shadow-overlapping method are acknowledged in the "Discussion" chapter of this paper.)

Stereo-pair Photogrammetry. This next method uses satellite image stereo pairs to obtain 3D city models from digital surface models (DSMs). In one study, this was done using the ERDAS IMAGINE Automatic Terrain Extraction with Dense Point Matching (eATE) software; an example stereo pair used by the authors in [18] can be seen in Fig. 5. The dense image matching procedure of the eATE tool finds the corresponding points for almost every pixel of the stereo pair and estimates their 3D coordinates in object space.

Stereo-pair photogrammetry is capable of creating highly detailed 3D city models of not just buildings, but trees and other natural features as well. However, while the ERDAS IMAGINE software used in this paper is non-free along with their proprietary algorithms for dense point matching, the general capability of automatic tie point extraction/automatic aerial triangulation is included in free photogrammetry software such as Meshroom or LISA-FOTO.

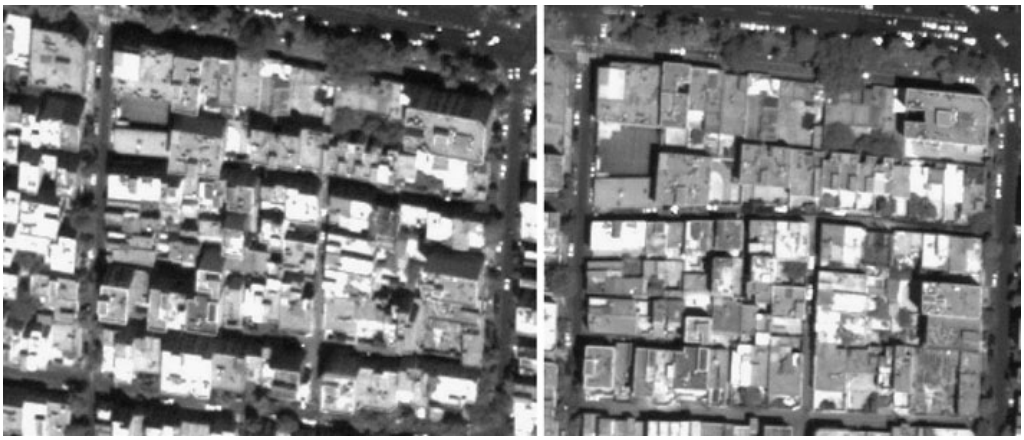


Fig. 5. Stereo pair of satellite images for an urban study area.

With that being said, this method using stereo pairs of VHR satellite images is generally more cost prohibitive as the images are more difficult to obtain than monocular satellite images, which can be found freely and in much greater quantities [14].

Depth Estimation in Computer Vision. However, when only a single image is available, it is not possible conduct 3D reconstruction using stereo geometry and its related photogrammetric techniques [19]. Instead, depth estimation algorithms relying on contextual cues (size, texture, position) can make strong assumptions about scene geometry. In the method proposed in [20], a depth estimation algorithm was used to create DSMs for urban areas, with examples of inputs shown in Fig. 6. Their algorithm relied on the idea that abrupt transitions in height often corresponded with abrupt changes in object type classification/labeling between adjacent objects and vice versa. With advances in deep learning and Convolutional Neural Networks (CNNs), it is now possible for these algorithms to learn these contextual cues from training data rather than by programming them in manually [19]. In both [20] and [21], the deep learning models were trained using ground truth-labeled DSMs.

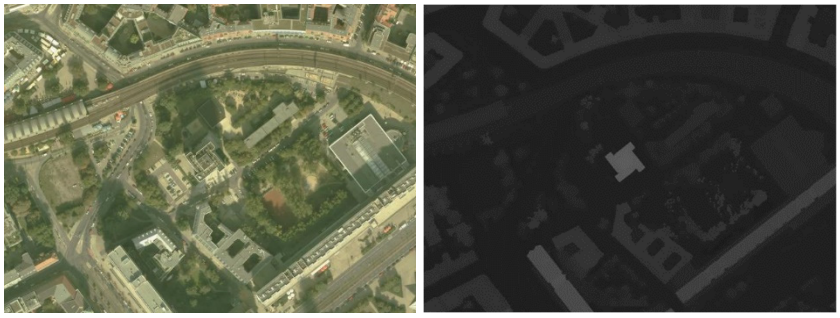


Fig. 6. Satellite image and its associated ground truth-labeled DSM for an urban study area.

Just as stereo-pair photogrammetry was able to, these depth estimation-based algorithms are also capable of creating 3D models of more than just buildings. However, disadvantages of this method include the high level of complexity involved with implementing deep learning models. Additionally, the time and resources required to train the model would also be significant. Lastly, these depth-estimation algorithms make assumptions instead of considering known information such as azimuth and elevation angle of the sun and may introduce errors when using images from different data sources (e.g., the appearance and location of rooftops in orthorectified photos will differ from images affected by relief displacement).

Applications of Prior Works in This Paper. The decision to use a shadow-based method was in part due to the vast quanity of monocular HSR satellite imagery available to geospatial professionals today, much of which can be easily obtained and spanning the majority of the world. Another potential advantage of methods using monocular satellite imagery is that change detection and analysis may be possible using the vast archives of historical monocular imagery whereas methods involving LiDAR and stereo imagery are limited by their more limited temporal and regional coverage. Specifically, the shadow-overlapping algorithm proposed in 2018 by N. Kadhim and M. Mourshed [13] was used in this paper's experimentation. The benefits of this algorithm over the other three discussed previously are that it only requires one

image, it is not very technically complex while still yielding highly accurate results for both low-rise and high-rise buildings, and that it allows for flexible integration with a variety of different shadow and footprint detection methods.

Shadow Detection Methods

The second major consideration when developing a shadow-based algorithm is the method by which shadows are extracted from source imagery. Four classes of varying levels of complexity have been identified in [10]—thresholding-based shadow detection, color model-based shadow detection, region-based shadow detection, and feature learning-based shadow detection—with the first two being pixel-based classification methods and the second two being object-based classification methods.

Thresholding-based Shadow Detection. Thresholding is currently the most widely used method of shadow detection. The advantage of this method is the simplicity and speed with which shadow regions can be identified since it works by simply applying mathematical operations to pixel values in the image [10]. In essence, it works by further darkening the dark parts of an image and lightening the light parts. However, due to the pixel similarity between shadow regions and dark features (e.g., water bodies, dark rooftops, fresh asphalt), it can be difficult to select suitable thresholds that will correctly identify shadow regions across a diverse range of geographic environments [22].

The authors in [13] used thresholding on imagery with both visible and NIR bands to help avoid the issue of dark colored features being mistaken for shadows. For each pixel, the proposed process computes the ratio between the visible and NIR bands. This works to discriminate between shadow regions and dark colored objects as objects that are dark in the

visible spectrum often have a much higher reflectance in the NIR band, and shadows would be dark in both [23]. Their method also uses a NDVI to filter out vegetation, producing a fairly accurate result at the cost of requiring multispectral imagery.

The first law of geography, as stated by Waldo R. Tobler, states that "everything is related to everything else, but near things are more related than distant things" [24]. This phenomenon, where values of a common variable are similar across nearby locations, is also called positive spatial autocorrelation. However, in general, pixel-based classification often fails to account for spatial autocorrelation—especially in many thresholding-based methods where a filter is globally applied to the image without considering local contextual clues and relationships [25]. Furthermore, many pixel-based analytic methods were developed for relatively coarse spatial resolution imagery, and with the increase of spatial resolution (such as with HSR images used in this paper's study), the accuracy of classification may decrease due to the increase of within-class variability [25]. This variability, which can further be described as high local spatial heterogeneity, often results in "salt-and-pepper" noise in the pixel-based classification result [26] and the application of a median filter may be required to reduce noise.

Color Model-based Shadow Detection. The transformation of an image from one color model to another can provide additional information to help detect shadows. Color spaces such as HSI (Hue, Saturation, Intensity) and c1c2c3 are commonly used because they produce images invariant to shadows [27] and CIELAB has gained popularity due to its similarity with human perception [10].

A cornerstone piece of research on shadow identification and classification using invariant color models was conducted in [28]. In their method, after initial RGB conversion to the c1c2c3 color invariant model, a Sobel operator is applied on the luminance component of the

input image to perform edge detection. The generated edge-map is used to find object contours and to extract candidate shadow regions. Dark regions that are not contained in the object contours are classified as cast shadow regions. A few limitations of this method, however, are that it generally only works for shadows cast on a flat, non-textured surface only, and that the process worked best for images with a single object according to the authors. Obviously, this would not be ideal for a dense urban landscape. As a whole, this method is better utilized in combination with other classes of shadow detection algorithms, rather than by itself.

Region-based Shadow Detection. This class of shadow detection algorithms seeks to solve the issue of classification not being adaptive to different regions across an image. In these methods, segmented regions are labeled as shadows or non-shadows based on the properties of individual regions as well as similarity with other regions, comparing properties such as color and texture histograms, intensity ratios, and distance between regions [10].

One significant implementation of a region-based method was proposed in [29]. In their research, a region growing algorithm utilizing two different color model transformations—RGB to c1c2c3 and RGB to HSV (Hue, Saturation, Value)—was used. Then, using a smoothed c3 channel, the saturation channel (S), and the gradient of the intensity channel (V) calculated using the Sobel operator, seed pixels are selected where they match the known properties of shadow regions. Next, the process recursively checks the eight neighbor pixels of a growing region's boundaries—adding them to the growing region if their saturation and intensity values are consistent with shadow pixels. Once no more neighbor pixels are added to a region, the growing algorithm stops. While this algorithm produces highly accurate results, a potential issue with this method is that a building's shadow will be missed entirely if at least one seed is not originally

detected in its shadow region, which can be an issue in image settings with many buildings depending on the density of starting seed points.

In sum, region-based shadow detection algorithms have the benefit of using contextual information to help identify shadowed areas in imagery and use more logic when classifying pixels as shadow pixels than simple thresholding, while not being as complex as feature learning-based methods.

Feature Learning-based Shadow Detection. Undoubtedly, the most complex algorithms for shadow detection in digital images are found in the subset of machine learning and deep learning-based methods. These methods are intended to solve some of the issues that arise in thresholding and color model-based algorithms such as difficulty in obtaining the optimal threshold value and distinguishing shadows from dark colored features [10]. Scholars in the field of computer vision have implemented approaches such as decision trees, neural networks, as well as supervised and unsupervised learning models to separate shadow and shadowless regions using trained classifiers. The authors in [25] used a machine learning random forest classifier on segmented Arctic sea ice imagery using a total of 13 spectral, texture, and spatial attributes to detect shadows. As an object-based classification technique, this method considers both spectral values of each pixel and spatial patterns that determine the texture, shape, and contextual properties of each feature, helping improve classification accuracy [25] and reducing "salt-and-pepper" noise by merging pixels into neighboring objects; however, these generalizations may come at the cost of losing some important detail information [26].

Furthermore, some feature learning-based shadow detection methods use predefined training data sets to identify shadows while others automatically learn the most relevant

characteristics of shadow regions at runtime (the discriminator between supervised and unsupervised classification algorithms).

Applications of Prior Works in This Paper. For this paper, eight shadow detection methods were selected for testing in the overall building height estimation process and are outlined below in Table 1. Additionally, the benefits of hybrid implmentations (where an algorithm pulls elements from more than one shadow method class) have been discussed previously. Those selected for this paper's experimentation follow this trend. Feature learning-based methods were not implemented in this study due to their complexity and their respective training data requirements. Only the first algorithm used in this paper may be considered an object-based classification method, with the remaining seven being pixel-based methods—a limitation that will be addressed further in the "Discussion" chapter.

