AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Information Systems-AIAA Infotech @ Aerospace

10.2514/6.2018-1236

Generating Multi-Purpose Rendered Environments from Unmanned Aerial System Data

Shida Xu*, Carter Kraus†, and Christopher W. Lum‡
Autonomous Flight Systems Laboratory
University of Washington, Seattle, WA, 98195, USA

I. Abstract

This paper proposes a novel approach to processing and utilizing aerial imagery and data for mapping. With the advancement of unmanned aerial systems (UAS), airborne data has gained considerable momentum in recent decades, revolutionizing methodologies of geographic research, law enforcement, agriculture, and mapping. Specifically, advances in consumer electronics significantly eased access to UAS and development in parallel computing has enabled machine learning on large data sets. This paper proposes an alternative method to the current practice of processing airborne data sets. Instead of obtaining aerial imagery of an object and merely presenting those objects as pixels on a map, the proposed method uses machine learning techniques to recognize the object, assign parameters to the object and render the object in a fashion that is most efficient and understandable to the end user. The object is no longer merely a set of pixels but a list of classified objects with associated parameters, an abstraction analogous to cognitive processes. This approach allows a single data set to be used in multiple fashions. For example, the data can be used to simulate changes in the environment or render the environment in different scenarios. This paper explores the methodology for collecting airborne data sets for this process and presents several cases of this using flight test data.

Nomenclature

AFSL	Autonomous Flight Systems Laboratory
CNN	Convoluted Neural Network
DEM	Digital Elevation Model
DSM	Digital Surface Model
GCS	Ground Control System
GIS	Geographic Information System
LiDAR	Light Detection and Ranging
NDVI	Normalized Difference Vegetation Index
PCL	Point Cloud Library
SDK	Software Development Kit
UA/UAS	Unmanned Aircraft/Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UW	University of Washington
UWAA	University of Washington Department of Aeronautics and Astronautics

*Research Assistant, William E. Boeing Department of Aeronautics and Astronautics, University of Washington; Guggenheim Hall Room 211, Box 352400, Seattle, WA 98195-2400, AIAA member.
†Research Assistant, William E. Boeing Department of Aeronautics and Astronautics, University of Washington; Guggenheim Hall Room 211, Box 352400, Seattle, WA 98195-2400, AIAA member.
‡Research Assistant Professor, William E. Boeing Department of Aeronautics and Astronautics, University of Washington; Guggenheim Hall Room 211, Box 352400, Seattle, WA 98195-2400, AIAA member.

II. Introduction

A. Problem Statement and Literature Review

Recent developments in consumer electronics have made UAS more versatile and accessible. UAS are widely used in applications such as search and rescue,[1,2] fire prevention and control,[3] safety and security,[4] archaeology,[5] meteorology,[6] precision agriculture,[7] and many other. Aerial imagery is one of the most popular UAS applications. Aerial mapping and surveying can be performed using sensors such as electro-optical, LiDAR, multi-spectral imaging and other methods of remote sensing. In this paper, these are collectively referred to as 'photo telemetry' techniques as they interpret incoming sources of light at varying wavelengths to remotely sense the area of interest. These techniques significantly ease jobs such as mapping, construction site selection, town structure planning, and progress monitoring of civil projects.[8] Specifically, UAV-based aerial photo telemetry is not only increasing close-range remote sensing applications, but it is also increasingly used as an alternative to conventional aerial photogrammetry.[9]

Advances in computer science have enabled artificial intelligence algorithms that are efficient in classifying objects and finding patterns. Implemented as a convoluted neural network (CNN), artificial intelligence mimics the human learning process by extracting features and patterns from a set of training data. It is widely used in applications such as photo recognition, classification, speech recognition.[10–12] With a CNN, given an abundant amount of training data and proper training and practice, an artificial intelligence algorithm has the potential to replicate the expertise of a person in a particular application. Some of the currently demonstrated abilities include playing chess[13] and recognizing objects in photos.[14]

Maps are used to convey and share geographically related information efficiently. Modern maps are divided into two categories, abstract and picturesque. Abstract maps use abstract symbols and color regimes to represent different objects and geographical features. For example, an abstract map could use hollow lines to designate roads, a book symbol to represent a school, the color blue to represent bodies of water, and contour lines to constitute a change in elevation. One needs specialized training to interpret abstract maps because the symbols are arbitrary and specialized knowledge is required to relate abstract symbols or patterns with their associated objects and features. Picturesque maps are more intuitive to interpret and are usually obtained through satellite imagery or aerial photography. Although high-quality aerial images can also be acquired using conventional platforms, such as satellites and aircraft, their temporal resolution is limited by the availability of aircraft platforms and orbit characteristics of satellites. Additionally, cloud coverage and air quality affect image acquisition. Therefore, this method is expensive and is limited in the environmental conditions under which operations could be conducted. Street view maps can serve as detailed supplements to satellite maps, but street view maps are often cumbersome to obtain as they can be limited by road access.

This project explores using UAS and artificial intelligence to produce abstract maps that are intuitive and efficient for a user to interpret. This is done in a way that mimics the human cognitive process; the environment is represented as a list of objects with meaning. In cognitive psychology, 'spatial ability' is used to refer to the concept of understanding, reasoning, and remembering the spatial relations among objects and space.[15] Spatial ability is essential to a persons daily life as it enables a person to navigate, perceive distance, and sense motion about themselves. Spatial ability is composed of four components: spatial perception, mental rotation, spatial visualization, and mental folding.[16–18] Spatial perception is defined as the ability to perceive spatial relationships among objects and features on oneself while rejecting interference.[16] It enables a person to perceive spatial information from the environment, such as shapes, features, texture, objects, and motion.[17] Some scholars argue that spatial perception involves attributing meaning to objects, implying recognizing individual objects and features is vital to achieving spatial perception.[18] Mental rotation describes the mental ability to quickly and accurately rotate objects without altering their features. For instance, a person could recognize both an upright and inverted letter, or recognize a house both standing in front of it and flying directly above it; change in perspective would not impair a person's ability to identify objective and features. Spatial visualization is a mental process during which a mental image is created to represent visual appearance of objects with respected locations and movements in an imaginary space. For instance, an engineer would be able to mentally animate how a machine he or she is designing is supposed to work.

The proposed approach is, in some aspects, matching spatial ability in its features. It would measure and perceive the environment as a list of classified objects with meaning. Then, the list would be generated from a three-dimensional reconstruction of the surveyed environment, thus enabling object classification from any field of view. This would make any analytic change analysis to or within the environment possible.

Recently, there has been significant interest in generating photo-realistic and other highly accurate maps from UAS. While these maps are useful for some applications, such as precision agriculture,[19] geological surveying,[20] or searching for a target,[2,21] they often contain more information than is needed for the task at hand. For example, city planners may only be interested in the location of roads or pavement on a map. Firefighters may just care about the location of fires or fuel sources in an environment.[3] Researchers of animal migration patterns may only be interested in the position of the animal flock, not in other irrevelent geographical features at this place. In these types of purposes, instead of generating a photo-realistic map of an environment, it is sufficient, and even desirable, to produce a simplified representation of the world with critical objects identified and labeled. This has the potential to increase situational awareness of users of this system by avoiding information overload and allowing the user to focus on relevant objects in the environment.[22,23] This method also has advantages over picturesque maps as they are more economical and less restricted to obtain, easier to manipulate, capable of producing a more detailed representation of the surveyed area, and can display the surveyed area under different conditions such as differentiated lighting or weather conditions.

B. Work-flow

The mapping method demonstrated in this paper generates labeled, rendered environments from data collected by UAS in four steps: data acquisition using a UAS, object identification and classification, map presentation with an objective-specific rendering scheme, and calibration/validation against maps obtained by other methods. Data can be collected from a variety of airborne sensors such as a camera (optical, multispectral, or thermal), LiDAR, and sonar. These raw sensor measurements are processed using commercially available techniques[24] to generate intermediate data sets such as digital elevation models (DEMs), normalized difference vegetation index (NDVI) map,[25] and point clouds. Lastly, due to inaccuracies that may occur during data acquisition, environmental variability, and interference, the cluster of objects would be calibrated and validated against existing accurate geographical records and ground truth data such as existing geographic information system (GIS) data, construction records, and environmental measurements. A process map of the work flow is shown as Figure 1.

C. Potential Applications and Example Scenarios

The proposed method produces a list of classified objects to represent a geographical area. This would enable three abilities: the ability to generate a map for different interfaces from a single data set, monitor change in environment with ease, and simulate changes to the environment.

For instance, a single data set could be used to generate a map for an arborist to highlight the positions and spread pattern of trees above a certain height, whereas for a farmer a map could be created that determines if a particular patch of crop is growing less than others.

Changes in the same environment could be more easily monitored because an automatic comparison between two scans of the same area is merely an arithmetic operation of the classified objects lists; the growth rate of individual trees within the forest could be conducted in this way.

The list of objects could be passed on to a simulated environment to allow the model or map to be rendered in different conditions, such as simulating forest fire progression at a specific forest. A real-estate agent could use a UAS to scan a property and its surrounding environment and showcase the property to potential clients with the surrounding environment in different lighting and season.

III. Data Acquisition

This section describes obtaining data from a UAS for use in this particular mapping process.

A. Aerial Data Acquisition

Mapping operations performed by UAS are usually conducted by flying the aircraft over an intended survey area while carrying various sensors. These sensors are used to take a series of prescribed measurements at specific locations. Such measurements often embed meta-data, such as the UAV's GPS position and sensor configuration. Flight paths can be pre-determined and flown autonomously, or the UAV can be controlled by a remote pilot. The example data set demonstrated in this paper was obtained using a DJI Phantom 3

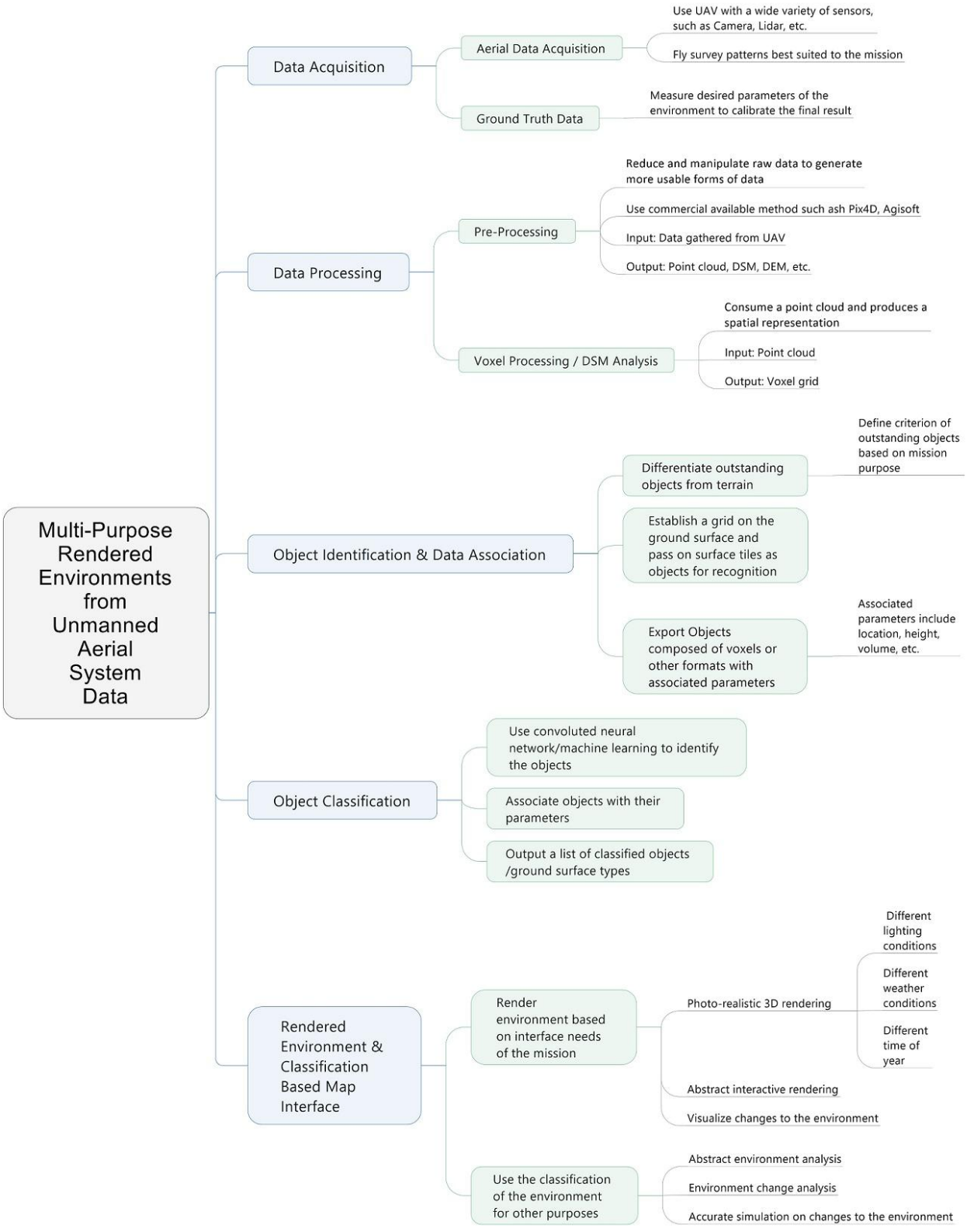
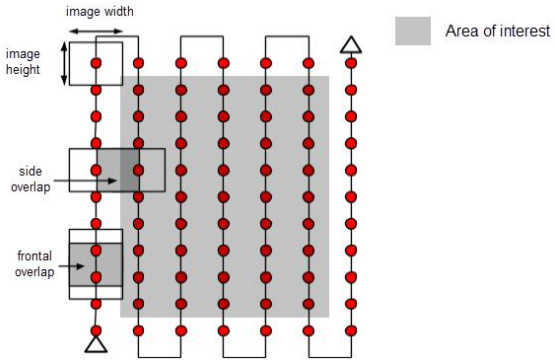


Figure 1. Process map of the work-flow.

