Mechatronic Systems and Control, Vol. 53, No. 10, 2025
An Open Access Paper

MONOCULAR VISUAL SLAM BASED ON DEEP LEARNING FEATURE POINTS

Wenhao Huang,* Songyi Lu,* Yifan Liu,* Guoyin Zhang,* and Quande Yuan**

Abstract

In view of the traditional monocular visual odometer in visual changes, light changes, poor robustness, low pose calculation accuracy, the feature matching module in ORB-SLAM replaced with feature matching based on SuperPoint network, and feature tracking, local map, key frame recognition, loop detection, pose estimation. Comparing the improved algorithm with the traditional ORB and SIFT on the public dataset KITTI, the absolute trajectory error was somewhat reduced, indicating that the method of integrating deep learning feature points is significantly better than the traditional visual SLAM in accuracy.

Key Words

Visual odometer, deep learning, SuperPoint, feature point extraction

1. Introduction

Synchronous location and map building Simultaneous localisation and mapping (SLAM) [1]. It refers to the mobile robot in the unknown environment, relying on the sensing information, establish a map consistent with the surrounding environment, while realising autonomous positioning. Visual SLAM is generally composed of four modules: front end (visual odometer), back end (non-linear) optimisation, loop detection, and drawing construction [2], As shown in Fig. 1. Traditional visual SLAM methods mainly rely on manually designed point features for inter-image matching and tracking [3], To recover the camera local motion geometry and correct the trajectory by loop detection. Visual SLAM can provide real-time and accurate positioning information for autonomous vehicles, capture road information through cameras, and combine with SLAM algorithms, vehicles can build a map of the surrounding environment in real time and accurately determine their own location in the map. However, it is unstable in environments such as direct or dim sunlight, lack of features, and dynamics.

* School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin, China; e-mail: huangwenhao@jlict.edu.cn; lusongyi@jlict.edu.cn; liuyifan@jlict.edu.cn; zhangguoyin@jlict.edu.cn
** School of Computer Technology and engineering, Changchun Institute of Technology, Changchun, Jilin, China; e-mail: yuanqd@ccit.edu.cn
Corresponding author: Quande Yuan

Recommended by Teng Li
(DOI: 10.2316/J.2025.201-0559)

Visual SLAM can be divided into feature point method and direct method from the implementation method. Although the direct method estimates the pose of the camera by calculating the minimum photometric error, although it saves a lot of computational time consumed in feature point extraction and matching, and thus has a fast running speed, the method is easy to be disturbed by external conditions such as illumination due to the strong assumption that the gray value is unchanged as a precondition, and the robustness is weak. The feature point method refers to the measurement and acquisition of road markings with marking properties in the image, and then the camera pose is evaluated by matching the feature points between two adjacent frames. Image features are generally divided into points, lines, edges and other features. The most common point features are oriented FAST and rotated BRIEF (ORB) [4], scale-invariant feature transform (SIFT) [5], speeded-up robust features (SURF) [6],etc. It has good versatility and robustness, but the extraction effect is poor in complex scenes such as obvious lighting changes. With the continuous optimisation of the neural network structure, the deep learning algorithm has a good matching effect in complex environments such as different lighting conditions, and has better robustness than traditional algorithms by extracting semantic information for feature matching.

Given the advantages of the above deep learning algorithms, more and more scholars are applying deep learning to visual SLAM. In order to solve the problem of acquiring the image depth information by monocular camera, scholars have proposed the method of predicting the image depth value by neural network (such as GCN-SLAM [7], UnDeepVO [8], SfM –Learner [9], etc.), and by using deep learning to replace traditional SLAM modules (such as CNN-SVO [10], DP-SLAM [11] class).

In 2016, Detone *et al.* [12] designed the convolutional neural network HomographyNet to directly estimate the monography between images, compared with ORB, proved the flexibility of deep learning methods and the universality of applicable scenarios, and made excellent contributions to feature extraction. In 2018, Detone *et al* [13] proposed SuperPoint based on MagicPoint, which has the characteristics of strong real-time performance, multi-task collaboration, and lightweight network. The ability to extract features from images in real time makes it ideal for applications that require real-time processing, such as

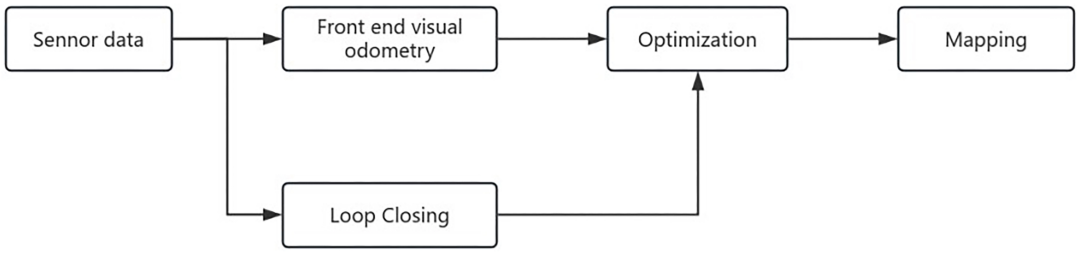

Figure 1. Visual SLAM framework diagram.

drones, robots, and autonomous driving. In addition, it can extract stable features in images with different lighting conditions, different rotation angles and different scales, so it has good robustness. SuperPoint can be used for both feature detection and descriptor extraction, which can share parameters in the encoder part, and can be calculated separately in the decoder part, which has a good synergy between the two tasks. SuperPoint's network structure is very lightweight, with only a few hundred thousand parameters, so it can be run on embedded devices. Due to its good real-time and robustness, it has been widely used in the field of computer vision.

