Pi Y., Nath N. and Behzadan A. (2020). "Deep neural networks for drone view localization and mapping in GPS-denied environments" In: *Proc. 37th CIB W78 Information Technology for Construction Conference* (CIB W78), São Paulo, Brazil, pp. 1-16. DOI: http://dx.doi.org/10.46421/2706-6568.37.2020.paper001

DEEP NEURAL NETWORKS FOR DRONE VIEW LOCALIZATION AND MAPPING IN GPS-DENIED ENVIRONMENTS

Yalong Pi[1], Nipun D. Nath[2], and Amir H. Behzadan[3]

Abstract: Geolocation information used to create ground survey maps is critical in many domains including construction, transportation, urban planning, disaster response, agriculture, forestry, ecology, mining, and defence. Rapid development of the UAV technology due in part to better camera resolution, longer flight range, and higher data storage capacity, has enabled more efficient, less expensive remote sensing at lower altitudes. While UAV remote sensing requires uninterrupted access to on-board GPS data, such information may not be always available (e.g., in GPS-denied environments). Also, UAV pilots and enthusiasts may not share location information due to privacy issues or lack of knowledge about geolocation meta-data. Advances in vision-based deep learning have created new opportunities in UAV data collection and navigation beyond the constraints of GPS-enabled environments. In particular, UAVs equipped with RGB cameras can be deployed to automatically recognize and map target objects and landmarks on the ground using deep learning methods. This paper presents a method for real-time aerial data collection and GPS-free mapping. This is achieved by passing UAV visual data through a convolutional neural network (CNN) to identify and localize target objects, followed by applying geometric transformation to instantaneously project detected objects on an orthogonal map, all without GPS data. Particularly, the pixel coordinates of ground objects and four reference points are first detected by the CNN model. Next, viewpoint transformation is applied to project detected objects from the UAV's perspective view onto an orthogonal view. Experiments conducted in an outdoor field yield a small average error of <2% along X and Y axes.

Keywords: Convolutional Neural Network (CNN), GPS-Denied, Unmanned Aerial Vehicle (UAV).

1 INTRODUCTION

1.1 Object Mapping Using Geolocation Information

Geolocation information are widely used by mapping platforms (e.g., Google Maps, OpenStreetMap, ArcGIS) to overlay objects on maps, and are critical components of decision support and research infrastructure in many fields including agriculture, forestry, ecology, marine industry, mining, commerce, construction, transportation, defense, disaster response, and urban planning. Existing methods of mapping data collection can be divided into two broad categories of ground survey and aerial remote

[1] Ph.D. Candidate, Department of Construction Science, Texas A&M University, College Station, TX 77843, USA, piyalong@tamu.edu
[2] Ph.D. Student, Zachry Department of Civil Engineering, Texas A&M University, College Station, TX 77843, USA, nipundebnath@tamu.edu
[3] Clark Construction Endowed Associate Professor, Department of Construction Science, Texas A&M University, College Station, TX 77843, USA, abehzadan@tamu.edu

sensing. Ground survey methods involve using total stations and ground photogrammetry. A total station is used to measure the coordinates of ground control points (GCPs) to produce detailed vector maps, which can reach an accuracy of <1 millimeter. However, processing data using this method is time consuming, and the range of data collection is relatively limited. Ground photogrammetry, on the other hand, involves using optical cameras, sonar, radar, and laser scanners to collect digital spatial information. This method can produce 3D models providing that high-accuracy GCP information is available (Adams et al. 2011). In contrast to ground survey methods, aerial remote sensing includes collecting rasterized images from satellites, airplanes, and unmanned aerial vehicles (UAVs). More recently, GPS-enabled light detection and ranging (LIDAR) equipment has been also used from low-altitude aircrafts for this purpose (Carter et al. 2012).

Using satellite imagery for real-time applications at micro-urban level of detail may pose certain challenges. In addition to high cost of acquiring and processing large volumes of satellite data, accessing such data in some locations or during certain times may be restricted, data could be interrupted due to cloud blocking (Adams et al. 2011), and update rate could be too slow (e.g., once every 24 hours) (NASA 2020). Moreover, majority of existing remote sensing data analysis techniques rely on the availability of GPS data, which may hinder their applications in GPS-denied environments (locations and times where GPS signal is weak or non-existent) for timely status monitoring, decision-making, resource deployment, and emergency management. Examples include locations inside heavily urbanized areas or surrounded by dense vegetation, indoor facilities (e.g., tunnels, sheltered spaces, underground, underwater), GPS-jammed locations (i.e., presence of radio signal interference impacting GPS signal reliability), unknown territories (e.g., battlefield, extraterrestrial exploration), or during severe weather.

In the past decade, rapid development of the UAV technology due in part to better camera resolution, longer flight range, and higher data storage capacity, has enabled more efficient, less expensive remote sensing methods at lower altitudes (Adams et al. 2011). Despite their affordability and ease of use, UAV remote sensing is still not a trivial task since it requires uninterrupted access to on-board GPS data to calculate location information and generate ground survey maps. However, in some cases, such information may not be readily available (e.g., in GPS-denied environments), and sometimes, UAV pilots and enthusiasts may not share location information due to privacy issues or the lack of knowledge about how to create and exchange geolocation meta-data. In light of these limitations, advances in computer vision (CV) and machine learning (ML) have created new opportunities for pushing the boundaries of UAV data collection and navigation beyond the constraints of GPS-enabled environments.

