Proceedings of the 3rd WRC Symposium
on Advanced Robotics and Automation 2021
Beijing, China, September 11, 2021

2021 WRC Symposium on Advanced Robotics and Automation (WRC SARA) | 978-1-6654-1823-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/WRCSARA53879.2021.9612673

ATFVO: An Attentive Tensor-compressed LSTM Model with Optical Flow Features for Monocular Visual Odometry

Hongwei Ren, Chenghao Li, Xinyi Zhang, Chenchen Ding, Changhai Man, and Hao Yu

***Abstract*—This paper proposes a new framework called ATFVO which can be deployed on the edge device to resolve monocular visual odometry problem. The vast majority of visual odometry algorithms using deep learning are equivalent to or beyond the traditional visual odometry algorithms in performance, however they do not consider the computing capability of edge equipment. In this paper, convolution neural network (CNN) and attentive tensor-compressed compression LSTM (A-T-LSTM) are used, with optical flow feature as input and a 6-DoF absolute-scale pose as output. The framework is fused with the spatio-temporal feature and deal with the overfitting problem of over-parameterized LSTM with high-dimensional inputs, and utilizes attention mechanism to get poses from the sequence output of T-LSTM. The poses are estimated from the original RGB images sequence without depending on any prior knowledge. The experimental outcomes at the KITTI dataset display that, in compared with the performance of the most advanced methods, the single T-LSTM model is 141× smaller than the original LSTM model, and the entire model is nearly one-seventh of DeepVO with a speed 23× faster than Flowdometry. The proposed VO is deployed to the robot based on raspberry pi, which can achieve real-time inference and navigate a cruise.**

I. INTRODUCTION

In robot simultaneous localization and mapping (SLAM), visual odometry (VO) is a very essential component [1]. The mission of VO is to estimate the movement of camera between adjacent images and the appearance of the local map. Monocular camera is cheap, but it provides much numerous information compared to other transducer. For instance, laser radar and inertial measurement unit (IMU). The monocular visual odometry has been widely concerned by academia and industry [2], and a large number of novel algorithms and applied products have sprung up. In terms of algorithm, traditional and deep learning algorithms are the main two major directions for estimating VO.

Traditional VO approaches, including camera distortion correction, feature extraction, pose calculation and local optimization etc., have been developed into a very mature system. These algorithms based on feature points or direct methods have achieved fantastic performance in accuracy and robustness [3]. Nevertheless, the above methods are always hard-coded with vital engineering works and every component in this approach is important enough to require careful design and fine-tuning to promote the property. Furthermore, for the sake of avoiding the serious shift problem, some extra information or prior knowledge should be provided when the algorithm estimates an absolute scale pose.

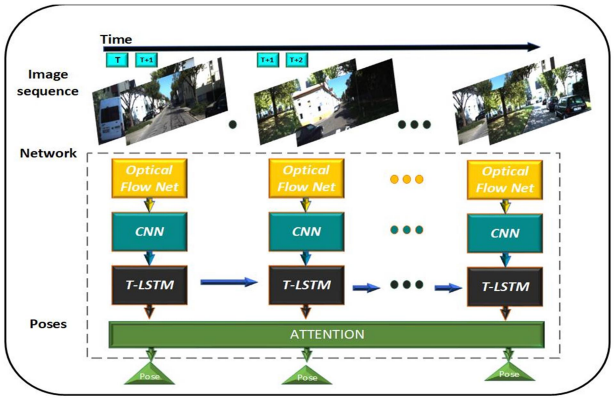


Figure 1. The framework of ATFVO system. Two consecutive image sequencess are overlaped as input to the optical flow network, as shown in this pipline steams [6][26]. The main contribution of CNN is to gain the spatial features and A-T-LSTM is to learn the temporal relationship of each frame in the video. The Output is in 6DoF pose format.

