Dual-stream VO: Visual Odometry Based on LSTM Dual-Stream Convolutional Neural Network

Yuan Luo, YongChao Zeng, RunZhe Lv, WenHao Wang

Abstract—**This paper studies the problem of visual odometry based on a deep recurrent convolutional neural network. A new visual odometry algorithm based on dual-stream convolutional neural networks with long short-term memory is proposed. The color stream of the convolutional neural network acquires the color features in the RGB image. The depth stream acquires the contour features in the depth image, generates fusion features through the feature fusion unit, and finally predicts the pose at the current moment through autonomous sequential modeling using recurrent neural networks. Experimental validation on the TUM dataset showed that the method introduces contour features into the system through a dual-stream architecture of neural networks, which provides higher accuracy and robustness compared to other convolutional neural network-based visual odometry systems, especially in the presence of motion blur and poor lighting.**

Index Terms—**mobile robot, position estimation, visual odometry, LSTM**

I. Introduction

IN the 5G era, robotics has been rapidly developed and is widely used in military, medical, service, aerospace and other fields [1]-[3]. Vision odometry is the core technology of robotics and is divided into two main categories: geometry-based and deep learning-based. There are three implementation methods for geometry-based visual odometry. The first is the direct method [4]-[6], which builds and optimizes a photometric error function to estimate the interframe motion based on the assumption of a constant grayscale. The second is the feature point method [7]-[9], which builds and optimizes the reprojection error function to estimate the interframe motion by extracting and matching the feature points of neighboring frames. The third is the fusion of the direct method and the feature point method, also known as the semidirect method [10], which extracts and matches the features of neighboring frames and builds and optimizes the photometric error function for the feature points to estimate the interframe motion.

Deep learning has been a major study topic in recent years and has taken the lead in computer vision research. However, most existing deep learning architectures are applied to object recognition and classification, and relatively little research has been done to accomplish pose estimation using deep learning. Compared to geometric visual odometry, deep-learning-based visual odometry does not require processes such as storing keyframes, matching features between frames, and global optimization. When exploring and building large-scale maps, the computation and storage of neural networks do not increase. This solves the problem of high computational pressure and storage pressure when geometry-based visual odometry is applied to large-scale maps. In addition, we introduce contour features from the depth image frames into the system through a dual-stream convolutional neural network (CNN) architecture, which gives the system better performance in challenging environments.

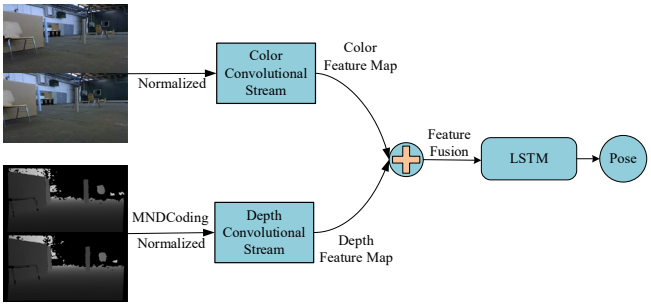


Fig. 1. Flow-chart of our proposed visual odometry. Color image sequences and depth image sequences are preprocessed and input to corresponding convolutional neural network stream to extract features for LSTM's sequential modeling to estimate pose. Image sequences are obtained from TUM dataset.

We propose new pose estimation methods based on dual-stream CNN with long short-term memory (LSTM). A flow-chart is shown in Figure 1. To extract contour features, the depth image is preprocessed and sent into a separate convolutional stream. The depth stream's contour feature maps and the color stream's color feature maps are created using the contour features of the depth stream as a compliment. This data is combined and sent into an LSTM for sequential modeling to estimate the current moment pose. This system has high accuracy and robustness. The main contributions of this paper are as follows. First, a new feature extraction method based on a dual-stream CNN is proposed to

Manuscript received November 4, 2021; revised April 11, 2022. This work was funded by the National Natural Science Foundation for Young Scholars of China (Grant No. 61803058).

Yuan Luo is a Professor at the Key Laboratory of Optical Information Sensing and Technology, School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: luoyuan@cqupt.edu.cn).

Yongchao Zeng is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: 1069995310@qq.com).

Runzhe Lv is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: 1109096063@qq.com).

Wenhao Wang is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: 1547889468@qq.com).

improve the accuracy and robustness of the system with regard to the color features of the color stream. Second, LSTM is used to autonomously model the anterior-posterior correlation and motion model of the image sequence. This effectively improves the accuracy of the system's positional estimation. Third, a dual-stream CNN and LSTM recurrent neural network are combined into a new pose estimation system with generalization capability to unknown environments.

II. Related Work

A. Geometry-Based Pose Estimation Methods

In 2007, Klein et al. proposed the parallel tracking and mapping (PTAM) system [11], which was the first simultaneous localization and mapping (SLAM) system to run localization and mapping in parallel and was the design standard for subsequent SLAM systems. PTAM introduces a keyframe mechanism in visual odometry. This requires only processing and storing keyframes from which keyframes are used for positional estimation to optimize the estimated path and map.

In 2014, J. Engle et al. proposed the Large Scale Direct Monocular SLAM (LSD-SLAM) system [12], which was the first SLAM system that used the direct method. LSD-SLAM acquires five points at equal distances on the polar line, measures its sum of squared distance (SSD) without computing feature points, and constructs a semidense map based on the assumption of constant grayscale.

In 2017, R. Mur-Artal et al. proposed the ORB-SLAM2 system [13], which is an upgraded version of ORB-SLAM and supports systems with RGB-D and binocular cameras. This system uses the same architecture as PTAM and is a relatively easy-to-use and well-designed modern SLAM system. This team then proposed ORB-SLAM3 [14] to improve the accuracy and robustness of the entire system by means of multisensor fusion.