In particular, UAVs equipped with standard RGB cameras can be deployed to automatically recognize landmarks (both manmade and natural) and targets of interest (ToIs) on the ground using deep learning methods such as convolutional neural networks (CNNs) (Adams 2011). This research aims at developing a method for GPS-free aerial data collection and mapping in real-time. This is achieved by passing the collected visual data through a CNN architecture, namely, RetinaNet (Lin et al. 2017) to identify and localize ToIs in each video frame, followed by applying geometric transformation to instantaneously project detected ToIs on an orthogonal map, all without involving GPS data. In this paper, current research progress on the relevant topics, as well as two approaches for collecting, analyzing, and mapping UAV visual data using ground

reference points are introduced and validated in real-world experiments. Potential applications of the proposed methods are also discussed.

1.2 Previous Work

CNNs have been used in recent years for image classification, recognition, and semantic segmentation (i.e., extracting semantic information from images). For example, Krizhevsky et al. (2017) proposed AlexNet, a CNN with 8 layers, which accomplished top-5 error of 15.3% on ImageNet dataset. Later, Simonyan and Zisserman (2016) introduced VGGNet with up to 19 layers and achieved higher accuracy. He et al. (2017) designed a network with residual blocks, namely ResNet, that outperformed human by 3.57% on ImageNet.

Object detection further expands upon object classification by localizing the pixel-level position of target classes in the image. Examples of state-of-the art algorithms for object detection include R-CNN (Girshick 2014) and Fast R-CNN (Girshick et al. 2015) that use region of interest (RoI) to improve prediction speed compared to the traditional sliding box methods. To further reduce the processing time, Ren et al. (2017) proposed Faster R-CNN by adopting region proposal network (RPN) that achieved mean average precision (mAP) of 73.2% on PASCAL dataset (Russakovsky et al. 2015) (containing everyday objects such as people and cars). These methods involve two stages, i.e., proposing candidate regions and classifying proposed regions, and can produce significantly accurate predictions. There are also one-stage detectors that are specifically designed to compute faster, but with smaller accuracy loss (Lin et al. 2017). For example, Redmon et al. (2018) introduced You-Only-Look-Once (YOLO) version 3, which is the latest version of YOLO (Redmon et al. 2016) and YOLO version 2 (Redmon et al. 2017) algorithms. In particular, YOLO adopts feature pyramid network (FPN) and anchor boxes in the gridded image to predict class and location simultaneously. Liu et al. (2016) introduced single shot detector (SSD) using pre-defined boxes and scaled feature maps, and achieved 76.8% mAP on the VOC dataset (Everingham et al. 2010). Recently, Lin et al. (2017) presented RetinaNet with a novel focal loss integrated with FPN to better learn hard examples, resulting in 37.8% average precision (AP) on the COCO dataset (Lin et al. 2014).

While the field of CV, and in particular, CNNs for object detection and segmentation is rapidly developing, few studies have focused on aerial object recognition using CNNs. For example, Han et al. (2014) proposed a new weakly supervised algorithm that achieved an overall 94.45% precision on Google Earth, and Landsat (dataset containing airplanes, vehicles, and airports). Cheng et al. (2016) introduced rotation-invariant CNN with a new objective function via enforcing training samples before and after rotating, resulting in 72.63% precision on the VHR dataset. Tang et al. (2014) used extreme learning machine (ELM) and deep neural network (DNN) to classify if satellite images contain boats, and reached 97.58% accuracy. However, these methods detect only pixel coordinates of objects in satellite images without providing any geographic coordinates.

In the field of UAV-based remote sensing, researches have mostly focused on photogrammetry and 3D mapping. For example, Friedel et al. (2018) showed the possibility of mapping landscape soils and vegetation through self-organizing map (SOM) of the aerial spectral information. Chao et al. (2012) demonstrated the use of monocular cameras to build visual simultaneous localization and mapping (SLAM) tools to navigate a UAV without GPS. Suh and Choi (2017) explored using GCPs and UAVs to produce maps with 14-cm error in a mining site. Ventuna et al. (2016) investigated 3D modeling of coastal fish nursery environment with drone cameras and GCPs with high accuracy

(89.1%). Cunliffe et al. (2016) used UAV-captured photos and structure-from-motion (SfM) to produce accurate grain biophysical data. While these approaches can produce local 3D maps of the topography and space, to obtain target information (quantity, coordination, and status), post-processing (e.g., map stitching and target marking) and GPS data are still needed. It is evident that in GPS-jammed sites (i.e., military sites or enemy territories), GPS-based mapping methods fail to provide instantaneous position, velocity, and time (PVT) information (Asher et al. 2011).

Some researchers have tried to leverage other onboard sensors such as inertial measurement unit (IMU), stereo cameras, Wi-Fi, radio-frequency identification (RFID) (Balamurugan et al. 2016), and sonar to extract PVT information in GPS-denied environments. For example, Pestana et al. (2013) utilized OpenTLD tracker to follow moving objects such as humans and cars with UAV cameras without GPS. Rajeev et al. (2019) investigated the use of prior knowledge and augmented reality (AR) to navigate in indoor GPS-denied areas. Bachrach et al. (2012) equipped drones with RGB-D cameras to automatically plan complex 3D paths. Scaramuzza et al. (2014) designed a visual–inertial multi UAV system with camera and IMU to autonomously fly and map the environment. While previous work has investigated sensor data fusion for GPS-denied UAV navigation, the problem of single monocular RGB camera-based UAV navigation using ToI recognition (locating, projection, counting) and mapping still remains unexplored. As previously discussed, the ability to locate and map objects using ordinary RGB cameras mounted on lightweight UAVs could be of significant value in many applications, and reduce the cost of data collection. Therefore, this research is geared toward the introduction and validation of a robust, computationally efficient method for identifying, locating, and mapping ToIs from UAV cameras in GPS-denied environments.

