What is That Object Out There? Automated Satellite Modeling and Alternate Reality

Zachary Bergen
Ball Aerospace
Naomi Owens Fahrner
Ball Aerospace
Carl Stahoviak
Ball Aerospace

ABSTRACT

We present a technique for automatically building a 3D model of a Resident Space Object (RSO) and determining its pose. This technique is applicable at a variety of ranges and the 3D point cloud representation increases in resolution with closing range. The pose estimation of the object provides useful intelligence. One example is for capture or docking. We will discuss the exploitation of our data as suitable for Virtual Reality (VR) and Alternate Reality (AR).

1. INTRODUCTION

In this paper, we answer the question of identifying a space satellite automatically from a sensing spacecraft. The cogent questions are: what is the shape? What direction is it headed? What is its orientation?
These considerations are at the heart of Space Domain Awareness (SDA). Individual object knowledge combined with known space objects allows us to determine en masse a predictive capability of encroachment or future proximity. Knowing what you are tracking greatly adds to the ISR (Intelligence Surveillance Reconnaissance) capability. 3D shape can help identify and assess the condition of a space object. An increasingly important role for SDA is to identify active satellites, expired satellites, and orbital debris. The whole idea of SBSS (Space-Based Surveillance Systems) is to keep a much closer watch on space from space itself.

We present a case study of techniques for remote spacecraft/satellite 3D modeling and pose analysis using a hybrid approach combining simulation, classical (photogrammetric bundle adjustment), and pose estimate using Perspective-n-Point (PnP) algorithms.

The study comprises three main areas:

- Exemplar data rendered for simulation input
- Photogrammetric algorithms for 3D point cloud generation
- Pose estimation using PnP algorithms

We have a variety of tools for simulating a satellite in motion in space. There are two simulation inputs:

- Exemplar as a CAD model
- Point cloud representation

1.1 Input Creation

In the first case of input creation, we render a CAD model at various orientations to produce image frames as seen from the sensor. In-house simulation programs are used. The orbit of the spacecraft collecting imagery in the simulation is designed to create reasonable parallax as the sensor motion is controllable. During orbit, a frame capture is simulated using a staring mode. Each frame image contains an aspect of the object that is used as input into a feature detection algorithm. A feature is a keypoint/descriptor pair (cite OpenCV since this is their terminology). A keypoint is a pixel location in an image. A descriptor is a vector that uniquely defines a keypoint. The features are matched (when visible) in each frame and results in a match set of features and frames.

The second case of input creation utilizes a point cloud imaged from the same sensor locations resulting in frames with known point correspondences for each frame. The match set of features is known in this case. The Blender CAD program was used to build point cloud exemplars from the CAD model by creating surface facet/ ray intersections (e.g., LIDAR simulation) to create noiseless object truth sets. We then image the point clouds in a simulation to produce synthetic images and arrive at the same output as in the first case: frames with known matched features.

1.2 Three-Dimensional Point Cloud Creation:

Given our set of matched features in images, we produce ray bundles from the camera perspective center at each collect location. These are input into a bundle adjustment as in [3]. A bundle adjustment optimizes (e.g., in a least-squares sense) the ray intersections between several rays from matching features in several images. The truth set allowed us to test the bundle adjustment without concern for the errors produced with feature matching.

Matched feature points are collected from the imagery using computer vision techniques and sorted for optimal parallax. The collecting craft interior orientation is used to project rays from features to form a ray bundle in inertial space. A least squares bundle adjustment is used along with feature points to produce a representative point cloud. The two point cloud creation options allow us to determine the effects of noise as we perform the pose.

For analyzing the impact of noise, we compared a standard triangulation method with the least squares bundle adjust with and without noise. The triangulation worked far better than the bundle adjustment without noise. When noise was added, the bundle adjustment worked much better.

1.3 Pose Estimation

Given our point cloud, we perform an analysis of pose. The point cloud, as well as the 2D camera projection points, are fed into a Perspective-n-Point (PnP) algorithm, producing a pose estimate. Characterization of the attitude/ephemeris of the satellite motion allows for maneuvers for docking. A variety of techniques used will be presented including model and point cloud generation using Blender and algorithms for combining the classical and ML information.

We implemented several PnP algorithms for the additional pose estimation algorithms. These approaches utilized Gröbner bases to optimize a given objective function. This allows us to find the quaternions for the rotation. After finding the rotation, we were able to substitute some known values to find the translation between the target body frame and the camera frame. The specific PnP algorithms implemented were the Unified PnP, O(n) PnP, Efficient PnP, and Perspective-3-Point.

1.4 Exploitation of the Process

The output of our process allows for exploitation of the data followed by dissemination. We discuss an in-house tool for rendering the objects and motions in AR/VR. This tool is useful for SDA when many objects are involved and are too distant to see without the aid of enhanced graphics. Such tools are being used for situational awareness.

Given a point cloud, it is also possible to compare the cloud sensed by our process with an exemplar. This step attempts to identify a known object given a point cloud. One of the authors holds a patent introducing a novel way of matching point clouds to objects. This would aid in identification of space objects and subsequent accurate rendering. Dissemination of intelligence data is more useful with an accurate rendering vs a point cloud.

We will present the results of our study including exemplar creation, algorithms, statistical plots of the outcomes of each area of analysis, and figures of the point clouds rendered from the sensor perspective.

The output of our process is useful for automated satellite docking, tracking to monitor maneuvers, and other SDA algorithms that determine trajectory vectors and proximity notification. We simulate docking process driven by our automated process.

The simplified component diagram for our modular process is shown in Figure 1. In the camera frame, we formulate algorithms for extracting features in images saved as keypoints (pixel space). From a collection of images, we obtain a point cloud (3D feature map) that is fed into a pose estimator. Subsequent sections detail these steps. Each step is modular in the sense that alternate feature detections, photogrammetric, and pose algorithms may be substituted. The cogent feature of this paper is the overall approach of combined steps to create a system providing utility as a whole and in part for other purposes such as onboard image processing capabilities to support autonomous missions such as resident space object (RSO) inspection, docking, and re-fueling.

