A Robust Learned Feature-based Visual Odometry System for UAV Pose Estimation in Challenging Indoor Environments

Leijian Yu, *Graduate Student Member, IEEE*, Erfu Yang, *Senior Member, IEEE*, Beiya Yang, *Graduate Student Member, IEEE*, Zixiang Fei, *Member, IEEE* and Cong Niu, *Member, IEEE*

***Abstract*—Unmanned Aerial Vehicles (UAVs) are becoming popular nowadays due to their versatility and flexibility for indoor applications, such as the autonomous visual inspection for the inner surface of a pressure vessel. Nevertheless, robust and reliable position estimation is critical for completing these tasks. Visual Odometry (VO) and Visual Simultaneous Localisation and Mapping (VSLAM) allow the UAV to estimate its position in unknown environments. However, traditional feature-based VO/VSLAM systems struggle to deal with complex scenes such as low illumination and textureless environments. Replacing the traditional features with deep learning-based features provides the advantage of handling the challenging environment, but the efficiency is ignored. In this work, an efficient VO system based on a novel lightweight feature extraction network for UAV onboard platforms has been developed. The Deformable Convolution (DFConv) is utilised to improve the feature extraction capability. Owing to the limited onboard computing capability, the Depthwise Separable Convolution (DWConv) is adopted to calculate the offsets for the deformable convolution and construct the backbone network to improve the feature extraction efficiency. Experiments on public datasets indicate that the efficiency of the VO system is improved by 30.03% while preserving the accuracy on embedded platforms with the feature points and descriptors detected by the proposed Convolutional Neural Network (CNN). Moreover, the proposed VO system is verified through UAV flying tests in a real-world scenario. The results prove that the proposed VO system is able to handle the challenging environments where both the latest traditional and deep learning feature-based VO/VSLAM systems fail, and it is feasible for UAV self-localisation and autonomous navigation in the confined, low illumination and textureless indoor environment.**

***Index Terms*—Unmanned aerial vehicles, Visual odometry, Deep learning-based features, Depthwise separable convolution, Improved deformable convolution.**

I. Introduction

The use of Unmanned Aerial Vehicles (UAVs) in indoor environments is gaining the increasing attention due to their flexibility, such as for inspecting the inner surface of a pressure vessel [1], [2]. For the UAV to fly steadily and perform tasks autonomously with increased efficiency and reduced human costs, providing accurate position information is crucial [3]–[5].

The Global Positioning System (GPS) is one of the most common solutions for estimating the position for UAV platforms [6]. However, GPS signals are not always available due to the possibility of being blocked by buildings, and large drifting errors affect the flying stability of the small-sized UAVs. To substitute the GPS system, various localisation methods based on Light Detection and Ranging (LiDAR) sensors, Time-of-Flight (ToF) sensors and visual sensors have emerged. Due to the stable performance in varied lighting environments and precise results, some works adopted LiDAR-based localisation technologies for their UAV platforms [7] [8]. However, the high cost and massive weight still cannot be ignored for small UAV platforms. The ultrasonic and Ultra-wideband (UWB) sensors are light and cost-efficient for UAV localisation in confined spaces. Nevertheless, the requirement for deploying the auxiliary anchor nodes restricts their applications. The advancements in cheaper and higher resolution cameras enabled the development of Visual Odometry (VO) and Visual Simultaneous Localisation and Mapping (VSLAM) technologies that can provide accurate position information at a low cost. In recent years, some advanced feature-based VO/VSLAM systems have emerged. Owing to the sparse feature points utilised for pose estimation, these VO/VSLAM systems are suitable for embedded computers on UAVs [9].

Traditional VO/VSLAM algorithms apply hand-engineered features that represent feature regions in the image to calculate the pose of the camera and map points of the surrounding environment. The Parallel Tracking and Mapping (PTAM) [10] is based on the Features from Accelerated Segment Test (FAST) [11] feature extraction approach. After that, the FAST feature extraction method is improved by combing with the Binary Robust Independent Elementary Features (BRIEF) and adopted into the most famous ORB-SLAM series [12]–[14]. These VSLAM systems are effective in general environments. However, when they are deployed into a dark and textureless scenario, such as inside of a pressure vessel, their performance is significantly degraded, and they may even be lose effectiveness for the localisation of the camera.

Recently, as deep learning has defined the state-of-the-art in many research areas [15] [16], adopting deep learning-based features into VO/VSLAM systems has gained increasing

This work is supported in part by the UK Net Zero Technology Centre (NZTC) under the LOCUST research project (2019-2021, Grant No.: AI-P-028). It is also supported partially by the Royal Society under the MOEA/D-PPR research project (2022-2024, Grant No.: IECNSFC211434). Mr. Leijian Yu is funded by the China Scholarship Council and the International Fees Only Studentship from the University of Strathclyde (2018-2021). (Corresponding author: Erfu Yang.)

Leijian Yu, Erfu Yang, Beiya Yang and Cong Niu are with the Department of Design, Manufacturing and Engineering Management, University of Strathclyde, Glasgow, G1 1XJ, United Kingdom (e-mail: leijian.yu@strath.ac.uk; erfu.yang@strath.ac.uk; beiya.yang@strath.ac.uk; cong.niu@strath.ac.uk).

Zixiang Fei is with the School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China (e-mail: zxfei@shu.edu.cn).

interest from researchers. Compared to traditional features, the features extracted or described by Convolutional Neural Networks (CNNs), such as the ASLFeat [17] and GIFT [18], show significant improvements in accuracy and robustness in challenging environments. Therefore, incorporating deep learning-based features has the potential to improve pose estimation accuracy [19]. The DBLD-SLAM replaces the ORB descriptor in conventional ORB-SLAM with a learned deep binary feature descriptor [20], and the LIFT-SLAM [21] integrates the deep features trained by the Structure-from-Motion (SFM), to outperform other state-of-the-art SLAM systems. The deployment of deep learning-based methods requires powerful Graphics Processing Units (GPU), thereby making them impractical for real-world use cases [22]. To some extent, efficiency is sacrificed to enhance the robustness and accuracy of VO/VSLAM systems. To this end, it is vital to strike a balance between accuracy and efficiency while deploying deep learning-based features into VO/VSLAM systems [23]–[25], especially for UAV platforms with limited payload capacity. To the best of our knowledge, the GCNv2-tiny based SLAM [26] is the only learned feature-based VSLAM system that achieves real-time performance on the most popular UAV onboard computing platform, the Nvidia Jetson TX2. It uses Shi-Tomasi corners [27], a type of hand-crafted feature point, as groundtruth for training the network, limiting its localisation performance in complex environments. Therefore, using VO/VSLAM to locate the UAV with onboard computing resources in low illumination and textureless scenes is still challenging.

Considering that monocular VO systems are particularly appealing for many robotic applications due to their use of a lightweight camera and their requirement for an easy calibration process [28], this paper proposes a robust and effective monocular VO system for UAV onboard platforms to be deployed in challenging indoor environments, such as inside of a pressure vessel. Specifically, an efficient CNN model is proposed for keypoint detection and description. The designed network deals with constraints on computation as the UAV has limited computing resources. Our system concentrates on the localisation of the UAV by processing and integrating data from imaging instruments and computing instruments on the UAV. This broadens the scope of instrumentation and measurement for the UAV in low illumination and textureless indoor environments. The main contributions of this paper are summarised as follows:

(1) A robust UAV onboard VO system is developed based on the features extracted by an efficient CNN model in complex environments.