B. Deep Learning Based Pose Estimation Methods

In 2015, Alex et al. proposed PoseNet [15] as a typical representative of early supervised learning methods. PoseNet batches the acquired images by structure from motion (SFM) and calculates the corresponding poses as the labels of the dataset. Then the network structure and parameters are designed based on GoogLeNet, and a regression model of 6-DOF poses is built by training. PoseNet obtains the estimated pose with high accuracy and without the support of a large number of labeled datasets by the migration learning method.

In 2017, DeepVO [16] proposed by Sen et al. directly mapped its corresponding pose from the original image sequence. It is able to learn not only the features of images by convolutional neural networks but also the dynamic relationships and intrinsic connections between images implicitly by recurrent neural networks.

In 2019, Almalioglu et al. proposed a generative adversarial-network-based model to learn image features by unsupervised learning to obtain a monocular VO system called GANVO [17]. The model does not require a large amount of calibration data compared to supervised learning and exhibited better performance than most traditional methods at that time.

III. Framework of Algorithm

Our proposed pose estimation system consists of a dual-stream convolutional neural network, feature fusion unit, and LSTM. The architecture of our proposed dual-stream VO system is shown in Figure 2. The dual-stream convolutional neural network consists of a color stream and a depth stream. The color stream extracts the color features in RGB image frames, and the depth stream extracts the contour features in depth image frames. The extracted color features and contour features are input to the feature fusion unit for feature fusion. Finally, the fused features are input to the LSTM for sequential modeling to estimate the pose at the current moment.

A. Feature Extraction Based on Dual-Stream CNN

Artificial intelligence and neural networks are the current research hotspots. Most existing convolutional neural network architectures are designed for object recognition, classification and tracking, so the features extracted by their convolutional neural networks are prepared for subsequent object recognition, classification, etc. To extract effective features applicable to visual odometry, we designed a color stream based on the VGGNet [18] architecture to extract color features of RGB image frames using the original RGB

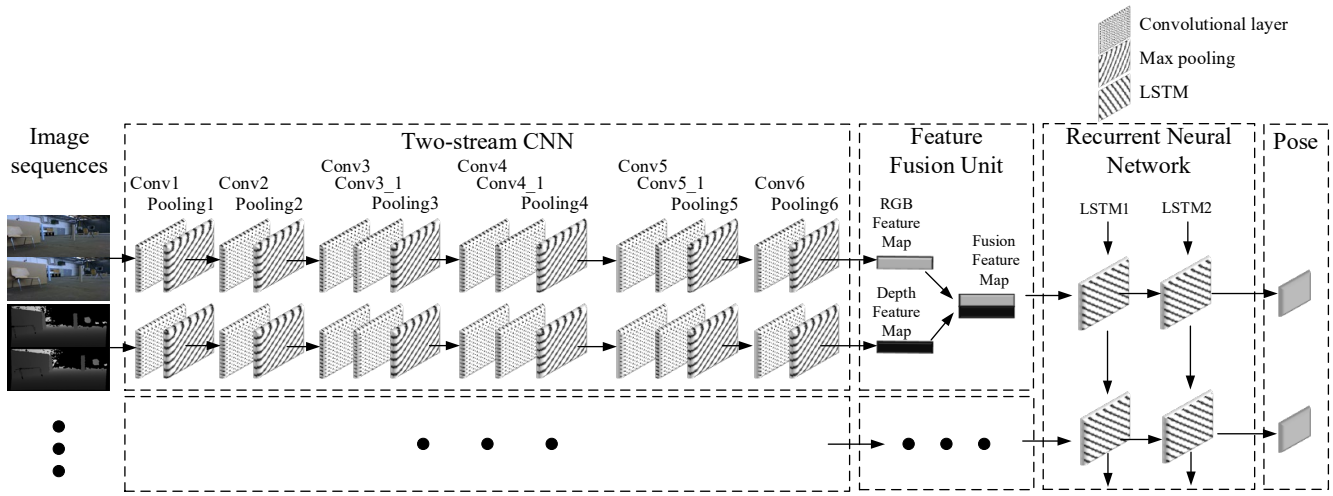


Fig. 2. Architecture of proposed dual-stream VO system. CNN should vary according to size of input image. RGB and depth images are from TUM dataset.

image frames as input. We also designed the depth stream with the same structure and configuration as the color stream. The contour features of the depth image frames are introduced into the pose estimation system through the dual-stream neural network architecture as a complement to the color features, which improves the accuracy and robustness of the system. Since visual odometry should have the ability to generalize in an unknown environment, the effective features of visual odometry should be based on several combinations of information about the object rather than relying on environmental information.

We designed a dual-stream CNN with the same structure and configuration for the color and depth streams. The detailed configuration of each convolutional and pooling layer is shown in Table I. The individual convolutional stream consists of nine convolutional layers and six maximum pooling layers. Each layer except Conv6 is followed by ReLU linear rectification function activation. To fully extract features at various levels in the image, the size of the convolution kernel of the method is reduced from 7×7 to 5×5 and finally to 3×3. Since this network is trained to learn effective features applicable to visual odometry, preprocessed completed color and depth images are processed by this network to compress high-dimensional image information into compact feature maps. The feature maps output from the two convolutional streams are fused by a feature fusion unit to generate fused features. The fused features are the basis for the subsequent LSTM for sequential modeling to estimate the pose.

TABLE I
CONFIGURATION OF EACH CONVOLUTIONAL AND POOLING LAYER

Layer	Kernel	Padding	Stride	Cannels
Convl	7×7	3	1	64
Pooling1	2×2	0	2	64
Conv2	5×5	2	1	128
Pooling2	2×2	0	2	128
Conv3	5×5	2	1	256
Conv3_1	3×3	0	1	256
Pooling3	2×2	0	2	256
Conv4	3×3	0	1	512
Conv4_1	3×3	0	1	512
Pooling4	2×2	0	2	512
Conv5	3×3	0	1	512
Conv5_1	3×3	0	1	512
Pooling5	2×2	0	2	512
Conv6	3×3	0	1	1024
Pooling6	2×2	0	2	1024

B. Sequential Modeling Based on LSTM

Most existing neural network architectures are designed for object recognition, classification and tracking. They generally use fully connected layers to generate probabilities of possible objects after feature extraction to recognize objects. However, this architecture does not apply to the pose estimation problem. We employ recurrent neural networks for autonomous sequential modeling of the motion relationship between image frames to estimate the pose information after feature extraction and fusion.