Table 1. Shadow Detection Algorithms Used in this Paper

Author(s)	Year	Title/Paper/Code	Algorithm Classes
V. Arévalo, J. González, and G. Ambrosio	2008	"Shadow detection in colour high-resolution satellite images" [29] [30]	Region, Color model, Thresholding
K. Deb and A. H. Suny	2014	"Shadow detection and removal based on YCbCr color space" [31] [32]	Color model, Thresholding
V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli	2017	"Automatic shadow detection in aerial and terrestrial images" [33] [34]	Color model, Thresholding
H. Ma, Q. Qin, and X. Shen	2008	"Shadow segmentation and compensation in high resolution satellite images" [35] [30]	Color model, Thresholding
S. Murali and V. K. Govindan	2013	"Shadow detection and removal from a single image using LAB color space" [36] [37]	Color model, Thresholding

Table 1 continued: Shadow Detection Algorithms Used in this Paper

Author(s)	Year	Title/Paper/Code	Algorithm Classes
G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo	2018	"Near real-time shadow detection and removal in aerial motion imagery application" [38] [39]	Color model, Thresholding
K. K. Singh, K. Pal, and M. J. Nigam	2012	"Shadow detection and removal from remote sensing images using NDI and morphological operators" [40] [41]	Color model, Thresholding
V. J. D. Tsai	2006	"A comparative study on shadow compensation of color aerial images in invariant color models" [42] [30]	Color model, Thresholding

METHODS

The research methods in this study are comprised of three parts: study site selection and generation of ground truth data, batch processing of imagery using eight different shadow detection algorithms, and the shadow-overlapping building height estimation process.

Study Area and Ground Truth Data

Study Area. The Missouri State University (MSU), Springfield, MO, campus and surrounding area was selected as the area of interest for testing the tools developed for this paper, as quality imaging and ground truth data was previously known to be available.

Study Image. March 2019 leaf-off imagery was obtained for the MSU study area for optimal view of shadows. The image was first exported from Google Earth Pro [43] and then georeferenced using ESRI ArcGIS Desktop software. The image capture date was provided by Google Earth Pro and the solar azimuth angle was measured from imagery as the back bearing of shadow angles. Further, the solar elevation angle was estimated using the NOAA Solar Calculator [44] using the known date and solar azimuth angle. Comprehensive metadata for the study image shown in Fig. 7 (left) is as follows:

- Spectral resolution – 3 bands/true color RGB
- Dimensions – 4800 x 8322 pixels
- Capture date/time – 3/15/2019 at 12:48:45 PM (estimated)
- Solar azimuth angle – 167°
- Solar elevation angle – 50° (estimated)
- Cell size – 0.585875426437609 ft

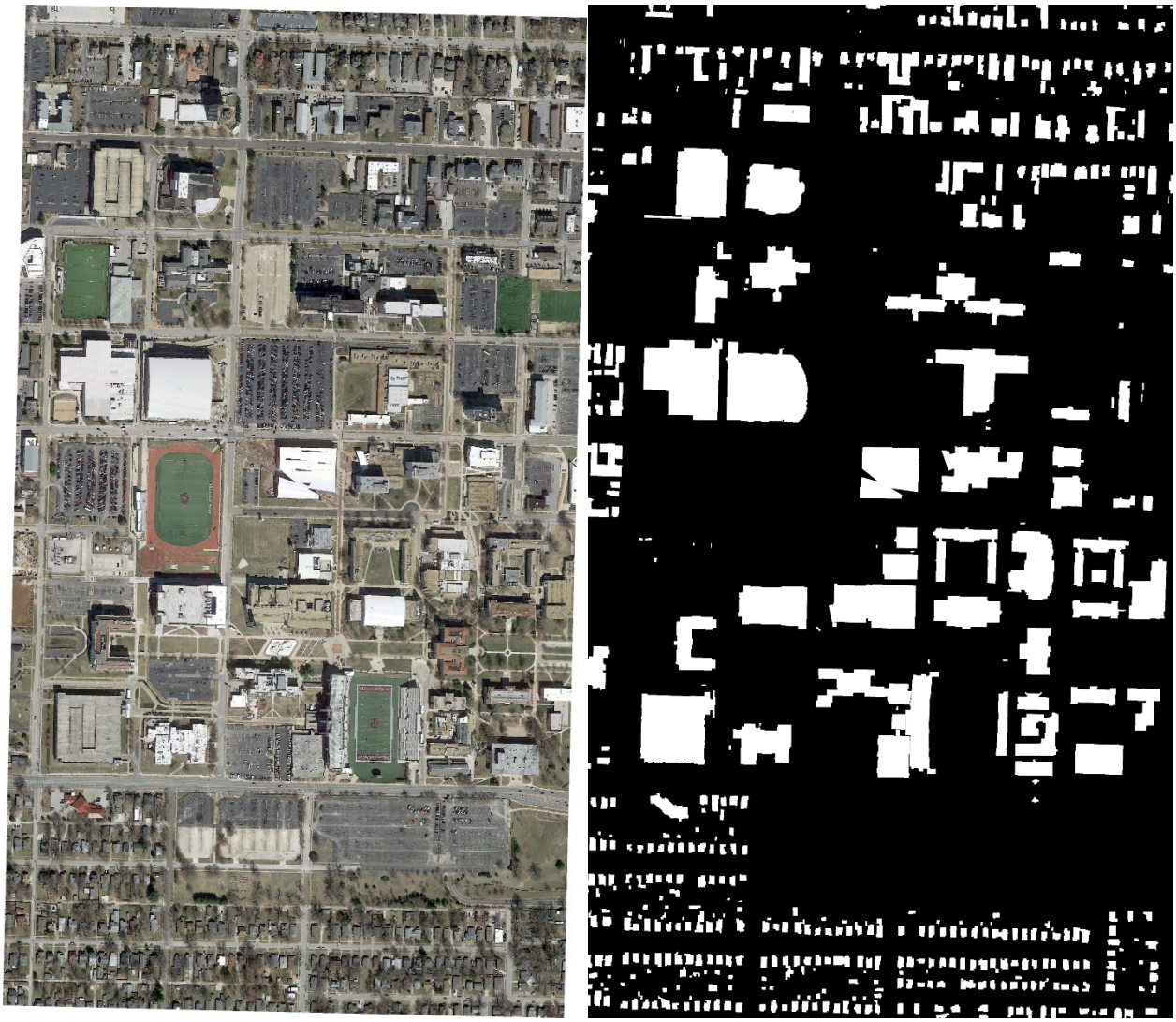


Fig. 7. March 2019 Google Earth image of MSU study area (left) and 495 ground truth building footprints derived from 2011 Greene County, MO, LiDAR and DEM data (right).

Ground Truth. To verify the accuracy of building heights estimated using the shadow-overlapping tool developed in this study, a ground truth dataset was derived from LiDAR and DEM data offered by the Missouri Spatial Data Information Service (MSDIS) [45]. This data was produced by the Sanborn Map Co, Inc. for their "Greene County, Missouri, 2011 Digital Mapping Project" [46]. The LAS and DEM data was processed according to the flow diagram in Fig. 8 and as described below. Additionally, while OpenStreetMap data or authoritiative

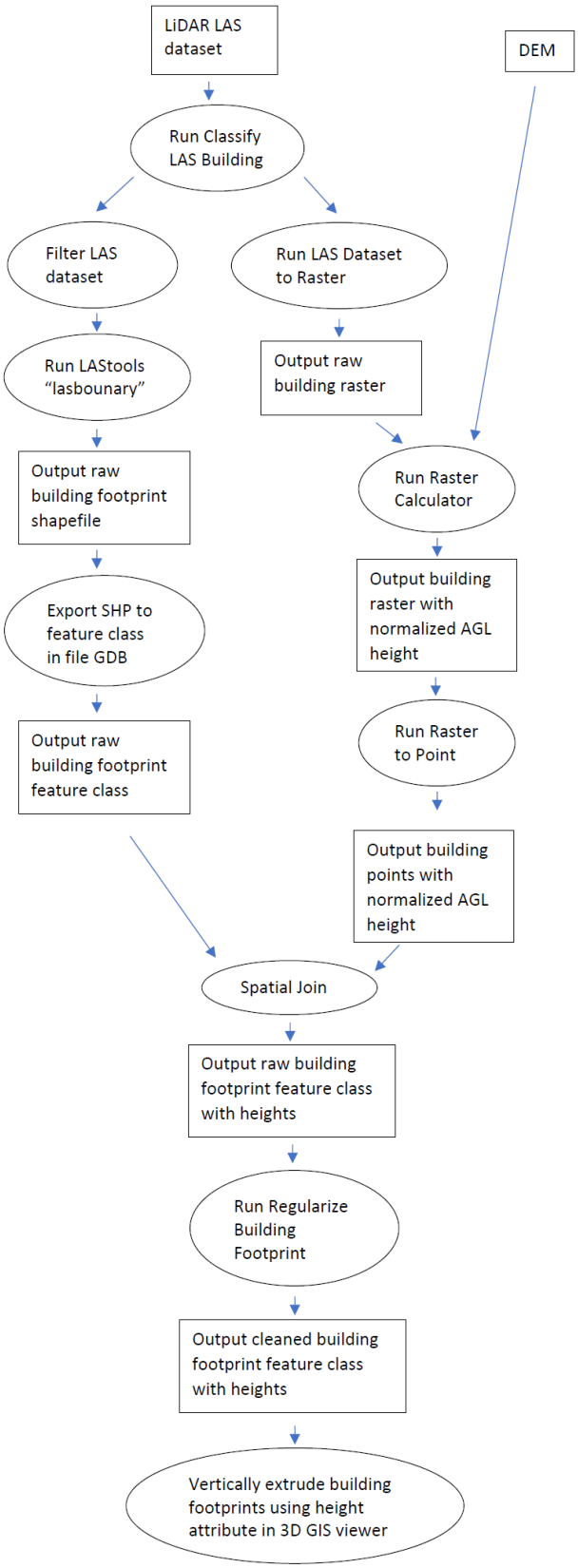


Fig. 8. Flow chart illustrating ground truth data generation from source LiDAR and DEM data.

building footprints from city or state government GIS data portals could have been used, it was decided to just create ground truth data from a single data source in the interest of minimizing errors that may occur during data conflation.

To create the vector building footprints required in this study as well as their ground truth heights, the ArcGIS Geoprocessing tool "Classify LAS Building" was first ran on the raw LAS point cloud data, which attributed buildings with a LiDAR classification of "6". Next, free, third-party LAStools (lasboundary) software [47] was used to create concave hull-based building footprint representations of building points, with the optimal tool parameters detailed below:

- Keep classification – 6 (outputs only building footprint polygons)
- Concavity – 8
- Disjoint – enabled (exports building footprints as discrete polygon features, not just one merged polygon)

Then, the ArcGIS tool "LAS Dataset to Raster" was ran, allowing the use of raster calcuations on what is essentially a LiDAR-derived digital surface model. Now, to normalize building elevation to "above ground level" (AGL) height, the following calculation was used on the output LAS raster and the DEM raster: HEIGHTS_AGL = OUTPUT_LAS_RASTER – DEM. At this point, a spatial join was used to attribute the building footprint polygons created using lasboundary with ground truth height values. The merge rule for this step was set to maximum, ensuring each building footprint was assigned the highest height value of those sampled from the LAS raster. Finally, to improve the geometry of the building footprint polygons, the ArcGIS Geoprocessing tool "Regularize Building Footprint" was used on the joined footprint polygons (with height attribution). The resulting 495 buildings with ground truth heights can be seen in Fig. 7 (right).

