Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5249449, 14 pages
https://doi.org/10.1155/2022/5249449

Research Article

A Lightweight Stereo Visual Odometry System for Navigation of Autonomous Vehicles in Low-Light Conditions

Jie Li,[1] **Zhenfei Kuang**,[1] **Guangman Lu**,[1] **Yuyang Peng**,[2] **Wenli Shang**,[1] **Jun Li**,[1] **and Wei Wei**[1]

[1] *Research Center of Intelligent Communication Engineering, School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China*
[2] *School of Computer Science and Engineering, Macau University of Science and Technology, Taipa, 999078 Macau SAR, China*

Correspondence should be addressed to Wenli Shang; shangwl@gzhu.edu.cn, Jun Li; lijun52018@gzhu.edu.cn, and Wei Wei; wei@gzhu.edu.cn

Received 19 July 2022; Revised 30 August 2022; Accepted 23 September 2022; Published 6 October 2022

Academic Editor: Mingqian Liu

Localization of vehicles in a 3D environment is a basic task in autonomous driving. In the low-light environments, it is difficult to navigate independently using a visual odometry for autonomous driving. The main reason for this challenge is the blurred images in the scenes with insufficient illumination. Although numerous works focused on this issue, it still has a number of inherent drawbacks. In this paper, we propose a lightweight stereo visual odometry system for navigation of autonomous vehicles in low-light situations. Contrary to the existing recovery methods, we aim to divide the captured image into the illumination image as well as the reflectance image and only estimate the illumination one, where the enhanced map of the low-light image is acquired by using the retinex theory. In addition, we further utilize a simplified and rapid feature detection scheme, which reduces the computation time by about 85% but maintaining the matching accuracy similar to that of ORB features. Finally, the experiments show that our average memory consumption of our proposed method is much less than the conventional algorithm.

1. Introduction

Localization is one of the tasks for autonomous driving, and it is also a necessity to achieve automatic navigation, whereas visual odometry (VO) and visual simultaneous localization and mapping (VSLAM) are considered to be the primary technologies to achieve this goal [1]. Visual odometry is the process of estimating the relative camera poses by observing two cameras sharing a common field-of-view [2]. A core part of it is to be able to track a sufficient number of points during the continuous camera movements. These points will be used to calculate the body pose (translation and rotation). Visual odometry can be broadly divided into two categories according to the method of processing the input images. On the one hand, it is an indirect method based on features; on the other hand, it is a direct method based on pixels [3]. The heart of the indirect method is to extract representative points from an image, which are often called features. Then, these points are tracked in the successive frames. The body pose is recovered by minimizing the reprojection error. In contrast, direct methods do not require features. The pixels are usually tracked directly. The body pose is achieved by minimizing the photometric error between pixels. However, one downside of the direct method is that it is highly sensitive to the light of environment. When the illumination changes dramatically, it is often impossible to track the correct points. Further, it might lead to a failure of body pose calculation, that is still a challenge for poor lighting conditions, although the indirect method overcomes this limitation. The intensity of image texture will be diminished in dimly lit scenes. In other words, the indirect methods need to select features based upon the difference of intensity between pixels. Therefore, the darker the image brightness, the smaller the overall difference in intensity, which is not conducive to feature extraction. This may result in the collapse of tracking eventually. Consequently, this paper has made a number of improvements to a visual odometry system for low-light environments to solve the described problems. Our target is to recover as much content as possible from a dark image. Then, robust features are extracted from the images. Ultimately, accurate pose is available.

There are two key components in our work: firstly, recovering detailed information of objects in low-light scenes; and secondly, improving the robustness of features as much as possible. To solve these two challenges, we first make use of the retinex theory [4] in the field of image. We add more realistic content to the recovered reflectance image. This enables us to obtain an enhanced image under low illumination situation. Then, a simple but effective detection scheme is used to extract features from the enhanced image. The feature computation time is cut down as far as possible while maintaining the matching accuracy. Furthermore, we chose the novel 4Seasons dataset [5] to evaluate our method, which contains a wide range of real-world conditions (see Figure 1). It is sufficient to verify the practicality of our approach. In addition, we have compared our method with ORB-SLAM2 [6] which has comparatively good performance at present. The results validate the effectiveness of our method.

The rest of this paper is organized as follows. We introduce the related work on visual odometry in Section 2. The main part of the article is given in Section 3, containing the low-light image enhancement scheme as well as the boosted feature detection method. Experiments and evaluation are illustrated in Section 4, comparing our proposed system in detail from various aspects. Finally, a summary of our work is given in Section 5.

2. Related Work

On the one hand, for feature-based VO/VSLAM systems, the abundant texture is a prerequisite for both accurate and robust tracking. However, the image may be blurred in low illumination scenes. We are unable to extract high-quality features in such cases, which will cause the failure of tracking. Thus, obtaining rich and realistic content from low-light conditions is an urgent challenge. Fortunately, there is a lot of research that addresses this problem. Histogram equalization (HE) and gamma correction (GC) are two widely used approaches at present. Through changing the histogram of an image to alter the distribution of pixel intensity, HE can enhance the low-light images to a certain extent. However, overenhancement is a potential risk in some parts of the image [7] as HE is a global process. In addition, for GC, the local area is not naturalized due to the uniform gamma coefficient used for the global image. A sea of work has patched the weakness of HE and GC, such as [8–12]. But there are still a few remaining problems. In recent years, image enhancement schemes based on retinex theory have been attracting the focus. Its core idea is to decompose the captured image into an illumination image and a reflectance image. An advanced low-light image enhancement algorithm (LIME) was proposed [13] based on this theory. Different from the previous method [14], LIME only estimates the illumination image. The complexity of the computation process is significantly reduced by using the Augmented Lagrangian Multiplier (ALM).

At present, most of VO/VSLAM utilize HE or GC when dealing with low-light scenes, like [15–18]. As mentioned before, they are not up to our requirements. Thanks to the retinex theory, which provides a new perspective for image enhancement, our demands are satisfied. We use the LIME image enhancement method to get the result.

On the other hand, one of the keys to achieving accurate localization for VO and VSLAM lies in selecting a series of features. They are tracked in the continuous frames. Then, the body pose is computed [19]. Although the Scale Invariant Feature Transform (SIFT) [20] has been proposed for almost twenty years, it remains of substantial interest due to its excellent performance in a variety of domains. However, an important drawback of SIFT features is that it imposes a large computational burden. This makes it difficult to employ for VO/VSLAM, which need to be processed in real-time. For this reason, a series of outstanding improvements have followed. In 2006, the notable Speeded Up Robust Features (SURF) was presented [21]. Compared to a previous work, SURF significantly cuts down on the time for feature extraction. Unfortunately, this comes at the cost of retaining the support of GPU devices [22].

Instead of focusing attention on features exclusively, some concentrate on key points or descriptors separately. For key points, the popular one is the Features from Accelerated Segment Test (FAST) detector [23]. This determines whether a pixel is a key point by comparing its intensity with the surrounding pixels. Contrast with other key points such as [24], the FAST key points only need to compare the difference in intensity. Therefore, it is quite fast to compute and holds promise for applications in scenarios in which real-time performance is required. Despite the fast computation speed of FAST, it also has shortcomings. Unlike [20, 21], there is no orientation information for FAST key points. And in a few cases, we want the change of the orientation not to affect the expression of the same key point when the observation angle undergoes a shift. So, FAST may encounter certain difficulties in this situation. For descriptors, the Binary Robust Independent Elementary Feature (BRIEF) [25] has received a lot of attention since it was proposed. The authors use a specific procedure to select multipaired image blocks (usually $9 * 9$ and Gaussian smoothed) centered on a key point. A string of binary is generated according to the discrepancy of the intensity between the image blocks. This is the main thought behind the BRIEF descriptor, which has the strength of being a binary descriptor. We can calculate the hamming distance between descriptors when matching features. It is a simple task for a computer. Unfortunately, as with FAST, BRIEF is also sensitive to rotation.