A recurrent neural network (RNN) processes sequential data. It differs from a convolutional neural network in its ability to process sequentially varying data. For example, the meaning of a word may be different depending on what is mentioned above, so to know the meaning of the word in context, you need to relate it to the previous context. The mathematical expression of RNN is as follows:

$$h_i = \mathcal{H}(w_{xh}x_i + w_{hh}h_{i-1} + b_i) \tag{1}$$

$$y_i = w_{hy}h_i + b_y \tag{2}$$

The above equation, h_i represents the state of the hidden layer at moment i, $\mathcal{H}$ is the activation function, w_{xh} is the weight coefficient of the input, w_{hh} is the weight coefficient of the previous hidden state, b_i is the bias vector, y_i is the output at moment i, w_{hy} is the weight coefficient of the hidden state and b_y is the bias vector. From the formula, it can be observed that RNN is theoretically capable of preserving the relationship between all items in the sequence. However, experiments have demonstrated that short-term memory has a large effect on the RNN, but long-term memory has little effect, which is the classical short-term memory problem of the RNN. To solve this problem, improved algorithms for RNNs have been derived. One of these algorithms is the LSTM [19].

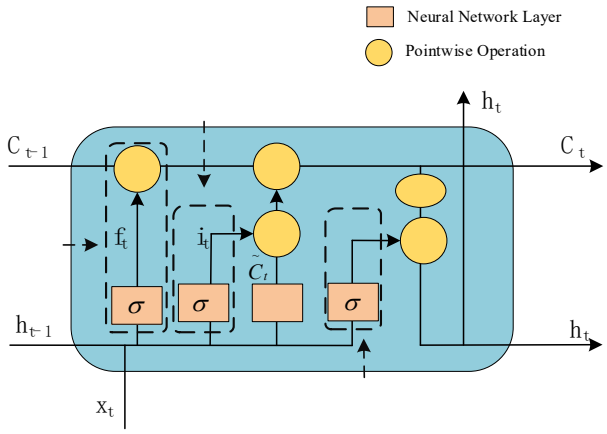


Fig. 3. Architecture of LSTM neural network unit. Source of data for LSTM neural network is fused feature maps output by feature fusion unit.

The long short-term memory neural network is an advancement of the recurrent neural network based on the gate circuit concept, which was originally designed to solve the problem of motion model and image sequence. Visual odometry, as a classical image sequence problem, fits very well with LSTM. The pose estimation network was built using two stacked LSTM layers, with the output of the previous LSTM layer as the input to the following LSTM layer and each LSTM layer having 1000 hidden states. After receiving the fused feature maps from the feature fusion unit, the pose information at the current moment is estimated and

output by the pose estimation model built by the LSTM. As the image sequence changes, there is pose information output at each moment. The architecture of the LSTM neural network unit is shown in Figure 3.

The LSTM has a module cell state similar to the RNN hidden layer, denoted by C_t, for storing previous and present information of high relevance. The previous cell state C_{t-1} and the current input signal x_t are processed together by LSTM to generate the current cell state C_t. LSTM design uses the idea of "gates" to introduce or remove information from cell states C_t. A "gate" is a method of processing information that allows signals to pass, not pass, partially pass or pass after being processed by the "gate." LSTM uses this principle to obtain the key information in a sequence and forget the less relevant information to achieve the purpose of long-term memory. An LSTM cell consists of a forget gate, input gate and output gate. The role of the forgetting gate is to eliminate information that is not very relevant in the cell state C_t at the current moment. The role of the input gate is to decide which useful information is added to the cell state C_t at the current moment from the new input x_t and the output h_{t-1} of the previous moment. The role of the output gate is to generate the current moment's output signal h_t by integrating the previous moment's cell state C_{t-1} after being processed by the forgetting gate and the input gate, the previous moment's output signal h_{t-1} and the current moment's input signal x_t. The following mathematical formulas are used to express the process of processing a signal by an LSTM layer:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6}$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

σ is the sigmoid activation function, W_f, W_i, W_o are the weight parameters, h_{t-1} is the output signal at the previous moment, x_t is the input at the current moment, b_f, b_i, b_o are the bias parameters, and tanh is the hyperbolic tangent activation function.

C. Feature Fusion

To improve the accuracy of visual odometry in challenging environments such as motion blur and poor lighting, we trained a separate convolutional stream to extract contour features from the depth images. Compared to RGB cameras, depth cameras are less affected in challenging environments. As a result, the contour features extracted from depth image frames have higher reliability in relatively complex environments. To a certain extent, the contour features are complementary to the color features extracted from RGB image frames. We improve the accuracy and robustness of the system in challenging environments by fusing colored and contour features to generate fused features for input into the LSTM for sequence modeling and estimating the current moment pose.

According to the sequence of fusion and prediction, feature fusion approaches may be characterized as early fusion [20-21] or late fusion [22]. Early fusion involves fusing the features of multiple layers and then training the predictor on the fused features. This type of method has "concat" and "add" operations. Late fusion is the improvement of detection performance by combining detection results from different layers. We use the "concat" feature fusion method to generate a fused feature map group by combining colored features and contour features, in preparation for the subsequent LSTM sequence modeling.