Deep learning VO approaches based on CNN and RNN have also blossomed some excellent systems [4]. Deep learning technology has developed rapidly, and achieved impressive results in the area of image classification, object detection and image segmentation. Different from CNN, which is the most important step of deep learning approach, the VO algorithms depend on low-level geometric features, while the former extract high-level information which is about appearance from images. This means that the existing pre-trained weights and models are not suitable for solving VO problems. In addition, temporal information is also the key to the VO problems. The single CNN network cannot extract the connection between frames, thus combining RNN can make up for the shortcomings by using CNN alone.

This research put forward a monocular visual odometry system named ATFVO which is according to CNN and A-T-LSTM. The most important work contribution results are divided into the following three aspects:

1) ATFVO system which utilizes the raw RGB images to estimate poses does not acquire the prior knowledge.

2) ATFVO system integrates CNN and A-T-LSTM to learn the spatial-temporal features of image pairs.

3) ATFVO system utilizes tensor-compressed and quantization methods to streamline the model so that the

*This work was supported by the National Natural Science Foundation of China (NSFC) (Key Program Grant No. 62034007), Innovative Team Program of Education Department of Guangdong Province (Grant No. 2018KCXTD028), the Key-Area Research and Development Program of Guangdong Province (Grant No. 2019B010142001) and Shenzhen Science and Technology Program (Grant No. KQTD20200820113051096).

The authors are with School of Microelectronics, and Engineering Research Center of Integrated Circuits for Next-Generation Communications, Ministry of Education, Southern University of Science and Technology, China.

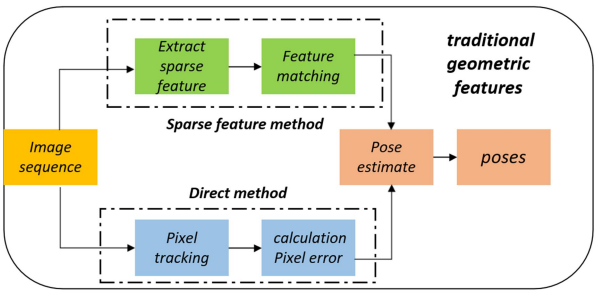


Figure 2. Sparse feature-based method and direct method.

system can be deployed on edge devices and maintain accuracy.

4) ATFVO system utilizes soft attention mechanism after T-LSTM to estimates poses.

The structure of this paper is relatively clear. The second part mainly introduces some related work in the past, the third part presents the basic situation of the RCNN model of tensor compression to readers, the fourth part is some experimental results and some analysis of our work, and the last part draws a conclusion and makes a summary.

II. RELATED WORK

In the following section, early work about the monocular VO will be reviewed and the differences between various algorithms will be discussed. Algorithms are mainly divided technically into two directions: geometry and learning methods. Many mature systems of Monocular VO have emerged. For instance, ORB-SLAM series and Direct Sparse Odometry (DSO) which depend on conventional method have been used in engineering area extensively, VINS-Mono is designed for drones, and DeepVO combines CNN with RNN [5].

A. Methods based on traditional Geometry features

Using rigorous mathematical derivation to estimate motion by geometric constraints extracted from image is called geometry based method. Based on this concept, many excellent systems have been derived. Moreover, whether to use sparse features and feature matching methods is to distinguish between sparse feature-based methods and direct methods.

Figure 2 represents a typical pipeline based on sparse features method. In 2004, Lowe D G. et al. propose scale-invariant feature transform (SIFT) [7]. The scale and rotation of SIFT features in images are invariant. And the experiment proved that the algorithm can achieve robust matching of affine distortion, three dimensional viewpoint changes, noise and illumination changes in a large range. Although the excellent performance of the SIFT algorithm has been proven, its slow calculation speed is unbearable in many real-time applications scenario. In order to solve this problem, in 2006 H. Bay et al. proposed a more computationally efficient accelerated robust feature named Speeded up Robust Features (SURF) [8]. Experiments show that the running speed of the algorithm is about three times faster than the SIFT algorithm. However, although SURF has increased the speed, it still has a long way to go in real-time applications. In 2011, E. Ruble et al. propose a more effective alternative named Oriented FAST and Rotated BRIEF (ORB) algorithm to SIFT. It is proved by experiments that the algorithm based on SIFT is two orders of magnitude slower than the algorithm based on ORB, and has rotation invariance and noise resistance [5]. Besides, the ORB feature does not have scale invariance, which will cause the estimated position to be different from the actual position by an uncertain scale in the case of monocular [9]. In summary, the extraction of sparse feature is extremely time-consuming, and all information except feature points is ignored when using this method which cannot make the best use of every pixel of the image. This method brings many limitations to the VO algorithm.

