Landmark Localization for Drone Aerial Mapping Using GPS and Sparse Point Cloud for Photogrammetry Pipeline Automation

Byeong Yeon Ryu, Won Nyoung Park, Donghwi Jung and Seong-Woo Kim

2022 International Conference on Electronics, Information, and Communication (ICEIC) | 978-1-6654-0934-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICEIC54506.2022.9748847

***Abstract*—Ground Control Point (GCP) rectification serves an important role in assuring the absolute and relative accuracy for drone photogrammetry generated data, yet the process of identifying and marking GCPs is still handled manually, hindering the scalability of the source photo processing pipeline. In this paper, we propose a method to accurately detect and automatically mark GCPs from aerial images using deep learning and photogrammetry-generated sparse point cloud to expedite the sourcephoto processing pipeline. Using SOTA Object Detection and Image Classification models, RetinaNet and Inception-ResNet-V2, we first accurately detect Ground Control Points from the collected source photos. The detected targets are then filtered and labeled by backward-projecting the detected image $x - y$ coordinates to 3D sparse point cloud, and comparing the 3D coordinate with the surveyed GCP. The GPS matching process on the sparse point cloud assures sub-centimeter level accuracy errors compared to traditional human rectification while exceeding the performance of other commercially available GCP detection methods.**

I. Introduction

Photogrammetry, the science of obtaining geospatial information from processing multiple overlapping images, has been an existing concept for many decades. Recently, the advancement of drone technology, the equipment of sensors, and high resolution cameras on drones have allowed for more convenient implementations of drone photogrammetry across many industries such as construction, agriculture, mining, disaster relief and more. In drone photogrammetry, drones are utilized to take aerial photos with relatively large overlapping regions, usually of around 75 percent. The computer recognizes patterns and features within each image and matches features from an image to another. Using the same feature from two or more different images, we are able to triangulate and determine the three dimensional location of the object. Likewise, through drone photogrammetry, we are able to initially construct a 3D point cloud and ultimately generate digital surface models, 2D/3D orthomosaics, and 3D mesh models to virtualize and digitize a target area.

GCPs (Ground Control Points) are used during the sourcephoto processing stage for improving the accuracy of drone photogrammetry data by establishing the relationship between a raw sourcephoto and the coordinate system [1]. When a drone takes its photos, the GPS location of the drone is recorded and stored as image data, which is later utilized during the initial source photo processing stage. However, due to the inherent inaccuracy of the drone's GPS and the internal distortion caused during the image stitching process, data accuracy is further reinforced by using GCPs. In specific, the 2D location within the photo is associated with the 3D location within the coordinate system, allowing the rectification process to reference the position using interpolation techniques.

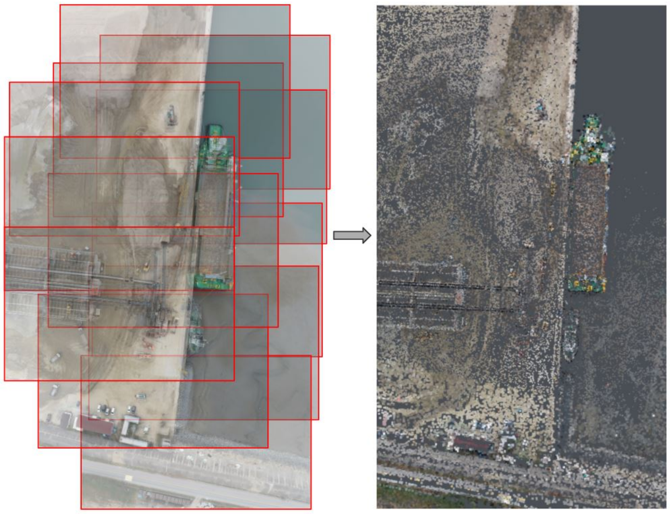

Fig. 1. Using overlapping aerial images captured by the drone, we create a three dimensional sparse point cloud during the first processing step.