2 METHODOLOGY

2.1 Transformation of Pixel Coordinates to Real-World Positions

In this research, the process of mapping UAV-captured scenes is carried out in two phases: (i) identifying ToIs in aerial views and locating them within the image (perspective) coordinate system (a.k.a., pixel coordinates), followed by (ii) projecting ToI positions to the real-world (orthogonal) coordinate system (a.k.a., real-world positions). To obtain the real-world orthogonal position of a point from its perspective pixel coordinates, four reference points (where any three points are non-collinear) with known pixel coordinates and corresponding real-world positions are needed.

Figure. 1 shows an example, where pixel coordinates of the reference points are (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4), and corresponding real-world position are (x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), and (x'_4, y'_4), respectively. Given M, the real-world position (x'_5, y'_5) of any point (i.e., ground ToI) can be calculated from its pixel coordinates (x_5, y_5) using Equations 6 and 7. In Equation 6, (x''_5, y''_5, w) represents the point's position in a homogenous coordinate system where $1/w$ is the distance of the point from camera (i.e., $w = 0$ represents that point is at infinite distance).

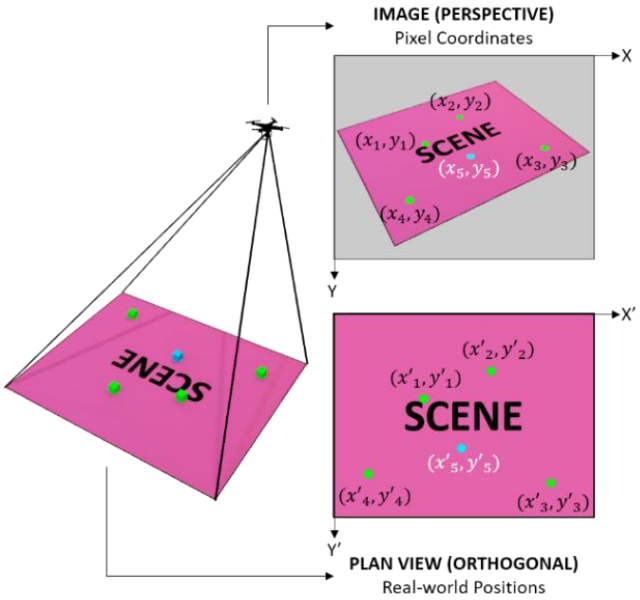

Figure 1. Perspective to orthogonal transformation.

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_4 \\ y_4 \\ 1 \end{bmatrix} \#(1)$$

$$A = \begin{bmatrix} a_1 . x_1 & a_2 . x_2 & a_3 . x_3 \\ a_1 . y_1 & a_2 . y_2 & a_3 . y_3 \\ a_1 & a_2 & a_3 \end{bmatrix} \#(2)$$

$$\begin{bmatrix} x'_1 & x'_2 & x'_3 \\ y'_1 & y'_2 & y'_3 \\ 1 & 1 & 1 \end{bmatrix} . \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} x'_4 \\ y'_4 \\ 1 \end{bmatrix} \#(3)$$

$$B = \begin{bmatrix} b_1 . x'_1 & b_2 . x'_2 & b_3 . x'_3 \\ b_1 . y'_1 & b_2 . y'_2 & b_3 . y'_3 \\ b_1 & b_2 & b_3 \end{bmatrix} \#(4)$$

$$M = B . A^{-1} \#(5)$$

$$\begin{bmatrix} x''_5 \\ y''_5 \\ w \end{bmatrix} = M . \begin{bmatrix} x_5 \\ y_5 \\ 1 \end{bmatrix} \#(6)$$

$$x'_5 = \frac{x''_5}{w}, y'_5 = \frac{y''_5}{w} \#(7)$$

2.2 Object Detection Framework

CNN models, trained on relevant datasets, can effectively detect objects based on visual features and output the pixel coordinates of detected objects. With proper training data, UAVs equipped with cameras can also detect objects on the ground (e.g., parking lots, people, landmarks, trees, buildings) from the perspective view. These objects can serve as reference points (Figure 1). The pixel coordinates of any ToI is also produced by the same CNN model. However, there is a trade-off between speed and accuracy of different CNN architectures, as shown in Table 1. In this research, the goal is to process images in real-time with high accuracy. Therefore, RetinaNet-50-500 (4) is selected since its focal loss function emphasizes on learning hard examples, leading to high accuracy without compromising real-time speed.

Table 1. Average precision (AP) vs. time on COCO test-dev dataset.

CNN model	AP (%)	Time (milliseconds)
YOLOv2 (Redmon et al. 2016)	21.6	25
SSD321 (Liu et al. 2016)	28.0	61
R-FCN (Dai et al. 2016)	29.9	85
RetinaNet-50-500 (Lin et al. 2017)	32.5	73
DSSD513 (Liu et al. 2016)	33.2	156
FRCN (Dai et al. 2016)	36.2	172
RetinaNet-101-800 (Lin et al. 2017)	37.8	198

2.3 Viewpoint Transformation

Once the CNN model identifies the pixel coordinates of the reference points and ToIs, the next step is to obtain each ToI's real-world position given the real-world positions of the reference points. The real-world positions of the reference points can be obtained either from prior-knowledge (e.g., using on-site measurements) or from a plan-view image (using the ratio between detected size of an object and its known actual physical size). In this Section, the problem of calculating the real-world position and orthogonal mapping of ToIs is approached from two angles: (i) projection from perspective to orthogonal based on reference objects' coordinates (PROC), and (ii) projection from perspective to orthogonal based on reference objects' size (PROS).