Considering the strength and weakness of FAST key points as well as BRIEF descriptors, the Oriented FAST and Rotated BRIEF (ORB) features was presented [26]. It keeps the speed merit of them. For their downsides, the

Figure 1: The 4Seasons dataset. We have selected several sequences, containing different environments and weather. The common trait of these sequences is that all of them are low-light scenes.

intensity centroid method [27] was applied. Orientation information is added to the FAST key points so that when the camera rotates, it still corresponds to the same place. Additionally, the descriptors depend on the key points. Therefore, when the key points are rotation invariant, the descriptors will naturally have orientation information as well. It is far less prone to features tracking loss when the image is spun.

Our feature detection scheme is similar to ORB. The difference is that we have chosen the Boosted Efficient Binary Local Image Descriptor (BEBLID) [28] as an alternative instead of BRIEF. The representative intensity pairwise tests were selected within local image regions via a Boosting scheme. Experimentally, our feature detection scheme has proven to reduce the computation time by more than 80% while guaranteeing matching accuracy.

3. Methodology

In this section, we will introduce the theoretical part of our proposed visual odometry system. The experimental part will be given in the next section. The framework of our proposed system is shown in Figure 2. Four main blocks are included: (1) an input data processing component focusing on low-light image enhancement, (2) a tracking component for features reprojection and matching (this is the core body of the system), (3) the initialization stage of the system (the principal function is to initialize the visual odometry system and generate map points), and (4) a map segment containing local map. Here, the map points for the body pose calculation and the candidates of unknown tracking quality are maintained primarily.

In the rest of this section, we will discuss the details of each individual part in more depth.

3.1. Low-Light Image Enhancement. In this paper, images are preprocessed using the LIME algorithm and the Fast Global Image Smoothing algorithm (FGS) [29]. Firstly, FGS is employed to process the low-light image in order to obtain the temporary one. Then, we used LIME to get the enhanced result. The method has been experimentally proven to be able to recover the information of low-light images successfully. It is beneficial to the feature extraction and matching.

The core idea of retinex theory is to decompose the captured image (**L**) into an illumination image (**T**) and a reflectance image (**R**), as shown in Figure 3. According to this, the relationship between these three is as follows:

$$\mathbf{L} = \mathbf{R} * \mathbf{T}, \tag{1}$$

where $*$ stands for element-by-element multiplication.

Let us do a simple transformation of Equation (1) to get

$$\mathbf{R} = \frac{\mathbf{L}}{\mathbf{T}}. \tag{2}$$

The notation '-' in Equation (2) stands for the element-by-element division. By a simple transformation, we change the objective of the solution from **R** to **T**. Thus, estimating **T** is the key to solving for **R**.

We define the problem of solving the illumination image as an optimization problem by minimizing the following

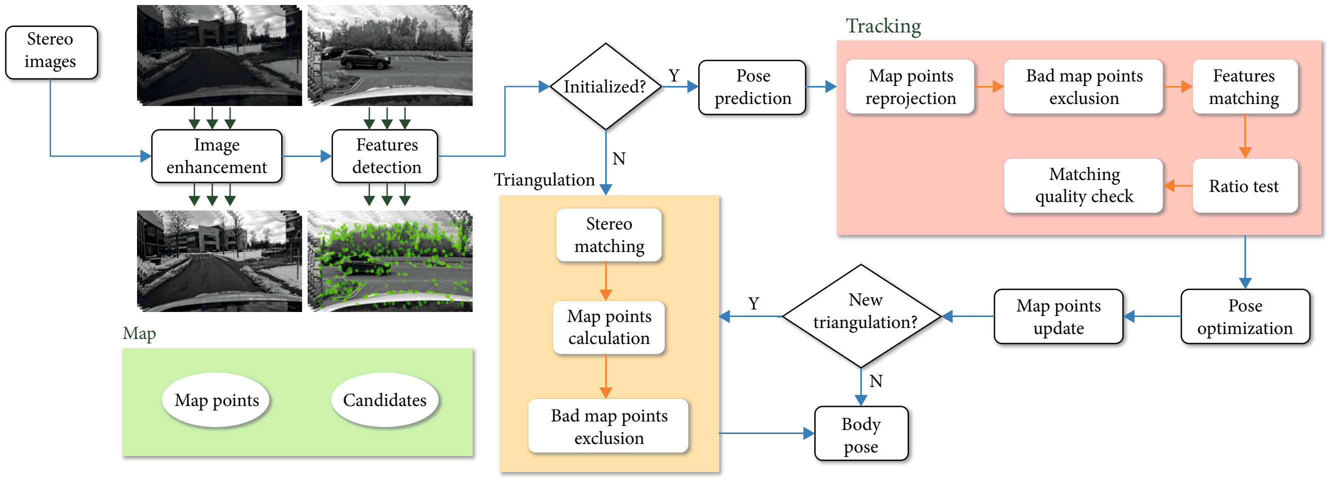


Figure 2: The framework of our proposed system. Our method consists of four parts: preprocessing, tracking, triangulation, and maps. We mainly did some work in the first part so as to improve the quality of the input data. For the other three parts, we implemented simple but useful strategies to realize decent performance.

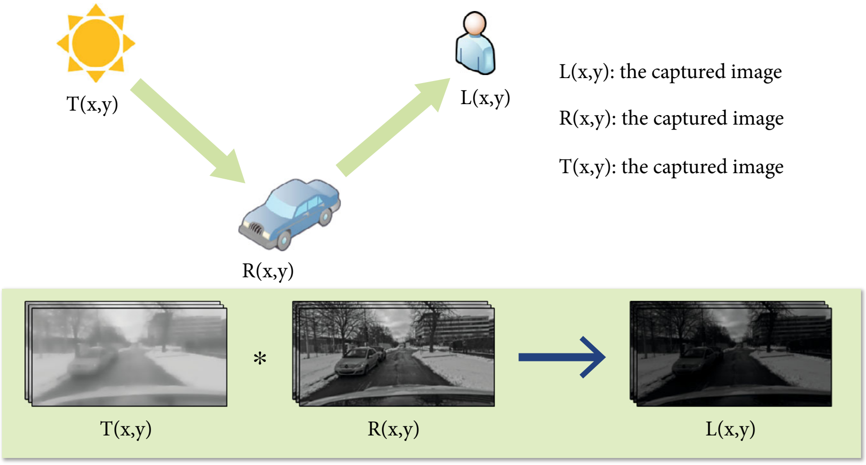

T(x,y)
L(x,y)
R(x,y)
L(x,y): the captured image
R(x,y): the captured image
T(x,y): the captured image
T(x,y)
*
R(x,y)
L(x,y)

Figure 3: The retinex theory. This theory divides the pictures we observed into illuminated and reflected pictures. It gives us a fresh perspective to understand the recovery of low-light images.

weighted least squares (WLS) function [29].

