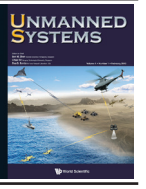
Unmanned Systems, Vol. 14, No. 2 (2026) 379–390
© World Scientific Publishing Company
DOI: 10.1142/S2301385026500068

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us



A Robust Visual-Inertial Navigation Method for Illumination-Challenging Scenes

Jiankang Qu*, Xu Lyu†,§, Ziyang Meng†, Pengfei Gu†, Jun Wang‡

*School of Instrument Science and Opto-Electronics Engineering, Beijing Information Science and Technology University, Beijing 100192, P. R. China

†Department of Precision Instrument, Tsinghua University, Beijing 100084, P. R. China

‡College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100089, P. R. China

Visual-inertial odometry (VIO) has been found to have great value in robot positioning and navigation. However, the existing VIO algorithms rely heavily on excellent lighting environments and the accuracy of robot positioning and navigation is degraded largely in illumination-challenging scenes. A robust visual-inertial navigation method is developed in this paper. We construct an effective low-light image enhancement model using a deep curve estimation network (DCE) and a lightweight convolutional neural network to recover the texture information of dark images. Meanwhile, a brightness consistency inference method based on the Kalman filter is proposed to cope with illumination variations in image sequences. Multiple sequences obtained from UrbanNav and M2DRG datasets are used to test the proposed algorithm. Furthermore, we also conduct a real-world experiment for the proposed algorithm. Both experimental results demonstrate that our algorithm outperforms other state-of-art algorithms. Compared to the baseline algorithm VINS-mono, the tracking time is improved from 22.0% to 68.2% and the localization accuracy is improved from 0.489 m to 0.258 m on the darkest sequences.

Keywords: Inertial navigation system; visual odometry; localization; Kalman filter; robustness.

US

1. Introduction

Autonomous navigation technology has become quite popular in recent years, driven by the limitations of the traditional GPS-based positioning method in blocked environments. The visual-inertial odometry (VIO) method, as a key technology of autonomous navigation, offers advantages such as low cost and small size. It combines the use of vision sensors and inertial measurement units (IMU) to provide complementary features. If the vision sensor fails during rapid movement, the IMU data can provide accurate localization results for a short period of time. The vision sensor is highly accurate when the system is moving at low speeds, which effectively reduces the cumulative error of the IMU over time. Due to these advantages, VIO has been rapidly developed and become one of the mainstream localization methods.

Currently, in the robotics community, VIO is primarily divided into two categories: the feature-based method and the direct method. The feature-based method detects feature points in the image that are invariant to rotation and scale, and estimates the camera's motion by the matched feature point pairs. The direct method, on the other hand, detects pixel points with discrimination in the image and directly estimates the camera's motion through pixel-level optical flow tracking. The most representative systems of these two methods are ORB-SLAM [1] and VINS-Mono [2], respectively.

However, the robustness of the VIO system relies on high-quality input images and is sensitive to illumination changes. Even the most advanced algorithms mentioned above must operate in an environment with the appropriate

Received 11 July 2024; Accepted 9 January 2025; Published 24 February 2025. This paper was recommended for publication in its revised form by editorial board member, Jinwen Hu.
Email Address: §lvclay@163.com

illumination and rich texture information. Otherwise, they may suffer from problems such as position drift or tracking failures. Such strict conditions limit the application of VIO in extreme illumination scenes, such as outdoor environments with few street lights, dark underground parking lots, and urban streets with sharp illumination changes. Developing a visual odometry that can function effectively in extreme illumination conditions is a challenging task. To enable localization tasks in such scenarios, it is necessary to develop a feature-matching method that is robust to illumination. To address the aforementioned issues, researchers have proposed numerous solutions, which can be generally classified into two categories: developing learning-based descriptors or enhancing image quality.

Schemes of developing learning-based descriptors do not change the image, but instead tend to obtain reliable feature points directly from low-light images through a learning-based approach. The early works attempt to use learned descriptors by converting pre-cropped image chunks into a fixed-length descriptor vector for matching via deep learning methods, e.g. L2Net'17 [3] and HardNet'17 [4]. Then, SuperPoint [5] and SuperGlue [6], extract both feature points and descriptors on the full image instead of operating on image blocks. Unlike the "extract-then-match" process, LoFTR [7] proposes an algorithm that builds matches directly, thus avoiding false matches in weakly textured regions due to insufficient repetition of feature point extraction. Some recent works, Lightglue [8] and Featurebooster [9], focus on accelerating the network of descriptors. Gluestick [10], considers line features, and S-Trek [11], considers iso-variation of rotations in the image plane. Efficient-LoFTR [12] uses a novel method for efficiently producing semi-dense matches across images, greatly improving the computational efficiency and accuracy of the previous detector-free matcher LoFTR. These works of developing learning-based descriptors consider a number of issues encountered in visual feature matching, including computational efficiency, weak texture, large-viewpoint change, and matching reliability, having accelerated the applications of VIO in extreme environments.

On the other hand, schemes of enhancing image quality transform low-quality images in illumination-challenging scenes into normal images. Darkness is seen as a cause of low-quality images in illumination challenging because it results in the loss of image texture. Traditional image enhancement methods are generally based on histograms or Retinex theory [13]. However, neither approach can achieve optimal results. The reason is that the histogram-based scheme is relatively simple, and the Retinex-based scheme cannot directly provide a precise measurement for the illumination component. Deep learning has been found useful for the task of image enhancement. RetinexNet [14] adopts a supervised learning approach, where paired images are fed into the network, and the network is trained by minimizing the reconstruction error, the reflection component consistency error, and the illumination component smoothing error. KinD [15] is based on RetinexNet, and obtains a more realistic enhanced image by compensating the degradation. RUAS [16] considers the intrinsic structure of the low-light image, estimating the enhancement and degradation components simultaneously, and performs lightweighting on the model. In addition, the approach based on generative adversarial network (GAN), EnlightenGan [17], is also used for image enhancement. This model does not require strictly matched image pairs, so it can be applied to a wider range of scenarios. Zero-reference depth curve estimation model is proposed by Li *et al.* [18]. The key technique of this algorithm is to train a lightweight deep network deep curve estimation network (DCE) to construct pixel-level and higher-order curves, after which the input image can be dynamically adapted. Jia *et al.* [19] propose a cycle-consistent GAN to address the problems of contrast distortion, loss of detailed information, and a significant amount of noise in night road scenes. DarkSLAM [20] compares the feature-tracking effects of the traditional gamma correction, histogram equalization scheme and the learning-based EnlightenGan scheme in dark scenes, and the results show that the EnlightenGan scheme is superior to others in the paper. TwilightSLAM [21] compares the effect of four deep learning-based image enhancement modules on visual SLAM, it also agrees that EnlightenGan is a good solution in dark scenes.

Despite the excellent works mentioned above, applying VIO in extreme illumination conditions remains challenging. For the schemes of developing learning-based descriptors, it is still difficult to apply the model to realistic localization tasks due to its high training cost, large size, texture feature similarity in the dark, and prevalence of mismatches. For the schemes of enhancing image quality, its advantage is that the original VIO framework does not need to be changed and the image enhancing method can be added modularly, but this undoubtedly increases the overall running time of VIO. In addition, both the amplified image noise after enhancing low-light images and the potential brightness changes between consecutive image frames also need to be addressed. For the former one, several works, such as RUAS [16], try to integrate a noise estimator into the image enhancement network for simultaneous enhancement and denoising, but the image noise cannot be removed entirely. For the latter one, there are several studies [22, 23] considering intrinsic factors of imaging sensors and lenses affecting brightness changes, but the effects of the lens and image sensors on the image brightness need to be additionally modeled. Another solution is to extract stable image features such as lines and edges in changing brightness conditions [24–26]. However, line-based methods have low

positioning accuracy, while there are few matching methods for edges.

