Learning-Type Anchors-Driven Pose Estimation for the Autolanding Fixed-Wing UAVs

Dengqing Tang, Lincheng Shen, *Member, IEEE*, Xiaojia Xiang, Han Zhou, and Tianjiang Hu*

Abstract—We propose a learning-type anchors-driven real-time pose estimation method for the autolanding fixed-wing unmanned aerial vehicle (UAV). The proposed method enables online tracking of both position and attitude by the ground stereo vision system in the Global Navigation Satellite System denied environments. A pipeline of convolutional neural network (CNN)-based UAV anchors detection and anchors-driven UAV pose estimation are employed. To realize robust and accurate anchors detection, we design and implement a Block-CNN architecture to reduce the impact of the outliers. With the basis of the anchors, monocular and stereo vision-based filters are established to update the UAV position and attitude. To expand the training dataset without extra outdoor experiments, we develop a parallel system containing the outdoor and simulated systems with the same configuration. Simulated and outdoor experiments are performed to demonstrate the remarkable pose estimation accuracy improvement compared with the conventional Perspective-N-Points solution. In addition, the experiments also validate the feasibility of the proposed architecture and algorithm in terms of the accuracy and real-time capability requirements for fixed-wing autolanding UAVs.

Index Terms—Pose estimation, autolanding fixed-wing UAVs, ground stereo vision system, block convolutional neural networks

I. INTRODUCTION

THE unmanned aerial vehicle (UAV)-based systems have recently emerged as efficient solutions in various application areas. Autonomous landing ability is one of the most essential functions in almost all UAV applications. Nevertheless, this function still faces major technical challenges. In addition to the control which is a difficult problem [1], the state estimation of the aircraft is also a challenging task [2], especially in Global Navigation Satellite System (GNSS)-denied or GNSS-inaccurate environments. Usually, extra sensors such as inertial navigation systems (INS) and visual navigation systems are adopted to aid the UAV state estimation during the landing [3]. Classical INS are however sensitive to the magnetic fields in the environment and its temperatures. Alternatively, the visual navigation system can be used that can also avoid drifting by combining prior information. By introducing a vision system, more rich environmental information is obtained, and it makes more robust self-state estimation possible [4]. Using an onboard or offboard camera, the vision navigation system provides UAVs

with the accurate and real-time self-states, such as visual odometry (VO) [5], visual simultaneous localization and mapping (VSLAM) [6]. Therefore, visual navigation is emerging as a viable autolanding solution due to its intrinsic ability to incorporate rich environmental information.

For the autonomous landing of the rotor-wing aircraft, the vehicle's pose is often estimated by detecting the markers painted on the static platform utilizing an onboard camera [7]. For more complex scenarios such as landing on ship deck, Wang *et al.* [8] realized autonomous landing of Parrot AR Drone on a vessel deck platform only relying on onboard sensors. They simulated the movement of ship deck by an attitude-programmable plate. Landing on a moving target [9] is more challenging compared with ship landing, in terms of localization, trajectory planning, and control. For the fixed-wing UAVs, however, it is challenging to track the ground marker throughout the entire landing process. This is because, unlike the rotor-wing UAVs, the fixed-wing UAVs are unable to hover. Landing the fixed-wing UAVs is further challenged [10] because even small errors in the guidance system may lead to system damage. The onboard vision was often used to detect the runway [11] and to estimate the relative UAV pose to the runway for autonomous landing. Without runway detection, the landmarks on the platform were tracked by the aircraft to provide information of the relative poses [12].

An alternative to the onboard vision-based systems is the ground vision-based guidance systems. It estimates the UAV pose and the pose data is then transmitted to the UAV for autolanding control. Generally, the ground systems are equipped with a vast computational capacity which enables real-time pose estimation. In addition, using ground-based systems also reduces the load of onboard processing resource, which is often limited especially for small aircrafts. Furthermore, onboard systems such as the barometer and inertial measurement unit (IMU) are significantly sensitive to temperature variations. The accuracy of the estimated UAV pose using the ground-based vision is also unaffected by the temperature variations.

Y. Gui [13] proposed a relay guidance scheme to land the UAV by placing three groups of cameras on both sides of the runway so that the total field of view of the cameras covers the whole landing area. The *AUTOLAND* project [14] focused on the solutions that enable the autonomous landing of a fixed-wing aircraft on a Fast Patrol Boat. The ground monocular vision system has been tested to generate the relative pose of the aircraft with respect to the camera. Relying

This work was supported in part by the National Natural Science Foundation of China under Grants 61703418 and 61973327. (*Corresponding author: Tianjiang Hu*).

on the ground stereo vision system, a saliency-inspired method [15] and a cascaded deep learning model [16] were proposed and developed to detect and track the aircraft in the images, and then used the Extended Kalman Filter (EKF) proposed in our previous work [17] to estimate the position of the aircraft. Paying attention to the ground stereo vision-based fixed-wing aircraft detection and localization for autonomous landing, we design a ground stereo vision guidance system for validation. Offline and online experiments demonstrate the feasibility and robustness of the proposed system [17], [17]. Only focusing on the 3D position estimation of the aircraft, these works explored vision-based guidance system schemes including monocular vision and infrared stereo vision.