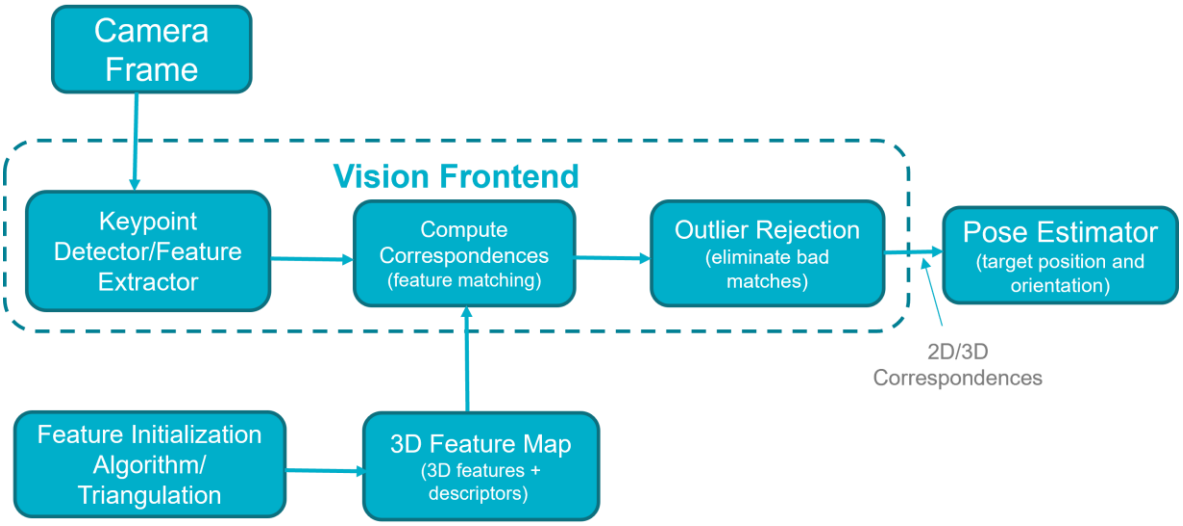


Fig. 1: System component diagram

2. OBJECT POINT CLOUDS

2.1 Image Processing

To begin our process for pose estimate and docking we must build a model of the RSO that gives us the ability to discern its shape with enough resolution to automatically recognize features during rendezvous. For the modeling step, we form a tasking methodology for collecting information about the RSO that can then be used to calculate a 3D representation of the RSO. We determine an orbit for scanning the RSO of interest. 2D images are collected at angular intervals at a fixed distance from the RSO. For each image, the exterior orientation of the sensor is recorded and matched with the collect. The collection strategy is to increase the angle between collect to establish a reasonable parallax for the stereo-space resection while maintaining visibility of features shared between subsequent images. We need a minimum of two rays from the sensor perspective center through each pixel location of an image feature to create a 3D space point. Given the controlled environment, we may select any angular separation for imaging.

2.2 Keypoint Detector/Feature Extractor

Given a sequence of images we step through the images and collect features. Features are pixel locations in images that represent the same features in two or more images of our RSO scan. We investigated several algorithms for feature extraction including Features from Accelerate Segment Test (FAST), Oriented FAST and Rotated BRIEF (ORB), Scale-Invariant Feature Transform (SIFT), and Shi-Tomasi “Good Features to Track”. For matching features we used brute force/greedy and Fast Library for Approximate Nearest Neighbors (FLANN)m (OpenCV algorithms).

Figure 2 shows image pairs of our RSO at different angles with lines between matched features.