Figure 2 illustrates the workflows of PROC and PROS approaches. In this Figure, four traffic cones are used as reference objects, and the goal is to calculate the orthogonal position of a moving person (i.e., ToI) from UAV-captured perspective views. In the PROC approach, first, Model-P is trained on a perspective video (PV1) and tested on another perspective video (PV2) to predict the pixel coordinates of the reference objects and ToI in each video frame. Next, from the known real-world positions of the reference objects, the real-world position of the ToI (i.e., x'_5, y'_5) is calculated in PV2 using Equations 1 through 7. While the PROC approach relies on the real-world positions of reference objects, the PROS approach uses the size of reference objects. The pixel coordinates of reference objects and ToI are obtained as before by using Model-P. However, the reference objects' real-world positions are calculated differently. In essence, another CNN model, Model-O, is trained on an orthogonal video (OV1) and tested on

another orthogonal video (OV2) to predict the boxes of the reference objects. Since in the PROS approach, the actual sizes of reference objects are known, the ratio between the sizes of the predicted boxes in pixel and their real sizes is used to transform the pixel coordinates of these objects to real-world positions. Next, the real-world position of the ToI is calculated in PV2 using Equations 1 through 7.

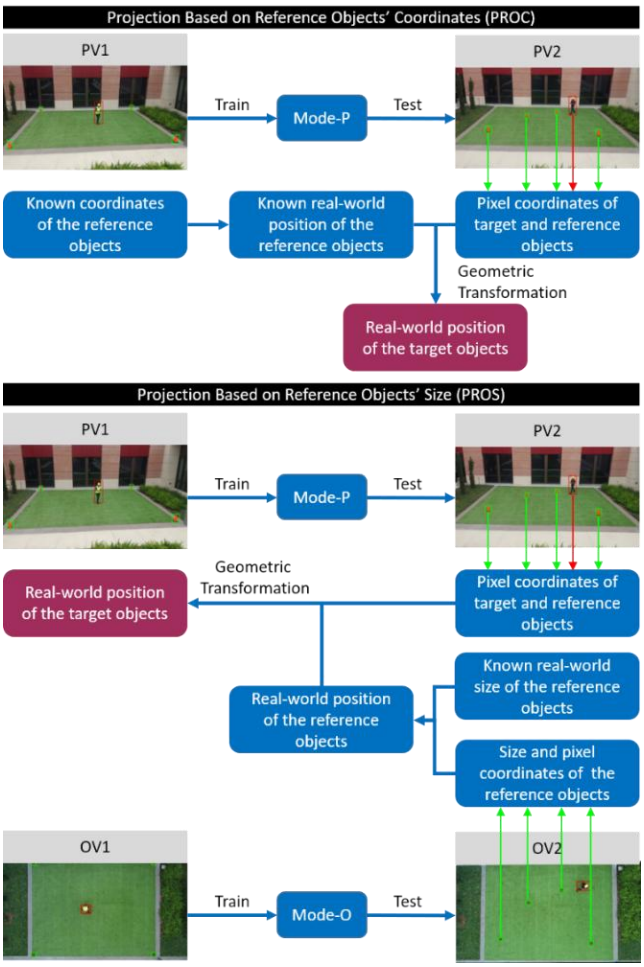

Figure 2. Reference objects in perspective and orthogonal views are used to calculate the real-world position of a walking person.

3 Applications

The large volume of available geocoded data (e.g., 3D building models with geographic information) and crowd-sourced video data, especially in urban areas, enables the training of CNN models to effectively recognize natural and manmade landmarks as ground reference objects. In some cases, where a sufficient number of references objects does not exist, manually placed markers (e.g., GCPs) can be also used. Nonetheless, with CNN models capable of recognizing multiple reference objects in perspective views and given information about the real-world positions of these objects, the PROC approach can be used to map multiple ToIs (both stationary and moving) in real-time.

In some scenarios, however, it might not be possible to obtain accurate real-world positions of the reference objects beforehand. In these cases, knowing the physical size of the reference objects, the PROS approach can be used to produce orthogonal maps of

ToIs. This approach requires (at least) one orthogonal image (e.g., from a second UAV camera, an airplane, or a passing satellite looking down vertically) to calculate the real-world positions of the reference objects. The PROS approach is therefore more generalizable compared to the PROC approach since most of the prior knowledge in PROS can be obtained beforehand or with little dependency on the actual site. Table 2 is a summary of potential applications of both approaches. Examples listed in this Table can be implemented using the PROC approach if there are adequate reference objects (4 or more). Examples marked with (*) indicate that the PROS approach is more suitable due to the potential of insufficient data about reference object positions (e.g., moving reference objects).

Table 2. Potential applications of PROC and PROS mapping methods.

Domain	Reference Points (Landmarks)	Potential TOIs	Application
Disaster management	Parking lot, intersection, building	People, car, debris, damaged roof	Search and rescue, damage estimate
Agriculture*	Tractor, building, road post	Livestock, crops, grass, vegetation	Growth estimate, fleet control
Forestry*	Cliff, river, equipment	Trees, animals	Fire control, animal migration
Marine systems*	Lighthouse, pier, building, vessels	Marine animals, corals, vessels, oil spill	Navigation, marine ecology
Construction management	Building element (wall, column), equipment	Material stocks, workers, equipment	Safety, work monitoring, productivity estimate
Transport systems	Road sign, toll plaza, control tower	Cars, trucks, trailers, planes	Traffic monitoring and control
Urban management	Parking lot, intersection, building	Buildings, trees, people	Urban planning, land survey
Military defence*	Military post, tent, installation	Adversary	Surveillance, target inspection

4 EXPERIMENTS

To train and test CNN models that can automatically detect reference objects and ToIs, two experiments are conducted. Experiment 1 involves collecting perspective video PV1 (Figure 3) and orthogonal video OV1 (Figure 4), which provide the training data for the perspective model (Model-P) and orthogonal model (Model-O). Experiment 2 contains two videos, PV2 (Figure 5) and OV2 (Figure 6), that serve as testing data.

