Locating Low-Altitude Obstacles to Aid in Drone Coverage Modeling and Routing

Aaron Pulver

MS Geography University of Utah

Abstract

Unmanned aerial vehicles (UAVs), more commonly referred to as drones, are becoming increasingly popular. There are many uses for drones such as remote sensing, aerial imagery collection, and transportation of goods. In addition to those, it is possible that medical drones can be used to increase response times for time-critical emergencies such as cardiac arrest. As drones navigate urban environments, they may encounter obstacles that alter the optimal route resulting in longer travel times and farther distance travelled. This study locates significant obstacles and shows the effect they have on travel time and distance. Morphological image processing was used to extract buildings and trees, representing common obstacles, from 0.5m LIDAR data in Salt Lake City. Buildings and trees were successfully detected and incorporated into service area projections. Analysis results show that at low altitudes (1-30m) obstacles lead to a significant increase in distance travelled. Results also show that obstacles significantly reduce the total area serviced by a drone.

Introduction

Unmanned aerial vehicles (UAVs) have routinely been used for aerial imagery collection and military purposes (Everaerts 2008). Recently UAVs or drones, as they are more commonly referred to by the public, have become increasingly popular in commercial business. Amazon, for example, has been developing drones to deliver packages to customers (Misener 2014). In addition to commercial applications, drones may be equipped to transport medical supplies such as blood derivatives and pharmaceuticals to hospitals, remote areas, and mass causality incidents (Thiels et al. 2015). In addition to transporting medical supplies, drones may have the ability to provide other medical treatments. In 2014, a prototype medical drone equipped with an AED was announced (Communication 2014). This particular drone can fly at speeds up to 100 km/h and can be deployed by emergency dispatchers at a moment's notice (Husten 2014). The drone flies to the patient using cell phone GPS coordinates as a target. Upon reaching the target, the dispatcher, through real-time video and audio, can instruct the bystander to use the Automated External Defibrillator (AED) on the patient. Since medical drones can travel faster than traditional ground transport EMS vehicles, they provide the capacity to drastically reduce travel time and overall response times for time-critical emergencies. Drones have limited range

and in order to minimize response times, a network of multiple drones is required to adequately provide service to a large area.

Drones can be outfitted with several different sensors, cameras, and onboard microprocessors to autonomously self-stabilize, follow given paths, and detect and avoid obstacles (Jimenez Lugo and Zell 2013). Computer vision and machine learning algorithms have been developed and experimentally tested to successfully navigate drones in narrow environments and in areas where GPS signals are non-existent (Jimenez Lugo and Zell 2013; Mercado, Castillo, and Lozano 2015; Krajnik et al. 2012). Although drones have the capabilities to fly autonomously and avoid obstacles, these obstacles may significantly reduce the estimated response times. This study uses Light Detection and Ranging (LIDAR) data to locate obstacles and to determine their effect on medical drone service ranges at low altitudes (1-30m).

Data

A subsection of Salt Lake City was chosen as the study area. Four contiguous sections of LIDAR data were downloaded from the Utah Automated Geographic Reference Center (AGRC). AGRC provides 0.5m resolution LIDAR datasets in the form of Digital Surface Models (DSM) and Digital Terrain Models (DTM). The average vertical accuracy is 0.04 meters. The DSM was generated using a TINing process as described in the metadata. In addition to the LIDAR data, EMS stations were geocoded from the Utah Department of Health using the Utah road network dataset provided by AGRC.

Methodology

This study breaks the problem of locating obstacles into two categories: locating buildings, and locating trees. Both of these obstacles were considered large enough that they may significantly increase travel times as drones fly around or above them. In comparison, telephone poles and small shrubs should not significantly alter a drone's optimal route.

Detecting Buildings

Several methods have been proposed to detect buildings using LIDAR as well as ancillary data such as NDVI. Demir et al. (2008) compared four different methods of building detection. The first method relies on combining NDVI data with the normalized DSM. The second method used supervised classification method based on a normalized DSM. The third method used the raw point cloud density in conjunction with NDVI data. The final and most successful method uses the vertical density of raw DSM LIDAR data. Morphological image processing has also been used to detect buildings using LIDAR data (Meng, Wang, and Currit 2009).