In this paper, we follow the schemes for enhancing image quality. First, we adopt a simple effective two-step algorithm to remove image degradation noise and get high-quality images. We use the DCE [27] to enhance the dark image, and a lightweight convolutional neural network [28] to denoise the enhanced image. Second, we propose a method inspired by Retinex theory to preserve brightness consistency. We decompose the image into reflection and brightness components, infer the brightness of the next frame from the previous frames' brightness components, and optimize the observed brightness with the Kalman filter algorithm. We also identify and adjust anomalous brightness frames to ensure they fall within a normal range. Third, we developed a robust VIO for illumination-challenging scenes using a constructed low-light image enhancement model and proposed a brightness consistency inference method. The proposed algorithm was evaluated on two publicly available datasets, UrbanNav [29] and M2DGR [30]. Furthermore, a real-world experiment is also conducted to verify the proposed algorithm. Both results indicate that the proposed algorithm can run in real-time at 15 fps on an RGB dataset with an image size of 640 $\times$ 480.

The contributions of this paper are summarized as follows. First, we construct an effective low-light image enhancement model using a DCE and a lightweight convolutional neural network. Second, we propose a brightness consistency inference method based on the Kalman filter, which can smooth out brightness changes in image sequences. Third, we verified the feasibility of the proposed method in illuminating challenging scenes through experiments on public datasets and real-world experiments.

The remaining parts of this paper are organized as follows. Section 2 provides a detailed description of the proposed VIO framework, DCE, lightweight convolutional neural network, and brightness consistency inference method. Section 3 provides the experimental results and their analysis. Section 4 provides a summary of the work presented in this paper.

2. Methodology

2.1. *Overview*

Figure 1 shows the proposed visual inertial navigation method. First, the camera captures a dark image which is then processed by the proposed low-light image

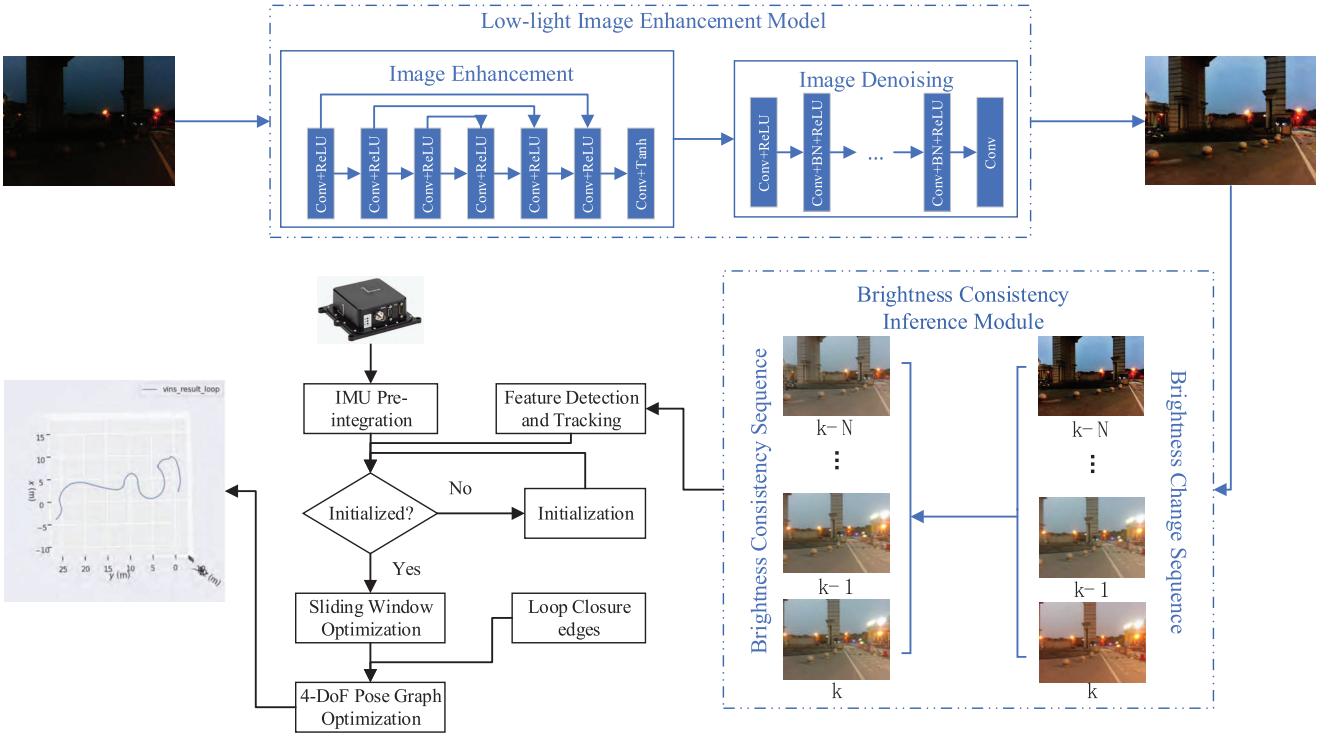


Fig. 1. Illustration of the full pipeline for the proposed visual-inertial navigation method. The blue figure is the contribution of the proposed algorithm compared with VINS-Mono [2], including the low-light enhancement model detailed in Sec. 2.2 and the brightness consistency inference module detailed in Sec. 2.3. The black figure is the pose estimator introduced from VINS-Mono.

enhancement model to restore texture information and denoise. This model includes an image enhancement network and an image denoising network, and the full implementation can be seen in Sec. 2.2. Then the proposed brightness consistency inference algorithm is performed to maintain consistent brightness across all processed images. The proposed algorithm uses a Kalman filter to eliminate sharp brightness changes between images, and its complete implementation can be seen in Sec. 2.3. Finally, the processed high-quality image sequence is input into the pose estimator to obtain the positioning result. The pose estimator consists of four parts: feature tracking, VIO initialization, sliding-window-based optimization for short-period localization and pose graph optimization to produce a globally consistent trajectory estimation.

Feature Tracking: We use the Lucas–Kanade method for feature tracking in this paper, which requires visual information to meet constant grayscale assumptions. The image processed by the low-light image enhancement model detailed in Sec. 2.2 and the brightness consistency module detailed in Sec. 2.3, defined as $\hat{X}$, already satisfies the constant grayscale assumption. This assumption implies that the gray value of the point remains constant after time Δt.

$$V(x+\Delta x, y+\Delta y, t+\Delta t) = V(x,y,t). \tag{1}$$

Here, (x,y) is the pixel coordinate, t is the time, and $V(x,y,t)$ is the gray value of the point in $\hat{X}$. Expanding the left-hand side of Eq. (1) using Taylor series and simplifying, Eq. (1) can be rewritten as

$$V_x u + V_y v + V_t = 0. \tag{2}$$

Here, $V_x = \frac{\partial V}{\partial x}$, $V_y = \frac{\partial V}{\partial y}$, and $V_t = \frac{\partial V}{\partial t}$ are the partial derivatives of the gray value with respect to x, y, and t, respectively. The variables u and v are the movement speed of the point on the horizontal and vertical coordinate axes. Given a static region containing n points, the feature tracking is achieved by using the least squares method to find the optimal movement speed (u^*, v^*).