Shadow Detection Toolset

The Python toolset for shadow detection on RGB images described herein utilizes numerous open-source image processing and spatial data libraries, most notably OpenCV, scikit-image, GDAL, and NumPy. Additionally, it should be noted that the hardware used in this study was a ThinkPad X1 Yoga 1st Signature Edition with an Intel Core i7-6600U CPU @ 2.60 GHz and 16.0 GB of memory running 64-bit Windows 10 Pro. Python 3.7 was used in an Anaconda environment.

Outside of updating functions to Python 3, threshold parameter adjustments, and adding spatial reference management for file input/output actions [48], the implementations of each of the eight shadow detection methods remained largely unchanged from their originals as cited in Table 1. The only significant changes included the addition of two morphological operations, erosion followed by dilation, at the end of each of the algorithms as a final cleanup to the edges of identified shadow segments.

The final toolset created for this paper allows users to batch process all eight shadow detection algorithms on an image using a simple command line interface. Tool runtime was consistently less than 10 minutes to produce all eight images for any given input image of size 4800 x 8322 pixels or less. See Appendix A for output shadow rasters for each of the eight methods.

Shadow-Overlapping Building Height Estimation Tool

For this study, an implmentation of the shadow-overlapping method outlined in [13] was written from scratch in Python (scikit-image). The tool requires five inputs: raster shadow image created using shadow toolset, raster building footprint image, cell size in feet (the user would

enter 1.64042 here if the image had 0.5 meter spatial resolution), solar elevation angle, and solar azimuth angle.

First, the shadow angle is calculated as the back bearing of the input solar azimuth angle. However, as this tool requires polar angles to draw the artificial shadows, the next step is to convert the shadow angle from a true bearing (compass bearing) to an angle in standard position. At this point, preprocessing is complete and the tool is now ready to iterate over each building footprint.

For each footprint, the tool creates a list of its boundary pixel coordinates using the algorithm from [49]. Then, the tool iterates over candidate height in an exhausive search process—starting at an initial height of 20 feet and increasing by 5 with each iteration. For each iteration, a blank "temporary image" of the same size as both the original and the shadow image is created in memory where the artificial shadow will be drawn in binary with shadow pixels set to 1. Shadow length (in pixels) is next calculated for the footprint given the candidate height and solar elevation angle. Now, for each boundary pixel of the footprint, a line is generated in the direction of the established shadow angle using the length found in the previous step. Once this is completed, the artificial shadow image is nearly complete, however, the illuminated side of the building will have a false "shadow" that appears in the center of the building that needs to be removed. A simple raster calculation is used to set the building footprint pixel values back to 0 in the binary image so all that remains is the artificial shadow. A Jaccard (intersection over union) score is then calculated between the artificial shadow image and the true shadow image, which is stored along with the candidate height in a list. The process continues in a loop until one of two conditions are met—the maximum candidate height has been reached or if the Jaccard score for the current candidate height has decreased from the previous. (An arbitrary value of 70 feet was

chosen as the maximum height to be tested for the sake of runtime as it reduced the total number of iterations ran.) In the latter case, the final height estimate for that building footprint would be set to the the previous candiate height.

For the 495 buildings in the study image of size 4800 x 8322 pixels, the tool consistently took just under 13 hours to run (or around 1.57 minutes per footprint). Due to the shadow-overlapping tool's long runtime on the available computer hardware, only five shadow outputs were tested:

- V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli [33]
- H. Ma, Q. Qin, and X. Shen [35]
- S. Murali and V. K. Govindan [36]
- G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo [38]
- V. J. D. Tsai [42]

Additionally, it must be acknowledged that the accuracy of the shadow detection methods were not objectively measured using a ground truth, only by height estimation accuracy in the overall shadow-overlapping process. As such, the study was limited by the fact that the five "best" shadow detection images used for testing were selected, subjectively, on the basis of visual image interpretation alone.

GitHub Repository of Python Tools

The tools created for and used in this paper can be found at: https://github.com/LonnieBIII/Shadow-Overlapping_Building_Height_Estimation_Toolset.

RESULTS

Using the LiDAR/DEM-derived ground truth data, accuracy metrics in this study were measured in a few different ways for each shadow detection algorithm and are detailed below:

- Percentage of buildings whose estimated height falls within the accepted margin of error (MOE) of 10 feet, or about one floor of a building (see Table 2).
- Scatterplot showing the relationship between true building height and height estimation error (calculated as: estimated height – true height) (see Fig. 9-Fig. 13).
- Scatterplot showing the relationship between building area (square footage of the area within the building footprint, not total building square footage of all floors) and height estimation error (estimated height – true height) (see Fig. 9-Fig. 13).
- Linear regression of the data displayed in both scatterplots and the x-intercept of the line of best fit (the building height or area where height estimation error is most likely to be zero) (see Fig. 9-Fig. 13).

Table 2. Accuracy Metrics by Shadow Detection Algorithm

Author(s)	Year	Within MOE	Outside MOE
V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli	2017	87/236 (36.9%)	149/236 (63.1%)
H. Ma, Q. Qin, and X. Shen	2008	54/236 (22.9%)	182/236 (77.1%)
S. Murali and V. K. Govindan	2013	75/236 (31.8%)	161/236 (68.2%)
G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo	2018	139/236 (58.9%)	97/236 (41.1%)
V. J. D. Tsai	2006	147/236 (62.3%)	89/236 (37.7%)

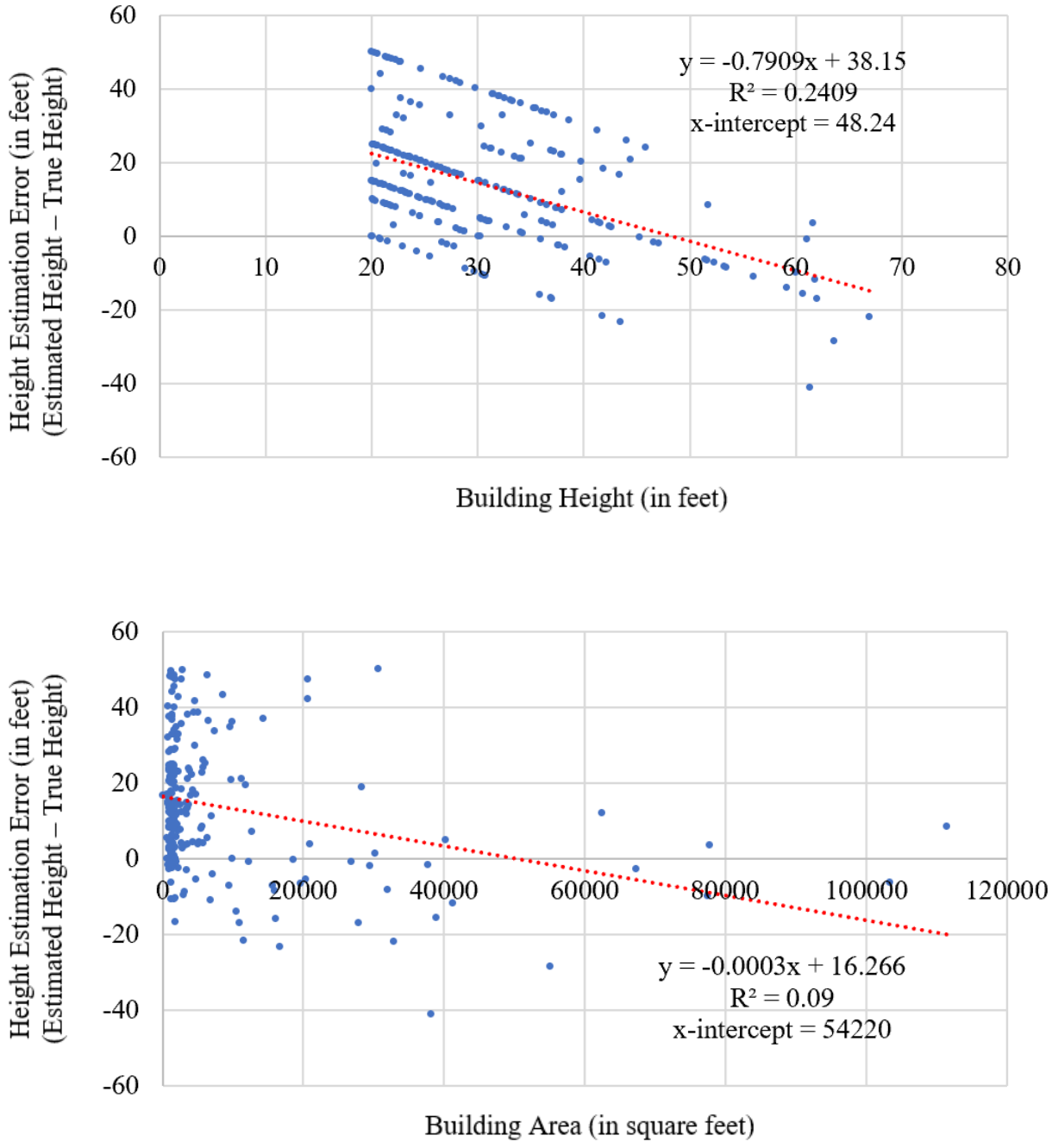


Fig. 9. Scatterplots illustrating the relationships between height estimation error and true building height (top) as well as building area (bottom) for V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli [33].

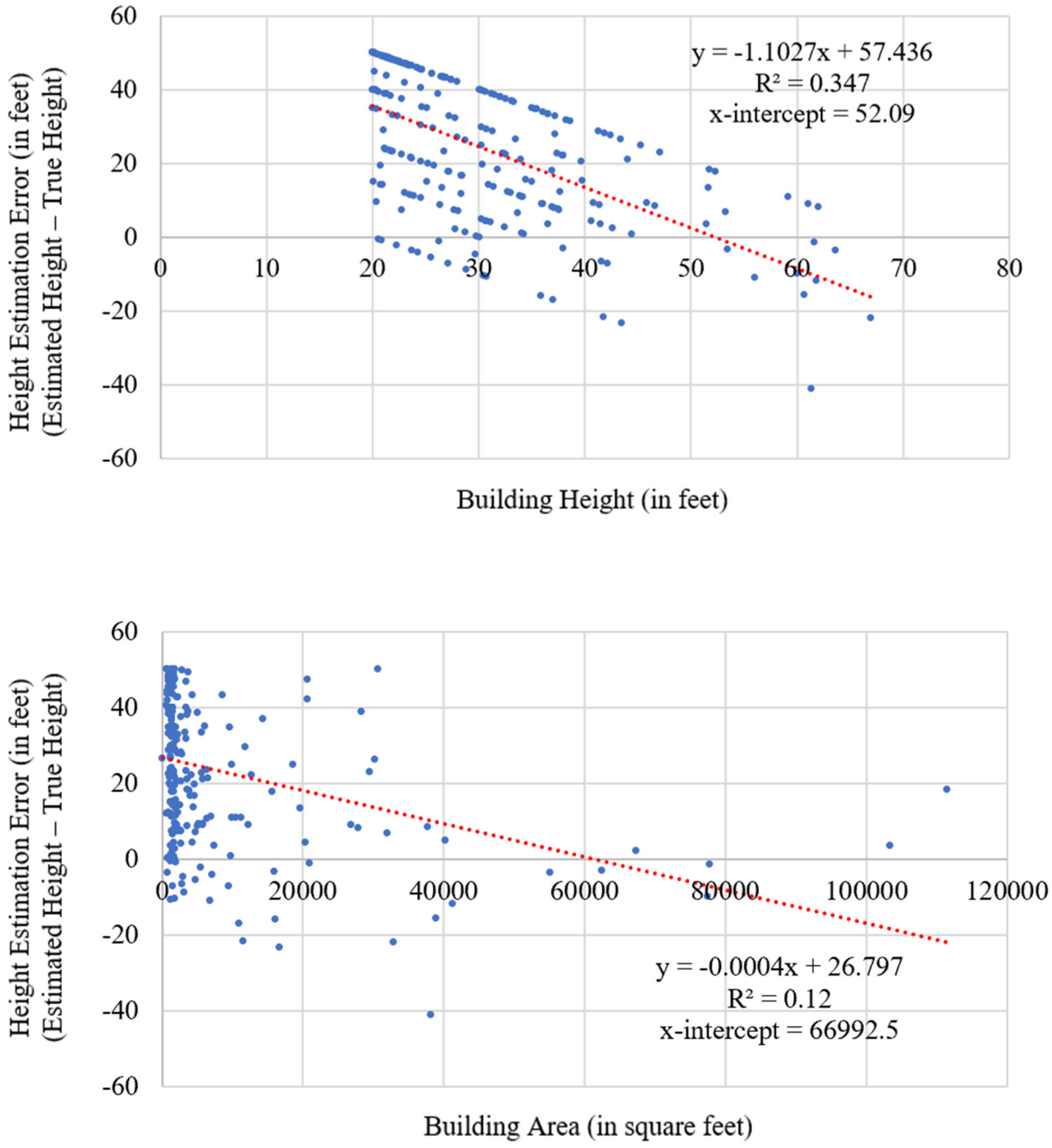


Fig. 10. Scatterplots illustrating the relationships between height estimation error and true building height (top) as well as building area (bottom) for H. Ma, Q. Qin, and X. Shen [35].