I used an approach based on Ekhari et al. (2008) and Rottensteiner et al. (2002) where the DSM was subtracted from the DTM resulting in a normalized DSM (Figure 1c). The slope of the normalized DSM was taken (Figure 1d) and classified into two categories: "Low-Rise" (0 to 15cm), and "High-Rise" (greater than 15cm) (Figure 1e). Then a morphological fill operation was performed to remove small holes. The resulting pixels were then converted to polygons.

Polygons less than 50 square meters in area were removed. Finally, bounding boxes were found for each remaining polygon; this simplifies the geometry to reduce computation time in later steps. Since most buildings are rectangular, it also removes irregularities in the raster (Figure 1f). The analysis was performed using ArcGIS and the provided arcpy Python library.

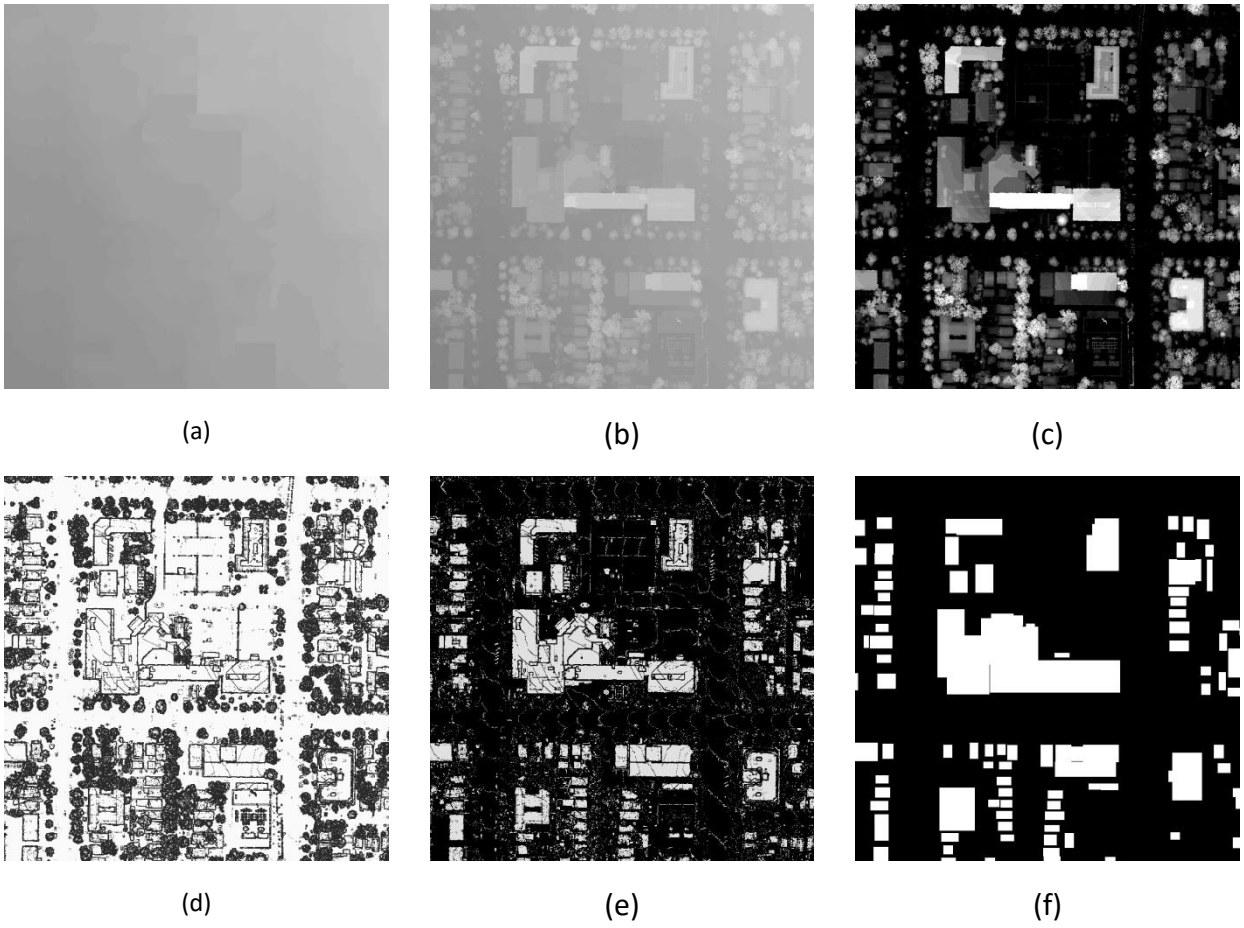

Figure 1. (a) the digital surface model, (b) the digital terrain model, (c) the normalized digital surface model, (d) the slope of the normalized digital surface model, (e) the reclassified slope, (f) the final raster showing the buildings in white.

Detection of Trees

Several methods have been proposed to both detect and delineate individual trees. Demir et al. (2008) analyzed the vertical density of DSM data to detect clusters representing trees. Koch et al. (2006) used a local maxima filter in conjunction with a pouring algorithm to delineate individual tree crowns. Rahman and Gorte (2008) used an inverse watershed algorithm, which

is similar to the pouring algorithm, to delineate individual tree crowns. Although these approaches work well, they are computationally expensive and complex.

This study uses a combination of surface roughness analysis and morphological image processing to detect trees. The first step was to compute the normalized DSM by subtracting the DSM from the DTM (Figure 2c). The slope of the normalized DSM was then taken, detecting the percent rise rather than the traditional degree measurement (Figure 2d). The slope was then classified into low (0-100 percent) and high (over 100 percent) groups (Figure 2e). The high slopes, generally represent pixels that have significant variation which is indicative of vegetation. The normalized DSM was also classified into low (0-1 meter) objects and high objects (greater than 1 meter). The normalized DSM raster containing tall objects was intersected with the raster containing high slope values. This resulted in raster showing pixels likely belonging to trees. The raster was then filtered using a median filter with a kernel size of 11. Since the input raster is binary, the median filter acts as a majority filter, and removes small groups of pixels and outlying pixels that are considered too small to be trees, resulting in a raster depicting trees (Figure 2f). The majority of the analysis was performed in ArcGIS using the arcpy library, however the image filtering was performed using scipy, a scientific Python library containing image processing functions.