Pro UAV with its standard camera configuration. Pix4D software, specifically Pix4D Capture, was used to design the UAV's survey area and autonomous flight path. Such software is compatible with commercially available hardware, such as a tablet. Thus, the process is relatively inexpensive and easily accessible. A sample flight path of this type, designed in Pix4D Capture, is shown below.

(a) Planning data acquisition using the Pix4DCapture app with a double grid pattern.



(b) Sample flight path with overlap parameters shown.

Figure 2. UAV flight plan interface and sample flight path.

(a) DJI Phantom 3 Pro, a consumer grade UAV.

(b) Aerial images acquired during a mission.

Figure 3. Sample UAS equipment and associated acquired aerial images.

B. Environment Sample Collection

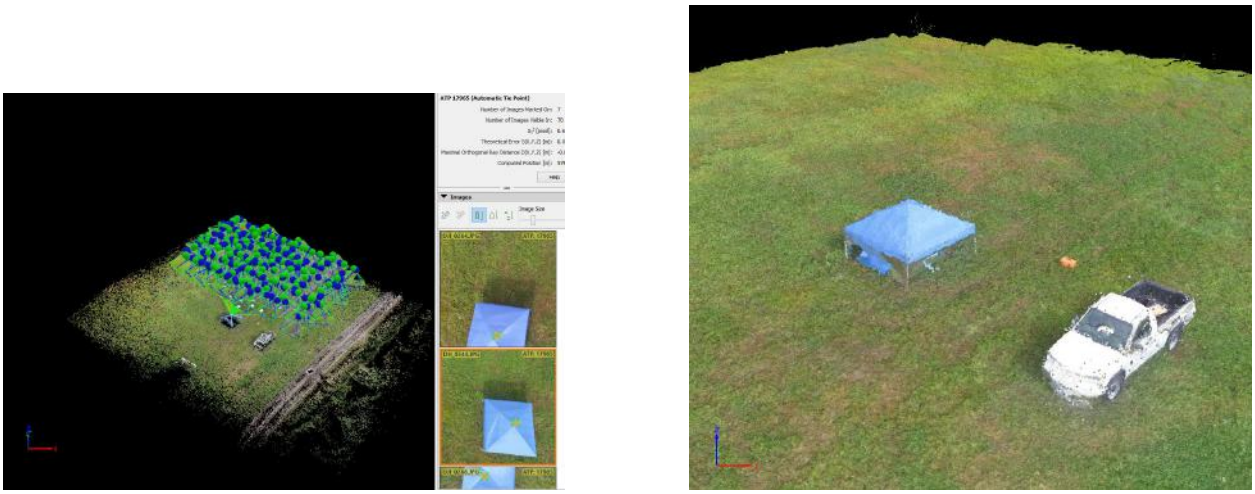
Depending on the application, an operator may require a different level of detail or sets of parameters from the final classification output. To satisfy case-specific demands and improve the accuracy of object classification, samples should be collected from the surveyed environment for application-specific needs. These samples may be close-up pictures of structures or surfaces, samples of vegetation, and weather conditions, etc. For instance, if a researcher wishes to determine the moisture content across a specific grass field, he or she could use the proposed approach to classify different types of grass, based on color, height, and relative location to a reference point. Based on a classified map, the researcher would easily be able to identify points of interest, collect samples for laboratory analysis, and apply the result to the entire field where grass conditions are classified as similar to samples collected.

The examples below demonstrate that the proposed approach is adaptive, as its capability could be extended by additional step to suit different needs. Figure 4 shows a sample of ground truth data collected for the point cloud depicted in Figure 6.

Figure 4. Ground truth data collected to provide more detail for the same environment which produced aerial images shown in Figure 2(b).

IV. Data Processing

To reduce aerial imagery data to a set of classified objects, one should establish a simplified volumetric profile spanning the survey space. Commercially available software, such as Pix4D or AgiSoft PhotoScan Pro can be used to achieve this task. The survey area is reconstructed by stitching together the aerial images acquired during flight missions by identifying matching key points in photos and overlapping the photos at these points. Photos from the DJI Phantom 3 Pro UAV are tagged with geographic information such as position and altitude, which is used for photo alignment. Intermediate steps in data processing, such as adding ground control points and optimizing camera settings, can be taken to improve the accuracy of the desired output, such as a DSM or point cloud. A point cloud could also be computed from other data sources, including data output from NDVI cameras, inferred cameras, and LiDAR. A sample point cloud is shown in Figure 6. To fill gaps in the point cloud, the point cloud is processed into a triangular surface mesh, and voids in the voxel grid are filled. An example of photo alignment and key point identification is shown in Figure 5(a) and a sample surface mesh is presented in Figure 5(b).



(a) Photo alignment process with camera positions.

(b) Resulting surface mesh from data processing.

Figure 5. Photo alignment process with camera positions and surface mesh created from the imagery data set.

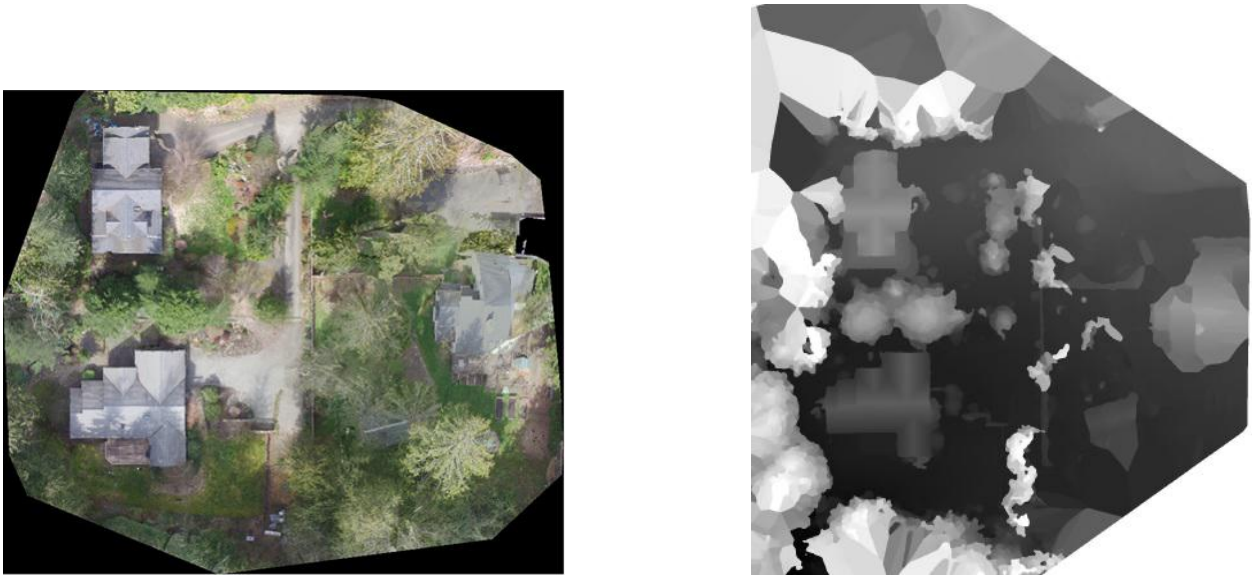
Figure 6. Sample of a point cloud.

A. Digital Surface Model Processing

A point cloud or surface mesh can be used to construct orthomosaics or digital surface models (DSMs), both of which are helpful to generate a deeper understanding of the environment. An orthomosaic is a composite image made up of the individual images collected by a UAV, aircraft, or satellite. The orthomosaic is geometrically corrected so that the distances between objects are true to scale. Thus, the orthomosaic can be useful in obtaining accurate dimensions of objects or distances between objects for the purpose of rendering the simulated environment. Additionally, color information can be obtained from an orthomosaic, which can be used to classify various objects. Figure 7(a) shows a sample orthomoaic generated from the surface mesh shown in Figure 8(a). A DSM portrays the relative elevation of a given environment. Relative elevations can be determined from aerial photogrammetry or the use of a LiDAR sensor.[26] A color code is used to represent the elevation of certain features on a DSM. For example, blue could represent low elevation while red represents a higher elevation. Grayscale DSMs are also common. In a grayscale DSM, the brighter the point is, the higher elevation it represents. A sample grayscale DSM is shown in Figure 7(b). Elevation data extracted from DSMs can help determine features from one another, determine growth rates, and be used to calculate volume. For the work outlined in the paper, Pix4D was used to create orthomosaic and DSM. Matlab was used to read and view the images for data analysis and manipulation.

B. Volume Pixel Based Processing

The point cloud, orthomosaic, and DSM can be used to generate a voxel grid of the environment. A voxel is a cube with associated color, position, and other properties.[27] Figure 8(b) shows a voxel representation of a car. Using voxels for further processing has distinct advantages in data size, processing time, noise suppression, and easier volumetric manipulations. Using a voxel grid for spatial representation can dramatically reduce data size, when compared to representing the same space with points, since one voxel can represent numerous points. Depending on accuracy requirement, a voxel grid is a down-sampled version of a point cloud. The point cloud could contain inaccuracies based on how the point cloud is acquired. For example, using a Phantom 3 Pro with standard camera attachment and the Pix4D software for processing, the raw data are photos taken by the UAV, which are assumed to have ± 5 meters of accuracy, to account for GPS inaccuracy, for photo geotaging. Pix4D would proceed to align each photo using variuos techniques to establish a point cloud in its own reference frame. The position of each point within the point cloud is calculated to a accuracy of 10^{-7} meter in all lateral, longitudinal, and vertical axis. It is reasonably deducible that noise should become a reasonable concern when one attempts to obtain results from the point cloud. Although

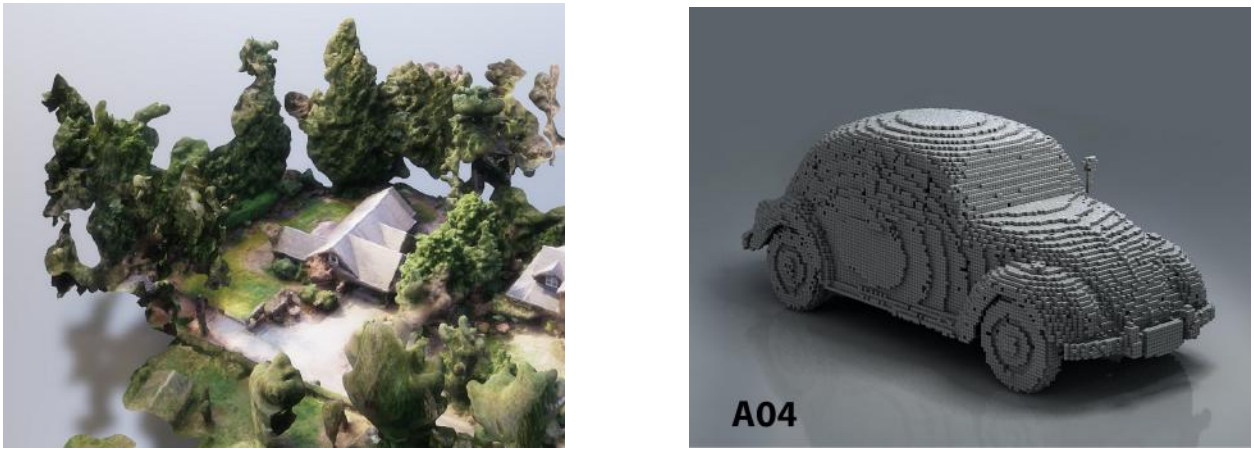


(a) Orthomosaic obtained by Pix4D.

(b) Resulting DSM generated from Pix4D.

Figure 7. Pix4D processed raw data in the form of point cloud and DSM.

there exist software that relies solely on photo-telemetry techniques to obtain the point cloud, such as AgiSoft, the resulting point cloud is still subject to noise and inaccuracies. Previous AFSL experiments has found that using the Phantom 3 Pro with its standard camera attachment to survey the same field with medium-density vegetation (grass) at almost the same time, and process raw data into point clouds using either Pix4D (photo geotag dependent) and AgiSoft (photo geotag independent), a difference in highest vertical point at the same latitude and longitude location of each point cloud varied between -0.01 and 0.03 meter. A number of factors, including GPS accuracy, wind, lighting condition, and camera distortion have been identified as possible attributes to the noise and inaccuracies. By downsampling the point cloud using a voxel grid with 0.02 meter lattice, the inaccuracies and noises within the point cloud would be largely eliminated, resulting in basically the same result across different scans of the same space. At the same time, because a voxel is a concentrated representation of all points within the volume of the voxel, the mesh gemoterty would be greatly simplified. As a result, representing the space within a voxel grid would greatly reduce the size of data set compared to using a point cloud. A surface mesh created from the point cloud shown in Figure 6 is shown in Figure 8(a). A model car made of voxels is shown in Figure 8(b).



