International Journal of Sensors and Sensor Networks
2022; 10(2): 16-24
http://www.sciencepublishinggroup.com/j/ijssn
doi: 10.11648/j.ijssn.20221002.11
ISSN: 2329-1796 (Print); ISSN: 2329-1788 (Online)

Helipad Pose Estimation Using Intelligent Helipad and LiDAR Camera Fusion (IHLCF)

Mohammad Sefidgar[*], **Rene Jr. Landry**

Department of Electrical Engineering, ETS University of Quebec, Montreal, Canada

Email address:
mohammad.sefidgar@lassena.etsmtl.ca (Mohammad Sefidgar), renejr.landry@etsmtl.ca (Rene Jr. Landry)
[*]Corresponding author

To cite this article:
Mohammad Sefidgar, Rene Jr. Landry. Helipad Pose Estimation Using Intelligent Helipad and LiDAR Camera Fusion (IHLCF). *International Journal of Sensors and Sensor Networks*. Vol. 10, No. 2, 2022, pp. 16-24. doi: 10.11648/j.ijssn.20221002.11

Received: May 28, 2022; **Accepted**: June 15, 2022; **Published**: September 19, 2022

Abstract: Pose estimation has evolved into a beneficial concept in autonomous systems. It refers to the techniques used by computers to detect and quantify certain features in an image. The present work proposes modified helipad intelligent detection and pose estimation, using a fusion of camera and LiDAR. The image data are first collected using Otsu thresholding through the downward drone camera and converted to a binary image. Next, Boundary Parametric Ellipse Fitting (BPEF) algorithm is employed to detect circles, which will turn into ellipses when there is a tangential distortion in an image. Then, Ellipses Region of Interest (EROI) is extracted from the images via the potential circles. The algorithm uses a modified version of the helipad with an arrow sign located outside of the helipad's circle. The arrow's centroid point is located on the axial line, which horizontally splits the word "H" and passes the word's centroid. Hence, using the proffering over-the-line-and-between-ellipses-check technique, potential arrows are extracted. A Support Vector Machine (SVM) is then trained to detect the helipad over 400 images of the word "H" and Arrow patterns. The "H" and the Arrow corners are detected and localized in the following phase. The projected LiDAR data is followingly utilized to find the corners depth information. Finally, the translational and rotational pose components are projected to obtain the corners' coordinates and the rigid body transformation. Software-in-the-Loop (SIL) is used to assess the method accurately. The experimental setup is tuned so that the drone stays motionless over the landing platform and conducts the pose estimation. The method was compared with the AprilTag Detection Algorithm (ATDA). A statistical Root Mean Square Error (RMS) is also used to gauge the accuracy of the proffered method. The analysis results confirmed a notable improvement in rotational and translational estimations.

Keywords: Pose Estimation, SVM, LiDAR Camera Calibration, UAV, Landing System, Drone

1. Introduction

This work covers the problem of drone landing through a modified version of the helipad and Artificial Intelligence (AI). The following sections discuss the problem statement and review state-of-the-art works addressing UAV landing approaches.

1.1. Problem Statement

Drone landing is one of the most challenging areas of research to have emerged over the past few years. It plays a pivotal role in today's world. Nowadays, drones are served in various tasks, including goods delivery, emergency acts, aerial imaging, to name a few. A myriad of components and actions are required to control and manipulate Unmanned Aerial Vehicles (UAVs). The current research focuses on the pose estimation section of the drone control system. It refers to spatial distance measurement of the aerial robots for a fixed origin, such as a landing sites, goods to be collected from stores' shelves.

While recent works use a variety of algorithms and sensors to operate this computation, in this work, a new pattern for helipad is designed, and fused data from the LiDAR and camera are employed for pose estimation. Initially, images and ranges data are collected via the downward camera and LiDAR sensors. The images are then converted to binary, stored in a single dimension matrix, from RGB image, represented by red, green, and blue matrices. Boundary Parametric Ellipse Fitting (PEF) technique followed by Ellipses Region of Interest (EROI) extraction is performed to eliminate unnecessary parts of the scene. The subsequent technique for removing redundant details from the image is called over-the-line-and-between-ellipses-check. This algorithm checks that the centroid of the arrow, located between the helipad ellipse and EROI, is selected. A Support Vector Machine (SVM) is also trained over 400 images of the arrow and the word "H" Region of Interest (ROI) for the detection of the helipad. After SVM confirms the detection, corners of "H" and arrows are extracted and localized. In the subsequent phase, the ranges data are projected to the image using the camera intrinsic parameters and spatial location of the LiDAR to the camera. The calculated corners and rigid body transformation estimation are finally utilized to the projection for the rotational and translational displacements of the landing platform to the drone.

1.2. Motivations

There are numerous downsides to the developed state-of-the-art pose estimations techniques, leading to the necessity of developing new hardware and software design. Monocular cameras pose estimation is deemed to be ill-posed, causing complications for pose estimation based on sole digital image processing techniques [1]. The next issue is that image base pose calculation relies on the scenes and the features, from textures and shapes; hence detection, for pose projection, based on a texture-less irregular object placed in a heavy-occlusion scene can lead to high computational cost, high latency, low accuracy, and even erroneous results [2]. Moreover, previous works used descriptors, such as SIFT and SURF, matching, and perspective-n-point (PnP) as critical components for their algorithms; however, this approach can only be successful, provided sufficient and appropriate matches are found [3]. Therefore, much research employs multiple sensors and algorithms to ameliorate the estimation. Global Positioning System (GPS) and Inertial Navigation System (INS) are two examples of these tools. While GPS is utilized in various applications, the signal is heavily reliant on weather conditions. It only can be used outdoor [4]. Even if excellent meteorological conditions exist, there are still two major issues of GPS signal spoofing, leading to inaccurate position calculation and signal jamming, which induces a high noise level on the sensory data. Moreover, the heaviness, bulkiness, and high-power consumptions are typical of Inertial Navigation System (INS) inclusion in design [5].