In terms of feature point matching, the ratio of the nearest neighbor and the two feature points to be matched is generally calculated from the set threshold, and the random sampling consistency (random sample consensus [RANSAC]) algorithm is used [14], [15]. Removal of the mismatch points was performed.

This paper first describes the overall algorithm process, then introduces the relevant theoretical knowledge of the SuperPoint algorithm, then designs a method for the fusion of deep learning feature points and traditional visual SLAM, and finally conducts experimental comparative analysis.

2. The Overall Algorithm Process

One of the most critical steps of visual odometry based on feature point method is to extract and match features, and its main process is shown in Fig. 2, including feature extraction, feature matching, pose estimation, and local pose optimisation, mainly according to the relevant feature matching relationship on the image to obtain the camera motion estimation between adjacent frames, and then the reprojection error function is constructed according to the matching feature relationship obtained therefrom, and the relative motion of the camera is obtained by minimising the error at the same time. In fact, it can be understood that the feature point method is divided into several key steps: feature detection, feature matching, motion estimation and optimisation.

In this paper, the SuperPoint network is used to replace the traditional ORB feature extraction method, and the feature points of the image are extracted and their descriptors are calculated. The improved model has three core threads, namely feature tracking, local mapping, and loop detection, as shown in Fig. 3, and completes the scheduling of the three threads through the total thread of the system, which can realise the functions of map

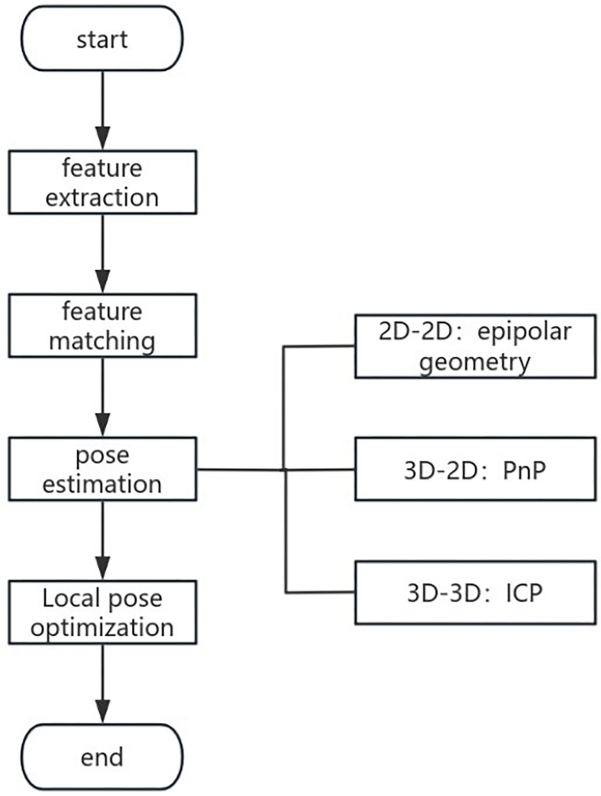

Figure 2. Flowchart of visual odometry based on feature point method.

reuse, loop detection, and repositioning, so as to realise the accurate positioning and mapping of the mobile robot.

The system framework is studied as follows:

(1) *Feature Tracking Thread:* The main function is to match and screen the map key frames, so as to improve the accuracy of the map construction. Using SuperPoint network to extract feature points from the current frame image, and in the current frame feature points match the previous frame feature points, introduce RANSAC algorithm to feature matching points error matching elimination processing, calculate the single stress matrix H and estimate the camera position, and then track the local map, the local map points and the current frame projection to get more match, optimise the position of the current frame attitude, and finally determine whether to insert a new key frame.