VIO Initialization: One of the drawbacks of the monocular vision scheme is that the scale is unobservable. To ensure the localization accuracy of the system, during initialization, the position estimator fuses the vision information and the IMU pre-integration information to calibrate the gyroscope bias (bias), velocity (v), gravity vector (g), and scale (s).

Epipolar geometric is used to solve the rotation and translation between two frames. Then, triangulation is used to obtain the 3D coordinates of observed feature points in the two frames. Based on these 3D feature points and the 2D feature points observed in other frames, the pose (translation, rotation), and 3D feature points of other frames in the sliding window are solved using Perspective-n-Point (PnP). The minimization error model is created by differentiating the rotation information of all frames in the sliding window from the IMU rotation information. The gyroscope bias is obtained using the least squares solution. Additionally, the system velocity, gravitational acceleration, and scale are calibrated by using the translation and velocity information of the vision and IMU as constraints.

Sliding-window-based Optimization: We use sliding window optimization for short-period localization. The localization result can be obtained by minimizing the following cost function:

$$\min_{\chi} \left\{ \sum_{l\in\upsilon} \left\| e_{\upsilon}(m^{b_l}_{b_{l+1}}, \chi) \right\|^2_{C^{bl}_{bl+1}} + \sum_{(i,j)\in \mathrm{B}} h(\left\| e_{\mathrm{B}}(m^{f_i}_i, \chi) \right\|^2_{C^{fi}_i}) + \left\| e_p - H_p \chi \right\|^2 \right\}. \tag{3}$$

Here, the first term denotes the IMU pre-integration constraint, χ denotes the optimization variable which contains the position, velocity, IMU bias, IMU-to-camera external reference, and the inverse depth of the 3D feature points for each frame within the sliding window. υ denotes the set of all IMU measurements within the current sliding window, $m^{b_l}_{b_{l+1}}$ denotes the IMU measurements between two consecutive frames b_l and b_{l+1}, e_{υ} denotes the IMU pre-integration measurement error, and $C^{b_l}_{b_{l+1}}$ denotes the IMU pre-integration covariance matrix of the integration noise. The second term denotes the visual reprojection constraint. B denotes the set of feature points that have been observed at least twice in the current sliding window, $m^{f_i}_i$ denotes the observation value of the ith feature point in the jth frame. e_{B} denotes the visual reprojection error, $C^{f_i}_i$ denotes the covariance matrix of the visual observation noise, and $h(\cdot)$ denotes the Huber norm. The third term denotes the priori constraint. e_p and H_p, denote the constraint relationship between the old frame and other frames in the sliding window.

Pose Graph Optimization: We use four-DOF pose graph optimization to produce a globally consistent trajectory estimation. The whole graph of sequential edges and loop closure edges are optimized by minimizing the following cost function:

$$\min_{P,\Psi} \left\{ \sum_{(i,j)\in S} \left\| r_{i,j} \right\|^2 + \sum_{(i,j)\in L} h(\left\| r_{i,j} \right\|^2) \right\}. \tag{4}$$

Here, $r_{i,j}$ denotes the residual of the edge between frames i and j, P denotes the relative position, Ψ denotes the yaw angle, S denotes the set of all sequential edges, L denotes the set of all loop-closure edges, and $h(\cdot)$ denotes the Huber norm.

2.2. *Low-light image enhancement model*

Image Enhancement Network: Deep curve estimation has been shown to be effective in enhancing image quality. This paper uses the DCE-Net [27] network model to recover texture information from dark images. The model comprises six hidden layers and an output layer, making it lightweight and suitable for deployment. In the first six convolutional layers, each convolutional layer consists of 32 convolutional kernels of size 3×3 and stride 1 followed by the ReLU activation function. The last convolutional layer consists of 24 convolutional kernels of size 3×3 and stride 1 followed by the Tanh activation function, which produces 24 curve parameter maps for eight iterations, where each iteration generates three curve parameter maps for the three channels. The DCE-Net network model was trained using a training dataset consisting of 2422 unpaired bright/dark images and paired bright/dark images obtained from the SICE dataset [31]. To obtain the mapping relationship between the input image and its best-fitting curve parameter, four loss functions were employed: spatial consistency loss, exposure control loss, color constancy loss, and illumination smoothness loss. The total loss function is given as

$$L_{\text{total}} = W_1 L_{\text{spa}} + W_2 L_{\text{exp}} + W_3 L_{\text{col}} + W_4 L_{\text{tva}}, \quad (5)$$

where W_i $(i = 1, 2, 3, 4)$ is the weight, L_{spa} is spatial consistency loss, L_{exp} is exposure control loss, L_{col} is color consistency loss, and L_{tva} is illumination smoothness loss. In particular, L_{spa} is defined as

$$L_{\text{spa}} = \frac{1}{K} \sum_{i=1}^{K} \sum_{j \in \Omega(i)} (|Y_i - Y_j| - |I_i - I_j|)^2, \quad (6)$$

where K is the number of local regions, $\Omega(i)$ is the four adjacent regions (top, bottom, left, and right) centered on region i. We denote Y and I as the average local region intensities in the enhanced and input images, respectively. Additionally, L_{exp} is defined as

$$L_{\text{exp}} = \frac{1}{M} \sum_{k=1}^{M} |Y_k - E|, \quad (7)$$

where E is the gray level in the RGB color space and is set to 0.6 based on prior experience, M represents the number of nonoverlapping local regions, and the average intensity value of each local area is denoted as Y. Additionally, L_{col} is defined as

$$L_{\text{col}} = \sum_{\forall (p,q) \in x} (J_p - J_q)^2; \quad x \in \{(R,G),(R,B),(G,B)\}, \quad (8)$$

where J_p denotes the average intensity value of p channel in the enhanced image, a pair of channels is represented as (p, q). Additionally, L_{tva} is defined as

$$L_{\text{tva}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{c \in \xi} (|\Delta_x A_n^c| + |\Delta_y A_n^c|)^2; \quad \xi \in \{R, G, B\}, \quad (9)$$

where A is curve parameter map, N is the number of iterations, and the horizontal and vertical gradient operations are represented as Δx and Δy, respectively.

Image Denoising Network: The images captured under low light conditions may contain various types of noise, including shot noise, read noise, and others. During the texture recovery process, although the texture information is improved, the noise is also amplified, making it challenging to extract subsequent feature points. To address this issue, we utilize FFD-Net [28] to denoise the enhanced image output by the image enhancement network, as it has demonstrated efficient image-denoising capabilities and rapid response times. The model comprises 12 layers. Each layer is composed of three types of operations: Convolution (Conv), Rectified Linear Units (ReLU), and Batch Normalization (BN). Specifically, "Conv+ReLU" is adopted for the first convolution layer, "Conv+BN+ReLU" for the middle layers, and "Conv" for the last convolution layer.

Most of the model-based denoising methods define the image-denoising task as

$$\hat{x} = \arg\min_x \frac{1}{2l^2} \|y - x\|^2 + m \times n(x), \quad (10)$$

where $\frac{1}{2l^2} \|y - x\|^2$ and $n(x)$ denote the data fidelity and regularization terms, respectively. The balance between these terms is controlled by m.

