Rooftop Detection using Aerial Drone Imagery

Kritik Soman
Department of Electrical Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India
kritiksoman@ieee.org

ABSTRACT

We present a rooftop detection algorithm using aerial RGBD and near infrared data which uses lower computational resources than algorithms requiring GPUs. The depth data is extracted from multiview images captured by drones using photogrammetry. Our approach is cost effective as compared to LIDAR surveying and has lower edge blurring. It is also novel in terms of segmenting clutter due to objects such as overhead water tanks on roofs. This helps is determining the actual free roof area that would be available for applications like solar panel deployment. The algorithm was evaluated on the aerial imagery of rooftops on a hill slope of size 12876×10533 pixels and an F1 score of 88.27% was obtained. The algorithm ran in under 2 minutes on a Google Cloud instance with Intel Xeon E5 processor.

CCS CONCEPTS

• **Computing methodologies → Image segmentation**;

KEYWORDS

Rooftop Detection, Photogrammetry, Intel Distribution for Python, Aerial Drone Imagery

ACM Reference format:
Kritik Soman. 2019. Rooftop Detection using Aerial Drone Imagery. In *Proceedings of CoDS-COMAD 19, Kolkata, India, January 3-5, 2019,* 4 pages.
https://doi.org/10.1145/3297001.3297041

1 INTRODUCTION

With greater demand for clean energy production, solar panel installations on rooftops has been rising. Companies like Tesla have commercialized solar roof tiles [6] and have begun widespread deployment in cities. For cities (especially in developing countries) to quickly switch to solar, low cost and fast rooftop surveying is highly in demand.

Aerial surveying can be done through Light Detection and Ranging (LIDAR) using Unmanned Aerial Vehicles (UAVs) and satellites. While LIDAR UAV surveying provides better accuracy, the Digital Elevation Model (DEM) generated using LIDAR suffers from edge blurring. Though this can be partially rectified by using reconstruction techniques [12], the overall cost associated with acquisition and maintenance are high. This makes LIDAR surveying less practical. Satellite aerial images have been available for a long time, but due to variability in building structures and shapes, rooftop detection using only RGB satellite images [9, 14] has not been highly successful. Newer satellites like WorldView-3 provide high resolution multispectral data [7].

In theory, computer vision algorithms can exploit the spectral reflectance signature [15] in the multispectral data of different materials and identify different regions. This typically requires heavy computational resources (including GPUs) for processing the 8 band multispectral data for a large ground area.

DEM can also be generated from high resolution multiview aerial imagery from drones using photogrammetry. These have much lower edge blurring as compared to LIDAR and can be generated at a lower cost. With newer CMOS optical sensors and high accuracy GPS, drone imagery can offer 1cm ground resolution [1], which is much higher than the 50cm resolution of the SpaceNet Dataset [5] obtained from satellite. This can be helpful in detecting small objects on rooftops like water tanks and determining the actual usable solar panel deployment area. Drone images are unaffected by clouds or atmospheric dust particles, can be captured at any time, and without much setup time as compared to satellite imagery. Nowadays, drones [3] also have infrared sensors which can help in detection of visible vegetation. Hence, rooftop detection using RGB, IR orthophoto and DEM from photogrammetry is an interesting and challenging problem. We captured aerial data [3] and used imagery with horizontal and vertical resolution of 5 cm and 1 cm respectively. The drone was flown at aroud 50m above ground level. Our main objective in this paper is to develop a cost efficient annotation algorithm for aerial imagery which uses minimal computational resources.

2 PROPOSED ALGORITHM

Our proposed algorithm can be be summarized as shown in the flow chart in Fig. 1. We began by collecting multiview aerial images [1] of a few places and then generated the corresponding DEMs. The datasets were then divided into test and train sets. Each step in the flow chart has been explained in detail in the following subsections.