In addition to the position, attitude also plays an important role in the landing guidance and controls system. For accurate attitude estimation, the Inertial Navigation System (INS) is commonly used [19]. Aided by the onboard inertial sensors, Yang *et al.* [19] proposed a bioinspired polarization-based attitude and heading reference system to self-determine the heading orientation in the GNSS-denied environments. However, the INS measurement is often affected by internal or external factors such as drift, magnetic field, and temperature variation [20]. To improve the robustness to environmental factors, the vision system is well-applied [5], [21]. To realize the tanker-UAV relative pose estimation during aerial refueling, Mammarell *et al.* [21] combined GPS and a machine vision-based system for a reliable estimation, where at least one order of magnitude improvement was achieved by using the EKF instead of other fusion algorithms.

In this paper, inspired by the solutions of the pose estimation for humans [22], human head [23], and rigid objects [24], we devise methods for the UAV 6D pose estimation with high accuracy, strong robustness, and real-time capability for autonomous landing guidance based on the ground vision system. A monocular vision-based solution and a stereo vision-based solution are presented in this paper. Similar to our work, Liu *et al.* [25] proposed an attitude estimation method for fixed-wing aircraft also using a ground vision system and validated the accuracy and real-time capability by experiments. In comparison, in addition to the attitude, our method can estimate the position and attitude jointly. Besides, in our outdoor experiments, the UAV realized successful autonomous landing in both GPS- and INS-denied environments, which means the onboard autonomous landing control system only relied on our estimation of UAV poses.

Compared with our previous work [17], which only focuses on the position estimation of the UAV, this paper aims to realize a joint estimation for the position and attitude that also plays an important role in the landing. In summary, the contributions of this paper are as the following.

1) *Robust and Accurate Pose Estimation*: During autonomous landing, the high-speed of the UAV in the three-dimensional space leads to rapid changes of the appearance, imaging background, etc. These temporal and spatial changes bring significant challenges for the ground vision-based UAV pose estimation, especially for the spatial position and attitude estimation. The proposed algorithm addresses these

issues and enables accurate and robust joint estimation of the UAV position and attitude, and was validated by the simulated and outdoor experiments.

2) *Training Data Augmentation*: The availability of a large amount of data for training is essential for the performance of neural networks. It is however almost impossible to collect the required data solely based on real experiments, e.g., due to the cost and challenging conditions. To address this issue, we build a simulated system with the same configuration as the outdoor system, and collect auto-labelled data under different environmental conditions for augmentation of the training dataset. The experiment results confirm the efficiency of this method.

3) *Accurate and robust anchor detection*: We propose a block convolutional neural network (Block-CNN), which greatly improves the detection accuracy and robustness by repeated anchors detection in several divided blocks and score-weighting of the detected anchors.

This paper is organized as the following. In Section 2 we present the problem of UAV pose estimation based on ground vision. The module details

including CNN training, anchors detection, and 6D pose estimation, are then presented in Section 3. In Section 4, three simulated experiments and two outdoor experiments are presented to validate the feasibility, real-time capability, the robustness of the proposed pose estimation method. The paper is finally concluded in Section 5.

II. PROBLEM FORMULATION

Accurate pose estimation and flight control are the main two challenges for the autonomous landing of UAVs. This paper focuses on the first component, which is estimating the UAV poses $\{P_u, A_u\}$ with high accuracy and strong robustness according to the sensor data. The sensor data includes the images I and pan-tilt-unit (PTU) attitude A_p . In general, the complete procedure of the pose estimation includes obtaining the object ROI first by object detecting, followed by detecting the object features and estimating the object poses. The procedure of the proposed pose estimation algorithm is summarized as the following mapping:

$$\{\text{ROI}, A_p, P_a\} \rightarrow \{P_u, A_u\} \quad (1)$$

where P_a denotes the system parameters gained by offline calibration.

In computer vision, the common extracted features are points, lines, and planes. During the autoland, there are several points of the UAV which are always in the field-of-the-view and remarkably distinctive. Since the anchors' distribution has a significant impact on the pose estimation accuracy, the selection of the anchors considers their characteristics and distribution (for details see, Section VI). Here, we consider the following five UAV anchors: the endpoints of the left-wing (LW), right-wing (RW), left tail (LT), right tail (RT), and the front tripod (FT) (see Fig. 1).