During a pre-setup process, GCPs are installed evenly throughout a jobsite prior to the drone flight. A surveyor then uses an accurate GNSS receiver to accurately pinpoint the center of the GCP to collect the precise Latitude, Longitude, and Altitude coordinates. These coordinates are utilized for each GCP label for rectification during the sourcephoto processing stage. Then, the following sourcephoto processing pipeline is used to produce drone photogrammetry-generated data. Aerial sourcephotos are first uploaded along with surveyed GCP coordinates. The sourcephoto processing pipeline consists of the following four phases: 1) Sparse point cloud build 2) GCP rectification 3) Point cloud, Mesh data build, and 4) Digital surface model, Orthomosaic build.

B. Ryu is with Columbia University, NewYork, USA (b.ryu@columbia.edu)

W. Park is with Angelswing Inc., Seoul, South Korea (peter@angelswing.io)

D. Jung and S. Kim are with the Smart City major in the Department of Civil and Environmental Engineering, and Integrated Major in Smart City Global Convergence, Seoul National University, South Korea. (donghwi-jung@snu.ac.kr, snwoo@snu.ac.kr)

Research conducted at Angelswing Inc. in coordination w/ SNU ARIL Labs, and supported in part by Korean Ministry of Land, Infrastructure and Transport as the Innovative Talent Education Program for Smart City.

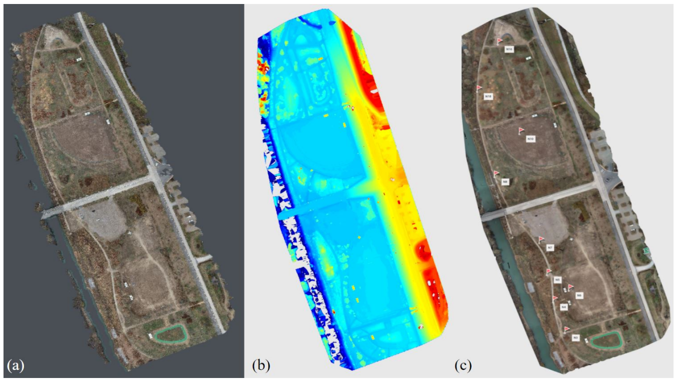

Fig. 2. Visualized results of the photogrammetry generated data in the forms of (a) Dense Point Cloud, (b) Digital Surface Model, and (c) Orthomosaic.

During the first phase of the pipeline, as illustrated in Figure 1, we process the sourcephotos to first identify the key points within the images and match the common tie points together to create a sparse point cloud. Here, the sparse cloud represents a rough estimation of the 3D model based on the camera position and orientations, thereby lacking relative and absolute accuracy. Poor absolute accuracy is observed especially for non-RTK drones, as the sparse cloud is built upon less accurate GPS data from the drone.

In order to improve the accuracy confidence, we proceed to the GCP rectification stage of the pipeline, which involves marking GCPs within the sourcephotos and referencing the exact survey points in the specified coordinate reference system. Although most of the sourcephoto processing pipeline is fully automated, the labor intensive GCP rectification step is still dealt manually, constraining the scalability, accuracy, and speed of the process.

For this reason, this paper focuses on automating the GCP rectification step by precisely locating GCPs within images and matching surveyed labels, and automatically rectifying GCPs for the remaining photogrammetry processes.

II. Related Works

Many studies have undertaken the assessment of photogrammetric mapping quality in order to obtain most accurate results by exploring numerous parameters including how different numbers and location variations of ground control points affect the accuracy of photogrammetry-generated data. In [2], the authors explore different variations of ground control point numbers and distributions to measure the role of GCPs. However, the process of identifying and marking GCPs from the drone images is dealt manually by placing cross markers on the center of GCP markers which only consist 10px$\times$10px in the images with an average GSD of 6.86 cm.