2.1 Preprocessing and DEM segmentation

The DEM was first interpolated to take care of some missing elevation values and then the RGB and IR orthophoto were downscaled using bilinear sampling to match the resolution of the DEM. To segment the DEM, we exploited the fact that all buildings will have step edges at their boundary. Simply applying an edge detector could miss buildings when the elevation data is blurred as shown in the white box in Fig. 2b. We therefore sharpened the DEM by using the following steps:

(1) Blurring original DEM with Gaussian filter, kernel size 6.
(2) Blurring output with Gaussian filter, kernel size 1

CoDS-COMAD 19, January 3-5, 2019, Kolkata, India

ACM ISBN 978-1-4503-6207-8/19/01...$15.00
https://doi.org/10.1145/3297001.3297041

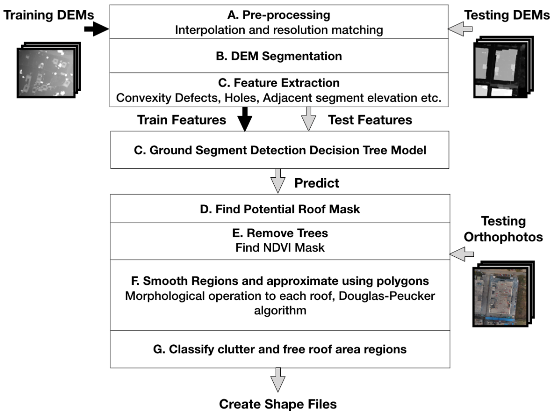
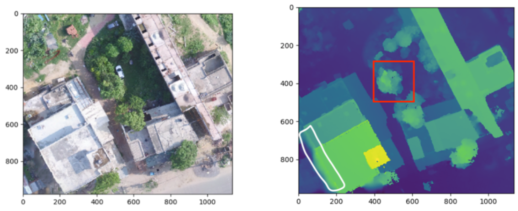
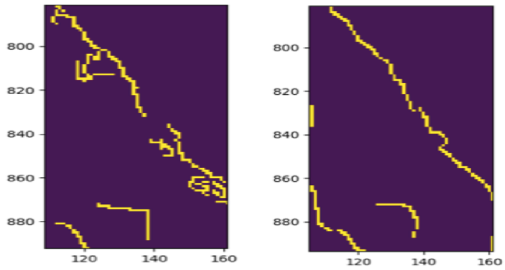

Training DEMs
A. Pre-processing
Interpolation and resolution matching
Testing DEMs
B. DEM Segmentation
C. Feature Extraction
Convexity Defects, Holes, Adjacent segment elevation etc.
Train Features
Test Features
C. Ground Segment Detection Decision Tree Model
Predict
D. Find Potential Roof Mask
E. Remove Trees
Find NDVI Mask
Testing
Orthophotos
F. Smooth Regions and approximate using polygons
Morphological operation to each roof, Douglas-Peucker algorithm
G. Classify clutter and free roof area regions
Create Shape Files

Figure 1: Flow chart of rooftop detection algorithm.



(a) (b)

Figure 2: Orthophoto and DEM.



(a) (b)

Figure 3: Effect of sharpening on output of edge detector.

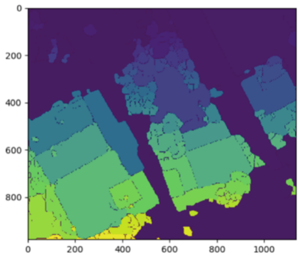
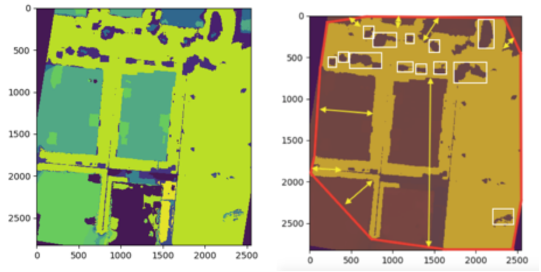


Figure 4: Segmented DEM.

(3) Constructing the sharpened DEM by adding the original DEM to the difference of the two blurred DEMs obtained above, scaled by a factor $\alpha = 20$.



(a) Label matrix of segmented DEM

(b) Ground segment.

Figure 5: A location in Noida, India.

We then use the canny edge detector on the sharpened DEM. The difference in edge detection before and after following the above steps in shown in Fig. 3. We might still miss some edges by a few pixels, so we dilate the result and then skelotonize it to prevent the final segments from being too eroded. Next we segment the DEM by subtracting the binary output of skeleton of the edges from a all-ones matrix and finally label each unconnected segment. The output has been shown in Fig. 4. Sharpening the DEM also helps in capturing the entropy in elevation data (shown in red box in Fig. 2b) that may be present for trees, by breaking segments into smaller ones corresponding to presence of trees, as shown in Fig. 4.