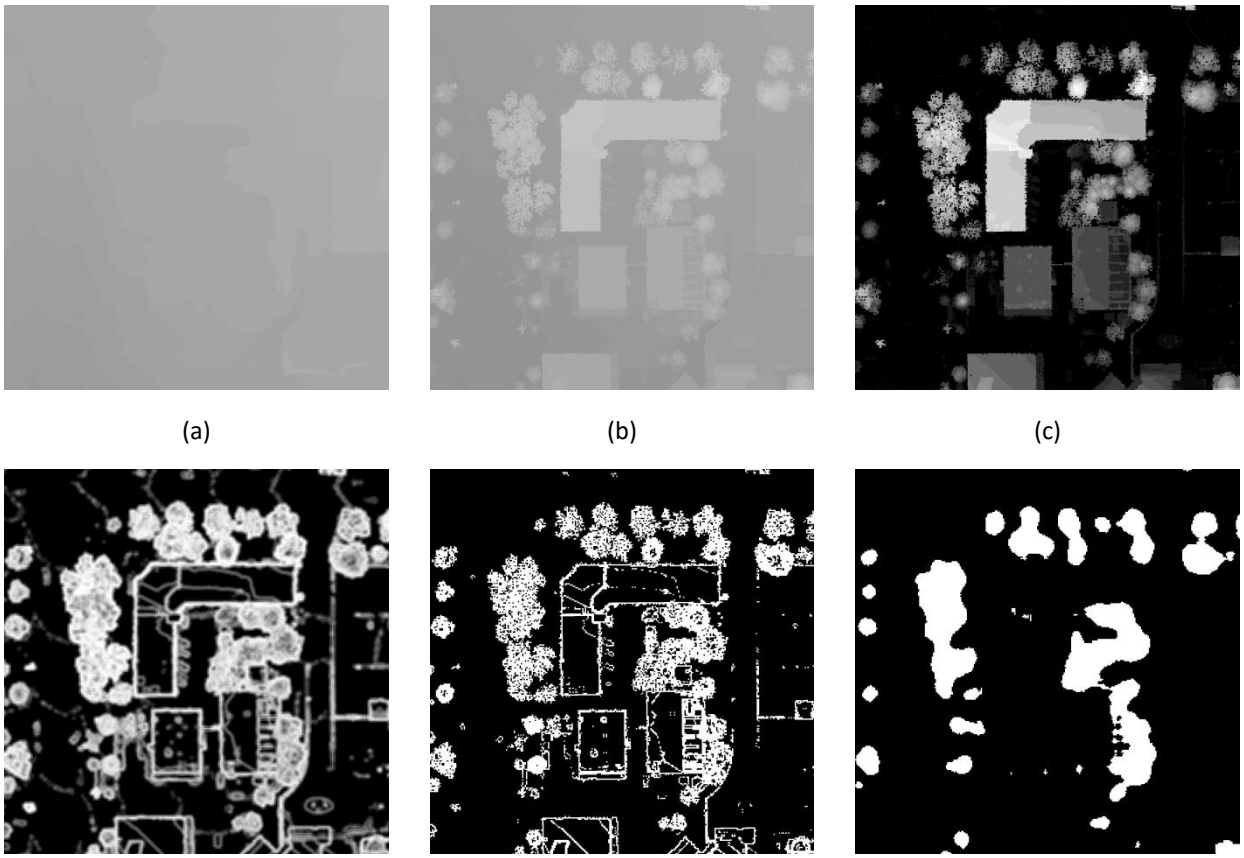

Figure 2. (a) the digital surface model, (b) the digital terrain model, (c) the normalized digital surface model, (d) the

slope of the normalized digital surface model, (e) the reclassified slope, (f) the final raster showing trees after ANDing high slope pixels with high terrain pixels and applying a median filter.

Determining Service Areas

Service areas represent areas that can be served within a prescribed time or distance threshold. These were determined by ORing the binary raster of building pixels with the binary raster of tree pixels to form a new raster representing obstacles. Current EMS locations were converted from points to raster pixels. The Path-Distance tool in ArcGIS was used to calculate the distance from each EMS pixel to each blank pixel in the obstacle raster. A cut-off of 2km was used when generating service areas, this represents the approximate range of drone with 1 minute of flying time at a top speed of 60 mph (Husten 2014). It is important to note that this algorithm only considers "Queens" movements meaning that a path must go through the center of pixels rather than just crossing a section of the pixel. This means that the calculated distance is over-estimated.

Results

As seen below in Figure 3, the majority of buildings were successfully detected. Since a ground-truth dataset of buildings was not available, no accuracy statistics are provided. However, by examining the aerial imagery, some errors can be found. As seen in the center of Figure 3, there is a section of grass that is classified as a building. This is the result of the bounding box, which assumes that all buildings are rectangular. There are very few false positives (detecting a building when a building is not present), and some false negatives (a building that should have been detected but was not).