Some algorithms include estimation techniques to have multiple sensors in one design. The most well-known examples of these approaches are Kalman Filter (KF), Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Particle Filter (PF). The suitability and practicality of the implementation based on them depend on the applications' nonlinearity. KF application for the linear system is perfectly proper. However, the slight nonlinearity can be easily handled by EKF approach to the extent that noise statistic such as covariance matrixes of the process noise is accurate. UKF profoundly overcome the insufficiencies of the EKF in system nonlinearity. However, it still suffers from the fact that the rounding error and approximation error increase when the system's nonlinearity grows. PF can manipulate a more complex and nonlinear system; nonetheless, the accuracy of the calculations largely depends on the number of particles. This increase can boost the computations, leading to the expensive and heavy system design [6].

1.3. Contributions

There are several contributions that the current research proffers, which can be listed as follows:

1) The proposing algorithm's first and most important contribution would be developing a fast circle detection technique based on the Boundary Parametric Ellipse Fitting (BPEF).
2) Second, boosting the accuracy of altitude estimation during the landing compared to the sole monocular camera-based methods such as AprilTag.
3) Third, the proposing method employs a low-resolution, low-weight LiDAR, eliminating the extra expenses of a powerful electrical feeding system and costly computers with Graphical Processing Units (GPUs).
4) Rotational translations are corrected via a combination of LiDRA and image data; and
5) Low-light-conditions-related inaccuracies can be compensated to a great extent through the data from the range sensor.

2. Related Work

The helipad detection problem was first addressed in 2007, in Nsogo et al. [7] article, after Sharp et al. [8] developed an algorithm for the landing of an unmanned aerial vehicle. Sharp proposed an algorithm that described the extraction of the main features such as corners, lines, and edges. That algorithm was inefficient, given that it was based on geometric moment descriptors, which were not sufficiently robust for helipad detection. In view of that, a novel Normalized Wavelet Descriptor (NWD) is proffered herein and compared to geometric moments and a Fourier Descriptor (FD). The NWD differs from the previously defined algorithm proposed by Sharp in several ways: with the NWD, the threshold value is obtained automatically for each image, edge features are used for the helipad, and the algorithms developed can be used for most types of landing targets, not just helipads. In 2009, a feature-based single-camera vision system for the safe landing of an unmanned aerial vehicle (UAV) was proposed [9]. This approach was based on the analysis of optical flow and of the mutual geometric position of different kinds of features, observed from different points of view. Later that year, a more robust algorithm was introduced in [10]. Zeng Fucen focused on object recognition and adaptive threshold selection in the vision system to make the algorithm more robust. A few years later, in 2012, another vision-based landing system was proposed, in the study of Lange et al. [11], to deal with the high cost and complexity of the Ground Controlled Approach (GCA) and Instrument Landing System (ILS) [12]. It proposes a real-time detection method, which aims to grant UAVs the ability to land, by recognizing the international standard helipad pattern (circle sur-rounding an H). Previous algorithms did not take into consideration image blur, irrespective of whether it was caused by weather conditions or by a foreign material covering the vision camera lens. This issue was addressed a few years later in 2016 in Zeng et al. [13] work. Their strategy for unblurring the image was to use the Weiner Filter. A monocular vision-based real-time target recognition was proposed by Lin [14]. In it, the helipad contours are used to perform point feature mapping and clustering. The algorithm can recognize the international landing target in a cluttered environment, by the four degree of freedom pose of the drone with respect to the helipad. Later that year, Cosimo conducted another helipad detection algorithm for accurate UAV pose estimation through a visual sensor [15]. Pose estimation has proven to be effective in many fields, but it stands out particularly in self-driving vehicles. The core component of a self-driving car is its ability to detect, identify, and process what its visual sensors perceive. Engaging these pose estimation techniques, these smart vehicles should be able to assess the difference between a person walking, running, or on a bicycle, and use this information accordingly. The main contribution of this work is its simplification of the process, allowing implementation and precise pose estimation.

3. Proposing Intelligent Helipad and LiDAR Camera Fusion (IHLCF)

A standard helipad is only composed of a circle, and the word "H" is located in the middle of the circle. However, the modified version of the helipad includes an arrow for better heading detection. While the symmetry of the standard helipad proposes no headings, the modified version only includes a single direction for heading. Figures 1(a) and 1(b) demonstrate a standard helipad pattern with no heading and modified versions.

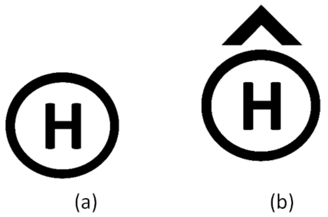


Figure 1. Standard helipad (a), proposing helipad.

IHLCF is a method for the modified version of the helipad's pose estimation, comprising preprocessing, detection, data fusion, and estimation. Figure 2 depicts the complete overview of the algorithm.

