A UAV-Based Aircraft Surface Defect Inspection System via External Constraints and Deep Learning

Yuanpeng Liu, Jingxuan Dong, Yida Li, Xiaoxi Gong, and Jun Wang

***Abstract*— In the field of aircraft maintenance, regular inspection of fuselage surface during the aircraft life cycle is a vital task to ensure the aircraft quality and flight safety. Currently, the inspection task is generally carried out manually in an indoor hangar, which is with low efficiency and reliability. In this article, a novel system based on the unmanned aerial vehicle (UAV) is presented to achieve automated aircraft surface inspection efficiently. The hardware is established with a lightweight and low-cost flight platform, on which a sensor containing an inertial measurement unit (IMU) and a camera is equipped for UAV localization. A high-resolution camera is equipped to collect images of fuselage for defect detection. Our inspection framework is mainly composed of two modules: the UAV localization module and the defect detection module. The localization module is designed to estimate the relative pose between the UAV and the aircraft, providing the foundation for image positioning on the aircraft surface. The existing visual–inertial odometry (VIO) approach is adopted to implement the pose estimation. To reduce the large drifts caused by the VIO approach, a novel method is proposed to deploy precalibrated ArUco markers around the aircraft, which serve as external constraints for the VIO objective to realize joint optimization of the camera pose. In addition, an adaptive weighting method is proposed, which takes into consideration the recognition effect of markers to balance the external constraints. The defect detection module aims to detect defects on the fuselage surface from images captured by the high-resolution camera, which is implemented based on deep learning. To address the issue of detection on a few training samples, the transfer learning strategy is exploited to first pretrain the model on a public defect dataset and then fine-tune it on our collected aircraft defect dataset. After detecting the defects, the defective region is reflected on the fuselage surface through the UAV pose on the corresponding frame provided by the localization module, realizing the accurate defect localization. Experiments on both the simulation environment and real data demonstrate the superiority of our proposed external localization module and the effectiveness of the crack detection module.**

***Index Terms*— Aircraft surface inspection, defect detection, pose estimation, unmanned aerial vehicles (UAVs), visual–inertial odometry (VIO).**

I. Introduction

AIRCRAFT surface defect detection is one of the most important tasks in the field of aircraft maintenance. The quality of the fuselage surface is a vital factor for the military aircraft stealth performance [1]. Furthermore, the appearance of defects, such as cracks and scratches, will cause severe safety hazards to civil aviation aircraft. Currently, fuselage cracks are detected and measured manually [2], [3]. Due to the large size of the aircraft, manual vision inspection is considerably time-consuming. In addition, such an approach would have a high incorrect detection rate, which is significantly unreliable. In recent years, robot-based inspection techniques are proposed for aircraft surface inspection, which makes great progress on automatic defect detection [4], [5]. However, since the robot has limited reachable space, the surface inspection of large-scale aircraft involves complicated planning of robot locations, which requires a lot of preparation time and cost.

In order to solve the above issues and improve the effectiveness of maintenance operations, an unmanned aerial vehicle (UAV)-based aircraft surface defect inspection system is proposed in this work. A high-resolution camera is equipped to collect images of the aircraft surface, which are then processed for defect detection. Compared with robot-based inspection systems, UAVs are easier to be manipulated and are more flexible to capture the aircraft surface. Another advantage of the UAV-based system is that the inspection can achieve a relatively large coverage on the aircraft surface since it is not limited by the restriction of navigation planning on the ground, such as the robotic inspection. Based on the system, this article addresses two tasks during the inspection. First, as in [6], the real-time pose of the UAV with respect to the aircraft is estimated, which outputs the real trajectory of the UAV. Second, defects are detected from the images collected by the high-resolution camera. The postures of the UAV that has captured the defect images are then obtained from the first task, with which the defective region can be reflected on the fuselage surface, realizing defect localization. A recapitulative illustration of our inspection system is shown in Fig. 1. Our system is a noncontact approach, which achieves crack detection and location with high accuracy and efficiency while causing no damage to the aircraft surface.

In our task, the accuracy of the UAV's pose relative to the aircraft is critical for the localization of the defect on the fuselage. When working outdoors, the RTK and GPS [7], [8] equipped on the UAV can achieve localization with high accuracy. However, in real applications, most of the maintenance works are implemented in an indoor hangar. Since indoor buildings block the transmission of GPS signals, the method of using RTK to achieve localization will fail. Other works have been carried out to use a laser radar installed on a UAV to achieve indoor localization. However, the laser radar is expensive, and due to its heavy weight, high stability of

Manuscript received 18 March 2022; revised 6 June 2022; accepted 28 July 2022. Date of publication 16 August 2022; date of current version 30 August 2022. This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20190016. The Associate Editor coordinating the review process was Dr. Shovan Barma. *(Yuanpeng Liu and Jingxuan Dong contributed equally to this work.) (Corresponding author: Jun Wang.)*

The authors are with the College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: nuaayp_liu@nuaa.edu.cn; jx.dong@nuaa.edu.cn; nuaa.lyd@nuaa.edu.cn; xiaoxigong.nuaa@gmail.com; wjun@nuaa.edu.cn).

Digital Object Identifier 10.1109/TIM.2022.3198713

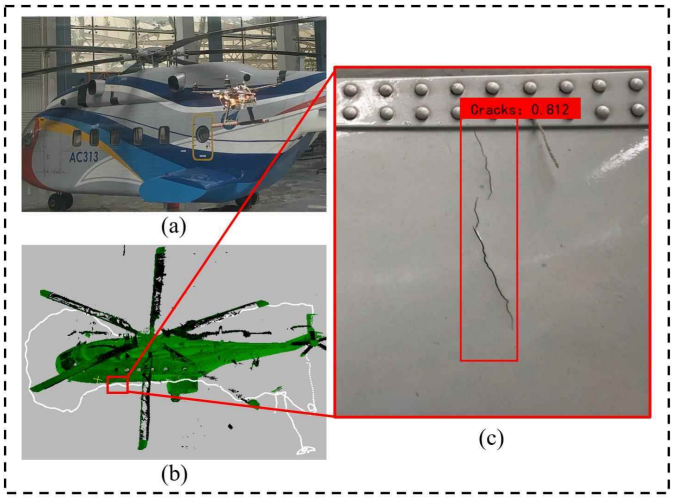


Fig. 1. Illustration of our UAV-based fuselage surface inspection system. The UAV is equipped with the camera flying around the aircraft to capture the image of fuselage surface, so as to implement the crack detection. (a) UAV is inspecting the helicopter. (b) Trajectory of the UAV is inferred by our method, shown in white. (c) Defect detection result. A crack is successfully detected and positioned.

the UAV's flight control system is required when the laser radar is mounted on the UAV. A typical solution is using a monocular camera and a low-cost inertial measurement unit (IMU) to achieve pose estimation of the UAV, which is known as the visual–inertial odometry (VIO) approaches [9], [10], [11], [12]. IMU measurements can not only recover scale information but also conduct auxiliary localization with visual measurement. Nevertheless, estimating the pose of a UAV is exceedingly challenging. The texture of aircraft surface is inadequate, resulting in few feature points in the images, which makes VIO methods fragile. Besides, UAV is required to fly more than one round for elaborative inspection. In the course of a long flight, VIO methods would yield large accumulated drifts. Therefore, solving the drift for camera localization is particularly important. Another challenge is detecting defects on the captured image while acquiring the real posture of the UAV. Motivated by the development of deep learning methods on object detection, a convolutional neural network (CNN) is employed to implement our crack inspection task. However, the lack of sufficient training samples for fuselage defects hinders the accurate detection of defects.

To tackle the above issues, a UAV-based aircraft surface inspection framework is developed, which is composed of a UAV localization module and a defect detection module. Compared with previous inspection systems, the framework proposed in this article can not only achieve real-time UAV localization accurately but also realize fast defect detection and localization on aircraft surface. Our localization module is inspired by the state-of-the-art VIO localization method, i.e., VINS-Mono [13], which can obtain accurate transformation of camera pose between two consecutive frames. However, applying the VINS-Mono method to position the UAV in our task would also face the challenge of large drifts as discussed above. We, therefore, propose to use ArUco markers [14], a robust binary marker in the field of visual navigation, as external constraints to improve the localization accuracy. The markers are deployed in the hanger and precalibrated before the inspection task. Once the monocular localization camera captures the frame containing markers, an external constraint would be added to the objective of the VINS-Mono, forming a least-squares residual formulation to provide more accurate pose estimation results. In addition, to balance all external constraints, an adaptive weighting method is presented, which establishes scoring mechanism for ArUco markers according to the recognition error of markers to improve the exactitude of constraints. For the detection module, the classical YOLOv4 [15] method is exploited, which shows remarkable performance on the real-time object detection. Since the training samples of fuselage surface defects are insufficient, which declines the inference effect of YOLOv4, we refer to the transfer learning strategy to first pretrain the network on public defect datasets and then fine-tune the model on our collected fuselage surface defect dataset. Our system explores the integration and data transmission of multiple instruments, including the UAV, the cameras, and the IMU, and utilizes cross-domain detection methods, significantly enriching the domain of instrumentation and measurement.

In summary, the main contributions of our inspection system are listed as follows.

1) A novel UAV-based fuselage surface inspection system is developed to realize real-time defect detection and localization with high accuracy.
2) A reliable external constraint-based module is proposed for UAV localization, which innovatively constructs external constraints using markers to overcome the problem of accumulated drifts that appeared in classical VIO localization methods, obtaining accurate and robust estimation results of postures of the UAV relative to the aircraft.
3) A novel scoring mechanism is designed for marker quality evaluation according to the recognition effect of markers, which is used for assigning adaptive weights to external constraints.
4) Defects on the fuselage surface with few training samples are effectively detected by the depicted transfer learning strategy, which is rarely used in the field of fuselage inspection and could have a moderate reference for the domain of measurement.

II. Related Work

In this section, the related works are divided into three parts: fuselage inspection systems, UAV localization methods, and defect detection methods.

A. Aircraft Surface Inspection Systems

In the past, aircraft surface inspection is typically conducted relying on periodic human vision, which is obviously inefficient and inaccurate. In order to address these issues, there has been ongoing studies [16], [17], [18] on aircraft inspection using automated robots since the 1990s. In particular, White *et al.* [19] developed a climbing robot for inspection of aerostructures. The robot utilizes vacuum to make it attached to vertical and overhanging surfaces. A traction system is designed for rapid movement of robot on planar and curved surfaces. However, their climbing robots need to contact with

the fuselage surface during the movement, which may cause damage to the delicate surfaces. Madsen *et al.* [20] proposed a robot climbing on nonferrous surfaces for aircraft wings and fuselage inspection. Vacuum suction cups are used for adhesion and two pairs of pneumatic cylinders are adopted for driving. However, the walking speed of the robot is limited by the cylinder stroke, which is of low efficiency for the inspection. Colin and Guibert [21] developed the Air-Cobot project that utilizes a ground robot to realize the inspection. The robot equips a pan-tilt-zoom camera and a lift-able 3-D laser scanner to collect 2-D images and 3-D space information as in [22] and [23]. Nevertheless, the inspection areas are limited by the movement of the robot on the ground, causing the upper surface of the aircraft undetectable.

