Multiresolution-Based 3-D Terrain Estimation Algorithms for Complex Urban Environments

Conference Paper · January 2019

DOI: 10.2514/6.2019-1194

4 authors, including:

Madhur Tiwari
Florida Institute of Technology
29 PUBLICATIONS **67** CITATIONS

SEE PROFILE

Richard Prazenica
Embry–Riddle Aeronautical University
91 PUBLICATIONS **876** CITATIONS

SEE PROFILE

AIAA SciTech Forum
7-11 January 2019, San Diego, California
AIAA Scitech 2019 Forum

10.2514/6.2019-1194

Multiresolution-Based 3-D Terrain Estimation Algorithms for Complex Urban Environments

Pedro Vergara, Madhar Tiwari, Andres Chavez, Richard Prazenica, and Troy Henderson

Embry-Riddle Aeronautical University, Daytona Beach, FL 32114

This paper investigates the development of multiresolution-based terrain estimation algorithms, focusing on complex three-dimensional urban environments. The algorithms generate functional terrain representations from point cloud data obtained from the processing of onboard vision and/or LIDAR sensors. Two classes of multiresolution algorithms are considered in this work. First, mathematical learning methods are developed and implemented to generate adaptive terrain representations in terms of piecewise-constant multiresolution basis functions. These algorithms have been applied in previous work for terrain estimation over two-dimensional domains and are extended in this paper for three-dimensional domains, resulting in terrain representations in terms of multiresolution occupancy grids. The second class of algorithms corresponds to global-orthogonal mapping (GLO-Map) algorithms, which have been developed in previous work to generate representations over two-dimensional domains in terms of smooth orthogonal functions with varying resolution. Representative terrain mapping examples are provided using simulated LIDAR data from a virtual urban environment as well as processed 3-D LIDAR data collected from a UAV flying through a representative obstacle environment.

I. Introduction

A critical enabling technology for autonomous UAV flight through complex urban environments is the ability to map the scene based on sensor data. An adaptively generated 3-D terrain map would enable an autonomous vehicle to plan and update 3-D trajectories around and through diverse features such as trees, power lines, buildings, bridges and overhangs, and potentially indoors. This paper focuses on developing multiresolution-based terrain representations suitable for the complex 3-D features that are commonly found in urban environments.

This paper investigates two classes of multiresolution terrain mapping algorithms, with a focus on developing implementations that are fast (real-time), recursive (i.e., no need to regenerate the entire terrain when data are received), and suitable for 3-D terrain features (i.e., can capture terrain features beyond a 2-D terrain skin representation). These algorithms operate on sensed 3-D point cloud data (i.e., a collection of 3-D points in an inertial reference frame) that can be obtained from LIDAR sensors, stereo vision systems, or structure from motion processing of monocular imagery. Two multiresolution-based approaches will be explored and modified for the purposes of generating complex 3-D terrain representations:

(1) Adaptive mathematical learning algorithms [1]: In previous work [2, 3], these algorithms have been implemented in the form of fast, recursive algorithms that provide multiresolution, piecewise-constant representations on 2-D domains. In this paper, the extension of these adaptive learning algorithms to 3-D terrain representations is investigated.

(2) Global-local orthogonal mapping (GLO-Map) [4 – 6]: These algorithms are currently in the form of efficient, multiresolution-based algorithms that employ smooth orthogonal functions over 2-D domains.

This paper describes both classes of multiresolution-based methods, and representative 3-D terrain representations are provided for complex urban environments. Examples are presented using simulated LIDAR and/or vision data from a virtual environment as well as using processed LIDAR from UAV data collection experiments.

II. Adaptive Mathematical Learning Algorithms

Mathematical learning algorithms, which have been developed and implemented in previous work [1 – 3], can be employed to generate an adaptive, multiresolution-based terrain map. The use of adaptive multiresolution greatly reduced the number of basis functions required in the terrain representation, increasing computational speed and efficiency. The algorithm is adaptive in that the terrain representation is locally updated as new data are collected, which leads to fast, real-time implementations.

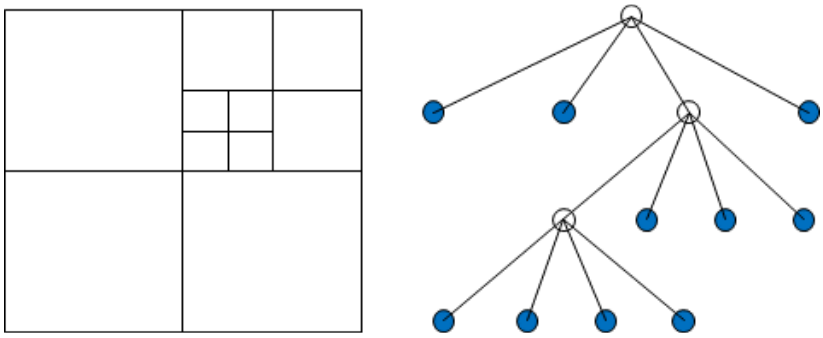


Figure 1: Adaptive Partitioning of a 2-D Domain.

Figure 1 illustrates the adaptive partitioning of a two-dimensional domain. The adaptive learning algorithm approximates the terrain in terms of piecewise-constant (multiresolution) functions that are supported over each rectangular subdomain:

$$f(x,y) \approx \sum_I c_I \chi_I(x,y)$$

where χ_I denotes the characteristic (constant) function over subdomain I, and the coefficient c_I is computed as the average height of all points that are in the subdomain I. It should be noted that other statistics can be used to define the value of these coefficients. The decision to subdivide a subdomain I into 4 higher-resolution subdomains is based on the variance of the data within the subdomain I:

$$\sum_{z \in I} (z_i - c_I)^T (z_i - c_I) > \tau_m$$

That is, if the variance exceeds a specified threshold, the subdomain is divided to achieve higher resolution. If the variance is low and does not exceed the threshold, higher resolution does not provide a significant benefit and the subdomain is not divided.