There exists a variety of efforts to automatically detect signalized targets from images using computer vision. [3] demonstrates the existing concept of auto calibrating cameras with the identification and localization of targets, and recently, fiducial markers have also been used for supervising autonomous UAV landing as studied in [4], [5]. However, the identification of fiducial markers using computer vision is usually conducted under strictly controlled environments where the camera is positioned relatively close to the target and the marker is clearly identifiable within the image. In experiments conducted by [6], the maximum detection range of AprilTags only extended 7.0 meters for a 0.167 meter wide target.

On the contrary, in aerial images taken from high altitudes where there are an abundance of features, distortions, obstacles, and varying lighting conditions, marker detection lacks the performance and reliability as the detectors rely heavily on extracting polygons from the contours of the image [7], [8]. Moreover, though many GCPs take the form of checkerboard mat boards, natural landmarks such as manholes are also often utilized when artificial marker installation is difficult or limited around and within target sites. Therefore, traditional fiducial marker detection is unsuitable for GCP marker detection.

In [9], the paper suggests a standardized L-shaped painted marker for detecting GCPs using edge detection and convolutional neural networks. Though the study relays relatively high detection accuracy, the proposal of a fixed GCP shape hinders the flexible placement of various GCP markers. Furthermore, it is unclear how the detection results are used for matching a GCP label and rectifying the target markers within the images.

While a commercially available GCP detection system is presented such as [10], the details of how the method is implemented is not revealed. Using the performance of this commercial system as baseline, we present a more comprehensive approach, utilizing deep learning and photogrammetry generated sparse point cloud to precisely identify and localize ground control points for automatic rectification.

III. Methods

Our proposed GCP detection and auto rectification method consists of three main modules which are integrated closely into the sourcephoto processing pipeline as illustrated in Fig. 3. In the figure, the first GCP detection step in (a) consists of searching through each source image to precisely detect ground control point markers. The detection phase occurs in parallel with the sparse point cloud build which is processed using Angelswing's sourcephoto processing pipeline [11]. Once the sparse point cloud build is complete, the 2D detection results are converted and projected to a 3D location in the point cloud. This location is then compared with and matched to a surveyed GCP label. Lastly, the automatic rectification sequence helps optimize the sparse point cloud to produce a rectified sparse point cloud.

A. Experimental Setup

One of the many challenges for detecting GCPs from aerial images is the inherent difficulty of small object detection from high resolution images with complex backgrounds. In order to train our deep learning models for high accuracy detection, we first preprocess substantial drone image datasets, which have been collected over time mainly from target construction sites.