(a) Surface mesh visualized in Windows 3D Viwer.

(b) Voxel representation of a car.[28]

Figure 8. Surface mesh created from point cloud and voxel representation of a car.

Compared to using polygons, using a voxel grid adds geometry and the capability of manipulating the spatial profile of the space. Both point clouds and polygons are superficial representations of a spatial object. As a result, the content below the surface of an object is usually not defined. Hence see-through projections would be produced by certain camera angles. In addition, any volumetric manipulations, or volume-based simulations such as lighting and fluid, are presumably either impossible or immensely complex to accomplish. For instance, if one wishes to drill a hole on a surface, one would have to develop sophisticated means to construct the additional surface added to the object by this drilling operation. A sample of erroneous rendering when placing the camera inside a structure defined by polygon surfaces is shown in Figure 9. This image is obtained by placing the camera inside the walls of a desert castle type structure defined by polygon surfaces in Source engine developed by Valve Inc. Although some structural elements are discernible, the rendering does not resemble a castle structure, thus the rendering is erroneous and ineffective. If one were to operate the same on a voxel grid, it would be a simple removal of affected voxels. For instance, in a Lego model, when one surface piece is removed, the space below the surface is not hollow unless by design. The appearance of the model does not require reconstructing of the surfaces to compensate to the removed surface piece because the internal structure naturally forms the new appearance.

The voxel grid could be obtained by commercial software development kits (SDK), such as Point Cloud Library (PCL).[29] A comparison of the point cloud obtained by Pix4D and the associated voxel grid with lateral length of 0.15 meter computed by PCL is presented in Figure 10.

Figure 9. Erroneous rendering when camera placed beneath polygon surface.

V. Data Analysis

Once the data is acquired and processed using the aforementioned techniques, it is analyzed to distill useful information from the data set. A portion of the data analysis necessary for object classification was done using Matlab. DSMs can be read in Matlab using built-in functions that are a part of the Matlab mapping toolbox. Each pixel of a DSM has a corresponding elevation value, and a matrix of these elevation values can be obtained as an output of one of these Matlab functions, namely 'geotiffread'. In addition to an elevation matrix, the function also outputs a reference object. This object contains valuable geographic information about the DSM file. Most importantly, the object provides the equivalent real-world size of a single pixel in the DSM. By combining the given pixel area with elevation data, the volume and area of specific objects can be calculated using simple multiplication. Custom Matlab functions were used to calculate the area and volume of objects in generated DSMs based on user input. The user is directed to select the outline of an object and its area and volume will be obtained. A sample of the selection process is shown in Figure 11(a). The area and volume of an object are unique properties that can be used to distinguish itself. For example, a tree has a different area and volume than a house. Thus, area and volume

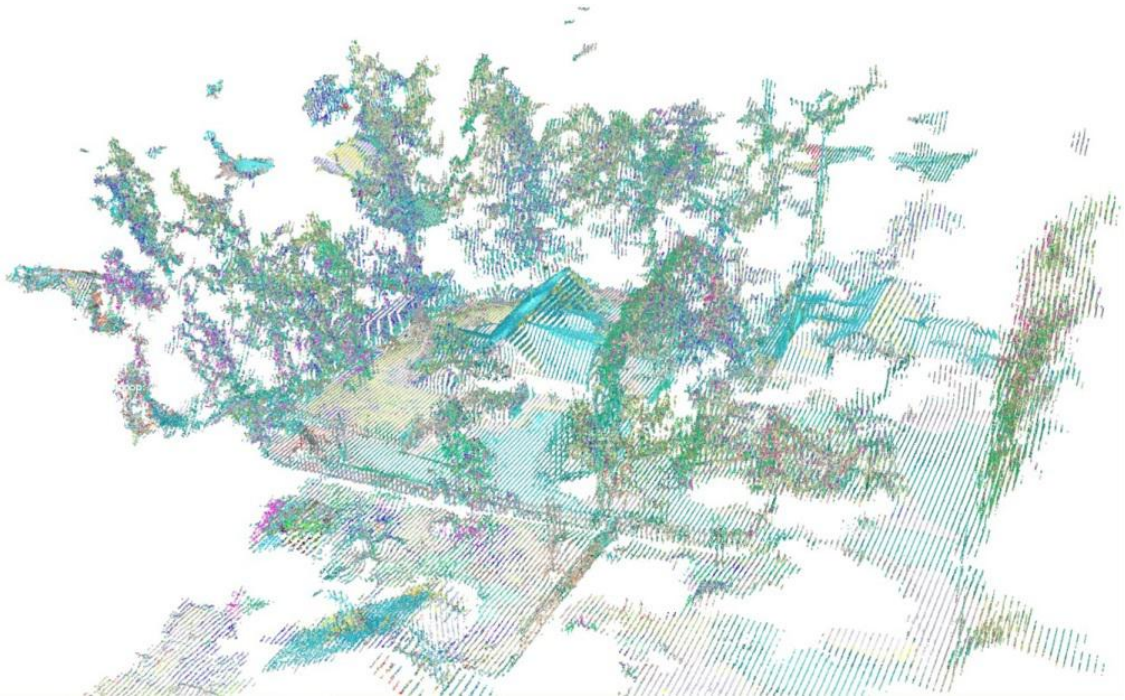
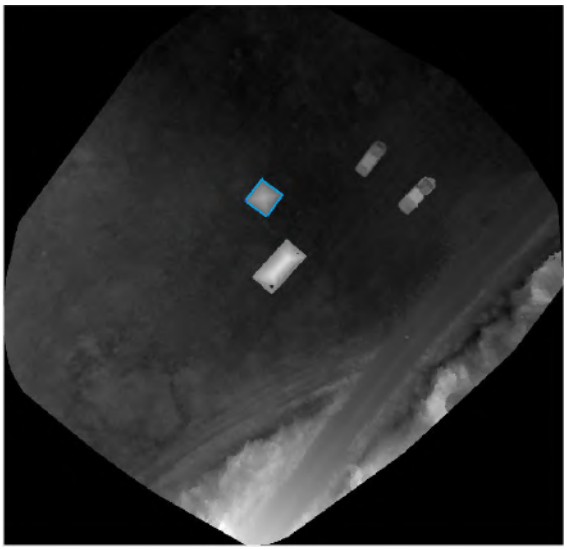


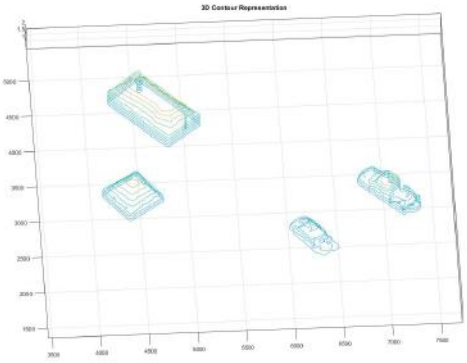
Figure 10. Voxel grid of the point cloud visualized in PCL cloud viewer.

can be used as inputs to object classifiers.

In addition to computing volumes, elevation changes can be used to identify different surfaces, such as differentiating between a road and the grass next to it.



(a) Volume calculation of an object in a DSM, the blue contour indicates user selected area the volume above which the user wishes to calculate.



(b) Contour visualization of objects, cross section of objects at different heights.

Figure 11. Selection of object in DSM from which volume and area measurements can be made and a contour visualization of a sample data set.

The first step in constructing a list of objects in the environment is to segment the environment into multiple spatial profiles representing individual objects so that each can be individually classified. Depending on data set properties there exists many means to accomplish this goal, two of which are presented below.

The base plane of the volume should be segmented into a grid. If one has obtained one voxel grid of a volume in which a ground datum could be identified, it would be reasonable to conclude that any cluster of voxel that appears obtrusive (having a significant height difference comparing to the reference datum and neighboring tiles) should be regarded as an independent object. Surface tiles within the grid which are not overlapped by an object should be passed on to the classification algorithm as a surface tile, and be classified as a surface type, such as gravel, vegetation, pavement, etc. This method is generally valid for mapping applications. The reference datum elevation is selected by the user according to a certain criteria, such as elevation of a user defined point or the most common elevation in the data set. Obtrusive objects, those with a higher elevation than ground, are therefore clearly visible and could be easily identified and segmented.

A contour profile of each object is obtained to classify the object. The contour profile is a set of cross section of an object obtained at different heights. A representation of the contour profiles of objects are shown in Figure 11(b).

VI. Object Classification

There exist multiple methods for object classification. Choosing an appropriate classification system tends to be heavily dependent on the particular mission. This section will present two general approaches that the authors deem most suitable for the studied UAV applications: correspondence grouping based object recognition and machine learning based object classification. This section is regarded as a separate section from data processing to highlight the difference between the paper's proposed mapping approach and conventional mapping approach. The previous data analysis section describes a process similar to conventional practice, as it focuses on calculations that are made directly from the data set. This section focuses on inferences made in those calculations.

A. Three-Dimensional Object Recognition Based on Correspondence Grouping

This method uses identified key points and associated features thereof to recognize objects. Specifics about the algorithm is extensively discussed by PCL documentation.[30] The processing pipeline and expected result are shown below.

While this method has shown desirable properties in tracking changes between scans, it imposes certain restrictions, as discussed in the previous section. In UAV applications, the data set may not always be complete, as shown in Figure 10. In addition, due to camera distortion, GPS inaccuracy, lighting variation, wind variation, changes in the survey volume during a scan, and other common sources of inaccuracies and error during UAV operations, changes between scans may not always be rigid, and noise level between data sets is usually not negligible unless substantial calibration efforts are applied. Also, it is not always possible, especially in complex environments or exploratory missions, to obtain a spatial profile of each expected object for a given operation. Hence the accuracy of the object recognition effort would be compromised because the hypothesis verification step of the algorithm could not be performed.

B. Object Classification Based on Machine Learning

Machine learning has been used successfully to classify data quickly and accurately. Therefore, we propose to use machine learning to efficiently classify large quantities of volumetric profiles of objects obtained through the data analysis step by considering the features of these object representations. One could offer two possible solutions. For common objects, classification could be completed by generating projections or images from the three-dimensional model of the object at strategically placed camera angles, and subsequently pass the pictures onto existing commercial image identification services, which are usually based on the convoluted neural network. For specific mission objectives, one could also develop custom machine learning algorithms to take advantage of particular features of the survey environment, and to maximize processing efficiency by tailoring classification to the most emphasized requirements of the mission objective.

C. Commercial Image Identification Service

Objects of interest for rendering can be segmented from the point cloud, DSM, or orthomosaic and made into an isolated object. These extracted objects can be passed through an image identification service to check against the output obtained from custom classifiers. Commercial photo identification engines, many of

which are based on a convoluted neural network and an abundance of training data, allow users to identify the primary objects within an image. By placing cameras at different locations around the volumetric profile of an object and using the commercial photo identification engines, one can classify the object within the images to a reasonable degree of certainty.

While this method is universal, fast and low in cost, it is restricted in control over classifiers and projection emphasis due to a lack of terminal user control over classification mapping, projection layer setup, and training data. For instance, a commercial photo identification service may place more emphasis on indices layout then the color pattern, whereas the mission objective requires differentiation of color instead of indices. Since the commercial service essentially operates as a whole, it is limited in its utility in some cases.

Figure 12 depicts an example of using Google Cloud Vision, a commercial image analysis API, to classify an object rendered with surface mesh at random camera angles. The dataset used to generate the photo of the car is acquired in real-world conditions further discussed in section IX, Experimentation, and Results. It is clear that the photo analysis API indicates that, with a reasonable degree of certainty, that the object depicted is a car.

Figure 12. Using CNN based commercial photo recognition API to classify objects rendered with point cloud based surface mesh.

D. Custom Built Machine Learning Algorithm for General Mapping Missions

At the inaugural stage of this process, there are distinct challenges to using deep learning for object classification. First, the object is represented as a three-dimensional volumetric profile, and little to noncommercial solution or experience for this kind of data exist as of this moment. Second, very few training data exists as of this moment, and gathering a significant amount of data for classification purposes for each mission during this phase is highly uneconomical and in some ways defeats the purpose of this approach.

To construct a custom machine learning algorithm, one could use commercial SDK, such as TensorFlow, as a template. This approach may be restricted in applicability as not all UAV mapping projects could guarantee the budget or practicability to gather enough training data to train a commercial CNN. Alternatively, for a project restricted in the availability of training data or provided with specialize emphasis, one could customarily construct a machine learning algorithm, as demonstrated in Figure 13.

The researchers involved with this paper had attempted to construct a machine learning algorithm which would require only a limited amount of training data, given specific restrictions on experimental environments.