(2) The Depthwise Separable Convolution (DWConv) is adopted to reduce the computing complexity, and the Deformable Convolution (DFConv) is improved by combing with the DWConv to extract feature points robustly and efficiently.

(3) Extensive evaluations on the public datasets, real-world scenarios and UAV flying tests are carried out to confirm the advancement of the proposed system.

The remaining paper is organised as follows: Section 2 introduces the related works. Section 3 addresses the proposed VO system based on the efficient feature extraction network. In Section 4, the experimental findings and analysation are provided. The whole paper is concluded in Section 5.

II. Related Works

As there are not many works aiming to deploy deep learning feature-based VO and VSLAM systems on UAVs, we will focus on introducing general deep learning feature-based VO and VSLAM systems in this section. In [23], a framework consisting of CNN and recurrent neural network (RNN) was developed to detect keypoints and their corresponding descriptors for pose estimation. The performance of the whole system is on par with the ORB-SLAM2. DF-SLAM [29] combines the FAST detector for keypoint detection with the TFeat network [30] for feature point description. The feature points and descriptors are then deployed into the ORB-SLAM2, and the DF-SLAM outperforms the ORB-SLAM in some image sequences. The HF-Net [31] is incorporated into the DXSLAM [32] to extract local and global features. Compared with traditional VSLAM systems, the robustness of the DXSLAM in different environments has been improved significantly. The SuperPointVO [33] follows the scheme of traditional systems while adopting the SuperPoint [34] to replace the hand-engineered feature extraction, and it achieves a close performance to the advanced VO systems. The LIFT-SLAM [21] employs the Learned Invariant Feature Transform (LIFT) [35] to replace the ORB feature extraction module, and the robustness of the VSLAM system is enhanced. These methods leverage the stability and robustness of the deep learning-based feature extraction module into the VO/VSLAM pipeline to obtain accurate localisation performance. However, these works focus on improving localisation while ignoring efficiency. In other words, integrating deep learning methods into the VO/VSLAM pipeline in performance-constrained platforms, such as UAV onboard platforms, is still an open problem [36]. There are also some works starting to improve the efficiency of the deep learning feature-based VO/VSLAM systems. To this end, several simplified networks have been developed and integrated into the VO/VSLAM pipeline. MobileNetV2 [37] is used as an encoder and trained using the knowledge distillation method in [38] to reduce the model size and increase the running speed greatly. A simple network whose backbone consists of four convolutional layers is proposed in [24]. A quantised local feature extraction module is described in [39]. To save processing time, only keypoints extracted by the SuperPoint are used in [4]. Even so, most of these methods still need more execution time than traditional VSLAM algorithms, and they are not suitable for embedded computing platforms.

III. Proposed Approach

A. *Overall Structure of the VO System*

Fig. 1 shows the overall structure of the VO system. It is adopted from SP-ORB-SLAM [40], which leverages the learned repeatability and description. The network to generate the keypoints and descriptors is shown in Fig. 2. The heatmap predicted by the keypoint detection branch is supposed to be the repeatability map, and it is sampled as 2D grids. After that, sparse features could be extracted. The repeatable features can

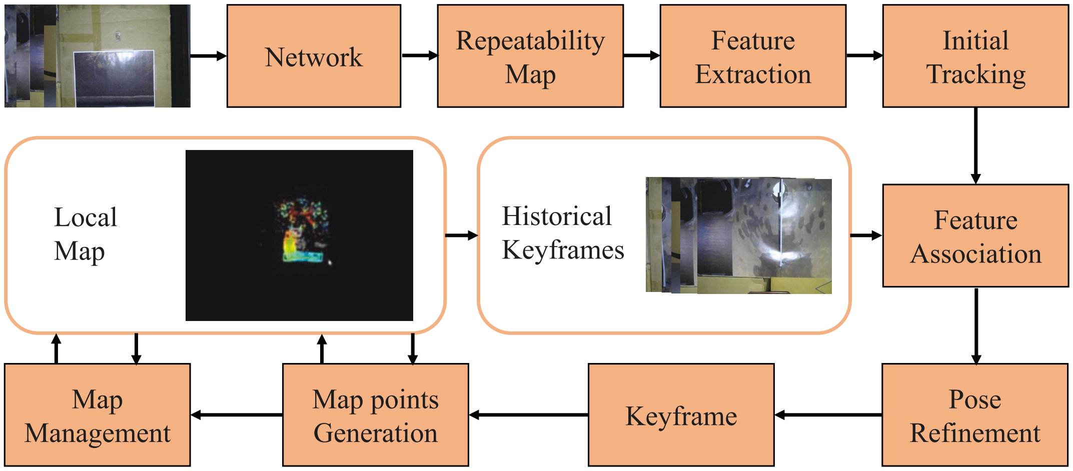


Fig. 1. The diagram of the VO system

be identified across different images, and their locations are considered as the local peaks of the repeatability map. The track-on-repeatability approach is adopted to directly localise the camera coarsely. After that, the landmarks are associated with local feature points, and the camera pose is optimised by minimising the reprojection error. The association step relies on matching feature descriptors significantly. The approximate nearest neighbour (ANN) search [41] and epipolar line search are deployed to match feature points. When a new keyframe is determined, the feature points observed by the previous keyframes are mapped as the map points. Finally, the map maintenance process culls redundant keyframes and deletes outliers.

This work mainly focuses on the network part to generate the reliable feature points and descriptors, which have a significant impact on subsequent processing. More details are given as follows:

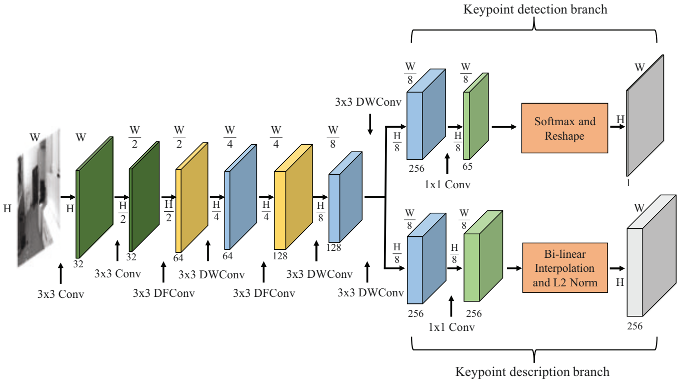


Fig. 2. The structure of the network for feature detection and description

B. Proposed Network

Considering that the deep learning feature-based VO system needs to focus on the deployment of UAV platforms in complex environments, such as the low illumination and textureless scenes, the efficiency and accuracy of the feature extraction module should be balanced. Inspired by the SuperPoint [34], which is trained by the self-supervision method and achieves the desirable homography estimation results,a novel efficient CNN model for feature extraction is proposed. The DFConv [42] is utilised to improve the feature extraction capability. Moreover, the DWConv [43] is adopted to reduce parameters, and it is also adopted to the DFConv to improve the feature extraction capability. The proposed feature detection and description module are shown in Fig. 2. It shows that the model includes a shared backbone network, followed by two sub-modules for feature point detection and description. Traditional convolutional layers, DWConv layers and DFConv layers are deployed to process input images. The details are introduced as follows:

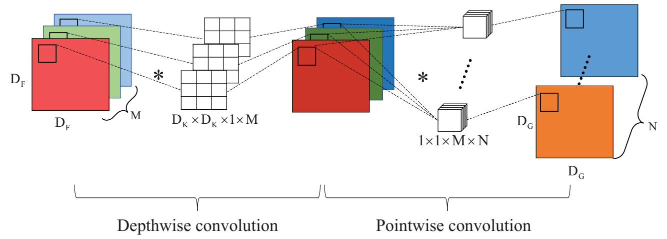


Fig. 3. Process of the DWConv

1) Depthwise Separable Convolution: To process the feature maps with weight and height represented by F_{iw} and F_{ih} and M channels to output feature maps with dimensions of $F_{ow} \times F_{oh} \times N$ by the $k \times k$ kernel, the computational cost for the standard convolution is

$$C_{sc} = k \times k \times M \times F_{ow} \times F_{oh} \times N \quad (1)$$

The core concept of the DWConv is to decompose the traditional convolution layer into a depthwise convolutional layer and a pointwise convolutional layer, which reduces the computational load and model size. The overall process of the DWConv is shown in Fig. 3.