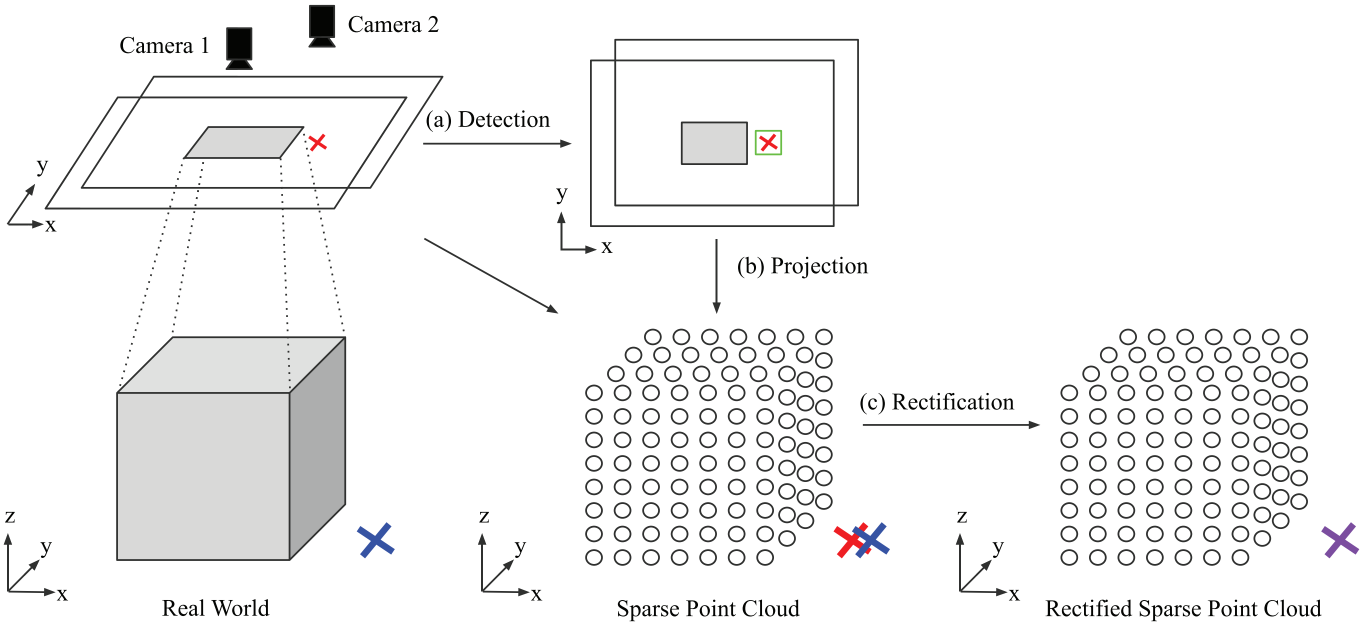

Fig. 3. The proposed automatic GCP rectification pipeline is composed of three main modules: (a) GCP object detection from images, (b) Coordinate conversion and projection from local image 2D to global sparse point cloud 3D (c) Surveyed GCP label matching for auto rectification.

TABLE I
TEST SITE SPECIFICATIONS

Test Site	Total GCPs	GCP Type	Total Area	Images	GSD	GPS
A	9	Checkerboard	0.132km^2	638	1.50cm/px	with RTK
B	8	Checkerboard	0.052km^2	247	1.38cm/px	with RTK
C	12	Checkerboard + Manhole	0.054km^2	548	3.23cm/px	without RTK
D	15	Checkerboard	0.155km^2	414	3.01cm/px	with RTK
E	6	Checkerboard + Manhole	0.201km^2	337	1.83cm/px	without RTK

Aerial sourcephotos were collected using a quadroter drone equipped with 20 megapixel camera. Images were taken with the mechanical shutter and a minimum overlapping ratio of 0.75. The drone's flight altitude varied depending on the job site, ranging from 30 m to 150 m above the point of takeoff.

We divide our dataset into training and validation sets by a 8:2 ratio prior to training. Due to images taken with large overlapping regions, the same GCP appears in multiple images from different angles and positions. Therefore, we categorize data sets by surveyed sites to prevent identical GCPs from being used for both training and validation. Additionally, five test sites documented in TABLE I were chosen for evaluating the performance of the deep learning models and auto rectification accuracy. Data sets used for testing were not used for training and validation.

GCPs are labeled using an open source tool, LabelImg. [12] Our dataset consists a wide variety of manholes and checkerboard GCPs including those that are painted on the surface. To process the source photos prior to training, we randomly crop the images from the original dimensions to smaller segments in order to prevent GPU memory overflow with the original high resolution image data during the model training stage. Unlike the more convenient method of resizing our images, since our target is already very minuscule compared to the image dimension, we approach the problem by cropping and then re-tiling the images upon inference. [13]

Construction job sites in an urban environment make object detection particularly challenging due to high density of tall buildings and objects. Though all photos were taken during the day, the abundance of shadows make our detection more difficult. Objects like building windows, roofs, sidewalk tiles, cars, and construction equipment easily resemble checkerboard ground control point markers and manholes. Therefore, we apply additional brightness augmentations of up to 50 percent in order to increase data set diversity. Random affine transformation with rotation is also applied to our images to accommodate for images taken from different camera orientations.