Since the structure and configuration of the color convolutional stream and the depth convolutional stream are identical, the output feature map after Pooling6 has the same structure and dimensionality. We use $X^{rgb} = [x_1, x_2, ..., x_i]$ to denote the color stream input sequence that has been preprocessed and $Y^{depth} = [y_1, y_2, ..., y_i]$ to denote the depth stream input sequence that has been preprocessed. The features extracted by the convolutional neural network are $\hat{X}^{rgb} = \left[\hat{x}_1, \hat{x}_2, ..., \hat{x}_i\right]$ and $\hat{Y}^{depth} = \left[\hat{y}_1, \hat{y}_2, ..., \hat{y}_i\right]$. We generate fused features by forming a feature map group Z_i^{fusion} from i moment color features $\hat{x}_i$ and i moment depth features $\hat{y}_i$. Considering that depth contour features have higher reliability under motion blur, poor lighting, etc., we let the depth contour feature profiles and the weight coefficient matrix be multiplied when fusing color features and depth contour features to increase the influence of depth contour features on the pose estimation network, and the weight coefficient matrix λ_i is updated automatically by network back-propagation. The expression of the formula for the fusion feature is shown below.

$$Z_i^{fusion} = \left[\hat{x}_i; \lambda_i \hat{y}_i\right] \tag{9}$$

Finally, the generated groups of fused feature maps are transferred to the LSTM for sequence modeling and estimation of the temporal pose.

D. Loss Function

In contrast to neural networks for target detection and classification purposes, we modeled the image sequences by LSTM recurrent neural networks to estimate the current moment pose instead of using a softmax classifier. The loss function for the network is the Euclidean loss of the pose. The pose information includes the camera position information p and orientation information q, so the mathematical expression of the Euclidean loss E of the pose is shown below.

$$E = E_p + W_e E_q \tag{10}$$

E_p is the positional Euclidean loss, E_q is the directional Euclidean loss and W_e is the equilibrium parameter of both. The Euclidean loss function is expressed in terms of the Euclidean distance between the true and estimated poses, as shown below.

$$E = \arg\min \frac{1}{N} \sum_{i=1}^{N} \sum_{i=1}^{t} \left\| \tilde{P}_t - P_t \right\|_2^2 + W_e \left\| \tilde{q}_t - q_t \right\|_2^2 \quad (11)$$

$\tilde{p}$ is the estimated position information, $\tilde{q}$ is the estimated direction information, p is the true position information, q is the true direction information and N is the training sample capacity.

IV. Experimental Evaluation

A. Experimental Platform

To evaluate the performance of our proposed LSTM dual-stream convolutional-neural-network-based visual odometry system, we implement the network structure based on the TensorFlow framework and train the pose estimation model on a CPU model Intel Xeon E5-2699 v4 with 2.2 GHz and GPU model GTX1080Ti server. To prevent overfitting, we also used the Adam optimizer to train the network, where the default values of 0.9 and 0.999 were used for both parameters β_1 and β_2, and the initial learning rate was set to 0.0002. The experiments were conducted using a laptop computer with an Intel Core i7-10875H CPU at 2.3 GHz, NVIDIA GeForce RTX2060 GPU, and Ubuntu 16.04.

B. Preprocessing of Image Frames

Our proposed LSTM-based dual-stream CNN consists of color images and depth images acquired by RGB-D cameras as inputs. Color images are input to the color stream of the dual-stream CNN to extract color features from the original RGB images. Therefore, color images do not need to undergo preprocessing such as optical flow, grayscale and binarization. However, to extract features that are more convenient for LSTM to build effective motion models and estimate the correct positional information, we need to cascade the adjacent two frames of the color image to form a deep RCNN tensor. The CNN we designed is based on a modified VGGNet architecture and therefore requires a fixed input image size of 224×224. The original image captured from Kinect One is 960×540, and the original image captured from Kinect 360 or Xtion is 640×480. In the system we designed, the original image is cascaded to 640×960 and then randomly cropped to a size of 224×224. We also tried adjusting the original image directly to 224×224 but found that the pose estimation performance was not as good as expected.

In our proposed dual-stream convolutional neural network, depth images from RGB-D cameras are introduced into a separate convolutional stream to learn contour features, thus enhancing system performance. The convolutional layer of a CNN, on the other hand, is intended exclusively for color image frames, where the pixel channels primarily convey light intensity. The pixel value of a depth image frame indicates the scaled distance between the camera's optical center and objects in the surroundings. It is difficult to extract effective contour features by directly using the original depth image as the input to the convolutional neural network. Therefore, the necessary preprocessing should be performed before feeding the depth image into the CNN. We used the minimized normal + depth (MND) method [23] proposed by Ruihao Li et al. to recover the depth image to a three-channel image. The results of processing depth images by MND are shown in Figure 4. It can be seen from the images that the image quality is significantly improved after MND encoding. The depth images also need to be cascaded adjacent to the MND encoding is completed, and then randomly cropped to a size of 224×224.

C. Training and Testing

We designed comparison experiments with the DeepVO and PoseNet algorithms to verify the effectiveness of our proposed method. The experiments were performed on the