The input of the algorithm is a cluster of voxels identified as objects received from data analysis phase with associated parameters, such as a cluster of voxels that represent a house, and its associated location

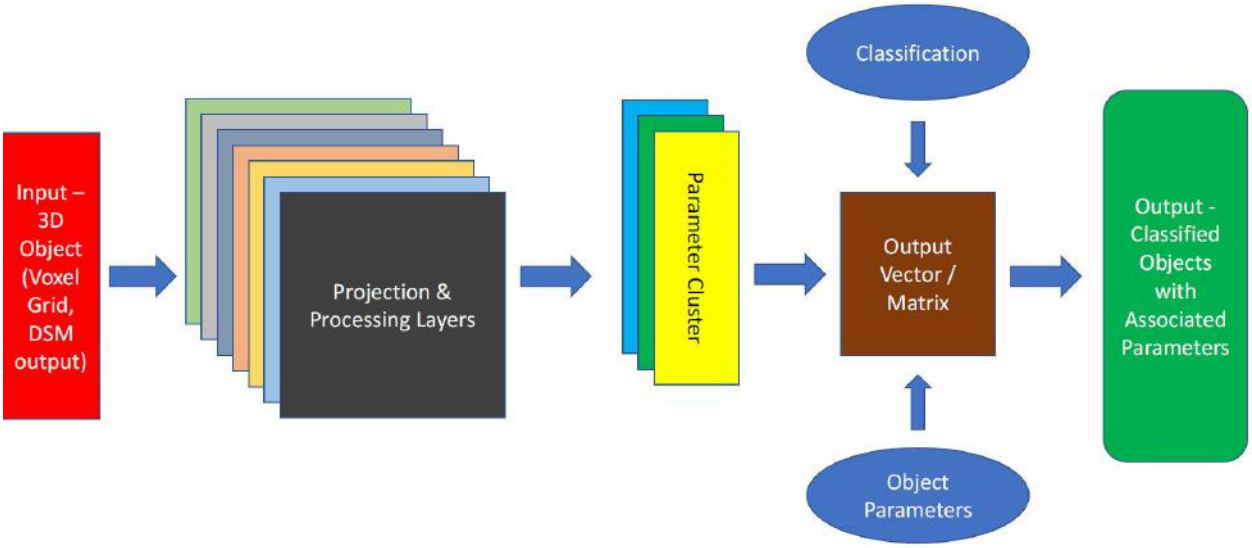


Figure 13. Flow chart of the custom built machine learning algorithm.

within the surveyed volume. Initially, the voxel cluster will be processed by multiple projection layers to extract features and parameters of the cluster. Projection layers would include volume, color composition, dimensions, and pictures of cross-sections of the object with specifications defined by mission requirement. Once the object is identified and classified, the program would continue to associate parameters with the project. For instance, if the object is classified as a cluster of trees, then the parameter association scheme of a group of trees would be invoked and parameters such as where are the highest branches, what is the location of the highest branches, what is the primary color or the trees, etc. would be considered. This process is illustrated in Figure 14.

For better processing efficiency, one could add convolution to the process. To add convolution in this algorithm, one could add a hidden layer, in the form of a weight matrix, to further process the parameter cluster. By using training data and an expected difference threshold to differentiate different categories of objects, the weight matrix should be adjusted so that after transforming the parameter cluster with the weight matrix, each object should be within the difference threshold when compared with the object of the came classification, and vice versa.

Finally, the parameters are calibrated against maps obtained with more accurate mapping methods such as GIS or ortho-mosaic. The two-dimensional maps would be projected on top of the spatial representation with voxels by manually picking distinct features as calibration points.

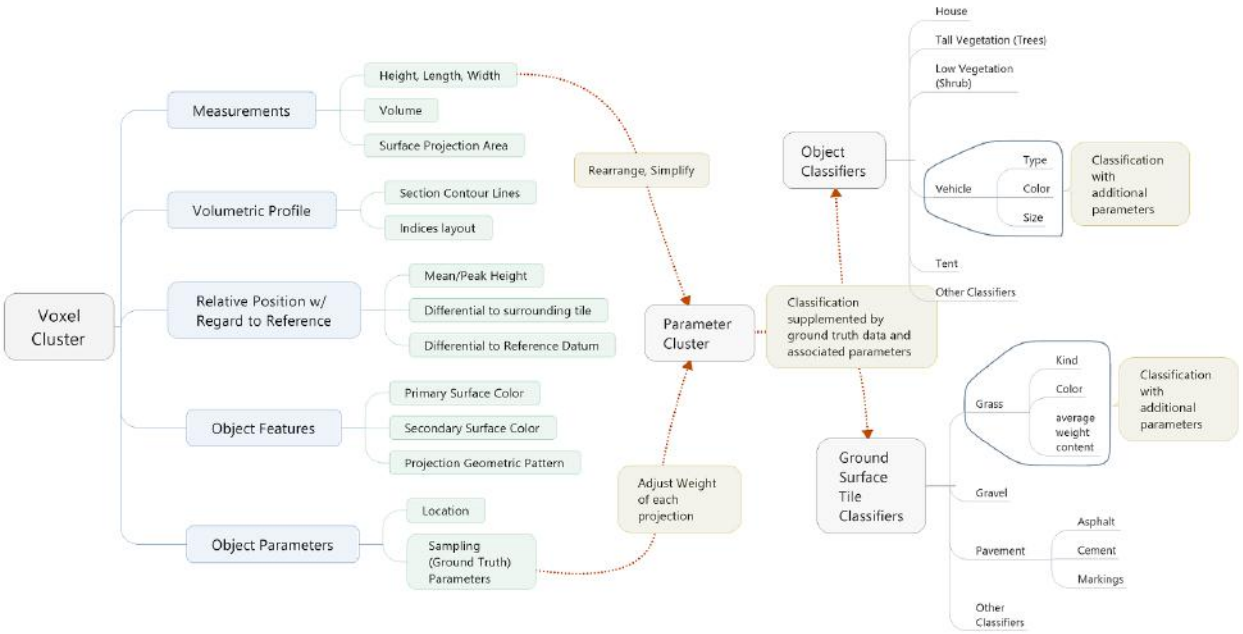


Figure 14. Projection layers and classifiers of voxel clusters.

VII. Environment Reconstruction

The final map is generated by taking in the classified objects and creating a map using a user-defined rendering scheme as shown in Figure 15. This figure is produced by manually aligning the classification result from the object classification step in the Cobra engine. The Cobra engine is developed by game developer Frontier, and is used in many modern personal computer games on the Windows platform such as Planet Coaster, and Elite Dangerous. To increase rendering accuracy, the model of the house and type of tree are calibrated with ground truth data collected from the environment, such as the pictures shown in Figure 4.

Since objects have distinct parameters associated with them, the objects rendered on the map will not be homogeneous. Objects can be dynamically modeled with a pre-defined set of building blocks. For example, to represent a tree on the final map, one could use four parameters to describe the classified tree: where the highest branch occurs, how tall is the highest branch, what is the primary color of the tree, and what is the largest circumference identified with the tree. A default trunk would be stretched to the determined height and placed onto the map at the specified location. Branches and leaves would be positioned in such way that satisfies the determined parameters. For instance, if primary color is green, leaves would be attached to the branch, if primary color is brown, no leaves would be attached to the branches, and if the primary color is black, a layer of grooved black texture would be added to the surfaces of the rendered tree so that it appears burnt.

One of the primary benefits of this approach is the ability to render the environment in different scenarios.[31] Since the environment is rendered from a list of classified objects, it is possible to simulate changes to the environment by editing the list of objects, such as adding artificial landscape to the above landscape. One could also simulate changes such as lighting condition, as shown in Figure 16.

VIII. Advantage and Potential Applications

One underlying principle of the mapping approach this paper proposes is transforming direct mapping data into a analytic based form. This section proceeds first to identify several advantages analytic based information has on its direct counterpart, and applies these advantages to one potentially applicable scenario in the end.

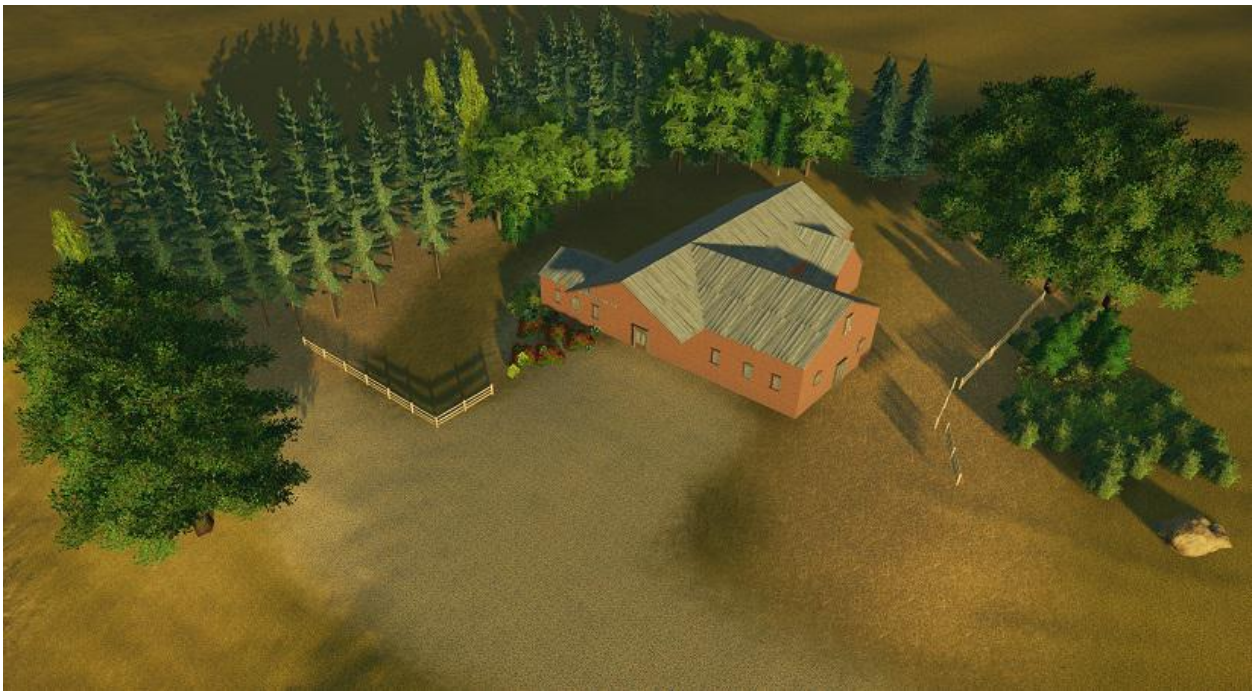


Figure 15. Photo realistic render of an survey area.

A. Advantage of analytic-based information comparing to direct information

When judging a traffic situation at a given location, one could access direct information such as traffic camera feeds, shown in Figure 5(a), or analytic-based information such as the traffic overlay on an abstract mapping interface such as Google Maps, shown in Figure 17(b). If one were to access the camera feed, one would obtain the most detail about the situation but has to analyze the traffic situation by oneself, whereas the Google Maps overlay does not provide the most detailed information, but the terminal user does not need to perform any analysis because the desired analytic outcome is directly presented, with red indicating slow traffic and green indicating fast traffic. With an increasing amount of information and an expanding amount of terminal users transmitting direct information to each user would use up a vast amount of bandwidth and is gruesome for the terminal user to make sense of the data.

Consider someone trying to figure out the traffic between Everett and Tacoma. One would have to go through each traffic camera on the route to determine the traffic condition of the course. If 100 people wish to perform similar tasks, the communications provider would have to transmit the video feed of all involved traffic cameras to all 100 individuals. Alternatively, if all video feed was transferred to an analytic service provider such as Google Maps and the analytic service provider distributes the information in analytic-based form, then the 100 individuals would have made a more comfortable effort trying to understand the traffic situation. Additionally, the data in analytic-based form is usually negligible in size when compared to the direct form (video feed v. overlay data). As a result, with the presence of analytic data, the terminal user would be able to more directly and quickly obtain needed data, and the communication service provider would free up bandwidth.

Analytic data could be presented in many different interfaces.