In the depthwise convolution process, each input channel is processed by a single convolution kernel. Thus, there will be M sets of convolutional kernels to process the input image. The computational cost for the depthwise convolution is

$$C_d = k \times k \times M \times F_{ow} \times F_{oh} \tag{2}$$

Then, the generated feature map will be processed by 1×1 traditional convolution. The pointwise convolution has a computational cost of

$$C_p = 1 \times 1 \times M \times F_{ow} \times F_{oh} \times N \tag{3}$$

The total parameter of the DWConv can be represented by:

$$C_{dw} = k \times k \times M \times F_{ow} \times F_{oh} + 1 \times 1 \times M \times F_{ow} \times F_{oh} \times N \tag{4}$$

The comparison of the computational costs for the DWConv and standard convolution can be expressed by:

$$\begin{aligned} C_r &= \\ &\frac{k \times k \times M \times F_{ow} \times F_{oh} + 1 \times 1 \times M \times F_{ow} \times F_{oh} \times N}{k \times k \times M \times F_{ow} \times F_{oh} \times N} \\ &= \frac{1}{N} + \frac{1}{k^2} \end{aligned} \tag{5}$$

It demonstrates that the DWConv has high efficiency while increasing the size and channel of the convolutional kernel. The original MobileNet uses Batch Normalisation [44] and Rectified Linear Units (ReLU) activation function [45] for both layers to enhance the feature extraction performance. To reduce the computing complexity, the Exponential Linear Unit (ELU) [46] is chosen as the activation function for pointwise layers. The expression for the ELU is:

$$f(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \tag{6}$$

The hyperparameter α controls saturates for negative inputs. Unlike the ReLU only has positive values, the negative values of ELU push the mean unit activations closer to zero, which accelerates the training speed and improves the stability of the training process.

2) Deformable Convolution: The DWConv scarifies the accuracy to reduce the computational cost, and traditional CNN is difficult to accommodate geometric variations properly owing to the fixed geometric structures. In contrast to the traditional convolution, the DFConv layer adds a 2D offset to the sampling grid, which enables the free form deformation of the convolutional kernel. As shown in Fig. 4, the DFConv consists of two feature preprocessing channels. The upper channel learns the sampling locations for the convolutional kernel. To reduce the number of parameters and improve the robustness of the DFConv, the traditional convolution is replaced by the DWConv to calculate the 2D offset matrix. Then, the convolutional operation is performed between the input data and the deformed convolutional kernel accordingly. Thus, the DFConv can extract features from non-uniform shapes effectively.

In a traditional 3×3 convolutional kernel with dilation 1, the convolutional grid $\mathcal{R}$ can be formalised as

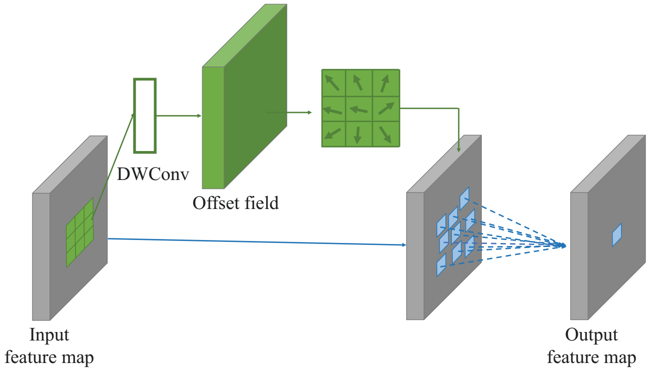


Fig. 4. Scheme of the DFConv

$$\mathcal{R} = \begin{Bmatrix} (-1,-1) & (-1,0) & (-1,1) \\ (0,-1) & (0,0) & (0,1) \\ (1,-1) & (1,0) & (1,1) \end{Bmatrix} \tag{7}$$

The output feature map on location p_0 can be obtained through:

$$y(p_0) = \sum_{p_r \in \mathcal{R}} w(p_r) \cdot x(p_0 + p_r) \tag{8}$$

where w indicates the convolutional weights. x represents the input feature map. p_r means the position in $\mathcal{R}$. In the DFConv, offsets $\{\Delta p_r | r = 1, 2, ..., |\mathcal{R}|\}$ is added to $\mathcal{R}$, and offset locations $p_r + \Delta p_r$ allows the convolutional kernel to form an irregular shape. Thereby, the DFConv is formulated as:

$$y(p_0) = \sum_{p_r \in R} w(p_r) \cdot x(p_0 + p_r + \Delta p_r) \tag{9}$$

As offset Δp_r learned by the DWConv is usually fractional, bilinear interpolation is implemented to revise the offset as an integer.

$$x(p) = \sum_{q} B(q, p) \cdot x(q) \tag{10}$$

where B denotes the bilinear interpolation kernel. p and q represent the fractional and integral location, respectively.

C. Training Process

To avoid the data labelling process, which is time-consuming and laborious, self-training is adopted. The training process is illustrated in Fig. 5. Firstly, the model provided in SuperPoint [34] is utilised to generate pseudo ground truth for unlabelled images. Moreover, the Homographic Adaptation (HA) [34] is employed to enlarge the dataset. The augmentation process can be represented by

$$\hat{A}(X; f_{ipa}) = \frac{1}{N_H} \sum_{i=1}^{N_H} H_i^{-1}(f_{ipa}(H_i(X))) \tag{11}$$

Where N_H is the number of the generated homography matrix. H and H^{-1} demonstrate the randomly generated homography

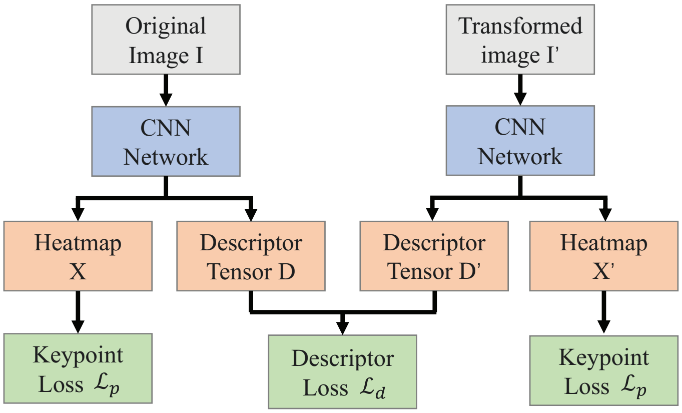


Fig. 5. Training process

matrix and the corresponding reverse, respectively. f_{ipa} indicates interest point adaption function.
The feature point loss function $\mathcal{L}_p$ is expressed as:

$$\mathcal{L}_p(X_o, L_o) = -\frac{1}{H_d W_d} \sum_{i=1,j=1}^{H_d,W_d} (l_{ij} log(x_{ij}) + (1 - l_{ij}) log(1 - x_{ij})) \tag{12}$$