(2) *Local Map Building Thread:* This is shown in Fig. 4, the main function is to continue to optimise the map

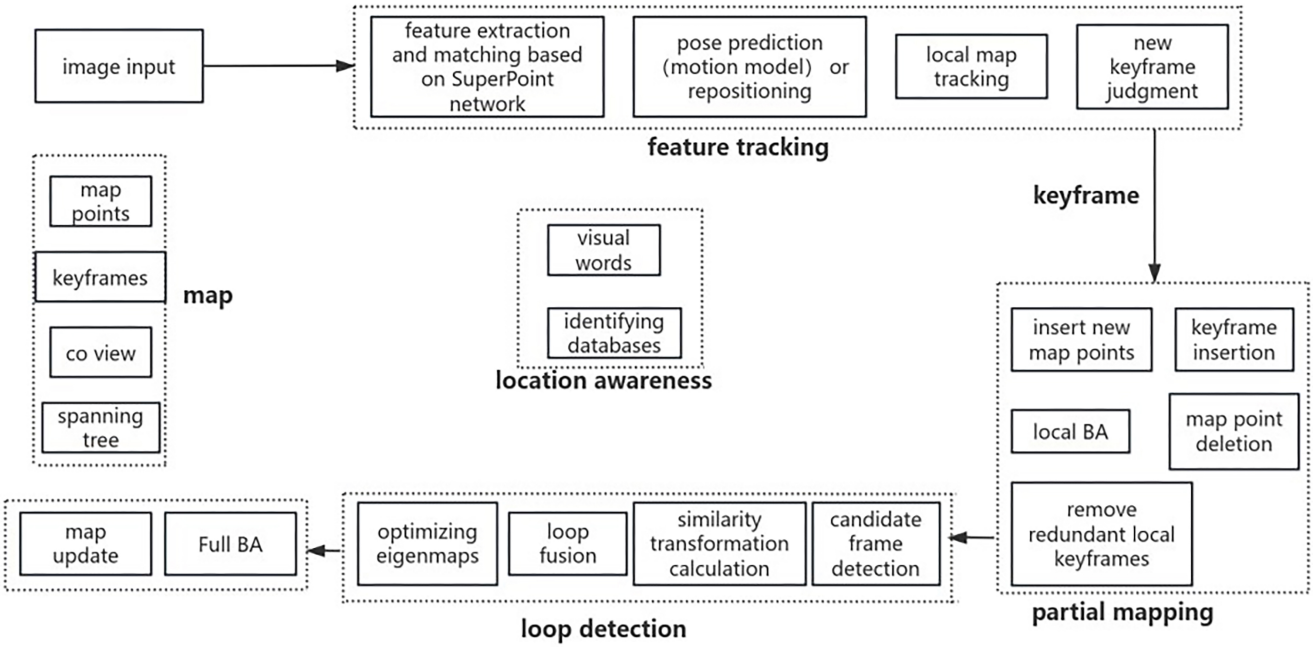

Figure 3. SuperPoint-SLAM system framework.

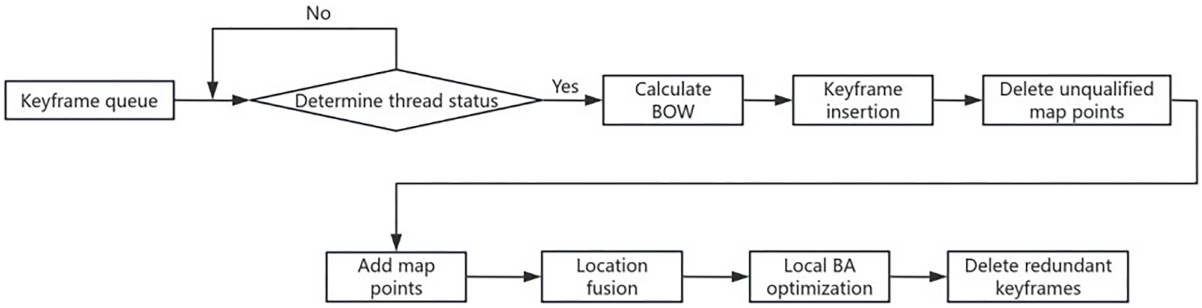

Figure 4. Drawing flow chart.

points and reduce the operation pressure of the return loop detection. After inserting a new keyframe, the local mapping thread first updates the common view and growth tree, calculates the word bag BOW, inserts it into the map, and then eliminates the redundant map points to avoid mismatching and wrong triangulation, and retain high-quality map points. At the same time, the feature points that are not matched in the new inserted frame are matched with the local map points, so that they meet the pole line constraints, and the forward depth test, disparity test, reprojection error test, and scale continuity test are conducted to create a new map point. Then, the map points and poses were optimised for local BA. Finally, the redundant key frames are removed to reduce the calculation pressure of the loop detection thread.

(3) *Closed-Loop Detection Thread:* The main function is to detect the closed-loop degree of the map points, and to optimise the closed-loop. First, the similarity between the current keyframe and the loop candidate frame is calculated, restricting the BOW vector similarity, the number of shared words, and the continuity. If there is a loop, match the common view of the current frame with the map points of the common view of the matching frame to establish a working relationship to fuse the duplicate map points. Then, the trajectory location was used to optimise it with this feature map. Finally, all keyframes and map points were optimised.