We divide the UAV pose estimation into three following operators: CNN training operator $FT(\cdot)$, anchors detection

Fig.1 The diagram of the proposed algorithm composed of the training operators $FT(\cdot)$, anchors detection operator $FD(\cdot)$, and pose estimation operator $FE(\cdot)$. The right trajectory plot shows the involved coordinate frames and three main periods for the autolanding of the UAV.

operator $FD(\cdot)$, and 6D pose filtering operator $FE(\cdot)$:

$$FT(\cdot) : \{D_T, N\} \rightarrow N_D \quad (2)$$

$$FD(\cdot) : \{ROI, N_D\} \rightarrow An \quad (3)$$

$$FE(\cdot) : \{An, A_p, P_a\} \rightarrow \{P_u, A_u\} \quad (4)$$

As depicted in Fig. 1, through offline training based on the training dataset D_T , the initial network N evolves to be the network N_D , which gains the ability to detect the anchors. The operator $FD(\cdot)$ detects the anchors and obtains the anchor locations An in the images I . This is mainly done by the anchor detection network N_D . The final operator $FE(\cdot)$ estimates the UAV poses $\{P_u, A_u\}$ according to the anchor locations An , system parameters P_a and real-time PTU attitude A_p .

The right part of Fig.1 displays the involved coordinate frames in the autonomous landing system. The objective of the proposed algorithm is to estimate the quickly varied transformation between the world coordinate frame and UAV body coordinate frame. For the left and right PTU coordinate frames, their origins in the world coordinate frame are known and constant. Since the camera is fixed on the PTU, the transformation between the camera coordinate frame and corresponding PTU coordinate frame is also constant.

III. METHODOLOGY

A. Anchors Detection Operator $FD(\cdot)$

Anchors detection is the core of the proposed pose estimation algorithm and its accuracy directly affects the pose estimation accuracy. Conventional anchors detection methods are often based on the classical feature point such as Scale Invariant Feature Transform (SIFT) [26] and Oriented FAST and Rotated BRIEF (ORB) [27]. These handcrafted representations are however suboptimal compared to statistically learned features. The CNN learns the features, therefore, archives significant advantages in computer vision applications in terms of accuracy and robustness [28]. One of the disadvantages is that the training of the CNN is computationally expensive and often needs GPU like Compute Unified Device Architecture. This is however an essential disadvantage that limits the applications of CNNs [29]. In contrast, such computational resources can be easily made available on the ground for vision-based applications. This justifies the use of CNN for accurate and

robust anchors detection in ground vision-based systems.

Fig. 2 illustrates the designed CNN: F with a 39×39 input and a 10×1 vector output. The deeper the convolutional layer the greater the accuracy. In practice, the trade-off between accuracy and real-time capability needs to be incorporated into the final design. Considering the above factors, we consider 4 convolution layers and 1 fully connected layer. The output vector represents the positions of the five anchors in the image. Conventionally, the anchors' positions are estimated by the F and can be directly used for estimating the next pose. Here, to improve the detection accuracy, we introduce the block strategy. Through partitioning the ROI, more pixel details are preserved after resizing as the network input, which helps the network extract more useful features. On the other hand, block strategy enables the repeated detection for the same anchors. Using the averaging on the repeated detection results, the outliers' negative impact on detection accuracy is also reduced.

Fig. 2. The Block-CNN architecture for anchors detection. The ROI is cut into 3 sub-ROIs. The anchors are then detected in sub-ROIs using the networks with the same structure. The network consists of 4 convolutional and a fully connected layer. The final detection results are generated after the fusion.

Since each anchor was distributed in a relatively fixed region of the ROI, several blocks are obtained by dividing the ROI. Each block contains some of the five anchors. As shown in Fig. 2, two blocks are cut from the original ROI and then resized to be with the size of 39×39 . The L-Block contains the anchors LW , LT , and FT . Also, the anchors RW , RT , and FT are within the R-Block. Another two networks (L and R) have almost the same structure with network F , are used to detect part of the anchors in the R-Block and L-Block, respectively. Compared with F , the only difference of the structure is that the outputs

are 6-dimensional vectors. To promote the detection accuracy, the anchors' locations in the ROI are then obtained by computing the score-weighted average of the Block-CNN outputs as described below.

For the frame k , the final FT location (u_{FT}, v_{FT}) is the average of all the networks outputs:

$$(u_{FT}, v_{FT})_k = \frac{(u_{FT}^F, v_{FT}^F)_k + (u_{FT}^L, v_{FT}^L)_k + (u_{FT}^R, v_{FT}^R)_k}{p} \quad (5)$$

Since three networks are used in this paper, the value of p is 3. Another 4 anchor locations are also predicted first, according to the historical anchor locations at step $k-1$, hence

$$(u, v)_{k|k-1} = (u, v)_{k-1} + ((u_{FT}, v_{FT})_k - (u_{FT}, v_{FT})_{k-1}) \quad (6)$$

Therefore, the final LW location computing steps are:

$$\Delta F_{LW} = \|(u_{LW}^F, v_{LW}^F)_k - (u_{LW}, v_{LW})_{k|k-1}\| \quad (7)$$

$$\Delta L_{LW} = \|(u_{LW}^L, v_{LW}^L)_k - (u_{LW}, v_{LW})_{k|k-1}\| \quad (8)$$

$$(u_{LW}, v_{LW})_k = \frac{\Delta L_{LW} (u_{LW}^F, v_{LW}^F)_k + \Delta F_{LW} (u_{LW}^L, v_{LW}^L)_k}{\Delta F_{LW} + \Delta L_{LW}} \quad (9)$$

Another three anchor locations are also computed following the same steps as the above.