According to researchers at Princeton University, intuitive interfaces have a low cognitive load, low mental effort is required. The lower the cognitive load, the more users can focus on getting their task done. So, intuitive interfaces have a real advantage regarding efficiency and the quality of the work. They also have the advantage of requiring less training and support; if an interface works the way users expect it to, they dont need to learn how to use it.[32]

With one classified list of objects and their associated parameters, one could create presentations of this dataset with the most efficient interface to different users. For instance, if one has a list aircraft within the controlled space of an airport and associated parameters which include but are not limited to position,

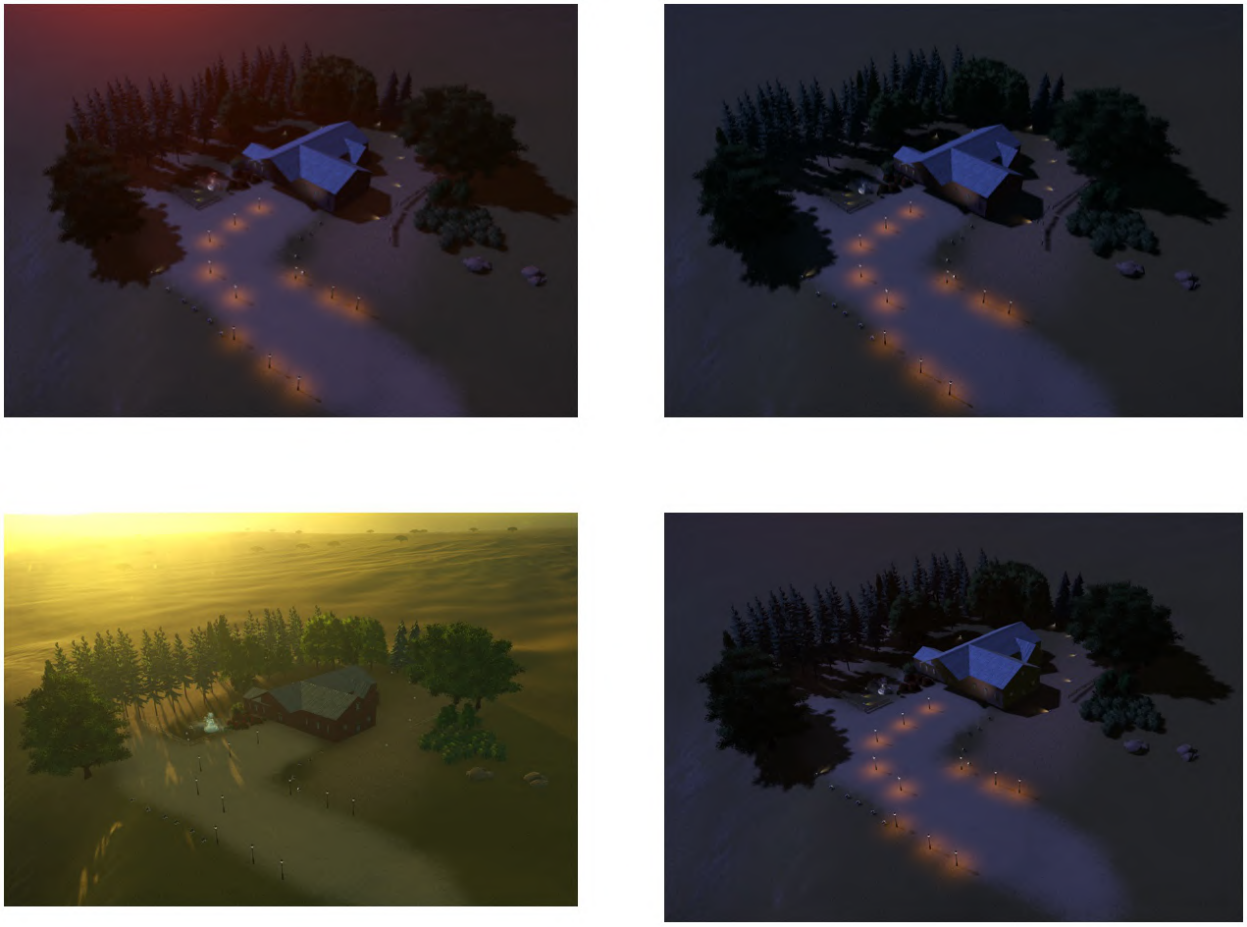


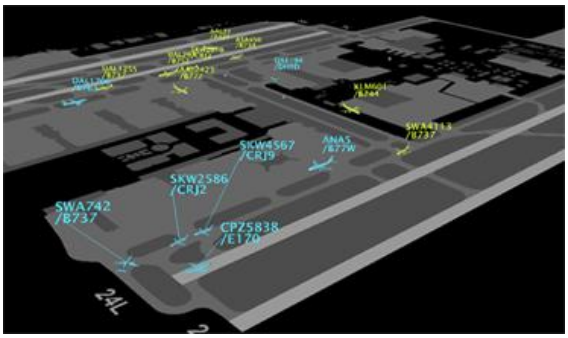
Figure 16. Rendering of the same environment under different lighting conditions.

(a) Highway traffic camera video stream.

(b) Analytical representation of traffic flow.

Figure 17. Analog and abstract means of representing traffic situation.

velocity vector, aircraft type, aircraft ownership, aircraft livery, call sign, and service status, one could create either an overview of the airport tarmac for air traffic control, shown in Figure 18(a), or an interactive photo-realistic rendering for the general public, shown in Figure 18(b).



(a) Surface Visualization Tool (SVT) used by air traffic controllers in Terminal Radar Approach Control (TRACON).[33]

(b) Interactive photorealistic presentation rendered in software Tower3DPro.

Figure 18. Different interfaces to present operational status of aircraft on the ground and within airspace around an airport.

Analytic data could enable simulation within the environment. To give an example, consider a situation where one is tasked with determining the placements of towers of a electrical power line in a remote hill area. Instead of merely looking at aerial photogrammetry data, if one were to look at a list of topographical terrain data and a classified list of vegetation with associated parameters such as vegetation type, size, and position, one could simulate the likelihood of towers being blown down by strong wind with a appropriate simulation algorithm. The same set of analytic data could be used to estimate and optimize total construction cost by simulating placement of access roads, total amount of vegetation to be cleared, and total material needed for construction with appropriate simulation models. Section IX presents a example which renders simulated changes to the environment.

B. Potential Applications

Transforming mapping data from direct to analytic produces many advantages, as discussed in this section. Among many reasonably deducible applications, which could utilize these advantages, one example is presented below.

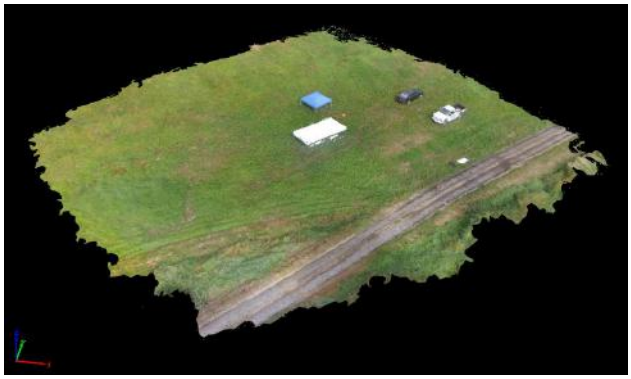
In disaster relief efforts, this kind of surveying system could provide a live presentation of a 3D model of the affected area, live tracking of emergency equipment, personal and supplies, rapid changes within the area, etc. Because the survey system recognizes objects and their parameters, and the render could be customized, the user of this surveillance data could select to see the data in the most relevant way, and all this data could be available to any affected person who owns a tablet or a phone. Imagine during an earthquake, each emergency responder has immediate access to which buildings have collapsed, which building has caught fire, and what roads has been cut off by debris. Each affected person has live access to where food and shelter is offered, where and what kind of medical service is offered, and the status of homes/workplaces of their friends and families.

IX. Experimentation and Results

To validate the work flow outlined in this paper, six image data sets were collected during flight tests conducted at Carnation Farms in Carnation, Washington. Objects such as boxes, tents, and cars were placed in a grass field, where a DJI Phantom III Pro UAV was flown over the area according to a flight-path programmed by either Pix4DCapture or DJI Go software and took pictures at specified geographical locations with a software determined gimbal's setting. The surveyed area was a square of approximately 60 feet by 60 feet, predominately grass field with an access road on the south side of the area and varying objects placed on the field for different datasets. Images acquired from the UAV were automatically geotagged. The UAV was set to fly at an altitude of 45 feet for the entire duration of its flight. Both side and front image overlap were set to 80% in the image acquisition software. Sample images taken by the UAV are shown in 19(a).

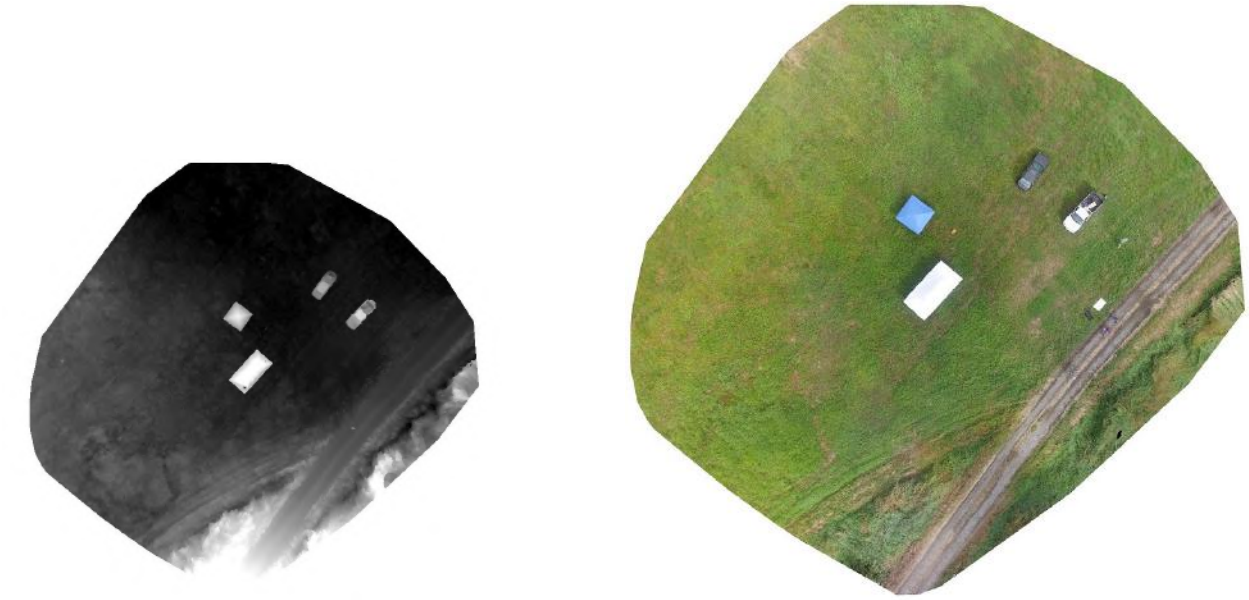
To process the images acquired from UAV into a three-dimensional model, the images were imported into Pix4D and were processed into point clouds. Pix4D matches key points in images during initial photo-alignment to stitch the photos together. Pix4D accounts for possible camera distortions by calibrating image alignment against a known camera profile. Within Pix4D, a point cloud is subsequently processed into a triangular surface mesh, which is a cluster of surfaces generated by linking points within a point cloud according to a filter setting. The resulted surface mesh is shown in Figure 19(b). Compared to the point cloud, the surface mesh will have fewer gaps. The surface mesh was subsequently used to produce a DSM, a digital surface model representing height information of a given point on the ground, and an orthomosaic, a uniform-scale colored representation of the surveyed area. An example of the DSM and orthomosaic is shown in Figure 20. In the DSM, the height is projected on grayscale, the more proximity a point is to white, the higher the point is. The default coordinate system was established in a local reference frame with east as positive x-direction and north as positive y-direction.

(a) Sample images taken at Carnation Farms.



(b) Surface mesh generated in Pix4D.

Figure 19. Raw data and mesh processed from collected aerial images.



(a) Sample DSM generated form Pix4D.

(b) Sample orthomosaic generated form Pix4D.

Figure 20. Raw data and mesh processed from collected aerial images.

A point cloud was also down sampled by PCL into a voxel grid. A comparison of a point cloud and its corresponding voxel grid is shown in Figure 21. For this project, the point cloud was down-filtered into a voxel grid with 0.15-meter lattice. This process significantly reduces file size, and processing time in later steps. A comparison between some points in a point cloud and number of voxels in the corresponding voxel grid is shown in Table 1.

(a) Sample point cloud of the experiment datasets.

(b) Sample voxel grid of the experiment datasets.

Figure 21. Raw data and mesh processed from collected aerial images rendered in PCL Cloud Viewer.

The DSM was analyzed in Matlab, using the mapping toolbox, to calculate the area and volume of test objects. The Matlab calculations have been shown to have good accuracy and were compared to area and volume calculations obtained from the voxel grid. These calculations were used as classifiers in the object classification step.

To perform object classification, the segmented area is segmented into a 0.5-meter side length square grid, and each grid is considered a surface tile. Granted that training data for this environment is limited, a semi-supervised machine learning technique named pseudo-labeling was used to classify the surface tile either as a kind of ground surface or grouped into a cluster to represent an object. The ground surface or object is further classified into different categories of each. A classification tree is shown in Figure 22. The corresponding identification number of each classifier is marked at the end of the classification.

A set of parameters was extracted from the surface tiles and objects and was used either to classify the

Table 1. Comparison between number of points in a point cloud and number of voxel in its corresponding voxel grid with 0.15 meter lattice.

Instance	1	2	3	4	5	6	7	8
Point#	5532494	3870560	4440971	5407455	9513317	8013878	8532906	6825064
Voxel#	106786	65419	78455	103684	157523	129985	150731	287388

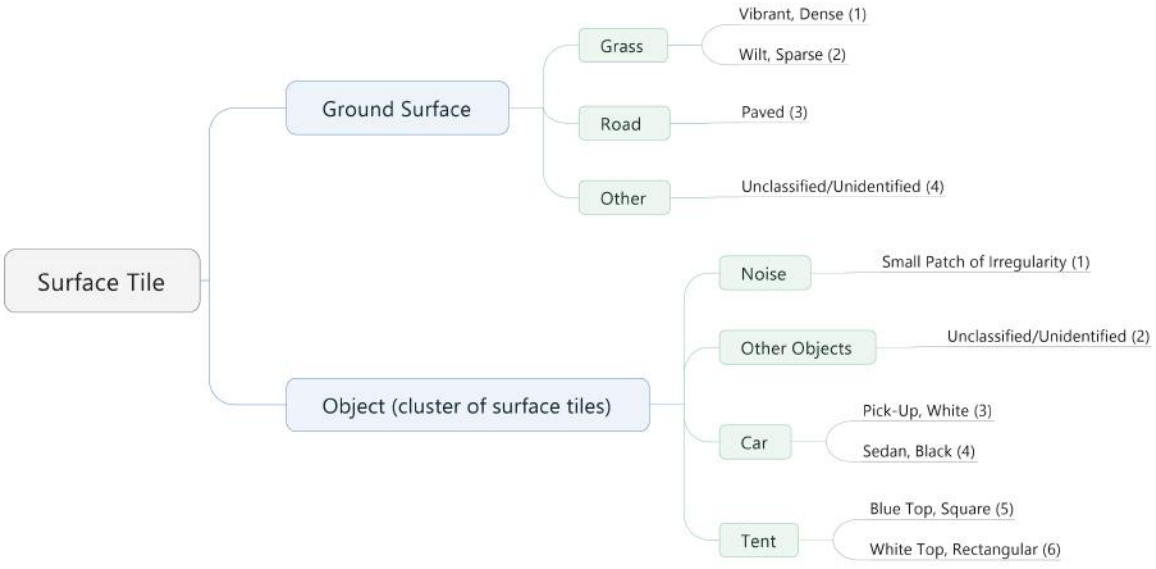


Figure 22. Classification tree used in this experiment.

surface tile or object or was used to define the features of the surface tile or object further. The parameters extracted from surface tiles and objects in this experiment are shown in Table 2. The projection layers used to classify surface tiles, and objects in this experiment are shown in Table 3.