$$J(T) = \sum_{p}\left(\left(T_p - Q_p\right)^2 + \lambda \sum_{q\in N(p)} \omega_{p,q}(L)\left(T_p - T_q\right)^2\right), \quad (3)$$

where $Q = \|L\|$; T_p represents the intensity of pixel p, $p = (x, y), 0 \le x < W, 0 \le y < H$; W, H are the width and height of the image, respectively; λ is a control parameter to balance the terms on both sides of the plus sign; $N(p)$ represents the neighbourhood pixels of p; and $\omega_{p,q}(L)$ is a weighting function defined on $\mathbf{L}$.

$$\omega_{p,q}(L) = \exp\left(\frac{-\left(L_p - L_q\right)^2}{\sigma}\right), \quad (4)$$

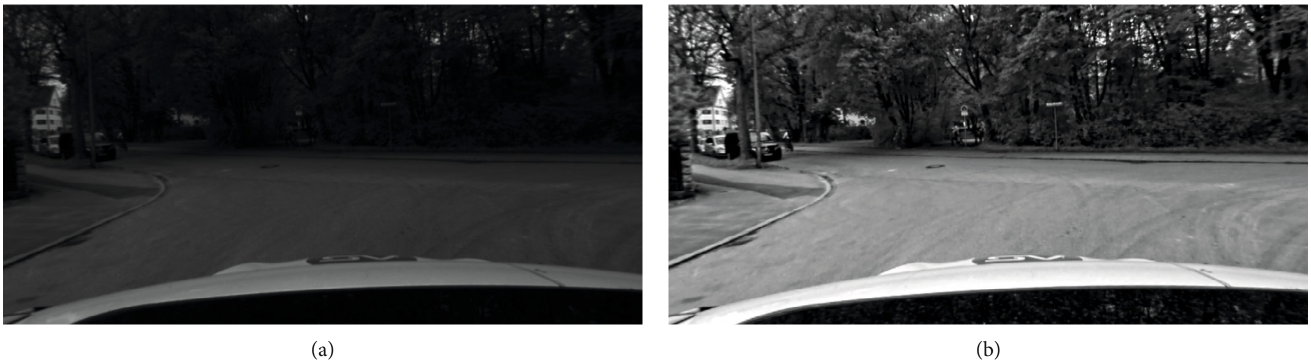
where σ is a range parameter. The effect of Equation (4) is to smooth the image texture at the rest of the location while preserving the object edge features.

To minimize the problem (1), let the derivative of $J(T)$ be zero. The following system of linear equations is derived:

$$(\mathbf{I} + \lambda\mathbf{A})\mathbf{T} = \mathbf{Q}, \quad (5)$$

where $\mathbf{I}$ represents the identity matrix; $\mathbf{T}$ and $\mathbf{Q}$ are $S \times 1$ -dimensional column vectors containing elements T and Q , $S = W \times H$; and $\mathbf{A}$ is a spatially varying Laplacian matrix of size $S \times S$ similar to the one defined in [30].

However, for a two-dimensional image, Equation (3) is a weighted L_2 norm objective function that is very difficult to solve directly. In order to satisfy the requirement of visual odometry, problem (3) can be decomposed into two subproblems (the vertical direction and the horizontal direction). As each subproblem is a one-dimensional linear system, the solution method is mature. Therefore, a fast and accurate calculation can be achieved.


(a)

(b)

Figure 4: The low-light scene enhancement method. (a) The original image. (b) The enhanced image.

Figure 4 shows the results of the low-light image processing, from which it can be indicated that our method can recover the image detail information. More comparative experiments can be found in Section 4.1.

3.2. Key Point Extraction and Descriptor Calculation. After the low-light image is enhanced, the next important step is to extract features from the enhanced one. We use a fusion of ORB key points and BEBLID descriptors. ORB key points are built on the famous FAST corners, and it performs the following detection procedure.

As shown in Figure 5, the central pixel is set to be p in a small adjacent area and its intensity is noted as I_p. Firstly, a circle with a radius of three is constructed by taking the pixel p as the center. Secondly, the sixteen pixels located on the circumference of the circle are selected, and their serial numbers $i\,(i = 1, \cdots, 16)$ are recorded in clockwise direction. Thirdly, a threshold T is set for comparing the discrepancy between the intensity of the central pixel and the individual circumference points. Fourthly, the absolute value of the difference in intensity between p and the sixteen points is calculated in turn $S_i = |I_p - I_i|$, and compare S_i with T. Finally, if there are N consecutive pixels with S_i greater than T, the central one is considered to be the key point.

For the descriptors, we use the BEBLID descriptor which was published recently. By choosing a series of specific weak learners (WLs) and using the integral image, it can outperform the fastest ORB descriptor in terms of speed. It is also comparable to SIFT in accuracy. The BEBLID descriptor is based on the work of the Boosted Efficient Local Image Descriptor (BELID) [31]. The major change between these is that the former converts the real-type descriptor into a binary one. Apart from this, it also uses the AdaBoost algorithm for WL selection and then combines all WLs to form a stronger message.

$$\mathscr{L}_{\text{BEBLID}} = \sum_{i=1}^{N} \exp\left(-\gamma l_i \sum_{k=1}^{K} h_k(\mathbf{x}) h_k(\mathbf{y})\right). \tag{6}$$

Equation (6) is the loss function for BEBLID, where γ is the weight of WLs, l_i is the training sample label, $\{\mathbf{x}, \mathbf{y}\}$ is a training set consisting of image block pairs, and $h_k(\mathbf{z}) \equiv h_k(\mathbf{z}\,; f, T)$ represents the k^{th} WLs, which depends on the feature extraction function $f : \mathscr{X} \longrightarrow \mathbb{R}$ and the threshold T. By giving these two parameters, we derive

$$h(\mathbf{x}\,; f, T) = \begin{cases} 1, & \text{if } f(\mathbf{x}) \le T, \\ 0, & \text{if } f(\mathbf{x}) > T. \end{cases} \tag{7}$$

In particular, the key for the BEBLID descriptor is the choice of $f(\mathbf{x})$ in Equation (7). The authors define it as the difference in the average intensity of the pixels between the two image blocks.

$$f(\mathbf{x}\,; p_1, p_2, s) = \frac{1}{s^2}\left(\sum_{q \in R(p_1, s)} I(q) - \sum_{r \in R(p_2, s)} I(r)\right), \tag{8}$$

where $I(t)$ denotes the intensity of pixel t and $R(p, s)$ represents a square adjacent area with pixel p as the center and side length s.

In summary, we are able to access the BEBLID descriptor. It is worth mentioning that, unlike the previous work, the weights of all WLs are set to the same value.

3.3. Feature Matching and Map Point Tracking. In this part, we will illustrate the step of features matching and map points tracking. The performance of features matching is closely related to the accuracy of the camera pose. Therefore, we adopt a coarse-to-fine approach. Firstly, the features in two images are matched roughly. Then, a ratio test is used to select the best descriptor among them. A typical feature matching result is shown in Figure 6, where it is clear that most of the initial matches are correct. This proves the effectiveness of our method. Also note that there are two parts of this system that use the features matching algorithm. One is that the system is not initialized. Stereo matching is employed for the first frame to initialize the entire visual odometry system. It is at this stage that the map points are generated. The second is after initialization. In the tracking phase, the map points are tracked between the two frames. These two parts will be described in detail below.