As for direct method, that is according to the gray invariant hypothesis to estimate the poses without calculating the features of images. Its typical pipeline is also shown in Figure 2. In 2013, J. Engel et al. propose an algorithm which is the first featureless monocular visual odometry method [10]. It can run in real-time on a CPU, and calculate the camera poses via way of means of reconstructing the semi-dense inverse depth map. On the other hand, there is a lot of uncertainty when using this method estimate poses because of the lack of loop closing detection. In 2011, a method which mainly reduced the uncertainty of poses estimation by precisely predicting the probabilistic depth of each pixel was proposed by R. A. Newcombe et al. to estimate the camera poses[11]. Nevertheless, this method is susceptible to the light illumination change because of the assumption of light intensity. In 2014, C. a semi-direct monocular visual odometry method was proposed by Forster et al., which extracts image blocks to obtain the camera pose [12]. This method sacrifices accuracy to increase speed. In conclusion, the direct method has the advantage of fast speed because it reduces the time of calculating features [13], and the disadvantage is touchy about illumination changes [14].

B. Methods based on Learning

This method accomplishes the same work by searching for data laws and obtaining the functional mapping relationship between the data and the target task [15]. In 2008, Roberts et al. first try to apply machine learning to deal with the VO problem. This method uses the K-Nearest Neighbors algorithm (KNN) to learn the speed and direction mapping from the sparse optical flow to the VO carrier [16]. In 2014, Ciarfuglia et al. first propose the use of Support Vector Machine (SVM) to perform self-motion pose regression and called it SVR-VO [15][17].

As is known to all, the traditional machine learning algorithm cannot complete the task well when confronted with a large amount of or highly nonlinear high-dimensional data [18]. However, deep learning algorithms have shown strong adaptability when learning large-scale data. This provides a novel direction for solving the VO problem. In 2015, Konda et al. put forward a deep convolutional neural network model which is end-to-end to appraise the changes in the velocity and direction of the camera, a simplex type of computing module and machine learning algorithms is used by the architecture in order to pick up information about visual motion, depth, and mileage from raw data [19]. In 2017, Muller et al. propose Flowdometry. The original optical flow calculated by Flownet

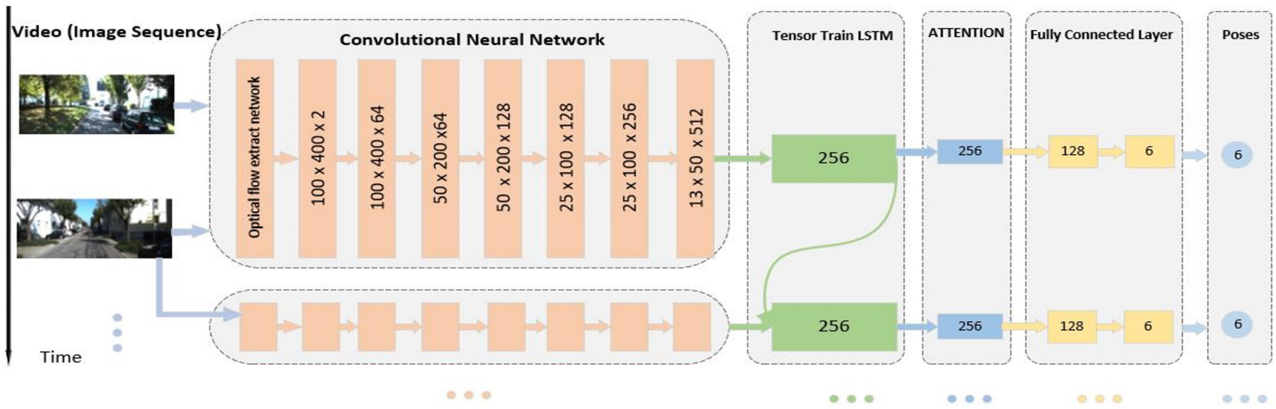


Figure 3. A monocular vision system structure is designed, which mainly depends on tensor compression theory. The dimension of tensor is given in the graph. The parameters from CNN to A-T-LSTM and fully connected layer are given as well.