Table 2. Parameters associated with objects and surface tiles

Type	Surface Tile	Object
Parameter 1	x location	x location of all tiles
Parameter 2	y location	y location of all tiles
Parameter 3	min height	height profile
Parameter 4	max height	orientation
Parameter 5	mean height	contour profile
Parameter 6	tile lattice length	area
Parameter 7	mean color scheme (RGB)	volume
Parameter 8	voxel count	primary color scheme
Parameter 9	classification	classification

Table 3. Projection layers used to classify surface tiles and objects.

Type	Surface Tile	Object
Projection 1	mean height	mean Height
Projection 2	relative position to asymptotic line	max Height
Projection 3	relative height to neighboring tiles	relative height to neighboring tiles
Projection 4	proximity to color Ref #1	ground/contour shape
Projection 5	proximity to color Ref #2	primary color scheme
Projection 6	proximity to color Ref #3	volume

A sample within each classification of surface tile is shown in Table 4, and a sample within each classification of surface tile are presented in Table 4.

Table 4. Sample surface tiles with associated parameters and classification.

Surface tile (unit)	x (m)	y (m)	min z (m)	max z (m)	mean z (m)	lattice (m)	mean color (RGB)	voxel count
classification 1	3.5	28.5	-0.562	-0.703	-0.638801	0.5	(76,125,78)	10
classification 2	4	35.5	-0.959	-1.02181	-0.993555	0.5	(36,144,90)	9
classification 3	36	9.5	0.130607	0.0921776	0.106836	0.5	(115,165,162)	10
classification 4	19	54	-1.4857	-1.619	-1.54669	0.5	(80,122,159)	12

A sample within each classification of objects is presented in Table 5.

For surface tiles, two data sets are labeled according to a set of parameters as training data and used to classify surface tiles of a third data set. The classified results are shown in Figure 23. This orthomosaic of this data set is depicted in Figure 20(b). This classification is accurate, as it is observable from the orthomosaic that the grass is more vibrant towards the center of the field, the grass tends to be more yellow and sparse on the northern and western edge of the area, and a patch of grass is wilted in front of the pickup truck. The color-coded map correctly displays these features, and it also correctly indicates the paved access road toward the south end of the field, the erroneous grass pattern towards the southwest corner of the field, some incomplete figures of people controlling the UAV on the edge of the access road, and the four distinct objects in the middle of the field. However, some inaccuracies are also present. The grass around objects tends to be classified as category withered, sparse (2) when in reality it was growing quite well. This may have been caused by people interfering with the grass conditions, i.e. tramping the grass while setting up the objects, or lighting interference, such as shadows, caused by the objects. Some small patches of dry grass

Table 5. Sample objects with associated parameters and classification.

Object (unit)	area (m^2)	voxel	mean z (m)	min z (m)	primary color (RGB)	volume (tile) (m^3)
classification 1	0.5	22	1.23118	0.5852	(29,157,83)	0.61559
classification 2	586.25	26244	1.2959	0.00159	(89,42,110)	759.657
classification 3	11.25	1551	1.5059	0.0126	(128,195,181)	16.7994
classification 4	10.5	187	1.3596	0.1203	(106,164,162)	12.8562
classification 5	14.25	1731	1.5184	0.0085	(148,135,188)	21.5124
classification 6	24.25	3413	1.9444	0.0059	(129, 215, 221)	46.999

were not correctly differentiated. This suggests a limit in result resolution of this particular experimental approach.

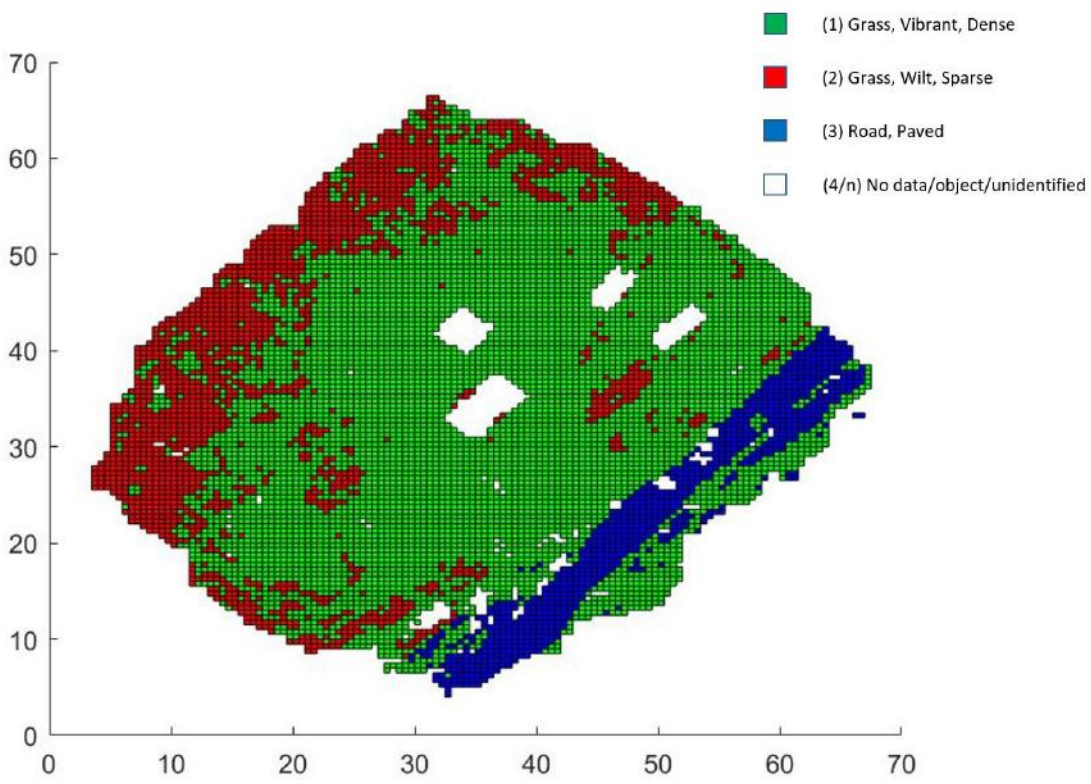


Figure 23. Color-coded representation of classified surface tiles within the surveyed field.

For objects, six data sets were gathered with objects of interest placed at different locations. Five of the data sets are manually labeled and one data set was trained using the semi-supervised machine learning technique described above in this section. The error of area and volume measurement against reference data are shown in Table 6 and the same set of measurements of the same object in different data sets are presented in Table 7. Reference data is obtained by a process described in previous AFSL research,[19] which determines the approach to be accurate. The error of measured volume against reference volume ranges from 3.56% to 8.75%, and the error of measured area against reference area ranges from 1.31% to 5.95%. The error in volume is influenced by choice of elevation reference datum and calculation inaccuracies on the edge of the survey area. In Figure 20(b), it is visible that on the northeast side, the boundary of both cars

is ambiguous. This inaccuracy occurred because there are only a few images taken of the edge of the survey area; it is also determined to be a contributing factor to the error of measured area against reference area. Another factor which impacted error in a measured area is processing resolution, because on the edge of the object the object does not occupy a complete tile. Applying a constant, determined by the circumference and orientation of the object, to calibrate the measured area reduces the error. Increasing processing resolution (reducing tile side length) could also reduce error in measured area.

Table 6. Error of area and volume measured with this experiment as opposed to the same parameters by another relatively accurate method.

	Measured		Actual		Error	
Object (unit)	Area (m^2)	Volume (m^3)	Area (m^2)	Volume (m^2)	Area	Volume
(3) Pickup, White	9.15	10.1415	8.7514	9.7081	4.554%	4.4643%
(4) Sedan, Black	7	6.9971	7.3483	6.8352	-4.7398%	2.3686%
(5) Tent, Blue Top, Square	10.65	14.7271	10.0515	13.5419	5.9543%	8.7521%
(6) Tent, White Top, Rectangular	19.25	33.1394	18.997	34.3638	1.3318%	-3.5631%

In Table 7, the data could conclude that the data acquisition and processing method could produce similar results across different data sets generated by different scans. The standard deviation of primary RGB column is obtained by averaging the standard deviation of each color spectrum. The measured volume of the same object varies about 3% across different data sets, measured area of the same object varies about 4% to 10% across different data sets, and primary color tone differs approximately 1% to 5.8% about one color spectrum. The area deviation is mainly the result of one extreme data point, whose probability of being an erroneous measurement is explained in the previous paragraph. By closely examining the original data sets, one could discover generic shift in color tone. Considering cloudy weather conditions and relative time at which the scan was performed, one could conclude the primary RGB variation of the blue tent is partially the result of changes in external lighting conditions. To calibrate for color variations, reference points representing pure red, green, and blue color should be placed in the survey area, and the color tone of the resulting data set should be adjusted so that the reference points show their intended color.

Table 7. Comparison of measurement of the same object across different data sets.

Object (unit)	Dataset #	Measured Area (m^2)	Measured Volume (m^3)	Primary color (RGB)
(3) Pickup, White	1	7.85	10.5331	128,195,181
(3) Pickup, White	2	9.15	10.1415	135,193,183
(3) Pickup, White	3	9.86111	10.6046	127,189,182
Standard Diviation		1.019823585	0.24930718	2.8046
(5) Tent, Blue Top, Square	1	10.65	14.7271	148,135,188
(5) Tent, Blue Top, Square	4	10.3594	14.3304	124,128,207
(5) Tent, Blue Top, Square	5	9.85	13.71	182,121,187
Standard Diviation		0.404956113	0.512633635	15.804

To summarize, the process described in this paper has been applied to a real environment, and the classification of both surface tiles and objects could be concluded as generally accurate. An error analysis of parameters measured with this process, when compared to parameters produced by a relatively accurate method, is performed. While the error is in a reasonably acceptable level, several sources of error are identified and approach to eliminate the sources or error are proposed. Therefore, one could conclude the process, however still primitive at this stage, work in the real environment to a reasonable degree of satisfaction. Subsequently, the lists of classified surface tiles and objects are passed on to the rendering step.

To demonstrate the versatility feature of the process described in the paper, two type of rendering is made for this experiment from the list of classified surface tiles and objects. The first is a photo-realistic rendering

using the Cobra engine with artificial object alignment; the second is a statistically accurate rendering which is generated automatically.

By manually aligning objects within the classification list in the Cobra Engine, the following rendered environment is obtained, as shown in Figure 24. Each tile of grass is represented as an object of grass, either green or yellow, and elevation of the grass is aligned with the highest voxel of the tile. The road is rendered with mud texture, and terrain elevation is alighted with the mean height of each tile labeled road. The objects are modeled and placed at specified locations, each with its height aligned with the classification list. A rendering with the camera more focused on the objects is shown in Figure 25(a), and a rendering with a different camera angle are shown in Figure 25(b).