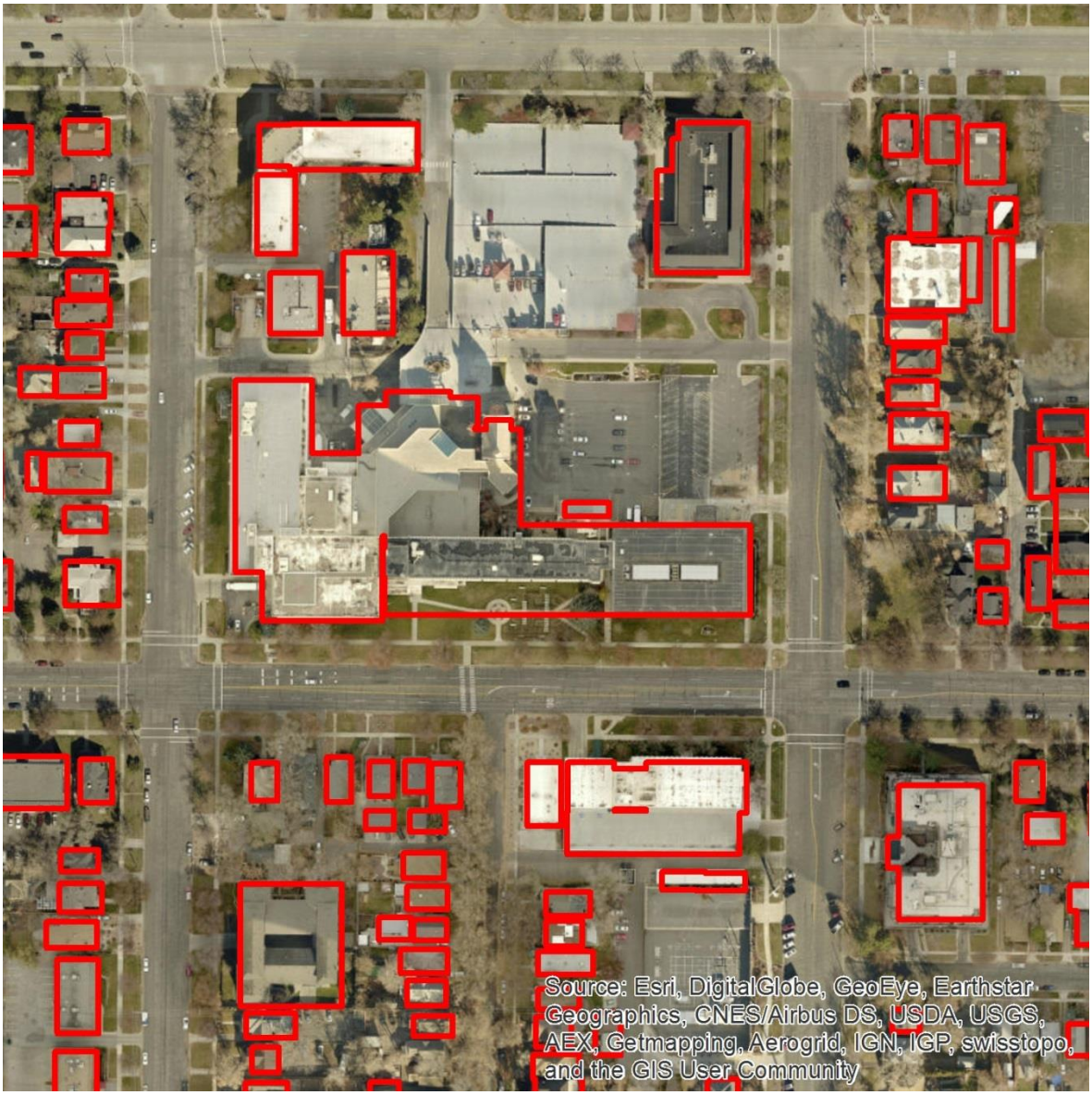

Figure 3. A zoomed in section of Salt Lake City where detected buildings are enclosed in red.

Trees were also successfully detected (Figure 4). Unfortunately, there was no ground-truth tree data to compare my results with. However, by examining Figure 4, most trees appear to be detected. Since the aerial imagery and the LIDAR data do not match up temporally, it is difficult to tell if there are over-estimates or under-estimates of trees. There are some small polygons that clearly do not represent trees but generally they represent small obstacles on or near buildings.

Figure 4. A zoomed in section of Salt Lake City showing the detected trees enclosed in red.

The derived service areas are shown in Figure 5. Greener areas represent areas closer to EMS stations. Black areas represent areas that were served before obstacles were included in the generation of service areas. There is a clear reduction in the range of service of medical drones when obstacles are avoided. It is worth noting that it is unlikely that a drone will maintain top speed as it avoids obstacles, leading to an over-estimation of the distance a drone can fly within 1 minute.