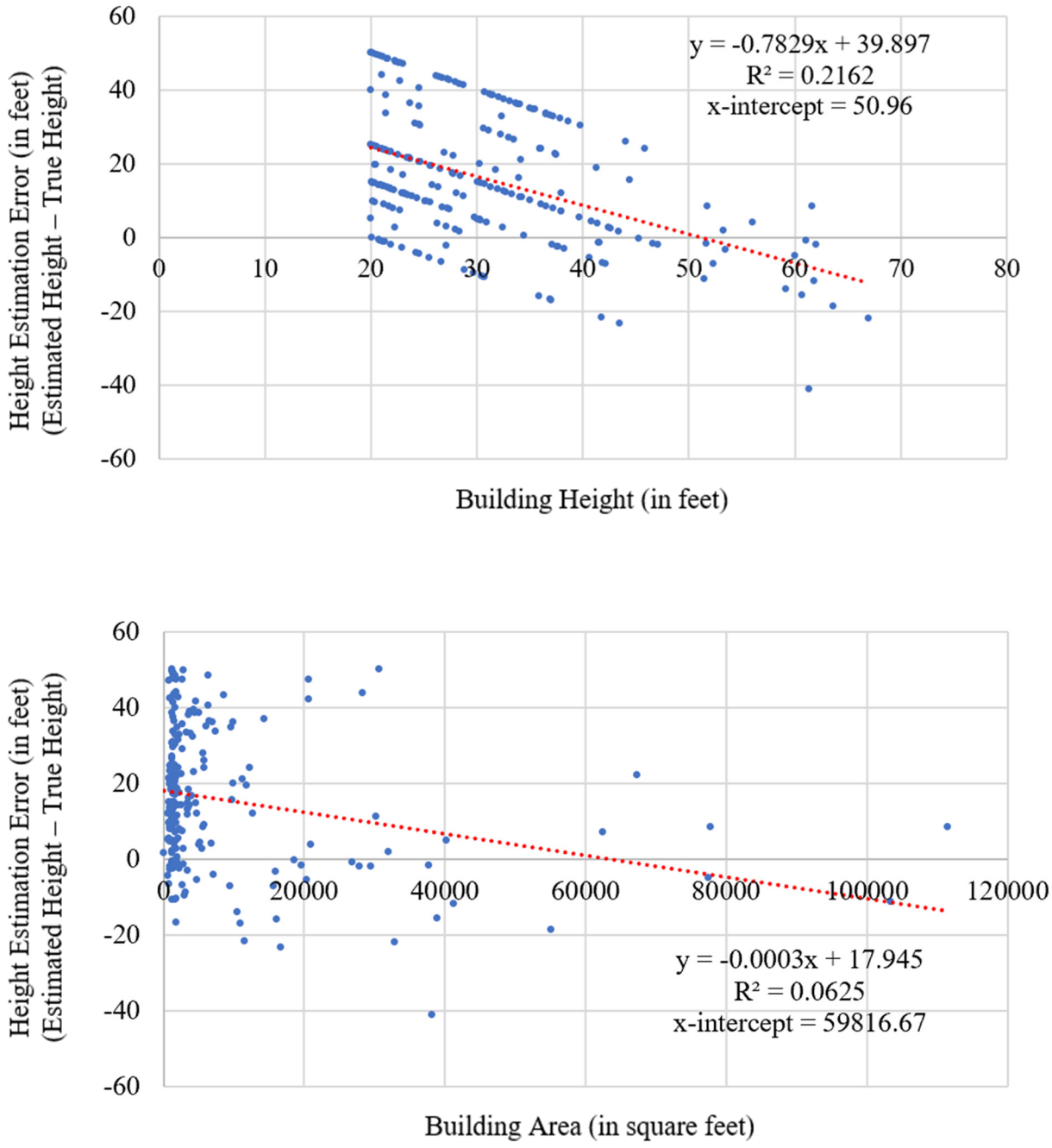


Fig. 11. Scatterplots illustrating the relationships between height estimation error and true building height (top) as well as building area (bottom) for S. Murali and V. K. Govindan [36].

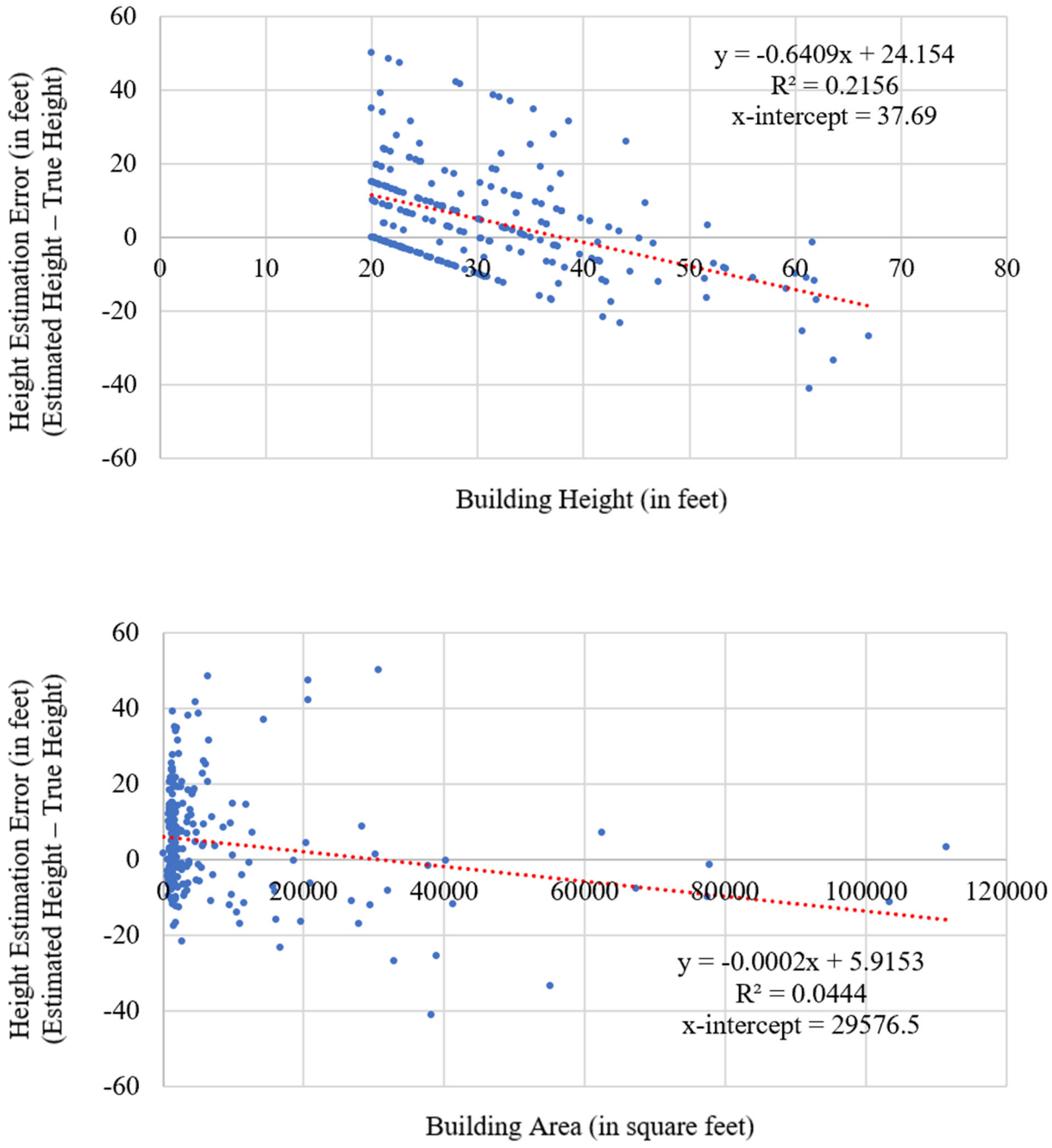


Fig. 12. Scatterplots illustrating the relationships between height estimation error and true building height (top) as well as building area (bottom) for G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo [38].

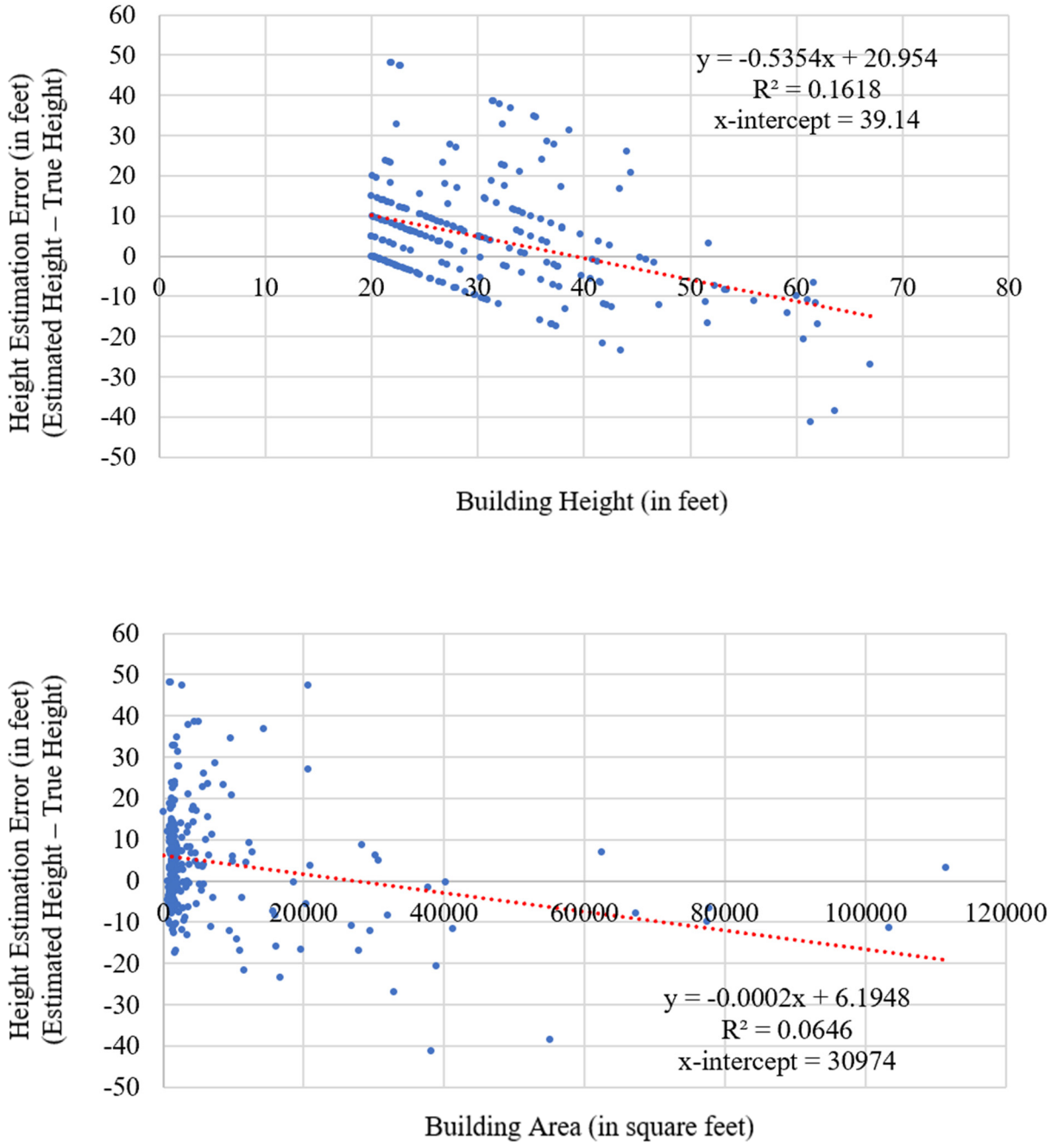


Fig. 13. Scatterplots illustrating the relationships between height estimation error and true building height (top) as well as building area (bottom) for V. J. D. Tsai [42].