3. SuperPoint Algorithm Framework and Principle

The traditional feature point detection algorithm usually calculates the feature points and feature descriptors of the image separately, without the ability to share the calculation and output, while the SuperPoint used in this paper can complete the detection of feature points and the extraction of descriptors at the same time. SuperPoint is a self-supervised network model based on a fully convolutional network divided into three phases as shown in Fig. 5. First, the simple geometry is trained to enable the SuperPoint network to identify simple feature point information; second, the trained encoder and feature point decoder are used to extract the real image, the missing feature points are sampled, then the feature points are restored to the original image by reverse single stress transformation; finally, the real image and the single stress

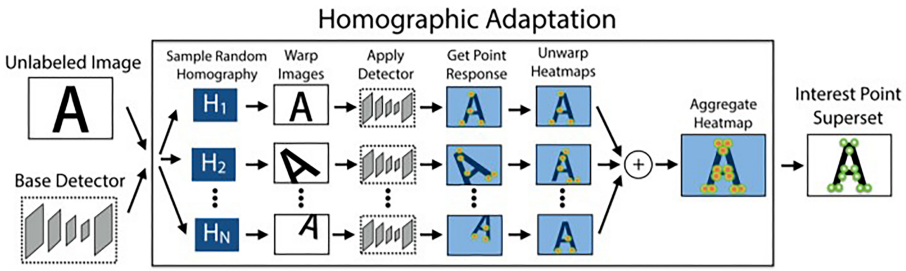

Figure 5. Adaptive monophonic transformation.

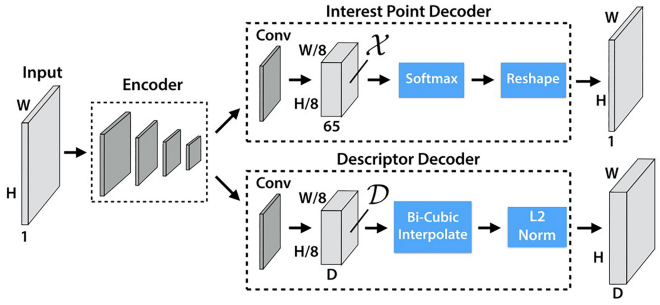

Figure 6. SuperPoint schematic sketch.

transformation image are input to the whole network, generating the position and descriptor.

The SuperPoint network is mainly composed of three parts: shared encoder, feature point decoder, and descriptor decoder, as shown in Fig. 6. The shared encoder is used to reduce the dimension of the image, and then the two decoders extract the image feature points and descriptors simultaneously according to the shared parameters of the encoder. Due to the parameter sharing of the feature points and descriptors, the computational amount is reduced and the computational efficiency is improved.

The output tensor of the VGG-style shared encoder is used as the input tensor of two decoders at the same time, the latter includes two branches, the feature decoder and the descriptor decoder, and the two decoders adopt different structures and learn different network parameters according to different tasks. Table 1 shows the specific network parameters, each row in the table has a convolutional channel, the first number is the input channel, the middle two numbers are the size of the convolution kernels, and the last number is the number of convolution kernels.

3.1 Shared Encoder

The encoder has a VGG-like structure with a total of eight convolutional layers of 3 × 3 size. The first four convolutional layers have 64 convolution kernels, and the last four convolutional layers have 128 convolution kernels, and the nonlinear activation function ReLU is connected behind each convolutional layer. After the second, fourth, and sixth activation functions, the maximum pooling of 2 × 2 is used to downsample the image. After eight convolutional layers and three maximum pooling layers, the size of the feature map decreases from W×H×1 to W/8×H/8×128 at the time of input, which reduces the dimension of the input image. This encoder structure strikes a good balance between computational efficiency and feature extraction capabilities, allowing SuperPoint to achieve better performance even with limited computing resources.

Table 1
SuperPoint Network Structure

Shared Encoder	Feature Point Decoder	Descriptive Sub Decoder
1×3×3×64	128×3×3×256	128×3×3×256
64×3×3×64	256×1×1×65	256×1×1×256
64×3×3×64	—	—
64×3×3×64	—	—
64×3×3×128	—	—
128×3×3×128	—	—
128×3×3×128	—	—
128×3×3×128	—	—

3.2 Feature Point Decoder

The input of the feature decoder is the output of the encoder, that is, W/8×H/8×128, which first passes through the CR module with the number of channels of 256, and then enters the convolutional layer with the number of channels of 65, and then outputs the tensor information of W/8×H/8×65. Due to the triple maximum pooling of the encoder, the information of one pixel corresponds to the pixel information of the non-overlapping 8 × 8 area size in the original image, and one channel is added for the case without feature point information, which is exactly 65 convolutional layer channels. The Softmax function is used to remove the channel without feature point information, and it becomes W/8×H/8×64. Finally, the size of the original image is restored by reshape, which is W×H×1, and the feature points are displayed on the original image. The value of each pixel in the output image represents the probability value of whether the corresponding pixel is a key point. Although the maximum pooling layer will reduce the resolution of the feature map, it does not actually reduce the amount of information of pixels, and

can also help the network learn more robust and abstract features, so as to improve the performance of detection and descriptor extraction.