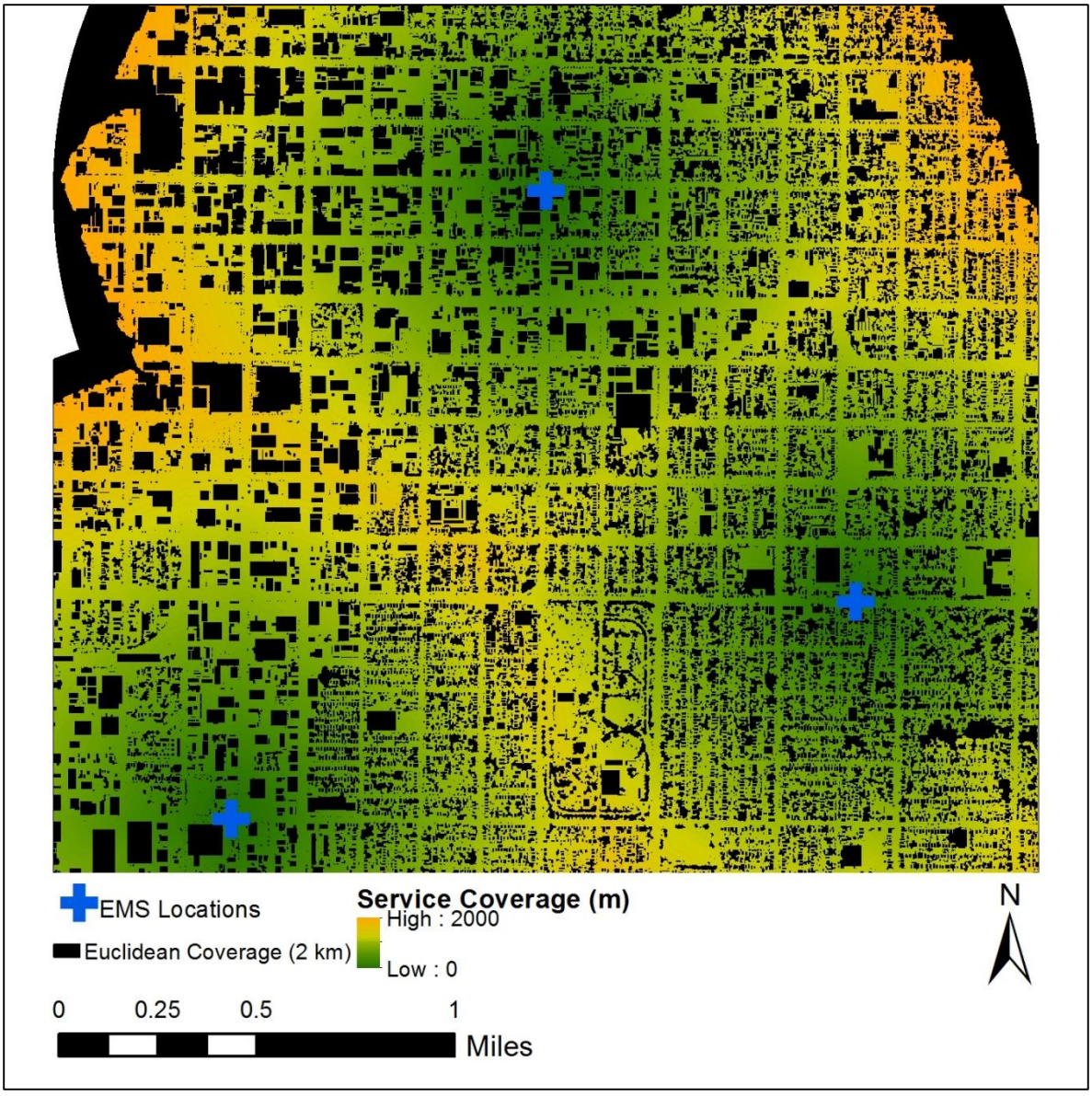

Figure 5. The derived service coverage from three EMS stations.