4.1 Data Collection and Description

The objective of both experiments is to project a moving ToI captured in UAV's perspective view into real-world (orthogonal) coordinates, using four reference points. Therefore, each experiment contains a continuously moving person (ToI) and four traffic cones (reference objects). The cones and the person are located on a turf (333 in × 426 in).

To obtain ground-truth information, one drone (Parrot Anafi) records the scene from an aerial angle pointing the camera vertically downward (videos OV1 and OV2). Another drone (Parrot Bebop 2) records the same scene from a perspective angle and an arbitrary altitude (videos PV1 and PV2). The cones' positions in Experiment 1 (PV1 and OV1) are different from Experiment 2 (PV2 and OV2). Also, the person changes outfit between the two experiments. With the upper left corner of the turf designated as the origin of the real-world coordinate system, the real-world positions of the four cones are shown in Figure 3 through 6. For example, cones in Experiment 1 are placed at coordinates (0, 0), (0, 333), (426, 0), and (426, 333), whereas in Experiment 2, they are placed at coordinates (56, 263), (148, 138), (258, 95), and (356, 277). The timestamps of PV2 and OV2 videos are carefully synchronized, leaving a total of 4,149 frames for projection. To train and test the CNN models for detection of cones and person, all collected videos are manually annotated with DarkLabel (DarkLabel 2020), frame by frame, as shown in Figure 3 through 6.

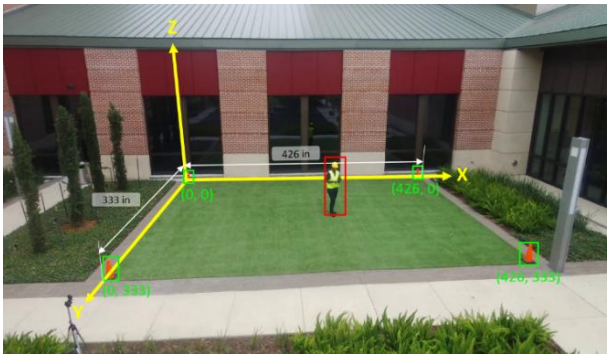

Figure 3. Perspective video 1 (PV1) annotation sample and setup.

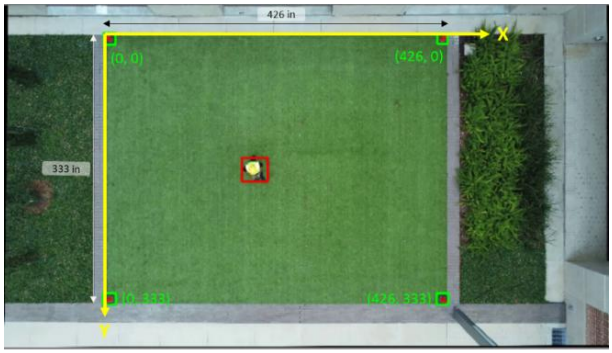

Figure 4. Orthogonal video 1 (OV1) annotation sample and setup.

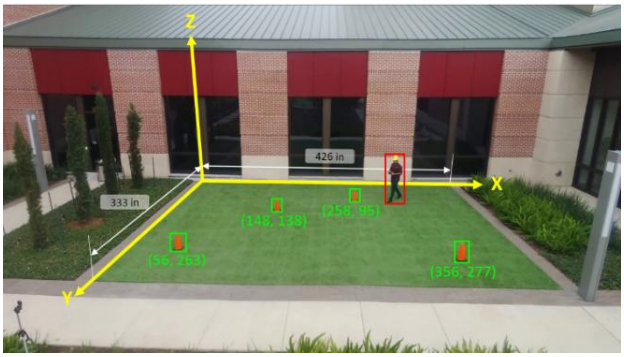

Figure 5. Perspective video 2 (PV2) annotation sample and setup.

Figure 6. Orthogonal video 2 (OV2) annotation sample and setup.

In these Figures, green rectangles represent cones, and red rectangle represents person. Description of the experimental data is shown in Table 3. The ground-truth real-world positions of the person are retrieved from the manually annotated OV2 video. All frames within PV1 and OV1 are used for training while the frames in PV2 and OV2 are used for testing. For the training of Model-P and Model-O, the learning rate is set to 10-4 and batch size is set as 4.

Table 3. Data statistics for experiments 1 and 2.

	Experiment 1		Experiment 2	
	PV1	OV1	PV2	OV2
Total frames	8,873	9,915	5,261	5,814
Person instances	8,813	8,223	4,492	4,773
Cone instances	33,362	32,892	19,650	19,092

4.2 Projection Based on Reference Objects' Coordinates (PROC)

Model-P is responsible for detecting the pixel coordinates of cones and person in PV2. Figure 7 shows an example of detection of cones (yellow boxes) and person (white box). All detected boxes are associated with a confidence level predicted by the model. The four cones and one person that are detected with the highest confidence levels are selected for further analyses.