When the visual odometry is started, the first frame is used for initialization. First of all, we follow the method in the previous two parts (see Sections 3.1 and 3.2) to enhance the low-light image and extract the features. Subsequently,

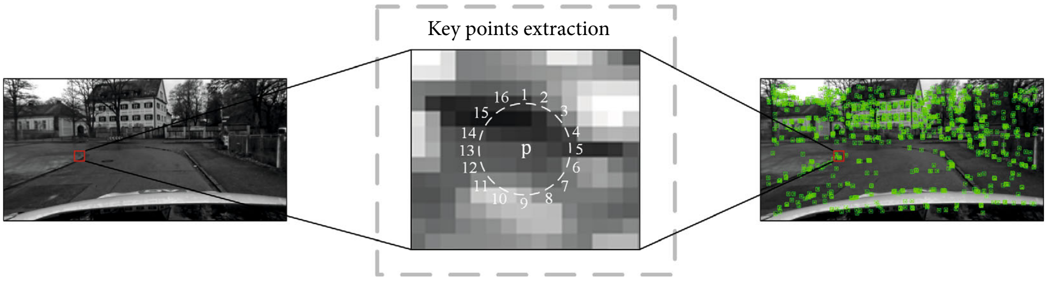


Figure 5: The FAST corner. We determine whether a central pixel p is a key point based on the difference between the intensity of that it and the sixteen circumference points in the adjacent domain.

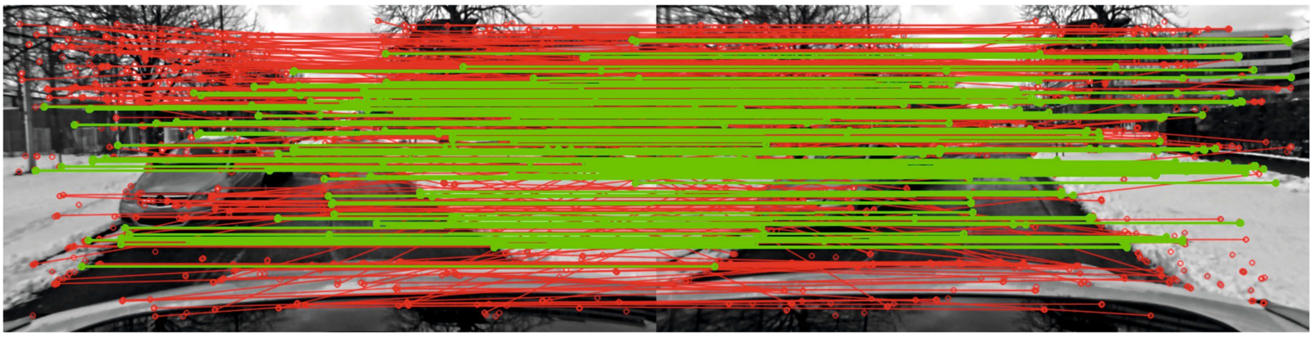


Figure 6: The result of image features matching. Green lines indicate correctly matched features. Conversely, wrong match relationships are represented using red lines.

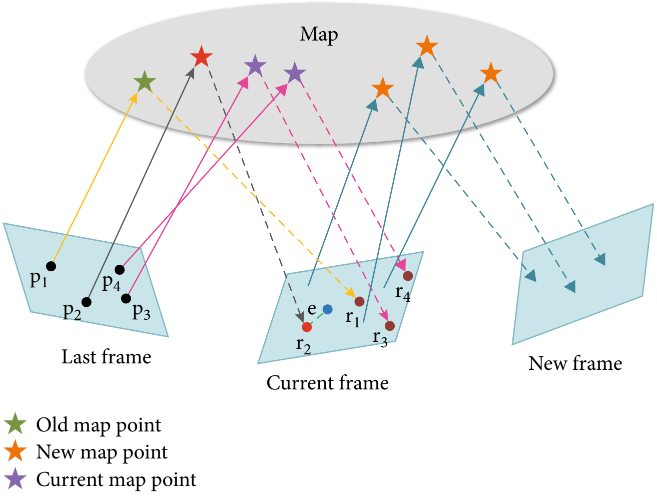


Figure 7: The reprojection model and local map. The spatial point, which is represented as a red pentagram, is reprojected into the current frame. The letter e denotes the reprojection error, which is also the variable to be optimized (see Section 3.4).

we find the initial matching points in the left and right images with the function from the OpenCV library. Next, we use the ratio test to pick out the best match. In particular, we used the local domain search method to reduce the load during searching. Once the features have been matched in the first frame, we should compute the corresponding map points according to the matched features and track them in the subsequent frames.

As soon as the second frame is available, there are already map points at this stage. Therefore, we employ a reprojection model $\pi(\cdot)$ to reproject the map points from the previous frame into the left image of the current frame, as shown in Figure 7. The red solid point indicates reprojection point, which is converted from a map point (red pentagon) using the reprojection model. Afterwards, the reprojection points are matched with features. Similar to the previous paragraph, we continue to reduce the search load as well as improve the matching accuracy by using local domain search and ratio test. If the amount of map points tracked successfully for the first time is inadequate, we also extend the search radius to twice the original size. The reprojection and matching of map points is performed again in the enlarged area. This is done in the hope that a sufficient number of correct matches can be found. Ultimately, the possibility of successful tracking is boosted.

3.4. Pose Estimation and Optimization. The prediction and optimization of camera pose is another central part of visual odometry. It receives matching information from the features and recovers the pose based on these matches. We set the camera pose to $\mathbf{T}^k$ for the k^{th} frame with consist of a quaternion $\mathbf{q}_k \in SO(3)$ as rotation and a position $\mathbf{t}_k \in \mathbb{R}^3$ as translation. We rely on the algorithm in [32] to predict the pose of the current frame.

Once the pose is estimated, the map points from the previous frame are reprojected into the current frame using the reprojection method (see Section 3.3). Then, we track between the reprojected points and the features in the current frame. Finally, the predicted poses are jointly optimized using the updated map points and the matched points. The cost function of the poses is defined as follows:

$$\{\mathbf{R}, \mathbf{t}\} = \underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \sum_{i \in S} \rho\left(\left\| p^i - \pi\left(\mathbf{R}P^i + \mathbf{t}\right) \right\|^2 \right). \tag{9}$$

In Equation (9), $\mathbf{R}, \mathbf{t}$ refers to the camera pose, which we split into a rotation matrix and a translation vector. S denotes the set of all matched points. ρ is the Cauchy cost function. p^i indicates the i^{th} features. $\pi(\cdot)$ stands for the reprojection model. As mentioned earlier, we use this model to compute the reprojection points of the previous map points. P^i represents the i^{th} spatial map point.

Lastly, we use the g2o optimization library [33] to solve problem (9). The objective is to obtain the rotation matrix and translation vector corresponding to minimize the cost function and use them as the optimized camera pose.