B. CNN Training Operator $FT(\cdot)$

This module consists of training data generation and network training. Generating the data in the simulation system significantly improves data production efficiency due to the high labor and time costs for conducting outdoor landing experiments. Considering the great simulation performance of Gazebo for UAV dynamic characteristics during landing, we construct a Gazebo-based simulation environment following the same configuration as the outdoor environment. Using this method, data under different conditions such as different weather and different landing path is efficiently generated. Furthermore, since the states of all objects, including UAV, PTU, camera, and involved parameters are known accurately in the simulation, the data supports autonomously labeling, and almost no manual labeling is needed.

Here we define the loss of network training by l_2 as the distance between the estimation with the ground truth. Before training, the samples are shuffled and every 10 samples were packed into a batch. We then train the network on a PC with 2 GPUs (GTX 3070). The network is trained with the maximum number of iterations of 40 k and an initial learning rate of 0.03, which is decreased by 10 at every 15 k iterations. Besides, we use the weight decay of 0.0001 in the training process.

C. 6D Pose Estimation Operator $FE(\cdot)$

This module aims to recover UAV spatial pose from several anchors. The Perspective-N-Points (PNP) problem solution is commonly used for the above problem. However, it does not consider the sensor data noise and takes advantage of the historical UAV states. To improve the robustness to the measurement error, such as the anchors' detection error, we establish an EKF to estimate the UAV position (x, y, z) and

attitude (Euler angle (ψ, ϕ, θ)) in the world coordinate frame.

Let state $\mathbf{x}$ be defined as:

$$\mathbf{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \psi, \phi, \theta, w_\psi, w_\phi, w_\theta]^T \quad (10)$$

where $(\dot{x}, \dot{y}, \dot{z})$ and $(w_\psi, w_\phi, w_\theta)$ are linear and angular velocities, respectively. The state $\mathbf{x}$ at step k is predicted by the process model F_k :

$$\bar{\mathbf{x}}_{k|k-1} = F_k \bar{\mathbf{x}}_{k-1|k-1} \quad (11)$$

$$F_k = \begin{bmatrix} I_{3 \times 3} & \Delta t & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & \Delta t \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (12)$$

where $\Delta t_{3 \times 3}$ denotes the 3×3 diagonal matrix with the diagonal element Δt defined as the time interval between steps $k-1$ and k . The state covariance matrix P is then obtained as:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + G_k Q_k G_k^T \quad (13)$$

The measurement $\mathbf{z}$ contains the detected anchor locations in the images captured by the two cameras:

$$\mathbf{z} = \left[\begin{bmatrix} u_{LW}^l & \vdots & u_{RT}^l \\ v_{LW}^l & \vdots & v_{RT}^l \\ 1 & \vdots & 1 \end{bmatrix}_{3 \times 5} \mid \begin{bmatrix} u_{LW}^r & \vdots & u_{RT}^r \\ v_{LW}^r & \vdots & v_{RT}^r \\ 1 & \vdots & 1 \end{bmatrix}_{3 \times 5} \right]_{3 \times 10} \quad (14)$$

where the upper index marks the left and right cameras. In terms of the measurement model $h(\cdot)$, the pinhole camera model is employed to project the anchors into the images according to the predicted state $\bar{\mathbf{x}}_{k|k-1}$:

$$\mathbf{z} = h(\bar{\mathbf{x}}_{k|k-1}) = \frac{1}{\lambda} N_c T_P^C T_W^P T_U^W P_U \quad (15)$$

where λ is a scaling factor and P_U is the anchor locations matrix in the UAV body coordinate frame. Matrix T indicates the homogeneous transformation between the two coordinate frames which is composed of the rotation matrix R and the translation vector $\mathbf{t}$:

$$T = \begin{bmatrix} R & \mathbf{t} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (16)$$

and N_c is the intrinsic matrix of the camera:

$$N_c = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (17)$$

where f , u_0 , v_0 , dx , and dy are the camera intrinsic parameters, which can be obtained in advance through offline calibration.

Due to the nonlinearity of the measurement model $h(\cdot)$, the Jacobian matrix $H(\cdot)$ is:

$$H_k = \frac{\partial h(\bar{\mathbf{x}}_{k|k-1})}{\partial \bar{\mathbf{x}}_{k|k-1}} \quad (18)$$

Hence, the Kalman gain K_k is:

$$S_k = H_k P_{k|k-1} H_k^T + R \quad (19)$$

$$K_k = P_{k|k-1} H_k^T (S_k)^{-1} \quad (20)$$

where R is the sensor's Gaussian noise covariance matrix for each measurement. The final step is to update the state $\mathbf{x}$:

$$\bar{\mathbf{x}}_{k|k} = \bar{\mathbf{x}}_{k|k-1} + K_k (\mathbf{z}_k - h(\bar{\mathbf{x}}_{k|k-1})) \quad (21)$$

Note that one camera is enough to obtain the pose estimation. To compare the monocular vision-based solution and the stereo vision-based solution, we must construct the monocular vision-based filter. The main difference between the stereo vision-based filter and the monocular vision-based filter is the size of the measurement vector $\mathbf{z}$. As shown in formula (14), for the monocular vision-based filter $\mathbf{z}$ is:

$$\mathbf{z} = \begin{bmatrix} u_{LW} & \vdots & u_{RT} \\ v_{LW} & \vdots & v_{RT} \\ 1 & \vdots & 1 \end{bmatrix}_{3 \times 5} \quad (22)$$

which is half of the $\mathbf{z}$ of the stereo vision-based filter.