Figure 2 presents a sample point cloud and images obtained from simulated LIDAR and vision sensors within a virtual urban environment. This point cloud was used to generate several terrain representations, as shown in Figures 3 and 4, by varying different parameters in the adaptive learning process. In this

implementation, a two-dimensional (north-east) domain is subdivided into rectangular regions, and the terrain height is approximated by a constant function over each region. Regions are locally subdivided into smaller regions based on new point cloud data. If the point cloud data that reside in a specific region show significant variance in the z-direction (height), the algorithm divides the region into 4 subregions and approximates the terrain in each subregion as a constant function. The overall adaptive learning process results in a multiresolution-based terrain representation in terms of piecewise-constant functions.

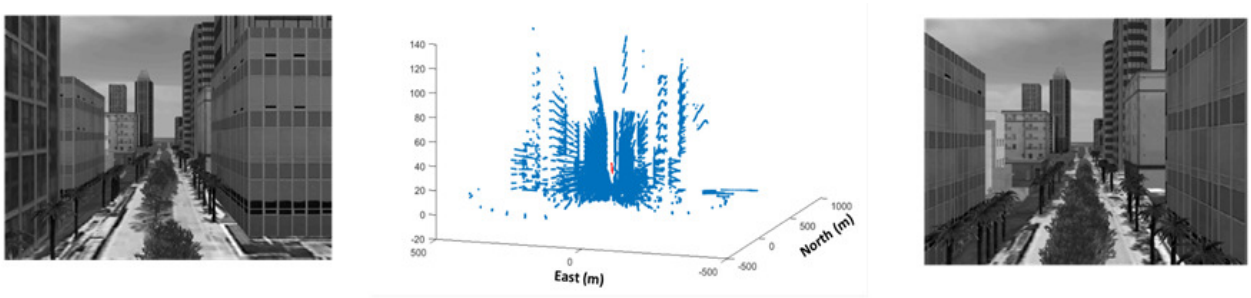


Figure 2: Simulated LIDAR Point Cloud and Images from the Virtual Urban Environment.

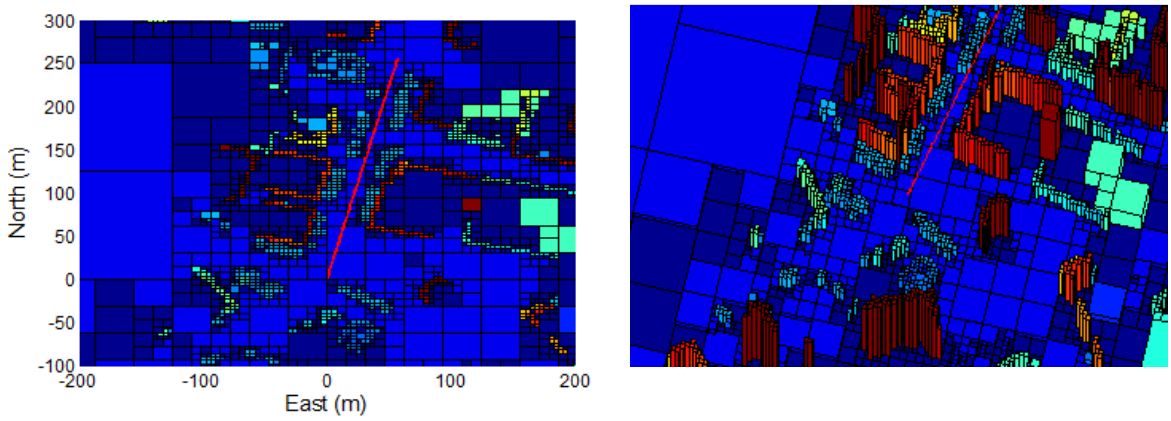
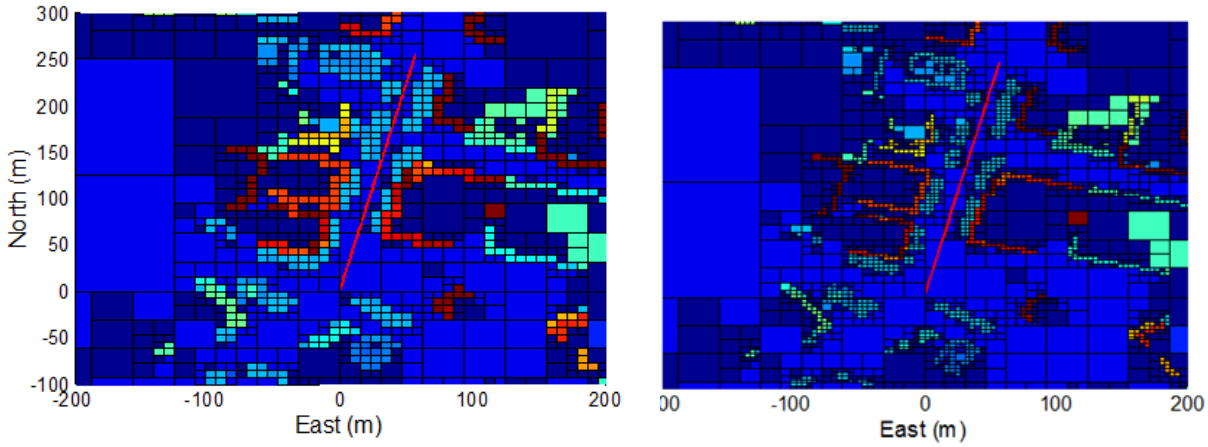


Figure 3: Multiresolution Terrain Representation using the Adaptive Learning Algorithm.