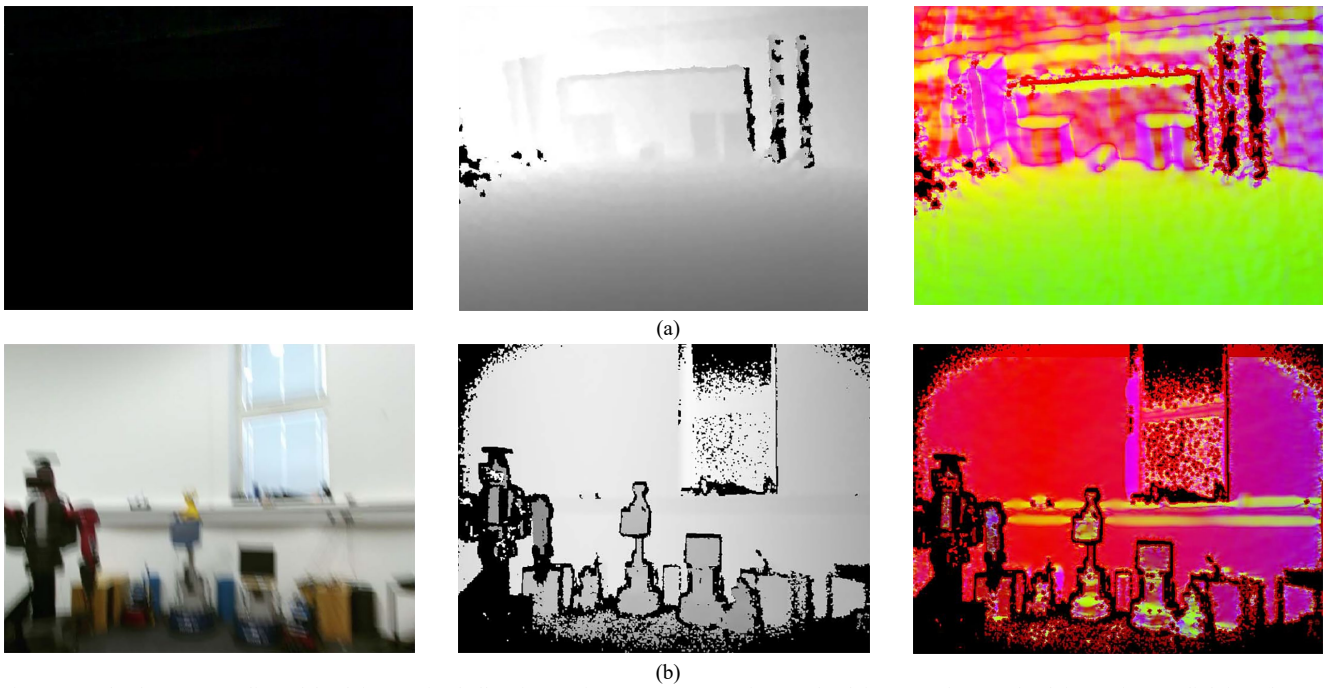


Fig. 4. Result of MND encoding of depth images in challenging environment. (a) RGB image, depth image and MND depth image in poor lighting. (b) RGB image, depth image and MND depth image with motion blur.

TABLE II
RESULTS OF TEST SEQUENCES

Seq.	Translational RMSE Drift(%)			Rotational RMSE Drift(°/100m)		
	DeepVO	PoseNet	Dual-stramVO	DeepVO	PoseNet	Dual-stramVO
r2/pioneer_360	7.421	**6.014**	6.218	5.962	**2.524**	2.536
fr2/pioneer_slam	5.267	4.234	**1.652**	2.876	0.739	**0.476**
fr2/pioneer_slam2	5.426	2.355	**1.437**	3.821	0.687	**0.411**
fr3/walking_rpy	22.376	19.691	**14.663**	9.534	7.681	**4.332**
fr3/walking_static	9.989	9.261	**8.951**	2.337	1.881	**1.648**
fr3/walking_xyz	20.115	18.793	**14.245**	7.452	6.548	**5.583**

Translational RMSE drift represents average RMSE drift error over corresponding sequence length. Rotational RMSE drift represents average rotational drift error per hundred meters on corresponding sequence.

TUM RGB-D dataset provided by the Technical University of Munich. Each sequence contains RGB images, depth images and real trajectories. The dataset is widely used to test various SLAM systems and VO algorithms. First, our proposed methods, DeepVO and PoseNet were used to train the model on the TUM testing and debugging sequence classes. The TUM testing and debugging sequence class contains fr1/xyz, fr1/rpy, fr2/xyz and fr2/rpy sequences with a total of 8473 samples. DeepVO and PoseNet train the model using only RGB images of the above sequence. Our proposed method, DeepVO and PoseNet were then tested for performance on the indoor TUM Robot SLAM sequence class and the challenging environment TUM Dynamic Objects sequence class, respectively. The TUM Robot SLAM sequence class uses the fr2/pioneer_360, fr2/pioneer_slam and fr2/pioneer_slam2 sequences, and the TUM Dynamic Objects sequence class uses the fr3/walking_rpy, fr3/walking_static and fr3/walking_xyz sequences.

D. *Evaluation Index*

We used the standard evaluation methods of relative pose error (RPE) [24] and absolute trajectory error (ATE) [24] for visual odometry to evaluate our proposed dual-stream visual odometer.

The relative pose error is a direct expression of the accuracy of the visual odometer by comparing the estimated pose and the real pose with a fixed time pose change. Using RPE for our proposed method, DeepVO and PoseNet on fr2/pioneer_360, fr2/pioneer_slam, fr2/pioneer_slam2, fr3/walking_rpy, fr3/walking_static and fr3/walking_xyz sequences were used for experiments. The experimental results are shown in Table Ⅱ. After analyzing the experimental data, it is observed that our proposed method improves performance by 64% in static environments and 28% in challenging environments compared to DeepVO. Relative to PoseNet, our proposed method improves performance by 30% in static environments and 20% in challenging environments.

To compare the performance differences between our proposed method, PoseNet and DeepVO, we used graphs to present the relative trajectory errors between the three methods. The experimental results on fr3/walking_rpy are shown in Figure 5, fr3/walking_static in Figure 6, and fr3/walking_xyz in Figure 7. We can see that our proposed method has the smallest relative trajectory drift and a better pose estimation effect.