However, it should be noted that since many buildings within the study area had true heights less than 20 feet or greater than 70 feet (minimum and maximum candidate heights for the shadow-overlapping tool), metrics were only computed on buildings whose true heights were within that range. (Essentially, buildings shorter than 20 feet would have had overestimated heights and buildings taller than 70 feet would have been underestimated.) As such, the following metrics depict only the remaining 236 buildings within the study area which are highlighted in Fig. 14. Additionally, it should be noted that a clustering pattern appears in the true height vs estimated height error scatterplots because of the five foot interval used between candidate heights. As such, points on the scatterplots appear to be "snapped" to multiples of five along the y-axis.

The shadow detection algorithms that resulted in the most accurate building height estimates were [42] and [38], with roughly 60% of building height estimates within 10 feet (or one floor) of the true height. Additionally, x-intercepts of the regression lines revealed that buildings whose heights were between 37-50 feet consistently had the lowest margins of error across all five shadow methods that were tested in this study. Perhaps a more interesting finding, however, is that shorter buildings' heights were systematically overestimated, and taller buildings were systematically underestimated—potentially underscoring a weakness with the shadow-overlapping method as a whole. A similar pattern was identified with building size/square footage with smaller buildings being overestimated and larger buildings being underestimated.

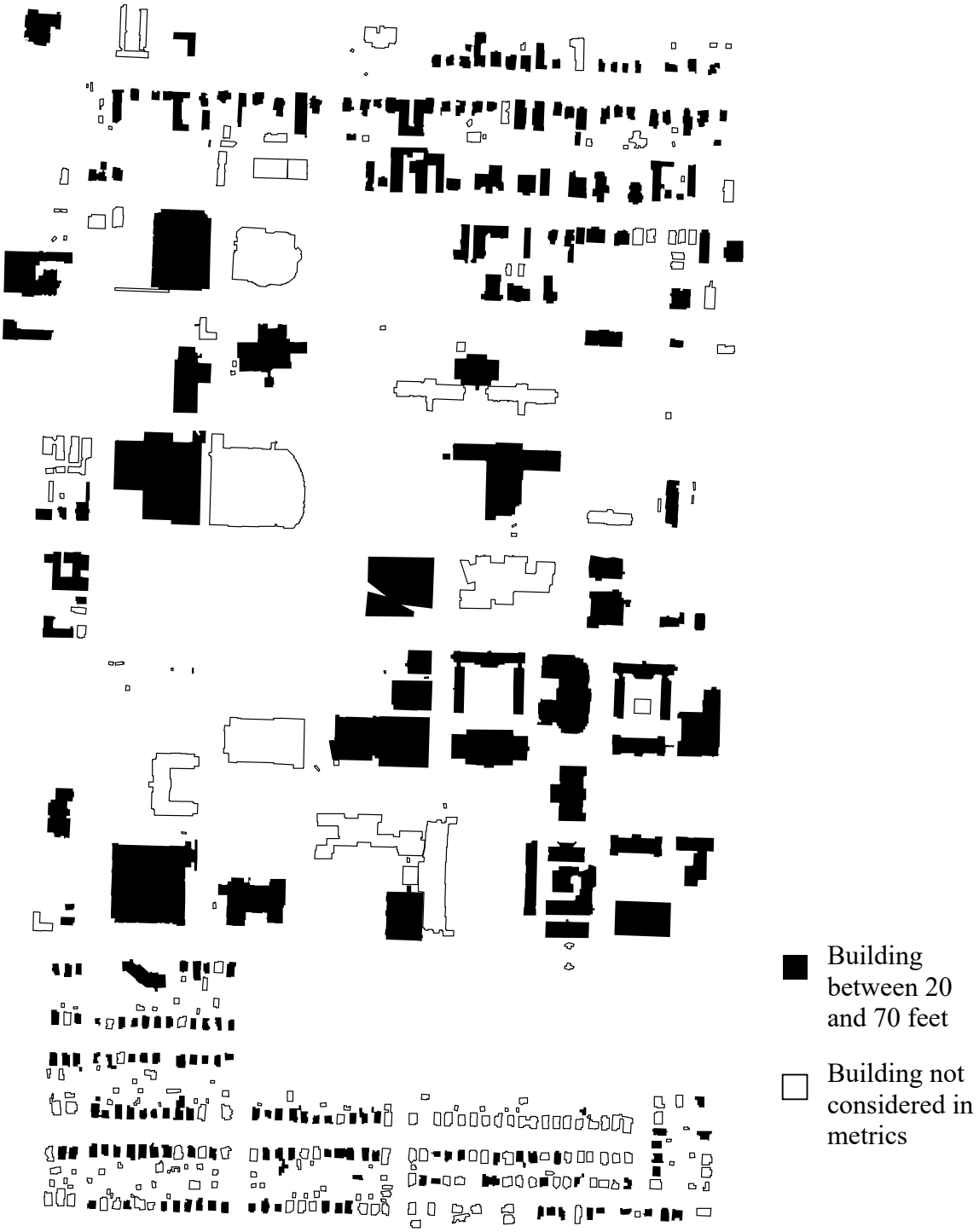


Fig. 14. Subset of 236 buildings within the MSU study area whose accuracy metrics were computed.

DISCUSSION

This study had a considerable number of limitations which will be addressed in this chapter as well as its author's overall conclusions.

Limitations Involving Imagery and Geography

Shadow detection in this study was performed on a single leaf-off image captured in March 2019. It is assessed that results from the shadow-overlapping process would have been significantly worse had shadows of homes in residential areas been obstructed by tree canopies. While this paper focused on the use of 3 band/RGB imagery due to its high availability, other authors have reported success with using imagery with a NIR band to help filter out vegetation from candidate shadow regions [11].

Further, the study image was selected in part due to the clarity of the shadow regions, with RGB pixel values of shadow areas near (0, 0, 0). It is assessed that results would likely have been worse with poorer shadow contrast. Also, it is difficult to determine if a preprocessing step (e.g., histogram equalization) should be included before detecting shadows on raw imagery without additional testing. In testing this paper's hypothesis, an image with optimal conditions was used under the assumption that, should the process fail for an image/scene with good conditions, the shadow-overlapping method would likely not be a suitable candidate in the cases of more challenging geographies or poorer image conditions.

The solar elevation angle of the study image (~50°) was optimal for the shadows in the MSU area—long enough shadow lengths for more accurate trigonometry to find building heights but not too long as to have shadows falling significantly onto adjacent buildings. Without testing

more images, it is not possible to determine whether different solar elevation angles will alleviate or worsen the systematic issue of shorter buildings' heights being overestimated and taller buildings being underestimated.

Additionally, a common phenomenon that occurs in cities is when tall buildings' shadows fall onto adjacent buildings (e.g., high rise apartment complexes). The MSU study area is not a dense urban area and, in this study, the impact this phenomeonon might have had on the shadow-overlapping process was not tested. This, however, would likely have introduced a significant amount of error and has previously been reported in the literature by multiple authors [12] [13]. With that being said, a potential workaround could include programmatically aggregating the heights of adjacent or nearby buildings of similar appearance. Another potential solution could be to use imagery captured when the sun is high in the sky and shadows are shorter so they are less likely to fall on adjacent structures. The impact shorter shadows would have in the shadow detection process and the sensitivity of height estimation results is also yet to be tested.

Lastly, this study is limited by the fact that the shadow-overlapping method was only tested against Western geography—specifically a mixed use area consisting of a college campus and surrounding residential-style buildings. However, it stands to reason that results might vary significantly across different global regions and across different images. Specifically, areas where buildings are densely packed in next to each other might have obscured shadow regions—reducing the effectiveness of any shadow-based height estimation methodology. With tall buildings, an aggregation strategy has been proposed above, however, in the case of densely packed low rise buildings such as the favela communities of Brazil, there may not be any viable

workarounds as there may not be any gaps between buildings where shadows could fall onto the ground and footprint delineation of individual structures might be challenging.

Limitations Involving the Python Tools and Overall Process

Currently, this process requires building footprints to be already available, however, implementing an automated (AI/ML) process for extracting clean building footprints from imagery is another important aspect of producing an entirely automated 3D building model extraction process.

Furthermore, as stated earlier in this paper, it should be recognized that the subjective visual interpretation of eight shadow detection outputs was a weak point in this study, as well as the reliance on pixel-based classification methods. While the results from the pixel-based methods used in this study were likely good enough for the MSU study image, the flexibility and robustness of a well-trained object-based classifier makes one almost certainly more appropriate for use in the overall shadow-overlapping process when considering the diversity of images that would be needed to create global building height data.

Another significant limitation of this study was that candidate height selection in the Python tool started at 20 feet and maxed out at 70 feet for the sake of improving runtime during the testing phase of experimentation. This brought down the total number of buildings analyzed down from 495 to 236, which is a number likely not statistically significant enough to support or reject this paper's hypothesis that the shadow-overlapping method can produce similar results to LiDAR-based methods. To that point, instead of using an arbitrary value of 70 feet, increasing the maximum candidate height to around 100 feet (or greater) would be suggested for future

testing, as the aerial ladders on fire trucks commonly have a maximum length of 100 feet and many cities limit building height based on that criteria.

Also, the methodology of creating artificial shadows in scikit-image by first finding the border pixels of the footprint and then, secondly, drawing lines off of each border pixel given the desired vector (angle and distance) is likely inefficient and leading to the substantial runtimes observed in this paper's experimentation. A previous attempt to create artificial shadows involving iterating over each pixel in the footprint and shifting them was found to be even more computationally complex and the process was quickly abandoned. Since both OpenCV and scikit-image do not have more efficient built-in functions to complete a task similar to the artificial shadow creation step using rasters, perhaps a better solution would be to process artificial shadows and perform Jaccard index calcuations using vectors. Python scripts using ESRI ArcGIS software have been published [50] [51] to simulate shadows for vector building footprint polygons with good success, however, the goal of this study was to only use free or open-source software. With that being said, it is entirely possible that a similar tool can be written using an open-source Python library designed to handle vector spatial data such as GDAL or Shapely.