Next, pixel coordinates, i.e., the center point of the bottom edge of each detection box (marked with dots in Figure 7) are determined as input for projection calculation. From the calculated pixel coordinates of four cones and their corresponding real-world positions, the person's real-world position is calculated (using Equations 1 through 7), and projected on the orthogonal map shown in Figure 8. Also, the annotated person's position in OV2 is projected on the same coordinate system which serve as ground-truth. As expected, there may be situations when the PROC approach fails to project, e.g., when Model-P cannot detect at least four cones in the frame, because they are either not within the camera view or obstructed by other objects. In this work, such frames are removed from analysis. The number of skipped frames, presented as the percentage of total number of frames, is termed frame loss (Equation 8) which describes how efficiently the model utilizes the input video data. Moreover, due to the noise in data and/or detection error, sometimes the detected boxes of the same object (e.g., cone) in two consecutive video frames could appear in two considerably different locations. To

identify such statistical outliers, X- and Y- coordinates of each objects are treated as a timeseries data. Given, a timeseries data of the coordinates of an object, $C = \{c_t; t = 1,2, \dots, n\}$, at frame $t = n$, we consider 15 previous consecutive data points, $C' = \{c_t, c_{t-1}, \dots, c_{t-14}\}$, and calculate its mean c'_{mean} and standard deviation c'_{STD}. If the current reading c_t is a statistical anomaly, defined as $c_t - c'_{mean} \geq k * c'_{STD}$ (two cases with k = 2 and 3) we consider the detection as an outlier. After removing the outlier predictions, the average projection error is recalculated.

$$frame\ loss\ (FL)\ \% = \frac{\#\ of\ skipped\ frames}{total\ \#\ of\ frames} * 100 \#(8)$$

Figure 7. Example of detections of person and cones.

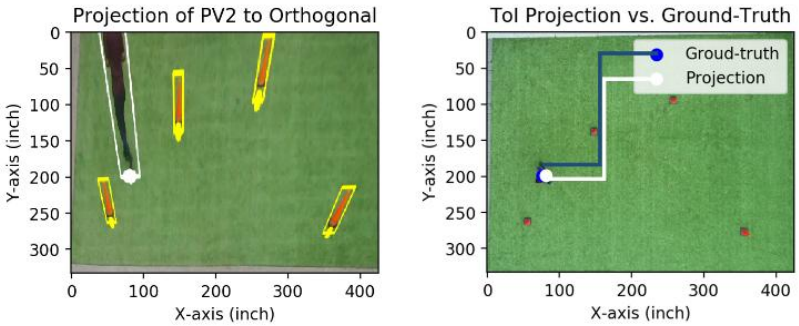

Figure 8. Perspective to orthogonal projection.

4.3 Projection Based on Reference Objects' Size (PROS)

As described earlier, the difference between PROC and PROS approaches is in the method for acquiring the reference objects' real-world positions. In the PROS approach, it is assumed that the size of an object in pixel units in the orthogonal image will represent a constant physical size throughout the entire video. Based on this assumption, from the known physical size of the target object, its actual position (in inches) can be calculated solely based on the model's predictions in the orthogonal view. In the experiments conducted, the actual dimensions of the traffic cone are known to be 10 inches (both length and width). However, due to the noise in data and/or human error, the annotated training boxes are slightly bigger than the actual size of the cones. From the training data (OV1), the width of the detection boxes for cones is calculated to be 14.56 inches on average. Therefore, it is assumed that the detected boxes for the same cones in the test data (OV2) would also have the same average physical size. During the

test, Model-O is applied to the first 30 frames (i.e., 1 second) of OV2 to obtain the sizes and coordinates of the boxes representing cones in pixel units. Next, knowing that the size of the boxes are 14.56 inches in real-world, coordinates are scaled accordingly to obtain the real-world positions of the boxes. The latter part (orthogonal projection) of the PROS approach is similar to the previously described PROC approach (Section 4.2).

5 Results

5.1 RetinaNet Model Testing

Model-P and Model-O are tested on PV2 and OV2 to evaluate their performance, as shown in Table 4. Overall, Model-O produces 51% mAP which is worse than 97% for Model-P. One reason for this disparity is that from orthogonal view (in Model-O), the person and cones appear less distinctive compared to perspective view (in Model-P). For individual classes, Model-P detects class person with 99.21% average precision (AP); however, Model-O produces only 45.89% AP. Similarly, for class cone, Model-P produces 96.11% detection AP, compared to 51.97% AP for Model-O.

Table 4. Comparison of Model-P and Model-O performance.

Model	mAP	AP (Cone)	AP (Person)
P	97.66%	96.11%	99.21%
O	48.93%	51.97%	45.89%

5.2 Projection Results

Projection results obtained from PROC and PROS approaches are summarized in Table 5, and illustrated in Figure 9. For both PROC and PROS, the remaining frames after removing non-projectable frames (because the Model-P detected less than four cones or one person while tested on PV2) is 3,735 with 9.98% frame loss for both projections. The Euclidian distance (in inches) between the person's real-world and projected positions is reported as projection error in each frame. The average projection error (APE) is calculated as the mean of frame-by-frame errors in all projectable frames. As shown in Table 5, the PROS approach achieves an APE value of 15.39 inches, which is slightly better than PROC's APE of 17.18 inches. Next, the projected person's real-world X- and Y- coordinates are compared with the corresponding ground-truth coordinates, frame by frame, and the errors are divided by the total length along each axis (i.e., 426 inches for X-axis and 333 inches for Y-axis) to present the error as a percentage. The errors in PROC approach are 9.52 inch (2.23%) along X-axis and 11.79 (3.54%) along Y-axis, which is slightly outperformed by PROS approach with errors of 8.62 inches (2.02%) along X-axis and 10.41 inches (3.12%) along Y-axis.

