An AI-based workflow for fast registration of UAV-produced 3D point clouds

Ka Lun Leung (Chinese University of Hong Kong), Yong Feng, Yuzhou Chen (City University of Hong Kong)

Mentor: Dr. Yingkui Li, Dr. Kwai Wong

Introduction

The innovation of unmanned aerial vehicle (UAV) technologies has allowed for the generation of very high-resolution three-dimensional (3D) point cloud data (up to millimeters) to detect and monitor surface changes. However, the existing point cloud registration methods, such as the iterative Closest Point (ICP) algorithm and Fast-Global Registration (FGR), do not perform well when dealing with large point cloud data with potential changes over time. Therefore, we develop a new workflow that only performs registration on a cropped segment of the point cloud.

Proposed Workflow

The following Figure 1 demonstrates our proposed workflow.

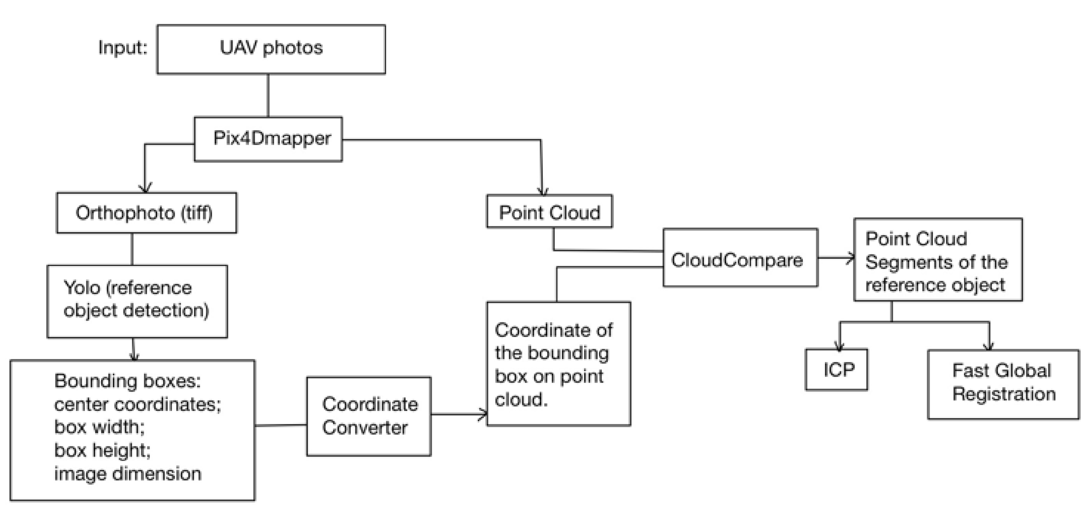

Figure 1. Proposed Workflow

Reference Detection Model

We use YOLO (You only look once) as the core algorithm of our reference detection model and use Roboflow as our server to deploy the YOLO algorithm. Generally, the YOLO algorithm performs faster regarding the object detection area [1]. Currently, we use our object detection model to detect drones, pipes, and houses in the studied plot. After receiving the Json file from the Roboflow server, the Model will extract the class names from the JSON file and set them as the dictionary key to loop them out on the user interface. Furthermore, under each key, it will store two lists, coordinates, and dimensions. Figure 2 shows the result of this model.

Figure 2. Reference Objects detected from the ortho-photo.

Coordinate Converter Model

The Coordinate Converter Model consists of two major functions, one is crop_coordinate(), and the other one is crop_las_file().

Projection

The Reference Detection Model provides bounding box properties for objects in the ortho-photo. However, the photo only covers a portion of the land, and its pixel widths/heights differ from the point cloud's metric units. The Coordinate Converter Model activates when the user selects classes. Its crop_coordinate() function calculates the objects' projected point cloud coordinates using the photo's bottom left corner and each box's bottom left corner. It stores these projected centers in projected_reference_center, and the bounding box ranges in reference_box. After obtaining the local coordinates on the point cloud, the Coordinate Converter Model will invoke the crop_las_file() function.

Figure 3. Ortho-photo on Point Cloud

Cropping

The model uses an iterative process to analyze the projected reference centers and bounding boxes. Different methods handle different detected objects. For "House", it reads the point cloud, generates a 2D array of x,y, and z coordinates called "points", and calculates a 3D bounding box centered on the reference coordinates. It filters the "points" array to only include points within the bounding box (cropping). For "Drone" and "Pipe", it performs white color segmentation and outlier removal, which is especially important for "Pipe" which can have noise. If the object is "Drone", it calculates the center point, returns it to the unsegmented drone cloud, finds the 500 points closest to the center, and crops those as the sub-point cloud. In summary, tailored methods crop optimal segments for each object type from the full point cloud.