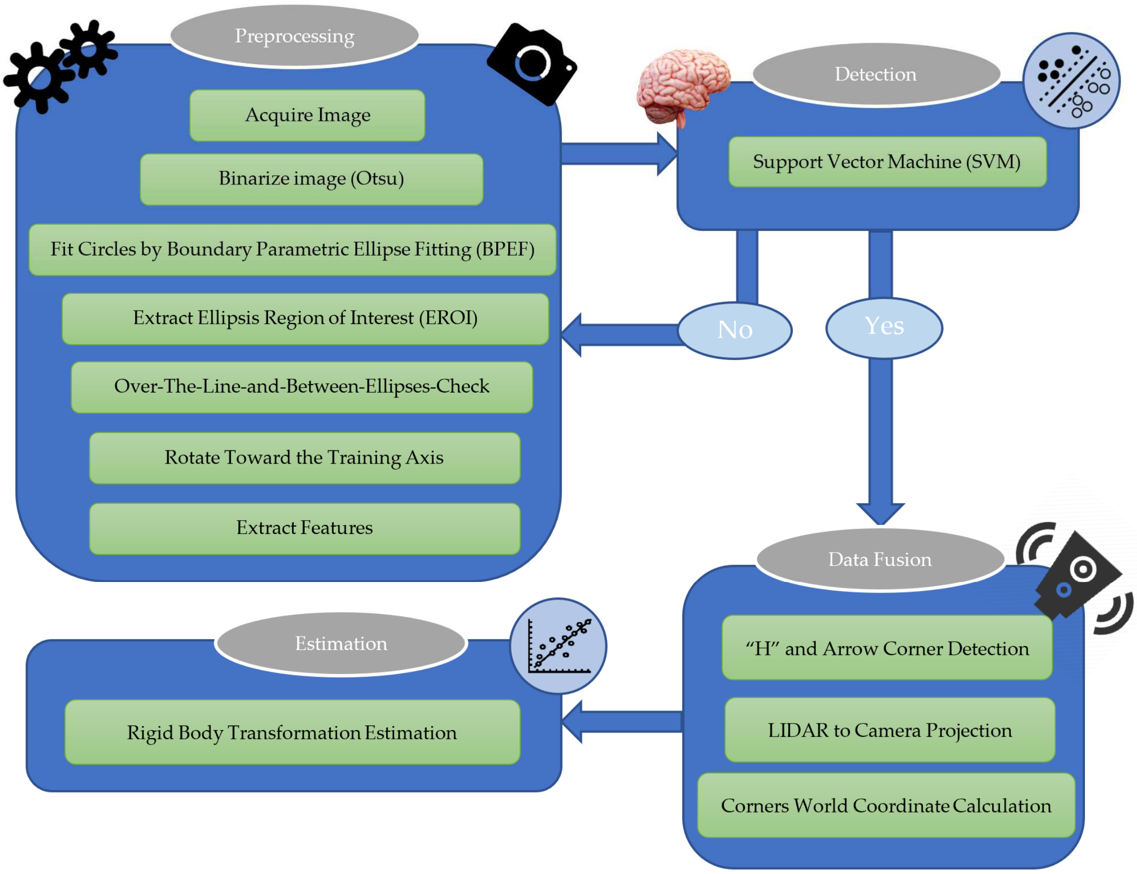

Preprocessing
Acquire Image
Binarize image (Otsu)
Fit Circles by Boundary Parametric Ellipse Fitting (BPEF)
Extract Ellipsis Region of Interest (EROI)
Over-The-Line-and-Between-Ellipses-Check
Rotate Toward the Training Axis
Extract Features
Detection
Support Vector Machine (SVM)
No
Yes
Data Fusion
"H" and Arrow Corner Detection
LIDAR to Camera Projection
Corners World Coordinate Calculation
Estimation
Rigid Body Transformation Estimation

Figure 2. Intelligent Detection and LiDAR Camera Fusion (IHLCF) overview.

3.1. Preprocessing

Preprocessing step is essential in advance of the detection step to extract the required features for the detection part.

3.1.1. Otsu Binarization

Acquired Images are presented in RGB format with three matrices for red, green, and blue colors' intensities. To binarize an image, first, it is essential to convert the image from RGB to grayscale.

3.1.2. Boundary Parametric Ellipse Fitting (BPEF)

In this phase, the circle of the scene is extracted. To implement BPEF, first, Euler number filtering is used to ensure that the hole in blobs is zero and one. Also, the suitable blob area for the current algorithm is taken between 30 and 60,000. Next, the boundaries of the binary blobs in the image are extracted. In the next phase of the calculation, the parametric equation of the rotated ellipse is considered as equation (1).

$$(\frac{(u-c_1)\times cos(a_3)+(v-c_2)\times sin(a_3)}{a_1})^2+(\frac{(u-c_1)\times sin(a_3)-(v-c_2)\times cos(a_3)}{a_2})^2=1 \quad (1)$$

The equation is converted to the optimization problem to validate the ellipse boundary. The parametric ellipse equation is defined as equation (1). Nonlinear least square is used to project the equation (2).

$$f_{\text{ellipse}}(a_1,a_2,a_3)=(\frac{(u-c_1)\times cos(a_3)+(v-c_2)\times sin(a_3)}{a_1})^2+(\frac{(u-c_1)\times sin(a_3)-(v-c_2)\times cos(a_3)}{a_2})^2-1 \quad (2)$$

$$f_{\text{ellipse}}(a_1,a_2,a_3) \quad (3)$$

where $a_1$is before-rotation ellipse axis in *v* direction, a_2 is before-rotation ellipse axis in *u* direction, and a_3 is the angle of rotation. c_1 and c_2 are the ellipse centroid coordinate address. The nonlinear least square method is used to solve Equation (3). Figure 3 depicts the ellipse plot and the concerning parameters.

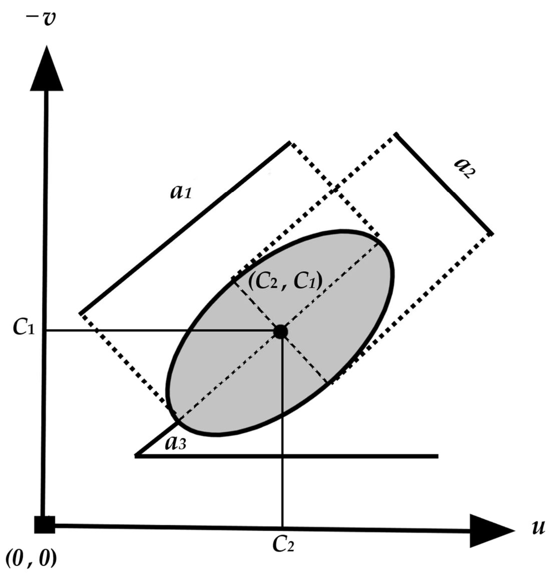


Figure 3. *Ellipse plot and parameters for parametric ellipse fitting.*

3.1.3. Ellipses Region of Interest (EROI) Extraction

To calculate the EROI two parameters namely, shrinkage factor (β) and number of sample points (n_{sp}) are considered. Equations (19)-(24) deliberates on finding EROI.