Equation (10) defines an implicit function for x. Set $\theta = n(x)$. Meanwhile, since m can be absorbed into l, Eq. (10) can be rewritten as

$$\hat{x} = F(y, M; \theta), \quad (11)$$

where F represents the implicit function and the variables are enclosed in bracket. The true-value image and noise image pair is created by adding Gaussian white noise to high-quality images. Those image pairs serve as the training dataset for the network. The loss function of the network is defined as

$$L_{\text{loss}} = \frac{1}{2N} \sum_{i=1}^{N} \|F(y_i, M_i; \theta) - x_i\|^2, \quad (12)$$

where N is the number of subimages, y_i is the noise subimage, and x_i is the original subimage. Finally, the ADAM [32] method is used to minimize the loss function L to optimize the model parameters.

2.3. *Brightness consistency inference method*

Consistent brightness is essential to ensure accurate matching and tracking of image features. However, due to

insufficient lighting in illumination-challenging scenes, it is impossible to guarantee consistent brightness for each image output by the low-light image enhancement model detailed in Sec. 2.2. Therefore, we have developed a brightness consistency inference method to maintain global brightness consistency between images.

According to Retinex theory [13], the brightness of an image can be estimated by

$$\gamma(X) = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij} \times g(i,j), \tag{13}$$

where X denotes the current image, M and N denote the rows and columns of the current image, x denotes the pixel point in X, $g(i,j)$ denotes the Gaussian surround function, and $\gamma(X)$ denotes the image brightness matrix of the current image, being the same size as X.

After the image processing by the low-light image enhancement model detailed in Sec. 2.2, the texture in low-light regions of the image has been restored and we obtained high-quality images. Based on this, we have to make the assumption that the variability of a real-world object across images is influenced only by illumination intensity. This assumption indicates that the pixel gradients in the common view region between images should be consistent. According to Eq. (13), it is clear that the image brightness matrix will also preserve this pixel gradient invariance. Then we rewrite the image brightness matrix $\gamma(X)$ as

$$\gamma(X) = \beta^a + \gamma'(X), \tag{14}$$

where β^a denotes the brightness mean matrix, a denotes the average brightness value of the image, and $\gamma'(X)$ denotes the zero-mean brightness gradient matrix. Based on the above derivation, it is known that $\gamma'(X)$ is the same and β^a is the main factor causing the different image brightness in the same view. However, the image brightness matrix should consist of the common-view component and the non-common-view component in the real scene. Thus, the image brightness matrix $\gamma(X)$ should be rewritten as

$$\gamma(X) = \gamma_s(X) + \gamma_d(X) = \beta_s^{a_s} + \gamma_s'(X) + \beta_d^{a_d} + \gamma_d'(X), \tag{15}$$

where s denotes the common-view component, and d denotes the noncommon-view component. The sum of their sizes equals the size of the image, i.e. $s + d = M \times N$. Here, $\beta_s^{a_s}$, $\gamma_s'(X)$, $\beta_d^{a_d}$ and $\gamma_d'(X)$ are all $M \times N$ matrices, there are non-zero values in the subscript region represented by $\{s, d\}$ and the rest is 0. Since the influence of the noncommon-view component, $\beta_s^{a_s}$ cannot be directly obtained. Instead, we augment $\beta_s^{a_s}$ and move the other terms to the other side of Eq. (15). It can be rewritten as

$$\beta_{M\times N}^{a_s} = \gamma(X) - \gamma_s'(X) - \gamma_d'(X) + \beta_d^{a_s} - \beta_d^{a_d}. \tag{16}$$

Here, the first three terms on the right-hand side equal $\beta_{M\times N}^{a}$, and the remaining two terms represent the average brightness value residual between the common-view component and the noncommon-view component. This residual is a minimum value that fluctuates around zero when the noncommon-view region is small.

Since all elements in the brightness mean matrix are the same, we can use the average brightness value a_s and a to represent $\beta_{M\times N}^{a_s}$ and $\beta_{M\times N}^{a}$. Then, the residual $\beta_d^{a_s} - \beta_d^{a_d}$ can be modeled as a zero-mean Gaussian noise $G(0, \sigma_1^2)$. If there is no illumination changes in the previous N frames of the current image, we can use the first-order linear regression function to infer the average brightness value of the current image. Then the observation equation and prediction equation for the average brightness value of the current image can be defined as

$$\begin{cases} a_s^k = a^k + G(0, \sigma_1^2), \\ \bar{a}_s^k = f(w, b; k) + G(0, \sigma_2^2), \end{cases} \tag{17}$$

where k denotes the time of the current image, $G(\cdot)$ denotes Gaussian noise, σ denotes the standard deviation of Gaussian noise, $\bar{a}_s^k$ denotes the predicted value of a_s at time k, f denotes the first-order linear regression function, and the parameters of this function are w and b. Then the Kalman filter is introduced to optimize the average brightness value, and its Kalman filter gain, estimated value and estimated error can be defined as

$$\begin{cases} K = \dfrac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}, \\ \hat{a}_s^k = \bar{a}_s^k + K \times (a_s^k - \bar{a}_s^k), \\ \hat{\sigma}_2^2 = (1 - K) \times \sigma_2^2, \end{cases} \tag{18}$$

where K denotes the Kalman filter gain, and $\hat{a}_s^k$, $\hat{\sigma}_2^2$ denotes the estimated value and estimated error of a_s at time k. In addition, we hope to perform Kalman filter optimization in different image channels separately, so the complete

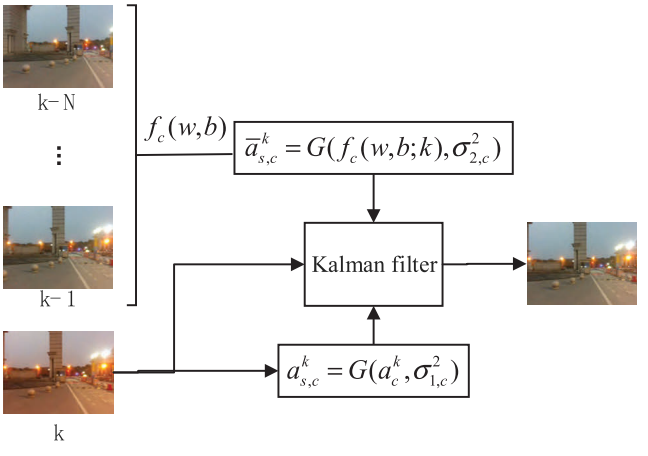


Fig. 2. Illustration of brightness correction.

brightness correction process can be defined as

$$\hat{X}^k = \sum_{c\in\{R,G,B\}} T(\hat{a}^k_{s,c}, \gamma'(X^k_c)), \tag{19}$$

where $\hat{X}$ denotes the image processed by the proposed brightness consistency inference method, c denotes the image channels, and $T(\cdot)$ denotes the conversion function from the image brightness matrix to the image. The whole brightness correction process is shown in Fig. 2.