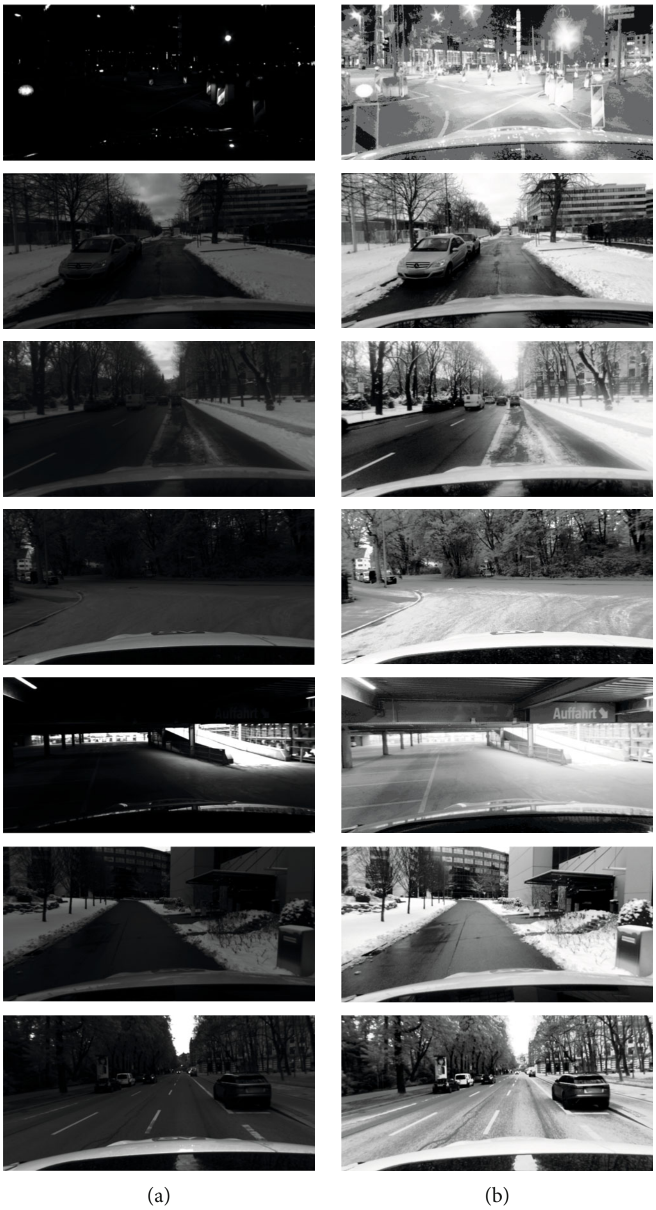
3.5. Local Map. To improve the performance of the proposed system, we also maintain a local map. In the local map, the map points for pose optimization are included. In other words, not all map points are in this space. We refer to map points that are not in the local map as “candidates”. They are always ready to be added to local map. All map points calculated by features matching are considered candidates initially. None of them belong to the local map. The map points are only added to the local map when they can be successfully tracked in a certain number of consecutive frames. We believe that the tracking quality of these map points is better. The new map point is a transformation from the candidates, which is represented by the orange pentagram in Figure 7. However, there is a special case. That is to say, if the number of map points in the local map falls below a threshold, we consider that there is a risk of tracking failure. Therefore, candidates should be joined to the local map immediately regardless of the quality of tracking to prevent failure. In particular, we directly add the map points generated in the first frame to the local map during the initialization phase. There are no map points in the local map at this time. In the subsequent step, the new map points are treated as candidates using the method described above. Tracking quality is used to judge whether or not it should be inserted into the local map.

4. Experiments

To evaluate our proposed method, the 4Seasons dataset is used and compared with the current superior performance of ORB-SLAM2. As a novel dataset, it has a wide range of abundant scenarios, from urban to rural and from parking to motorway. Unlike some previous datasets [34], the 4Seasons dataset also includes a variety of weather and lighting conditions. Meanwhile, the 4Seasons dataset utilizes a simple data acquisition system consisting of a stereo camera vision system (Basler acA2040-35gm), an inertial measurement unit system (Analog Devices ADIS16465), and an RTK-GNSS system (mosaic-X5). Finally, the fusion of the visual system with the RTK-GNSS data provides centimetre-level positioning accuracy, which will greatly contribute to the performance evaluation of the algorithm. More details on the 4Seasons dataset can be found in [5]. In addition, ORB-SLAM2, one of the more outstanding SLAM algorithms, has received considerable attention from a broad mass of researchers since it was proposed. Different from our proposed algorithm, ORB-SLAM2 is a complete SLAM system that supports multiple functions, including loop closing, place recognition, and so on. However, it is the power of its features that brings a heavy computation burden. In the experimental section, qualitative and quantitative results will be given in order to demonstrate the merits of our proposed system.

4.1. Dark Image Recovery. We investigate the capabilities of our low-light image enhancement algorithm. To demonstrate the superiority, we compare other low-light image restoration schemes commonly used today, including histogram equalization and adaptive gamma correction. We selected a number of representative raw images from the 4Seasons dataset, containing various scenes, weather, etc., as shown in Figure 8. The strengths of ours can be clearly observed by doing different processing methods on the same image. We are able to maintain both the detail information and the global quality of the image in comparison to the other two methods. In particular, there is hardly any illumination in the first row of images. Although our result is not as bright as the other two, we succeed in recovering the full content of the image. For the other two methods it fails, as shown at the top of the image. There are a few completely dark areas that indicate the recovery was not successful. For the rest of low-light environments, our method takes into account the local details while keeping the whole content consistent. This will be analysed in the following.

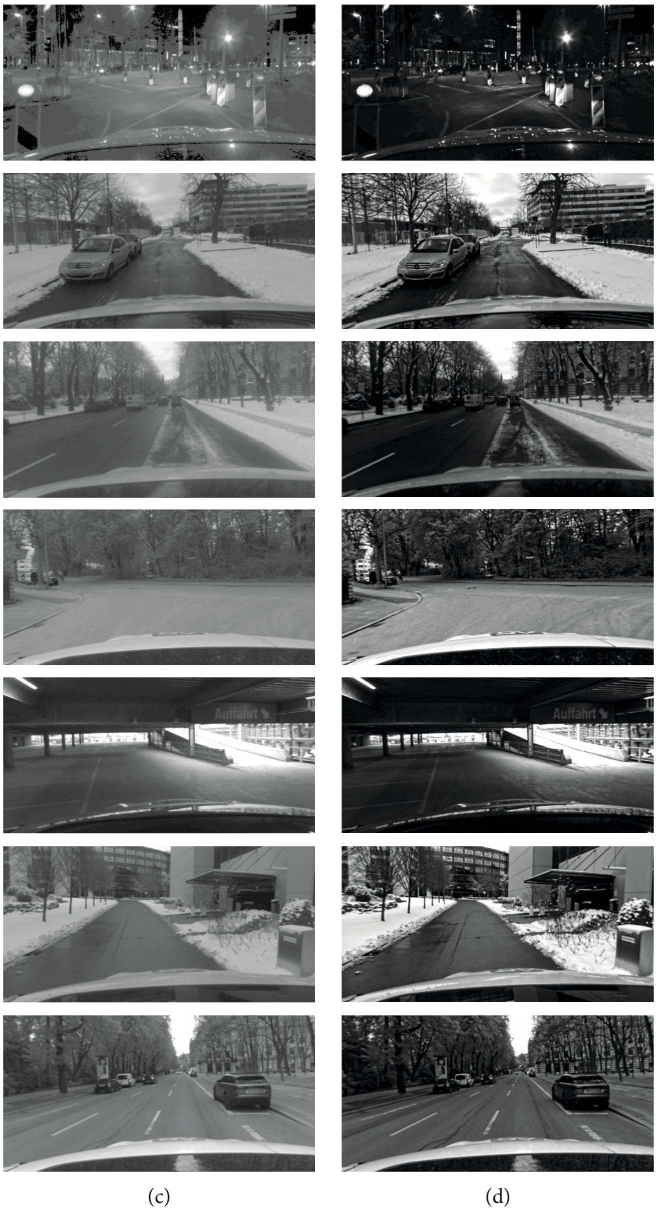
With histogram equalization, the inherent drawbacks lead to a situation where the image appears exposed unnaturally. This is shown by the snow in the bottom right corner of the third row of images and the edge of the road on the left in the fourth row of images. The details are also not handled well enough. Examples include the tree branches in the



(a)

(b)

Figure 8: Continued.