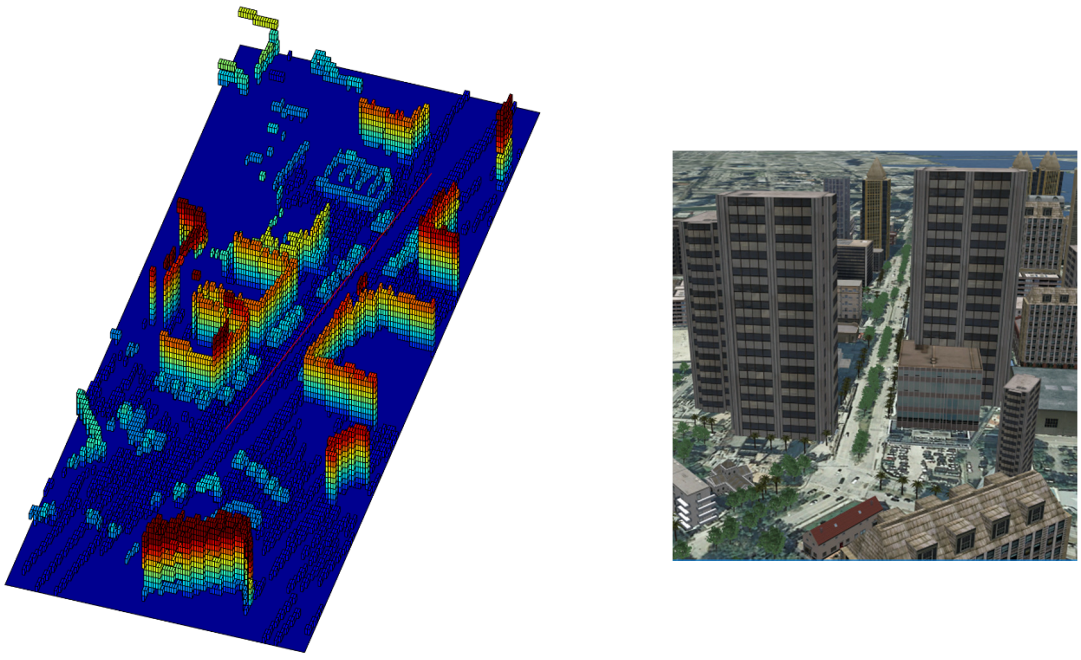
(a) Level 7 Resolution **(b) Level 8 Resolution**

Figure 4: Adaptive Terrain Maps using Different Resolution Levels.

During the simulations, the vehicle with sensors was translated through an urban canyon. The terrain representations in Figures 3 and 4 show that the LIDAR-based terrain representations capture the sides of several buildings in the canyon as well as the tops of several trees that are in the middle of the canyon, below the flight path. Figure 4 shows the effect of increasing the maximum resolution level by a factor of two. The Level 8 representation clearly provides higher resolution and more detail, but it also increases the number of basis functions from 1942 to 3940.

These adaptive learning algorithms were then extended by the authors to provide three-dimensional terrain representations. The three-dimensional algorithm required extending the adaptive two-dimensional subdivision algorithm (see **Error! Reference source not found.**), which resembles a multiresolution quadtree representation, to the adaptive partitioning of a three-dimensional domain, resembling a multiresolution octree representation. Therefore, an overall three-dimensional domain is defined and then adaptively partitioned into subdomains that resemble cubes or rectangular prisms.

A significant difference in the three-dimensional terrain algorithm compared to the two-dimensional implementation is that the terrain representation over each cubic subdomain is now a binary function; that is, each subdomain is marked as occupied or unoccupied based on the point cloud data. In contrast, the two-dimensional implementation employs piecewise-constant basis functions, so that each subregion has a corresponding height. In the three-dimensional algorithm, the decision to subdivide a given cubic region into 8 subregions is based on the number of 3-D points that are within that region. In the current implementation, once the number of points in a cubic region exceeds a specified threshold N_S, that region is partitioned into 8 subregions. A subregion is marked as occupied if the number of points within that region exceed a specified number N_P. In other words, a subregion is considered open (i.e., obstacle free) if there are not at least N_P sensed 3-D points within that region. Suitable choices for the parameters N_P and N_P were investigated under different conditions.



(a) 3-D Adaptive Terrain Representation **(b) Image of Virtual Environment**

Figure 5: Terrain Representation of a Virtual Urban Environment using 3-D Adaptive Learning.

Figures 5 and 6 present examples of three-dimensional terrain representations of the scene depicted in Figure 2. These representations are in terms of occupied rectangular prisms at different resolution levels, with color variation to denote the maximum height of each occupied region. One can see that the urban canyon is represented at higher resolution and some of the adjacent buildings are represented by larger subregions. The terrain representation in Figures 5 and 6 differ in terms of the maximum allowable resolution employed. It should be emphasized that, as in the two-dimensional algorithm, the three-dimensional terrain implementation is recursive (locally modified as new point cloud data are sensed and processed) and suitable for real-time application.