$$P_{EROI_i}=(1+\beta)\times(p_{b_i}-p_c)+p_c \quad (4)$$

$$SP_{EROI_i}=\left\lfloor\frac{i_{EROI}}{i_S}\right\rfloor, 1\le i_{EROI}\le SZ_B \& n_{sp}\le i_S\le SZ_B \quad (5)$$

$$ZM=Zeros(SZ_{img}) \quad (6)$$

$$SP_{MASK}=SUB(SP_{EROI_i},1) \quad (7)$$

$$EROI_{MASK}=CONV(SP_{MASK}) \quad (8)$$

$$EROI=EROI_{MASK}\odot Oimg \quad (9)$$

ZM, SZ_{img}, SP_{EROI_i}, SP_{MASK}, $EROI_{MASK}$, $Oimg$, and $EROI$ are a zero mask, original image size, sampled points boundary for $EROI$, sample points mask by substituting one in zero mask, mask for $EROI$, the original image, and the image with Ellipses Region of Interest, respectively. Also, $\odot$ is Hadamard product notation, $CONV$ is convex hull notation, $\lfloor\ \rfloor$ is rounding notation, and SUB is pixel substitution.

3.1.4. Over-the-Line-and-Between-Ellipses-Check (OLBEC)

This is an essential step to make certain that the centroid of the potential arrow point is located over the axis developed by the current technique. OLBEC concept is depicted in the Figure 4.

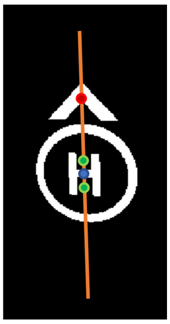


Figure 4. *OLBEC presentation.*

To calculate the equation of the line represented by orange in Figure 4, the algorithm undergoes below procedures:

1. logical negation of the labeled blob of the word "H";
2. Extraction of two centroids from the negated blob; and
3. Using the two points found in 2 and the word "H" centroid to find the orange line equation: least square algorithm is employed to find the line equation.

Above procedures makes sure that the arrow is located on the helipad OLBEC line.

Each point on the *u*-*v* plane has three statuses, namely, out of the ellipse, inside the ellipse, and on the ellipse. Following two Equations (10) - (12) determine each centroid location.

$$f_{\text{ellipse}}(v, u, a_3) < 0 \tag{10}$$

$$f_{\text{ellipse}}(v, u, a_3) > 0 \tag{11}$$

$$f_{\text{ellipse}}(v, u, a_3) = 0 \tag{12}$$

The whole centroids are checked to confirm the primary helipad detection. From all centroids in the scene, only two must satisfy equation (10) because circle and letter "H" centroids are inside the ellipse. At least one centroid must satisfy the equation (11) since the arrow centroid is outside the ellipse. Points that are accepted in Equation (12) are located on the boundary of the ellipse. These Equations also guarantee that the arrow centroid is located between ellipses, namely EROI and the helipad ellipse.

3.1.5. Rotate Toward the Training Axis

It is essential to rotate the helipad binary image toward a fixed axis to have a stable feature. Here training axis is formed by the helipad circle centroid and the same point with zero v; hence training axis is variable but parallel to the v axis of the image plane. Figure 5 depicts the training axis in the image plane.

In the final step, the image is rotated with the angle of $-\theta_{HA}$. θ_{HA} is the angle between helipad axis and training axis, represented by green line and blue line in Figure 5 respectively.

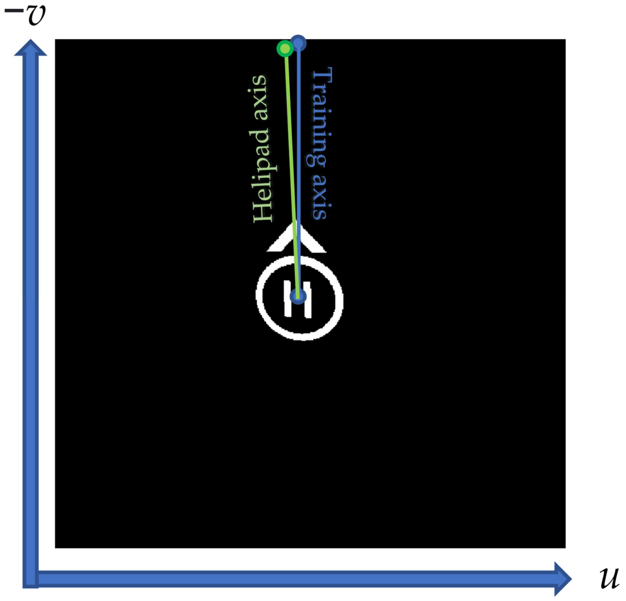


Figure 5. *Training axis, in blue line, and helipad axis, in green line.*

3.1.6. Feature Extraction

In this phase of the methodology following arrow and the word "H" are extracted from the image and resized to one sixth of the original image. Figure 6 shows this rescaling.

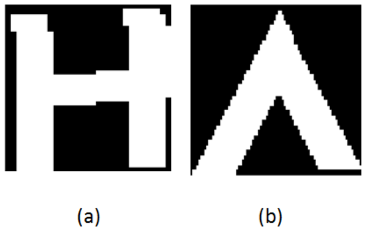


Figure 6. *Rotated and rescaled version of the H and the arrow presented in (a) and (b) respectively.*

The feature vector is the extracted from the H and potential arrows. These features are represented in a vector format in Equation (13).

$$F_{vect} = (F_{Ed}, F_{Sol}, F_E, F_{EC}) \tag{13}$$

$F_{vect}, F_{Ed}, F_{Sol}, F_E, F_{EC}$ are are the training feature vector, equivalent diameter, solidity, extent, and eccentricity, respectively.

3.2. Detection

This part explores the procedures for the detection of the modified helipad. The classifier must be trained once; then, it can be used for detection. Support Vector Machine is used as a detector. The data were trained over 200 images of the arrow and 200 images of the word "H" and tested with 200 different images. The test of the trained SVM model issued 100% of accuracy for the binary classification of the word" H" and the arrow images, with one sixth size of the original image.