(c) (d)

Figure 8: Results of the histogram equalization, adaptive gamma correction, and low-light image enhancement schemes. (a) The original image. (b) The results of histogram equalization. (c) The outcome of adaptive gamma correction. (d) The results of low-light image enhancement we used.

second row and the road in front of the vehicle in the seventh row. The histogram equalization comes closest to the performance of our method in the sixth row of images, but the image contrast is still inadequate for minor information such as roads and bushes.

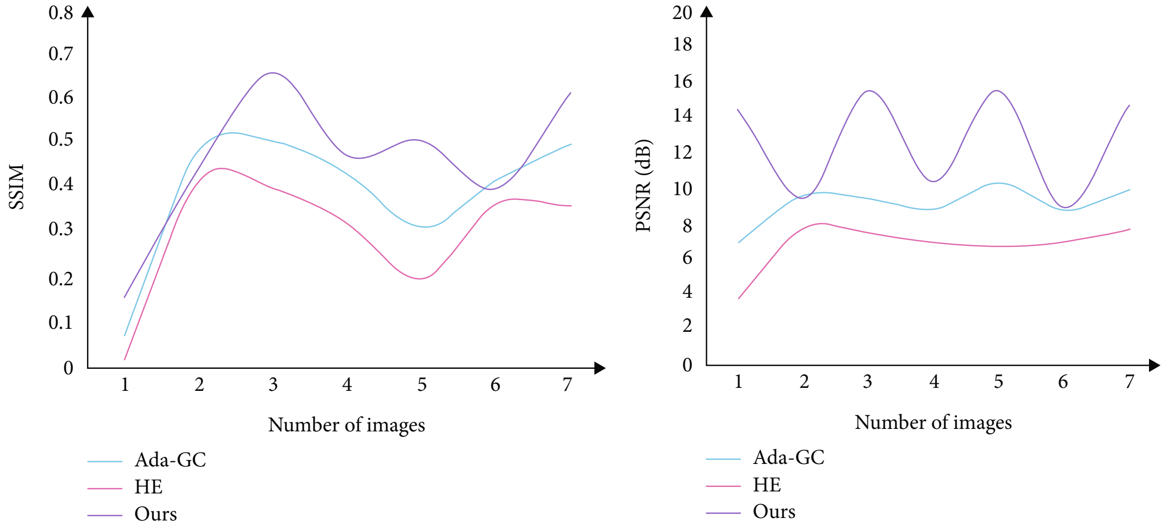
For adaptive gamma correction, it can automatically adjust the gamma parameter according to the image content, avoiding the shortcomings of fixed parameter. However, there are still challenges. The most noticeable of these can be identified in all images. The results are not sharp sufficiently and look like a haze image. This causes a deterioration in the general quality and is not conducive to the feature extraction and matching.

In our method, it is possible to restore local details while taking into account the global content of the image. Especially for almost dark scenes (as in the first line of Figure 8), we recovered all parts of the image successfully, whereas the other two methods failed. In the fifth row, our method does not seem to work as well as histogram equalization, for example, in the upper part of the image. Nevertheless, our aim is to make the enhanced image more realistic. It avoids the local distortion of content. Yet histogram equalization has led to a partial overexposure of information, as in the case of the garage exit.

Besides, in order to quantify the superiority of ours, we make use of PSNR and SSIM [35] as evaluation metrics.

Table 1: Comparison of PSNR&SSIM for adaptive gamma correction, histogram equalization and our method.

Images	Ada-gamma correction		Histogram equalization		Ours	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
1st row	7.8633	0.0837	4.3233	0.0229	16.1813	0.1802
2nd row	10.8508	0.5457	8.7224	0.4692	10.6581	0.5003
3rd row	10.5836	0.5674	8.4229	0.4509	17.4225	0.7377
4th row	9.9263	0.4876	7.8555	0.3623	11.6864	0.5310
5th row	11.6111	0.3531	7.5862	0.2236	17.4387	0.5689
6th row	9.9133	0.4696	7.8530	0.4106	10.0635	0.4492
7th row	11.1430	0.5586	8.5902	0.4063	16.5346	0.6859
Average	10.2702	0.4380	7.6219	0.3351	14.2836	0.5219



(a)

(b)

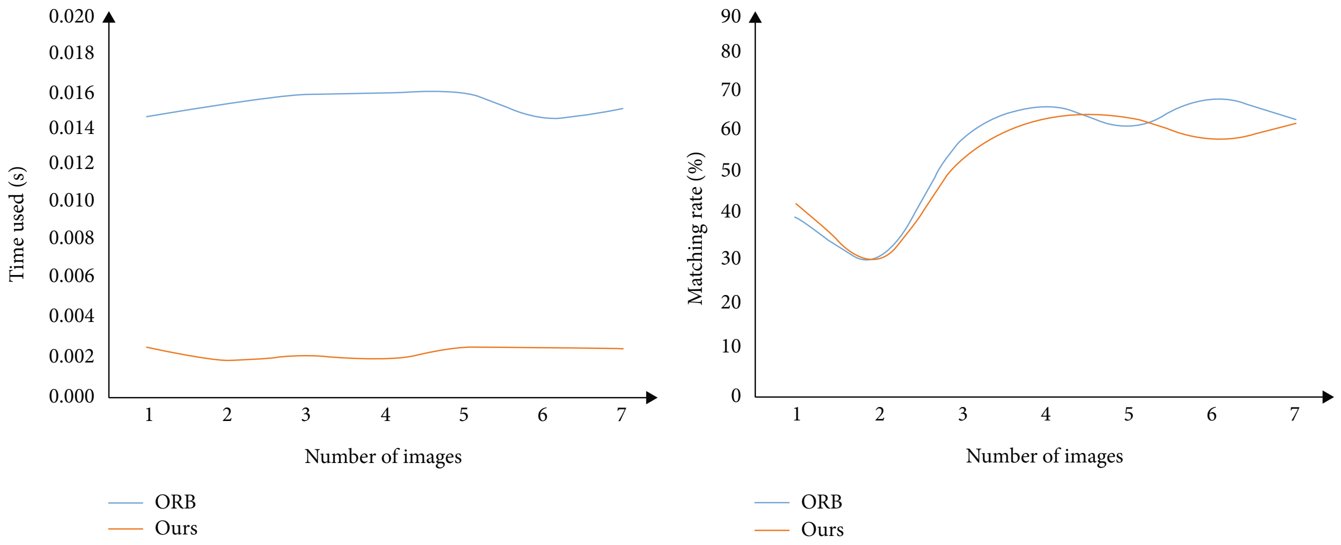
Figure 9: Comparing PSNR and SSIM for adaptive gamma correction, histogram equalization, and our method. Ada-GC: adaptive gamma correction; HE: histogram equalization; Ours: for low-light image recovery method we used. (a) Comparison of SSIM. (b) Comparison of PSNR.

SSIM means structural similarity. It compares the similarity between the processed result and the original image in three different aspects. A larger value of SSIM means a higher similarity between the two images. PSNR, on the other hand, represents the peak signal-to-noise ratio. Similar to SSIM, the higher the value, the better the result.

Table 1 shows the outcome of our PSNR and SSIM compared to the others. From the table, it can be concluded that our low-light image enhancement algorithm achieves decent performance on both PSNR and SSIM. Although for the second image the adaptive gamma correction is processed optimally, our results are similar to it. Not to mention that we have pretty good results in all the other images. In Figure 9, we visualize the performance of three algorithms. As reflected in the table, neither the adaptive gamma correction nor the histogram equalization performs as well as ours.