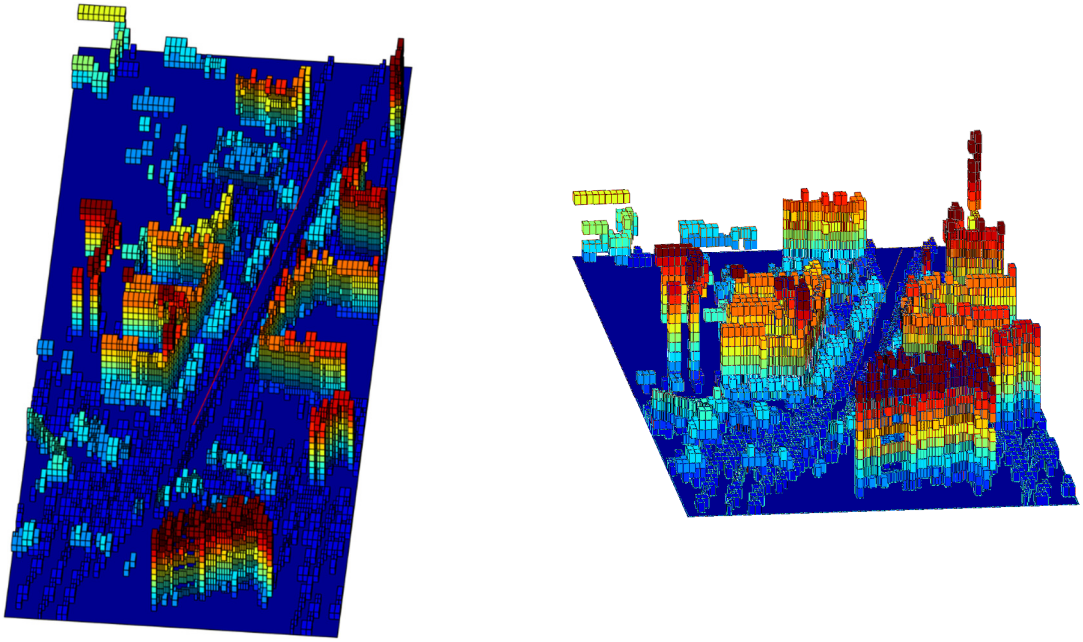


Figure 6: Terrain Representation using 3-D Adaptive Learning (two different views).

III. 3-D Terrain Results from Flight Data

3-D terrain representations were generated from sensor data acquired during UAV data collection flights through an obstacle environment designed to resemble an urban canyon. The UAV and environment are shown in Figure 7. An inertial 3-D point cloud was computed from the LIDAR data collected during the flight. The point cloud computation required using the measured position and attitude of the UAV, as well as the fixed orientation of the LIDAR with respect to the UAV (set to a 20° downward pitch angle), to map the LIDAR data into an inertial reference frame. A representative point cloud from one of the test flights is illustrated in Figure 8. For clarity of presentation, the figure only shows points within the obstacle environment that are at an altitude of 3 ft AGL or higher. Therefore, most ground points are not shown, and most of the points correspond to the various obstacles within the scene.

Figure 7: UAV and Obstacle Environment.

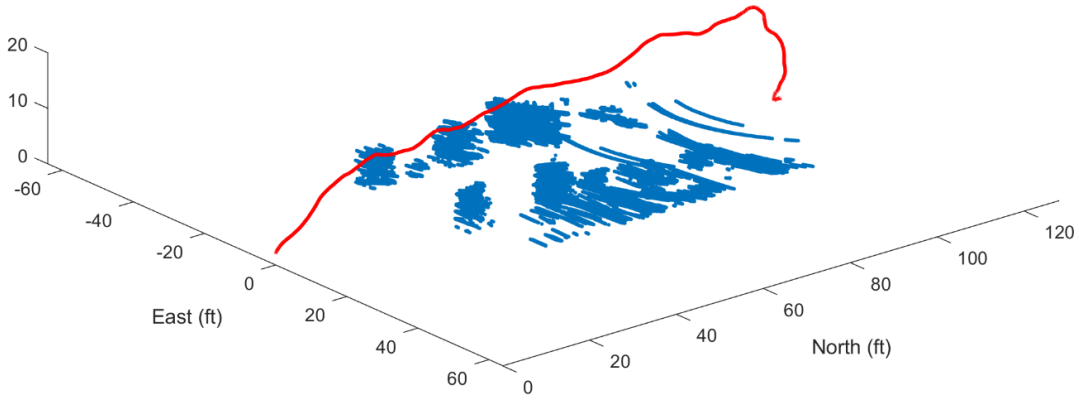


Figure 8: Inertial 3-D Point Cloud from Flight LIDAR Data.

Figures 9 – 12 depict 3-D terrain representations that were computed at different lateral resolution levels, ranging from Level 5 to Level 8. The resolution levels are dyadic, so Level 5 corresponds to dividing the lateral domain into a 32 x 32 array of lateral subregions, Level 6 corresponds to a 64 x 64 array, Level 7 corresponds to a 128 x 128 array, and Level 8 yields a 256 x 256 array. Note that the mapping is performed over a 128 x 128 ft^2 domain, so the physical sizes of each subregion in the Level 5 – 8 resolutions correspond to 4 x 4 ft^2, 2 x 2 ft^2, 1 x 1 ft^2, and 0.5 x 0.5 ft^2, respectively. In all cases, a fixed number of vertical regions was selected at a resolution of 1 ft in the vertical direction. The UAV flight path is shown in red in each figure, showing that the vehicle was flown through the urban canyon at an altitude that was slightly higher than the tallest obstacles (about 10 ft). As expected, the figures show that increased resolution provides more detail in the terrain representations.