UAV-based methods have been introduced into the field of aircraft inspection in recent years, due to its convenient operation and flexible movement that is not affected by the environment. Bjerregaard [24] utilized a UAV equipped with cameras to obtain aircraft surface information. Multiple 3-D laser scanners are mounted on the UAV for obstacle avoidance and distance measurement to the aircraft. However, the heavy laser scanners result in a large load, limiting the velocity of the UAV. Miranda *et al*. [6] proposed to use the UAV to acquire precise images coming with useful metadata and a CNN-based method to inspect screws from images. The localization is achieved through the lidar technology. Although relatively accurate localization can be achieved using laser scanners or lidars, their systems are considerably complex and expensive. Contrastively, instead of relying on laser scanners, our system employs a lightweight and low-cost hardware inspired by the success of VIO for localization.

B. UAV-Based Localization Methods

The UAV empirically relies on GPS to achieve localization in the outdoors considering that the signal can be completely covered. RTK can assist GPS in providing higher accurate localization results. Nevertheless, due to the signal attenuation, GPS and RTK will be unavailable indoors, while indoor inspection is common for aircraft maintenance. Consequently, one needs to locate the UAV based on the onboard equipments, instead of relying on external signals. Motivated by 3-D laser scanners for navigation, Grzonka *et al*. [25] used an open-hardware quadrotor platform equipped with a Hokuyo laser range finder, an XSens IMU, a Gumstix computer, and a laser mirror to achieve autonomous flight and obstacle avoidance in an indoor environment. They cleverly use a laser mirror to reflect the laser beam to the ground to achieve height measurement. However, the localization is implemented on a 2-D map based on the 2-D navigation algorithm, which is limited. Özaslan *et al*. [26], [27] employed two orthogonally placed 2-D laser scanners and an IMU to inspect objects with few geometry cue and poor illumination, such as tunnels. Compared with 3-D laser scanner, 2-D laser scanners are lighter for installation on a UAV. The cost is lower. Nevertheless, using multiple 2-D laser scanners to achieve scene perception requires accurate external parameter calibration.

In recent years, visual sensors with their inherent advantages in price and weight have been gradually applied in the field of UAV localization and navigation [28], [29]. Mur-Artal *et al*. [30] and Engle *et al*. [31] came up with a monocular to achieve camera localization through the use of feature tracking between adjacent frames. However, a monocular-based vision-only system is limited in scale recovering, making it incapable of being implemented in real world. By contrast, a low-cost IMU is gradually applied for scale observation, which can assist the visual measurements. As a representative work, Campos *et al*. [32] integrated monocular, stereo, and RGB-D cameras and an IMU to perform visual, visual–inertial, and multimap simultaneous localization and mapping (SLAM). Although they achieved a remarkable performance, the proposed system is extremely complex. Zuo *et al*. [33] utilized a prior LiDAR map to achieve the localization based on a camera. However, the method has numerous requirements for the invariance of the scene, and different planes parked in the hanger will affect the results. Qin *et al*. [13] and its extending work [34] presented a tightly coupled method termed VINS-Mono, which optimizes camera poses from both IMU measurements and visual measurements to improve the accuracy of localization results. Nevertheless, in our application, the fuselage surface is texture-less, causing the failure of VINS-Mono in the aspect of visual tracking. Moreover, the long flight distance of UAV during the inspection will yield drifts due to the IMU integration errors. Instead, we propose an external constraint-based localization module (ECLM) to eliminate the drifts for aircraft inspection.

C. Fuselage Detection

Due to the high efficiency and accuracy, 2-D image-based fuselage defect detection has attracted extensive researcher attention [35], [36], [37], [38]. Jovančević *et al.* [39] implemented the fuselage detection task based on shape edges. Hough transformation is applied for closed detection of oxygen bay handle, while the EDCircles method is used for air-inlet vent inspection. Their method is less robust to noise and occlusion. Rice *et al.* [40] detected the missing screws and fasteners on the fuselage by matching the predefined patterns. The effect of feature extraction has a great impact on the recognition accuracy.

Considering the huge success of deep learning algorithms [41] on object detection, learning-based methods achieve more accurate detection results compared to feature-based methods. Miranda *et al.* [6], [42] showed the capability of screw detection by using deep neural networks. The basic idea of the above two works is to find a cluster of interested zones and employ a deep CNN for precise location of the defects. However, their networks suffered low detection efficiency due to the complicated designs. Moreover, CNNs are limited by the insufficient training data, which often occurs in the aviation industry defects detection scenes. Inspired by the above works, we employ an object detection model, Yolov4 [15], for real-time fuselage defect detection. A data augmentation method is utilized to tackle with problem of insufficient training data.

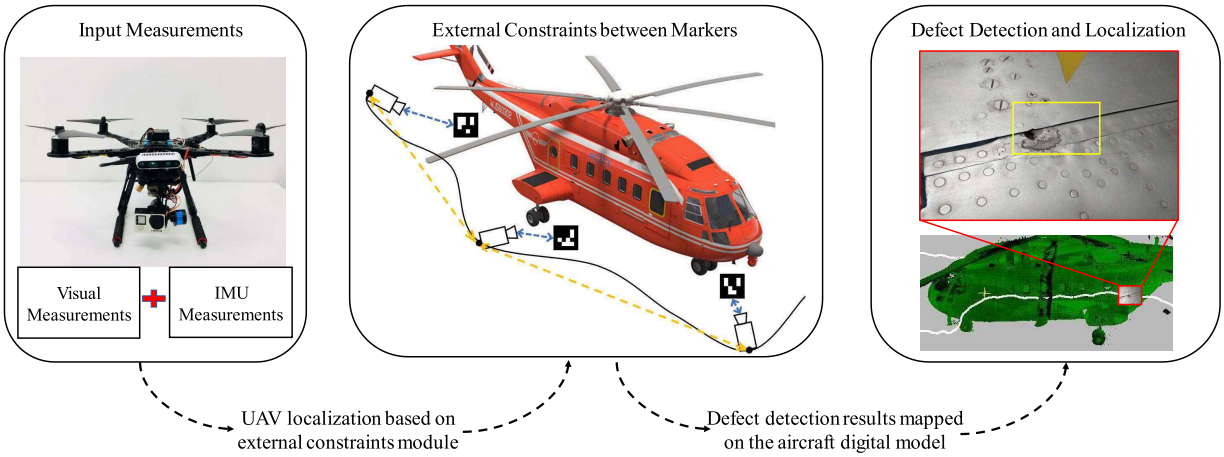


Fig. 2. Overview of the proposed UAV-based aircraft crack inspection system. Our system consists of two main stages: camera localization and crack detection. The camera localization is accomplished by the proposed adaptive external constraints module, which is based on the VIO. The defect detection part contains a data augmentation step and transfer learning strategy based on an object detection model to recognize defects. Finally, the detected defects are mapped on the fuselage surface, according to the accurate UAV pose with respect to the aircraft.

III. Overview

In this section, a brief overview of the proposed aircraft surface inspection system is illustrated. Our system employs a UAV equipped with cameras and IMU to achieve defect detection and localization on the fuselage. Fig. 2 shows the pipeline of our proposed system, which essentially includes two modules. In order to localize the defects, the UAV needs to be first localized during the inspection. Hence, an ECLM is presented to provide the high-accuracy localization of the UAV. After that, a defect detection module based on one of the state-of-the-art object detection models is presented to detect defects with the input of the images of aircraft surface.

A. External Constraint-Based Localization Module

Taking as input the measured values of a localization camera and an IMU, specifically the RGB image streams from the camera and the 3-D acceleration and 3-D angular velocity streams from the IMU, our method estimates camera poses through a VIO approach. To deal with the problem of large drifts occurred in most VIO approaches, external constraints are innovatively introduced into the existing VIO optimization objective. More specifically, several ArUco markers are deployed in the hanger in advance, and their relative poses are then calibrated. During the inspection, the camera continuously captures the precalibrated markers as the UAV moves. Through detecting the marker, the camera pose relative to the detected marker can be inferred. Therefore, the constraints between the markers can be transferred to the constraints between the camera frames capturing the markers through the perspective-n-point (PnP) algorithm [43]. According to the cases that the external constraints are established on different markers and the same marker, two kinds of constraint models are therefore proposed. Furthermore, the marker quality in the captured image could have a significant impact on the reliability of the constraint. The continuous capture of markers also results in enormous constraints. In order to balance them, an adaptive weighting method is devised, which assigns a weight to each constraint based on the score of the marker in the image. The score function is constructed based on the distance and the deflection of the marker relative to the camera.

B. Defect Detection Module

The camera used for defect detection would collect the images of fuselage surface during the inspection. A deep learning method is then employed to detect defects from these images. For the network, a classical and lightwise object detection architecture, Yolov4, is adopted. Taking into consideration the insufficient training data for defects on aircraft surfaces, we turn to exploit the transfer learning strategy for our task. In particular, the Yolov4 network is first pretrained on a publicly available defect dataset. After that, the network head is frozen and the classification model is fine-tuned using our collected dataset of fuselage surface.

IV. Effective Camera Localization

In this section, the hardware of our UAV system is first introduced for aircraft surface inspection. A classical VIO method, i.e., VINS-Mono [13], is then reviewed, which achieves the state-of-the-art performance on estimating postures of a UAV in real time. The shortcomings of the VINS-Mono are also illustrated when applying the method in our scenario. After that, our localization module is illustrated in detail, which utilizes adaptive external constraints to accomplish pose estimation with higher accuracy and robustness.