4.2. Stereo Matching. This part compares the benefits of the fusion methods we used for key points and descriptors. We still selected images in the dark image recovery part as test data. Figure 10 depicts the correct matching rate and the time loss of descriptor computation between our method and the ORB features with the better performance at present. We do feature extraction on the low illumination enhanced images. Then, the right match was found between the stereo images. In the figure on the right, it can be noticed that our approach is up to par with the ORB features with respect to the correct matching rate. In the test data, the average right match percentage for the ORB features is about 63.27%, while ours is about 61.25%. The graph on the left represents the time loss of descriptor computation for both methods. We can identify the upper hand of our method clearly. In terms of speed, our method improves by about 84.52%. Thus, our time loss is substantially reduced at a similar correct matching rate to the ORB features. It is sufficient to demonstrate the strengths of our method.

4.3. Accuracy. We have examined the performance of our system in the recently released 4Seasons dataset. Eleven sequences were selected to suit our requirements. We have



(a)

(b)

Figure 10: Comparison of ORB features with the method we used. (a) Comparison of descriptor computation time. (b) Comparison of correct matching rate.

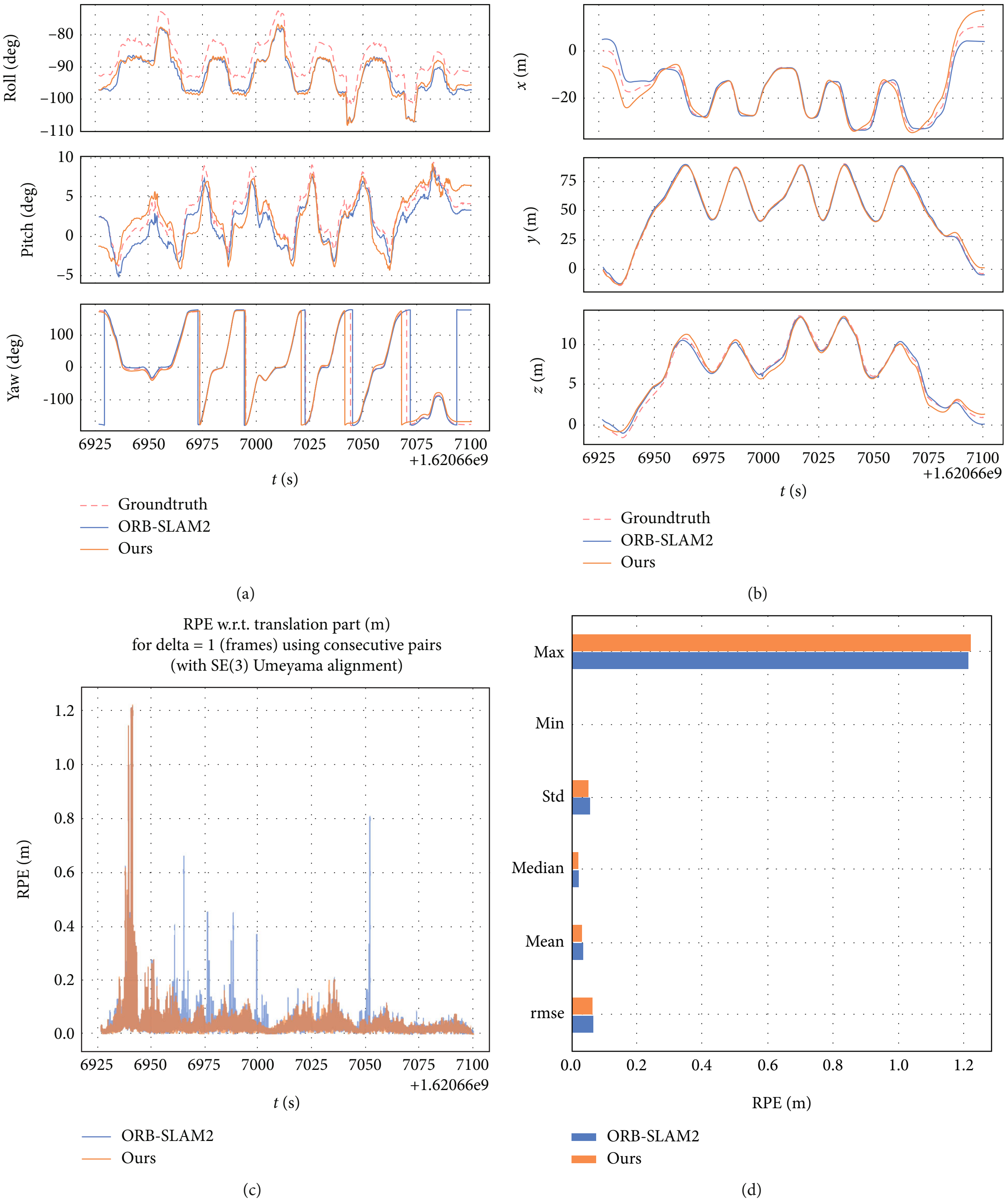
Table 2: Comparison of accuracy and memory in the 4Seasons dataset.

Scenes	Sequences	Number of images	ORB-SLAM2		Ours	
			RMSE (m)	MEMORY (GiB)	RMSE (m)	MEMORY (MiB)
OLD TOWN	2021-01-07-10-49-45	24629	0.084	4.9	0.079	66.2
	2021-05-10-21-32-00	24658	*	*	0.605	71.8
	2020-10-08-11-53-41	28999	0.053	3.5	0.044	79.1
COUNTRYSIDE	2021-01-07-13-30-07	14729	0.111	3.8	0.115	48.3
OFFICE LOOP	2021-01-07-12-04-03	13746	0.034	2.2	0.034	51.2
BUSINESS CAMPUS	2021-01-07-13-12-23	12023	0.064	1.3	0.064	24.3
NEIGHBORHOOD	2021-05-10-18-02-12	11674	0.036	1.6	0.034	43.4
	2021-05-10-18-32-32	10760	0.037	1.6	0.034	44.0
	2020-12-22-11-54-24	9775	0.057	2.8	0.056	40.1
PARKING GARAGE	2021-05-10-19-15-19	5257	0.069	1.0	0.063	35.9
	2020-12-22-12-04-35	7793	0.050	1.5	0.029	31.1
AVERAGE [1]			0.059	2.4	0.055	46.4

*This symbol means the algorithm failed to work. [1]The AVERAGE represents a sequence without trace failure.

run our method in an Intel(R) Core (TM) i7-10700 desktop computer with 16.0GB RAM. For accuracy evaluation, we have chosen the relative pose error (RPE) proposed in [36] as the indicator. Table 2 shows the results of our method compared with ORB-SLAM2. From the experiments, our results are pretty similar to those of ORB-SLAM2. For some sequences, our method is even better than it which is a complete visual SLAM system. In particular, ORB-SLAM2 does not work for the second sequence in OLD-TOWN, because the system is unable to track a sufficient number of features. In contrary, our method achieves stable tracking. This indicates that our system is robust.

Furthermore, we have also compared the memory consumption to highlight the lightness of our method. In Table 2, MEMORY indicates the memory usage during the algorithm running. We can observe that the minimum memory requirement for our method is approximately 24.0 MiB, while the minimum for ORB-SLAM2 is 1.0 GiB. For all sequences (excluding the untraceable one), our average memory usage is 46.4 MiB, which is an accuracy of 0.055 m at this point. Correspondingly, the comparison algorithm is 2.4 GiB, with an average accuracy of 0.059 m. We can claim that we achieved the approximate localization accuracy with a comparatively minor memory consumption.