Furthermore, two scenarios are considered for removing the outliers: k = 2 and k = 3 (Section 4.2). For k = 3, after removing outliers, a total of 1,493 (FL = 64.02%) and 1,488 (FL = 64.14%) frames remain in PROC and PROS, respectively, and the overall APE is improved to 13.86 inches (for PROC) and 11.86 inches (for PROS). When removing outliers with k = 2, calculated APEs (overall, X-, and Y-) for both approaches are greater than the case of k = 3, but smaller FL (26.63% for PROC and 26.71% for PROS) is achieved. Errors along X- and Y-axis for individual frames are documented in Figure 9. In general, it is observed that removing outliers leads to higher frame loss but smaller

APE in both approaches. Nonetheless, for all cases, the average error along any dimension is less than 4% which indicates the robustness of the proposed methods.

Table 5. Test results of PROC and PROS mapping methods.

	Method	Direct Projection	Outlier (k = 2)	Outlier (k = 3)
Remained Frame # (FL %)	PROC	3,735 (9.98%)	3,044 (26.63%)	1,493 (64.02%)
	PROS	3,735 (9.98%)	3,041 (26.71%)	1,488 (64.14%)
Overall APE (inch)	PROC	17.18	14.35	13.86
	PROS	15.39	12.31	11.85
X-axis APE (inch and %)	PROC	9.52 (2.23%)	7.11 (1.66%)	6.53 (1.53%)
	PROS	8.62 (2.02%)	5.89 (1.38%)	5.37 (1.26%)
Y-axis APE (inch and %)	PROC	11.79 (3.54%)	10.89 (3.27%)	10.87 (3.26%)
	PROS	10.41 (3.12%)	9.55 (2.86%)	9.50 (2.85%)

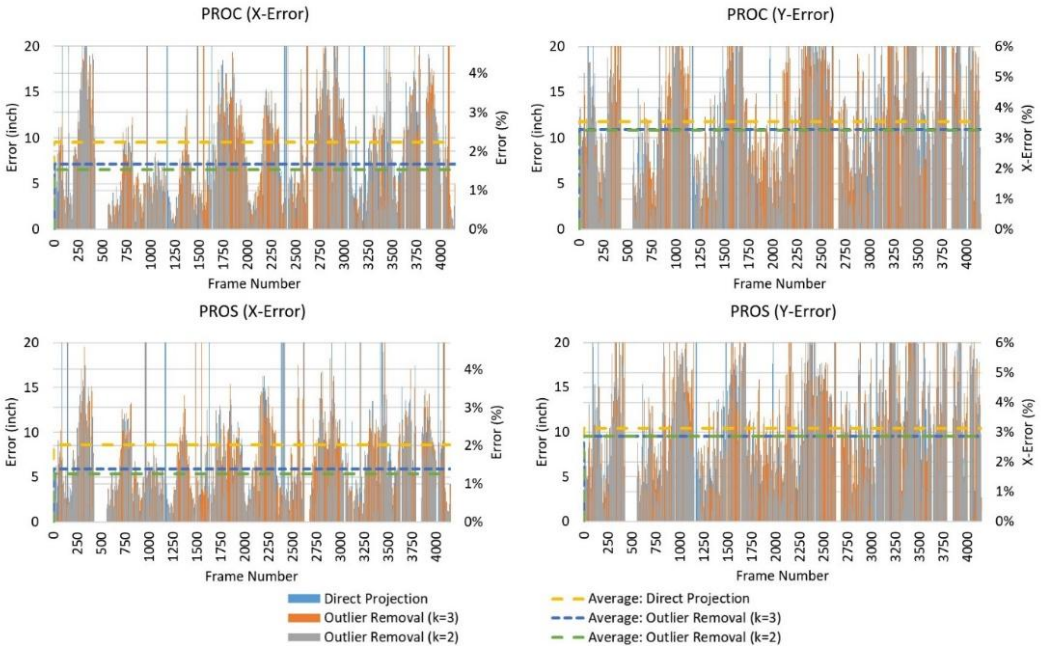

Figure 9. Frame-by-frame projection error in X- and Y- axes for PROC and PROS methods.

6 CONCLUSION

The research presented in this paper was motivated by the need for robust object detection and mapping from UAV-captured scenes in GPS-denied environments. By proposing a vision-based UAV localization and mapping framework, this work provided an alternative for GPS-denied target projection based on reference objects' coordinates or sizes. In addition, potential applications of GPS-denied UAV mapping in various domains were discussed. Separate experiments were conducted to produce training dataset (PV1 and OV2) and testing dataset (PV2 and OV2) for CNN models, and two

mapping (perspective to orthogonal projection) approaches (namely PROC and PROS) were designed and validated. The PROC approach used the real-world coordinates of four reference points (e.g., natural or manmade objects on the ground) to project ToIs onto an orthogonal map. The PROS approach, on the other hand, was based on information about the physical size of reference objects (e.g., cars, land parcels, buildings). Based on this distinction, the PROS approach is deemed more suitable in scenarios with limited prior knowledge. Both approaches achieved APEs as small as 11.85 inches (PROS) and 13.86 inches (PROC) (1.26% and 2.85% along X- and Y- axes in PROS, compared to 1.53% and 3.26% along X- and Y- axes in PROC) which makes them a feasible solution for real-time localization and mapping of ToIs solely from RGB camera's inputs.