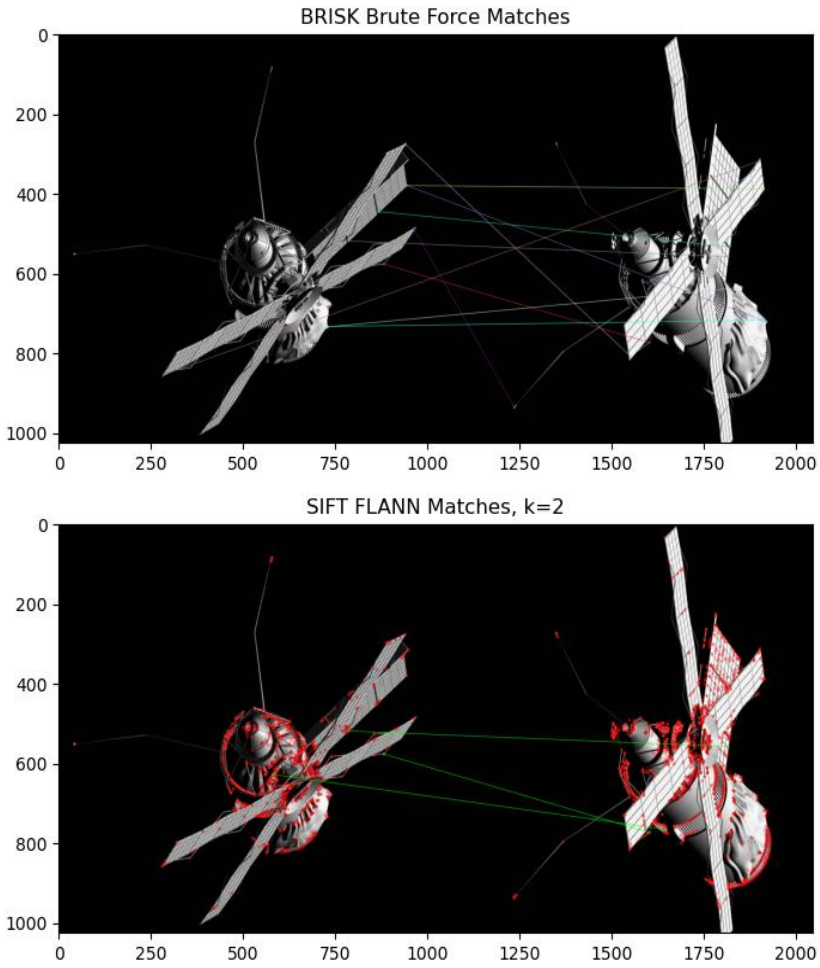


Fig. 2: Feature Matching using OpenCV tools

2.3 Triangulation/Bundle Adjustment

Given a set of image points with matching feature locations, we may form rays from the camera perspective center into space. The set of rays for a given feature in several images are called a feature bundle (citations?). Given a bundle of rays for a single feature we can calculate the best (Least Squares sense) estimate or the 3D location of the feature in the images. In photogrammetry this technique for reconstruction is called a bundle adjustment and can typically combine tie points, ground control points, and complex camera models in a solution where several parameters are tunable within some tolerance to achieve the best fit across an image. For the purposes of this study, we used a least squares formulation for solving the ray bundle to get 3D points that result in a point cloud expression of the RSO [3]. Traditional bundle adjustments are also available for possibly increased accuracy at the risk of slower speed and higher numerical complexity.

Given the features extracted from the imagery described earlier, we create rays from each camera perspective center to form ray bundles with matched features for each bundle. The ray projections from exterior to interior orientation through the perspective camera center to the focal plane are rearranged to accommodate the least squares solver formulation. For a single feature and image pair we have the geometry shown in Figure 3. The point p is a feature in 3D and the vectors r1 and r2 emanate from p to the respective camera perspective centers. The intersection of each ray with the image plane results in a pixel location. We collect these feature image points (2D), sort them by feature, and reverse the process in our bundle adjustment to get an estimate of p (3D).

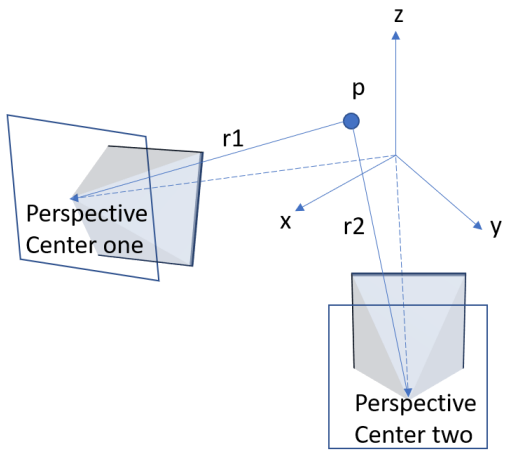


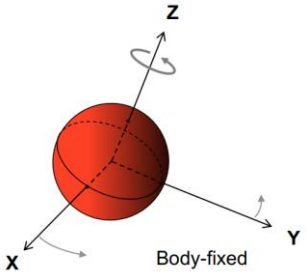
Fig. 3

First, there are three important coordinate systems used in the Feature Initialization (Eq. 1):

- Body
- Inertial
- Camera

Body-fixed frames are tied to a named body and rotate with it. The camera frame is fixed with respect to the camera body and has its origin at the perspective center. The inertial frame is common to the body and camera frames and provides a relative orientation between the two.

With these coordinate frames of reference, it's easy to formulate the transformation between a 3D body point and a ray in the camera frame. The ray in the camera frame may be projected on the focal plane given the pixel size and focal length. It's common to use a pinhole camera model, although it is easy to add a radial distortion model to the camera matrix.



The equation for translation into the 2D camera frame into the 3D RSO feature location is (Equation 25 in [3]):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{C}_{j/cam} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{C/I} \left(\begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix}^{B/I} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^{B}_{j/targ} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}_{targ/cam} \right) \quad (1)$$

Starting in the parenthesis, the term on the left represents rotating a point in the body frame into the inertial frame. After the rotation our body point is in the inertial frame. When we add the translation vector (t) to this point, we create a ray originating at the body point (red) and projecting through the camera perspective center as shown in blue below.

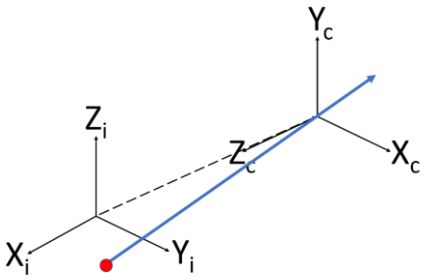
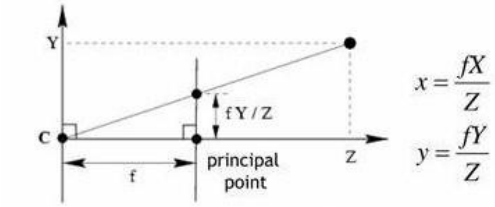


Fig. 4

Next up in Equation 4 is the left-most matrix. This matrix rotates the ray into the camera frame. In photogrammetry this step is called moving from the exterior orientation to the interior orientation of the camera. The transformed ray is now relative to the camera system. We may take this ray and multiply it by the camera matrix. Doing so gives us a location on the focal plane. In general, the camera matrix is formed with the focal length in pixels and the focal plane locations are normalized by the z value of the ray. This results in pixel locations. The center of the focal plane is called the principal point. In most applications the principal point is defined by an offset from the image corner. IMPORTANT: in [1] the principal point is assumed at (0,0) with no pixel offset from the image corner.

Referring to the figure below, we look at the Z/Y camera frame plane. The perspective center is at C. Our ray goes through C and intersects the focal plane at a axial distance f from C. By similar triangles and the camera matrix (shown bottom of figure) we see that the y component of the intersection is given by fY/Z and similarly for x (fX/Z). With a focal length, f, in pixels we see that the ratios Y/Z and X/Z are dimensionless, and we have the ray projection from the original body point in pixels with the principal point as the origin.