A. UAV Platform

The structure of our UAV platform is shown in Fig. 3. The platform is based on an S500 quadrotor platform equipped with a Pixhawk flight controller. A Nvidia Jetson TX2 microcomputer is served as the onboard computation device. The UAV platform includes two cameras: the detection camera and the localization camera. Fig. 4 visualizes the data communication among the sensors. The detection camera is employed to collect fuselage surface data for defect detection, which is

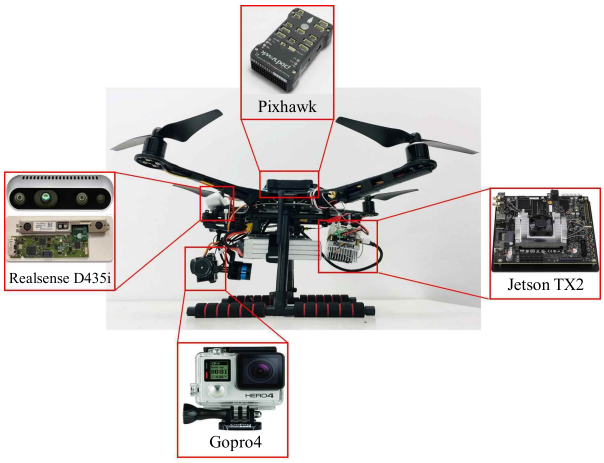


Fig. 3. Four-rotor flying platform equipped with a Realsense D435i sensor, a Gopro4 high-resolution camera, a Nvidia Jetson TX2 microcomputer, and a PX4 flight control. The Realsense D435i sensor contains an IMU and a monocular camera to carry out the pose estimation of the UAV. The Gopro4 camera is utilized for defeat detection. This platform weights 1.7 kg and can fly for approximately 18 min.

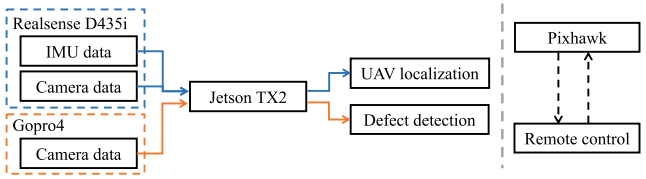


Fig. 4. Visualization of the data communication among the sensors. The blue arrows show the data processing of the localization module, while the red arrows represent the detection module.

demanded for high resolution. To this end, the Gopro Hero4 black sport camera is selected due to its remarkable antishake performance and high resolution of images. The resolution of the detection camera is 2592 × 1944. In order to obtain better quality images of the fuselage surface, a Tarot 2-Axis brushless gimbal is deployed to stabilize and control the Gopro camera precisely. The localization camera is used for UAV localization. An Intel RealSense D435i camera is adopted, which has a built-in IMU module and a monocular camera for both the IMU measurements and visual measurements. The resolution of the camera is 1280 × 720, which is capable of providing image data at a rate of 30 frames/s for visual measurements. The IMU equipped on the D435i camera provides the 3-D acceleration and the 3-D angular velocity data at a sampling frequency of 200 Hz. D435i is installed on the top of Gopro 4 to make sure that the D435i can capture more features of the fuselage to achieve localization. After Gopro4 and D435i are installed on the UAV, their extrinsic parameters are calibrated. Once the pose of the D435i camera is obtained through the VIO method, the pose of the Gopro4 camera can be inferred through the extrinsic parameters, which is utilized for defect localization on the fuselage surface.

B. Notations

Before illustrating the proposed method, the following notations used in this work are defined. More specifically, we denote $(\cdot)^v$ as the camera trajectory coordinate frame.

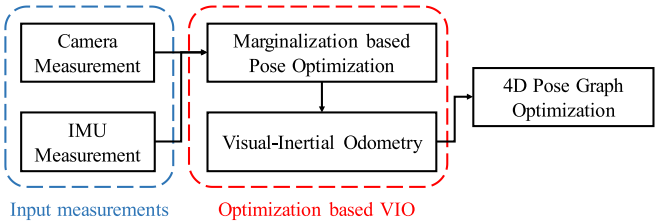


Fig. 5. Overview of the VINS-Mono [13].

$(\cdot)^a$ is the aircraft 3-D digital model coordinate frame. $(\cdot)^c$ represents the camera coordinate frame. $(\cdot)^m$ is the marker coordinate frame. $\hat{(\cdot)}$ denotes the reference value of a measurand. It can be obtained by the PnP algorithm in the VINS-Mono or inferred from the precalibrated markers in our method. $\mathbf{R}$ and $\mathbf{p}$ are employed as the rotation term and position (translation) term, respectively, where $\mathbf{R}$ is a 3 × 3 matrix and $\mathbf{p}$ is a 1 × 3 vector. For example, $\mathbf{R}_c^m$ and $\mathbf{p}_c^m$ are the rotation and translation terms from the camera frame to the marker frame. In particular, ψ is considered as the yaw angle of Euler angles in our pose optimization system.

C. Review of VINS-Mono

Since our localization module is based on the VINS-Mono, one of the state-of-the-art VIO systems, this section briefly introduces the method. The relied hardware is an IMU and a camera, which is significantly cheap and light. Compared with conventional visual odometry systems, the equipped IMU can effectively contribute to auxiliary localization with visual measurement. An overview of the VINS-Mono algorithm is shown in Fig. 5.

The input data for VINS-Mono is the RGB image stream from the camera and the 3-D acceleration and the 3-D angular velocity streams from the IMU. VINS-Mono adopts a marginalization method, which optimizes inertial and visual measurements in a couple way to calculate camera pose. Specifically, once a new camera frame comes, Harris features [44] are detected for visual tracking. The marginalization method takes the new frame into the sliding window for optimization and removes the old frame that has been optimized. The prior cues yielded by old frames also participate in the optimization process. The optimization of camera pose estimation is formulated as

$$\min_{\mathcal{X}} \{r_{\mathcal{P}} + r_{\mathcal{B}} + r_{\mathcal{C}}\} \tag{1}$$

where $r_{\mathcal{P}}$ denotes the prior information generated by the marginalized old frame. $r_{\mathcal{B}}$ is the residuals of the preintegrated IMU values between two consecutive IMU frames. $r_{\mathcal{C}}$ depicts the reprojection errors produced by tracking the same Harris features observed in two consecutive frames, which is computed through the optical flow algorithm. $\mathcal{X}$ represents the states of the camera, which contains 3-DoF position and 3-DoF orientation. In addition, since the IMU can directly provide observations of the roll and pitch angles of the UAV by the gravity vector, VINS-Mono only performs the estimations of the position and the yaw angle, forming a 4-DoF estimation. Hence, the residual of the sequential frames proposed by

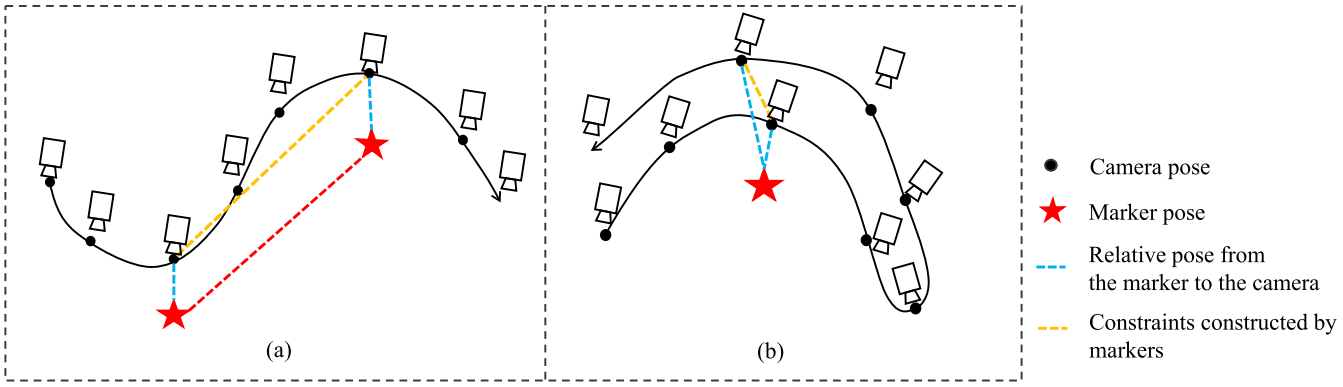


Fig. 6. Illustration of our proposed external constraints module. (a) Constraints constructed by using two adjacent markers. (b) Constraints constructed by using the same marker, which is captured by the camera multiple times. In addition, the marker pose is obtained by using the laser tracker in advance. The relative pose from the marker to the camera can be estimated based on the PnP algorithm.

VINS-Mono is defined as

$$\begin{aligned}\hat{\mathbf{p}}_{c_i c_j}^{c_i} &= \hat{\mathbf{R}}_{c_i}^{v\,-1}\left(\hat{\mathbf{p}}_{c_j}^{v} - \hat{\mathbf{p}}_{c_i}^{v}\right)\\ \hat{\psi}_{c_i c_j} &= \hat{\psi}_{c_j} - \hat{\psi}_{c_i}\\ \gamma_{\text{vins}}(\mathbf{p}_{c_i}, \psi_{c_i}, \mathbf{p}_{c_j}, \psi_{c_j}) &= \begin{bmatrix} \mathbf{R}_{c_i}^{v\,-1}\left(\mathbf{p}_{c_j}^{v} - \mathbf{p}_{c_i}^{v}\right) - \hat{\mathbf{p}}_{c_i c_j}^{c_i} \\ \psi_{c_j} - \psi_{c_i} - \hat{\psi}_{c_i c_j} \end{bmatrix}\end{aligned} \tag{2}$$

where c_i and c_j denote the ith and jth camera frames, respectively, and $\hat{\mathbf{R}}_c^v$ and $\hat{\mathbf{p}}_c^v$ represent the camera pose relative to the camera trajectory coordinate system, which are premeasured values through the PnP algorithm. $\mathbf{R}_c^v$ and $\mathbf{p}_c^v$ are terms that need to be optimized. By minimizing the residuals, the camera poses can be optimized. In practice, long-term preintegration of IMU would inevitably cause accumulated drifts. Therefore, VINS-Mono adopts a global pose graph optimization [45] to reduce the accumulated drifts. When the camera moves back to the starting position, forming a closed-loop constraint, the current camera pose is reoptimized using a PnP-based method. We refer readers to [45] for more details.

D. External Constraint-Based Localization Module

Inspired by the success of camera localization of the VINS-Mono, we seek to integrate the VINS-Mono to the UAV for aircraft defect inspection. A UAV is ubiquitously equipped with an IMU and a camera, which meets the hardware requirements of the VINS-Mono. The input is the RGB image data and IMU data, specifically the 3-D acceleration and the 3-D angular velocity, as in VINS-Mono. However, since the fuselage surface is texture-less, Harris features cannot be effectively detected. Therefore, directly applying VINS-Mono suffers great challenges. Moreover, VINS-Mono only generates graph optimization when the camera returns to the starting position, causing deviations in the previous calculation of the correct trajectory. In our experiments, directly using the VINS-Mono algorithm to calculate camera pose causes a large drift in the produced trajectory, which drives our work.