B. Deep Learning Pipeline

1) RetinaNet Training: We first utilize a deep Convolutional Neural Network, RetinaNet with a ResNet-152 backbone pretrained on COCO datasets. We also experimented with ResNet-50, though the shallower network returned slightly

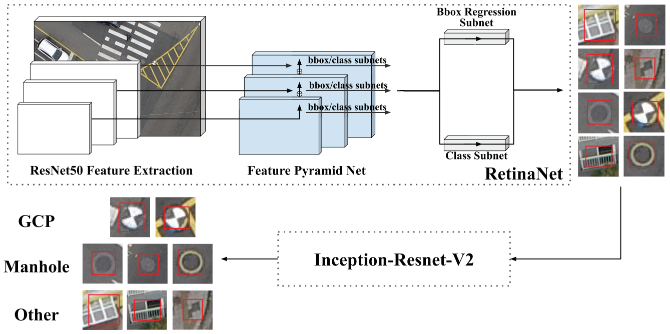

Fig. 4. Illustration of the deep learning pipeline, which consists of an initial object detection sequence using RetinaNet and an additional layer of Inception-ResNet-V2 for reinforced GCP classification results.

inferior mAP values upon validation. RetinaNet's introduction to focal loss for dense object detections proves most effective for small object detection with varying scales with the Feature Pyramid Network. [14] Furthermore, the model better addresses the background and foreground class imbalance issue with its focal loss function. We use the default anchor configuration since the aspect ratios and scales of our target classes are fairly uniform. [15]

2) Inception-Resnet-V2 Training: Though RetinaNet returns a relatively high mAP, we reinforce our detection results by passing them through an Image Classification network, Inception-ResNet-V2. The model recognizes 1000 different classes pretrained on the ImageNet Large Scale Visual Recognition dataset with an input size of 299×299. We aim to use transfer learning to train our Image Classification model to correctly identify Ground Control Points from other similar looking objects.

In order to prepare training data for the Image Classification model, we use results from the RetinaNet to overcome the biases created during object detection. As RetinaNet returns all candidate objects and their bounding boxes within the source photos, we then crop all the detection results and compile them to be trained for Inception-ResNet-V2. [16] Here, we purposefully lower the detection threshold of our object detection model so that our Inception-ResNet-V2 is exposed to as many false detections confused by the RetinaNet during the training phase. In addition to the GCP and manhole classes used in RetinaNet, we also create an ”other” class to distinguish non-GCP objects.

C. Coordinate Conversion and Label Matching

Detection results from RetinaNet and Inception-ResNet-V2 output the position of GCP candidates in the form of image x,y coordinates by taking the center of the resulting bounding box. Recognition of ground control point markers within the images is, however, not enough for GCP auto-rectification because 1) detected GCPs are not associated with a correct surveyed GCP label and 2) false positive detections from the deep learning pipeline are critical to the ground control point rectification step since the result would lend association of image tie points to the wrong three dimensional position. Since surveyed GCP positions are recorded in a specified coordinate reference system in 3D whereas detection results are recorded in 2D image $x - y$ pixel positions, we must convert the detection results to 3D. As mentioned earlier, using triangulation, the points within the images are mapped to three dimensional space in the sparse cloud. This implies that for every pixel $x - y$ of an image, we can assign the location of the pixel in three dimensional space of the sparse cloud. For coordinate conversion, we utilize backward projection of image coordinates to the photogrammetry generated sparse point cloud. Once the detected GCPs are position in the same 3D space as the surveyed GCPs, we match the detections to a corresponding GCP label based on a distance threshold. The results are then used for automatic rectification of GCPs and optimization of the point cloud.