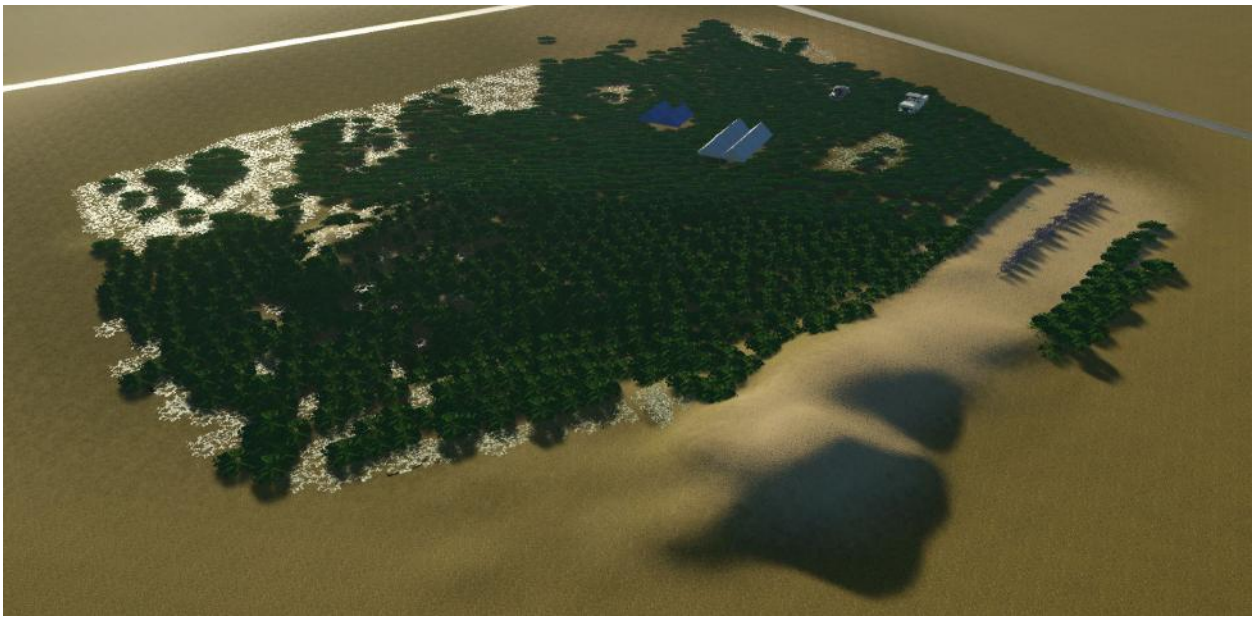
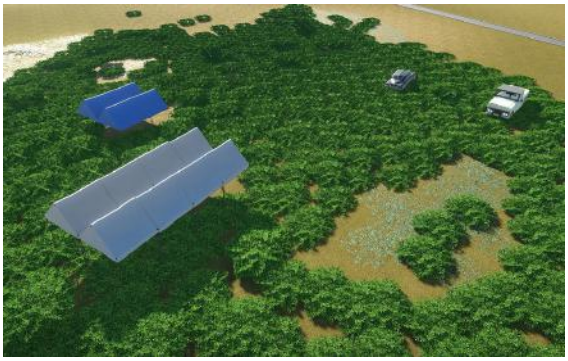
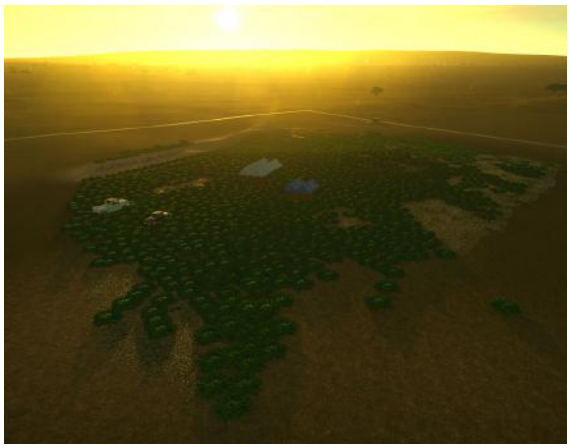


Figure 24. Rendered environment of the Carnation Farms data set.



(a) Render with more focus on the objects.



(b) Render of the surevy area at an different angle.

Figure 25. Raw data and mesh processed from collected aerial images.

As discussed in the paper, one of the advantages of acquiring the map as a list of classified list of surface tiles and objects, when compared to the conventional approach, is the additional capability of simulating changes in the environment. Figure 26 shows the same area under different lighting conditions corresponding to different times of the day. This ability has many practical applications, and this experiment attempted to determine lighting equipment placements for an end-of-year party at the surveyed area with real-time

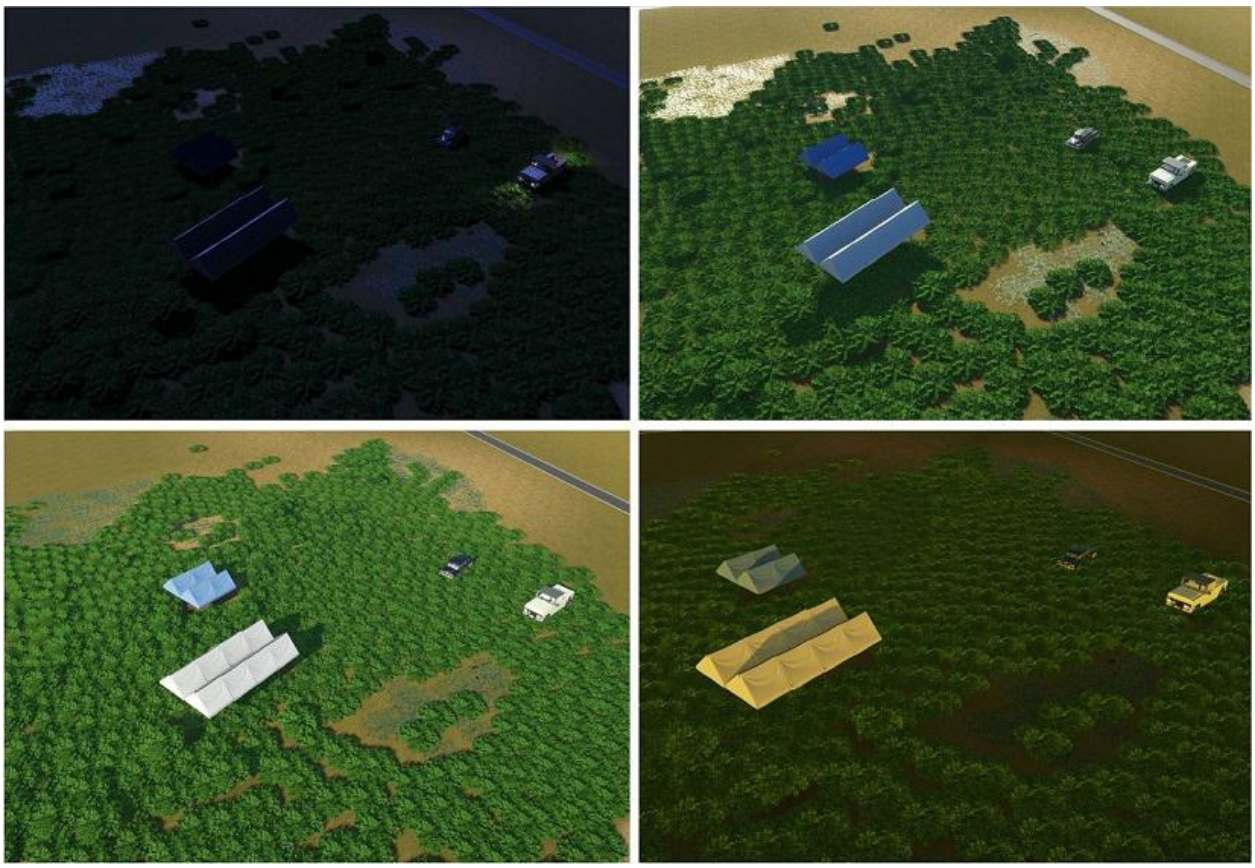
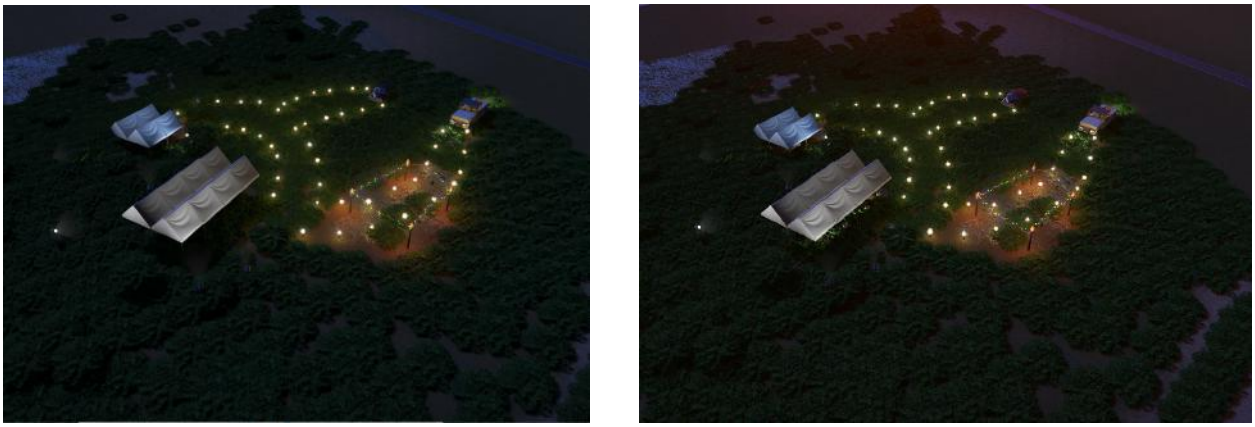


Figure 26. Rendered environment at different time of day under different lighting conditions.

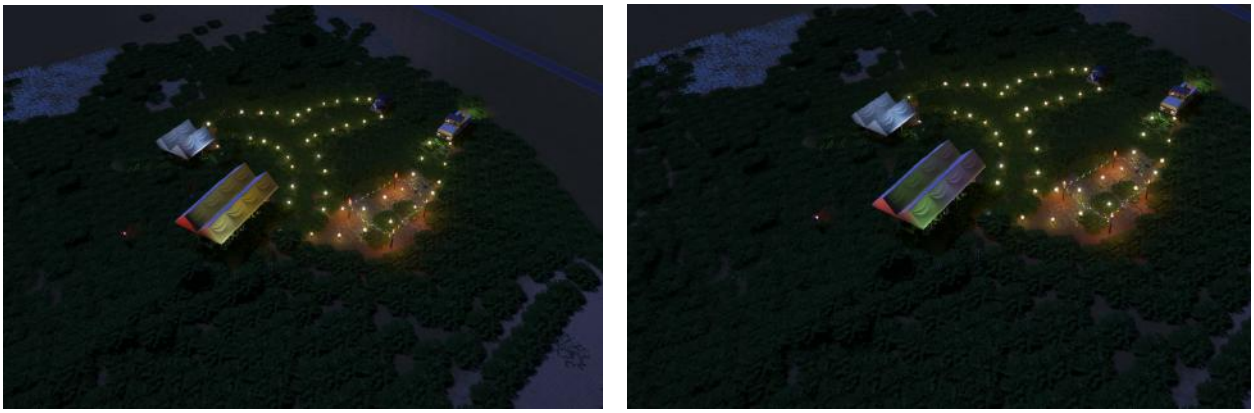
simulation of the effects of lighting placement patterns. In Figure 27(a), time is set at midnight. Torches and colorful lights are placed so that a central well-illuminated area is established, sparkling lights are placed to illuminate paths and create a wonderland-like atmosphere, and illumination lights are placed around the tents. Figure 27(b) shows the same setting in the twilight, evident by the orange halo on the edge of the image. It is thus conclusive that the same lighting setup would function in twilight as well as in the dark, since by comparison, the effect of changed lighting conditions, midnight to twilight, does not seem to interfere much with the lighting effects. One could also experiment with other changes, such as changing the color of the illumination lights, as shown in Figure 28. All changes in the environment described above are conducted in real time, whose result is also presented in real time.



(a) Render of simulated changes to the surevy area.

(b) Render of simulated changes to the surevy area under different external lighting condition.

Figure 27. Raw data and mesh processed from collected aerial images.



(a) Render of simulated changes to color of projection light on the white tent to yellow fading to red.

(b) Render of simulated changes to color of projection light on the white tent to purple fading to red.

Figure 28. Render of simulated changes to color of projection light on the white tent.

One could also simulate changes in the environment between different states of the environment. Figure 29(a) introduces a car at the end of the road. Assume the car has moved forward a certain distance along the road. If one assumes the road is muddy, and grass would be knocked down if a car has driven over it, one could roughly simulate a deep tire track, and grass knocked lower. Figure 29(b) shows the result.

The rendering process could be automated. This experiment demonstrates the ability to automatically generate statically accurate renders of an environment using the Matlab software. Components used to generate the render are shown in Figure 30. The tents, shown in Figure 30(a), are modeled as triangular roofs supported by poles at the corner. The pole starts at the elevation reference datum and stops at the lower end of the roof. The highest point of the roof is aligned with the height voxel of the object. The grass tiles, shown in Figure 30(b), are obtained as approximating each individual grass as a vertical line, starting

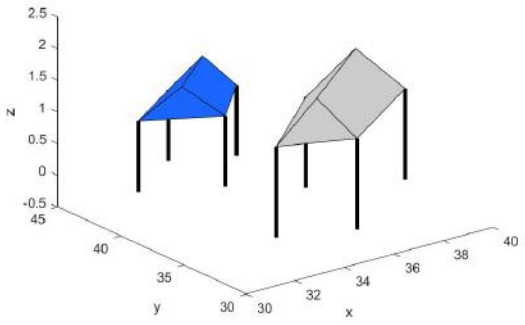
(a) Render of simulated placement of a car in the surevy area.

(b) Render of simulated changes to the surevy area if a car has moved along the road.