3. Experiments and Results

3.1. *Setup*

Fifteen dark sequences from UrbanNav [29] and M2DGR [30] datasets, including UrbanNav-HK-Data20190428, Gate_01-02, Darkroom_01-06, Street_01, Street_03-06 and Street_08. Two dark sequences were collected from the real world. Three normal illumination sequences from the M2DGR dataset, including Room_01-03. All the data sequences mentioned above are used to comprehensively test the proposed algorithm in this paper. Note that for simplicity, we also use the first letter of the scene corresponding to the sequence plus the serial number to represent the original sequence name mentioned above. To illustrate, "UN" denotes "UrbanNav-HK-Data20190428," "G01" denotes "Gate_01," "D01" denotes "Darkroom_01", and "S01" denotes "Street_01." Besides, the gate sequence of the M2DRG dataset is not completely dark. Therefore, we utilize the method in [33] to convert it into a fully dark scene, and we add a segment with uniformly varying illumination and another segment with randomly varying illumination. The M2DGR dataset often experiences severe image dropout, which can cause the algorithm to fail due to the loss of data. Therefore, only the data segments before such a case happens are used in the experiments for evaluations. All experiments in this paper were conducted on a computer with Ubuntu 18.04 as the operating system, an Intel i7 9th generation CPU as the computing unit, and a consumer-grade, low-priced graphics card, GeForce RTX 3050 for the GPU.

3.2. *Comparison algorithms*

In this paper, we classify all eight comparison algorithms used into three categories. The first category is the general visual algorithm, including VINS-Mono [2], OpenVINS [34] and R2VIO [35]. The second category is the visual algorithm for enhancing image quality, including DarkSLAM [20], VINS-Mono-RUAS and VINS-Mono-ZD. The third category is the visual algorithm for developing learning-based descriptors, including SP_SLAM [36] and AirVO [37]. For a fair comparison, we modified the algorithms belonging to the second and third categories, and the corresponding modifications are described in detail as follows.

For the second category, we construct the image preprocessing modules based on various image enhancement models, and we append these image preprocessing modules to the image input of VINS-Mono to create the modified algorithms. The position of the added image preprocessing module in the entire algorithm pipeline is the same as the blue part in Fig. 1. According to the above ideas, we retain the image enhancement scheme EnlightenGan [17] used by DarkSLAM and obtain the modified DarkSLAM. Additionally, two other image enhancement modules, RUAS [16] and Zero-DCE [19], use the same modifications above to get VINS-Mono-RUAS and VINS-Mono-ZD. Figure 3 displays some results from datasets, in comparison with the above three schemes called DarkSLAM, VINS-Mono-RUAS and

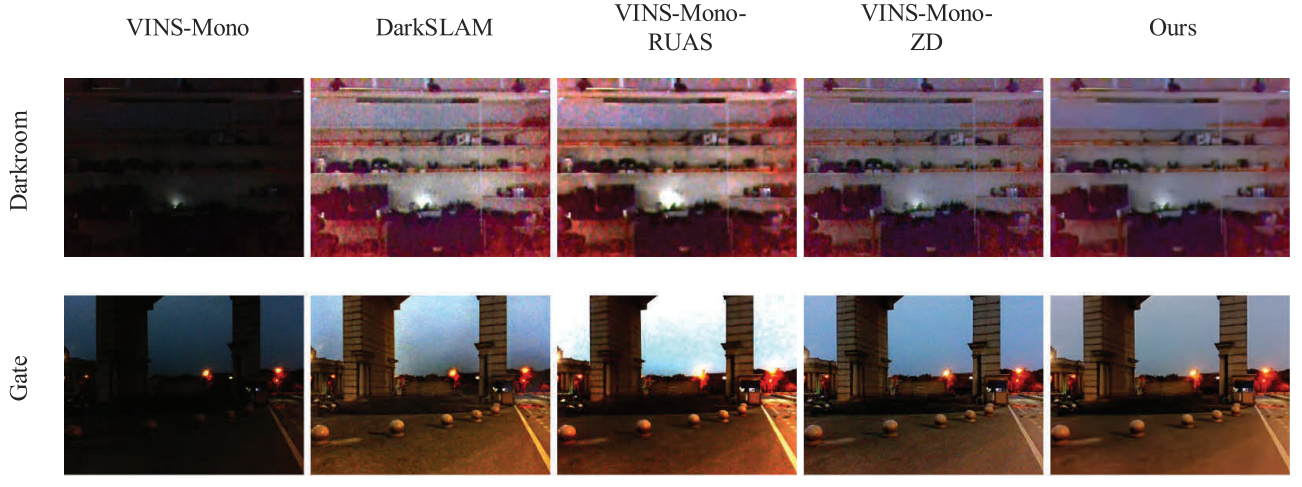
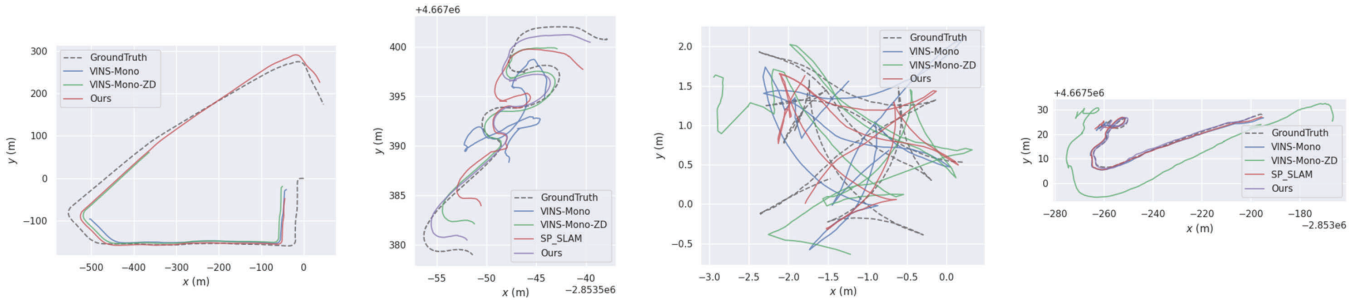


Fig. 3. Comparison of the proposed algorithm and state-of-the-art methods.



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

Fig. 4. Trajectories of various methods using the sequences in the M2DGR dataset and UrbanNav dataset. (a), (b), (c), and (d) represent the results of positioning trajectory on UrbanNav-HK-Data20190428, Gate_01, Darkroom_01 and Street_01, respectively.

VINS-Mono-ZD. Figure 3 shows that DarkSLAM, VINS-Mono-RUAS and VINS-Mono-ZD amplify image noise when processing dark images. In contrast, the proposed method removes image noise and outputs high-quality images.

For the third category, the modified SP_SLAM and AirVO are, in essence, integrations of the beneficial components of the original SP_SLAM and AirVO for illumination-challenging scenes into the VINS-Mono framework. For the modified SP_SLAM: We have replaced VINS-Mono's original Shi-Tomasi corner detector with the more reliable SuperPoint model used by the initial SP_SLAM. For the modified AirVO: We have replaced the front-end visual feature tracking scheme of VINS-Mono with the combination of SuperPoint and SuperGlue from the initial AirVO. In both modifications, we strictly controlled the maximum number of features to be 150, with a minimum feature distance of 30 pixels, to ensure that the parameter settings were consistent across all compared algorithms.

Figure 4 shows the trajectories of various methods using the sequences in the M2DGR dataset and UrbanNav dataset. (a), (b), (c), and (d) represent the results of positioning trajectory on UrbanNav-HK-Data20190428, Gate_01, Darkroom_01 and Street_01, respectively. To avoid the difficulty of distinguishing trajectories due to too many algorithms to compare, we only compare the best algorithm in each category mentioned above during testing. In (a), the SP_SLAM trajectory does not appear, indicating that the positioning has failed. The complete test results can be viewed from Tables 1–3. Note that the modified algorithm cannot directly represent the performance of the original algorithm.