3.3 Descriptor Decoder

The input of the descriptor decoder is also the output of the encoder, and the dimension of the descriptor is 256, which is used to compare the similarity between different feature points. After passing through the CR module with 256 channels, entering the convolutional layer with the same number of 256 channels, the output is a feature map of W/8×H/8×256, then the bicubic interpolation is used to change to W×H×256, and finally the L2 norm is used to normalise each descriptor. The output of this process is the descriptor for each key.

3.4 Loss Function

The loss function consists of two parts [16], as shown in (1), where L_p is the loss function of the decoder that extracts the key points, and L_d is the loss function of the decoder that generates the descriptor.

X and D are the feature map and descriptor sub-feature map of the image output by the network, Y is the label value of the image feature point, X', D', Y' is the label value of the feature point feature map, the descriptor sub-feature map and the image feature point output after the image is input into the network after the monography transformation, S is the feature point judgment matrix, and λ is the hyperparameter, which is used to balance the feature point detection loss and the descriptor loss.

$$L\left(X, X', D, D'; Y, Y', S\right) = L_p\left(X, Y\right) + L_p\left(X', Y'\right) + \lambda L_d\left(D, D', S\right). \quad (1)$$

The loss function of the key points is calculated using the cross-entropy loss:

$$L_p\left(X, Y\right) = \frac{1}{H_c W_c} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} l_p\left(x_{\mathrm{hw}}; y_{\mathrm{hw}}\right) \quad (2)$$

$$l_p\left(x_{\mathrm{hw}}; y\right) = -\log\left(\frac{e^{x_{\mathrm{hwy}}}}{\sum_{k=1}^{65} e^{x_{\mathrm{hwk}}}}\right). \quad (3)$$

H_c, W_c are the height and width of the key point features, respectively;

$x_{\mathrm{hw}} y_{\mathrm{hw}}$ are the values and label values of the image X in the (h, w) coordinates, respectively;

K is the number of channels;

$x_{h\mathrm{wk}}$ is the value of the image X at the (h, w) position of the kth channel.

The loss function of the descriptor is as follows:

$$L_d\left(D, D', S\right) = \frac{1}{\left(H_c W_c\right)^2} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} \sum_{\substack{h'=1 \\ w'=1}}^{H_c, W_c} l_d\left(d_{\mathrm{hw}}, d'_{h'w'}; s_{\mathrm{hw}h'w'}\right) \quad (4)$$

$$s_{\mathrm{hw}h'w'} = f\left(x\right) = \begin{cases} 1, \ if \left\| \widehat{HP_{\mathrm{hw}}} - P_{h'w'} \right\| \le 8 \\ 0, \qquad \text{otherwise} \end{cases}. \quad (5)$$

$$l_d\left(d, d'; s\right) = \lambda_d * s * \max\left(0, m_p - d^T d'\right) + (1 - s) * \max\left(0, d^T d' - m_n\right) \quad (6)$$

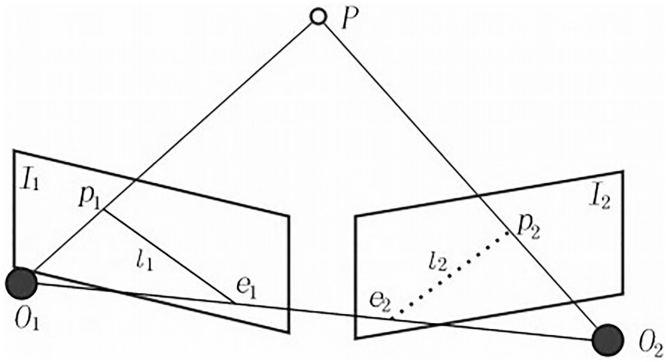

Figure 7. Polar geometric constraints.

$s_{\mathrm{hw}h'w'}$ is 0 or 1;

$d'_{h'w'} d_{\mathrm{hw}}$ are the values of the descriptor feature plots D' at (h', w') and D at (h, w);

p_{hw} is the central pixel location of the cell (h, w);

$m_p m_n$ are the thresholds corresponding to the forward and reverse directions, respectively;

λ_d is the hyperparameter that balances the internal positive and negative losses of the descriptor.

After the original image is downsampled, the points on the feature map correspond to an 8×8 cell in the original map, $s_{\mathrm{hw}h'w'}$ is used to determine whether the pixel position of the centre of the cell corresponding to the d_{hw} and $d'_{h'w'}$ is similar, "1" means that the position is similar, and "0" means that the position is opposite.

3.5 Camera Pose Estimation and Optimisation

In this paper, the pose estimation is the motion of the monocular camera, and the input information is 2D pixel coordinates. Since the motion estimation is based on two sets of 2D points, the pole constraint is used to solve the motion [17]. As shown in Fig. 7, I_1 and I_2 represent the imaging planes of the previous frame and the current frame image, O_1 and O_2 represent the camera centre, l_1 and l_2 are the polar lines of the feature point x_1 and x_2, and the intersection points of the connection line with the O_1 and O_2 are the pole e_1 and e_2. The geometric constraint equation for the poles is

$$\begin{cases} x_2^T F x_1 = 0 \\ x_2 = H x_1 \end{cases}. \quad (7)$$

F is the base matrix; H is the response matrix. When the feature points are matched correctly and the P points are not in the spatial plane, the normalised plane coordinates and the base matrix satisfy the (7). If the feature point cannot fall on the pole line due to the influence of mismatching, you need to calculate the distance from the x_1 and x_2 to the l_1 and l_2 of the polar line, respectively, and the point is the outer point when the distance is greater than the threshold. In order to ensure the calculation accuracy of the basic matrix,

the image feature points were filtered out according to the minimum distance threshold method, and then the error matching was further filtered out by the RANSAC algorithm.