Figure 6 shows estimated paths taken by a drone flown a few meters off of the ground from three EMS stations to 3 simulated incidents. The yellow lines, are the ideal naïve paths that do not consider any obstacles. The red lines show the paths taken by drones to avoid trees and buildings. As expected, each path that considered obstacles resulted in a longer flight distance (Table 1). In best case scenarios, such as flying along a road, the difference is negligible since there are few, if any obstacles. However, in most cases, the drone will have to fly around trees and buildings experiencing a significant drop in service coverage.

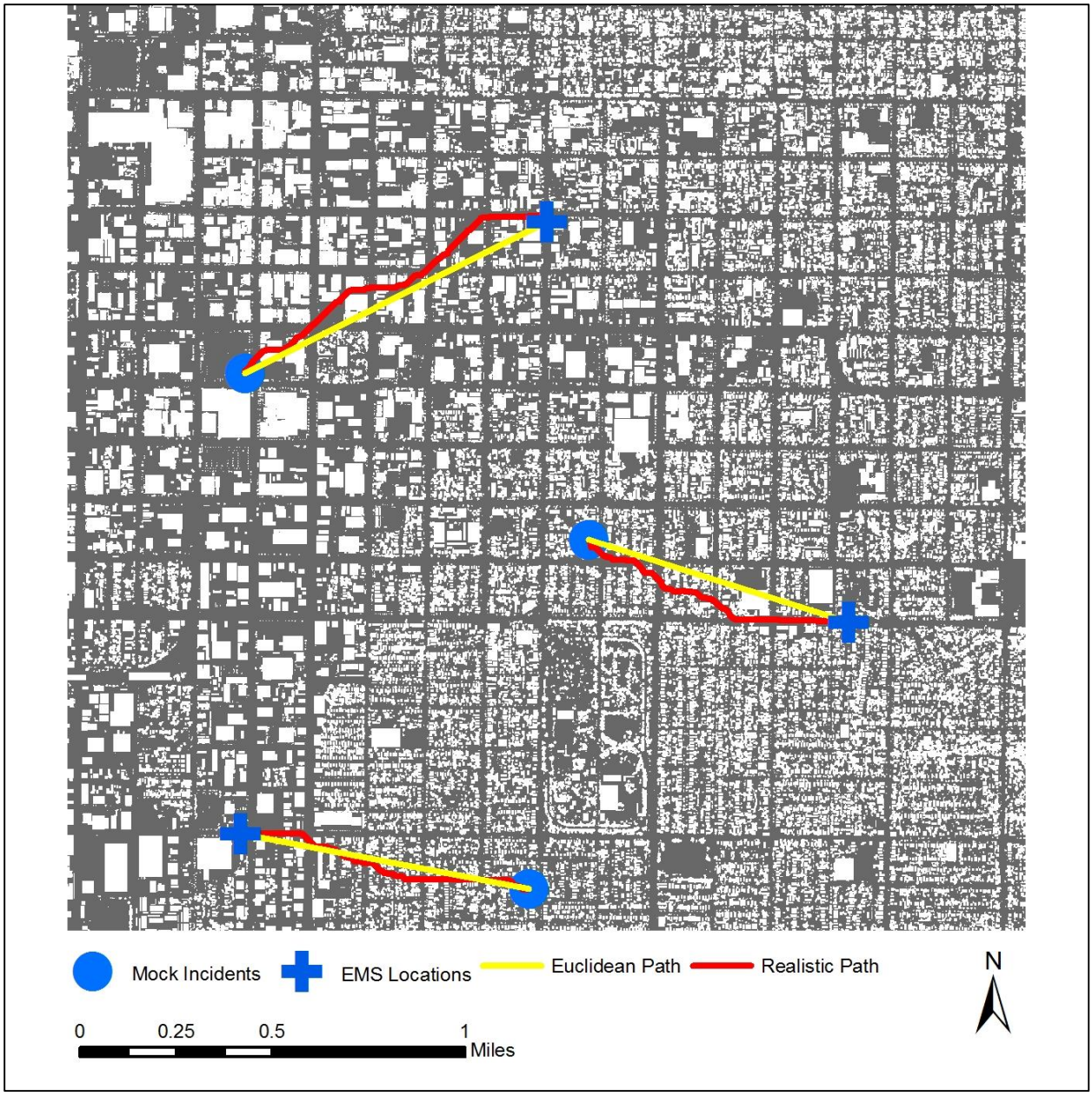

Figure 6. This shows the paths taken by a drone flying a few meters off the ground from an EMS station to a simulated incident. The yellow lines show a direct flight. The red lines show a realistic path that accounts for buildings and trees. The basemap shows buildings and trees in white.