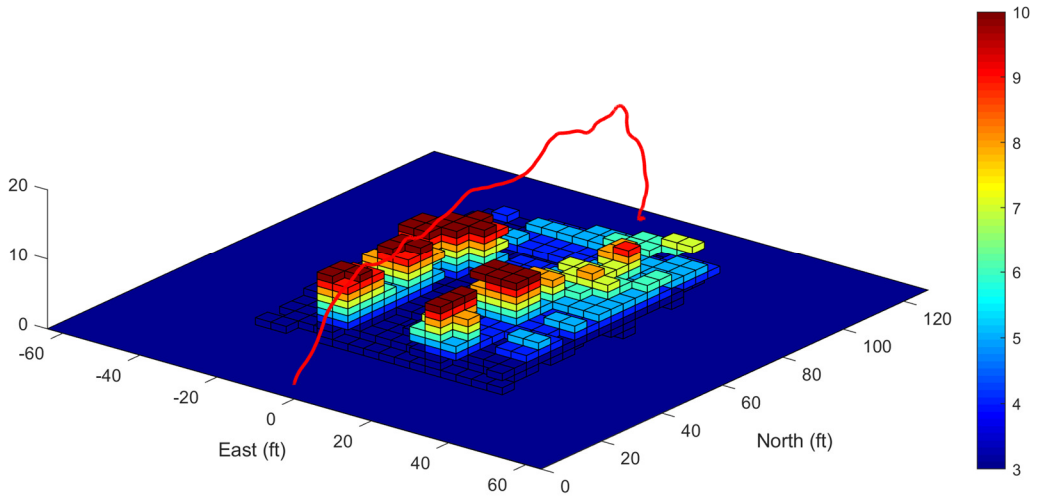


Figure 9: 3-D Terrain Representation – Level 5 Resolution.

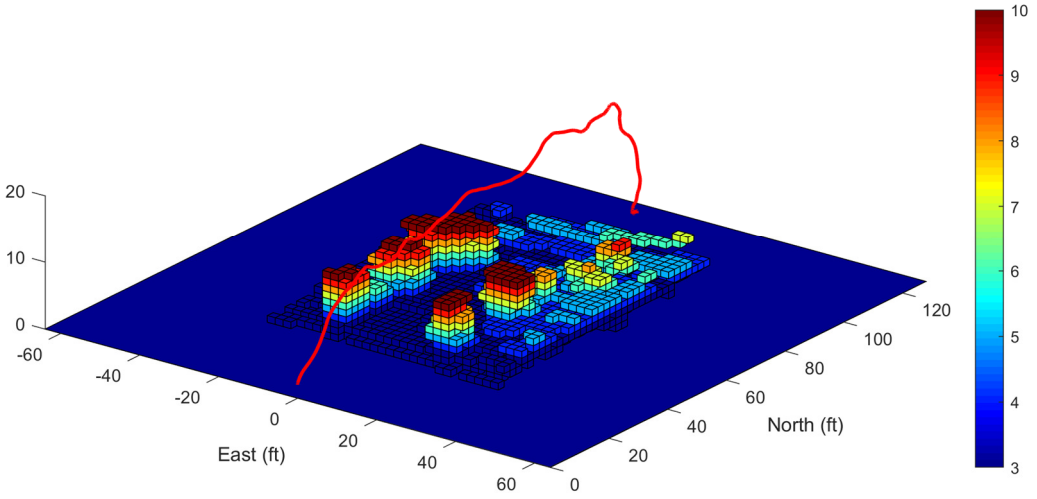


Figure 10: 3-D Terrain Representation – Level 6 Resolution.

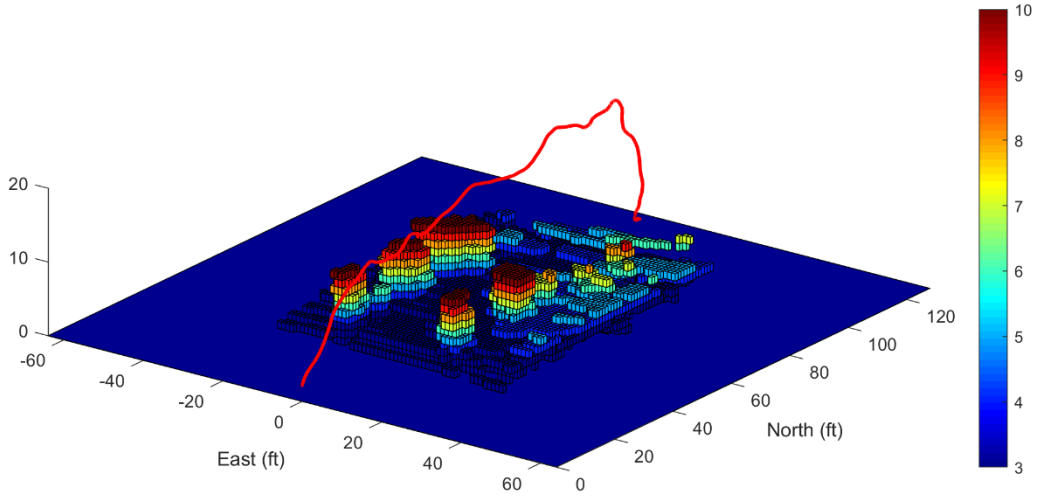


Figure 11: 3-D Terrain Representation – Level 7 Resolution.

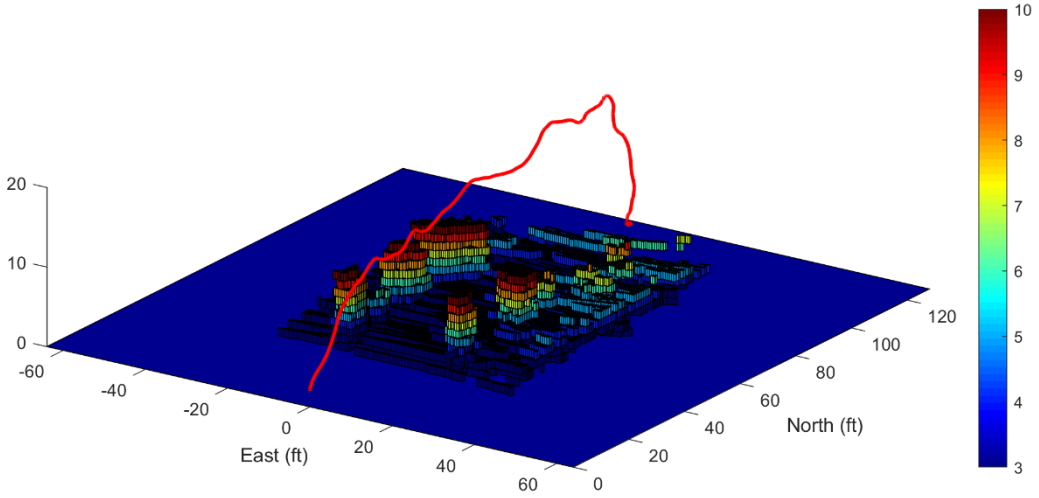


Figure 12: 3-D Terrain Representation – Level 8 Resolution.