IV. EXPERIMENTS

To validate the proposed algorithm and generate training dataset, we build a parallel system as shown in Fig. 3. This system is comprised of the guidance system and the fixed-wing UAV. The guidance system includes two 2-freedom Pan-Tilt-Units (PTUs), two cameras mounted on the PTUs, and a laptop with i9-9900k (CPU) and NVIDIA GTX2080 (GPU)). In the outdoor experiment, the PTUs are placed on both sides of the runway with a 10.77 m baseline. The PTUs attitude measurement resolution reaches 0.00625 degrees and its highest rotary speed is 50 degree/s. The camera DFK 23G445 (Germany) turns along with the PTU for extending the field-of-view and generates 640 X 480 pixel video with 60 frames per second. The fixed-wing UAV Pioneer has a wingspan of 2.3 m and a total mass of 14 kg with petrolic propulsion. The simulation follows the same configuration as the outdoor environment, including the UAV model and the guidance system. Furthermore, the real UAV autopilot PX4 has been also introduced to establish a hardware-in-loop simulation system for a more realistic flight.

Fig. 3. The parallel system containing the outdoor and Gazebo-based simulated environments. It is composed of the guidance system and fixed-wing UAV, and the outdoor and simulated environments have the same configuration.

Online experiments are conducted both in simulated and outdoor environments to validate the performance. A complete landing is composed of sloping, flaring, and taxiing, and there are different requirements of pose estimation accuracy in

different phases. Therefore, the performances of pose estimation are analyzed for all three phases. The dataset which is used for offline training is generated from 3 simulated and 2 outdoor landings. For autonomous ROI extraction, YOLO-v4 [30] is used to detect the UAV and generate ROIs. Experimental results show that YOLO-v4 obtains accurate UAV ROI detection in our landing scenes.

A. Anchor Detection

To validate the anchor detection accuracy through the training data augmentation, three different training datasets are used to train the networks. These datasets include the real dataset (RD), the simulated dataset (SD), and the mixed dataset (MD) combining the RD and SD. As shown in TABLE I, the conventional network (only F part shown in Fig. 2) and the proposed Block-CNN are both trained using the above three datasets and 6 networks with different parameters are obtained accordingly. Their anchor detection accuracy is then tested in the outdoor data. The anchor detection error e is defined as:

$$e = \frac{\|(u, v) - (\tilde{u}, \tilde{v})\|}{w} \quad (23)$$

where (u, v) and $(\tilde{u}, \tilde{v})$ are the detection and ground truth of the anchor locations in the image frames, respectively. As shown in Fig. 4, e indicates the pixel distance between the detection and ground truth of the anchor, and w is the width of the ROI. For each anchor, a detection with $e > 5\%$ is regarded as a failure. For a complete test, assuming that the number of the total anchors is N and the failure number is n , the failure rate f is:

$$f = \frac{n}{N} \quad (24)$$

TABLE I
THE FAILURE RATES DEFINED AS FORMULA (24)

Test data	Anchor	Networks					
		Conventional network			Block-CNN		
		SD	RD	MD	SD	RD	MD
outdoor	LW (%)	6.59	7.41		2.48	1.93	
	LT (%)	6.58	6.07		2.20	1.97	
	FT (%)	5.20	5.20		2.20	1.48	
	RT (%)	9.07	6.86		4.41	2.11	
	RW (%)	7.97	7.69		3.31	3.86	
	Average	7.08	6.65		2.92	2.27	

TABLE I presents the detection failure rates of the five UAV anchors. Compared with the networks trained by the only real dataset, the networks trained by the mixed dataset performs detection accuracy improvement both for the conventional network (average failure rate from 7.08% to 6.65%) and the Block-CNN (average failure rate from 2.92% to 2.27%). Also, under the premise of using the same training dataset, the Block-CNNs show failure rate reductions of 58.8% (RD) and 65.9% (MD) respectively, compared with the conventional networks, which are remarkable accuracy improvement.

Fig. 4 also displays several detection samples in different landing phases using network F . According to the images captured in the simulated and outdoor environments, the UAV shows the more distinguishable feature in the simulated cases. This indicates that the dataset acquired from the simulation is more efficient than that of the outdoor environment. This is because the experimental conditions in the simulated

environment are more controllable. Therefore, generating simulated data not only expands the dataset but also improves the dataset quality, hence contributes to the improvement of the anchor detection accuracy.