(a)

(b)

(c)

(d)

Figure 11: The accuracy comparison of ORB-SLAM2 and ours. (a, b) The accuracy comparison in the roll, pitch, yaw, and x, y, z aspects. (c, d) The relative pose error and other metrics comparison.

In order to demonstrate the contrast qualitatively, we chose the packet 2021-05-10-19-15-19 from the PARKING GARAGE scenario to compare their trajectories and other various metrics separately. We present a summary of how our method compares to ORB-SLAM2 and real trajectories as shown in Figure 11. The top row indicates the comparison of our method with ORB-SLAM2 in the roll, pitch, and yaw dimensions and the x-, y-, and z-axes. The strengths of our method can be observed by comparing each of the six different perspectives. In terms of rotation error, the mistake of ORB-SLAM2 is larger than ours. The advantage of our approach is particularly evident in the yaw dimension. The comparison of the RPE is shown in the bottom row, where the left panel illustrates the absolute translation root-mean-square error (RMSE) of our method against ORB-SLAM2. The graph reveals that our error is smaller than that of the comparison method. In the right picture, a contrast of other metrics such as the mean is included. It can be noted that our results are approximately the same as ORB-SLAM2. As stated earlier, it is a complete SLAM system. In other words, we have achieved a comparable result to the full system for a lower cost, which shows the excellence of our method.

5. Conclusion

In this paper, we proposed a lightweight stereo visual odometry system based on the indirect methods for low-light scenes. The image decomposition is applied to our proposed system according to retinex theory. Specifically, we first utilized the thought of LIME to obtain the enhanced image of a low-light scene and only estimate the illumination image. This reduces the computational burden of the proposed system to a large extent. Then, we applied an efficient detection scheme to acquire the high-quality features, which significantly reduces the calculation time. Meanwhile, a coarse-to-fine process was employed to find out the best match in the points matching phase by sorting the descriptors according to their Hamming distance. In addition, an efficient local map for pose optimization was maintained to keep the tracking accuracy. Moreover, we defined an optimization function to minimize the reprojection error for pose estimation. Finally, the experiments using the 4Seasons datasets showed that our proposed approach is superior to the existing methods. It should be noted that we will apply the proposed method to intelligent vehicular networks in our future work [37–39].

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62173101 and 61905045, in part by the Open Research Project of Zheijiang Laboratory under Grant 2021KF0AB06, in part by the Guangzhou Science and Technology Project under Grants 202102010501 and 202102010416, in part by the International Collaborative Research Program of Guangdong Science and Technology Department under Grant 2020A0505100061, and in part by GZU-HKUST Joint Research Program under Grant YH202110.

References

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: a survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[2] H. Alismail, M. Kaess, B. Browning, and S. Lucey, "Direct visual odometry in low light using binary descriptors," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2017.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[4] E. H. Land, "The retinex theory of color vision," *Scientific American*, vol. 237, no. 6, pp. 108–128, 1977.

[5] P. Wenzel, R. Wang, N. Yang et al., "4Seasons: a cross-season dataset for multi-weather SLAM in autonomous driving," in *Proceedings of the DAGM German Conference on Pattern Recognition*, pp. 404–417, Tübingen, 2020.

[6] 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.

[7] M. Kim and M. G. Chung, "Recursively separated and weighted histogram equalization for brightness preservation and contrast enhancement," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1389–1397, 2008.

[8] D. J. Ketcham, R. W. Lowe, and J. W. Weber, *Image Enhancement Techniques for Cockpit Displays*, Defense Technical Information Center, Fort Belvoir, VA, USA, 1974.

[9] S. M. Pizer, E. P. Amburn, J. D. Austin et al., "Adaptive histogram equalization and its variations," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 355–368, 1987.

[10] S. Srinivasan and N. Balram, "Adaptive contrast enhancement using local region stretching," in *Proceedings of the 9th Asian Symposium on Information Display*, pp. 152–155, New Delhi, 2006.

[11] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi, and M. Shoyaib, "An adaptive gamma correction for image enhancement," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, p. 13, 2016.

[12] S.-C. Huang, F.-C. Cheng, and Y.-S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1032–1041, 2013.

[13] X. Guo, Y. Li, and H. Ling, "LIME: low-light image enhancement via illumination map estimation," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 982–993, 2017.

[14] X. Fu, D. Zeng, Y. Huang, X.-P. Zhang, and X. Ding, "A weighted variational model for simultaneous reflectance and illumination estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2782–2790, Las Vegas, 2016.

[15] C. Won, H. Seok, Z. Cui, M. Pollefeys, and J. Lim, "OmniSLAM: Omnidirectional localization and dense mapping for wide-baseline multi-camera systems," in *Proceedings of the 2020 IEEE International Conference on Robotics Automation*, pp. 559–566, Paris, 2020.

[16] Q. Fu, H. Yu, X. Wang et al., "Fast ORB-SLAM without keypoint descriptors," *IEEE Transactions on Image Processing*, vol. 31, pp. 1433–1446, 2022.

[17] S. Ji, Z. Qin, J. Shan, and M. Lu, "Panoramic SLAM from a multiple fisheye camera rig," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 169–183, 2020.

[18] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots Systems*, pp. 1861–1868, Macau, 2019.

[19] R. Wang, M. Schworer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3903–3911, Venice, 2017.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *Proceedings of the European Conference on Computer Vision*, pp. 404–417, Graz, 2006.

[22] S. N. Sinha, J.-M. Frahm, M. Pollefeys, and Y. Genc, "GPU-based video feature tracking and matching," in *Proceedings of the Workshop on Edge Computing Using New Commodity Architectures*, p. 4321, Chapel Hill, 2006.

[23] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision*, pp. 430–443, Graz, 2006.

[24] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, Manchester, 1988.

[25] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Proceedings of the European Conference on Computer Vision*, pp. 778–792, Heraklion, 2010.

[26] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, 2011.

[27] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[28] I. Suárez, G. Sfeir, J. M. Buenaposada, and L. Baumela, "BEBLID: boosted efficient binary local image descriptor," *Pattern Recognition Letters*, vol. 133, pp. 366–372, 2020.

[29] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do, "Fast global image smoothing based on weighted least squares," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5638–5653, 2014.

[30] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[31] I. Suárez, G. Sfeir, J. M. Buenaposada, and L. Baumela, "BELID: boosted efficient local image descriptor," in *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*, pp. 449–460, Madrid, 2019.

[32] M. Aladem and S. A. Rawashdeh, "Lightweight visual odometry for autonomous mobile robots," *Sensors*, vol. 18, no. 9, 2018.

[33] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: a general framework for graph optimization," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, pp. 3607–3613, Shanghai, 2011.

[34] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the Kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, Vilamoura-Algarve, 2012.

[37] L. Zhang, W. Zhou, J. Xia et al., "DQN-based mobile edge computing for smart Internet of vehicle," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 10, 2022.

[38] J. Lu, L. Chen, J. Xia et al., "Analytical offloading design for mobile edge computing-based smart internet of vehicle," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, no. 1, p. 10, 2022.

[39] L. Chen, R. Zhao, K. He, Z. Zhao, and L. Fan, "Intelligent ubiquitous computing for future UAV-enabled MEC network systems," *Cluster Computing*, vol. 25, no. 4, pp. 2417–2427, 2022.