Where $H_d = H/8$ and $W_d = W/8$. X_o is the heatmap generated by the keypoint detector branch. L_o indicates the ground truth for keypoints. To improve the training efficiency, M_p positive pairs and M_n negative pairs of descriptor cells are sampled from $(H_d W_d)^2$ of positive and negative pairs to train the model. Variables with $'$ indicate that these variables are extracted from the transformed image. The encoded descriptor tensors D is generated from the original image. u_{ij} represent the centre of the descriptor vector d_{ij}. The correspondence of the descriptor pair $\{d_{ij}, d'_{i'j'}\}$ can be calculated through:

$$c_{iji'j'} = \begin{cases} 1, & ||H(u_{ij}) - u'_{i'j'}||_2 \leqslant 8 \\ 0, & otherwise \end{cases} \tag{13}$$

With the positive margin m_p and negative margin m_n, a hinge loss for descriptor loss can be defined as:

$$\mathcal{L}_d(D, D'; H) = \frac{1}{(H_d W_d)^2} \sum_{i=1,j=1}^{H_d,W_d} \sum_{i'=1,j'=1}^{H_d,W_d} (\lambda c_{iji'j'} max(0, m_p - d_{ij}^T d'_{i'j'}) + (1 - c_{iji'j'}) max(0, d_{ij}^T d'_{i'j'} - m_p)) \tag{14}$$

Finally the training loss could be represented by

$$\mathcal{L}_{joint} = \mathcal{L}_p(X_o, L_o) + \mathcal{L}_p(X'_o, L'_o) + \mathcal{L}_d(D, D'; H) \tag{15}$$

IV. Experiments and Analysation

To test the effectiveness of the proposed system, experiments including feature point extraction, localisation performance analysation and UAV flying tests are carried out. A laptop is utilised for training the model, which is equipped with the Intel Core i7-7700HQ CPU, 16 GB of Random Access Memory (RAM) and an external GPU enclosure connected via Thunderbolt 3. In particular, the crate is equipped with an Nvidia Titan RTX GPU. The MS-COCO 2014 dataset, which consists of 80000 images, is utilised to train the proposed model, and images are resized to 240 × 320. The training is done using PyTorch 1.9.1 with CUDA 10.2. The model is optimised by the adam solver [47] with a learning rate of 0.0001. To improve the generalisation performance, data augmentation techniques such as random contrast and motion blur are adopted to enlarge the training dataset. The total training integration is 200000.

A. Datasets

HPatches [48] is a novel benchmark for local feature descriptor evaluation that contains 116 image scenes. There are 57 scenes that show large photometric changes, and the other 59 sequences have large viewpoint changes. Each sequence consists of 6 images and 5 ground-truth homographies between the first image and the others.
11 sequences captured by an AscTec Firefly drone make up the EuRoC dataset [49]. In a large machine hall, 5 sequences are collected using a Leica Multistation as ground truth. The remaining 6 sequences are recorded in a small room using a Vicon system to produce the ground truth. Without the multi-map system, most methods cannot accomplish the sequences V103 and V203, which threaten the safety of robots in real-world applications. Therefore, the effectiveness of VO/VSLAM approaches is evaluated by the remaining 9 sequences.
The ICL-NUIM dataset [50] includes image sequences under various lighting scenarios. As a result, it is appropriate to verify the effectiveness of VO/VSLAM systems in different illumination scenarios. In this study, office room sequences with static, local variation, global variation and local and global variation illumination conditions are employed to test VO/VSLAM systems.

TABLE I
Feature point extraction comparison

Feature extraction methods	Detector metric		Descriptor metric		Homography estimation		
	Rep.	MLE	MAP	MS	ϵ = 1	ϵ = 3	ϵ = 5
ORB	0.64	1.03	0.51	0.18	0.14	0.40	0.49
SIFT	0.51	1.16	0.70	0.27	0.63	0.76	0.79
SuperPoint	0.61	1.14	0.81	0.55	0.44	0.77	0.83
GCNv2	0.64	1.14	0.78	0.44	0.45	0.73	0.81
deepFEPE	0.63	1.07	0.78	0.42	0.46	0.75	0.81
Proposed method	0.63	1.20	0.75	0.41	0.38	0.70	0.78

B. Evaluation of Feature Point Extraction and Description

To evaluate the feature point detection and matching ability of the proposed model, detector metrics including the repeatability (Rep.) and Mean Localisation Error (MLE), descriptor metrics consisting of Mean Average Precision (MAP), Matching Score (MS) and the homography estimation metrics with