$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

Fig. 5

For the pixel location of the ray projection on the focal plane, it's common to use a focal plane in front of the perspective center to ease the understanding of similar triangles. The formulation here has the camera matrix multiplication built-in as shown in [3] Equation 5:

$$\overline{z}_j = \begin{bmatrix} u_j \\ v_j \end{bmatrix} = g(\overline{s}, X_j) + \overline{n}_c = \frac{foc}{z^C_{j/cam}} \begin{bmatrix} x \\ y \end{bmatrix}^C_{j/cam} + \overline{n}_c \quad (3)$$

From this equation we can see that there is no principal point offset in the calculation. We address this translation in both the forward and backward calculation. Although a simple pinhole camera projection is used, the rotations from body, to inertial, to camera, and then to the focal plane need to follow the formulation exactly for the stacked equations and least squares solver to work properly. Of course, any explicit decomposition of a series of similar steps would also work for the bundle solver.

The convention use for the geometry for the interior to exterior orientation is that the focal plane is forward of the camera perspective center. This has the effect of changing the sign of the pixel locations for a feature in the image plane. Also, the principal point is calculated from an image corner as the origin. The pixel locations must be corrected accordingly to use the least squares bundle solver.

2.4 Exemplars

The RSO to be modelled and estimated is represented as a point cloud. To garner the points in 3D to describe the shape we follow several steps involving image capture at different angles in an orbit around the RSO and capture images that are used to find features. The features can be matched between two to several images and 3D point locations calculated. The point set represents our object. In order to test the algorithms after the point cloud is obtained, we use a perfect point cloud set representing the RSO: exemplar. The RSO to be modelled can be represented in an ideal noiseless state as a point cloud comprised of 3D points on the surface of the object. The noiseless point cloud representation is an exemplar. To garner the points from the model we use a CAD representation (e.g., wavefront file) and use a Python script to gather facet points representing perfect known features on the RSO.

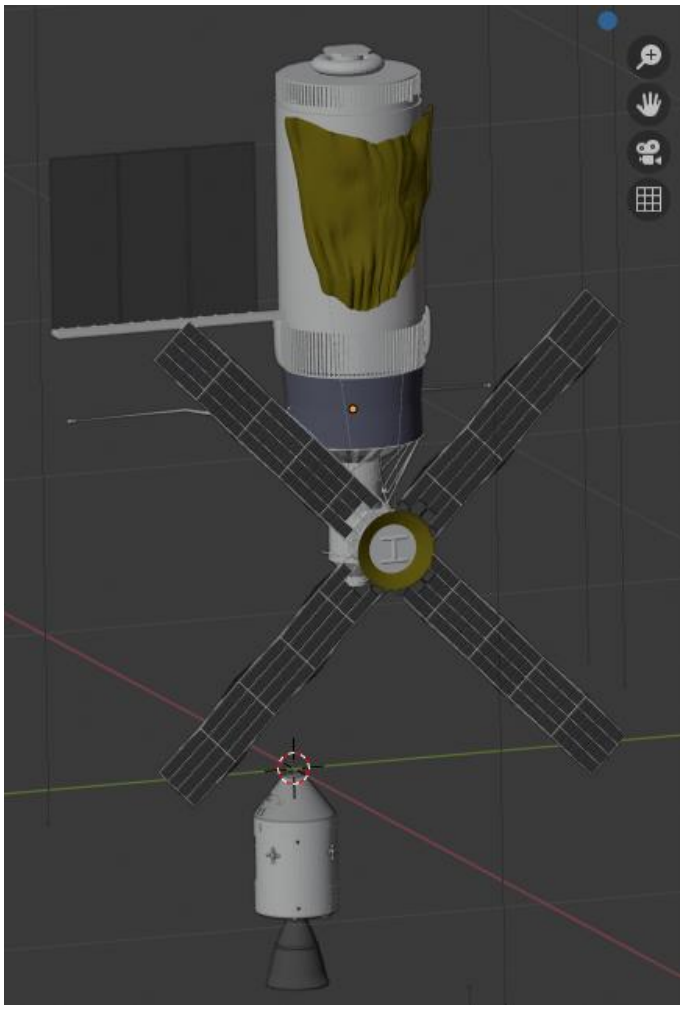


Fig. 6 CAD model render of satellite and docking capsule (lower)

Figure 6 is a CAD rendering of a satellite. Any CAD file may be used. The CAD object is built with millions of facets as shown in Figure 7. For our study we generated facet midpoints on the exterior facets of the satellite model and saved them to a point cloud file. The point cloud is easily decimated by voxelization or other techniques to manage the number of points (i.e., resolution).