IV. Results

A. Accuracy Evaluation

As per metrics used commonly to evaluate photogrammetry generated data, we use the root mean square error of ground control points to evaluate the proposed automatic GCP rectification accuracy [2], [17], [18]. For each surveyed GCP, we take the residuals between the surveyed reference coordinates and the resulting position within the point cloud after ground control point rectification. Here, we measure the error values based on the measured GCP coordinate values, which are the ground truth. The RMSE error values are then each calculated in the $x - y - z$ directions. $RMSE_X$, $RMSE_Y$, $RMSE_Z$ can each be expressed as the following equations:

$$RMSE_X = \sqrt{\sum_{i=1}^{n} \frac{(X_{surveyed} - X_{rectified})^2}{n}}, \qquad (1)$$

$$RMSE_Y = \sqrt{\sum_{i=1}^{n} \frac{(Y_{surveyed} - Y_{rectified})^2}{n}}, \qquad (2)$$

$$RMSE_Z = \sqrt{\sum_{i=1}^{n} \frac{(Z_{surveyed} - Z_{rectified})^2}{n}}, \qquad (3)$$

where n is Total number of ground control points.

Total RMSE in each of the three dimensions is calculated as follows:

$$RMSE_T = \sqrt{RMSE_X^2 + RMSE_Y^2 + RMSE_Z^2}. \qquad (4)$$

Since RMSE values could vary greatly per test site depending on site and flight specific parameters such as ground sampling distance (GSD), camera resolution, number of ground control points, ground control point distribution, and more as addressed in [17] and [19], we compare $RMSE_T$ values within the same dataset depending on the following rectification methods: no GCP, human, baseline, and proposed method. For when no GCPs are used for rectification, we use the surveyed ground control points as check points rather than reference points for calculating the RMSE values.

TABLE II
GROUND CONTROL POINT ERROR TABLE

Test Site	Rectification Method	RMSE$_X(mm)$	RMSE$_Y(mm)$	RMSE$_Z(mm)$	RMSE$_T(mm)$
A	Without	549.17	840.66	258.53	1036.89
	Human	4.376	1.681	0.960	4.785
	Baseline	10.527	7.758	9.381	16.094
	Proposed	**7.07081**	**4.48643**	**6.521**	**10.614**
B	Without	3017.85	1903.76	1820.56	4005.76
	Human	7.894	5.681	2.212	9.974
	Baseline	6.420	11.151	14.695	19.532
	Proposed	**9.213**	**8.073**	**2.079**	**12.424**
C	Without	233.872	205.116	67.310	318.276
	Human	5.432	3.622	2.049	6.843
	Baseline	12.116	10.889	15.247	22.312
	Proposed	**4.239**	**5.671**	**2.993**	**7.687**
D	Without	1610.8	3733.2	4057.2	5743.9
	Human	4.918	3.591	0.982	6.168
	Baseline	2959.128	913.555	48.107	3097.310
	Proposed	**6.612**	**6.138**	**2.014**	**9.244**
E	Without	6915.7	13069.1	43932.8	46354.3
	Human	3.065	4.647	2.742	6.206
	Baseline	8.960	9.971	2.897	13.7148
	Proposed	**5.512**	**5.434**	**4.461**	**8.934**

B. Deep Learning Model Accuracy

We assess the deep learning model accuracy for RetinaNet and Inception-ResNet-V2, respectively. The objective of the deep learning pipeline is to be as accurate as possible by increasing both precision and recall, which are metrics often used to measure the accuracy of object detection and image classification models. With the same number of true positive detections, higher precision is achieved by reducing the number of false positive counts while higher recall is achieved by reducing the number of false negative detections as seen in the equations below:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{5}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

However, for GCP rectification, precision is the more important accuracy assessment standard than recall. This is because, though marking as many GCPs within the images will help improve the accuracy, the 3D position of GCPs can be rectified by marking a minimum of two images per GCP marker. Therefore, the number of false negative detections do not influence the accuracy greatly. On the other hand, false positives are critical since the erroneously detection result will misguide the rectification process.