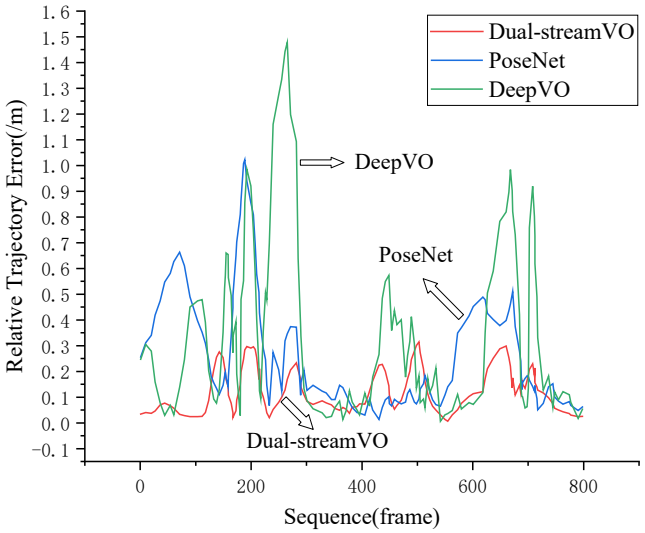


Fig. 5. Results of trajectory drift comparison for fr3/walking_rpy.

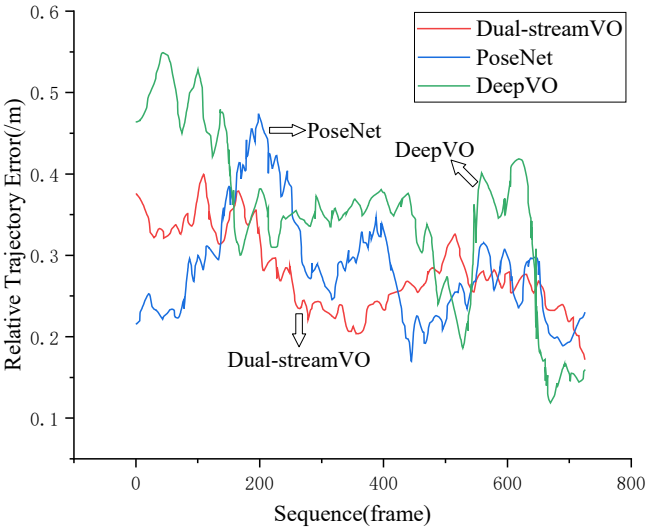


Fig. 6. Results of trajectory drift comparison for fr3/walking_static.

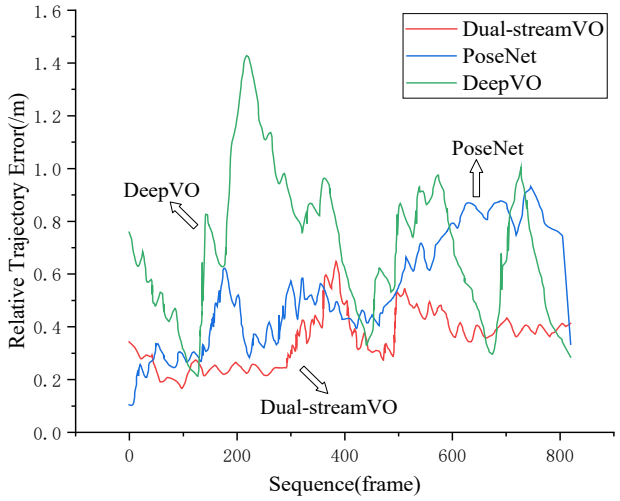


Fig. 7. Results of trajectory drift comparison for fr3/walking_xyz.

The absolute trajectory error is the difference between the estimated and true positional information and may be used to visualize the algorithm's correctness and the trajectory's overall consistency. We have evaluated our proposed method using the ATE evaluation method. The experimental results on fr3/walking_rpy are shown in Figure 8, fr3/walking_static in Figure 9, and fr3/walking_xyz in Figure 10. In the plot of the experimental results, we record in a two-dimensional coordinate system the paths estimated by our proposed method (marked as estimated), the true paths of the sequence (marked as ground truth), and the difference between the estimated paths and the true paths (marked as difference). To evaluate the performance of our proposed method more comprehensively, we also used the ATE evaluation method for DeepVO and PoseNet on the sequences fr3/walking_rpy, fr3/walking_static, and fr3/walking_xyz, and the experimental results are shown in Figure 11. By comparing the ATE experimental result plots of our proposed method, DeepVO, and PoseNet, it is clear that our proposed method has significantly fewer errors in challenging environments.

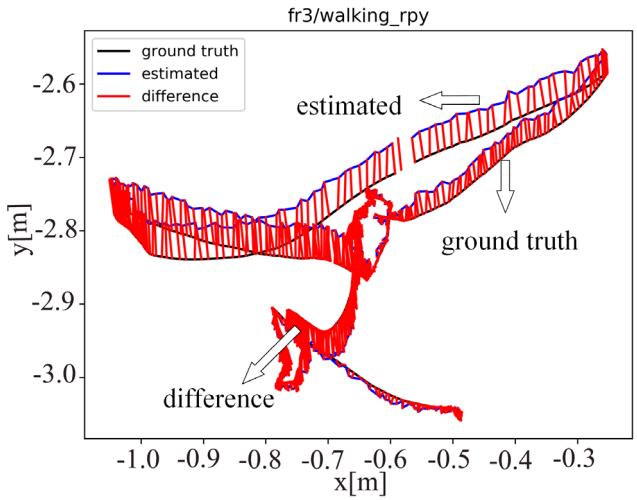


Fig. 8. ATE evaluation plot of our proposed method on sequences fr3/walking_rpy.

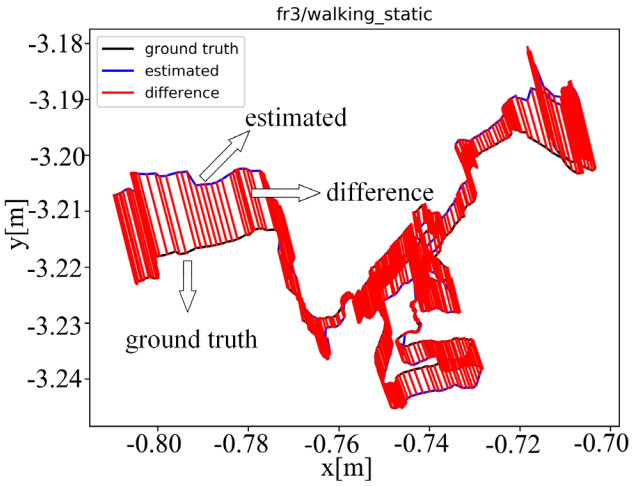


Fig. 9. ATE evaluation plot of our proposed method on sequences fr3/walking_static.