The presented work poses some limitations. For example, the designed methods utilize video input from only one UAV for detection and mapping, and work best under adequate illumination (RGB camera provides little information in dark or not well-lit scenes). Also, while both mapping methods require reference objects on the ground, such objects may not be readily available in some circumstances such as UAV missions in unknown territories or remote locations. Future research will investigate solutions to these and other problems including the possibility of using other visual bandwidths (e.g., thermal imagery) as well as creating ad-hoc reference points using multiple cooperative UAVs.

7 Acknowledgments

The authors would like to acknowledge the Texas A&M University's High Performance Research Computing (HPRC) for providing necessary computational resources for model training. The authors are also thankful to Mr. Jerome Bouvard, Director of Strategic Partnerships at Parrot Inc. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily represent the views of HPRC, Parrot Inc., or the individual named above.

8 References

Adams, S.M., Friedland, C.J. (2011). A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. In: 9th International Workshop on Remote Sensing for Disaster Response. Stanford, CA: pp. 15-16.

Asher, M.S., Stafford, S.J., Bamberger, R.J., Rogers, A.Q., Scheidt, D., Chalmers, R. (2011). Radionavigation alternatives for US Army Ground Forces in GPS denied environments. International Technical Meeting of The Institute of Navigation. San Diego, CA: pp. 24-26.

Bachrach, A., Prentice, S., He, R., Henry, P., Huang, A.S., Krainin, M., Maturana, D., Fox, D., Roy, N. (2012). Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. The International Journal of Robotics Research 31(11), 1320–43.

Balamurugan, G., Valarmathi, J., Naidu, V. (2016). Survey on UAV navigation in GPS denied environments. In: International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, Paralakhemundi, India. pp. 198-204.

Carter, J., Schmid, K., Waters, K., Betzhold L., Hadley, B., Mataosky, R., Halleran, J. (2012). Lidar 101: An introduction to lidar technology, data, and applications. National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center. Charleston, SC.

Chen, J., Qiu, J., Ahn, C. (2017). Construction worker's awkward posture recognition through supervised motion tensor decomposition. Automation in Construction 77, 67-81.

Cheng, G., Zhou, P., Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 54(12), 7405-15.

Cunliffe, A.M., Brazier, R.E., Anderson, K. (2016). Ultra-fine grain landscape-scale quantification of dryland vegetation structure with drone-acquired structure-from-motion photogrammetry. Remote Sensing of Environment 183, 129–43.

Dai, J., Li, Y., He, K., Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379-387. Barcelona, Spain.

DarkLabel1.3 (image labelling and annotation tool). (2017). DarkLabel, [online] Available at: https://darkpgmr.tistory.com/16, [accessed January 16, 2020].

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. International Journal of Computer Vision 88(2), 303-338.

Friedel, M.J., Buscema, M., Vicente, L.E., Iwashita, F., Koga-Vicente, A. (2018). Mapping fractional landscape soils and vegetation components from Hyperion satellite imagery using an unsupervised machine-learning workflow. International Journal of Digital Earth 11(7), 670–90.

Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH: IEEE, pp. 580-587.

Girshick, R. (2015). Fast R-CNN. In: IEEE International Conference on Computer Vision. Boston, MA: IEEE, pp. 1440-1448.

Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J. (2014). Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Transactions on Geoscience and Remote Sensing 53(6), 3325-37.

Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM 60(6), 84-90.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In: IEEE International Conference on Computer Vision. Venice, Italy: IEEE, pp. 2980-2988.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L. (2014). Microsoft COCO: Common objects in context. European Conference on Computer Vision. Zurich, Switzerland: Springer, pp. 740-755.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A. C. (2016). SSD: Single shot multibox detector. In: European Conference on Computer Vision. Amsterdam, The Netherlands: Springer, pp. 21-37.

NASA. (2020). Earthdata, [online] Available at: https://earthdata.nasa.gov, [accessed January 6, 2020].

Pestana, J., Sanchez-Lopez, J.L., Campoy, P., Saripalli, S. (2013). Vision based GPS-denied object tracking and following for unmanned aerial vehicles. In: IEEE International

Symposium on Safety, Security, and Rescue Robotics (SSRR). Linkoping, Sweden: IEEE, pp. 1-6.

Rajeev, S., Wan, Q., Yau, K., Panetta, K., Agaian, S.S. (2019). Augmented reality-based vision-aid indoor navigation system in GPS denied environment. In: Mobile Multimedia/Image Processing, Security, and Applications, Baltimore, MD: p. 109930P.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV: IEEE, pp. 779-788.

Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH: IEEE, pp. 7263-7271.

Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:180402767.

Ren, S., He, K., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1137-1149.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3), 211-252.

Scaramuzza, D., Achtelik, M.C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M.W., Chli, M., Chatzichristofis, S., Kneip, L., Gurdan, D. (2014). Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. IEEE Robotics and Automation Magazine 21(3), 26–40.

Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556.

Suh, J., Choi, Y. (2017). Mapping hazardous mining-induced sinkhole subsidence using unmanned aerial vehicle (drone) photogrammetry. Environmental Earth Sciences 76(4), 144.

Tang, J., Deng, C., Huang, G.B., Zhao, B. (2014). Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. IEEE Transactions on Geoscience and Remote Sensing 53(3), 1174-85.

Ventura, D., Bruno, M., Lasinio, G.J., Belluscio, A., Ardizzone, G. (2016). A low-cost drone based application for identifying and mapping of coastal fish nursery grounds. Estuarine, Coastal and Shelf Science 171, 85–98.

Wang, C., Wang, T., Liang, J., Chen, Y., Zhang, Y., Wang, C. (2012). Monocular visual SLAM for small UAVs in GPS-denied environments. In: IEEE International Conference on Robotics and Biomimetics (ROBIO). Guangzhou, China: IEEE, pp. 896-901.