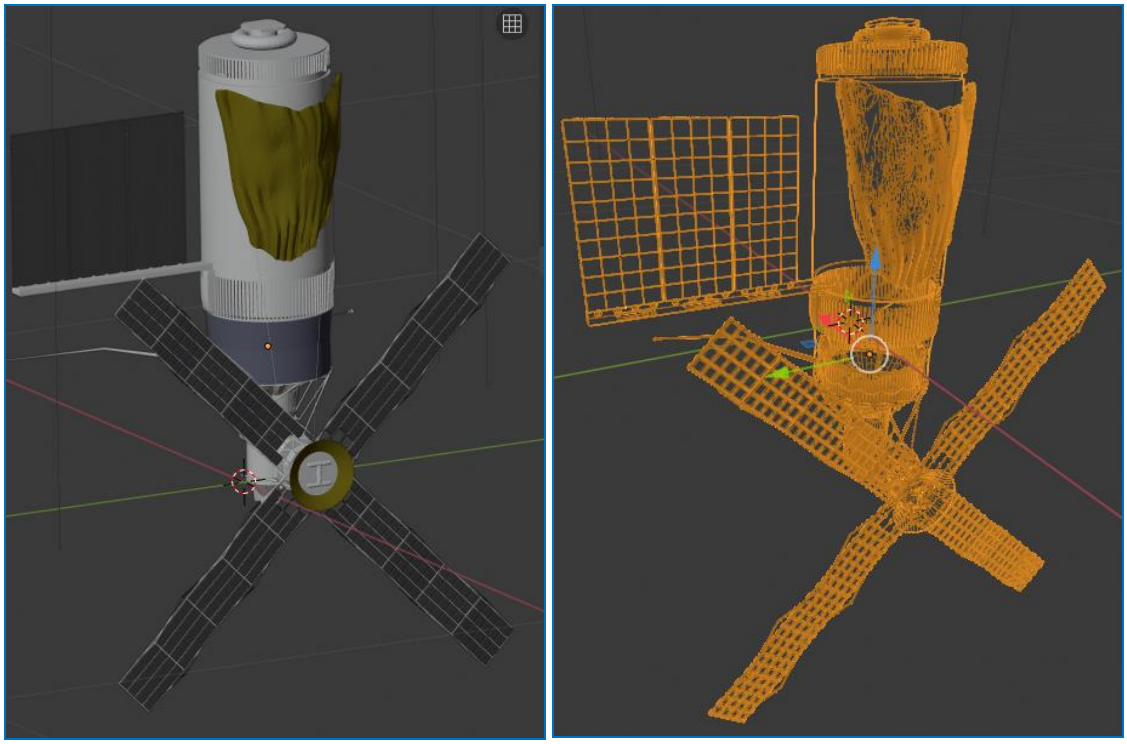


Fig. 7 satellite render and satellite model facets

The point cloud may be used to bypass the point cloud creation for the purposes of testing the photogrammetry with a noiseless test set. It may also be used as a blueprint for comparison. For example, if a satellite has been modified or damaged there exist techniques [1] for comparing point cloud of objects at desired resolutions (e.g., on the order of the solar panel size or smaller depending on desired detail).

The system we envision can utilize point clouds from multiple sources. Point clouds may be obtained by image features and bundle adjustments as we have shown. Point clouds can be obtained from LIDAR and SAR. The modular aspect of our process allows for each step to be replaced with any suitable or preferred algorithm. The sum total of our process is a vision for a system that allows for deviation at any step to explore more detail at each step. For instance, the point clouds can be voxelized to a certain digital accuracy and use to catalog the condition of a satellite over the period of performance. Such digital signatures can be used to monitor the condition and orientation of satellite components.

2.5 Results

We used our RSO model to generate images in a capsule (with camera) orbit around the RSO and ran our full process of feature extraction from imagery, feature matching and ray bundle formation, bundle adjustment, and 3D point cloud collection. The result is a point cloud representation of the RSO. Figure * shows the result of reconstructing our RSO without/with noise. The left image of Figure * is a perspective view of the point cloud reconstructed from the ensemble of image feature matches. Obviously with a noiseless feature set a point cloud could be obtained with a simple stereo-space resection of rays. However, when we add noise, the advantage of the least square bundle is evident in the right hand image of Figure 8. If we further voxelize the points at some minimum needed resolution, we will average more noise over the scale of the RSO and preserve an overall pose estimate which is discussed in the next section.

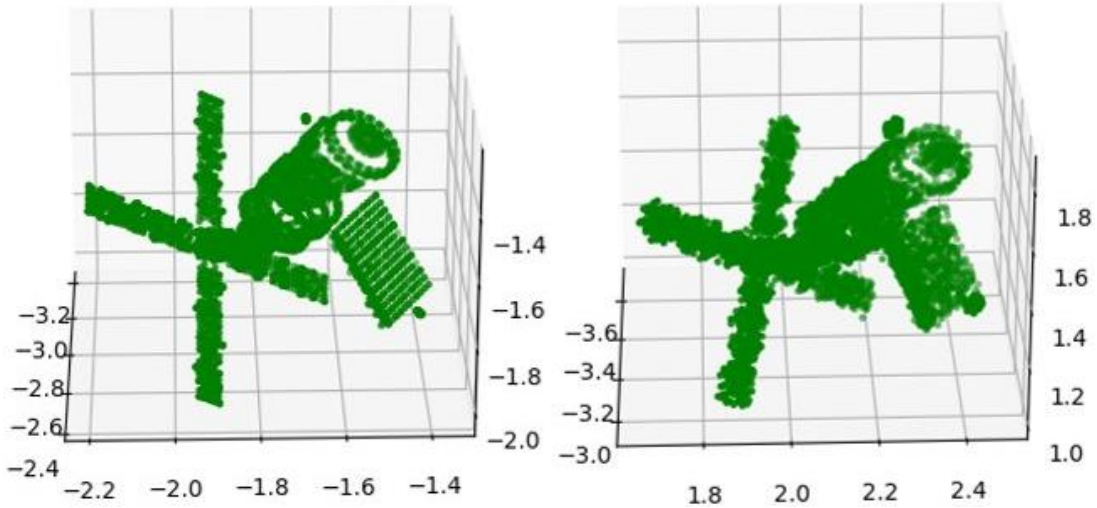


Fig. 8 Model point cloud reconstruction without (left) and with noise (right)

3. POSE ESTIMATION

Given the accumulated point solutions from the bundle adjustment (point cloud) we explore several algorithms to solve the Perspective N-Points problem (PnP). This is the problem of estimating the pose of an object (or camera) when both the 2D projections and 3D points of features are given. The goal of PnP algorithms is to fit the absolute position and orientation of the camera to the measurement data given a certain number of correspondences between 3D world points and 2D image measurements. The PnP algorithm takes inputs of the 3D point locations with respect to a reference/world frame, and the 2D camera projection points. The outputs are the rotation from the reference/world frame to the camera frame, the position of world/reference origin seen from camera, and the depth from the camera center to each point. To find the required parameters, we need a rotation from world frame to camera frame that is consistent with the mappings from world points to corresponding image points.

We have explored the following PnP algorithms: Unified PnP (UPnP) [8], Optimal solution to PnP (OPnP) [9], efficient PnP (EPnP) [10], and Perspective-d-Points (P3P) (cite OpenCV). The component diagram for this process is shown in Figure 9. The algorithms are well-known, and we present only the results of our search for the best relative performance for our use case.