Segment Registration Model

We commenced our study by researching the two most popular cloud registration algorithms: Fast Global Registration (FGR) and Iterative Closest Points (ICP). Once we grasped their basic principles, we extracted flowcharts illustrating their algorithmic logic.

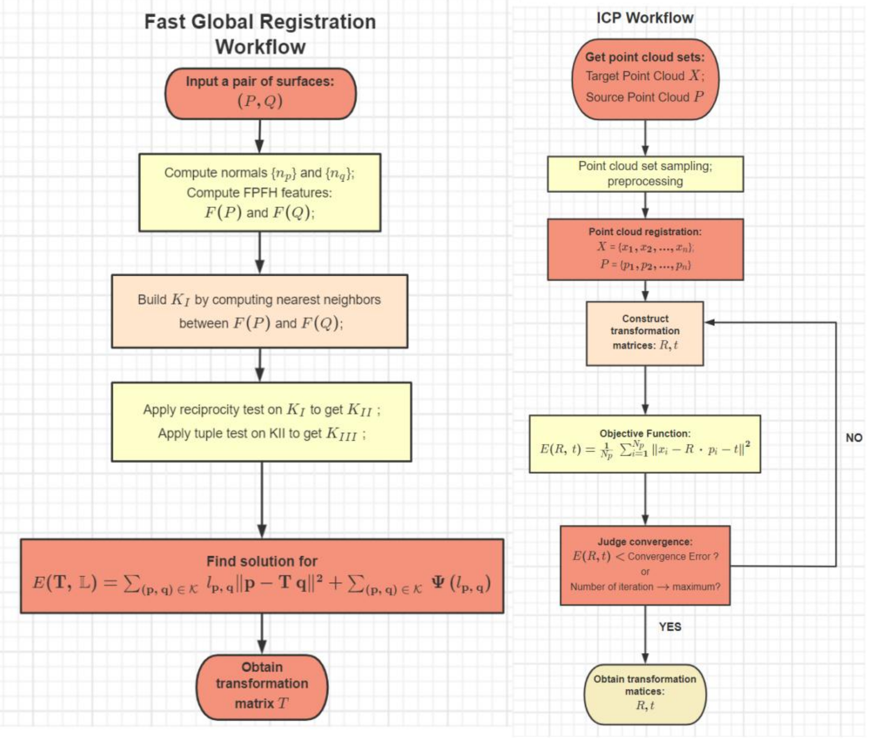

Figure 4. Workflow of ICP and FGR

In general, both ICP and FGR have their own benefits and drawbacks. In this specific case, after conducting several matching tests, we observed that ICP performs better with fewer cloud points, whereas FGR yields more desirable results when more cloud points are applied.

Experimental Result

Our model is tested on the data UAV data collected from three erosion plots at the East Tennessee Research and Education Center. Given the whole point cloud, we can detect and extract houses, pipes, and drones out. We will give our test results, based on two sets of data in table 1 .

	ICP on D	ICP on E	FGR on D, P and H
Distance (m)	0.00684889	0.297478	0.196512
Standard Deviation (m)	0.0301869	0.500817	0.0752717

Table 1. The data collected in 8/9/2021 is aligned to that in 23/4/2021. The former set contains 15,201,529 points while the latter set contains 16,536,909 points.

The running time of each appraoch is listed below:

- ICP on Drones: 5 seconds
- ICP on Entire Point Cloud: 50 seconds
- FGR on Drones, Pipes and Houses: 40 seconds (Normal Approximation) + 75 seconds (FGR)
- FGR on Entire Point Cloud: 40 minutes (Normal Approximation) + $\geq$5 hours (FGR)

Acknowledgement

This project was sponsored by the National Science Foundation through Research Experience for Undergraduates (REU) award, with additional support from the Joint Institute of Computational Sciences at the University of Tennessee Knoxville.

References

[1] F. Joiya. Object detection: Yolo vs Faster R-CNN. International Research Journal of Modernization in Engineering Technology and Science, volume 04, issue 09, 2022. doi:10.56726/IRJMETS30226

[2] P. J. Besl, Member, IEEE, and N. D. McKay. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 14, no. 2, p. 239-256, Feb. 1992, doi: 10.1109/34.121791

[3] T. Jost and H. Hügli. Fast ICP algorithms for shape registration. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, volume 2449, p.91-99, 2002. doi: https://doi.org/10.1007/3-540-45783-6_12