3.3. Data Fusion

This section elaborates on procedures for combining data from the camera and LiDAR. The data fusion phase follows three steps as below.

1. "H" and arrow corner detection;
2. LiDAR to Camera Projection; and
3. Corners' world coordinate calculation.

In the first step, the Harris corner detection algorithm and is employed to localize the H and arrow blobs in the image plane. By a sliding window and image gradient, the Harris-based corners are detected. Since the camera and LiDAR are in the real world, the rigid body transformation is required for the projection calculation. Hence, in the next step, LiDAR to camera projection is conducted. Equations (14) and (15) deliberates the procedures for calculating the projection.

$$P_{LC} = P_L \times R_{LC} + T_{LC} \tag{14}$$

$$pp_{LC} = C \times P_{LC} \tag{15}$$

where R_{LC}, T_{LC}, P_{LC}, pp_{LC} and C are rotation matrix of LiDAR to the camera, translation matrix of LiDAR to the camera, point clouds of LiDAR in the camera frame, projected point clouds in image frame, and camera matrix respectively. R_{LC} and T_{LC} are related to the system's design and are taken as Equations (16) and (17).

$$R_{LC} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (16)$$

$$T_{LC} = (-0.15 \quad 0.05 \quad 0.16) \quad (17)$$

In the final part of the data fusion, part 3, the nearest neighbor method finds the nearest projected points pp_{LC} to the helipad corners to find the helipad corners' world coordinates.

3.4. Rigid Body Transformation Estimation

Corner points' real-world coordinates are taken as the second point sets to calculate the rigid body transformation. Then real corners' measurements of the modified helipad (modified helipad dimensions) are the first points sets. The estimation algorithm calculates the transformation between these two sets of points. Rigid body transformation is estimated by these two sets of points and the Singular Value Decomposition Algorithm (SVD) technique.

4. Empirical Setup

The research is conducted in the Software-in-the-Loop (SIL). CoppeliaSim API framework transfers image and LiDAR data from the simulator to MATLAB in real-time. The simulation time step is selected as the default timing, 50 ms. Figure 7 depicts the interaction between MATLAB and CoppeliaSim environment in real-time.

In addition, there are five primary coordinate frames in the simulation environments. Figure 8 shows these frames. The first frame is the world coordinate, and the rest of the frames are calculated relative to this system. While the second frame is the drone frame, the same as the camera frame, and the third system belongs to the helipad. The fourth system is pixel frame, with coordinates of u and v. Finally, the fifth system is for LiDAR, which has x_l and y directions.

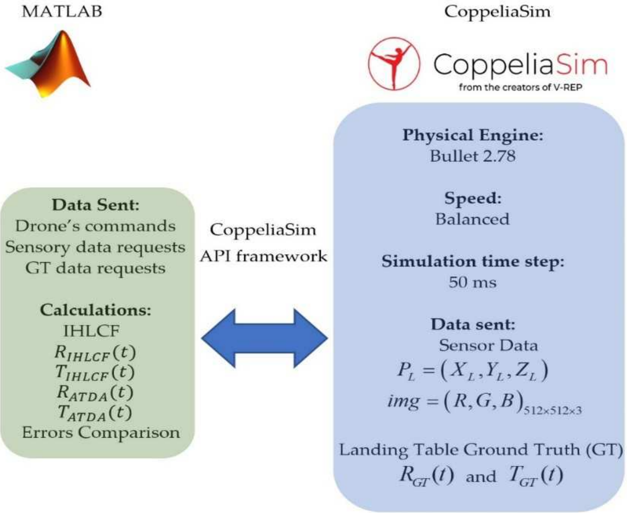


Figure 7. *SIL interaction between MATLAB and CoppeliaSim details.*

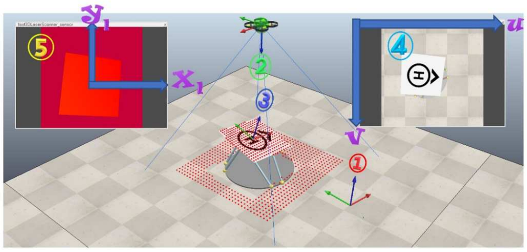
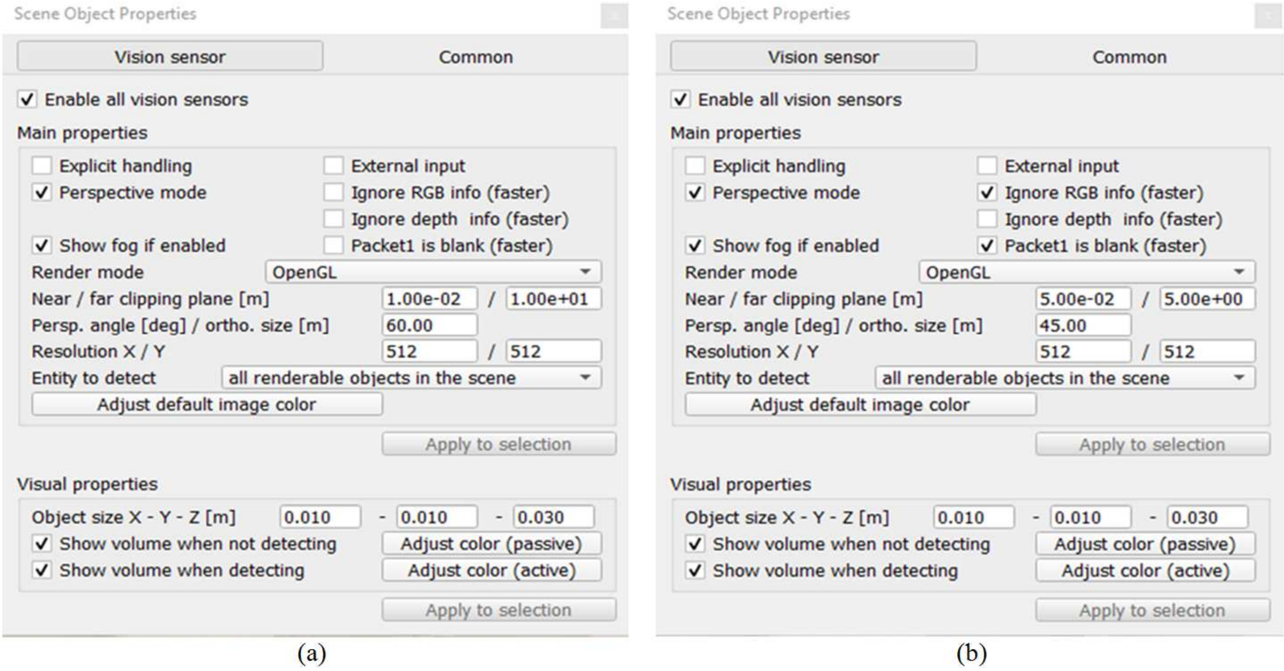