Figure 13 provides an example of adaptive multiresolution. In this case, the lateral domain was originally divided into a 32 x 32 array of regions (i.e., Level 5 resolution) and 10 vertical regions. The lateral regions were then subdivided based on the point cloud data, with a maximum allowable resolution level of 8. The resulting terrain representation is in terms of 26,520 occupied cells. In general, this multiresolution example and the terrains generated at uniform resolution provide reasonable approximations of the environment from which the individual obstacles can be identified. It is also evident, however, that mainly due to errors in the 3-D point cloud, individual obstacles in the terrain map are rendered larger than their actual size (i.e., the point cloud is spread out). For example, the opening under the square arch is not identified in the terrain representations. This is most likely due to inaccuracies in the inertial point cloud resulting from errors in the Pixhawk navigation data and time synchronization errors between the Pixhawk and LIDAR data logging.

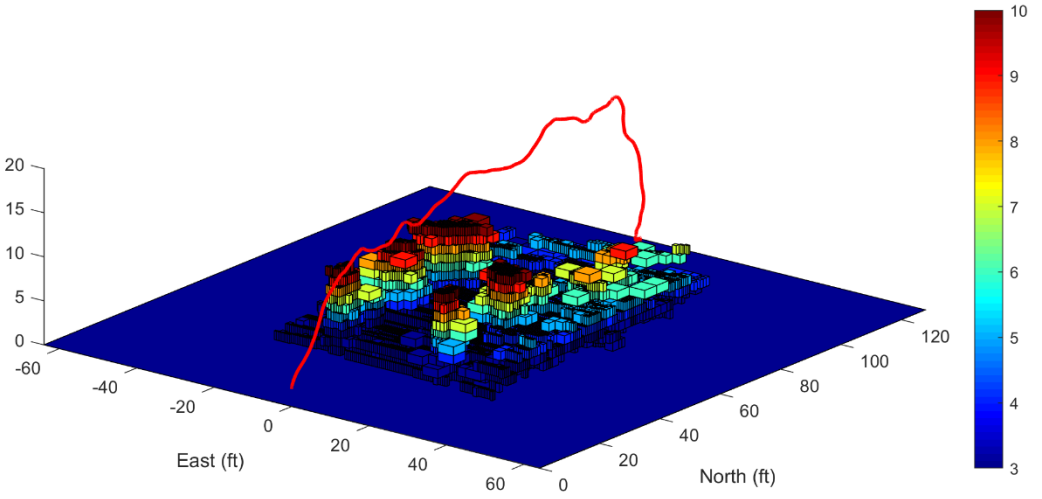


Figure 13: Multiresolution 3-D Terrain Representation.

As an example application of this work, the terrain representations were used to provide obstacle constraints to a path planning algorithm. The path planning algorithm required a simplified obstacle map that identifies the centroids and radii of distinct obstacles. To satisfy this requirement, a 2-D terrain map was first generated using the adaptive learning algorithm over a 2-D domain. A representative example terrain map from the flight data is shown in Figure 14. This representation was then thresholded by terrain height, which yields a binary representation that identifies areas that are occupied by obstacles, as shown in Figure 15a. A clustering algorithm was then applied to the binary map in order to group occupied cells in the binary map into distinct obstacles. Finally, the centroid of each obstacle cluster was computed and the radius of each obstacle is determined as the maximum distance from the cluster centroid to any point in the cluster, which is an intentionally conservative estimate. Figure 15b depicts the obstacle map corresponding to the terrain map shown in Figure 14.