2.2 Features for ground segmentation

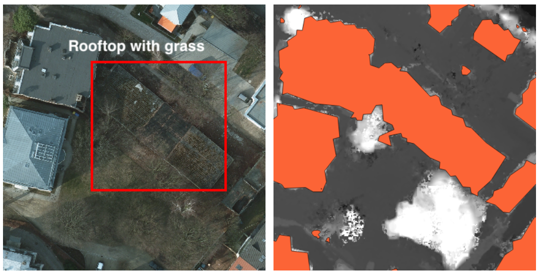
The most common practice [8, 10] is to obtain the Digital Terrain Model (DTM) of the ground area and subtract it from the Digital Surface Model (DSM) to obtain the normalized DSM (nDSM). DTM can be obtained from old satellite data but that will have much lower resolution than the DSM obtained from photogrammetry. Hence, this would result in errors in elevation data on performing subtraction. In case of LIDAR, ground segmentation is performed using Progressive Morphological Filter [17] available in Point Data Abstraction Library (PDAL) [4]. However, since we are using photogrammetry, we cannot use this method for generating DTM.

Therefore, in our method, we only identify the ground segments and check which adjacent segment is greater than its elevation by the minimum building height (h_{min}). Adjacency in the labelled matrix was determined by the following steps for each segment in its bounding box:

(1) Create mask of current segment and dilate it.
(2) Multiply the dilated mask matrix with the current segment matrix.
(3) Find unique labels in the resultant matrix other than the label of the current segment.

Multi-threading was used to identify the list of adjacent segments for each segment in the segment labelled matrix. The adjacency list of each segment was stored and the following features were computed for each segment:

(1) Convexity defects: The ground segments usually have convexity defects with large depth and may also have a large number of total number of defects. This is common in cases when a ground segment has many adjacent building segments. An example has been shown in Fig. 5b. The ground



(a) Orthophoto **(b) Rooftop overlaid on DEM**

Figure 6: Rooftops with grass in ISPRS dataset

segment is highlighted in yellow with its convex hull plotted in red. The large convexity defects have also been marked with yellow lines.

(2) Holes: Ground segments tend to have many holes in them due to presence of buildings, trees, cars etc. The example in Fig. 5b also has many holes marked with a white bounding box. The number of holes, average area of holes, and number of large holes were used as features.

(3) Number of adjacent segments which are taller by h_{min}.

A decision tree was trained with the above features generated from the training DEM and used to identify which of the segments were ground in the testing DEM.

2.3 Creating potential roof mask

The remaining non ground segments in the testing DEM were then checked if they were taller than any adjacent ground segment by h_{min} and a mask was created for those potential roof segments. For each non ground segment, the adjacent ground segment was found simply by finding the intersection of its adjacency list and ground segment label list. To prevent missing out on building segments that are not directly adjacent to a ground segment, we iterated through the remaining segments and checked if they were taller than the ground segment of their adjacent building segment, if any. Using this method, we were able to skip the interpolation required in the step of generation of DTM from point cloud and the generation of nDSM.

2.4 Removing false positives due to trees

A simple approach to segment vegetation was to transform the RGB image to LAB color space for better perceptual uniformity and threshold the LAB values. But this method missed out several trees when its LAB values were outside the chosen threshold. Since vegetation usually had low reflectance in the blue and red (R) wavelength regions of the spectrum (due to absorption by chlorophyll for photosynthesis) and high reflectance in the near infrared (NIR) region (due to the cellular structure in the leaves), vegetation can be identified by high NIR and low visible reflectance [15].

This effect is captured by the Normalized Difference Vegetation Index (NDVI) [16], which is defined as shown in 1.

$$NDVI = \frac{NIR - R}{NIR + R} \tag{1}$$

Its value lies between -1 and 1 and can be threshold between 0.2 and 0.9 to find the regions with vegetation. The binary resultant matrix is then multiplied to the labelled roof mask matrix and for each segment, we find the ratio of the segment area covered in NDVI to the area of the entire segment. If the ratio is more than 0.5 and the total segment area is small, we simply remove those segments from the roof mask. This is done to only remove trees from roof mask since false positives in NDVI due to presence of grass on roofs could be present in the dataset as shown in Fig. 6. The idea behind doing this is that tree canopies have entropy in elevation and the edge detector we use will break the tree regions into small segments.