In summary, the experimental results suggest that the training dataset augmentation contributes to the anchor detection improvement. Furthermore, compared with the conventional network, the proposed Block-CNN results in significant improvement in the accuracy of the anchor detection in the UAV landing scenes.

Fig. 4. Detection samples in different landing phases using network F . Once the pixel distance between the detection and ground truth of the anchor exceeds $5\% \cdot w$ (ROI width), this detection is regarded as a failure.

B. Pose Estimation

1) Simulations

Here, we evaluate the pose estimation performance of the proposed algorithm and make comparisons with the

conventional PNP method in the simulated environment. Three methods are tested on the simulated guidance system including monocular PNP (MP), monocular EKF (ME), and stereo EKF (SE) proposed in this paper. Since MP and ME only need monocular vision, we consider the average results of the two cameras as their final results.

In Fig. 5, we present the three simulated landing scenes (S_1 , S_2 , S_3) that are designed to validate the performance of the proposed algorithm. The first landing is completed without wind disturbance. To simulate the crosswind in the outdoor environment, a continuous crosswind is created during the second and third landing. Besides, in the third landing, a going around the process is also simulated which is common during the actual landing. The estimated pose error in sloping, flaring, and taxiing phases is showed as Fig. 5. Since the UAV gradually approaches the guidance system, the position estimation error of the ME and MP is gradually reduced from sloping to taxiing phase. In other words, the ME and MP are sensitive to the distance between the UAV and the camera. On the contrary, the distance almost does not affect the positioning error of the proposed SE. Remarkable positioning error of the MP exceeding 20 m (S_2 at the X- and Y-axis) which is likely to result in the deviation from the runway. In the flaring period, the estimation of the height above the ground (Z-axis) is also important. Except for the SE, other methods have significant errors along the Z-axis.

The attitude estimation error shows greater volatility than that of the positioning error. The pitch estimation error reaches 5° in the sloping phase of S_2 . It probably causes the UAV to descend too fast to successfully land. A high initial yaw estimation error is also seen in S_2 . However, the errors of SE

Fig. 5. Three landing trajectories in the simulated environment and the pose estimation error for the three simulated landing.

and ME are gradually converged before entering the flaring period, whereas the error of MP deems volatile.

Fig. 6 displays the RMSE for three landing simulations. For the conventional method MP, the positioning RMSE reaches 20.10 m at the X-axis and 1.40 m at the Z-axis, shows an unacceptable level of accuracy for the landing process. In addition, the yaw RMSE of about 6 degrees in the simulations S_2 and S_3 , cannot support a successful landing. By comparison, ME achieves accuracy improvement for both the position and attitude estimation. Furthermore, significantly remarkable pose RMSE reduction is achieved by the SE. The RMSE at the X- and Y-axes did not exceed 1.0 m, which ensures that the UAV is completely within the runway range. More importantly, the RMSE at the Z-axis remained about 0.05 m. It also laid a key foundation for a successful landing.

Fig. 6. The RMSE of pose estimation for the three simulated experiments.

2) Outdoor Evaluation

For autoland, the estimation accuracy at the Z-axis is the most important in terms of position estimation, especially in the flaring period. Once the UAV is taxied on the runway which is usually detected by the landing detector, the UAV controller would not consider the location at Z-axis. In our outdoor experiments, the runway width is about 10 m. This means that the maximum error at the X-axis in the flaring and taxiing phases should not exceed 5 m.

Two outdoor experiments have been conducted for performance evaluation. In the outdoor experiments, we compare and discuss the results of left PTU-based PNP (LP), right PTU-based PNP (RP), and SE. Since the computer

platform on which the algorithms run is the same as the one used in simulation, the real-time capability in simulation (30 fps) is also valid for the outdoor evaluation. Fig. 7 illustrates the position and attitude estimation results of one of the two outdoor experiments. For a successful landing, smooth temporal positioning is a prerequisite. According to the position estimation results, the error curves of LP and RP show more fluctuations compared with the proposed method SE. For instance, the maximum errors of LP and RP even reach 279 m at the Y-axis and 19 m at the X-axis. This can easily deviate the UAV from the 10 m width runway. On the contrary, the maximum error of SE at the X-axis does not exceed 5 m, which is necessary to ensure that the aircraft is always within the range of the runway in flaring and taxiing phases. For the algorithms LP and RP, there is a higher error at the Y-axis than that of the other axes. This is because the Y-axis has a high degree of coincidence with the direction of the cameras' optical axis. This makes the positioning at the Y-axis more sensitive to measurement errors. The details have been discussed in our previous work [17]. For the attitude estimation, the results of the three methods show comparable temporal fluctuation. The images and detected anchors from two cameras are also shown at the point A, B, and C in Fig. 7. The red and yellow anchors are ground truth and detection results, respectively. It is seen that the anchor RW is out-of-the-field-view in the left camera at point C. This causes a significant Z-axis positioning error.