In our inspection task, the aircraft is commonly parked in the hanger. On this basis, an ECLM is proposed and integrated into the VINS-Mono objective, to address the issues caused by the original VINS-Mono method as discussed above. Specifically, in the preparation phase, markers are deployed surrounding the aircraft in the hanger. The markers can provide external constraints to facilitate camera localization. The ArUco marker is adopted in this work, due to its convenience and robustness for detection. Each ArUco marker has a predefined coordinate frame built by its inner coded matrix. A laser tracker is then used to calibrate the precise relative poses between the planar markers. Once the localization camera captures a marker in the inspection phase, the relative pose between the camera and the marker can be computed. A reference pose is then obtained through the calibrated relative pose between markers, serving as the external constraints to commendably optimize the trajectory of the camera. Taking into consideration the constraints established by the same marker and different markers, two kinds of external constraints are devised to address the situation of whether a marker is captured or not before. Note that since the measured values of the IMU make the pitch and roll angles of the camera always observable, similar to VINS-Mono, our goal focuses on the optimization of the 3-D pose and yaw angle of the current frame relative to the reference frame, i.e., a 4-DOF optimization.

1) Constraints Based on Different Markers: In the inspection phase, the localization camera will capture and record different markers deployed in the scene as the UAV moves. Through detecting markers, each marker is capable of reflecting an estimation of the camera pose. Since the relative poses of markers are precalibrated, the relative poses of the camera can be obtained at the frames capturing the markers. These poses would serve as constraints for the VINS-Mono algorithm. A visualized illustration is shown in Fig. 6(a). More specifically, a reference marker m_i is first set and further captured by a camera keyframe termed the reference frame c_i. The relative pose $\mathbf{P}_{c_i}^{m_i}$ (1×6 vector) from the camera frame to the reference marker frame is then estimated by detecting the marker. Once a new marker m_j is captured by the camera, the relative pose $\mathbf{P}_{c_j}^{m_j}$ between the camera and the new marker captured by the current camera keyframe c_j can be estimated. In our implementation, the reference marker is initially set to the first deployed marker and is then updated as the last record marker. Combining the two relative poses $\mathbf{P}_{c_i}^{m_i}$ and $\mathbf{P}_{c_j}^{m_j}$ with the precalibrated relative pose of the two markers, denoted as

$\mathbf{P}_{m_j}^{m_i}$, the estimation of the relative pose of the camera between the two keyframes c_i and c_j can be inferred and formulated as

$$\hat{\mathbf{P}}_{c_i c_j}^{c_i} = \mathbf{P}_{c_j}^{m_j} \mathbf{P}_{m_j}^{m_i} \left(\mathbf{P}_{c_i}^{m_i}\right)^{-1} \tag{3}$$

where $\hat{\mathbf{P}}_{c_i c_j}^{c_i}$ denotes the estimation. Subsequently, the relative position $\hat{\mathbf{p}}_{c_i c_j}^{c_i}$ and yaw angle $\hat{\psi}_{c_i c_j}^{c_i}$ regarding keyframe c_j and the previous keyframe c_i can be obtained through

$$\begin{aligned} \hat{\mathbf{p}}_{c_i c_j}^{c_i} &= \left(\hat{\mathbf{R}}_{m_i}^{m} \hat{\mathbf{R}}_{c_i}^{m_i}\right)^{-1} \left(\hat{\mathbf{R}}_{c_j}^{m_j} \hat{\mathbf{p}}_{m_j}^{m} + \hat{\mathbf{p}}_{c_j}^{m_j} - \hat{\mathbf{R}}_{c_i}^{m_i} \hat{\mathbf{p}}_{m_i}^{m} - \hat{\mathbf{p}}_{c_i}^{m_i}\right) \\ \hat{\psi}_{c_i c_j} &= \mathrm{Yaw}\left(\hat{\mathbf{R}}_{c_i c_j}^{c_i}\right) \end{aligned} \tag{4}$$

where $\hat{\mathbf{R}}_{c_i c_j}^{c_i}$ is the matrix representing rotation from keyframe c_j to c_i.

$$\hat{\mathbf{R}}_{c_i c_j}^{c_i} = \hat{\mathbf{R}}_{m_i}^{c_i} \hat{\mathbf{R}}_{m_j}^{m_i} \hat{\mathbf{R}}_{c_j}^{m_j}. \tag{5}$$

$\mathrm{Yaw}(\cdot)$ is a function that transfers the rotation matrix to Euler angles and then picks the yaw angle. The reason for picking the yaw angle is that only the yaw angle in Euler angles and the translation of the camera are integrated into pose optimization, as in VINS-Mono. From the perspective of the camera trajectory, the yaw angle and translation of the camera to be estimated can be defined as

$$\begin{aligned} \mathbf{p}_{c_i c_j}^{c_i} &= \left(\mathbf{R}_{c_i}^{v}\right)^{-1} \left(\mathbf{p}_{c_j}^{v} - \mathbf{p}_{c_i}^{v}\right) \\ \psi_{c_i c_j} &= \mathrm{Yaw}\left(\left(\mathbf{R}_{c_i}^{v}\right)^{-1} \mathbf{R}_{c_j}^{v}\right) \end{aligned} \tag{6}$$

where $\mathbf{R}_{c}^{v}$ and $\mathbf{p}_{c}^{v}$ are the camera rotation and position to be optimized, respectively. On this basis, a constraint can be constructed and integrated into the VINS-Mono formulation to optimize the current camera pose, which can be defined as

$$\gamma_{\text{external\_diff}}\left(\mathbf{p}_{c_i}^{v}, \psi_{c_i}, \mathbf{p}_{c_j}^{v}, \psi_{c_j}\right) = \begin{bmatrix} \mathbf{p}_{c_i c_j}^{c_i} - \hat{\mathbf{p}}_{c_i c_j}^{c_i} \\ \psi_{c_i c_j} - \hat{\psi}_{c_i c_j} \end{bmatrix}. \tag{7}$$

2) Constraints Based on the Same Marker: During the normal inspection process with a UAV, one marker could be captured by the camera many times, especially when the UAV needs to repeat multiple cycles of its flight to better cover the surface of the fuselage. Once a recorded marker is repetitively captured by the camera on the UAV, a constraint between the current keyframe and the reference keyframe is then established. Note that the keyframe where the marker is last recorded is chosen as the reference frame in this case. More specifically, we denote m_k as the repetitively detected marker, c_{k_i} as the reference keyframe, and c_{k_j} as the current frame for capturing the marker m_k. The relative poses $\mathbf{P}_{c_{k_i}}^{m_k}$ and $\mathbf{P}_{c_{k_j}}^{m_k}$ from the camera frames c_{k_i} and c_{k_j} to the reference marker coordinate system are then estimated. On this basis, the rotation and translation terms from the current camera keyframe c_{k_j} to the reference keyframe c_{k_i} are computed by

$$\begin{aligned} \hat{\mathbf{p}}_{c_{k_i} c_{k_j}}^{c_{k_i}} &= \left(\hat{\mathbf{R}}_{c_{k_i}}^{m_k}\right)^{-1} \left(\hat{\mathbf{p}}_{c_{k_j}}^{m_k} - \hat{\mathbf{p}}_{c_{k_i}}^{m_k}\right) \\ \hat{\mathbf{R}}_{c_{k_i} c_{k_j}}^{c_{k_i}} &= \hat{\mathbf{R}}_{m_k}^{c_{k_i}} \hat{\mathbf{R}}_{c_{k_j}}^{m_k} \\ \hat{\psi}_{c_{k_i} c_{k_j}} &= \mathrm{Yaw}\left(\hat{\mathbf{R}}_{c_{k_j}}^{c_{k_i}}\right) \end{aligned} \tag{8}$$

where $\hat{\mathbf{R}}_{c_{k_i} c_{k_j}}^{c_{k_i}}$ and $\hat{\mathbf{p}}_{c_{k_i} c_{k_j}}^{c_{k_i}}$ represent the rotation and position terms of transformation from the frame c_{k_j} to c_{k_i}, respectively.

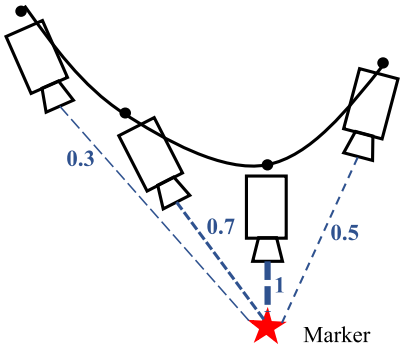


Fig. 7. Illustration of the proposed adaptive weight method. When the camera is far from the marker or the angle between the camera optical axis and the plane normal of the marker is large, the estimated weight is small, considering that the quality of the image involves markers that will affect the accuracy of constructing external constraints.

$\hat{\psi}_{c_{k_i} c_{k_j}}$ is the yaw angle decomposing from the rotation matrix. Similar to the constraints based on different markers as mentioned above, the constraints based on the same marker are defined as

$$\gamma_{\text{external\_same}}\left(\mathbf{p}_{c_{k_i}}^{v}, \psi_{c_{k_i}}, \mathbf{p}_{c_{k_j}}^{v}, \psi_{c_{k_j}}\right) = \begin{bmatrix} \mathbf{p}_{c_{k_i} c_{k_j}}^{c_{k_i}} - \hat{\mathbf{p}}_{c_{k_i} c_{k_j}}^{c_{k_i}} \\ \psi_{c_{k_i} c_{k_j}} - \hat{\psi}_{c_{k_i} c_{k_j}} \end{bmatrix} \tag{9}$$

where $\mathbf{p}_{c_{k_i} c_{k_j}}^{c_{k_i}}$ and $\psi_{c_{k_i} c_{k_j}}^{c_{k_i}}$ are camera position and yaw angle to be optimized, respectively. Their definitions are given in (6).