Figure 8. *Employed coordinates system in the simulation.*



(a)

Scene Object Properties

Vision sensor | Common

☑ Enable all vision sensors

Main properties

☐ Explicit handling
☑ Perspective mode
☑ Show fog if enabled
☐ External input
☐ Ignore RGB info (faster)
☐ Ignore depth info (faster)
☐ Packet1 is blank (faster)
Render mode: OpenGL
Near / far clipping plane [m]: 1.00e-02 / 1.00e+01
Persp. angle [deg] / ortho. size [m]: 60.00
Resolution X / Y: 512 / 512
Entity to detect: all renderable objects in the scene
Adjust default image color
Apply to selection

Visual properties

Object size X - Y - Z [m]: 0.010 - 0.010 - 0.030
☑ Show volume when not detecting
☑ Show volume when detecting
Adjust color (passive)
Adjust color (active)
Apply to selection

(b)

Scene Object Properties

Vision sensor | Common

☑ Enable all vision sensors

Main properties

☐ Explicit handling
☑ Perspective mode
☑ Show fog if enabled
☐ External input
☑ Ignore RGB info (faster)
☐ Ignore depth info (faster)
☑ Packet1 is blank (faster)
Render mode: OpenGL
Near / far clipping plane [m]: 5.00e-02 / 5.00e+00
Persp. angle [deg] / ortho. size [m]: 45.00
Resolution X / Y: 512 / 512
Entity to detect: all renderable objects in the scene
Adjust default image color
Apply to selection

Visual properties

Object size X - Y - Z [m]: 0.010 - 0.010 - 0.030
☑ Show volume when not detecting
☑ Show volume when detecting
Adjust color (passive)
Adjust color (active)
Apply to selection

Figure 9. *The camera (a) and the LiDAR (b) settings.*

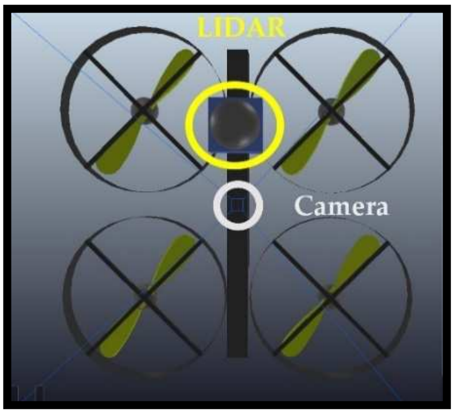


Figure 10. Drone sensor installations and arrangements.

The sensor's parameters settings are illustrated in Figure 9. Figure 9(a) belongs to the camera parameters, while Figure 9(b) is for the LiDAR settings.

This experiment has two primary sensors, including a camera and LiDAR, with the layout demonstrated in Figure 10.

5. Results and Discussion

The IHLCF is compared with ATDA for translational and rotational components to evaluate the algorithm's viability. Figure 11 illustrates the translational calculations and ground truth. It is apparent from the figure that the LiDAR range data significantly ameliorates the translational components in the X, Y, and Z directions. The most notable error is in the Z – coordinate of ATDA during all simulation periods disregarding the initial results. The following highest displacement estimation error is for ATDA in the Y-direction. However, the X-direction shows a lower error than the other ATDA translational estimations.

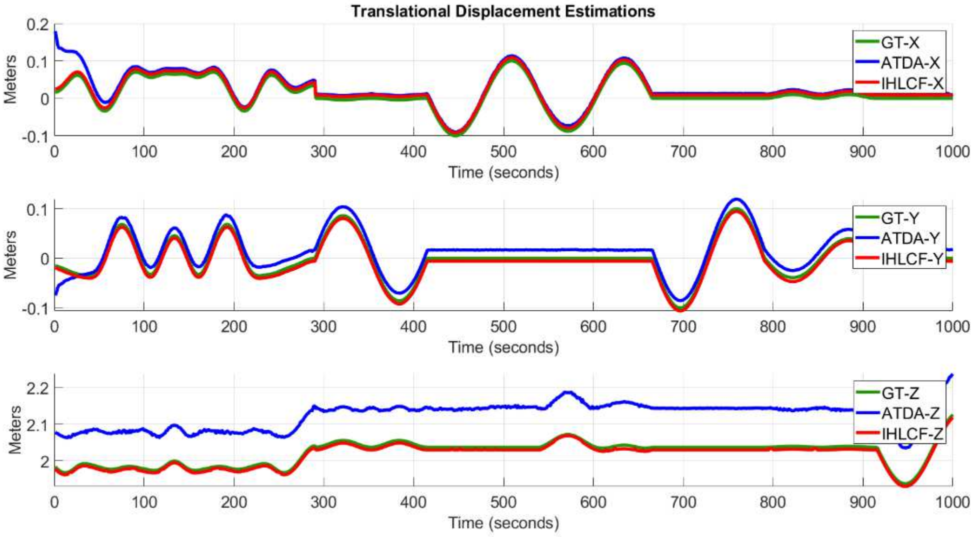


Figure 11. Translational estimation for GT (green), ATDA (blue), and IHLCF (red) – For Z, Y, and Z.