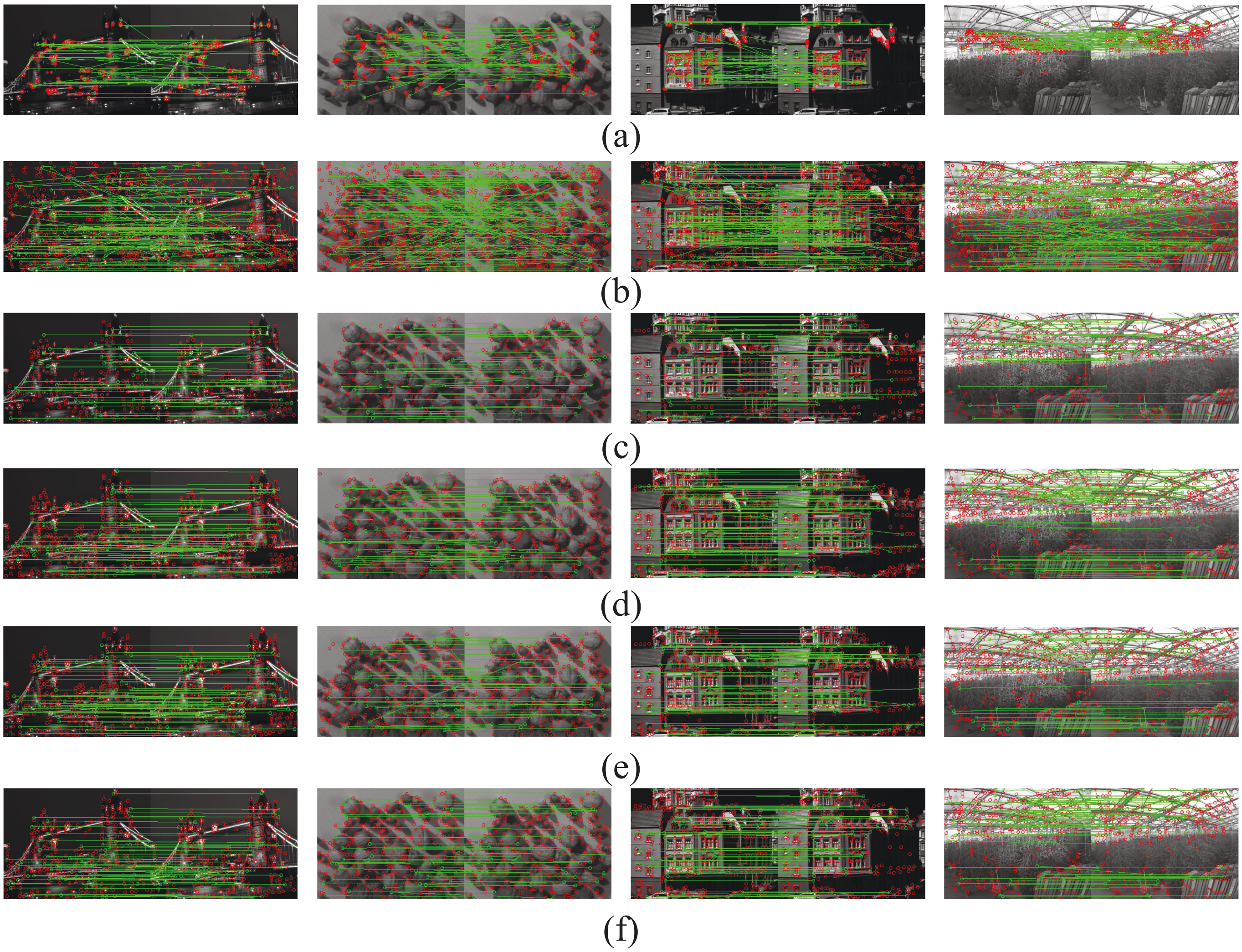


Fig. 6. Feature point extraction and matching: (a) ORB, (b) SIFT, (c) SuperPoint, (d) deepFEPE, (e) GCNv2, (f) the proposed model.

different thresholds are measured on the HPatches dataset. The proposed model is compared with the state-of-the-art deep learning-based feature extraction algorithms, including SuperPoint [34], deepFEPE [48] and GCNv2 [26], as well as traditional feature extraction methods such as ORB [51] and SIFT [52] implemented by OpenCV on the laptop used to train the model. The results are presented in Table I. The visualised feature extraction and matching results are shown in Fig. 6. The ORB achieves the highest Rep., but scores for descriptor-focused metrics are the lowest. Therefore, it cannot perform well in the homography estimation task. The SIFT detects feature points with sub-pixel accuracy. To this end, the best performance in homography estimation with the $\epsilon = 1$ is achieved. Compared with traditional feature extraction methods, the learned descriptors outperform artificially designed representations. In the homography estimation, SuperPoint achieves the best score with a tolerance threshold of 3 and 5. Our method is slightly less accurate than other learned features but still better than the ORB. Comparisons of the learned features towards model size (MB) and Floating Point Operations (FLOPs) are shown in Fig. 7. It indicates that the proposed method contains fewer parameters, and the FLOPs are reduced significantly compared to other methods. This is owing to the proposed efficient network structure, which achieves the balance between efficiency and accuracy.

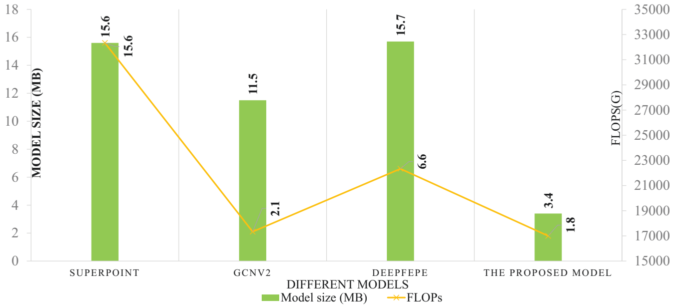


Fig. 7. Model size and FLOPs comparison of different VO/VSLAM methods

C. Evaluation of Trajectory Estimation

To verify the localisation accuracy of VO/VSLAM systems, the Absolute Trajectory Error (ATE) [53] is estimated for comparison. The ATE represents the absolute distance between the true trajectory and the calculated path. In the following parts, the mean ATE and Root Mean Square (RMS) ATE will be calculated to represent the localisation accuracy of VO/VSLAM systems. The comparisons of the proposed algorithm against the ORB-SLAM3, SP-ORB-SLAM, GCNv2-SLAM, HSO [54] and DPVO [22] are carried out on the EuRoC dataset and the ICL-NUIM dataset with simulated lighting changes. As we focus on the applications on the UAV platform, the loop closure and relocalisation modules are disabled.

1) Evaluation on the EuRoC Dataset: The evaluation on the EuRoC dataset has been carried out on a laptop with an Intel(R) Core (TM) i7-8750H CPU, 20GB of RAM and a GeForce GTX 1050 Ti 4GB graphics card. Similarly, the software used in the evaluation includes CUDA 10.2 and PyTorch 1.9.1. Table II gives the median value of the localisation results across 10 runs. The GCNv2-SLAM is unable to maintain a consistent scale for monocular sequences and as a result, it fails in all scenarios. In several sequences, the ORB-SLAM3 achieves the highest localisation accuracy. However, it is unable to finish the V102 and V202 sequences due to the fast motion and relatively low texture environment. Both the HSO and DPVO accomplish all sequences. Without photometric calibration, the accuracy of the HSO is significantly degraded in the dark environment. The DPVO achieves the worst performance among most of the tests. The best average localisation results are achieved by the SP-ORB-SLAM. The proposed method completes all the sequences and obtains the sub-optimal average localisation accuracy.

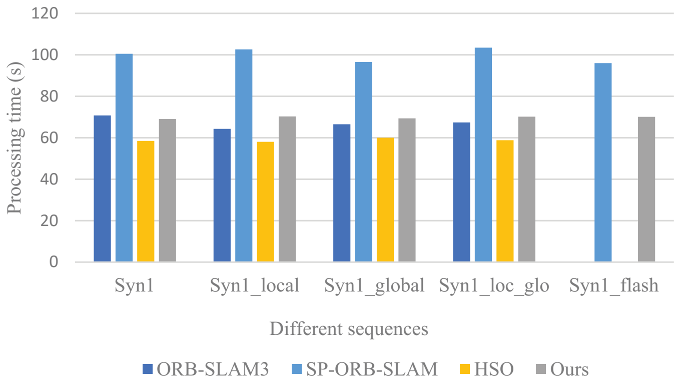


Fig. 8. Time usage comparison

2) Evaluation on the ICL-NUIM Dataset with Simulated Lighting Changes: To obtain the results of the deep learning-based VO/VSLAM methods, a powerful GPU is preferred. Because of the payload and power constraints on the UAV onboard platform, the high-power GPU is unavailable. Thus, a small-sized Nvidia Jetson TX2, which is the most popular onboard platform for UAVs, is utilised to verify the proposed method. It features a 256-core Pascal GPU and a 64-bit hex-core ARMv8 CPU, along with 8 GB of RAM. The Jetson TX2 is flashed with the Jetpack 4.6, and PyTorch 1.9.1 is also installed to run deep learning embedded VO/VSLAM approaches. Considering the limited computing resources of Jetson TX2, all images in the ICL-NUIM dataset with simulated lighting changes are resized to 320 × 240. Due to the DPVO only achieving around 2 fps on the laptop and its low localisation accuracy, it is not deployed on the Jetson TX2. The ORB-SLAM3, SP-ORB-SLAM, GCNv2-SLAM and HSO are utilised for comparison with the proposed method. Similarly, the median localisation accuracy of successful trajectory estimation from 10 executions is presented in Table III. Unlike the results obtained on the powerful laptop, the proposed method obtains the best performance on the Jetson TX2. Due to the observation of less texture information in the scenario with the flashlight, the ORB-SLAM3 fails to initialise and track the pose of the camera. Owing to the limited computing resources, the performance of the SP-ORB-SLAM is restricted. The HSO could keep a relatively stable performance under several different lighting conditions. Due to the poor initialisation performance in the scene with the flashlight, the HSO fails in this sequence. Comparisons for time usage and power consumption are depicted in Fig. 8 and Table IV, respectively. The HSO is the most efficient method. However, it cannot achieve the robust and accurate localisation performance. Compared to the SP-ORB-SLAM, the efficiency of the proposed method has improved by 30.03%, with a decrease in total power consumption by 37.31%.