3) Adaptive Weighting: Considering that the localization camera can continuously capture the marker when the UAV passes through it, multiple external constraints based on the marker will be integrated into the pose optimization. Since the relative pose from the marker frame to the camera frame is obtained through the existing detection algorithm, the marker quality in the captured images has a significant impact on the computation of camera poses. Assigning each established constraint with a fixed weight would obviously amplify recognition errors from low-quality markers and thus cause drift in our localization module. Consequently, an adaptive weighting method is proposed, which adaptively assigns a weight to the captured keyframe according to the quality of the captured marker, thus improving the localization accuracy. Our method is motivated by an observation that the marker detection accuracy typically depends on two factors: 1) the angle between the camera's optical axis and the normal of the marker planner and 2) the distance between the camera and the marker. Hence, a score function is designed based on the angle and distance, which is defined as

$$\omega = \frac{\cos\theta}{1 + e^{k(d - d_0)}} \tag{10}$$

where θ represents the angle between the normal vector of marker plane and the optical axis of the camera. d denotes the distance from the marker center to the camera center. k and d_0 are chosen coefficients, which are positive and determined by the characteristic of a camera. When the angle is close to 0 or the distance between the camera and the marker decreases, the weight is close to 1 such that the external constraints have a greater effect on the optimization of the camera trajectory. A visualized illustration is given in Fig. 7. Through the adaptive weighting method, the external

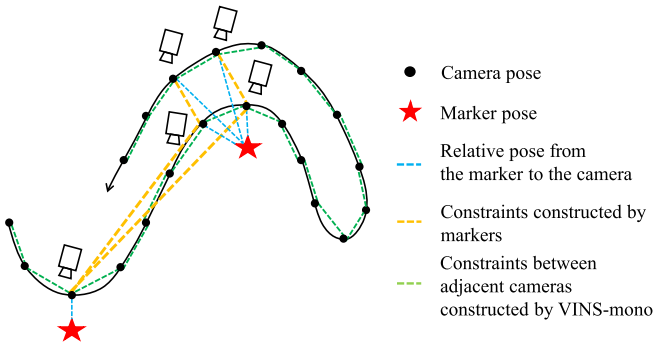


Fig. 8. Illustration of the camera pose optimization formulation, which consists of the external constraints presented by our method and pose residuals of adjacent cameras generated by the VINS-Mono. Based on VIO results, the proposed ECLM can effectively eliminate the drifts during the flight of the UAV.

constraints error caused by the computation of the relative pose between the camera and the marker could be effectively reduced.

4) Optimization Objective: In the final form of the formulation for camera pose optimization, the residuals between the sequential frames output by the VINS-Mono method is retained. The proposed external constraints are additionally introduced to eliminate the drift caused by the VINS-Mono algorithm. A visual illustration is shown in Fig. 8. The overall formulation taking into account of above constraints is defined as the following least-squares problems:

$$\min_{\mathbf{p},\psi} \left\{ \sum_{(i,j)\in\mathcal{S}} \|\gamma_{\text{vins}}\|^2 + \sum_{(i,j)\in\mathcal{L}} \omega(\|\gamma_{\text{external}}\|^2) \right\} \tag{11}$$

where $\mathcal{S}$ represents the sequential frames. Their poses are estimated by VINS-Mono. $\mathcal{L}$ denotes frames that observed markers with high quality during the camera movement. An adaptive weight ω is assigned to the frames that detect markers to balance the established constraints, as introduced in Section I, and therefore to improve the accuracy and the robustness of the optimal camera trajectory. The algorithm of the proposed ECLM is described in Algorithm 1. The optimization method we adopt is the gradient descent algorithm.

Algorithm 1 ECLM

Input: The number of camera frames, N; The camera poses for every frame output by VINS-Mono, $\mathbf{P}_c = \{\mathbf{P}_{c_i}^v\}_{i=1}^N \subset \mathbb{R}^{4\times4}$; The captured image by the localization camera, $F = \{f_i\}_{i=1}^N$; The capturing time stamp for the corresponding image, $T = \{t_i\}_{i=1}^N$; The threshold of marker score for accepting the constraint, η; The threshold for the minimum time interval of two adjacent frames, τ;

Output: The accurate camera poses for each frame, $\hat{\mathbf{P}}_{c_i}^v$;

```
Initialization: marker set ID* ← ∅;
for P^v_{c_i} ∈ P_c do
  build the constraint from VIMS-Mono according to Equation 2;
  if frame f_i contains a marker then
    detect the marker id ID_i;
    compute the score ω_i according to Equation 10;
    if ω_i ≥ η then
      if ID_i ∉ ID* then
        add γ_external_diff according to Equation 7;
        ID* ← ID_i
      else
        get the time stamp t_p of the last frame recording the same marker ID_i;
        if t_i − t_p > τ then
          add γ_external_diff according to Equation 7;
        else
          add γ_external_same according to Equation 9;
        end if
      end if
    end if
  end if
end for
start optimization with the gradient descent method;
return the optimized pose {P̂^v_{c_i}}_{i=1}^N.
```

V. Aircraft Surface Defect Detection

After acquiring images of the aircraft surface, it is a vital task to detect the defects, i.e., the cracks and scratches in our work. Our method is based on deep learning, which has led to great success in the field of object detection. However, cracks and scratches occurred in images are usually small, which hinders the accurate detection. Therefore, Yolov4 [15] architecture is adopted in our method, which has a satisfactory ability to address the challenges. More specifically, Yolov4 utilizes a pyramid structure to extract multiscale features from an input image such that it can detect small defects. Furthermore, Yolov4 integrates a self-antagonism training data enhancement module and an attention module, which is capable of augmenting the dataset and improving the detection accuracy and robustness.

A. Training Data

The training data are captured from eight types of aircraft using the detection camera on the UAV, to alleviate the impacts caused by different surface textures and the types of rivets. To further increase the robustness of the deep learning model, additional images of the aircraft surface are supplied, which are captured with angles and distances that a UAV cannot reach. Due to the high resolution, the images are cropped into small parts with a resolution of 480 × 480 to fit for the network input. We then invite several experts in the aviation field to identify the two defects in the dataset and annotate them manually. Next, a data augmentation method, including image flipping, cropping, and illumination variation, is utilized to expand our dataset, improving the generalization of our network. Details of the training data are given in Table I.

B. Training Details

Due to the insufficient training data, the transfer learning strategy is employed, which utilized a pretraining and fine-tuning scheme to increase the robustness and generalization of the detection model. Since the shallow layer of the

TABLE I
DETAILS OF THE TRAINING DATA IN OUR SYSTEM

Defect species	Number of images
Normal	4600
Cracks	1100
Scratches	950
Total	6650

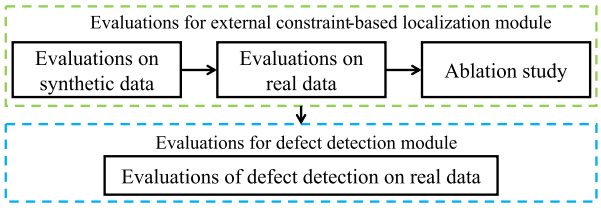


Fig. 9. Evaluation process of our system.

network mainly extracts low-level semantic features of the image, such as edge information, the entire network is first pretrained on a public concrete defect dataset. After that, the network is trained on the collected aircraft surface dataset for the first 120 epochs. The parameters of the shallow layer of the network are then frozen, and the deep classification network is subsequently trained for the rest epochs. Through the training strategy, the model can achieve more accurate defect detection for insufficient training data.

VI. RESULTS

In this section, we primarily evaluate the proposed external constraint-based localization algorithm on the synthetic data and real data. First, qualitative and quantitative results in terms of predicted camera trajectories produced by our algorithm are displayed. Next, comparisons with several state-of-the-art visual–inertial localization algorithms are implemented. An ablation study is then carried out to demonstrate the effectiveness of the proposed adaptive weighting method. Finally, the deep learning-based detection module is evaluated by marking the detected defects on the collected images. The evaluation process is shown in Fig. 9.

A. Experiments on Synthetic Data

1) Experimental Settings: We first evaluate the localization results in a simulated scenario. The scene is built using the AUTODESK 3DS MAX software.[1] A large flight hanger is constructed, in which a real-scale A350 airliner is parked. The airliner is painted with textures similar to those in reality to match the actual situation. Taking into consideration the sizes and wingspan of the airliner, five markers are used to serve as the external constraints. The size of markers is set to 0.5 m $\times$ 0.5 m. The simulated scenario is shown in Fig. 10(a). Note that although more than five markers are deployed in the scene, only five markers are selected to realize our ECLM, as shown in Fig. 10(b). Other markers are utilized to determine the relationship between the markers and the airliner model, providing convenience to align the camera trajectory to the airliner model coordinate frame.

TABLE II
APE AT DIFFERENT DISTANCES IN THE SIMULATED SCENE

Distance (m)	ORB-SLAM3 [32]		VINS-Mono [13]		Ours	
	APE (*trans*: m, *yaw*: °)					
	trans.	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*
150	0.19	**2.81**	**0.18**	2.83	**0.18**	**2.81**
300	0.52	5.96	0.71	6.12	**0.41**	**5.92**
450	0.98	9.51	1.16	9.52	**0.51**	**9.28**

After that, the scenario is exported into a 3-D dynamic simulator, i.e., Gazebo,[2] in which a simulated UAV equipped with an IMU and a camera is built. The parameters of the simulated IMU and camera are set to be consistent with those in the real hardware mentioned in Section IV. Their extrinsic parameters are subsequently calibrated. Since the experiments in this section are designed to evaluate the performance of the proposed localization module, the detection camera for defect detection is not simulated.

2) Implementation Details: The virtual camera is controlled to capture the first marker, which is regarded as the starting location. The inspection is then carried out around the simulated airliner. During the movement, the camera gradually captures the specified markers. Each marker is detected and provides an external constraint according to Algorithm 1.

3) Metrics: Since the ground truth of the camera trajectory can be obtained in the simulation, the metric of absolute pose error (APE)[3] is employed to conduct the quantitative evaluations. More specifically, given a measured pose and the ground truth, APE observes the pose errors between them in two aspects: the rotation and the translation. The rotation error is typically represented as the error of Euler angles, specifically, the yaw angle γ in our case. The translation error represents the Euclidean distance between the predicted position of a frame and the ground truth position of the corresponding frame. The corresponding frames are matched according to the timestamps of all recorded frames.