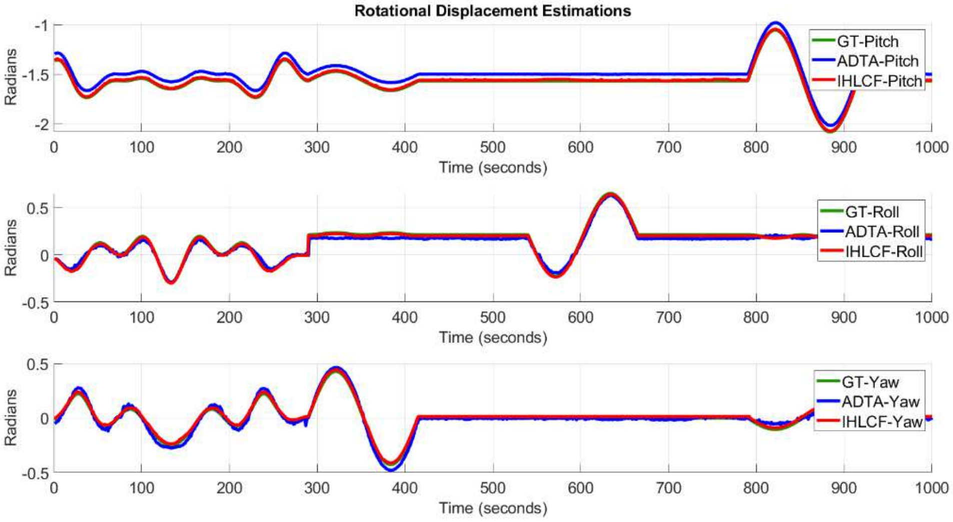


Figure 12. Translational estimation for GT (green), ATDA (blue), and IHLCF (red) - translational estimation for GT (green), ATDA (blue), and IHLCF (red) – For pitch, roll, and yaw.

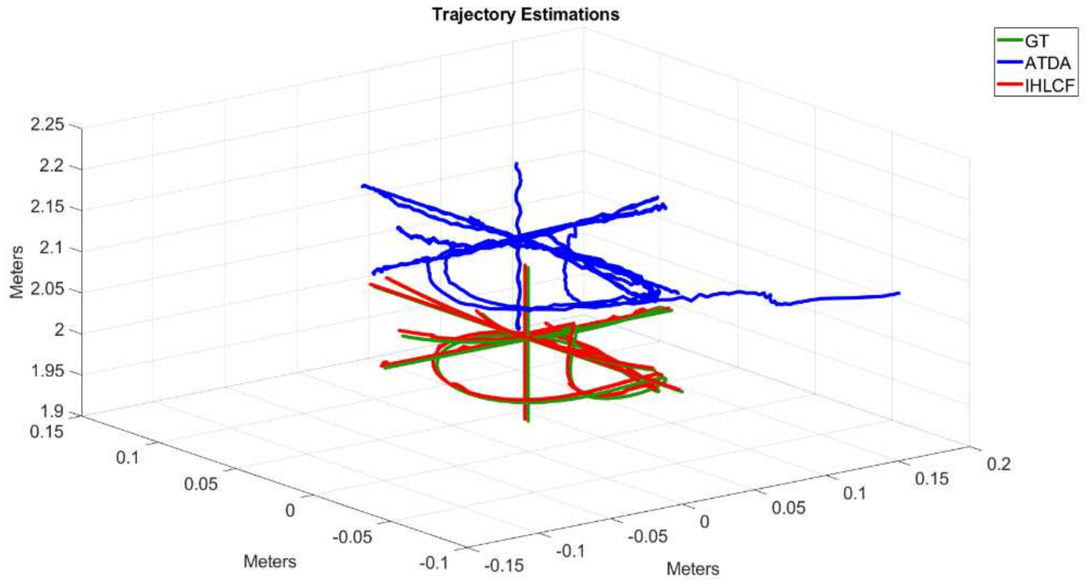


Figure 13. *Trajectory (3D) plot for GT (green), ATDA (blue), and IHLCF (red).*

Next, IHLCF is tested for rotational angles, presented in Euler angles, estimations. Figure 12 demonstrates the rotational estimation results. While the IHLCF algorithm proves feasible in translational estimation, it slightly fluctuations over the GT for Euler angles estimation. ATDA demonstrates remarkable error during all periods in pitch estimation, although it proposed a better roll tracking. In addition, the highest yaw estimation error is for ATDA at 380 seconds of simulation.

The trajectory plot of GT, ATDA, and IHLCF is demonstrated in Figure 13. It is transparent from the plot that IHLCF proposed a better trajectory tracking. ATDA demonstrates a considerable inaccuracy for altitude estimation, confirming the suitability of the LiDAR range data inclusion for translational estimations correction. Equation (18) describes the Mean Absolute Errors (MAEs) calculation over the simulation period.

$$\text{MAEs} = \frac{\sum_{t=t_1}^{t_n} |y_t - \tilde{y}_t|}{n} = \frac{\sum_{t=t_1}^{t_n} |e_t|}{n} \qquad (18)$$

y_t is the correct value, $\tilde{y}_t$ is the estimated value, and e_t is the error at the time "t". The mean absolute errors (MAEs) of the proposing IHLC and ATDA are summarized and compared in Table 1.

Table 1. *Summary of IHLCF and ATDA MAEs.*

Name	**Error**	
	Translational absolute error (m) (X, Y, Z)	**Rotational absolute error (rad) (pitch, roll, yaw)**
ATDA	0.0210, 0.0168, 0.1027	0.0708, 0.0357, 0.0271
IHLCF	0.0033, 0.0037, 0.0039	0.0162, 0.0228, 0.0196

From Table 1, MAEs errors are negligible compared to the ATDA algorithm. This is because the low-angle, low-density LiDAR provides range data which is beneficial to the correction of the pose estimation inaccuracies. These improvements are more prominent in the translational pose estimation. While low-angle, low-density LiDAR is less expensive and light-weight, it is a suitable sensor for camera-based sensor fusion for landing system design.