2.5 Smoothing roof tops and approximating using polygons

On removing the tress from the roof mask, several unwanted holes could be generated. This can be corrected by selectively applying morphological operations to each roof region. To do this, the roof mask was first labelled and the bounding box (with some extra margin) of the mask of each region was taken and, dilation and erosion were applied iteratively to remove very small regions and join very close ones. The resultant mask was then assembled into a new roof mask. Finally, each region was approximated into a polygon using the Douglas-Peucker algorithm [11] with tolerance equal to 1% of the contour length of a roof and a maximum tolerance of 10 pixels. The regions with very small area were then classified as clutter and the remaining as free roof area. These steps were also multi-threaded to increase the overall speed. The resulting polygons are then saved as a shape file by using the same geo-reference as the DEM and orthophoto after pre-processing.

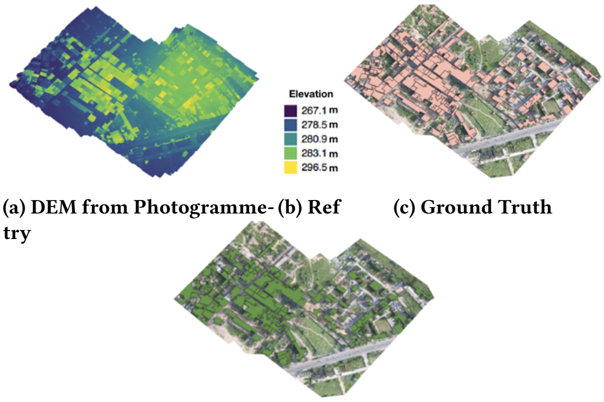
Since we are were not using a GPU and only relied on Intel Xeon E5 v3.2.3 GHz CPU on Google Cloud, we decided to use Intel Distribution for Python (IDP) to take advantage of NumPy, SciPy and Scikit-learn that were accelerated by Intel Math Kernel Library (MKL).

3 RESULTS

We tested our algorithm on a dataset where there is slope in the ground level. The DEM of the region (Surajpur, Greater Noida, Uttar Pradesh, India) is shown in Fig. 7a below. The elevation level (around 280m above sea level) of the buildings in the left is comparable to the elevation level of the ground on the right. The result of running the algorithm on this data has been overlayed on the orthophoto and shown in Fig. 7d. By clustering the DEM and comparing each potential roof segment to nearest adjacent ground segment, the variation in the elevation of the ground level is taken care of. We then manually labelled the ground truth by visual inspection (shown in Fig. 7c) in QGIS [13] and found the **precision, recall and F1 score to be 87.33, 89.23 and 88.27% respectively.**

We also ran the algorithm on the International Society for Photogrammetry and Remote Sensing (ISPRS) Potsdam dataset [2]. The results overlayed on the orthophoto is shown in Fig. 8. The dataset consisted of multiple tiles, so the algorithm was run individually on each tile.

It is important to note here that we used the DSM and not the nDSM and that the source of the elevation data in this dataset was from LIDAR. We observed that the edges of buidings in this dataset were much more blurred unlike our datasets (as shown in Fig. 9),



(a) DEM from Photogrammetry **(b) Ref** **(c) Ground Truth**

(d) Result imposed on orthophoto.

Figure 7: Results from our approach.

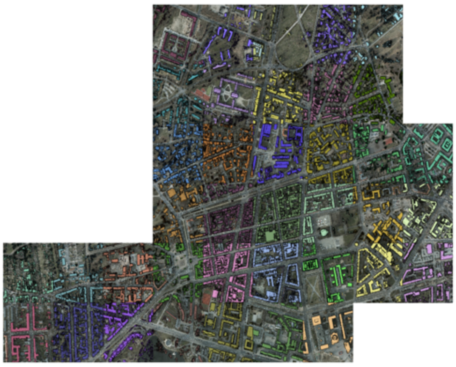
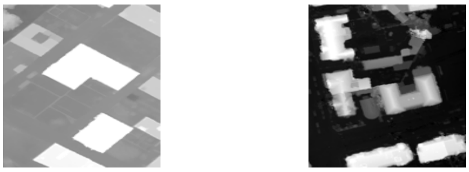


Figure 8: Rooftop detection on ISPRS Potsdam dataset.



(a) Photogrammetry **(b) LIDAR**

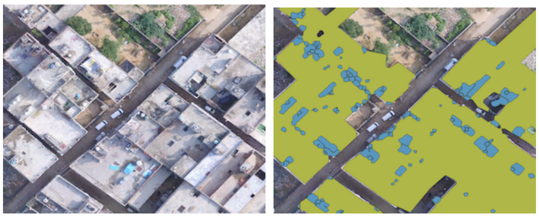
Figure 9: Blurring of edges in DEM

resulting in improper segmentation. Hence, our algorithm did not generate results comparable to the photogrammetry generated DEM.