TABLE IV

THE RMSE IN OUTDOOR PERFORMANCE EVALUATIONS

RMSE	Solution	Sloping		Flaring		Taxiing	
		Exp 1	Exp 2	Exp 1	Exp 2	Exp 1	Exp 2
X axis (m)	SE	3.2	2.8	0.3	0.5	1.9	1.9
	PNP	5.8	3.1	2.6	6.4	2.1	2.8
Y axis (m)	SE	11.9	10.8	3.7	4.9	17.8	12.4
	PNP	54.8	39.0	48.3	37.5	41.5	49.8
Z axis (m)	SE	5.4	2.9	0.4	0.3	7.5	5.9
	PNP	7.6	2.8	1.1	2.1	8.3	9.8
Roll (°)	SE	21.0	19.3	0.7	1.5	0.8	1.5
	PNP	22.5	25.1	2.9	2.4	3.5	2.9
Pitch (°)	SE	19.5	22.8	0.5	0.8	1.1	0.9
	PNP	22.2	21.5	0.7	2.8	1.3	2.0
Yaw (°)	SE	11.5	10.3	1.8	2.2	2.2	1.4
	PNP	7.4	8.4	4.1	4.5	2.5	3.5

TABLE IV shows the RMSE of the two outdoor experiments in the sloping, flaring, and taxiing, respectively. The RMSE in

Fig. 7. The estimation results of one of the outdoor experiments by LP, RP and SE. Estimated trajectories in sloping, flaring, and taxiing are depicted. Besides, several exemplary images captured from cameras during autoland and the anchors detection results are also shown.

the sloping phase is higher than that of in another two landing phases. This is because the UAV is the farthest away from the cameras in the sloping phase, and our previous work [17] has demonstrated that for a given detection error, the further the UAV from the cameras, the greater the location error. Besides, the roll RMSEs of SE are 11.5° and 10.3° , which are even higher than that of the PNP-based solution (7.4° and 8.4°). The reason is that the initial UAV pose of the proposed method SE is roughly estimated, and it needs to take several steps to converge. The RMSEs of the height estimation (at the Z-axis) in the flaring period by the PNP-based solution reach 1.1 m and 1.2 m, thus may lead to failed landing. In contrast, the SE results do not exceed 0.4 m, thus satisfies safe landing requirement.

In summary, the outdoor experimental results show that compared with the conventional PNP-based solution, the proposed method achieves high accuracy improvement of position and attitude estimation. The outdoor experiments also confirm the feasibility of the proposed solution SE to provide the real-time poses of the UAV during autonomous landing.

V. CONCLUSIONS

We presented a deep learning-enabled UAV pose estimation algorithm for autonomous guidance based on the ground stereo vision system. A pipeline of CNN-based anchors detection and anchors-driven pose estimation is constructed. For anchors detection, a Block-CNN learning-type detection algorithm was proposed and developed. Using the blocking mechanism, the proposed algorithm provides higher accuracy and robustness for anchors detection. Considering the high outdoor experimental cost, we built a parallel system including the simulated and outdoor environments with the same configuration to expand the network training dataset through simulated experiments. The pose estimation was realized by an EKF estimator using the detected anchors' locations. The simulation and experimental results showed significant accuracy improvement by the proposed pose estimation method compared with the conventional algorithm. Furthermore, the experiments also demonstrated that the proposed algorithm satisfies the landing navigation requirements in terms of accuracy, real-time capability, and robustness. A likely next step is to consider special scenarios such as different weather conditions and complicated backgrounds.