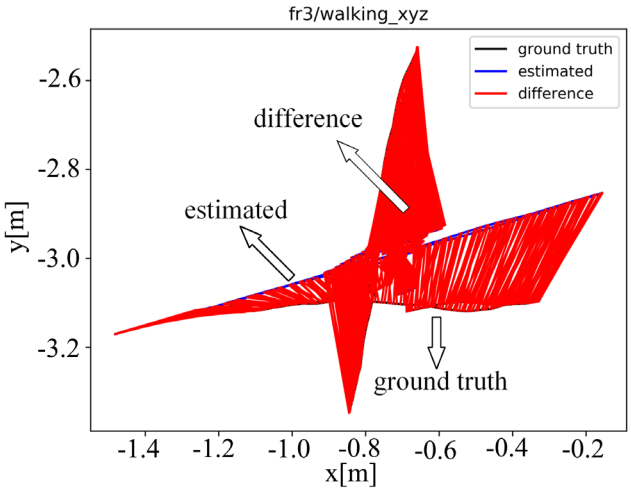
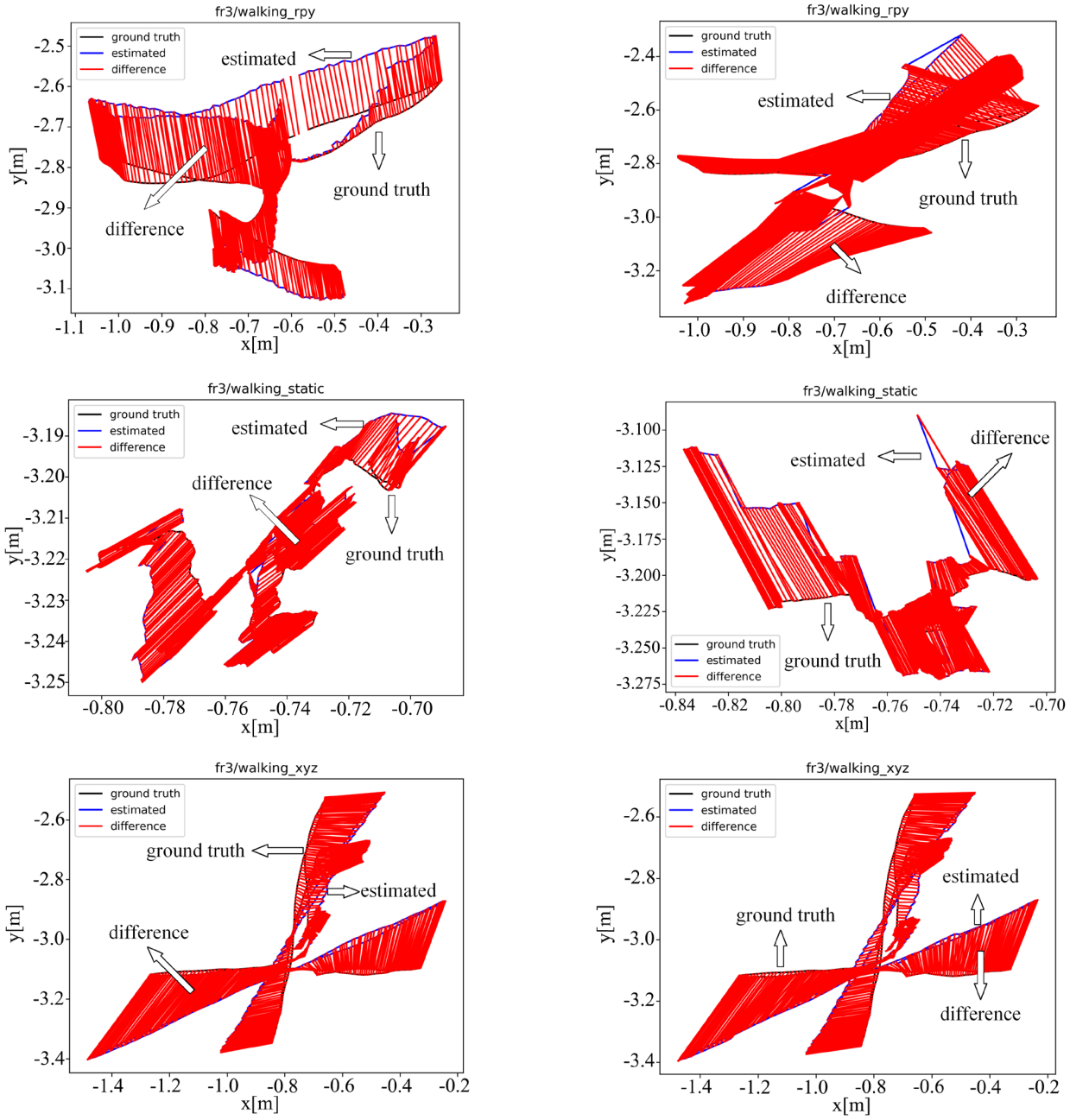


Fig. 10. ATE evaluation plot of our proposed method on sequences fr3/walking_xyz.

V. Conclusions

We proposed a new method for pose estimation. In the feature extraction stage, we extract the color features of color image frames and the contour features of depth image frames with separate convolutional streams to generate the corresponding feature maps. In the feature fusion stage, we combine the color feature maps and depth feature maps into a fused feature map group. In the pose estimation stage, we estimate the pose by modeling the motion model of the image sequence using LSTM. Our experiments on the TUM public dataset showed that our proposed method has higher accuracy and robustness and performs better than other convolutional-neural-network-based pose estimation methods, especially in challenging environments.