is directly used as the input to the pose prediction network, and then the 2-DoF pose estimation between frames is directly obtained through the calculation of the fully connected layer [20][21]. In 2017, a model named DeepVO was proposed by S. Wang et al., DeepVO links the CNN and a unidirectional LSTM together for the first time [22]. This model utilizes the function of Recurrent Neural Network (RNN) which can handle the relationship between continuous video sequences well. In 2018, an end-to-end Bi-directional Recurrent Convolutional Neural Network (B-RCNN) called Magic VO was proposed by Jian Jiao et al. [23]. The behave of this method has even surpassed that of the traditional method [24].

Above all, these methods using deep learning do not consider the computational force of edge equipment, so they cannot be deployed on edge devices effectively. This paper proposes a pose estimation that combines CNN with A-T-LSTM together. The model can maintain certain accuracy, and the size of the model is so light that it can be deployed on edge devices easily. Excluding the optical flow network model, the size is only 29.5 MB and it can reach a speed of 37 FPS.

III. MODEL FOR ATTENTIVE TENSOR-COMPRESSED RCNN

In this section, the attentive tensor-compressed RCNN, which is a model framework for camera pose estimation is described in detail. From Figure 3, we can see the framework includes a CNN structure, an A-T-LSTM structure, and two Fully Connected Layers. The network can extract low-level geometric features and time series relationships well.

A. Architecture of the Proposed RCNN

The architecture of ATFVO is shown in Figure 3. Firstly, we compress the image to half of the original (150, 500, 3), and we superimpose two adjacent frames as input to optical flow extract network. Because normalization can improve the generalization ability and the convergence rate of the model, we subtract the pixel mean and then divide it by the variance from the optical flow feature vector (150, 500, 2). Secondly, the CNN structure is used to extract low-level geometric and spatial features, and the output of CNN structure is transmitted to the A-T-LSTM structure as its input. Then, the A-T-LSTM structure extracts spatial-temporal fused features and the output of A-T-LSTM structure is regarded as the input of the fully connect structure. Finally, the output of fully connect structure is the 6-DoF pose.

B. CNN Architecture

VO problem relies on more geometric features. However, most existing classical network structures are oriented to complete the tasks that require high-level features of classification, detection and segmentation [24][25]. In consequence, we can't use similar networks like InceptionNet, ResNet, and VGGNet to extract features that are utilized to resolve VO problem. The inspiration of the CNN structure comes from the optical flow estimation network [26][27][28]. To make the model lighter, we take a step further to compress the extracted optical flow vectors. The shape of CNN structure's output tensor is (13, 100, 512). Above all, the feature information obtained from the last convolution is transferred to the RNN for extracting temporal features.

C. A-T-LSTM Architecture

A attentive tensor-compressed LSTM is constructed to learn the spatial-temporal information after the CNN structure [29][30]. Each cell in A-T-LSTM[31] can track an internal state to represent its memory, and can update its state based on time as well as the current input and past state through learning, as shown below:

$$
\begin{aligned}
I_t &= \sigma_{gt}\left(W_i f_f + U_i S_{t-1} + b_i\right) \\
P_t &= \sigma_{gt}\left(W_p f_f + U_p S_{t-1} + b_p\right) \\
O_t &= \sigma_{gt}\left(W_o f_f + U_o S_{t-1} + b_O\right) \\
R_t &= I_t \odot R_{t-1} + P_t \odot \sigma_{ht}\left(W_r f_f + U_r S_{t-1} + b_r\right) \\
S_t &= O_t \odot \sigma_{ht}(R_t)
\end{aligned} \tag{1}
$$

where the input optical feature f_f is multiplied by weight matrices W_* to update three gates I_t, P_t, $O_t$and memory R_t. S_t and R_t are the current hidden state and memory, while S_{t-1} and R_{t-1} are previous hidden state and memory respectively, $\odot$ is the element-wise product and σ_{gt} is sigmoid activation function and σ_{ht} is tangent activation function.