REFERENCES

- [1] B. Xiao and S. Yin, "A New Disturbance Attenuation Control Scheme for Quadrotor Unmanned Aerial Vehicles," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2922-2932, 2017.
- [2] J. Henawy, Z. Li, W. Yau and G. Seet, "Accurate IMU Factor Using Switched Linear Systems for VIO," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7199-7208, 2021.
- [3] B. Herisse, T. Hamel, R. Mahony and F. Rusotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *IEEE Trans. Robotics*, vol. 28, no. 1, pp. 77-89, 2012.
- [4] C. Zhang, L. Chen and S. Yuan, "ST-VIO: Visual-Inertial Odometry Combined With Image Segmentation and Tracking," *IEEE Trans. Instrumentation and Measurement*, vol. 69, no. 10, pp. 8562-8570, 2020.
- [5] C. Zhang, L. Chen and S. Yuan, "ST-VIO: Visual-Inertial Odometry Combined With Image Segmentation and Tracking," *IEEE Trans. Instrumentation and Measurement*, vol. 69, no. 10, pp. 8562-8570, 2020.
- [6] J. Garforth and B. Webb, "Visual Appearance Analysis of Forest Scenes for Monocular SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1794-1800.
- [7] D. Wager, B. Remes and G. Croon, "Optical-Flow Based Self-Supervised Learning of Obstacle Appearance Applied to MAV Landing," *Robotics and Autonomous Systems*, vol. 100, pp. 78-94, 2018.
- [8] L. Wang, X. Bai and P. Doody, "Quadrotor Autonomous Approaching and Landing on a Vessel Deck," *Journal of Intelligent & Robotic Systems*, vol. 92, pp. 125-143, 2018.
- [9] Y. Qi, J. Jiang, J. Wu, J. Wang, C. Wang and J. Shan, "Autonomous Landing Solution of Low-Cost Quadrotor on a Moving Platform," *Robotics and Autonomous Systems*, vol. 119, pp. 64-76, 2019.
- [10] M. Lungu, "Auto-Landing of Fixed Wing Unmanned Aerial Vehicles Using the Backstepping Control," *ISA Trans.*, vol. 95, pp. 194-210, 2019.
- [11] K. Abu-Jbara, G. Sundaramorthi and C. Claudel, "Fusing Vision and Inertial Sensors for Robust Runway Detection and Tracking," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 9, pp. 1929-1946, 2018.
- [12] Y. Meng, W. Wang, H. Han and M. Zhang, "A Vision/Radar/INS Integrated Guidance Method for Shipboard Landing," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8803-8810, 2019.
- [13] Y. Gui, "Research on Key Techniques of Airborne Vision-Based Navigation for Autonomous Landing of A UAV on A Ship Deck," Ph. D. dissertation, Aeronautical and Astronautical Science and Technology Graduate School of National University of Defense Technology, Changsha, China, 2013.
- [14] N. P. Santos, V. Lobo and A. Bernardino, "AUTOLAND Project: Fixed-Wing UAV Landing on a Fast Patrol Boat Using Computer Vision," *OCEANS 2019 MTS/IEEE SEATTLE*, 2019, pp. 1-5.
- [15] Z. Ma, T. Hu and L. Shen, "Stereo Vision Guiding for the Autonomous Landing of Fixed-Wing UAVs: A Saliency-Inspired Approach," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 2, pp. 43-55, 2017.
- [16] M. Li and T. Hu, "Deep learning Enabled Localization for UAV Autoland," *Chin. J. Aeronaut.*, vol. 34, no. 5, pp. 585-600, 2021.
- [17] D. Tang, T. Hu, L. Shen, D. Zhang, W. Kong and K. H. Low, "Ground Stereo Vision-Based Navigation for Autonomous Take-Off and Landing of UAVs: A Chan-Vese Model Approach," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 2, pp. 67-80, 2016.
- [18] W. Kong, D. Zhou, Y. Zhang, D. Zhang, X. Wang, B. Zhao, *et al.*, "A Ground-Based Optical System for Autonomous Landing of a Fixed Wing UAV," *IROS*, 2014, pp. 4797-4804.
- [19] J. Yan, T. Du, X. Liu, B. Niu and L. Guo, "Method and Implementation of a Bioinspired Polarization-Based Attitude and Heading Reference System by Integration of Polarization Compass and Inertial Sensors," *IEEE Trans. Ind. Electron.*, vol. 67, no. 11, pp. 9802-9812, 2020.
- [20] K. Li, L. Chang and Y. Chen, "Common Frame Based Unscented Quaternion Estimator for Inertial-Integrated Navigation," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2413-2423, 2018.
- [21] M. Mammarella, G. Campa, M. Napolitano, M. Fravolini, Y. Gu and M. Perhinschi, "Machine Vision/GPS Integration Using EKF for the UAV Aerial Refueling Problem," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 38, no. 6, pp. 791-801, 2008.
- [22] L. Diogo, P. David and T. Hedi, "Multi-Task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2752-2764, 2021.
- [23] R. Valle, J. Buenaposada and L. Baumela, "Multi-Task Head Pose Estimation in-the-Wild," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2874-2881, 2021.
- [24] W. Lin, A. Anwar, Z. Li, M. Tong, J. Qiu and H. Gao, "Recognition and Pose Estimation of Auto Parts for an Autonomous Spray Painting Robot," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1709-1719, 2019.
- [25] F. Liu, Z. Wei and G. Zhang, "An Off-Board Vision System for Relative Attitude Measurement of Aircraft," *IEEE Trans. Ind. Electron.*, 2021.
- [26] H. Wu and J. Zhou, "Privacy Leakage of SIFT Features via Deep Generative Model Based Image Reconstruction," *IEEE Trans. Information Forensics and Security*, vol. 16, pp. 2973-2985, 2021.
- [27] C. Campos, R. Elvira, J. Rodriguez, J. Montiel and J. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimodal SLAM," *IEEE Trans. Robotics*, pp. 1-17, 2021.
- [28] L. Nanni, S. Ghidoni and S. Brahmam, "Handcrafted vs. Non-Handcrafted Features for Computer Vision Classification," *Pattern Recognition*, vol. 71, pp. 158-172, 2017.
- [29] D. Tang, Q. Fang, L. Shen and T. Hu, "Onboard Detection-Tracking-Localization," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1555-1565, 2020.
- [30] B. Alexey, W. Chien-Yao and M. Hong-Yuan, "Yolov4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020, arXiv:2004.10934.