3.3. *Experimental evaluation of localization accuracy*

We next quantitatively evaluate the proposed algorithm. The metric of absolute pose error (APE), root mean squared error (RMSE), is used to evaluate the localization accuracy of the proposed algorithm. The RMSE metric represents the average error between the position estimated by the algorithm and the true position. The smaller the RMSE metric, the higher the localization accuracy. It is important to note that the RMSE value gradually increases as the execution time increases. In this paper, we only compare the RMSE value of the data segment that can run stably in each scheme.

Table 1 presents the experimental results, and the best result and the second-best result of the sequence are highlighted in red and blue, respectively. It is noted that the

Table 1. Test result of RMSE(m) on UrbanNav dataset and M2DGR dataset. Red is the best result, and blue is the second-best result. The symbol "—" indicates that the algorithm diverges. Sequences where all algorithms failed are not shown. Ours_BCI denotes that brightness consistency inference method is performed.

	UN	G01	G02	D01	D02	D03	S01	S03	S04	S05	S06	S08
Vins-Mono	32.985	7.601	16.440	0.489	0.504	0.341	2.549	—	12.264	2.412	7.654	—
OpenVINS	—	—	—	2.323	4.171	2.809	—	—	—	24.221	29.862	—
R2VIO	—	7.383	29.933	2.136	—	3.304	28.346	12.883	—	21.460	11.148	—
DarkSLAM	34.135	2.416	5.848	1.507	0.595	1.119	3.825	—	23.263	5.998	0.794	—
VM-RUAS	33.032	3.915	7.359	0.314	0.575	0.446	8.919	4.088	18.086	21.926	1.158	11.359
VM-ZD	28.795	2.281	9.552	0.484	0.524	0.383	21.438	3.143	57.536	1.735	0.346	—
Ours	31.980	2.198	4.676	0.341	0.407	0.385	1.349	2.962	19.118	1.103	0.647	5.737
Ours_BCI	25.694	1.457	3.725	0.258	0.407	0.371	1.349	3.029	17.756	1.150	0.583	4.386

Table 2. Test result of CR(%) on UrbanNav dataset and M2DGR dataset. Red is the best result, and blue is the second-best result. VM-RUAS denotes VINS-Mono-RUAS. VM-ZD denotes VINS-Mono-ZD. Ours_BCI denotes that the brightness consistency inference method is performed.

	UN	G01	G02	D01	D02	D03	S01	S03	S04	S05	S06	S08
Vins-Mono	23.7%	0	3.3%	22.0%	54.9%	62.0%	17.8%	0	20.8%	20.8%	27.4%	0
OpenVINS	0	0	0	2.7%	0.5%	0	0	0	0	0	1.7%	0
R2VIO	0	1.6%	0	3.1%	0	4.6%	0.7%	3.9%	0	1.5%	22.0%	0
DarkSLAM	22.8%	37.5%	24.6%	0	33.3%	8.6%	12.7%	0	22.4%	5.0%	66.3%	0
VM-RUAS	24.0%	6.6%	26.0%	58.8%	41.7%	51.6%	0	5.7%	26.8%	50.0%	65.2%	4.7%
VM-ZD	25.8%	48.7%	33.8%	44.7%	46.7%	61.9%	0	4.6%	2.6%	69.8%	70.9%	0
Ours	47.7%	50.2%	23.2%	49.4%	58.0%	52.7%	73.7%	6.3%	21.0%	74.3%	75.0%	5.2%
Ours_BCI	60.0%	54.3%	44.2%	68.2%	58.0%	52.5%	73.7%	5.1%	28.9%	74.5%	73.2%	17.9%

Table 3. Test results of CR(%), RMSE(m), TFF(pairs) and COST(ms) on M2DGR dataset. Red is the best result, and blue is the second-best result. The symbol "—" indicates that the algorithm diverges. Ours_BCI denotes that the brightness consistency inference method is performed.

	Gate_01				Darkroom_01				Street_01			
	CR↑	RMSE↓	TFF↑	COST↓	CR↑	RMSE↓	TFF↑	COST↓	CR↑	RMSE↓	TFF↑	COST↓
SP_SLAM	13.2%	4.357	46	26.00	55.3%	0.384	53	21.56	5.6%	3.120	44	21.10
AirVO	9.6%	2.545	42	110.30	18.9%	0.510	49	68.10	0	—	36	58.40
Ours	50.2%	2.198	84	34.74	49.4%	0.341	95	37.12	73.7%	1.349	46	35.97
Ours_BCI	54.3%	1.457	84	40.57	68.2%	0.258	95	38.77	73.7%	1.349	46	37.07

original VINS-Mono performs poorly in illumination-challenging scenes, but it is still the best among the algorithms of the first category. On the other hand, the results indicate that VINS-Mono-ZD performs the best in the second category, while the DarkSLAM scheme performs poorly, and is even inferior to VINS-Mono on some sequences. The proposed algorithm achieves the best performance among all algorithms, particularly for the sequences with illumination changes, such as UrbanNav, Gate_01, and Darkroom_01.

3.4. *Experimental evaluation of algorithm robustness*

The second experiment involves a comparative analysis of algorithm robustness. Algorithm robustness is not only related to localization accuracy, but also closely related to algorithm execution time, especially in illumination-challenging scenes.

The metric, correct rate (CR) [33], is used to evaluate the robustness of the proposed algorithm. The CR metric represents the proportion of execution time for the SLAM algorithm when the absolute translation error (ATE) values and the absolute orientation error (AOE) values do not exceed the specified threshold. It can be calculated using

$$\mathrm{CR}^{\varepsilon,\varphi} = \frac{\sum_{k=0}^{N-1}(t_{k+1}-t_k)\times c^{\varepsilon,\varphi}(p_k)}{t_N - t_0}, \tag{20}$$

where

$$c^{\varepsilon,\varphi}(p_k) = \begin{cases} 1, & \text{if } \mathrm{ATE}(p_k) \le \varepsilon \text{ and } \mathrm{AOE}(p_k) \le \varphi \\ 0 \end{cases}, \tag{21}$$

N denotes the number of nodes in the positioning result, t denotes time, and ε, ϕ denotes the specified threshold of ATE and AOE. The larger the CR metric, the better the algorithm robustness.

When calculating the CR metric, we set the ATE threshold to 1% of the ground truth of the trajectory length and the AOE threshold to 180. Table 2 presents the experimental results, and the best result and the second-best result of the sequence are highlighted in red and blue, respectively. It is noted that high localization accuracy does not necessarily mean that it is more robust, such as the results on street_05. In general, the proposed algorithm still has the best performance.

3.5. *Comparison with learning-based descriptor schemes*

The third experiment involves a comparative analysis with different learning-based algorithms. It should be noted that both SuperPoint and SuperGlue are lightweight and can be executed in the proposed framework. In this experiment, in addition to utilizing the RMSE metric and CR metric, we also add tracked features per frame (TFF) metric and cost metric, which represent the average number of the tracked feature points between frames and the average time to process each frame, respectively.

During testing, we strictly controlled the maximum number of features to be 150, with a minimum feature distance of 30 pixels, to ensure that the parameter settings were consistent across all compared algorithms. Table 3 demonstrates the experimental results evaluated by the above four metrics, and the best result and the second-best result of the sequence are highlighted in red and blue, respectively. The proposed method is still the best in terms of robustness and localization accuracy. In terms of feature detection and tracking, the proposed algorithm also tracks the most feature points. Superpoint_SLAM outperforms the proposed algorithm in terms of spending on processing images, and AirVO takes the longest time.