Path	Euclidean Distance (m)	Realistic Distance (m)	Difference (m)
1	1407.6	1557.9	150.3
2	1137.1	1250.8	113.7
3	1225.7	1298.3	72.6

Table 1. A summary of the path distances and the extra distance required to avoid obstacles.

Conclusions

LIDAR data were successfully used to classify low altitude obstacles as either trees, buildings, or neither. The classification results were verified using aerial imagery and despit some misclassifications, trees and buildings were sucessfully detected. Service areas were derived which specifically avoid the detected obstacles. Based on Figure 6 and Table 1, obstacles signficantly affect the range of medical drones. It is worth noting that in many cases the drones will be flying above the trees and buildings, however obstacles may affect potential landing and take-off zone, and short range trips. This study can be improved in several ways. First, using more advanced image processing filters can probably reduce the amout of misclassifications and provide a better set of obstacles. Secondly, using a better algorithm to determine service areas that does not rely on "Queens" movements will provide better travel distance and time estimates. For example, visiblity graphs could be generated and then shortest path algorithms such as Dijsktra and A* could be used to determine more optimal flight paths. Although there is room for improvement, the methodologies used in this study were successful and provided meaningful results.

References

Abd Rahman, MZ, and BGH Gorte. 2008. Individual tree detection based on densities of high points of high resolution airborne LiDAR. Paper read at GEOBIA 2008: Pixels, Objects, Intelligence GEOgraphic Object based Image Analysis for the 21st Century, 5-8 August 2008, Calgary, Canada.

Demir, N, D Poli, and E Baltsavias. 2008. Extraction of buildings and trees using images and LiDAR data. *ISPRS Proceedings, Beijing*.

Ekhtari, Nima, MR Sahebi, MJ Valadan Zoej, and A Mohammadzadeh. Automatic building detection from LIDAR point cloud data.

Everaerts, J. 2008. The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37:1187-1192.

Husten, Larry. 2014. Grad Student Invents Flying Ambulance Drone to Deliver Emergency Shocks. *Forbes*.

Jimenez Lugo, J., and A. Zell. 2013. Framework for autonomous onboard navigation with the AR.Drone. Paper read at Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, 28-31 May 2013.

Koch, Barbara, Ursula Heyder, and Holger Weinacker. 2006. Detection of individual tree crowns in airborne lidar data. *Photogrammetric Engineering & Remote Sensing* 72 (4):357-363.

Krajnik, T., M. Nitsche, S. Pedre, L. Preucil, and M. E. Mejail. 2012. A simple visual navigation system for an UAV. Paper read at Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on, 20-23 March 2012.

Meng, Xuelian, Le Wang, and Nate Currit. 2009. Morphology-based building detection from airborne LIDAR data. *Photogrammetric Engineering & Remote Sensing* 75 (4):437-442.

Mercado, D. A., P. Castillo, and R. Lozano. 2015. Quadrotor's trajectory tracking control using monocular vision navigation. Paper read at Unmanned Aircraft Systems (ICUAS), 2015 International Conference on, 9-12 June 2015.

Misener, Paul. 2014. Amazon Petition for Exemption.

Rottensteiner, Franz, and Christian Briese. 2002. A new method for building extraction in urban areas from high-resolution LIDAR data. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34 (3/A):295-301.

Thiels, Cornelius A, Johnathon M Aho, Scott P Zietlow, and Donald H Jenkins. 2015. Use of unmanned aerial vehicles for medical product transport. *Air medical journal* 34 (2):104-108.

Webredactie Communication. 2014. TU Delft's ambulance drone drastically increases chances of survival of cardiac arrest patients. Delft, The Netherlands: TU Delft.