In Table III, the number of true positive and false positive detections are each tabulated for RetinaNet and Inception-ResNet-V2. For RetinaNet, the counts are made on bounding boxes with IoU $\geq$ 0.50 and detection confidence $\geq$ 0.50. As seen in results, the precision value significantly increases across all test sites after passing the results of RetinaNet back into Inception-ResNet-V2. Specifically, in test site E, 85 false positive detections are reduced from the original object detection result, thus increasing the precision dramatically from 0.4670 to 0.8737. The reduced false positive detections are reconsidered as true negative detections. Indeed, the number of true positives also reduce upon running through the image classifier, consequently increasing the number of false negative detections. As expressed in Eq. 6, less true positives and more false negatives contribute to lowering the recall. However, this reduction in recall does not influence the overall rectification accuracy by much since finding as many accurate detections is more important than the sheer number of detections. Likewise, We are able to conclude that the additional image classification step helps with significantly increasing the precision of the deep learning pipeline, which contributes to increased accuracy.

TABLE III
DEEP LEARNING MODEL PERFORMANCE

Test Site	Model	TP	FP	$Precision$
A	RetinaNet	204	40	0.8361
	Inception-ResNet-V2	204	8	**0.9623**
B	RetinaNet	222	91	0.7093
	Inception-ResNet-V2	220	22	**0.9091**
C	RetinaNet	1454	258	0.8347
	Inception-ResNet-V2	1442	17	**0.9890**
D	RetinaNet	492	251	0.6622
	Inception-ResNet-V2	384	11	**0.9722**
E	RetinaNet	85	97	0.4670
	Inception-ResNet-V2	83	12	**0.8737**

C. GCP Rectification Results

We assess the proposed automatic GCP rectification accuracy on the five test sites as detailed in Table II with the evaluation method described in IV-A. For each test site, RMSE values are calculated based on rectification by human, baseline, and the proposed method. By additionally measuring

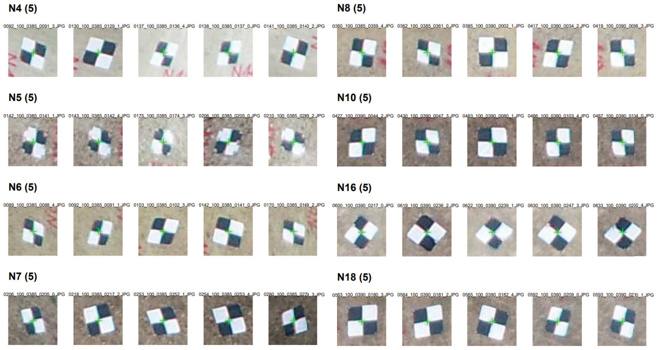

Fig. 5. GCP Detection Result for Test Site A

the RSME value for when no GCPs are used for rectification, we can observe that data without GCP rectification generates error as big as 46.35 m in test site E. Although total GCP number, GCP type, test site area, number of images, ground sampling distance, and the GPS specification varied, similar accuracy levels are observed across the five test sites. It can be seen that the proposed rectification accuracy is nearing human performance, trailing the $RMSE_T$ value by only a few millimeters, well within the acceptable error range of photogrammetry generated data [20]. In fact, all $RMSE_X$, $RMSE_Y$, $RMSE_Z$ values for the five test sites came well within a centimeter, only lending millimeter level error values. The commercially available baseline method was also quite accurate, though the performance was slightly behind compared to the proposed method. Fig. 5 depicts detection results of the eight GCPs used for rectification for test site A.

Test site D lends a relatively large $RMSE_T$ value of 3098.31 mm for the baseline method compared to other test sites. Such substantial error value is produced due to the use of manholes as GCPs. As the commercially available system does not support detection of markers other than checkerboards, the system erroneously confused a neighboring checkerboard marker as the manhole GCP, thus creating large RMSE values upon rectification. We are able to observe here that misled detection and label matching results lead to a significant increase in error values. Remarkably, with deep learning based manhole detection, our proposed method accurately detected and matched the manhole GCP and presented an impressive error of merely 9.2444 mm.