The minimum distance threshold method refers to the distance test of the feature point pair in the image, and the nearest feature point pair is selected as the minimum distance. Equation (8) is used to judge the distance of the feature point matching pair, and when the condition is satisfied, it is judged to be a correct match, otherwise the matching pair is eliminated.

$$D_i < \alpha D_{\min}. \tag{8}$$

D_i indicates the ith matching pair;α is the set threshold;$D_{\min}$ is the minimum matching distance in the matching set.

The RANSAC algorithm was used to eliminate feature mismatching, and firstly, four groups of non-collinear matching points were randomly selected from the feature point set to calculate the corresponding homology matrix H. Then, determine whether the number of feature points in the current inner point set is greater than the number of optimal inner point sets, and if so, update the optimal inner point set and update the number of iterations. Finally, the number of iterations is judged according to the set threshold, and if it is greater, the update outside point is retained, otherwise it will continue to iterate to meet the requirements.

The ultimate goal of VO is to get a precise trajectory. Ideally, for a pair of matching pixels p_i^k, p_{i+1}^k, there must be $p_{i+1}^k = T_i p_i^k$. However, in real life, noise is unavoidable, and the error ξ must exist, and there is $\xi_i = p_{i+1}^k - T_i p_i^k$. Our goal is to minimise the reprojection error ξ_i, leading to the objective function:

$$\xi^* = arg\ \frac{1}{2}\sum_{i=1}^{k}\left\|p_{i+1}^j - T_i p_i^j\right\|_2^2, \tag{9}$$

where T_i is the pose transformation between the ith frame and the i+1st frame.

Therefore, for the camera pose T_i obtained from the single response matrix H and the basic matrix F, we choose the smaller value of the minimised reprojection error ξ^* as the camera pose.

4. The Experiment and the Results Were Analysed

Experiments quantitatively evaluated the entire monocular visual odometer system using the publicly available dataset KITTI and compared the trained SuperPoint model with the SIFT, ORB algorithms. The hardware platform of this experiment is Inter I7 12650H CPU and NVIDIA RTX4060, and the system version is Ubuntu20.04.

4.1 Experimental Dataset

The KITTI dataset was jointly founded by foreign authoritative institutions and has wide recognition and influence. It provides rich multimodal data, including image data, point cloud data, camera correction data, and label data, which provide a comprehensive training and testing environment. In addition, the KITTI dataset covers a variety of real-world driving scenarios such as urban, rural, and highway, and each image may contain up to 15 vehicles and 30 pedestrians, with various degrees of occlusion and truncation. This diversity helps to improve the performance and reliability of deep learning algorithms in visual SLAM.

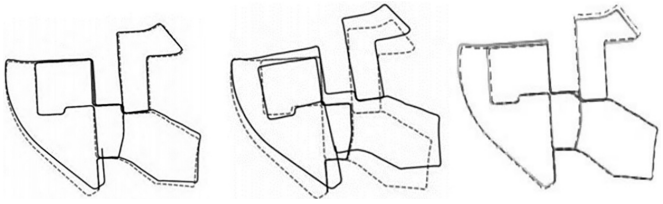

Figure 8. Comparison of the trajectories of the three algorithms: (a) ORB; (b) SIFT; and (c) The algorithm of this paper.

In this experiment, the visual SLAM highway data set of KITTI open outdoor scenes is used, and the data set consists of realistic scenes with the same perspective and the illumination changes in different scenes. The KITTI dataset contains images from the left and right perspectives. In order to ensure the comparison and evaluation of the experiment, Sequence with five sequences of 00-04 in the left perspective were selected as the validation dataset.

4.2 Track Error

The scale of monocular visual odometer is uncertain, and the back-end optimisation is not done in this paper, so there will be a cumulative error in the algorithm operation. In this experiment, the absolute trajectory error is used to evaluate the algorithm performance. The absolute trajectory error is calculated using the root mean square error, and the calculation formula is

$$s = \sqrt{\frac{\left(\sum_{i=1}^{n}\left\|s_i^e - s_i^t\right\|^2\right)}{n}}. \tag{10}$$

s_i^e represents the estimated coordinates of pose estimation based on the i-frame image, and s_i^t represents the true coordinate value of the i-frame image.