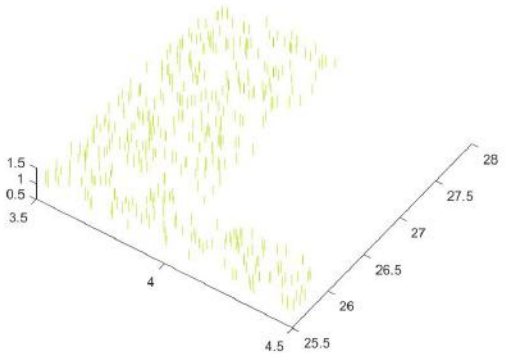
Figure 29. Render of simulated changes to the environment of placing and driving a car.

at a certain height above the reference datum, which is determined by the difference in height of the highest voxel and the lowest voxel of that tile. The line ends at a height between the height voxel and the lowest voxel. This terminal height is determined by random sampling, the result is a normal distribution with population mean at the mean height of the tile and a 100% confidence interval spanning the gap between height of the highest voxel and the lowest voxel. The voxel count is used to determine the relative density of the grass. A constant is multiplied to the voxel count to determine the number of total grass lines placed in a certain tile. Due to the high entropy nature of location of each individual grass plant relative to a grass field, the lines representing individual grass are placed at randomly generated locations within the boundary of its tile. The paved road is represented as a square mesh spanning the tile at the mean height of the tile. For processing simplicity, a uniform grey texture is used to represent pavement.

This rendering, hence, properly accounts for differences between different tiles, resulting in a statically accurate rendering of the environment. A complete render of the survey area, which produced an orthomosaic shown in Figure 31 is presented as Figure 32.



(a) Sample render of objects in the experiment data sets.



(b) Automatic render of six sample grass tiles.

Figure 30. Components from which automatic render of the environment is generated.

Figure 31. Orthomosaic of the environment used to generate the automatically rendered environment.

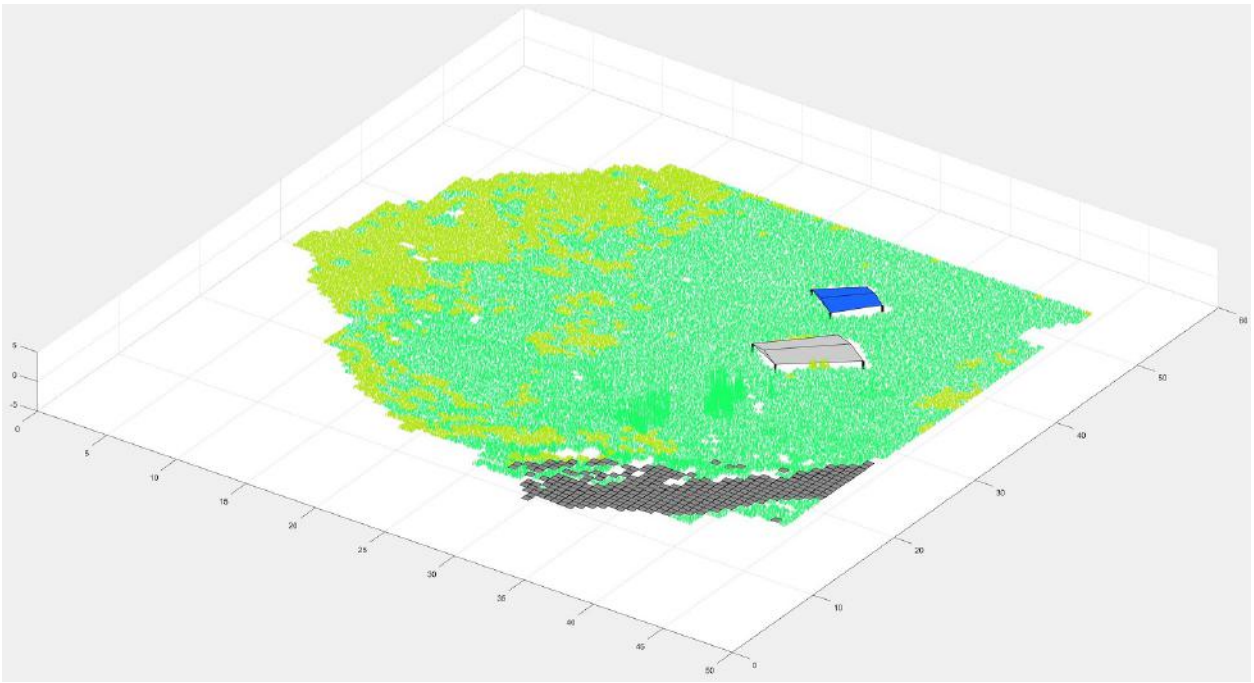


Figure 32. Automatic rendered environment of the experiment data set.

X. Conclusions

This paper proposed and demonstrated a UAV mapping process which uses photo telemetry sensors, such as a camera, multi-spectral sensors, or LiDAR, onboard a UAV to acquire a three-dimensional representation of the environment. The system then used machine learning to classify objects and surface tile types and finally render the environment in a user-defined interface. An experiment was conducted to verify that the process works in a real environment. The proceedings and results of the experiment were discussed in Section IX. Using a consumer grade UAV with an attached camera, the classification of surface tiles are generally reasonable, and classification of objects are accurate. Measurements and calculations obtained from the experiment are within an acceptable margin of those parameters measured by a relatively reliable method.

In the experiment setup described in Section IX, changes in environmental lighting conditions, such solar zenith, clouds and shadows, cause color variation in data sets collected at different times; in future practice color calibration points should be setup within the survey area to correct each data set to a uniform color tone. Additionally, changes within the survey area due to environmental factors such as wind and personnel movement cause disturbances in the resulting data set. To ensure accurate results, data acquisition should be collected as fast as possible and people should completely vacate the survey area during UAV scans. Furthermore, it is observed that objects near the edge of the survey area tend to contain more artifacts than objects towards the center of the survey area. To suppress this source of error, an operator should extend the boundary of survey area further than desired, so that all objects of interest are accurately represented in the resulting data set.

The UAV mapping process demonstrated in this paper, in principle, aims at transforming direct mapping data into an analytic based form, so that one set of mapping data could be rendered in multiple interfaces, and could enable simulation of changes to the survey area. Both abilities will increase utility of mapping data, and, as demonstrated in the example in Section VIII, enable further, never-before-seen solutions to real-world problems.

Acknowledgments

The software used for data processing include Pix4D, Matlab, and Visual Studio with licenses provided by UW.

The software used to render the displayed environments include Microsoft Windows 3D Viwer, Minecraft, Unity Engine and Planet Coaster, authored by authors' personal possession of standard release licenses of the above mentioned products, or other form of usage authorization.

The authors would like to thank Hannah Rotta, the AFSL Flight Operations Director, for coordinating and conducting flight tests associated with the project.

Many members of the the AFSL contributed to the research including Gavin Cai, Sicong Huang, Richard Jiang, and Matthew Lemelin.

References

[1]Erdos, D., Erdos, A., and Watkins, S. E., "An experimental UAV System for Search and Rescue Challenge," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 28, No. 5, 2013, pp. 32–37.

[2]Lum, C. W. and Vagners, J., "A Modular Algorithm for Exhaustive Map Searching Using Occupancy Based Maps," *Proceedings of the 2009 Infotech@Aerospace Conference*, Seattle, WA, April 2009.

[3]Lum, C. W., Summers, A., Carpenter, B., Rodriguez, A., and Dunbabin, M., "Automatic Wildfire Detection and Simulation Using Optical Information from Unmanned Aerial Systems," *Proceedings of the 2015 SAE Aerotec Conference*, Seattle, WA, September 2015.

[4]Man, Q., Ma, S., Xia, L., and Wang, Y., "Research on security monitoring and health management system of medium-range UAV," *Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on*, IEEE, 2009, pp. 854–857.

[5]Saleri, R., Pierrot-Deseilligny, M., Bardiere, E., Cappellini, V., Nony, N., De Luca, L., and Campi, M., "UAV Photogrammetry for Archaeological Survey: The Theaters Area of Pompeii," *Proceedings of the Digital Heritage International Congress (DigitalHeritage)*, IEEE, 2013, pp. 497–502.

[6]Rogers, K. J. and Finn, A., "Frequency Estimation for 3d Atmospheric Tomography Using Unmanned Aerial Vehicles," *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, IEEE, 2013, pp. 390–395.

[7]Tokekar, P., Vander Hook, J., Mulla, D., and Isler, V., "Sensor Planning for a Symbiotic UAV and UGV System for Precision Agriculture," *IEEE Transactions on Robotics*, Vol. 32, No. 6, 2016, pp. 1498–1511.

[8]Ying-cheng, L., Dong-mei, Y., Xiao-bo, D., Chang-sheng, T., Guang-hui, W., and Tuan-hao, L., "UAV Aerial Photography Technology in Island Topographic Mapping," *Image and Data Fusion (ISIDF), 2011 International Symposium on*, IEEE, 2011, pp. 1–4.

[9]Liu, X., Chen, P., Tong, X., Liu, S., Liu, S., Hong, Z., Li, L., and Luan, K., "UAV-Based Low-Altitude Aerial Photogrammetric Application in Mine Areas Measurement," *Earth Observation and Remote Sensing Applications (EORSA), 2012 Second International Workshop on*, IEEE, 2012, pp. 240–242.

[10]Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11]Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12]LeCun, Y., Bengio, Y., et al., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, Vol. 3361, No. 10, 1995, pp. 1995.

[13]Cireşan, D. C., Meier, U., and Schmidhuber, J., "Transfer learning for Latin and Chinese characters with deep neural networks," *Neural Networks (IJCNN), The 2012 International Joint Conference on*, IEEE, 2012, pp. 1–6.

[14]Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., "Caffe: Convolutional architecture for fast feature embedding," *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 675–678.

[15]Lohman, D. F., "Spatial Ability: A Review and Reanalysis of the Correlational Literature." Tech. rep., DTIC Document, 1979.

[16]Donnon, T., DesCôteaux, J.-G., and Violato, C., "Impact of Cognitive Imaging and Sex Differences on the Development of Laparoscopic Suturing Skills," *Canadian journal of surgery*, Vol. 48, No. 5, 2005, pp. 387.

[17]Simmons, A., "Spatial Perception from a Cartesian point of view," *Philosophical Topics*, Vol. 31, No. 1/2, 2003, pp. 395–423.

[18]Jeannerod, M. and Jacob, P., "Visual Cognition: a New Look at the Two-Visual Systems Model," *Neuropsychologia*, Vol. 43, No. 2, 2005, pp. 301–312.

[19]Lum, C. W., Mackenzie, M., Shaw-Feather, C., Luker, E., and Dunbabin, M., "Multispectral Imaging and Elevation Mapping from an Unmanned Aerial System for Precision Agriculture Applications," *Proceedings of the 13th International Conference on Precision Agriculture*, St. Louis, MO, August 2016.

[20]Lum, C. W., Rysdyk, R. T., and Pongpunwattana, A. A., "Autonomous Airborne Geomagnetic Surveying and Target Identification," *Proceedings of the 2005 Infotech@Aerospace Conference*, AIAA, Arlington, VA, September 2005.

[21]Lum, C. W., Vagners, J., and Rysdyk, R. T., "Search Algorithm for Teams of Heterogeneous Agents with Coverage Guarantees," *AIAA Journal of Aerospace Computing, Information, and Communication*, Vol. 7, January 2010, pp. 1–31.

[22]Ueunten, K. K., Lum, C. W., Creigh, A. A., and Tsujita, K., "Conservative Algorithms for Automated Collision Awareness for Multiple Unmanned Aerial Systems," *Proceedings of the 2015 IEEE Aerospace Conference*, Big Sky, MO, March 2015.

[23]Lum, C. W., Rowland, M. L., and Rysdyk, R. T., "Human-in-the-Loop Distributed Simulation and Validation of Strategic Autonomous Algorithms," *Proceedings of the 2008 Aerodynamic Measurement Technology and Ground Testing Conference*, Seattle, WA, June 2008.

[24]Costa, D. B. and Mendes, A. T., "Lessons Learned from Unmanned Aerial System-Based 3D Mapping Experiments," 2016.

[25]Rouse, J., Haas, R., Scheel, J., and Deering, D., "Monitoring Vegetation Systems in the Great Plains with ERTS," *Proceedings of the 3rd Earth Resource Technology Satellite (ERTS) Symposium*, Vol. 1, 1974, pp. 48–62.

[26]Priestnall, G., Jaafar, J., and Duncan, A., "Extracting urban features from LiDAR digital surface models," *Computers, Environment and Urban Systems*, Vol. 24, No. 2, 2000, pp. 65–78.

[27]Ashburner, J. and Friston, K. J., "Voxel-based morphometrythe methods," *Neuroimage*, Vol. 11, No. 6, 2000, pp. 805–821.

[28]"3D Pixel / Voxel," Bilderzucht Visual Experiments `http://www.bilderzucht.de/blog/3d-pixel-voxel/`, Accessed Dec 11, 2017.

[29]"Point Cloud Library (PCL) Documentation," Point Cloud Library (PCL) - Documentation `http://pointclouds.org/documentation/`, Accessed Dec 11, 2017.

[30]"Point Cloud Library (PCL) Documentation - 3D Object Recognition based on Correspondence Grouping," Point Cloud Library (PCL) - Documentation `http://pointclouds.org/documentation/tutorials/correspondence_grouping.php#correspondence-grouping`, Accessed Dec 11, 2017.

[31]Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S. N., Rosaen, K., and Vasudevan, R., "Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks?" *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[32]"The Intuitive Interface," The Trustees of Princeton University `https://ux.princeton.edu/designing-experience/intuitive-interface`, Accessed Dec 11, 2017.

[33]"NextGen Opens a Window to Airports," Federal Aviation Administration (FAA) - Documentation `https://www.faa.gov/nextgen/snapshots/stories/?slide=38`, Accessed Dec 11, 2017.