4) Comparisons: Our approach is compared with two state-of-the-art visual–inertial localization methods, VINS-Mono [13] and ORB-SLAM3 [32], in terms of the APE under different moving distances. The quantitative results are reported in Table II. As can be seen, at the beginning of the traveling (at a moving distance of 150 m), three methods achieve similar localization errors. As the moving distance increases, VINS-Mono and ORB-SLAM3 show relatively large localization errors, while ours is obviously smaller. Specifically, at the moving distance of 300 m, the translation error of our method is 0.11 m smaller than ORB-SLAM3 and 0.3 m smaller than VINS-Mono. The yaw angle error of our method is 0.2° smaller than that of VINS-Mono. When the moving distance reaches 450 m, the translation error of our method is 0.47 m smaller than ORB-SLAM3 and 0.65 m smaller than

[1] https://knowledge.autodesk.com/support/3ds-max

[2] https://gazebosim.org/

[3] https://github.com/MichaelGrupp/evo

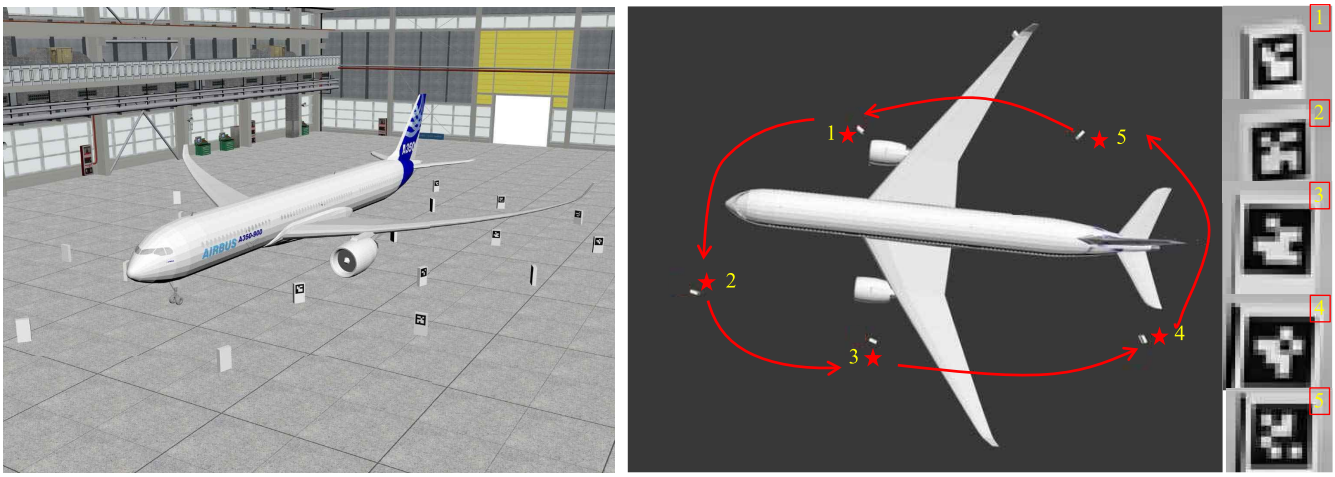


Fig. 10. (a) Simulated hangar scene. (b) Markers are deployed as external constraints in the simulated scene.

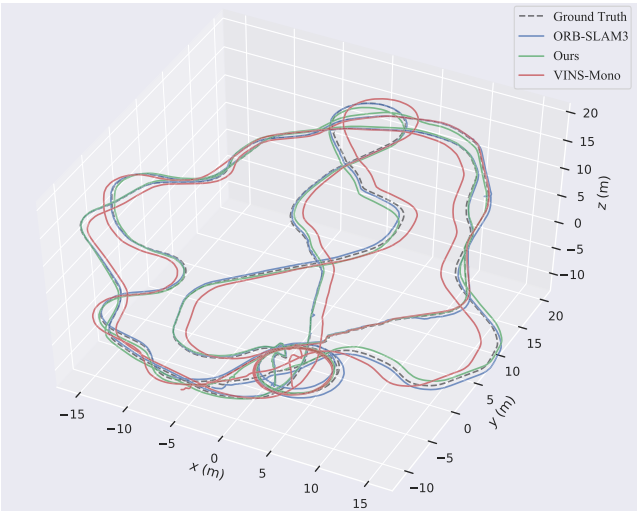


Fig. 11. Comparisons of the camera trajectories in the simulated scenario, which are displayed in a coordinate grid. The blue, red, and green lines denote the results of ORB-SLAM3 [32], VINS-Mono [13], and our method, respectively. The black dotted line represents the ground truth. (Please zoom in for better viewing.)

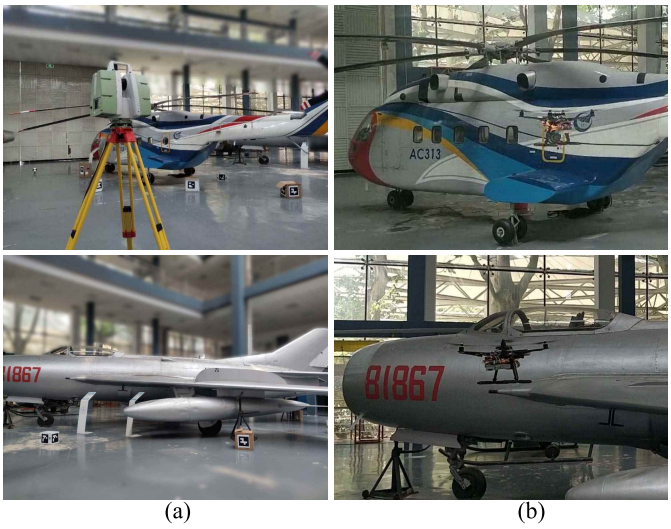


Fig. 12. Scenes for real inspection experiments. The first row is the scene for the helicopter. The second row is for the fighting jet. (a) Laser scanner is used to capture the aircraft and markers for calibration (real scene). (b) UAV is inspecting the aircraft.

VINS-Mono. It can be deduced that due to the large moving distance and the texture-less airliner surface, classical VIO methods can hardly track features on the images and therefore produce large accumulated drifts. Our method can effectively reduce accumulated drifts by introducing external constraints into the optimization, thereby improving localization accuracy.

The camera trajectories of our method and the competitors are visualized in Fig. 11. The results of ORB-SLAM3 and VINS-Mono are shown with the blue and red lines. Ours is shown with the green lines. Ground truth is displayed with the black dotted line. It can be seen that VINS-Mono has a large drift with the ground truth on most locations. Both ORB-SLAM3 and our method are close to the ground truth. However, ORB-SLAM3 shows large drifts on multiple small loops. Moreover, since the feature extraction and matching stage of ORB-SLAM3 is considerably time-consuming, the method is hard to achieve real-time localization in our experiment.

B. Experiments on Real Data

1) Experimental Settings: To evaluate the effectiveness of the proposed localization algorithm in real applications, we conduct experiments on two types of aircraft: a fighter jet and a helicopter [see Fig. 12(a)]. The above two types of aircraft have smaller sizes than the A350 airliner simulated in the synthetic scenario. Hence, four markers with the size of 0.15 m $\times$ 0.15 m are deployed in the scene to provide external constraints for each aircraft. In contrast to the synthetic scene, the ground truth of the camera trajectory cannot be obtained in the real scene. Consequently, to verify the effectiveness of our localization algorithm and other competitors, additional auxiliary markers are deployed. The auxiliary marker is not

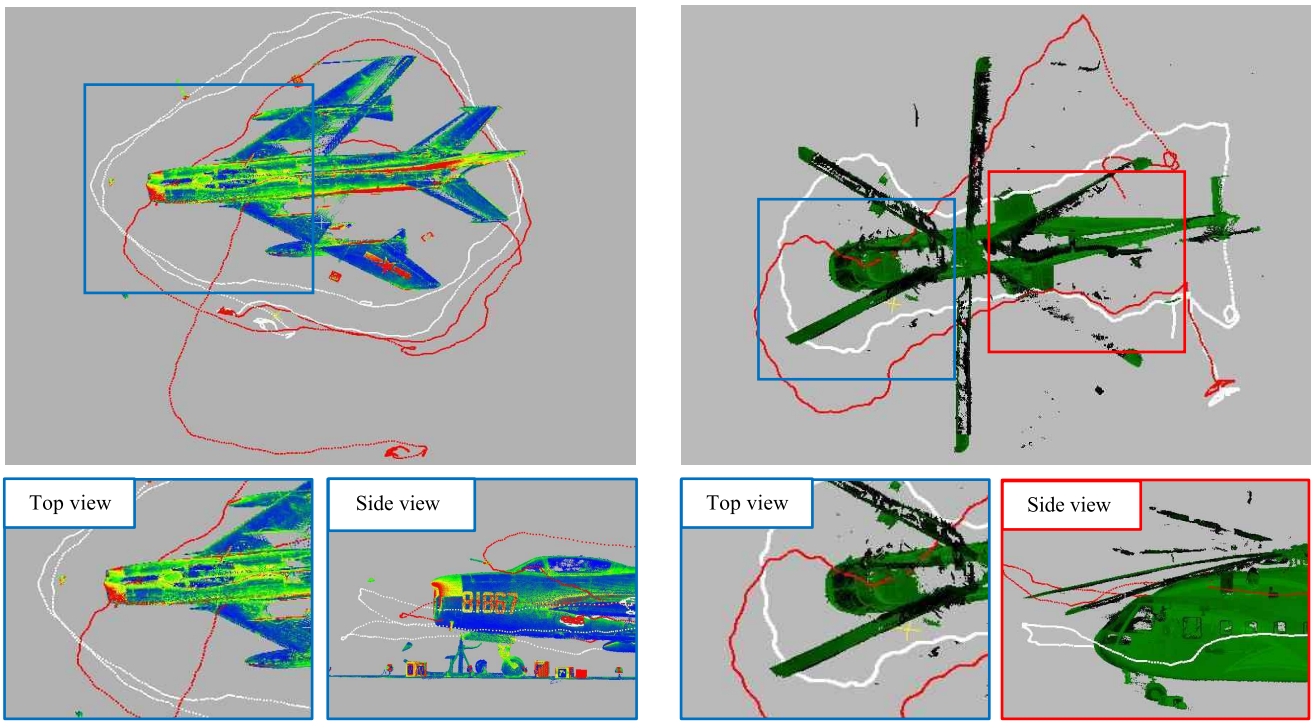


Fig. 13. Comparisons of the estimated trajectories between the VINS-Mono [13] (the red curves) and ours (the white curves). It can be seen that compared to VINS-Mono, our approach has fewer drifts when inspecting the aircraft.

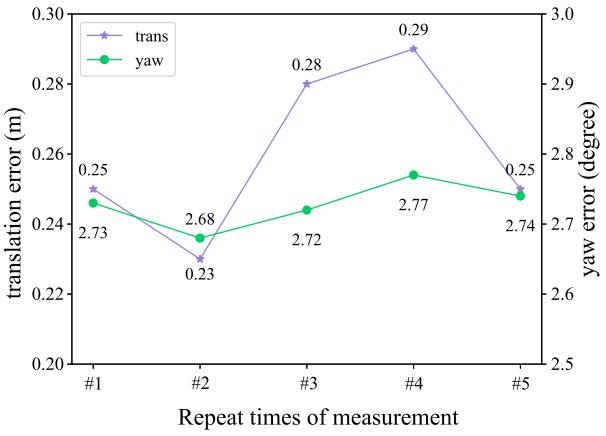


Fig. 14. Evaluations for repeated accuracy of our system. Five experiments are conducted. The purple curve shows the errors of translation, and the green curve represents the errors of the yaw angle.

involved in the optimization procedure and is intended to provide the ground truth. Through capturing and detecting the auxiliary markers, the camera poses can be inferred, which serve as reference values to evaluate the predicted values from optimization. In our experiments, five auxiliary markers are additionally deployed for both two aircraft. In addition, an extra marker is used to realize coordinate system alignment for trajectory visualization. For our adaptive weighting method, based on the characteristics of the localization camera, k and d_0 in (10) are set to 1.7 and 4, respectively.