In the experiment, the difficult data path KITTI Sequence 00 was selected as the evaluation sample, and the ORB algorithm, SIFT algorithm and the visual odometer of this algorithm were used for the map trajectory comparison experiment. The evaluation tool was used to evaluate the absolute trajectory error of different algorithms. The experimental results are shown in Fig. 8. In Fig. 8, the dashed line represents the real map trajectory; the solid line represents the pose estimation trajectory obtained by each algorithm. From Fig. 8, it can be found that from the absolute error trajectory diagram, the SIFT effect is the worst, although the error trajectory of the ORB algorithm is small in the early stage, but with the increase of the number of frames, the trajectory error gradually

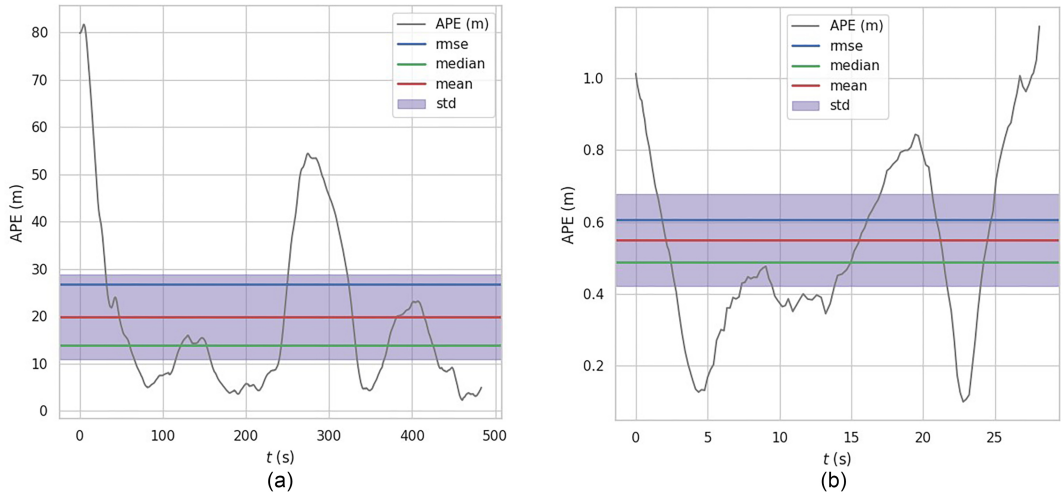

Figure 9. Trajectory error plot for part of the sequence 02 and 04.

Table 2
Track Error Comparisons Results Under the KITTI Dataset

	00	01	02	03	04	Mean
ORB	6.6535	494.0831	26.8237	1.1475	0.6071	105.8629
SIFT	28.7192	32.6023	35.6984	5.4582	1.2564	20.7469
SuperPoint	8.8623	30.7521	26.8647	0.7952	0.5894	13.5727

increases, and the fitting degree with the real trajectory decreases. The estimated trajectories obtained by the proposed algorithm are almost consistent with the real trajectories. This is due to the drift phenomenon caused by the accumulation of errors, and the larger the accumulated error, the more obvious the drift. The results show that the proposed algorithm has higher accuracy in pose estimation, is least affected by the cumulative error, and has better robustness.

Figure 9 shows the trajectory error diagram of the proposed algorithm on sequences 02 and 04, including APE, RMSE, Median, Mean, Std.

This experiment tested the KITTI Sequence 00-04 dataset, and the experimental results are shown in Table 2. Because the KITTI dataset data contains complex lighting changes and perspective change scenes, the Superpoint algorithm has good stability and robustness compared with the other two algorithms due to its unique feature matching method.

As can be seen from Table 2, compared with the traditional feature point matching algorithm (ORB, SIFT), the error of the algorithm on the five maps is somewhat reduced, and the average absolute trajectory error is reduced by 87.2% compared with that of ORB and 34.6% lower than that of SIFT. This paper improves the trajectory error accuracy and improves the robustness.

5. Conclusion

In this paper, self-supervised convolutional neural network SuperPoint is used to extract feature points and fuse them with visual odometer, then optimise the feature matching results according to the RANSAC optimisation algorithm, and finally make the camera pose estimation according to the polar geometry constraints. On the KITTI dataset, its absolute trajectory error decreased by 87.2% and 34.6% compared with the conventional methods ORB and SIFT, respectively. Deep learning feature points can express deeper image information, and SLAM fused with deep learning feature points has higher accuracy and robustness. Future work will start from the following two aspects:

(1) Due to the introduction of deep learning model, the complexity of the algorithm is higher than that of the visual odometer based on ORB and SIFT, the coding layer structure of the network can be adjusted, and the lightweight attention model is used to downsample the image features to reduce the computational cost and optimise the running speed of the algorithm.

(2) The bag of word model describes the whole image by determining which words are defined in the dictionary appear in an image, so as to transform all descriptors in an image into a vector, avoiding direct comparison of descriptors, and the vector represents the presence of features or not, which has nothing to do with the spatial position and arrangement order of the object. However, SuperPoint can extract the feature points and descriptors simultaneously, and the extracted descriptors are similar to the descriptors output by traditional algorithms, so it can also be used to construct the word bag model.