3.6. *Real-world experiments*

Whether the proposed algorithm can be effective in the real world is also what our work is interested in. To verify this, we conducted a real-world experiment in a 5 m × 6 m dark room. First, we used the Realsense D435i camera to collect the image data and the IMU data. Then, the Optical Positioning Motion Capture Camera System RTS 2000 was used to provide the ground truth. Finally, we collected two illumination-challenging sequences and evaluated the performance of all compared algorithms using the RMSE metric and the CR metric. If you are interested in the data we collect, you can find it at "https://gitee.com/mycat-123/illumination_-rubust_-visual_-inertial_-odometry.git".

Table 4 demonstrates the experimental results evaluated by the above two metrics, and the best result and the second-best result of the sequence are highlighted in red and blue, respectively. The experimental result shows that our algorithm maintains good performance even when most other algorithms fail.

3.7. *Evaluation under normal illumination*

In order to comprehensively evaluate the proposed algorithm, we conduct an experiment under normal illumination to compare the proposed algorithm with the baseline algorithm VINS-Mono. Three data sequences from M2DGR, Room_01-03, are used to test.

In this experiment, we only use the RMSE metric for evaluation because there is no illumination challenging factor, and the real-time performance has been verified in the above experiments. Table 5 demonstrates the experimental results. The results show that the proposed algorithm maintains comparable or even better performance in localization accuracy compared to VINS-Mono under normal illumination.

Table 4. Experimental result of RMSE(m) and CR(%) on collected sequences.

	Sequence_01		Sequence_02	
	RMSE↓	CR↑	RMSE↓	CR↑
VINS-Mono	—	0	—	0
OpenVINS	—	0	—	0
R2VIO	37.096	0	—	0
DarkSLAM	0.445	64.9%	—	0
VINS-Mono-RUAS	0.610	53.7%	—	0
VINS-Mono-ZD	0.415	72.6%	0.332	35.8%
SP_SLAM	0.485	50.3%	—	0
AirVO	—	0	—	0
Ours	0.386	73.6%	0.259	74.8%
Ours_BCI	0.372	79.8%	0.260	76.5%

Table 5. The RMSE(m) of the proposed algorithm and VINS-Mono on the good illumination sequences.

	Room_01	Room_02	Room_03
VINS-Mono	0.340	0.441	0.457
Ours	0.333	0.435	0.483
Ours_BCI	0.328	0.436	0.470

4. Conclusion

In this paper, we propose a robust visual-inertial navigation method that can operate robustly under illumination-challenging scenes, such as low light and illumination changes. First, we construct an effective low-light image enhancement model using a DCE and a lightweight convolutional neural network. This model helps recover texture information of low-quality images in illumination-challenging scenes. Second, we propose a brightness consistency inference method based on a Kalman filter algorithm, which ensures that image sequences are brightness-consistent, and feature points are tracked correctly. We compare the proposed method with other state-of-the-art

methods from multiple perspectives on public datasets and the real world. The results indicate that the proposed algorithm can run in real-time at 15 fps on an RGB dataset with an image size of 640 × 480. Compared to the baseline algorithm VINS-Mono, the tracking time is improved from 22.0% to 68.2% and the localization accuracy is improved from 0.489 m to 0.258 m on the darkest sequences.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62303040, in part by China Postdoctoral Science Foundation 2023M741882, and in part by the National Natural Science Foundation of China under Grant No. 62273195.

ORCID

Jiankang Qu https://orcid.org/0009-0006-0998-1679
Xu Lyu https://orcid.org/0000-0003-4171-1864
Ziyang Meng https://orcid.org/0000-0002-3742-0039
Pengfei Gu https://orcid.org/0009-0008-8071-7701
Jun Wang https://orcid.org/0000-0002-3686-9272

References

[1] R. Mur-Artal and J. D. Tardós, ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras, *IEEE Trans. Robot.* **33** (2017) 1255–1262.

[2] T. Qin, P. Li and S. Shen, VINS-Mono: A robust and versatile monocular visual-inertial state estimator, *IEEE Trans. Robot.* **34** (2018) 1004–1020.

[3] Y. Tian, B. Fan and F. Wu, L2-Net: Deep learning of discriminative patch descriptor in Euclidean space, *2017 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, Honolulu, HI, USA, 2017), pp. 661–669.

[4] P. Chao, C. Y. Kao, Y. S. Ruan, C. H. Huang and Y. L. Lin, Hardnet: A low memory traffic network, *2019 Int. Conf. Computer Vision* (IEEE, Seoul, South Korea, 2019), pp. 3552–3561.

[5] D. DeTone, T. Malisiewicz and A. Rabinovich, Superpoint: Self-supervised interest point detection and description, *2018 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, San Diego, California, USA, 2018), pp. 224–236.

[6] P. E. Sarlin, D. DeTone, T. Malisiewicz and A. Rabinovich, Superglue: Learning feature matching with graph neural networks, *2020 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Seattle, Washington, USA, 2020), pp. 4938–4947.

[7] J. Sun, Z. Shen, Y. Wang, H. Bao and X. Zhou, LoFTR: Detector-free local feature matching with transformers, *2021 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Nashville, Tennessee, USA, 2021), pp. 8922–8931.

[8] P. Lindenberger, P. E. Sarlin and M. Pollefeys, Lightglue: Local feature matching at light speed, *2023 Int. Conf. Computer Vision* (IEEE, Paris, Île-de-France, France, 2023), pp. 17627–17638.

[9] X. Wang, Z. Liu, Y. Hu, W. Xi, W. Yu and D. Zou, Featurebooster: Boosting feature descriptors with a lightweight neural network, *2023 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, Vancouver, BC, Canada, 2023), pp. 7630–7639.

[10] R. Pautrat, I. Suárez, Y. Yu, M. Pollefeys and V. Larsson, Gluestick: Robust image matching by sticking points and lines together, *2023 Int. Conf. Computer Vision* (IEEE, Paris, Île-de-France, France, 2023), pp. 9706–9716.

[11] E. Santellani, C. Sormann, M. Rossi, A. Kuhn and F. Fraundorfer, S-TREK: Sequential translation and rotation equivariant keypoints for local feature extraction, *2023 Int. Conf. Computer Vision* (IEEE, Paris, Île-de-France, France, 2023), pp. 9728–9737.

[12] Y. Wang, X. He, S. Peng, D. Tan and X. Zhou, Efficient LoFTR: Semi-dense local feature matching with sparse-like speed, *2024 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE, Seattle, Washington, USA, 2024), pp. 21666–21675.

[13] E. H. Land, The retinex theory of color vision, *Sci. Am.* **237** (1977) 108–129.

[14] C. Wei, W. Wang, W. Yang and J. Liu, Deep retinex decomposition for low-light enhancement, arXiv:1808.04560.

[15] Y. Zhang, X. Guo, J. Ma, W. Liu and J. Zhang, Beyond brightening low-light images, *International Journal of Computer Vision* **129** (2021) 1013–1037.

[16] R. Liu, L. Ma, J. Zhang, X. Fan and Z. Luo, Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement, *2021 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (Nashville, Tennessee, USA, 2021), pp. 10561–10570.