2) Calibrations: The detection camera has a field angle greater than 180°, which is suitable for the fish-eye model.

TABLE III
APE AT DIFFERENT DISTANCES (UNIT: m) IN REAL SCENES

Aircraft	Distance	ORB-SLAM3 [32]		VINS-Mono [13]		Ours	
		APE (*trans*: m, *yaw*: °)					
		trans.	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*
Fighter jet	50	0.21	1.35	0.20	**1.31**	**0.19**	1.33
	100	0.32	2.81	0.29	**2.73**	**0.25**	**2.73**
	200	0.50	5.11	0.49	4.92	**0.41**	**4.14**
Helicopter	50	**0.20**	2.18	0.21	**1.83**	**0.20**	1.92
	100	0.29	3.57	0.31	3.94	**0.25**	**3.41**
	200	0.52	6.01	0.54	6.27	**0.43**	**5.39**

The MEI method [46] is thus applied to calibrate the camera. The localization camera is calibrated using a pin-hole model, specifically, Zhang's method [47]. All calibrations are achieved by Camodocal toolbox [48]. In order to calibrate the relative pose of markers, a laser scanner (Leica ScanStation P20) is first used to obtain the point cloud of the hangar, as shown in the first row of Fig. 12(a). The markers are then searched in the scene cloud. A local coordinate frame is built on each marker in the point cloud, according to the predefined frame of the marker. The relative pose between markers is finally computed. Note that, according to our localization method, only the relative pose of adjacent markers is computed.

3) Comparisons: Since our method is based on VINS-Mono, our algorithm is first compared with the original VINS-Mono to evaluate the effect of the external constraints-based localization module. The qualitative results are shown in Fig. 13. The left shows the results on the fighter jet, while the right depicts the results on the helicopter. It can

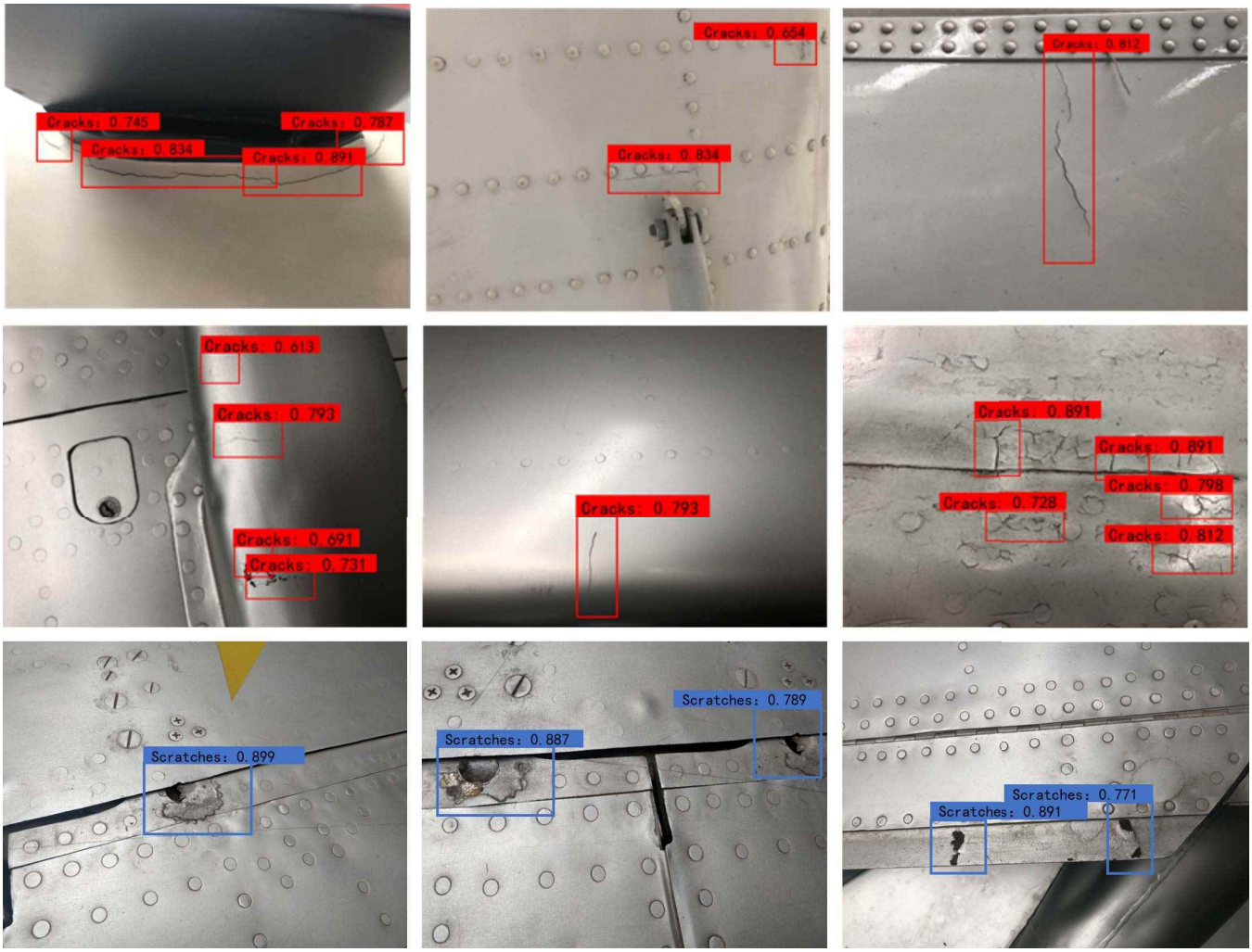


Fig. 15. Defect detection results for aircraft surface. The first and second rows are the results for cracks. The third exhibits the results for scratches.

be observed from the top views that the camera trajectory predicted by VINS-Mono (shown with the red curves) has passed through the aircraft in both cases, which would not appear in real situations, indicating its inaccuracy. In contrast, the camera trajectories of our method (shown with the white curves) are more accurate. The results indicate that our method is considerably better than VINS-Mono. The reason can be inferred that the low-texture aircraft surface has a significant impact on the VINS-Mono algorithm for tracking images using Harris features, causing large drifts when the camera constantly moves.

For quantitative results, our method is compared with VINS-Mono and ORB-SLAM3 using the APE metric as in synthetic experiments. The results are reported in Table III. Since the aircraft used for experiments are smaller than the A350 airliner, the trajectory length in the real scene is shorter. It can be observed that at the beginning of the inspection, the drifts of all methods are close. When the moving distance increases, VINS-Mono and ORB-SLAM3 yield large drifts due to the low-texture aircraft surface and feature mismatching. Comparatively, our method detects markers to assist the optimization of camera poses, generating less drifts and achieving the best performance. Moreover, the translation error of our method does not exceed 0.45 m, which can meet the requirements for accurate localization of aircraft defects.

In order to evaluate the repeated accuracy of our system, multiple repeated experiments are further conducted. The aircraft for evaluation is the fighter jet and the inspection distance is 100 m. The inspection is repeated five times. The results are reported in Fig. 14. It can be observed that the maximum fluctuation range is 0.06 m for the translation error and 0.09° for the yaw angle error, indicating a relatively good stability of our system. The reason is that the deployed markers provide effective constraints to reduce the drifts and thereby improve the accuracy.

C. Ablation Study

In our method, the confidences of the external constraints fundamentally depend on the detection accuracy of markers. If a marker is detected inaccurately, the constraint generated by the marker would have a large error, misleading the optimization of the camera poses. Consequently, weights are needed to balance all external constraints. In this section, ablation experiments are carried out to verify the effectiveness

TABLE IV
COMPARISONS ON APE AT DIFFERENT DISTANCES BETWEEN A FIXED WEIGHT AND THE ADAPTIVE WEIGHTS

Distance (m)	Fixed weight										Adaptive weight	
	0.2		0.4		0.6		0.8		1.0			
	APE (*trans*: m, *yaw*: °)											
	trans.	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*	*trans.*	*yaw*
150	1.21	6.36	2.35	7.54	3.17	9.09	3.86	9.93	9.12	9.95	**0.19**	**2.81**
300	3.01	9.17	4.52	10.54	4.99	12.48	5.21	12.99	5.23	12.99	**0.41**	**5.94**
450	4.63	12.01	5.68	15.90	6.34	17.48	6.80	17.98	6.92	18.17	**0.51**	**9.44**

of the proposed adaptive weighting method used in the ECLM. We generate several competitors which use different fixed weights ranging from 0 to 1. Since the ground truth of the camera trajectory can be obtained in the simulation scene, the experiments are conducted on the synthetic data for comparisons. Table IV shows the comparative results under the metric of APE. It can be concluded that using adaptive weights significantly outperforms fixed weight methods. The reason is inferred that using a fixed weight equalizes different detection errors caused by different marker qualities. While a large weight ignores the detection errors from a low-quality marker, a small weight impairs the external constraints such that the VINS-Mono algorithm plays a major role. Contrastively, we present a score function [see (10)] to assess the detection accuracy according to the quality of the marker in an image, as illustrated in Section IV-D. The weights are assigned to constraints according to the detection score of the marker, which is more reasonable and more effective.

D. Defect Detection

The results are shown in Fig. 15. It can be seen that the cracks and scratches are successfully detected with high confidence. For multiple instances of defect, our method can also effectively detect them. Note that, for a vimineous crack, our method tends to recognize it as several cracks (see the top left of Fig. 15). It should be explained that a threshold has been set in the network to reduce the rate of error identification, which causes insensitivity for the network to detect vimineous cracks. In real applications, such a case does not affect the maintenance. The results indicate that our method has a satisfactory discernment for the two defects, which proves the feasibility of Yolov4 [15] combined with the transfer learning strategy for accurate detection of aircraft surface defects. Furthermore, comparisons with manual inspection are carried out on the real fighter jet aircraft. We invite an experienced worker to inspect the aircraft carefully. The results are shown in Table V. Note that the time of our approach represents the total time in the localization stage and the detection stage. It can be concluded that our method significantly improves the aircraft inspection accuracy and efficiency, which is consequential for aircraft maintenance.