With a faster process in place to draw artificial shadows, testing on more images and spanning more diverse geographies can more easily occur. Additionally, while this study used exhaustive search to find candidate heights as was done by [13], optimization algorithms have been reported to improve runtimes as well by [14] and [15] and can be implemented as a next step.

Lastly, while this study leaves many questions unanswered, the work has considerable merit in terms of the Python toolset developed which facilitates further fine tuning and

development of the algorithms used within. The toolset was written in a "plug and play" manner whereby users can easily switch between different shadow detection algorithms and play with different parameters. A significant portion of the workload in this study was implementing the shadow-overlapping building height estimation tool in Python and making edits to the shadow detection scripts to ensure cross compatability and updating them to the latest version of Python. With a toolset now available, future work can be done much more quickly with the tool processing time being the only major time cost. A final positive outcome of this research is that the batch shadow detection tool can be used for a variety of other remote sensing problems and is not limited in application to just building height estimation.

Conclusion

Currently, it does not appear that the shadow-based method tested in this research is reliable enough to produce height estimations comparable to LiDAR-derived methods. Unfortunately, this is expected to be the case in the target regions of this research—developing nations—where densely packed, low-rise buildings obstructed by tree canopies are likely to be found and poor view of shadows/footprint delineation could complicate the shadow-overlapping process. Lastly, the results from the shadow-overlapping algorithm discussed in this paper should not be considered to be suitable for downstream calculations (e.g., estimating number of floors of a building/building capacity, line of sight analysis, or flood damage modeling). However, for a simple/generalized 3D cartographic representation of an area that is not a dense urban landscape, it appears that this low-cost method could produce adequate results. Examples of these visualizations can be found in Appendix B.

REFERENCES

[1] "How GIS can help with emergency management," *USC GIS Blog*, 24-Mar-2021. [Online]. Available: https://gis.usc.edu/blog/how-gis-can-help-with-emergency-response/. [Accessed: 20-Jun-2022].

[2] A. Tiwari and K. Jain, "A detailed 3D GIS architecture for disaster management," *International Journal of Advanced Remote Sensing and GIS*, vol. 4, no. 1, pp. 980–989, 2015.

[3] CBS News, "Inside the model shop behind bin Laden raid," *CBS News*, 13-Sep-2020. [Online]. Available: https://www.cbsnews.com/video/inside-the-model-shop-behind-bin-laden-raid/. [Accessed: 20-Jun-2022].

[4] W.-R. Su, C.-H. Huang, and C.-H. Yang, "3D GIS helps Taiwan prepare for flood events," *ArcNews*, 2020. [Online]. Available: https://www.esri.com/about/newsroom/arcnews/3d-gis-helps-taiwan-prepare-for-flood-events/. [Accessed: 20-Jun-2022].

[5] J. Lesparre and B. G. Gorte, "Simplified 3D city models from LiDAR," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIX-B2, pp. 1–4, 2012.

[6] J. Pánek, "The commercialisation of public data – How does participatory data-mining look on a global scale?," *South African Journal of Geomatics*, vol. 2, no. 3, pp. 231–245, 2013.

[7] F. E. Sandnes, "Determining the geographical location of image scenes based on object shadow lengths," *Journal of Signal Processing Systems*, vol. 65, no. 1, pp. 35–47, 2010.

[8] *OpenStreetMap*. [Online]. Available: https://www.openstreetmap.org/. [Accessed: 20-Jun-2022].

[9] "Building footprints," *Bing Maps*, 12-Feb-2020. [Online]. Available: https://www.microsoft.com/en-us/maps/building-footprints. [Accessed: 20-Jun-2022].

[10] S. Murali, V. K. Govindan, and S. Kalady, "A survey on shadow detection techniques in a single image," *Information Technology And Control*, vol. 47, no. 1, 2018.

[11] M. Teke, E. Başeski, A. Ö. Ok, B. Yüksel, and Ç. Şenaras, "Multi-spectral false color shadow detection," *Photogrammetric Image Analysis*, pp. 109–119, 2011.

[12] Y. Shao, G. N. Taff, and S. J. Walsh, “Shadow detection and building-height estimation using IKONOS data,” *International Journal of Remote Sensing*, vol. 32, no. 22, pp. 6929–6944, 2011.

[13] N. Kadhim and M. Mourshed, “A shadow-overlapping algorithm for estimating building heights from VHR satellite images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 1, pp. 8–12, 2018.

[14] A. N. Trekin, V. Y. Ignatiev, and P. Y. Yakubovskii, “Deep neural networks for determining the parameters of buildings from single-shot satellite imagery,” *Journal of Computer and Systems Sciences International*, vol. 59, no. 5, pp. 755–767, 2020.

[15] M. Izadi and P. Saeedi, “Height estimation for buildings in monocular satellite/airborne images based on fuzzy reasoning and genetic algorithm,” *2009 10th Workshop on Image Analysis for Multimedia Interactive Services*, 2009.

[16] M. Mitchell, *An introduction to genetic algorithms*. Cambridge, MA: MIT Press, 1996.

[17] N. Donges, “Gradient descent: An introduction to 1 of machine learning's most popular algorithms,” *Built In*, 23-Jul-2021. [Online]. Available: https://builtin.com/data-science/gradient-descent. [Accessed: 20-Jun-2022].

[18] D. Poli and I. Caravaggi, “3D modeling of large urban areas with stereo VHR satellite imagery: lessons learned,” *Natural Hazards*, vol. 68, no. 1, pp. 53–78, 2013.

[19] T. Van Dijk and G. De Croon, “How do neural networks see depth in single images?,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[20] S. Srivastava, M. Volpi, and D. Tuia, “Joint height estimation and semantic labeling of monocular aerial images with CNNS,” *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[21] L. Mou and X. X. Zhu, “IM2HEIGHT: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network,” Sep. 2017.

[22] K.-L. Chung, Y.-R. Lin, and Y.-H. Huang, “Efficient shadow detection of color aerial images based on successive thresholding scheme,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 2, pp. 671–682, 2009.

[23] N. M. Kadhim, M. Mourshed, and M. T. Bray, “Shadow detection from very high resoluton satellite image using GRABCUT segmentation and ratio-band algorithms,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W2, pp. 95–101, 2015.

[24] W. R. Tobler, “A computer movie simulating urban growth in the Detroit Region,” *Economic Geography*, vol. 46, p. 234, 1970.

[25] X. Miao, H. Xie, S. F. Ackley, D. K. Perovich, and C. Ke, “Object-based detection of Arctic sea ice and melt ponds using high spatial resolution aerial photographs,” *Cold Regions Science and Technology*, vol. 119, pp. 211–222, 2015.

[26] N. T. Q. Trang, L. Q. Toan, T. T. H. Ai, N. V. Giang, and P. V. Hoa, “Object-based vs. pixel-based classification of mangrove forest mapping in Vien An Dong Commune, Ngoc Hien District, Ca Mau Province using VNREDSat-1 images,” *Advances in Remote Sensing*, vol. 05, no. 04, pp. 284–295, 2016.

[27] T. Gevers and A. W. M. Smeulders, “Color-based object recognition,” *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[28] E. Salvador, A. Cavallaro, and T. Ebrahimi, “Shadow identification and classification using invariant color models,” *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*.

[29] V. Arévalo, J. González, and G. Ambrosio, “Shadow detection in colour high-resolution satellite images,” *International Journal of Remote Sensing*, vol. 29, no. 7, pp. 1945–1963, 2008.

[30] N. Refes, “NassimREFES/shadow-detection-algorithms,” *GitHub*, 08-Jun-2021. [Online]. Available: https://github.com/NassimREFES/shadow-detection-algorithms. [Accessed: 20-Jun-2022].

[31] K. Deb and A. H. Suny, “Shadow detection and removal based on YCbCr color space,” *The Smart Computing Review*, vol. 4, no. 1, 2014.

[32] M. Mostipan, “Mykhailo-mostipan/shadow-removal,” *GitHub*, 25-Dec-2018. [Online]. Available: https://github.com/mykhailo-mostipan/shadow-removal. [Accessed: 20-Jun-2022].

[33] V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli, “Automatic shadow detection in aerial and terrestrial images,” *Bulletin of Geodetic Sciences*, vol. 23, no. 4, pp. 578–590, 2017.

[34] V. L. S. Freitas, “Vanderfreitas/shadowdetection,” *GitHub*, 28-Nov-2019. [Online]. Available: https://github.com/vanderfreitas/shadowdetection. [Accessed: 20-Jun-2022].

[35] H. Ma, Q. Qin, and X. Shen, “Shadow segmentation and compensation in high resolution satellite images,” *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, 2008.

[36] S. Murali and V. K. Govindan, “Shadow detection and removal from a single image using LAB color space,” *Cybernetics and Information Technologies*, vol. 13, no. 1, pp. 95–103, 2013.

[37] Y. Doğan, "YalimD/image_shadow_remover," *GitHub*, 20-Feb-2022. [Online]. Available: https://github.com/YalimD/image_shadow_remover. [Accessed: 20-Jun-2022].

[38] G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo, "Near real-time shadow detection and removal in aerial motion imagery application," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 104–121, 2018.

[39] T. W. W. Hong, "Thomaswangweihong/shadow-detection-algorithm-for-aerial-and-satellite-images," *GitHub*, 18-Nov-2019. [Online]. Available: https://github.com/ThomasWangWeiHong/Shadow-Detection-Algorithm-for-Aerial-and-Satellite-Images. [Accessed: 20-Jun-2022].

[40] K. K. Singh, K. Pal, and M. J. Nigam, "Shadow detection and removal from remote sensing images using NDI and morphological operators," *International Journal of Computer Applications*, vol. 42, no. 10, pp. 37–40, 2012.

[41] M. Thakkar, "Shadow detection using skimage · Issue #3214," *GitHub*, 21-Jun-2018. [Online]. Available: https://github.com/scikit-image/scikit-image/issues/3214. [Accessed: 20-Jun-2022].

[42] V. J. D. Tsai, "A comparative study on shadow compensation of color aerial images in invariant color models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 6, pp. 1661–1671, 2006.

[43] Google Earth Pro, *MSU study area satellite image*. Springfield, MO, 2019.

[44] "NOAA Solar Calculator," *ESRL Global Monitoring Laboratory*. [Online]. Available: https://gml.noaa.gov/grad/solcalc/. [Accessed: 20-Jun-2022].

[45] "Missouri LiDAR Data," *MSDIS*. [Online]. Available: https://www.msdis.missouri.edu/data/lidar/index.html. [Accessed: 20-Jun-2022].

[46] "Greene County, Missouri, 2011 digital mapping project - LiDAR & survey report." [Online]. Available: https://wustl.app.box.com/s/gqqtejfi32uj6gqy0sifqec0ob6d8c1o/file/502448229582. [Accessed: 20-Jun-2022].

[47] "Lasboundary," *rapidlasso GmbH*. [Online]. Available: https://rapidlasso.com/lastools/lasboundary/. [Accessed: 20-Jun-2022].

[48] S. Erle, F. Warmerdam, and E. Rouault, "Gdalcopyproj.py," *OSGeo SVN Repositories*, 2005. [Online]. Available: https://svn.osgeo.org/gdal/trunk/gdal/swig/python/samples/gdalcopyproj.py. [Accessed: 20-Jun-2022].