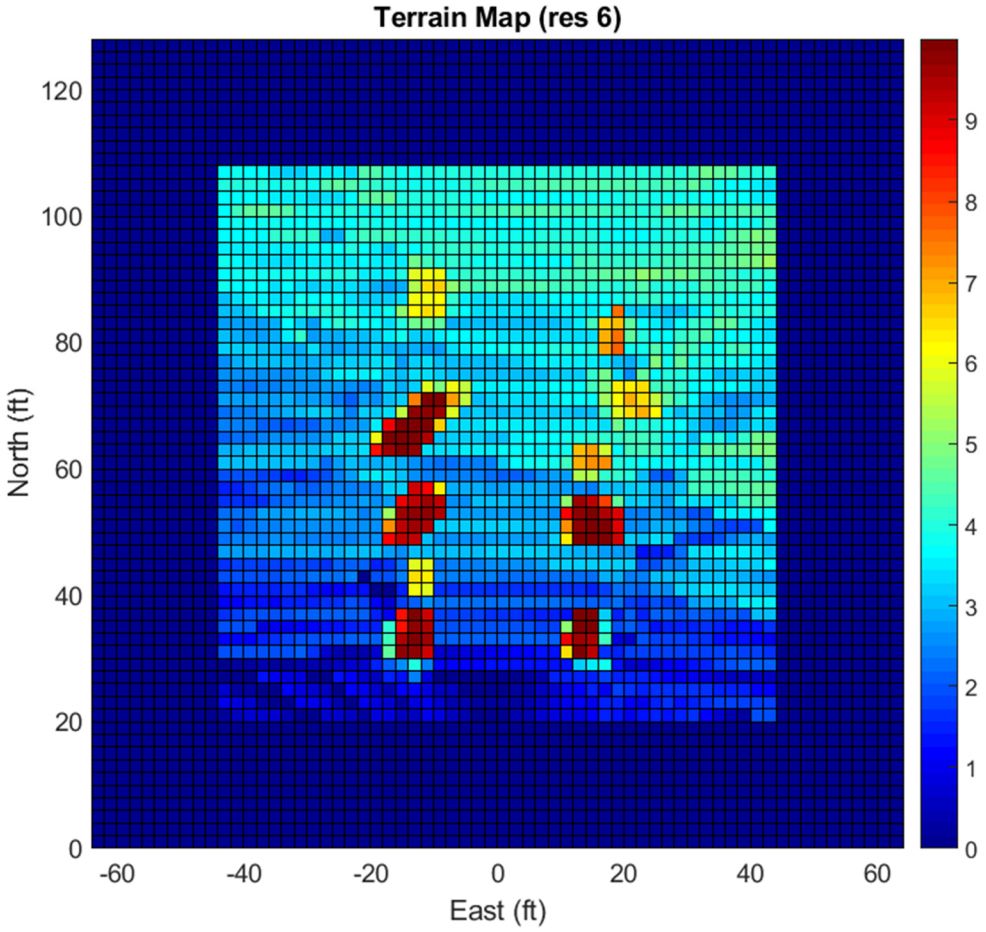
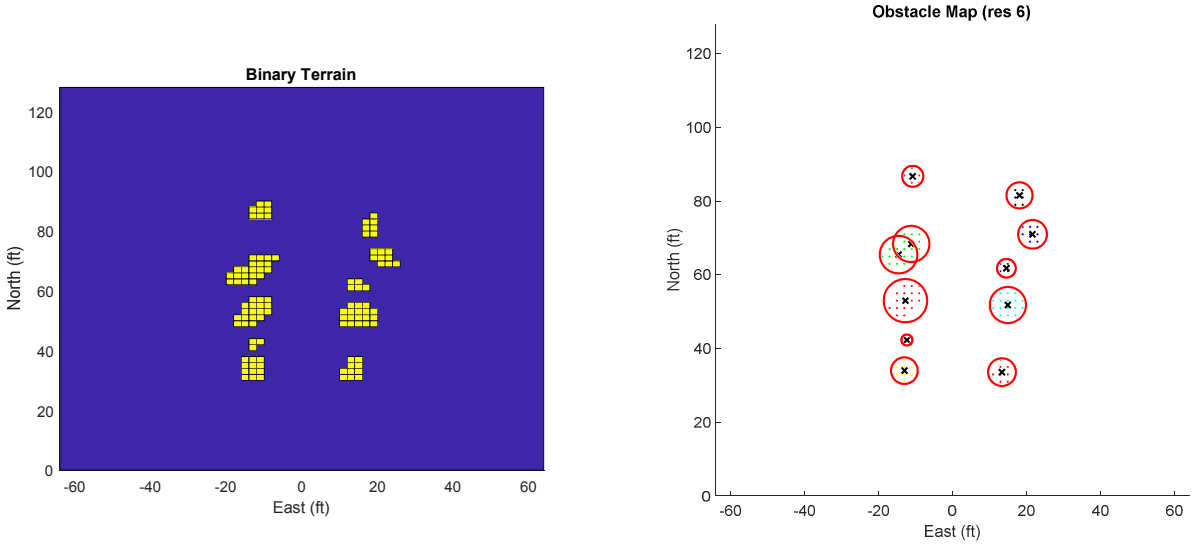


Figure 14: 2-D Terrain Map Generated from LIDAR Data.



(a) Binary Terrain Map **(b) Obstacle Map**

Figure 15: Obstacle Mapping for Path Planning Application

IV. Global-Local Orthogonal Mapping (GLO-Map)

Previous work in multiresolution functional representations of point cloud data [4 - 5] is currently being extended to consider 3-D terrain representations. The multiresolution properties of current algorithms have led to broad and useful approximation approaches in numerous problems. The Global-Local Orthogonal Mapping (GLO-Map) method [6] uses an orthogonal weighting function technique, as illustrated in Figure 16, that produces a global family of overlapping preliminary approximation methods whose centroids lie on the vertices (i.e., the data from the point cloud). Figure 17 shows an example of generating a point cloud from stereo vision images of a spacecraft, and the resulting functional representation from GLO-Map. The functional representations are being investigated due to their reduced requirement on memory, and rigorous comparison methods.