Referring to Figure 10, we take the results of the bundle solution and use them in PnP algorithms to generate pose estimates in the camera frame. Given the known points from either noiseless data or the exemplar we may produce relative performance metrics in rotational angles and translation.

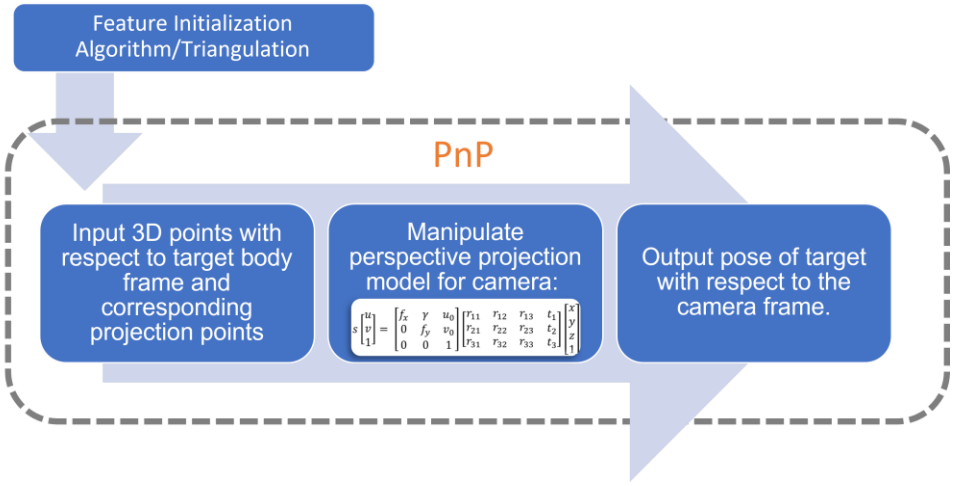


Fig. 9 Pose estimation diagram

The results are shown in Figure * (Error in relative dimensionless units).

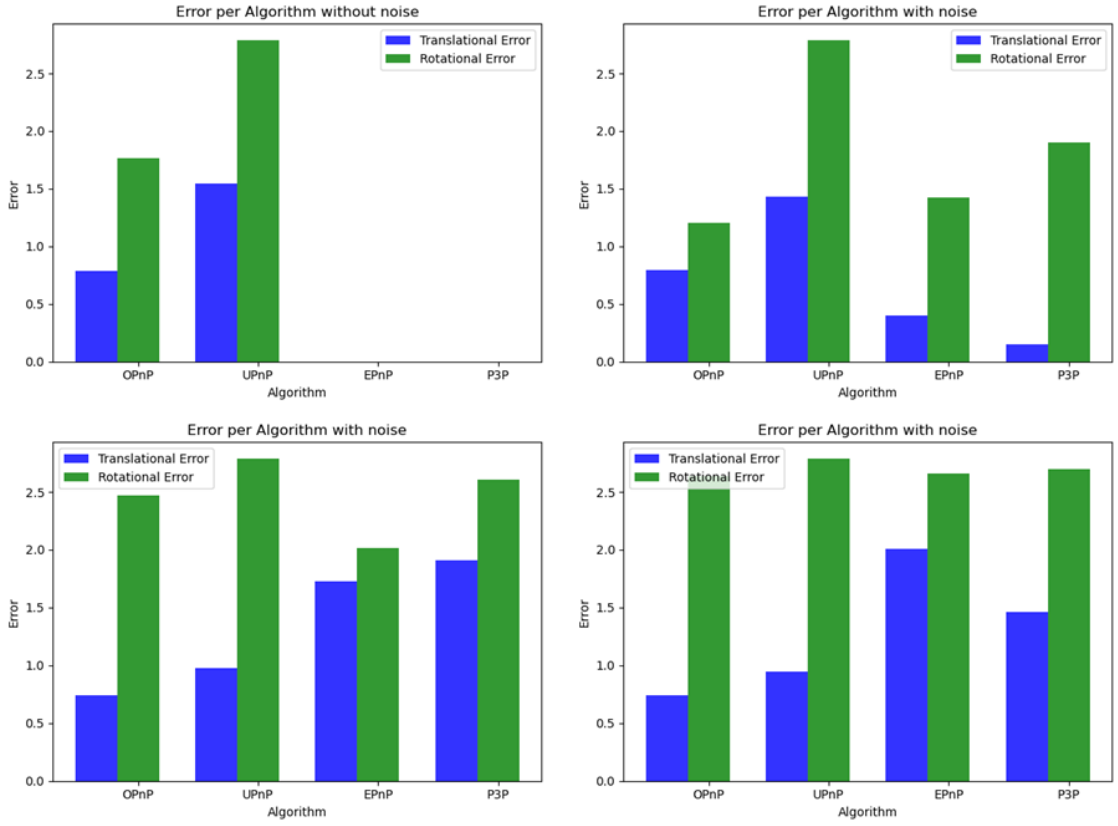


Fig. 10 PnP Comparisons: Upper Left: no noise, Upper Right: absolute magnitude 1.0 Gaussian noise, Bottom Left: 5.0 Gaussian noise, Bottom Right: 10.0 Gaussian noise

Figure 10 reveals choices for the pose estimation depending on the most important docking factors. The entire point cloud formation Feature Initialization Algorithm (FIA), and pose estimation and reconstruction (SLAM-inspired Pose Estimation and Reconstruction - SPEAR) is detailed in [3].

4. SIMULATION

Given our process for pose analysis, it is useful to run this end-to-end process in a physics-based simulation. Given an adequate RSO model and pose, we may simulate hosting the software on our capsule in a docking scenario. The geometry for our scenario is shown in Figure *. For the approach capsule (the chaser and rendered in Figure 1 bottom), we have a camera, Inertial Reference Unit (IRU), Star Tracker, and GPS. The approach capsule also has a dynamics simulator and actuator modes. The RSO (target) is a simplified object with enough shape to calculate pose uniquely. For the algorithm presented here, we employ our feature point extractor, 12 DOF target state estimator, and feed that information back to the capsule for guidance and control. In practice an on-board process would contain the entire algorithm process. We separate the algorithm from the physics to exercise our process flow in a test harness with known information.