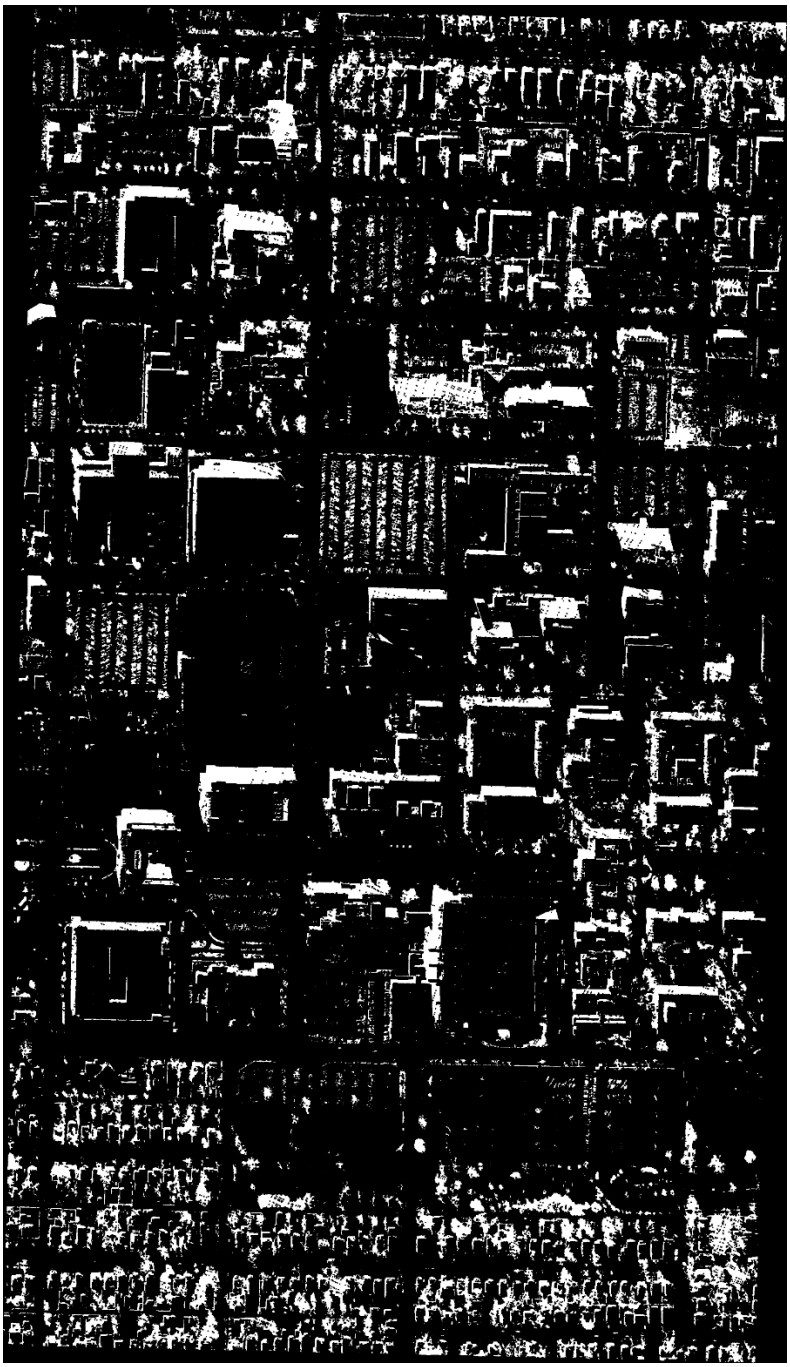
[49] S. Wallkötter, “Firefoxmetzger/ipynb_boundary_tracing,” *GitHub*, 30-Aug-2020. [Online]. Available: https://github.com/FirefoxMetzger/ipynb_boundary_tracing. [Accessed: 20-Jun-2022].

[50] “Producing building shadows using ArcGIS Desktop?,” *Geographic Information Systems Stack Exchange*, 22-Nov-2011. [Online]. Available: https://gis.stackexchange.com/questions/17155/producing-building-shadows-using-arcgis-desktop. [Accessed: 20-Jun-2022].

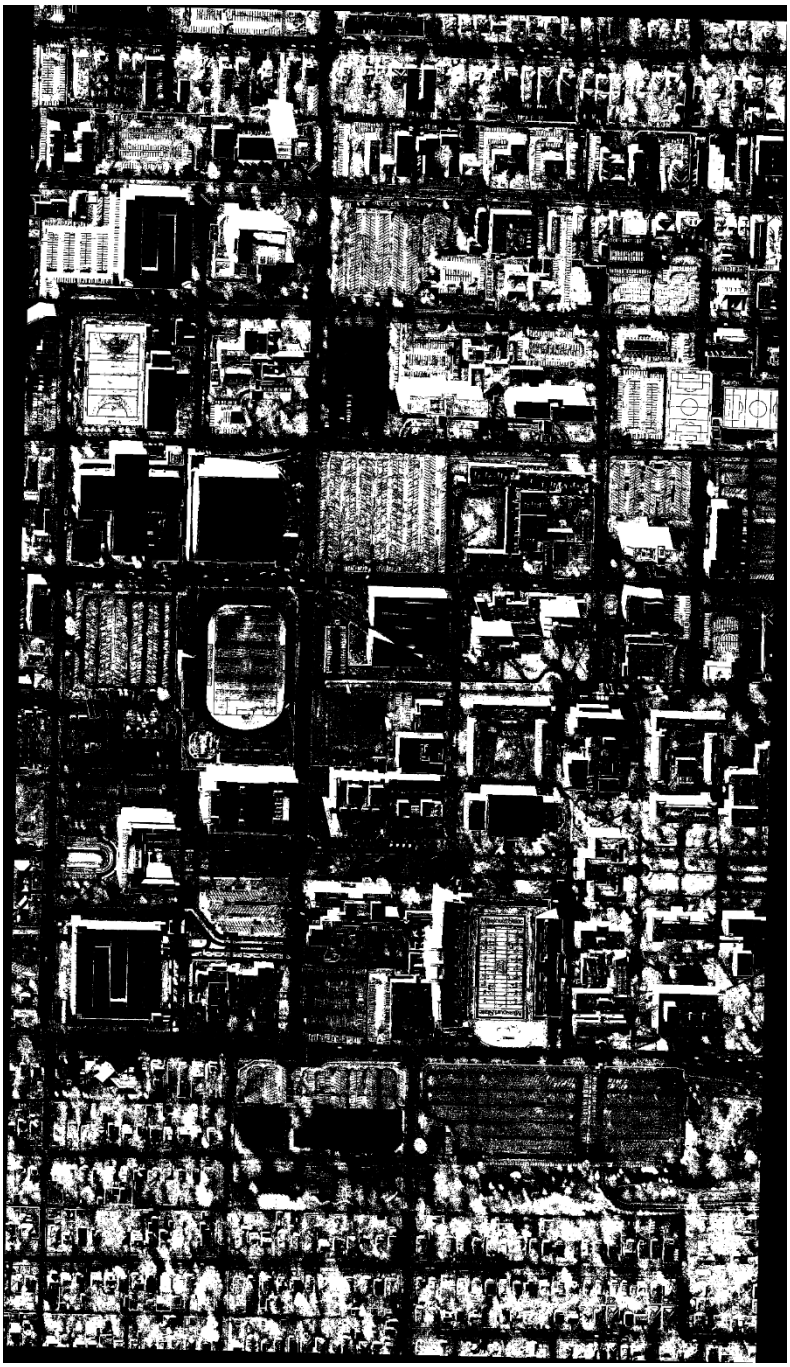
[51] “Porting Avenue code for producing building shadows to ArcPy/Python for ArcGIS Desktop?,” *Geographic Information Systems Stack Exchange*, 07-Feb-2012. [Online]. Available: https://gis.stackexchange.com/questions/19935/porting-avenue-code-for-producing-building-shadows-to-arcpy-python-for-arcgis-de. [Accessed: 20-Jun-2022].

APPENDICES

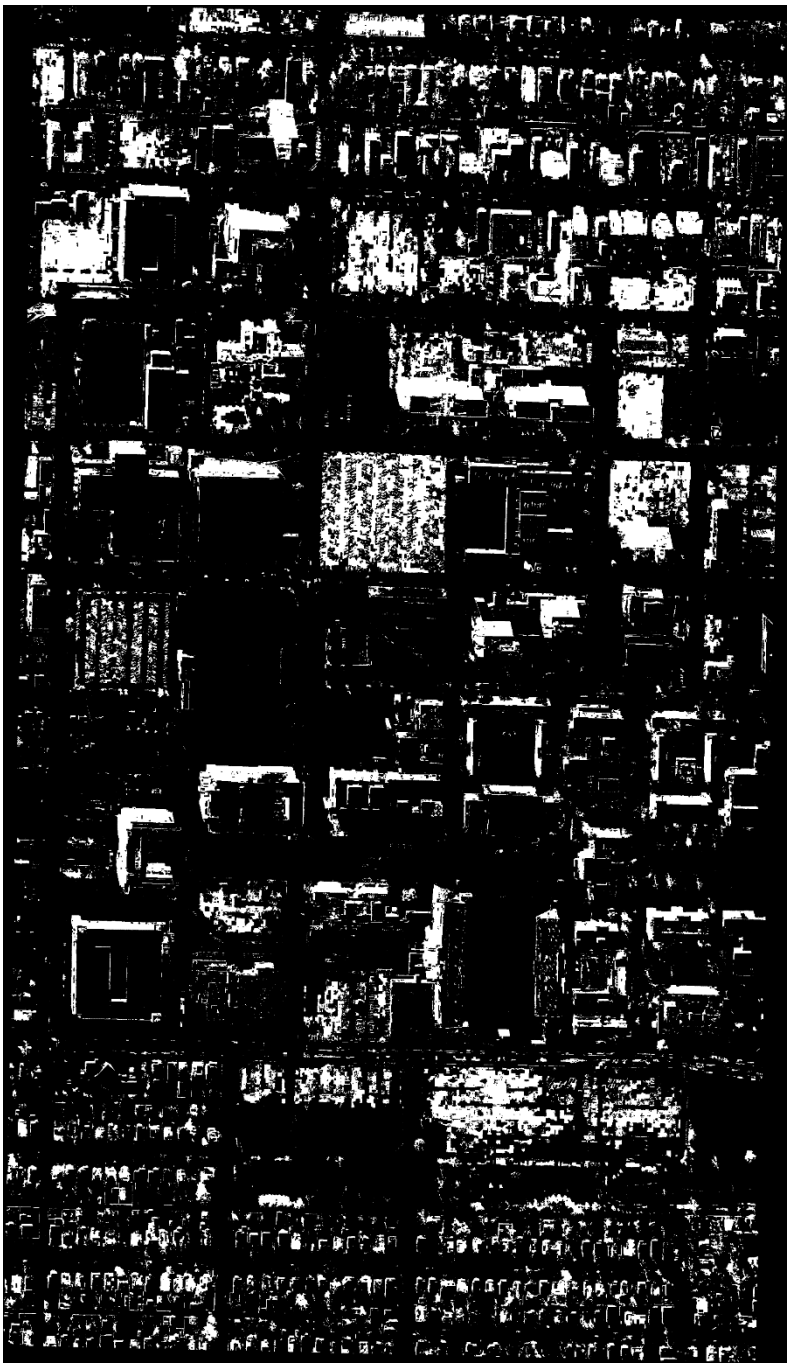
Appendix A: Results of the Eight Shadow Detection Algorithms