The LSTM structure can adequately extract temporal features in sequences [32].On the other hands, it becomes high

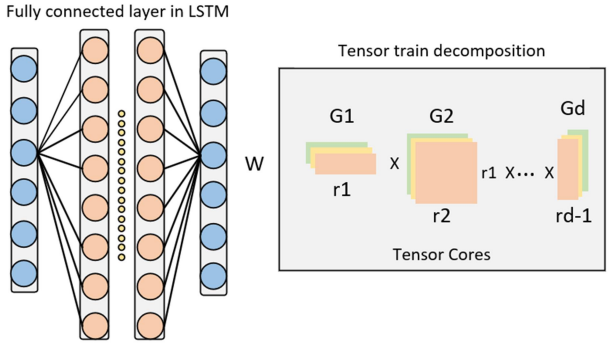


Figure 4. Tensor train format of weight matrix in the fully-connected layer in LSTM.

computation force requirements because of the large size of the resulting network. In the task of VO, video frame features usually form a high-dimensional input features, which makes the size of the input-to-hidden matrix in LSTM extremely large. A great number of undefined network parameters exist in the fully-connected layer, which is the main computing power consumption unit in LSTM. In response to this serious problem, this paper proposes a tensor-compressed method to solve this problem, which can greatly reduce the parameters of the model.

Tensor Train Preliminaries: By using the tensor train decomposition, a high-dimension tensor can be decomposed as the product of a sequence of low-dimension tensors. In detailed, given a k dimensional tensor $\mathcal{F} \in \mathcal{R}^{\mathrm{l_1 \times l_2 \times \ldots \times l_k}}$, it can be factorized and represented using a collection of d tensor cores $\mathcal{G} \in \mathcal{R}^{\mathrm{r_{k-1} \times l_k \times r_k}}$, where $\mathrm{k} \in [1, \mathrm{d}]$. Each element in $\mathcal{F}$ can be renewed as following:

$$\mathcal{F}(h_1, h_2, \ldots, h_d) = \mathcal{G}(h_1)\mathcal{G}(h_2) \ldots \mathcal{G}(h_d) \tag{2}$$

where $\mathcal{G}_p(h_p) \in \mathcal{R}^{r_{p-1} \times r_p}$ is a part of matrix from the high-dimensional tensor $\mathcal{G}_p$ and r_p is the rank of $\mathcal{G}_p$. For each integer l_k, it can be further decomposed as $l_p = m_{p\times} n_p$. As a result, each core of tensor $\mathcal{G}_p$ can be transformed into $\mathcal{G}_p{}^* \in \mathcal{R}^{r_{p-1} \times r_p \times m_p \times n_p}$. Therefore, the decomposition of the d dimensional tensor $\mathcal{F} \in \mathcal{R}^{m_1 \times n_1 \times m_{2\times} n_{2\times} m_{d\times} n_d}$ can be rewritten as:

$$\mathcal{F}\big((x_1, y_1), (x_2, y_2), \ldots, (x_d, y_d)\big) = \mathcal{G}_1^*(x_1, y_1)\mathcal{G}_2^*(x_2, y_2) \ldots \mathcal{G}_d^*(x_d, y_d) \tag{3}$$