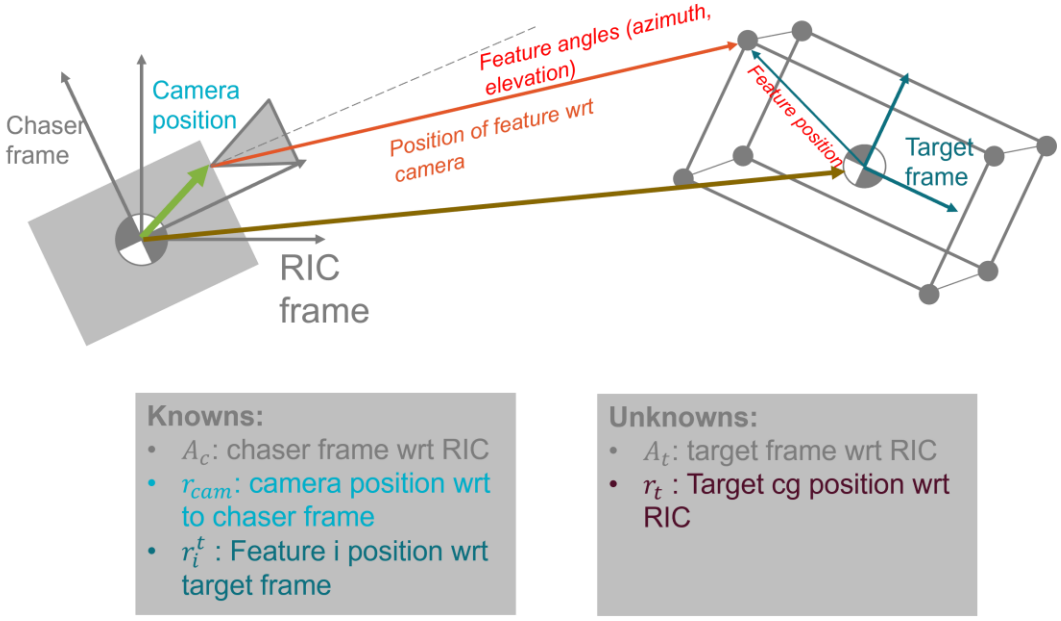


Fig. 11 Reference frames used in the docking simulation

During the simulation loop, we estimate the kinematic state and the 3D location of the body frame features. These states are propagated forward in time given an updated motion model at each step (Equation 3 [3]). For the guidance, navigation, and controls we developed a Joint Probabilistic Data Association Filter (JPDAF) with promising results and utilized it in a simulation to demonstrate a full closed loop simulation. The simulation loop diagram is shown in Figure 12.

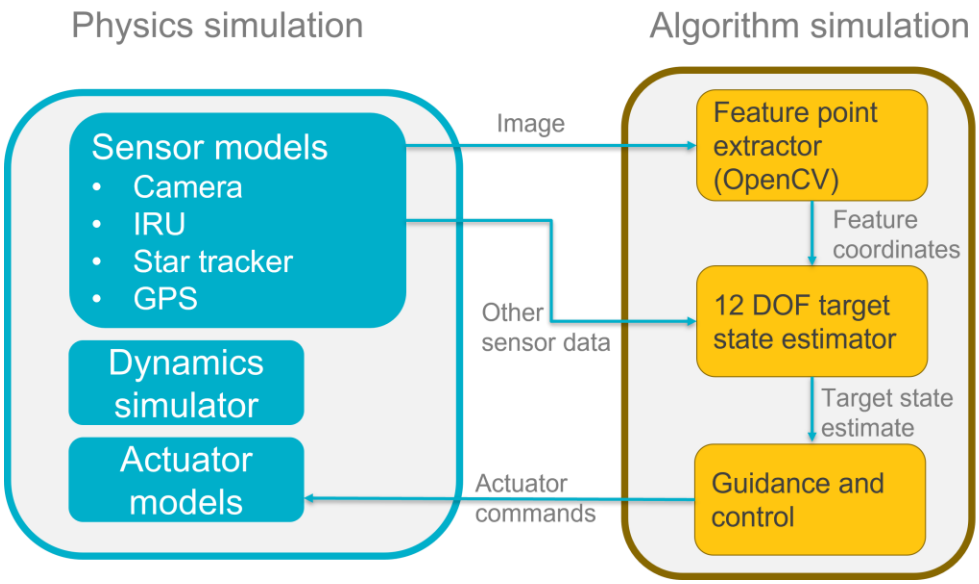


Figure 12 Close loop simulation

The simulation uses a simplified RSO (screenshot shown in Figure 13) with enough axial differentiation to allow for a unique pose estimation from a capsule viewpoint. The graphical output on the right shows the error rates of position and attitude over time during the capsule approach to the RSO.