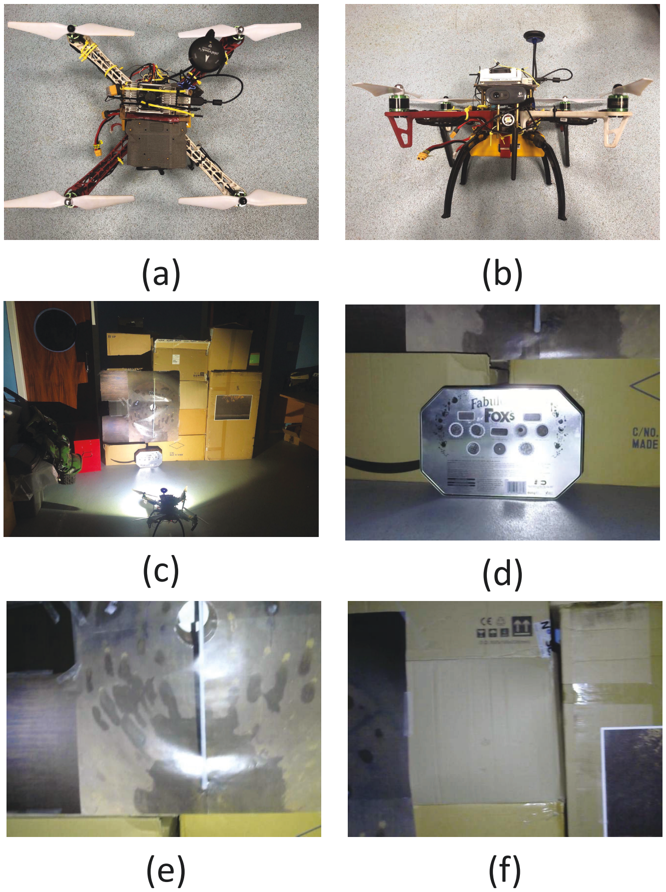


Fig. 9. An experimental environment setup. (a) The top view of the quadrotor, (b) the front view of the quadrotor, (c) the experimental environment, (d) a box for VSLAM initialisation, (e) and (f) two sample images captured by the onboard camera.

TABLE II
COMPARISON OF THE MEAN ATE (M) AND RMS ATE (M) IN THE SELECTED EUROC DATASET.

EuRoC	ORB-SLAM3		SP-ORB-SLAM		HSO		DPVO		The proposed method	
benchmark	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE
MH01	0.020	0.022	**0.011**	**0.012**	0.030	0.037	0.106	0.116	0.014	0.017
MH02	0.016	0.018	**0.011**	**0.012**	0.040	0.051	0.043	0.050	0.012	0.014
MH03	0.027	0.031	**0.023**	0.027	0.047	0.057	0.133	0.148	**0.023**	**0.026**
MH04	**0.074**	**0.081**	0.091	0.102	1.163	1.237	0.194	0.210	0.087	0.098
MH05	**0.036**	**0.041**	0.042	0.047	0.512	0.545	0.152	0.175	0.050	0.058
V101	0.031	**0.033**	0.032	0.034	**0.030**	**0.033**	0.045	0.049	0.035	0.038
V102	-	-	0.229	0.248	**0.058**	**0.075**	0.148	0.161	0.201	0.215
V201	0.026	**0.023**	**0.022**	0.024	0.024	0.028	0.064	0.075	0.027	0.036
V202	**-**	**-**	**0.035**	**0.052**	0.059	0.065	0.064	0.071	0.104	0.127
Average	**0.033***	**0.036***	0.055	0.062	0.218	0.236	0.105	0.117	0.061	0.070

* The system did not complete all sequences or failed during the execution.

TABLE III
COMPARISON OF THE MEAN ATE (M) AND RMS ATE (M) IN THE SELECTED ICL-NUIM DATASET WITH SIMULATED LIGHTING CHANGES.

ICL-NUIM	ORB-SLAM3		SP-ORB-SLAM		HSO		The proposed method	
benchmark	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE	Mean ATE	RMS ATE
Syn1	0.335*	0.779*	0.053	0.062	0.128	0.173	**0.041**	**0.046**
Syn1-local	0.479*	0.545*	**0.040**	**0.046**	0.130	0.176	0.043	**0.046**
Syn1-global	0.128*	0.148*	0.059	0.066	0.129	0.174	**0.048**	**0.054**
Syn1-loc-glo	0.271*	0.330*	**0.028**	**0.035**	0.127	0.172	**0.028**	0.042
Syn1-flash	-	-	0.128*	0.141*	-	-	**0.102**	**0.112**
Syn1-average	0.303*	0.451*	0.062	0.070	0.129*	0.174*	**0.052**	**0.060**

* The system did not complete all sequences or failed during the execution.

TABLE IV
POWER CONSUMPTION COMPARISON

SLAM System	Memory (GB)	CPU (Power/mW)	GPU (Power/mW)	Total power consumption (Power/mW)
ORB-SLAM3	0.50	2504.15	0.00	3168.16
SP-ORB-SLAM	2.83	1342.80	3853.97	6893.92
HSO	0.24	2065.78	0.00	2629.76
Proposed method	2.74	1104.94	2112.66	4321.89

3) Evaluation on the Real-world Sequence: To further verify the effectiveness of the proposed method, the analysation of the performances of different VO/VSLAM methods on the real-world scenario is provided. Fig. 9 (a) and (b) show the top and front views of the quadrotor, which is used to capture image sequences, and the overall environment is shown in Fig. 9 (c). The quadrotor is based on the PX4 autopilot, and the Nvidia Jetson TX2 is used as the onboard computer. Some boxes and images captured in a pressure vessel are utilised to set up the environment. To make the environment more challenging, the lights are turned off. An OLIGHT Baton 3 is adopted to provide light for the environment. The Logitech C270 is used as the image input sensor. Considering the limited computing resources of the Nvidia Jetson TX2, the quadrotor is held in hand to record an image sequence. Some recorded images are shown in Fig. 9 (d), (e) and (f). The box (shown in Fig. 9 (d)) contains the rich texture information, and it is utilised to initialise the VO/VSLAM system. The image shown in Fig. 9 (e) contains motion blur and reflections. Fig. 9 (f) indicates the textureless environment. The Jetson TX2 is used to verify different VO/VSLAM methods. The ORB-SLAM3 cannot accomplish this sequence due to the insufficient feature points that can be extracted and matched from the scene in textureless regions. The quality of feature points used by HSO is affected by motion blur and reflection, making it difficult to maintain a consistent scale of the trajectory in this scenario. Moreover, the SP-ORB-SLAM cannot extract enough feature points in a short amount of time due to the high demand for computing resources, leading to the failure to initialize the system. The proposed network does not need as many computing resources required by the SP-ORB-SLAM. The extensive and reliable feature points could be extracted and described quickly, which makes the VO system can locate the UAV in this challenging environment. The trajectory can be seen in Fig.10. More information about how they behave differently can be found through https://youtu.be/cCg1NOMM14U.