When using RGB-D cameras as sensors to acquire color and depth images, invalid depth data will inevitably occur. This data impacts the accuracy of the system. Future work will investigate the processing of invalid depth data generated during image acquisition by the RGB-D camera to reduce the error in the system's pose estimation.



(a) (b)

Fig. 11. Results of ATE assessment. (a) ATE evaluation plot of PoseNet on sequences fr3/walking_rpy, fr3/walking_static and fr3/walking_xyz. (b) ATE evaluation plot of DeepVO on sequences fr3/walking_rpy,fr3/walking_static and fr3/walking_xyz.

REFERENCES

[1] Cadena C, Carlone L, Carrillo H, et al, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-perception Age," IEEE Transactions on robotics, vol. 32, no. 6, pp1309-1332. 2016

[2] Bresson G, Féraud T, Aufrere R, et al, "Real-time Monocular SLAM with Low Memory Requirements," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp1827-1839, 2015

[3] Bresson G, Alsayed Z, Yu L, et al, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," IEEE Transactions on Intelligent Vehicles, vol. 2, no. 3, pp194-220. 2017

[4] Yang S, Scherer S, "Direct Monocular Odometry Using Points and Lines," IEEE International Conference on Robotics and Automation (ICRA), 29 May-3 June, 2017, Singapore, pp3871-3877.

[5] Engel J, Stückler J, Cremers D, "Large-scale Direct SLAM with Stereo Cameras," International Conference on Intelligent Robots and Systems (IROS), 28 October, 2015, Hamburg, Germany, pp1935-1942

[6] Engel J, Koltun V, Cremers D, "Direct Sparse Odometry," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 3, pp611-625, 2017

[7] Calonder M, Lepetit V, Strecha C, et al, "Brief: Binary Robust Independent Elementary Features," European conference on computer vision, 5-11 September, 2010, Springer, Berlin, pp778-792

[8] Leutenegger S, Chli M, Siegwart R Y, "BRISK: Binary Robust Invariant Scalable Keypoints," International Conference on Computer Vision, 6-13 November, 2011, Barcelona, Spain, pp2548-2555

[9] Mur-Artal R, Montiel J M M, Tardos J D, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Transactions on Robotics, vol. 31, no. 5, pp1147-1163, 2015

[10] Forster C, Pizzoli M, Scaramuzza D, "SVO: Fast Semi-direct Monocular Visual Odometry," IEEE international conference on robotics and automation (ICRA), 31 May, 2014, Hong Kong, China, pp15-22

[11] Klein G, Murray D, "Parallel Tracking and Mapping for Small AR Workspaces," IEEE & Acm International Symposium on Mixed & Augmented Reality, 13-16 November, 2008, Nara, Japan, pp225-234

[12] Engel J, Schps T, Cremers D, "LSD-SLAM: Large-scale Direct Monocular SLAM," European Conference on Computer Vision, 6-12 September, 2014, Zurich, Switzerland, pp834-849

[13] Mur-Artal R, Tardós J D, "Orb-slam2: An Open-source Slam System for Monocular, Stereo, and Rgb-d Cameras," IEEE transactions on robotics, vol. 33, no. 5, pp1255-1262, 2017
[14] Campos C, Elvira R, Rodríguez J J G, et al, "Orb-slam3: An Accurate Open-source Library for Visual, Visual–inertial, and Multimap Slam," IEEE Transactions on Robotics, vol. 37, no. 6, pp1874-1890, 2021
[15] Kendall A, Grimes M, Cipolla R, "PoseNet: A Convolutional Network for Real-time 6-dof Camera Relocalization," IEEE International Conference on Computer Vision (ICCV), 7-13 December, 2015, Santiago, Chile, pp2938-2946
[16] Wang S, Clark R, Wen H, et al, "DeepVO: Towards End-to-end Visual Odometry with Deep Recurrent Convolutional Neural Networks," IEEE International Conference on Robotics and Automation (ICRA), 29 May-3 June, 2017, Singapore, pp2043-2050
[17] Almalioglu Y, Saputra M R U, de Gusmao P P B, et al, "Ganvo: Unsupervised Deep Monocular Visual Odometry and Depth Estimation with Generative Adversarial Networks," International conference on robotics and automation (ICRA), 20-24 May, 2019, Montreal, QC, Canada, pp5474-5480
[18] Simonyan K, Zisserman A, "Very Deep Convolutional Networks for Large-scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014
[19] Shi X, Chen Z, Wang H, et al, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," arXiv preprint arXiv:1506.04214, 2015
[20] Bell S, Zitnick C L, Bala K, et al, "Inside-outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," Proceedings of the IEEE conference on computer vision and pattern recognition, 27-30 June, 2016, Las Vegas, NV, USA , pp2874-2883
[21] Kong T, Yao A, Chen Y, et al, "Hypernet: Towards Accurate Region Proposal Generation and Joint Object Detection," Proceedings of the IEEE conference on computer vision and pattern recognition, 7-30 June, 2016, Las Vegas, NV, USA, pp845-853
[22] Yang M, Yu K, Zhang C, et al, "Denseaspp for Semantic Segmentation in Street Scenes," Proceedings of the IEEE conference on computer vision and pattern recognition, 18-23 June, 2018, Salt Lake City, UT, USA, pp3684-3692
[23] Li R, Liu Q, Gui J, et al, "Indoor Relocalization in Challenging Environments with Dual-stream Convolutional Neural Networks," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp651-662, 2017
[24] Sturm J, Engelhard N, Endres F, et al, "A Benchmark for the Evaluation of RGB-D SLAM Systems," Intelligent Robots and Systems (IROS), 7-12 October, 2012, Vilamoura-Algarve, Portugal, pp573-580