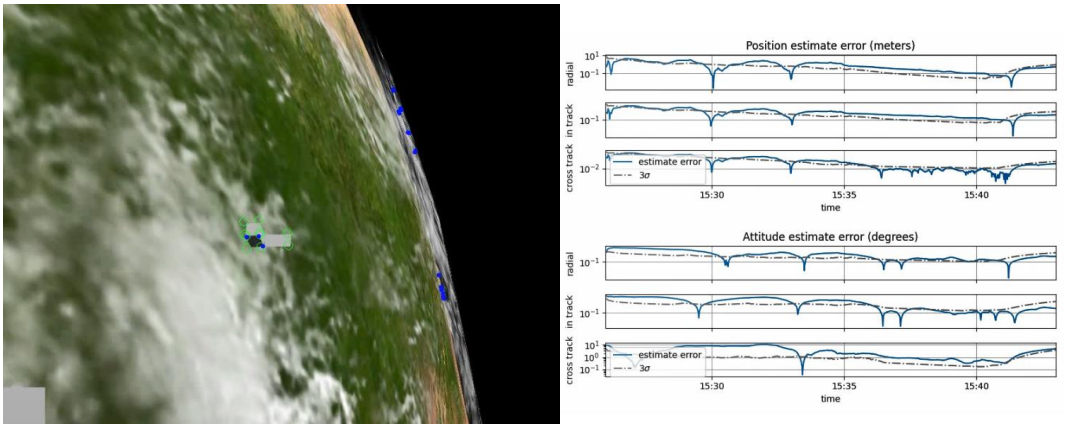
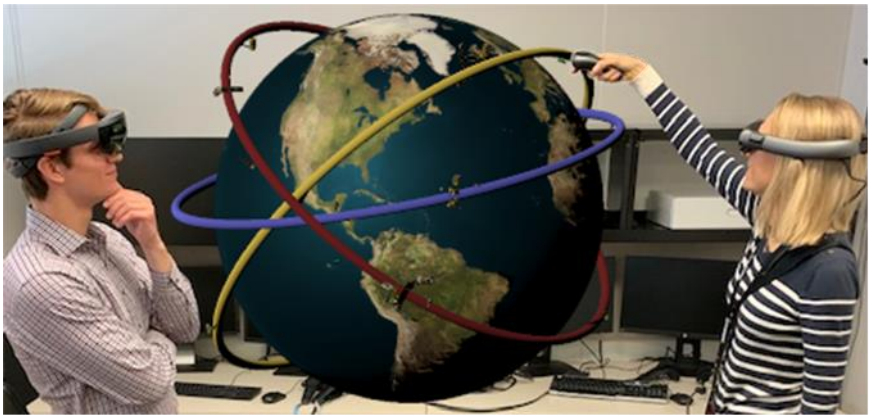


Fig. 13 Docking Simulation

5. ALTERNATE REALITY (AR) / VIRTUAL REALITY (VR)

COTS software is increasingly offering AR/VR options. Some Graphical User Interfaces (GUIs) now have a VR mode for viewing data. VR and AR offer an immersive experience to better understand time-space data. VR assists the interpretation of SDA by allowing for timeline orbit and encounter visualizations and changing observer views with respect to scenarios. The SDA is enhanced via arbitrary observer locations for perspective/scale in space-time and full volumetric degree of freedom independent of RSO activity.



6. CONCLUSION

A hybrid technique for automated docking has been presented. Options for intermediate uses and exploitation are included. An end-to-end automated process has been discussed, including feature initialization using AI image processing tools, a least squares bundle adjustment, and pose estimation algorithms. This process culminates in a target state estimation driven by the inputs from previous steps. We have included a closed loop simulation to demonstrate the efficacy of the algorithm.

7. REFERENCES

[1] A Alahi,, R Ortiz, and P Vandergheynst, FREAK: Fast Retina Keypoint, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE, June 2012

[2] S Augenstein, Monocular Pose and Shape Estimation of Moving Targets, for Autonomous Rendezvous and Docking, PhD Thesis, Stanford University, June 2011

[3] S Augenstein and S Rock. Improved Frame-to-Frame Pose Tracking during Vision-Only SLAM/SFM with a Tumbling Target, *2011 IEEE International Conference on Robotics and Automation*, Shanghai International Conference Center, May 9-13, 2011, Shanghai, China.

[4] S Augenstein and S Rock. Simultaneous Estimation of Target Pose and 3-D Shape Using the FastSLAM Algorithm, *AIAA Guidance, Navigation, and Control Conferenc*e, 10-13 August 2009.

[5] H Bary, T Tuytelaars, and L Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding, Volume 110, Issue 3, June 2008, Pages 346-359

[6] Z Bergen. *Object Point Cloud Comparison and Matching*, U.S. Patent 11,106,936 B2, August 30, 2021.

[7] Ebrahim Karami, Siva Prasad, and Mohamed Shehata, Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images, arXiv:1710.02726

[8] L Kneip1, H Li1, and Y Seo, UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability, ECCV 2014, Part I, LNCS 8689, pp. 127–142, 2014

[9] V Lepetit, F Moreno-Noguer, and P Fua, EPnP: An Accurate O(n) Solution to the PnP Problem, Int J Comput Vis, DOI 10.1007/s11263-008-0152-6

[10] S Leutenegger, M Chli and R Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints, 2011 IEEE International Conference on Computer Vision

[11] C Liu and W Hu, Relative Pose Estimation for Cylinder-Shaped Spacecrafts Using Single Image, IEEE Transactions on Aerospace and Electronic Systems, VOL. 50, NO. 4 OCTOBER 2014

[12] D Lowe, Distinctive Image Features from Scale-Invariant Keyponits, International Journal of Computer Vision, 2004

[13] F Moreno-Noguer, Deformation and Illumination Invariant Feature Point Desctiptor, Deformation and illumination invariant feature point descriptor, CVPR 2011, 2011, pp. 1593-1600, doi: 10.1109/CVPR.2011.5995529.

[14] F Moreno-Noguer, V Lepetit, and P Fua, Accurate Non-Iterative O(n) Solution to the PnP Problem, Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision, January 2007

15] A Mousavian, D Anguelov, J Flynn, an J Kosecka , 3D Bounding Box Estimation Using Deep Learning and Geometry, IEEE Conference on Computer Vision and Pattern Recognition, 2017
[16] Rekhil M Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7668-7673
[17] E Rublee, V Rabaud, K Konolige, and G Bradski, ORB: An Efficient Alternative to SIFT or SURF, 2011 IEEE International Conference on Computer Vision
18] S Umeyama, Least-Squares Estimation of Transformation Parameters Between Two Point Patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 13, No. 4, April 1991
[19] H. Yang, J. Shi, and L. Carlone, TEASER: Fast and Certifiable Point Cloud Registration, IEEE Transactions on Robotics (T-RO), 2020
[20] Y Zheng Y Kuang, S Sugimoto, K Åström, and M Okutomi, Revisiting the PnP Problem: A Fast, General and Optimal Solution, 2013 IEEE International Conference on Computer Vision