D. UAV Flying Test

Finally, the flying test with the quadrotor and the environment introduced above is carried out. Through the conclusion obtained in the prevision, only the proposed methods can provide position information for the quadrotor in this challenging environment. Thus, only the proposed method is verified by the flying test. During the experiments, the scale information of the developed VO system is obtained from the PX4 autopilot, and a rectangle trajectory consisting of horizontal and vertical movement is utilised. Experimental results indicate that the UAV is capable of following the desired path in this low-illumination and textureless environment. Fig.11 displays the

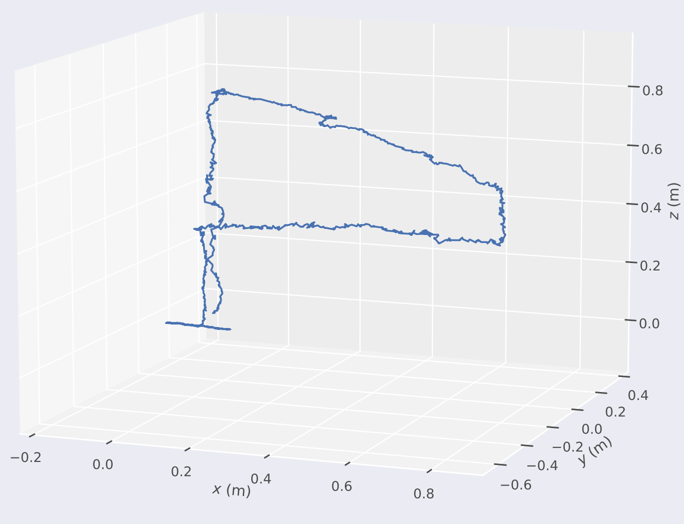
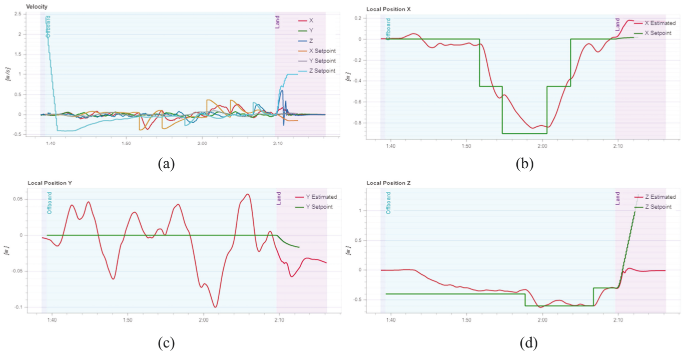


Fig. 10. Trajectory estimation by the proposed VO system



(a) (b) (c) (d)

Fig. 11. UAV trajectory estimation

velocity and position estimation results over time in the trajectory following process. In summary, the experimental results further attested that the proposed method is capable of locating the UAV in a dark and textureless environment. Additionally, the proposed network was trained on the MS-COCO 2014 dataset, which does not focus on indoor scenarios and does not include the image sequences used in the test experiments. Despite this, the performance of the developed CNN model is relatively good, as the VO system has demonstrated the outstanding performance in a variety of environments, especially in the environments used for the flying test. Therefore, the proposed network has achieved the good generalisation capability for unseen scenes. As a result, a robust and efficient VO system for the UAV onboard platform in challenging indoor environments has been achieved.

V. Conclusion and Future Work

In this work, a robust monocular VO system based on an efficient deep learning-based feature extraction network for UAVs has been presented. In the proposed feature extraction network, deformable convolution was utilised to enhance the feature extraction capabilities, and depthwise separable convolution was implemented to reduce parameters. Moreover, the deformable convolution has been improved by deploying the depthwise separable convolution to calculate the offsets. The proposed CNN model improves the feature extraction efficiency while preserving the ideal accuracy, especially for embedded platforms. The proposed network was incorporated into a VO system to improve its performance. Extensive experiments on public datasets, the real-world scenario and the flying test confirmed the efficiency and effectiveness of the whole VO system for UAV platforms.

Although the proposed system has been verified by UAV platforms, it still cannot support UAV moving fast in challenging environments. In further work, the efficiency of the feature extraction network can be further improved via descriptor quantisation or knowledge distillation. What is more, the estimated trajectory refinement and optimisation will be investigated, and the UAV equipped with the VO system will be tested in large areas.

Acknowledgments

Thanks are given to the supports from Strathclyde NZTC robotics research team, especially Dr Gordon Dobie and Dr Charles MacLeod.

References

[1] B. Yang, E. Yang, L. Yu, and A. Loeliger, “High-precision uwb-based localisation for uav in extremely confined environments,” *IEEE Sensors Journal*, vol. 22, no. 1, pp. 1020–1029, 2021.

[2] R. La Scalea, M. Rodrigues, D. P. M. Osorio, C. Lima, R. D. Souza, H. Alves, and K. C. Branco, “Opportunities for autonomous uav in harsh environments,” in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2019, pp. 227–232.

[3] Y. Liu, J. Dong, Y. Li, X. Gong, and J. Wang, “A uav-based aircraft surface defect inspection system via external constraints and deep learning,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[4] C. H. Quach, M. D. Phung, H. V. Le, and S. Perry, “Supslam: A robust visual inertial slam system using superpoint for unmanned aerial vehicles,” in *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*. IEEE, 2021, pp. 507–512.

[5] Z. Wang, S. Liu, G. Chen, and W. Dong, “Robust visual positioning of the uav for the under bridge inspection with a ground guided vehicle,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–10, 2021.

[6] X. Liu, X. Liu, W. Zhang, and Y. Yang, “Interacting multiple model uav navigation algorithm based on a robust cubature kalman filter,” *IEEE Access*, vol. 8, pp. 81 034–81 044, 2020.

[7] Y. J. Choi, I. N. A. Ramatryana, and S. Y. Shin, “Cellular communication-based autonomous uav navigation with obstacle avoidance for unknown indoor environments,” *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, pp. 344–352, 2021.

[8] J. Ho, S. Phang, and H. Mun, “2-d uav navigation solution with lidar sensor under gps-denied environment,” in *Journal of Physics: Conference Series*, vol. 2120, no. 1. IOP Publishing, 2021, p. 012026.

[9] J. Delmerico and D. Scaramuzza, “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2502–2509.

[10] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*. IEEE, 2007, pp. 225–234.

[11] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[14] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[15] X. Tao, L. Han, M. Paoletti, S. Roy, J. Plaza, J. M. Haut, and A. Plaza, “Multiple incremental kernel convolution for land cover classification of remotely sensed images,” in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 2021, pp. 2548–2551.

[16] F. Gao, Q. Liu, J. Sun, A. Hussain, and H. Zhou, “Integrated gans: Semi-supervised sar target recognition,” *IEEE Access*, vol. 7, pp. 113 999–114 013, 2019.

[17] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, “Aslfeat: Learning local features of accurate shape and localization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6589–6598.