Apart from the above downside, our method using photogrammetry has the benefit of segmenting clutter on roofs. We classify small segments present on the roof mask as clutter. This helps in estimating the actual roof area that is available for potential solar panel deployment. An example of the roof mask and clutter that is segmented is shown in Fig. 10. This would be hard to achieve on using DEM generated from LIDAR or satellite.

4 CONCLUSIONS AND FUTURE WORK

We presented a rooftop detection algorithm which uses comparatively lower computational resources as no GPU was used and is cost effective as compared to LIDAR based methods. The algorithm is able to handle cases when grass is present on roofs and is also able to segregate clutter present on rooftops. It also does not require a DTM for detecting roof segments. As future work, support for LIDAR DEM source would also be incorporated.



(a) Orthophoto **(b) Result**

Figure 10: Clutter segregated from actual rooftop solar installation area

REFERENCES

[1] [n. d.]. 3DR Site Scan Drone Survey Accuracy White Paper. https://3dr.com/resources/books-guides/white-papers/site-scan-accuracy/. ([n. d.]). Accessed: 2018-08-30.

[2] [n. d.]. ISPRS Test Project on Urban Classification and 3D Building Reconstruction. http://www2.isprs.org/commissions/comm3/wg4/detection-and-reconstruction.html. ([n. d.]). Accessed: 2018-08-30.

[3] [n. d.]. New Thermal Camera Integration for Solo. https://3dr.com/blog/new-thermal-camera-integration-for-solo/. ([n. d.]). Accessed: 2018-08-30.

[4] [n. d.]. Point Cloud Processing and Analysis with PDAL. https://pdal.io/workshop/exercises/analysis/dtm/dtm.html. ([n. d.]). Accessed: 2018-08-30.

[5] [n. d.]. SpaceNet Challenge. https://spacenetchallenge.github.io/. ([n. d.]). Accessed: 2018-08-30.

[6] [n. d.]. Tesla:Solar Panels. https://www.tesla.com/solarpanels. ([n. d.]). Accessed: 2018-08-30.

[7] [n. d.]. WorldView. https://www.satimagingcorp.com/satellite-sensors/worldview-3/. ([n. d.]). Accessed: 2018-08-30.

[8] Emmanuel Baltsavias, Scott Mason, and Dirk Stallmann. 1995. Use of DTMs/DSMs and orthoimages to support building extraction. In *Automatic extraction of man-made objects from aerial and space images.* Springer, 199–210.

[9] Hayk Baluyan, Bikash Joshi, Amer Al Hinai, and Wei Lee Woon. 2013. Novel approach for rooftop detection using support vector machine. *ISRN Machine Vision* 2013 (2013).

[10] Liang-Chien Chen, Tee-Ann Teo, Yi-Chen Shao, Yen-Chung Lai, and Jiann-Yeou Rau. 2004. Fusion of LIDAR data and optical imagery for building modeling. *International Archives of Photogrammetry and Remote Sensing* 35, B4 (2004), 732–737.

[11] David H Douglas and Thomas K Peucker. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization* 10, 2 (1973), 112–122.

[12] Ahmed F Elaksher, James S Bethel, et al. 2002. Reconstructing 3d buildings from lidar data. *International Archives Of Photogrammetry Remote Sensing and Spatial Information Sciences* 34, 3/A (2002), 102–107.

[13] GIS Quantum. 2013. Development Team.(2013). Quantum GIS geographic information system. Open Source Geospatial Foundation Project. (2013).

[14] Parvaneh Saeedi and Harold Zwick. 2008. Automatic building detection in aerial and satellite images.. In *ICARCV*. 623–629.

[15] Gary A Shaw and Hsiaohua K Burke. 2003. Spectral imaging for remote sensing. *Lincoln laboratory journal* 14, 1 (2003), 3–28.

[16] Fei Yuan and Marvin E Bauer. 2007. Comparison of impervious surface area and normalized difference vegetation index as indicators of surface urban heat island effects in Landsat imagery. *Remote Sensing of environment* 106, 3 (2007), 375–386.

[17] Keqi Zhang, Shu-Ching Chen, Dean Whitman, Mei-Ling Shyu, Jianhua Yan, and Chengcui Zhang. 2003. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE transactions on geoscience and remote sensing* 41, 4 (2003), 872–882.