[17] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen and Z. Wang, Enlightengan: Deep light enhancement without paired supervision, *IEEE Trans. Image Process.* **30** (2021) 2340–2349.

[18] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong and R. Cong, Zero-reference deep curve estimation for low-light image enhancement, *2020 Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2020), pp. 1780–1789.

[19] Y. Jia, W. Yu, G. Chen and L. Zhao, Nighttime road scene image enhancement based on cycle-consistent generative adversarial network, *Sci. Rep.* **14** (2024) 14375.

[20] A. Savinykh, M. Kurenkov, E. Kruzhkov, E. Yudin, A. Potapov, P. Karpyshev and D. Tsetserukou, Darkslam: Gan-assisted visual slam for reliable operation in low-light conditions, *2022 IEEE 95th Vehicular Technology Conf. (VTC2022-Spring)* (Helsinki, Finland, 2022), pp. 1–6.

[21] S. P. Singh, B. Mazotti, S. Mayilvahanan, G. Li, D. M. Rajani and M. Ghaffari, Twilight SLAM: A comparative study of low-light visual SLAM pipelines, arXiv:2304.11310.

[22] P. Liu, X. Yuan, C. Zhang, Y. Song, C. Liu and Z. Li, Real-time photometric calibrated monocular direct visual SLAM, *Sensors* **19** (2019) 3604.

[23] Y. Chen, S. Liu, H. Ji and G. Qin, Design radiosity restoration algorithm based on visual SLAM, *2022 IEEE 10th Joint Int. Information Technology and Artificial Intelligence Conf. (ITAIC)* (Chongqing, China, 2022), pp. 1143–1147.

[24] R. Gomez-Ojeda, F. A. Moreno, D. Zuniga-Noël, D. Scaramuzza and J. Gonzalez-Jimenez, PL-SLAM: A stereo SLAM system through the combination of points and line segments, *IEEE Trans. Robot.* **35** (2019) 734–746.

[25] C. Kim, J. Kim and H. J. Kim, Edge-based visual odometry with stereo cameras using multiple oriented quadtrees, *2020 IEEE/RSJ/ Int. Conf. Intelligent Robots and Systems (IROS)* (IEEE, Las Vegas, Nevada, USA, 2020), pp. 5917–5924.

[26] B. Zhang, Y. Wang, W. Yao and G. Sun, HTSR-VIO: Real-time line-based visual-inertial odometry with point-line hybrid tracking and structural regularity, *IEEE Sens. J.* **24** (2024) 11024–11035.

[27] C. Li, C. Guo and C. C. Loy, Learning to enhance low-light image via zero-reference deep curve estimation, *IEEE Trans. Pattern Anal. Mach. Intell.* **44** (2021) 4225–4238.
[28] K. Zhang, W. Zuo and L. Zhang, FFDNet: Toward a fast and flexible solution for CNN-based image denoising, *IEEE Trans. Image Process.* **27** (2018) 4608–4622.
[29] L. T. Hsu, N. Kubo, W. Wen, W. Chen, Z. Liu, T. Suzuki and J. Meguro, UrbanNav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas, *Proc. 34th Int. Technical Meeting of the Satellite Division of the Institute of Navigation* (2021), pp. 226–256.
[30] J. Yin, A. Li, T. Li, W. Yu and D. Zou, M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots, *IEEE Robot. Autom. Lett.* **7** (2021) 2266–2273.
[31] J. Cai, S. Gu and L. Zhang, Learning a deep single image contrast enhancer from multi-exposure images, *IEEE Trans. Image Process.* **27** (2018) 2049–2062.
[32] D. P. Kingma, Adam: A method for stochastic optimization, arXiv: 1412.6980.
[33] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long and Q. She, Are we ready for service robots? The openloris-scene datasets for lifelong slam, *2020 IEEE Int. Conf. Robotics and Automation (ICRA)* (Paris, France, 2020), pp. 3139–3145.
[34] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang and G. Huang, Openvins: A research platform for visual-inertial estimation, *2020 IEEE Int. Conf. Robotics and Automation (ICRA)* (2020), pp. 4666–4672.
[35] Z. Huai and G. Huang, Square-root robocentric visual-inertial odometry with online spatiotemporal calibration, *IEEE Robot. Autom. Lett.* **7** (2022) 9961–9968.
[36] C. Deng, K. Qiu, R. Xiong and C. Zhou, Comparative study of deep learning based features in SLAM, *2019 4th Asia-Pacific Conf. Intelligent Robot Systems (ACIRS)* (Nagoya, Japan, 2019), pp. 250–254.
[37] K. Xu, Y. Hao, S. Yuan, C. Wang and L. Xie, Airvo: An illumination-robust point-line visual odometry, *IEEE/RSJ International Conf. Intelligent Robots and Systems (IROS)* (IEEE, Detroit, Michigan, USA, 2023), pp. 3429–3436.



Jiankang Qu received the B.Eng. degree from Changsha University, China, in 2022. He is currently pursuing a master's degree at Beijing Information Science and Technology University. His current research interests include deep learning, visual SLAM, and autonomous robotics systems.



Xu Lyu (Member, IEEE) received the B.Sc. and M. Sc. degrees in control theory and control engineering from the Department of Electrical Engineering, Liaoning University of Technology, Jinzhou, China, in 2014 and 2019, respectively, and Ph.D. degree in navigation, guidance and control from the Department of Navigation Engineering, Naval University of Engineering, Wuhan, China, in 2022.

He is currently a Postdoctoral Fellow with the Department of Precision Instrument, Tsinghua University, Beijing, China. His main research interests include inertial navigation systems, integrated navigation, visual inertial navigation, cluster collaborative navigation and control.



Ziyang Meng (Senior Member, IEEE) was born in 1984. He received the B.S. degree (Hons.) from the Huazhong University of Science and Technology, Wuhan, China in 2006, and the Ph.D. degree from Tsinghua University, Beijing, China, in 2010. He was an Exchange Ph.D. Student with Utah State University, Logan, UT, USA, from 2008 to 2009. He is currently an Associate Professor with the Department of Precision Instrument, Tsinghua University. Prior to joining Tsinghua University, he was a Post Doctoral Researcher, a Researcher, and a Humboldt Research Fellow with Shanghai Jiao Tong University, Shanghai, China, the KTH Royal Institute of Technology, Stockholm, Sweden, and the Technical University of Munich, Munich, Germany, respectively, from 2010 to 2015. His research interests include distributed control and optimization and intelligent navigation technique. He is a fellow of IET. He serves as an Associate Editor for Systems & Control Letters and IET Control Theory & Applications.



Pengfei Gu received the B.E. degree in measurement and control technology and instrumentation from the Department of Precision Instrument, Tsinghua University, Beijing, China in 2020. He is currently working toward the Ph.D. degree with the Department of Precision Instrument, Tsinghua University. His research interests include state estimation, sensor fusion, visual-inertial localization and mapping.



Jun Wang is currently a lecturer with the College of Information Science and Technology, Beijing University of Chemical Technology, China. He received his B.S degree from Beijing Information Science and Technology University, Beijing, China, in 2013. He received his Master's degree and Ph.D. degree in the Department of Human and Artificial Intelligent Systems from the University of Fukui, Fukui, Japan, in 2016 and 2019, respectively. Prior to joining Beijing University of Chemical Technology, he held a postdoc at Tsinghua University, Beijing, China. His research interests include state estimation, SLAM, particle filter, sensor fusion, and applied robotics.