V. Conclusion

In this paper, we present the concept of automated landmark detection from aerial drone images for photogrammetry pipeline automation. With our implementation and architecture, we outperform the commercially existing ground control point detection method and approached the accuracy of human rectification within a few millimeters. By achieving full automation in the sourcephoto processing pipeline, we are also able to dramatically increase the scalability and productivity of the photogrammetry process while maintaining the necessary accuracy levels. Moreover, the deep learning based approach for detecting and classifying targets allows for a flexible selection and addition of ground control point types, potentially removing artificial checkerboard markers.

References

[1] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, "A dense stereo matching using two-pass dynamic programming with generalized ground control points," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 1075–1082.

[2] P. Martínez-Carricondo, F. Agüera-Vega, F. Carvajal-Ramírez, F.-J. Mesas-Carrascosa, A. García-Ferrer, and F.-J. Pérez-Porras, "Assessment of uav-photogrammetric mapping accuracy based on variation of ground control points," *International journal of applied earth observation and geoinformation*, vol. 72, pp. 1–10, 2018.

[3] A. Habib, Z. Lari, E. Kwak, and K. Al-Durgham, "Automated detection, localization, and identification of signalized targets and their impact on digital camera calibration," *Revista Brasileira de Cartografia*, vol. 65, no. 4, 2013.

[4] P. H. Nguyen, M. Arsalan, J. H. Koo, R. A. Naqvi, N. Q. Truong, and K. R. Park, "Lightdenseyolo: A fast and accurate marker tracker for autonomous uav landing by visible light camera sensor on drone," *Sensors*, vol. 18, no. 6, p. 1703, 2018.

[5] J. Wubben, F. Fabra, C. T. Calafate, T. Krzeszowski, J. M. Marquez-Barja, J.-C. Cano, and P. Manzoni, "Accurate landing of unmanned aerial vehicles using ground pattern recognition," *Electronics*, vol. 8, no. 12, p. 1532, 2019.

[6] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.

[7] A. Duda and U. Frese, "Accurate detection and localization of checkerboard corners for calibration." in *BMVC*, 2018, p. 126.

[8] M. Rufli, D. Scaramuzza, and R. Siegwart, "Automatic detection of checkerboards on blurred and distorted images," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3121–3126.

[9] A. Jain, M. Mahajan, and R. Saraf, "Standardization of the shape of ground control point (gcp) and the methodology for its detection in images for uav-based mapping applications," pp. 459–476, 01 2020.

[10] Pix4D. [Online]. Available: https://cloud.pix4d.com/

[11] A. Inc., "Angelswing." [Online]. Available: https://app.angelswing.io

[12] Tzutalin, "Labelimg." [Online]. Available: https://github.com/tzutalin/labelImg

[13] F. Ozge Unel, B. O. Ozkalayci, and C. Cigla, "The power of tiling for small object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[15] M. Ahmad, M. Abdullah, and D. Han, "Small object detection in aerial imagery using retinanet with anchor optimization," in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2020, pp. 1–3.

[16] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. [Online]. Available: http://arxiv.org/abs/1602.07261

[17] E. Sanz-Ablanedo, J. H. Chandler, J. R. Rodríguez-Pérez, and C. Ordóñez, "Accuracy of unmanned aerial vehicle (uav) and sfm photogrammetry survey as a function of the number and location of ground control points used," *Remote Sensing*, vol. 10, no. 10, p. 1606, 2018.

[18] J. Tomaštík, M. Mokroš, Š. Saloň, F. Chudý, and D. Tunák, "Accuracy of photogrammetric uav-based point clouds under conditions of partially-open forest canopy," *Forests*, vol. 8, no. 5, p. 151, 2017.

[19] S. Gindraux, R. Boesch, and D. Farinotti, "Accuracy assessment of digital surface models from unmanned aerial vehicles' imagery on glaciers," *Remote Sensing*, vol. 9, no. 2, p. 186, 2017.

[20] F. Da, C. Nogueira, L. Roberto, T. Körting, and E. Shiguemori, "Accuracy analysis of orthomosaic and dsm produced from sensor aboard uav," 05 2017.