E. Limitations

Although our system generally exhibits remarkable performance, there are two major limitations. One limitation is that when certain cracks are too slight to be recognized from the collected images (even by humans), our system would fail to detect the cracks. A possible solution is to increase the resolution of the detection camera such that slight cracks can be clearly imaged. Another limitation is that the UAV is typically restricted to a safe distance from the aircraft during the inspection to ensure the safety of both the UAV and the aircraft. Consequently, for the defects located in narrow or hidden areas inaccessible to the UAV, our system is difficult to detect. Nevertheless, it should be noted that the unreachable areas only take a small percentage of the whole fuselage surface. As an approach to auxiliary inspection, our system can still significantly improve inspection efficiency.

TABLE V
COMPARISONS OF OUR METHOD AGAINST MANUAL INSPECTION IN ASPECTS OF DEFECT NUMBER AND TIME

Method	Number of detected defects	Time
Manual inspection	31	3.1 hours
Our approach	87	12 minutes

VII. CONCLUSION AND FUTURE WORK

In this work, a novel aircraft surface inspection system based on the UAV is presented to improve the aircraft maintenance efficiency and defect localization accuracy. The advantages of our system include three aspects. First, compared to the state-of-the-art methods, the localization drifts are significantly reduced through utilizing the proposed ECLM. Second, the localization accuracy is further improved by designing a score mechanism for ArUco marker to evaluate the quality of markers in the camera view and generate adaptive weights for external constraints. Third, the defects on the fuselage surface are detected with high accuracy and efficiency through the presented deep learning method with the transfer learning strategy. For real applications, our system can effectively serve as an auxiliary approach for rapid aircraft surface inspection, which is of great significance to the field of aircraft maintenance. Furthermore, the integration of multiple sensors and the methods transferring from other fields, such as computer vision and pattern recognition considerably, enrich the domain of instrumentation and measurement.

Our system has an insufficient defect detection capability when the defect is extremely slight or when the defect locates in a hidden or narrow area, as illustrated in Section VI. Consequently, in the future, we would like to introduce image preprocessing methods into our pipeline to enhance the recognition effects for slight defects. Our network would also be

improved to obtain better robustness. Moreover, the localization accuracy of our algorithm would be further improved to reduce the dependence of visual localization on features in the scene.

REFERENCES

[1] M. Pan, Y. He, G. Tian, D. Chen, and F. Luo, "PEC frequency band selection for locating defects in two-layer aircraft structures with air gap variations," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 10, pp. 2849–2856, Oct. 2013.

[2] D. A. Marx and R. C. Graeber, "Human error in aircraft maintenance," in *Aviation Psychology in Practice*. Abingdon, U.K.: Routledge, 2017, pp. 87–104.

[3] C. Drury, "Human reliability in civil aircraft inspection," State Univ. New York Buffalo Dept. Ind. Eng., Albany, NY, USA, Tech. Rep. ADP010442, 2001.

[4] M. W. Siegel, W. M. Kaufman, and C. J. Alberts, "Mobile robots for difficult measurements in difficult environments: Application to aging aircraft inspection," *Robot. Auto. Syst.*, vol. 11, nos. 3–4, pp. 187–194, Dec. 1993.

[5] Q. Xie *et al.*, "RRCNet: Rivet region classification network for rivet flush measurement based on 3-D point cloud," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[6] J. Miranda, S. Larnier, A. Herbulot, and M. Devy, "UAV-based inspection of airplane exterior screws with computer vision," in *Proc. 14th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2019, pp. 1–7.

[7] S. Jung *et al.*, "Toward autonomous bridge inspection: A framework and experimental results," in *Proc. 16th Int. Conf. Ubiquitous Robots (UR)*, Jun. 2019, pp. 208–211.

[8] A. Khaloo, D. Lattanzi, K. Cunningham, R. Dell'Andrea, and M. Riley, "Unmanned aerial vehicle inspection of the placer river trail bridge through image-based 3D modelling," *Struct. Infrastruct. Eng.*, vol. 14, no. 1, pp. 124–136, 2018.

[9] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 957–964.

[10] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 3923–3929.

[11] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 298–304.

[12] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[13] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[14] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[16] M. Siegel and P. Gunatilake, "Remote enhanced visual inspection of aircraft by a mobile robot," in *Proc. IEEE Workshop Emerg. Technol., Intell. Meas. Virtual Syst. Instrum. Meas. (ETIMVIS)*, May 1998, pp. 49–58.

[17] C. Alberts, C. Carroll, W. Kaufman, C. Perlee, and M. Siegel, "Automated inspection of aircraft," Carnegie-Mellon Inst. Res., Pittsburgh PA, USA, Tech. Rep. ADA350525, 1998.

[18] M. Siegel and P. Gunatilake, "Remote inspection technologies for aircraft skin inspection," in *Proc. IEEE Workshop Emergent Technol. Virtual Syst. Instrum. Meas.*, May 1997, pp. 78–79.

[19] T. S. White, R. Alexander, G. Callow, A. Cooke, S. Harris, and J. Sargent, "A mobile climbing robot for high precision manufacture and inspection of aerostructures," *Int. J. Robot. Res.*, vol. 24, no. 7, pp. 589–598, Jul. 2005.

[20] J. Shang, T. Sattar, S. Chen, and B. Bridge, "Design of a climbing robot for inspecting aircraft wings and fuselage," *Ind. Robot, Int. J.*, vol. 34, no. 6, pp. 495–502, Oct. 2007.

[21] N. Colin and F. Guibert, "Collaborative robot for visually inspecting an aircraft," U.S. Patent 9952593, Apr. 24, 2018.

[22] M. Futterlieb, V. Cadenat, and T. Sentenac, "A navigational framework combining visual servoing and spiral obstacle avoidance techniques," in *Proc. 11th Int. Conf. Informat. Control, Autom. Robot.*, Sep. 2014, pp. 57–64.

[23] M. Lakrouf, S. Larnier, M. Devy, and N. Achour, "Moving obstacles detection and camera pointing for mobile robot applications," in *Proc. 3rd Int. Conf. Mechatronics Robot. Eng. (ICMRE)*, Feb. 2017, pp. 57–62.

[24] L. Bjerregaard. *Aircraft Drone Inspection Technology*. Accessed: Aug. 23, 2022. [Online]. Available: https://aviationweek.com/mro/aircraft-drone-inspection-technology

[25] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 90–100, Feb. 2012.

[26] T. Ozaslan *et al.*, "Towards fully autonomous visual inspection of dark featureless dam penstocks using MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 4998–5005.

[27] T. Ozaslan *et al.*, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1740–1747, Jul. 2017.

[28] Y. Lu, Z. Xue, G. S. Xia, and L. Zhang, "A survey on vision-based UAV navigation," *Geo-Spatial Inf. Sci.*, vol. 21, no. 2, pp. 21–32, 2018.

[29] H. Duan and Q. Zhang, "Visual measurement in simulation environment for vision-based UAV autonomous aerial refueling," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2468–2480, Sep. 2015.

[30] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[31] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 834–849.

[32] C. Campos, R. Elvira, J. J. Gómez Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," 2020, *arXiv:2007.11898*.

[33] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang, "Visual-inertial localization with prior LiDAR map constraints," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3394–3401, Oct. 2019.

[34] Y. Lin *et al.*, "Autonomous aerial navigation using monocular visual-inertial fusion," *J. Field Robot.*, vol. 35, no. 1, pp. 23–51, Jan. 2018.

[35] G. C. Giakos, L. Fraiwan, N. Patnekar, S. Sumrain, G. B. Mertzios, and S. Periyathamby, "A sensitive optical polarimetric imaging technique for surface defects detection of aircraft turbine engines," *IEEE Trans. Instrum. Meas.*, vol. 53, no. 1, pp. 216–222, Feb. 2004.

[36] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, "Automatic detection and classification of sewer defects via hierarchical deep learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1836–1847, Oct. 2019.

[37] Q. Xie, D. Li, Z. Yu, J. Zhou, and J. Wang, "Detecting trees in street images via deep learning with attention module," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5395–5406, Aug. 2020.

[38] D. Li *et al.*, "Automatic defect detection of metro tunnel surfaces using a vision-based inspection system," *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101206.

[39] I. Jovancevic, S. Larnier, J.-J. Orteu, and T. Sentenac, "Automated exterior inspection of an aircraft with a pan-tilt-zoom camera mounted on a mobile robot," *J. Electron. Imag.*, vol. 24, no. 6, Nov. 2015, Art. no. 061110.

[40] M. Rice *et al.*, "Automating the visual inspection of aircraft," in *Proc. Singapore Aerosp. Technol. Eng. Conf. (SATEC)*, vol. 7, Feb. 2018, pp. 1–5.

[41] J. Chen, Z. Liu, H. Wang, A. Nunez, and Z. Han, "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 2, pp. 257–269, Dec. 2017.

[42] J. Miranda, J. Veith, S. Larnier, A. Herbulot, and M. Devy, "Machine learning approaches for defect classification on aircraft fuselage images aquired by an UAV," *Proc. SPIE*, vol. 11172, Jul. 2019, Art. no. 1117208.

[43] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the PNP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, p. 155, 2009.

[44] J. Shi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[45] T. Qin, P. Li, and S. Shen, "Relocalization, global optimization and map merging for monocular visual-inertial SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1197–1204.

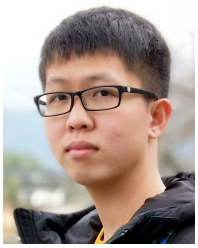
[46] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3945–3950.
[47] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Sep. 1999, pp. 666–673.
[48] L. Heng, B. Li, and M. Pollefeys, "CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1793–1800.

Yuanpeng Liu received the bachelor's degree in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2016, where he is currently pursuing the Ph.D. degree.

His research interests include robotics and deep learning.

Jingxuan Dong received the bachelor's degree in aircraft manufacturing engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2018, where she is currently pursuing the master's degree.

Her research interests include simultaneous localization and mapping (SLAM), and deep learning.



Yida Li received the bachelor's degree in aircraft manufacturing engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2019, where he is currently pursuing the master's degree.

His research interests include computer version and robotics.

Xiaoxi Gong received the bachelor's and master's degrees in electrical and electronic engineering from the University of Bradford, Bradford, U.K., in 2012 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China.

His research interests include robotics, vision-based simultaneously localization and mapping, and autonomous driving systems.

Jun Wang received the bachelor's and Ph.D. deg rees in computer-aided design from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 2002 and 2007, respectively.

He is currently a Professor with NUAA. From 2008 to 2010, he conducted research as a Post-Doctoral Scholar at the University of California, Davis, CA, USA, and the University of Wisconsin, Milwaukee, WI, USA. From 2010 to 2013, he worked as a Senior Research Engineer at Leica Geosystems, Norcross, GA, USA. In 2013, he paid an academic visit to the Department of Mathematics, Harvard University, Cambridge, MA, USA. His research interests include geometry processing and geometric modeling.