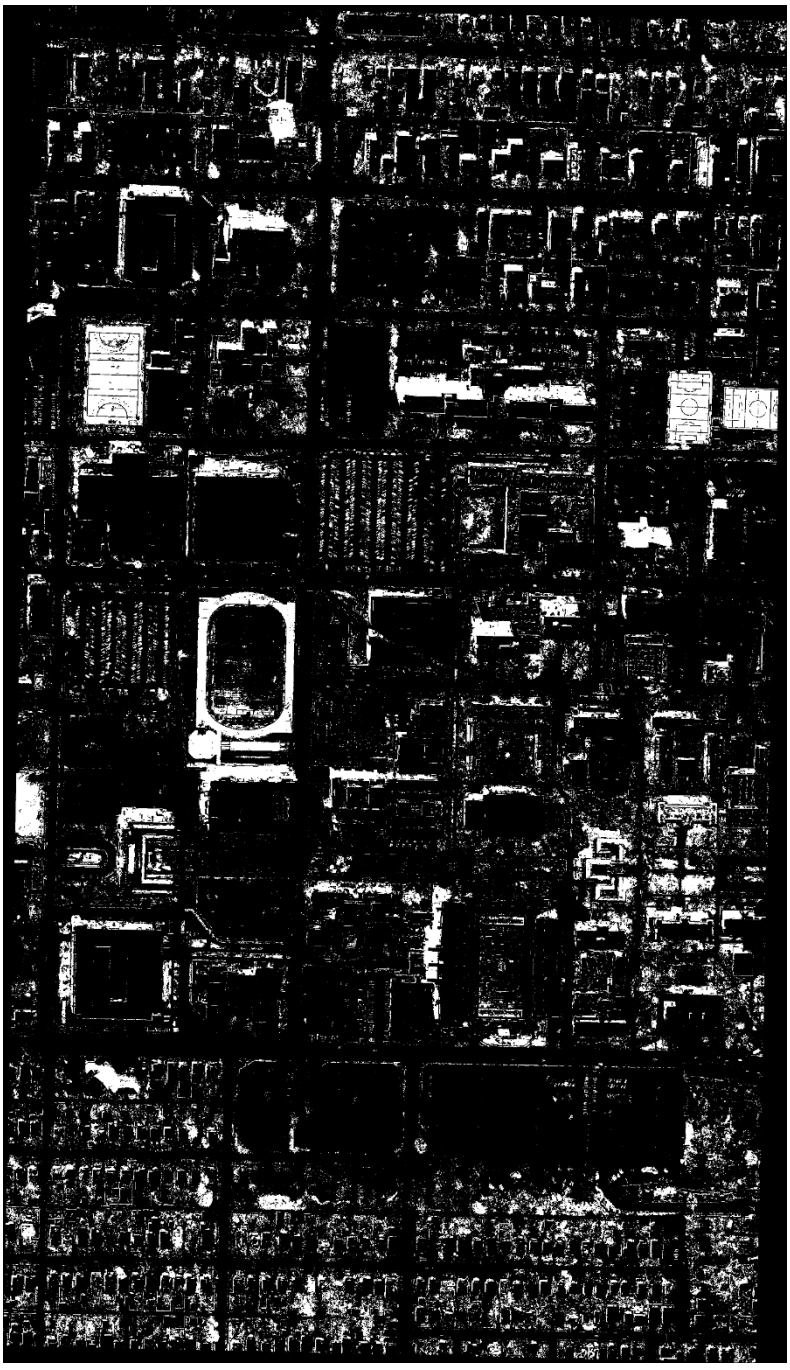
Shadow detection result from V. Arévalo, J. González, and G. Ambrosio [29].



Shadow detection result from K. Deb and A. H. Suny [31].

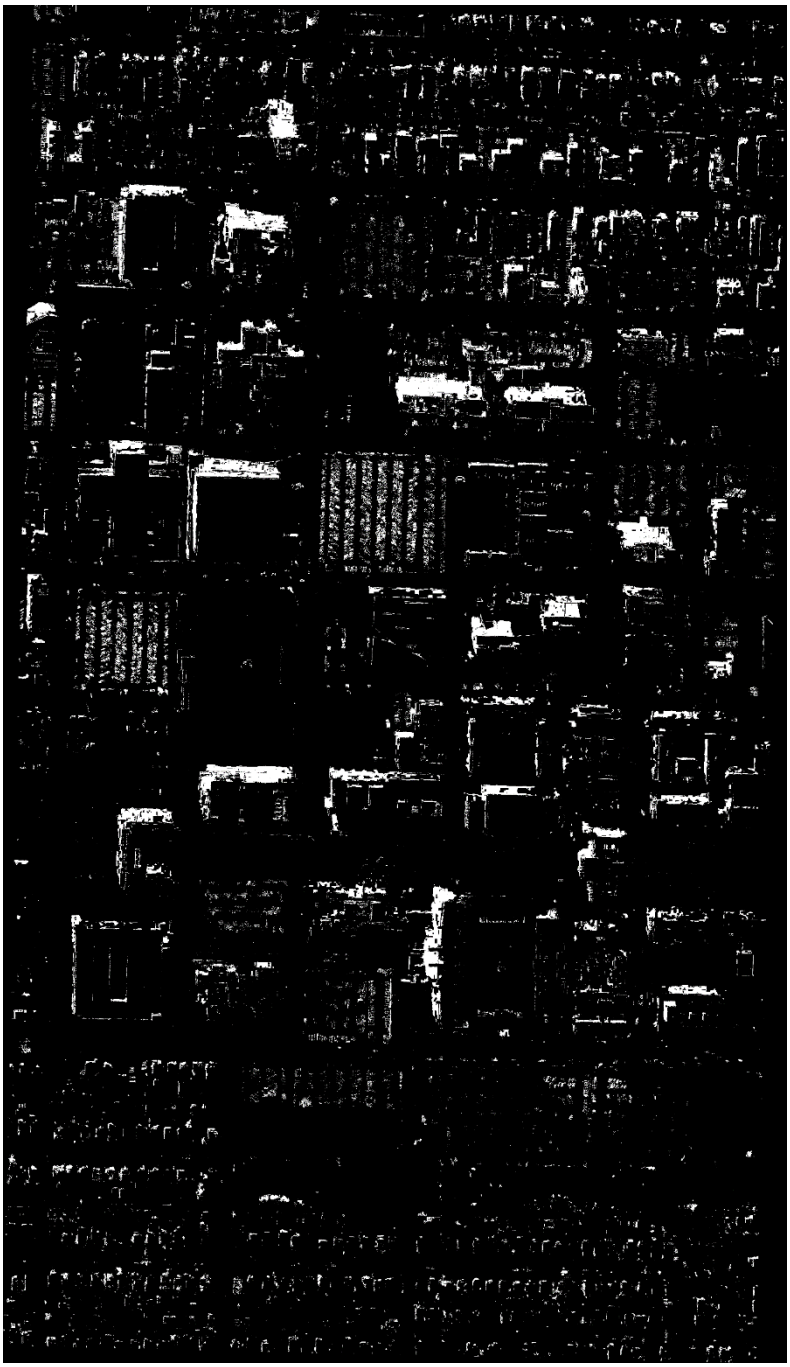


Shadow detection result from V. L. S. Freitas, B. M. F. Reis, and A. M. G. Tommaselli [33].



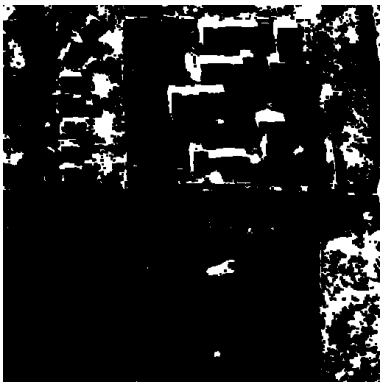
Shadow detection result from H. Ma, Q. Qin, and X. Shen [35].

Shadow detection result from S. Murali and V. K. Govindan [36].



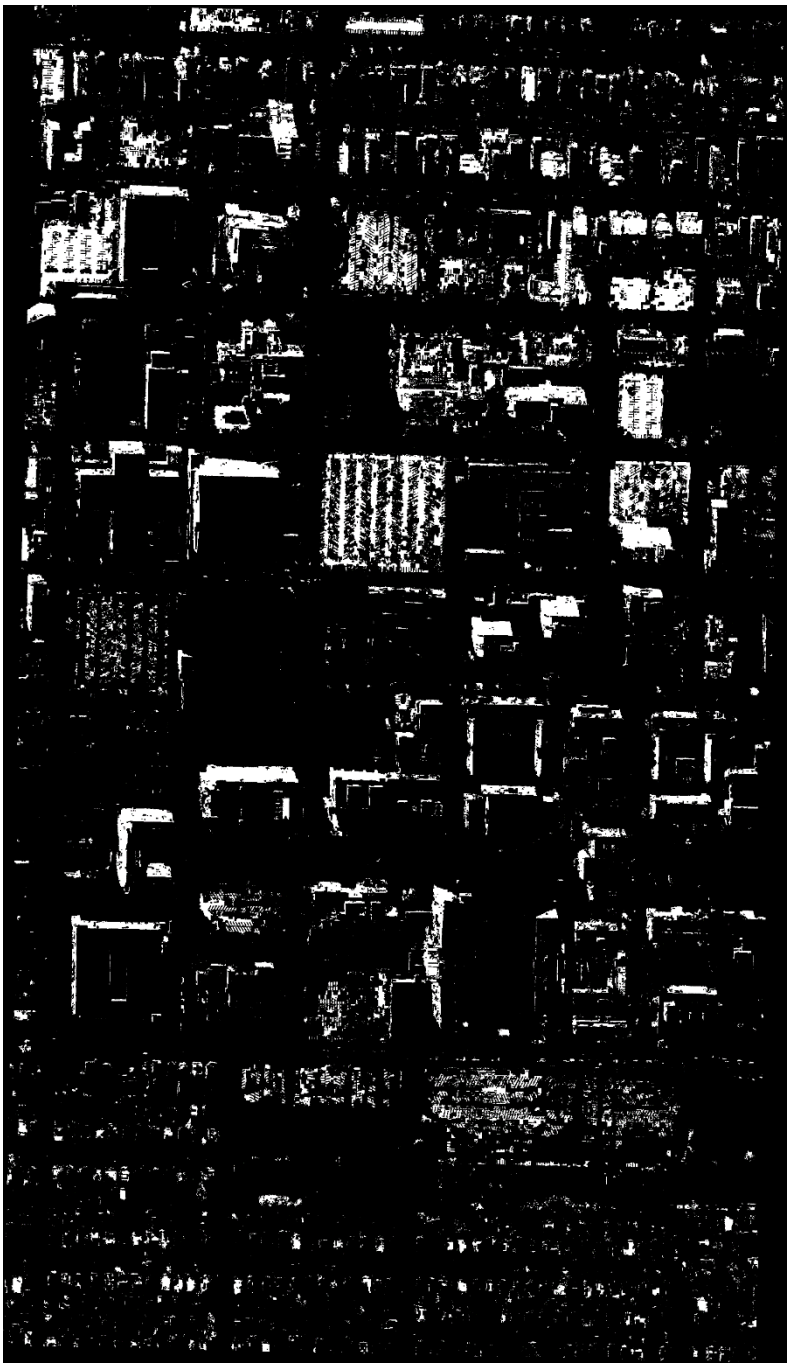
Shadow detection result from G. F. Silva, G. B. Carneiro, R. Doth, L. A. Amaral, and D. F. G. Azevedo [38].

Smaller test image that worked successfully



Successful shadow detection in smaller test image (512 x 512 pixels)

Shadow detection result from K. K. Singh, K. Pal, and M. J. Nigam [40]. This algorithm did not produce a suitable result for the MSU study image but did work successfully for other smaller images in testing, such as with the example seen above to the right.



Shadow detection result from V. J. D. Tsai [42].

Appendix B: 3D Visualizations of the Final Outputs

Rudimentary 3D model of the MSU study area created by vertically extruding footprints. Renderings have been included using ground truth heights (top) and with estimated heights (bottom) for all 495 buildings using V. J. D. Tsai [42].

Rudimentary 3D model of the MSU study area created by vertically extruding footprints. Renderings have been included using ground truth heights (top) and with estimated heights (bottom) for the subset of 236 buildings using V. J. D. Tsai [42] whose heights were between 20 and 70 feet and metrics were computed for.