6. Conclusions and Future Works

In this work, the helipad pose estimation problem was addressed via modifying an arrow for heading estimation and including LiDAR range data. The main inspiration behind this research is to develop a system for aerial vehicles that can estimate the landing surface attitude and position. The application of the method would be beneficial, especially for rotary aerial vehicles' emergency landing in challenging situations such as marine environments. After the image data acquisition, the algorithm converts the image from RGB to grayscale and binary images. Then, developed State of the Art (SOTA) BPEF is used to extract the circles from the scene. Next, two measures, including EROI, over-the-line-check-and-between-ellipses-check, are taken to eliminate unnecessary candidates' binary blobs. In the following stage, the helipad binary pattern is rotated. Also, relevant features, namely, equivalent diameter, solidity, extent, and eccentricity, are extracted from the blobs. Next, the detection is conducted utilizing SVM. The H and arrow corners are extracted and localized if the helipad is detected. In the following step, the data from the LiDAR are projected into the image plane. Then, the nearest projected point clouds points toward the H and arrow corners are found via nearest neighbor search in Cartesian coordinates of the camera frame. In the final stage, the pose of the helipad is estimated through the SVD algorithm. The empirical findings support a noticeable improvement in the pose estimation by modifying the helipad and integrating a LiDAR sensor. In addition, in this study, we used a low density, low angle LiDAR, which substantially reduces the computational and cost expenses compared to when high density, high angle LiDAR is utilized.

There are several propositions for future works. First, in this work, the IHLCF method is trained over the image with lower computation; however, we also recommend training deep learning or machine learning model on 3D LiDAR data and fusing the model with image data. In this manner, problems of camera calibration inaccuracies are rectified. Next, the outstanding potential of Artificial Intelligence (AI) in many areas has been confirmed over the most recent research, and complex landing system development is not an exception.

Reinforcement learning is one of these methods that use an agent, which behaves like a human that gains experience in an environment, representing the world. This experience acquisition process is how the agent interacts with the environment and obtains acting knowledge through being rewarded or penalized. In addition, a State of the Art (SOTA) technique based on the imitation learning method can also be used for the landing task learning. In this type of learning, a human supervisor teaches the robots to conduct a task.

References

[1] Cui, X.-Z., Feng, Q., Wang, S.-Z., & Zhang, J.-H. (2022). Monocular Depth Estimation with Self-Supervised Learning for Vineyard Unmanned Agricultural Vehicle. Sensors, 22 (3), 721. https://doi.org/10.3390/s22030721

[2] Zhang, Z., Zhou, C., Koike, Y., & Li, J. (2022). Single RGB Image 6D Object Grasping System Using Pixel-Wise Voting Network. Micromachines, 13 (2), 293. https://doi.org/10.3390/mi13020293

[3] Li, M., Chen, R., Liao, X., Guo, B., Zhang, W., & Guo, G. (2020). A Precise Indoor Visual Positioning Approach Using a Built Image Feature Database and Single User Image from Smartphone Cameras. Remote Sensing, 12 (5), 869. https://doi.org/10.3390/rs12050869

[4] Basan, E., Basan, A., Nekrasov, A., Fidge, C., Sushkin, N., & Peskova, O. (2021). GPS-Spoofing Attack Detection Technology for UAVs Based on Kullback–Leibler Divergence. Drones, 6 (1), 8. https://doi.org/10.3390/drones6010008

[5] Sefidgar, M., & Landry, R. (2022). Unstable Landing Platform Pose Estimation Based on Camera and Range Sensor Homogeneous Fusion (CRHF). Drones, 6 (3), 60. https://doi.org/10.3390/drones6030060

[6] Sefidgar, M., & Landry, R. (2022). Landing System Development Based on Inverse Homography Range Camera Fusion (IHRCF). Sensors, 22 (5), 1870. https://doi.org/10.3390/s22051870

[7] G. F. Nsogo, K. Kith, B. J. van Wyk, and M. A. van Wyk, “Robust Helipad Detection Algorithm (January 2007),” AFRICON 2007, Sep. 2007, doi: 10.1109/afrcon.2007.4401634.

[8] C. S. Sharp, O. Shakernia, and S. S. Sastry, “A vision system for landing an unmanned aerial vehicle,” Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), 2021, doi: 10.1109/robot.2001.932859.

[9] S. Lee, J. Jang and K. Baek, "Implementation of Vision-Based Real Time Helipad Detection System," 2012 12th International Conference on Control, Automation and Systems, 2012, pp. 191-194.

[10] P. M. Fitts, R. E. Jones, J. L. Milton, and AIR MATERIEL COMMAND WRIGHT-PATTERSON AFB OH, “Eye Fixations of Aircraft Pilots. III. Frequency, Duration, and Sequence Fixations When Flying Air Force Ground-Controlled Approach System (GCA).,” DTIC, 2021. https://apps.dtic.mil/sti/citations/ADA329371 (accessed Aug. 25, 2021).

[11] S. Lange, N. Sunderhauf and P. Protzel, "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," 2009 International Conference on Advanced Robotics, 2009, pp. 1-6.

[12] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti and S. Longhi, "Vision-based autonomous navigation and landing of an unmanned aerial vehicle using natural landmarks," 2009 17th Mediterranean Conference on Control and Automation, 2009, pp. 910-915, doi: 10.1109/MED.2009.5164661.

[13] Zeng Fucen, Shi Haiqing and Wang Hong, "The object recognition and adaptive threshold selection in the vision system for landing an Unmanned Aerial Vehicle," 2009 International Conference on Information and Automation, 2009, pp. 117-122, doi: 10.1109/ICINFA.2009.5204904.

[14] A. Cesetti, Emanuele Frontoni, Adriano Mancini, and Sauro Longhi, “A single-camera feature-based vision system for helicopter autonomous landing,” ResearchGate, Jul. 26, 2009. https://www.researchgate.net/publication/224567916_A_single-camera_feature-based_vision_system_for_helicopter_autonomous_landing (accessed Aug. 25, 2021).

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-Up Robust Features (SURF),” Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346–359, Jun. 2008, doi: 10.1016/j.cviu.2007.09.014.