[18] Y. Liu, Z. Shen, Z. Lin, S. Peng, H. Bao, and X. Zhou, “Gift: Learning transformation-invariant dense visual descriptors via group cnns,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[19] Z. Xu, J. Yu, C. Yu, H. Shen, Y. Wang, and H. Yang, “Cnn-based feature-point extraction for real-time visual slam on embedded fpga,” in *2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2020, pp. 33–37.

[20] X. Gu, Y. Wang, and T. Ma, “Dbld-slam: A deep-learning visual slam system based on deep binary local descriptor,” in *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2021, pp. 325–330.

[21] H. M. S. Bruno and E. L. Colombini, “Lift-slam: A deep-learning feature-based monocular visual slam method,” *Neurocomputing*, vol. 455, pp. 97–110, 2021.

[22] Z. Teed, L. Lipson, and J. Deng, “Deep patch visual odometry,” *arXiv preprint arXiv:2208.04726*, 2022.

[23] J. Tang, J. Folkesson, and P. Jensfelt, “Geometric correspondence network for camera motion estimation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1010–1017, 2018.

[24] G. Li, L. Yu, and S. Fei, “A deep-learning real-time visual slam system based on multi-task feature extraction network and self-supervised feature points,” *Measurement*, vol. 168, p. 108403, 2021.

[25] S. Jin, X. Dai, and Q. Meng, ““focusing on the right regions”-guided saliency prediction for visual slam,” *Expert Systems with Applications*, p. 119068, 2022.

[26] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, “Gcnv2: Efficient correspondence prediction for real-time slam,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3512, 2019.

[27] J. Shi *et al.*, “Good features to track,” in *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE, 1994, pp. 593–600.

[28] N. Brasch, A. Bozic, J. Lallemand, and F. Tombari, “Semantic monocular slam for highly dynamic environments,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 393–400.

[29] R. Kang, J. Shi, X. Li, Y. Liu, and X. Liu, “Df-slam: A deep-learning enhanced visual slam system based on deep local features,” *arXiv preprint arXiv:1901.07223*, 2019.

[30] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, “Learning local feature descriptors with triplets and shallow convolutional neural networks.” in *Bmvc*, vol. 1, no. 2, 2016, p. 3.

[31] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, “From coarse to fine: Robust hierarchical localization at large scale,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[32] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, “Dxslam: A robust and efficient visual slam system with deep features,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4958–4965.

[33] X. Han, Y. Tao, Z. Li, R. Cen, and F. Xue, “Superpointvo: A lightweight visual odometry based on cnn feature extraction,” in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*. IEEE, 2020, pp. 685–691.

[34] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[35] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, “Lift: Learned invariant feature transform,” in *European conference on computer vision*. Springer, 2016, pp. 467–483.

[36] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[38] L. Chun, L. Hongfei, Z. Qi, M. Zhenzhen, T. Sisi, and W. Yaping, “Binocular slam based on learning-based feature extraction,” in *Proceedings of the 2020 3rd International Conference on Robot Systems and Applications*, 2020, pp. 25–29.

[39] S. Li, S. Liu, Q. Zhao, and Q. Xia, “Quantized self-supervised local feature for real-time robot indirect vslam,” *IEEE/ASME Transactions on Mechatronics*, 2021.

[40] H. Huang, H. Ye, Y. Sun, and M. Liu, “Monocular visual odometry using learned repeatability and description,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8913–8919.

[41] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, “An optimal algorithm for approximate nearest neighbor searching fixed dimensions,” *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[42] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, “Deformable convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[43] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[44] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[45] A. F. Agarap, “Deep learning using rectified linear units (relu),” *arXiv preprint arXiv:1803.08375*, 2018.

[46] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and accurate deep network learning by exponential linear units (elus),” *arXiv preprint arXiv:1511.07289*, 2015.

[47] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[48] Y.-Y. Jau, R. Zhu, H. Su, and M. Chandraker, “Deep keypoint-based camera pose estimation with geometric constraints,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4950–4957.

[49] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[50] S. Park, T. Schöps, and M. Pollefeys, “Illumination change robustness in direct visual slam,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4523–4530.

[51] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[52] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[53] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[54] D. Luo, Y. Zhuang, and S. Wang, “Hybrid sparse monocular visual odometry with online photometric calibration,” *The International Journal of Robotics Research*, vol. 41, no. 11-12, pp. 993–1021, 2022.

Leijian Yu received the B.Eng. degree in electrical information engineering and the M.Sc. degree in information and communication engineering from the China University of Petroleum (East China), Qingdao, China, in 2015 and 2018, respectively. He has been working toward the Ph.D. degree in robotics and autonomous systems for asset visual inspection with the Department of Design, Manufacturing and Engineering Management, University of Strathclyde, Glasgow, U.K., since 2018.

His current research interests include machine learning with applications on unmanned aerial vehicles (UAV), UAV vision-based autonomous navigation, and image contrast enhancement.

Erfu Yang (Senior Member, IEEE) received the Ph.D. degree in robotics from the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., in 2008.

He is currently a Senior Lecturer with the Department of Design, Manufacturing and En-gineering Management (DMEM), University of Strathclyde, Glasgow, U.K. He has more than 160 publications in these areas, including more than 80 journal papers and 10 book chapters. His main research interests include robotics, autonomous systems, mechatronics, manufacturing automation, signal and image processing, computer vision and applications of machine learning and artificial intelligence, etc.

Dr. Yang was the recipient of more than 15 research grants as Principal investigator (PI) or coinvestigator (CI). He is the Fellow of the U.K. Higher Education Academy, Member of the U.K. Engineering Professors' Council, Senior Member of the IEEE Society of Robotics and Automation, IEEE Control Systems Society, Publicity Co-Chair of the IEEE U.K., and Ireland Industry Applications Chapter, Committee Member of the IET SCOTLAND Manufacturing Technical Network. He is currently an Associate Editor for the Cognitive Computation journal published by Springer.

Beiya Yang (Graduate Student Member, IEEE) received the B.Eng. degree in electronic information engineering from Northwestern Polytechnical University, Xi'an, China, in 2013 and the M.Sc degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2015. He has been working toward the Ph.D. degree in high-precision UAV positioning for autonomous inspection with the Department of Design, Manufacturing and Engineering Management (DMEM), University of Strathclyde, Glasgow, U.K., since 2019.

His current research interests include indoor localisation technology, unmanned aerial vehicles (UAV) localization, and wireless sensor networks.

Zixiang Fei (Member, IEEE) received the bachelor's degree from Liverpool John Moores University, Liverpool, U.K., in 2012, the master's degree from the University of York, York, U.K., in 2014, and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K., in 2020.

He is currently a Lecturer with Shanghai University, Shanghai, China. His major research interests include computer vision, machine learning, object recognition, and deep learning.

Cong Niu (Member, IEEE) received the B.Eng.(Hons.) degree in electronic engineering from the University of Central Lancashire, Preston, U.K., in 2014, the M.Sc. degree in embedded digital system from the University of Sussex, Brighton, U.K., in 2015, the Ph.D. degree in robotics and autonomous systems for agriculture applications from the Department of Design, Manufacturing and Engineering Management (DMEM), University of Strathclyde, Glasgow, U.K., in 2021.

He is currently a Research Associate with DMEM. His current research interests include field and indoor path planning, modeling and simulation, unmanned ground and aerial vehicles, and smart factory.