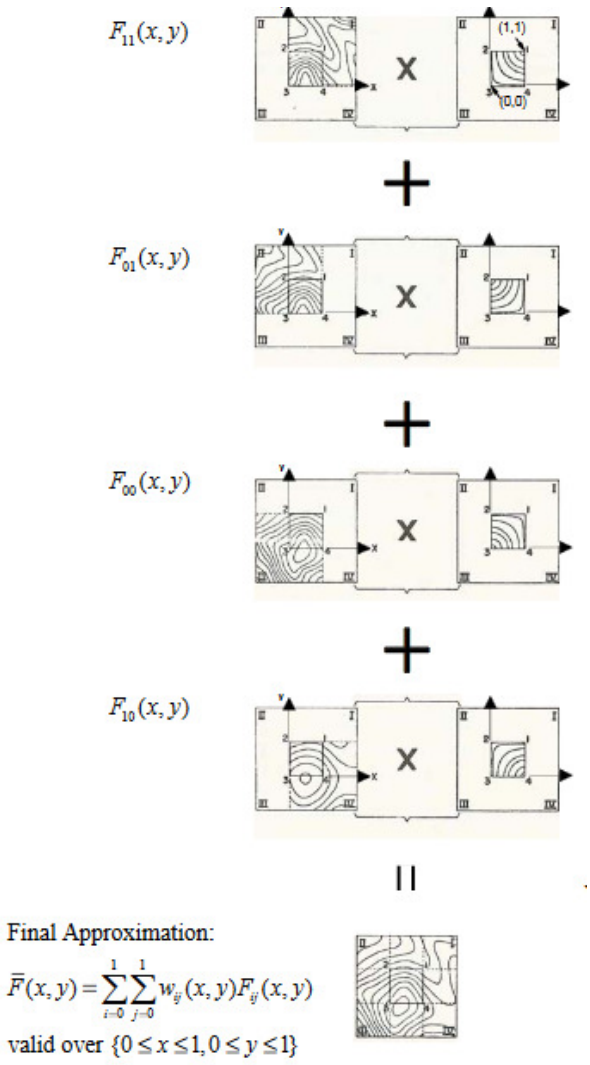
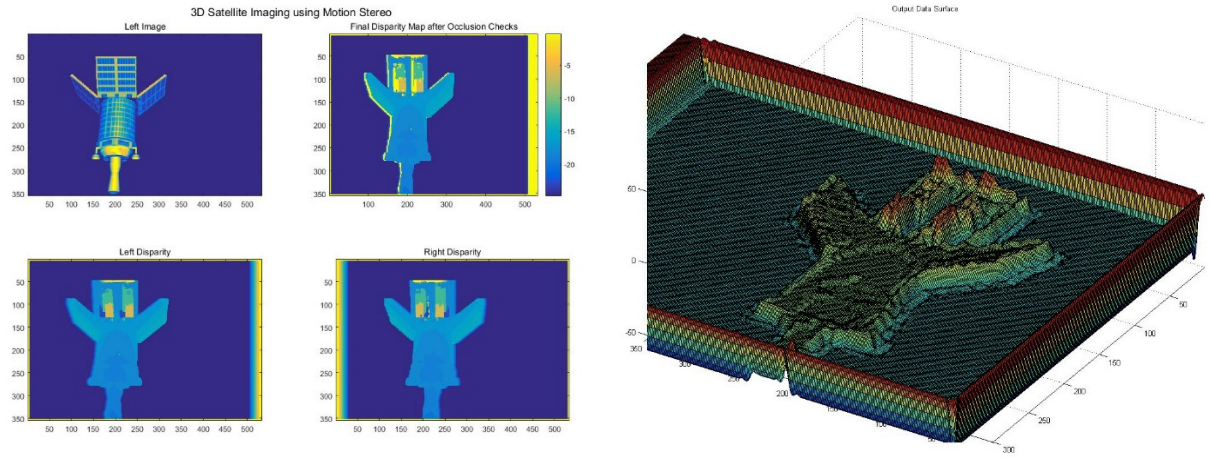


Figure 16: GLO-Map Algorithm with Weighted Orthogonal Basis Functions.



(a) Point Cloud **(b) Functional Representation**

Figure 17: Stereo Vision and Functional Representations of a Satellite.

Since the weighting function generates a global family of overlapping preliminary approximations, these naturally local models are defined independently of each other with the choice of any classical basis functions (using a priori data). An averaging method ensures that the final approximations are globally piecewise continuous. The global/local approach to mapping makes sense because sensors, in particular vision-based sensors, may fail over water or near mirror-like buildings. As such, the local approximations would be a much more attractive basis for local analysis, such as path planning.

Currently, a batch GLO-Map code is operational and has been tested against relevant data sets to quantify the efficiency and accuracy. Future work includes expansion to a sequential algorithm, which does not require re-running the code when new data is provided. The extension of the algorithm to non-functional spaces, such as under a bridge or overhang, is currently under investigation. One potential method is to mesh B-spline functions on top of the global/local map that can handle multi-valued functional representations

V. Conclusion

This paper has investigated the development of 3-D terrain mapping algorithms suitable for the representation of complex 3-D environments such as an urban canyon. Two classes of multiresolution-based algorithms were considered in this research: adaptive mathematical learning algorithms and global-orthogonal mapping (GLO-Map). Representative terrain mapping examples were presented using simulated LIDAR data from a virtual urban environment as well as processed 3-D LIDAR data collected from a UAV flying through a representative obstacle environment.

Acknowledgements

This material is partially based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D16PC00110.

References

[1] Binev, P., Cohen, A., Dahmen, W., DeVore, R., and Temlyakov, V., "Universal Algorithms for Learning Theory, Part I: Piecewise Constant Functions," *Journal of Machine Learning Research*, Vol. 6, September 2005, pp. 1297–1321.

[2] Prazenica, R.J., Kurdila, A.J., and Sharpley, R.C., "Receding Horizon Control for MAVs with Vision-Based State and Obstacle Estimation," invited paper, *AIAA Guidance, Navigation, and Control Conference*, AIAA 2007-6830, Hilton Head, SC, August 2007.

[3] Prazenica, R.J., Hielsberg, M., Sharpley, R.C., and Kurdila, A.J., "3-D Implicit Terrain Mapping and Path Planning for Autonomous MAV Flight in Urban Environments," *AIAA Guidance, Navigation, and Control Conference*, AIAA 2013-4792, Boston, MA, August 2013.

[4] Singla, Henderson, Junkins, and Hurtado, "A Robust Nonlinear System Identification Algorithm using Orthogonal Polynomial Network," *15th AAS/AIAA Space Flight Mechanics Conference*, Copper Mountain, Colorado, 23-27 January 2005.

[5] Junkins, Singla, Griffith, and Henderson, "Orthogonal Global/Local Approximation in N-Dimensions: Applications to Input/Output Approximation," *6th International Conference on Dynamics and Control of Systems and Structures in Space*, Cinque-Terre, Italy, 2004.

[6] Singla and Junkins, *Multi-Resolution Methods for Modeling and Control of Dynamical Systems*, CRC Press, 2008.