(3) Because the feature points in the moving region will greatly reduce the accuracy and robustness of

positioning, and it is necessary to retain the feature points in the static region as much as possible, it is necessary to introduce an object detection algorithm to detect dynamic targets to further reduce the error of pose estimation, improve the positioning accuracy, and improve the visual SLAM algorithm.

Acknowledgement

This work was supported by Jilin Province Science and Technology Development Plan Project (No.20210201049GX), National Natural Science Foundation of China (U22A2045).

References

[1] F. Debbat and L. Adouane, Formation control and role assignment of autonomous mobile robots in unstructured environment, *Mechatronic Systems and Control, 44*(2), 2016.

[2] C. Jianxian, G. Penggang, W. Yanxiong, and G. Zhitao, Mobile robot navigation using monocular visual-inertial fusion, *Mechatronic Systems and Control, 49*(1), 2021, 36–40.

[3] H. He, W. Xiang, and H. Liu, Autonomous navigation path planning of service robot based on multi-sensor fusion, *Mechatronic Systems and Control, 52*(2), 2024, 76–82.

[4] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, *60*(2), 2004, 91–110.

[5] H. Bay, T, Tuytelaars, and L.J.V. Gool, SURF: Speeded up robust features, *Computer Vision and Image Understanding*, *110*, 2008, 346–359.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, ORB: An efficient alternative to SIFT or surf, *International Conference on Computer Vision IEEE Computer Society*, Barcelona, 2011, 2564–2571.

[7] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, GCNv2: Efficient correspondence prediction for real-time SLAM, *IEEE Robotics and Automation Letters*, *4*(4), 2019, 3505–3512.

[8] R. Li, S. Wang, Z. Long, and D. Gu, UnDeepVO: Monocular visual odometry through unsupervised deep learning, in *Proceeding of the IEEE International Conference on Robotics and Automation*, Brisbane, QLD, 2017, 7286–7291.

[9] L. Zhang, G. Li, and T.H. Li, Temporal-aware SfM-learner: Unsupervised learning monocular depth and motion from stereo video clips, in *Proceeding of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Shenzhen, 2020, 253–258.

[10] S.Y. Loo, A.J. Amiri, S. Mashohor, S.H. Tang, and H. Zhang, CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction, in *Proceeding of the International Conference on Robotics and Automation (ICRA)*, Montreal, QC, 2019, 5218–5223.

[11] A. Li, J. Wang, M. Xu, and Z. Chen, DP-SLAM: A visual SLAM with moving probability towards dynamic environments, *Information Sciences*, *556*, 2021, 128–142.

[12] D. Detone, T. Malisiewicz, and A. Rabinovich Deep image homography estimation, 2016, *arXiv:1606. 03798v1*.

[13] D. Detone, T. Malisiewicz, and A. Rabinovich, Toward geometric deep SLAM, *ResearchGate*, 2017, to be published.

[14] P.H.S. Torrand A. Zisserman, MLESAC: A new robust estimator with application to estimating image geometry, *Computer Vision and Image Understanding*, *78*(1), 2000, 138–156.

[15] O Chum and J Matas, Optimal Randomized RANSAC, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *30*(8), 2008,1472–1482.

[16] D. DeTone, T. Malisiewicz, and A. Rabinovich, Toward geometric deep SLAM, 2017, *arXiv:1707.07410*.

[17] J. Huang, S. Yang, Z. Zhao, Y.K. Lai, and S.M. Hu, ClusterSLAM: A SLAM backend for simultaneous rigid body clustering and motion estimation, *International Conference on Computer Vision*, 2021, 87–101.

Biographies

Wenhao Huang was born in Nantong, Jiangsu, China. He received the bachelor's degree in engineering from Nanjing University of Technology in June 2021. He is currently pursuing the master's degree in electronic information engineering with Jilin Institute of Chemical Technology. His current research interest is robotic SLAM.

Quande Yuan was born in Heze, Shandong, China. He received the Ph.D. degree in computer application technology from Harbin Institute of Technology. He is a Professor with Changchun Institute of Technology. His research interests include robotics and artificial intelligence.

Songyi Lu was born in Wuhan, Hubei, China. He received the bachelor's degree in engineering from Wenzheng College, Soochow University in June 2022. He is currently pursuing the master's degree in electronic information engineering with Jilin Institute of Chemical Technology. His major research direction is Laser SLAM.

Yifan Liu was born in Qingyang, Gansu, China. He received the master's degree in electronic information engineering from Jilin Institute of Chemical Technology in June 2024. Since July 2024, he has been an Engineer with Huali Integrated Circuit Manufacturing Co., Ltd. His research focusses mainly on visual SLAM and deep learning.

Guoyin Zhang was born in Datong, Shanxi, China. He received the master's degree in electronic information engineering from Jilin Institute of Chemical Technology in June 2024. Since July 2024, he has been an Engineer with China National Heavy Duty Truck Manufacturing Company. His research focuses mainly on robot positioning and navigation.