Double-index trick as shown above is the core function to decompose the fully-connected computation in LSTM cells. In general, the large-scale tensor multiplication is the most expensive computation part in the LSTM cell, it can be generically denoted as:

$$\mathrm{y} = \mathcal{W}x + \mathrm{b} \tag{4}$$

where $\mathrm{x} \in \mathcal{R}^u$ is the input vector, $\mathrm{b} \in \mathcal{R}^i$ is the bias vector, $\mathrm{y} \in \mathcal{R}^i$ is the output vector, and $\mathcal{W} \in \mathcal{R}^{i \times u}$ is the weight matrix. To approximates $\mathcal{W}x$ with much fewer parameters, the weight matrix W is reshaped into a tensor $\mathcal{W} \in \mathcal{R}^{(i_1 \times i_{2\ldots} \times i_{d)}(u_1 \times u_2 \ldots \times u_d)}$, where $\mathrm{I} = \prod_{d=1}^{d} i_k$ and $\mathrm{U} = \prod_{d=1}^{d} U = u_k$. Similarly, x and b can be reshaped into d dimensional tensors $\mathcal{X} \in \mathcal{R}^{u_1 \times u_{2\times} \ldots \times u_d}$ and $\mathcal{B} \in \mathcal{R}^{i_1 \times i_2 \times \ldots \times i_d}$. The result is that the output y can also become a tensor $\mathcal{Y} \in \mathcal{R}^{i_1 \times i_2 \times \ldots \times i_d}$ with d dimension. Therefore, the multiplication of matrix-vector in LSTM cells can be reformulated as:

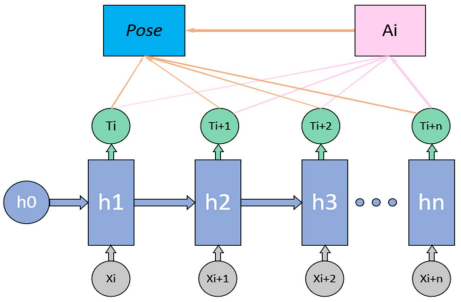


Figure 5. The soft attention mechanism with the T-LSTM output

$$\mathcal{Y}(x_{1,}x_{2,\ldots,}x_d) = \sum_{y_1=1}^{u_1} \sum_{y_1=2}^{u_2} \ldots \sum_{y_d=1}^{u_d} [\mathcal{G}_1^*(x_1, y_1)\mathcal{G}_2^*(x_2, y_2) \ldots \mathcal{G}_d^*(x_d, y_d) * \mathcal{X}(x_{1,}x_2, \ldots, x_d)] + \mathcal{B}(b_1, b_2, \ldots, b_d) \tag{5}$$

In this tensor decomposition way, the computational complexity in the tensor-compressed LSTM turns out to be $\mathcal{O}(\mathrm{dr}_{\mathrm{max}}^2 \mathrm{i_u})$ instead of $\mathcal{O}(\mathrm{i_u^d})$, where $\mathrm{r_{max}}$ represents the maximal rank of cores $\mathcal{G}_\mathrm{k}$ and $\mathrm{i_u}$ represents the maximal model size $\mathrm{i_k \cdot u_k}$ of weights $\mathcal{W}$.

Behind the T-LSTM output, we apply the soft attention mechanism to learn the softmax probability of the last output of each time status output as shown in Figure 5. We define $\mathrm{T} = [\mathrm{t_1}, \mathrm{t_2}, \mathrm{t_3} \ldots \mathrm{t_n}]$ as the output of T-LSTM, and then calculate the attention distribution $\mathrm{A_i} = \mathrm{softmax(c(t_i, q))}$, c is attentive scoring mechanism and q is query vector, in this paper, the c is defined as following:

$$\mathrm{c}(t_i, \mathrm{q}) = {t_i}^T q \tag{6}$$

Finally, we get the output through $\sum_{i=1}^{n} tiAi$.

D. Cost Function and Optimization

In the deep learning training, the setting of loss function and super parameter is essential. The proposed architecture of monocular VO system called ATFVO can compute the poses increment relative to the previous frame $\mathcal{Y}_\mathrm{t} = (\mathrm{y_1}, \mathrm{y_2}, \ldots, \mathrm{y_d})$ given RGB image data in a certain time series up to time t in $\mathcal{X}_\mathrm{t} = (\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_d})$ the conditional probability formula:

$$\mathrm{P}(\mathcal{Y}_t|\mathcal{X}_t) = \mathrm{P}\,(y_1, y_2, \ldots, y_d|x_1, x_2, \ldots, x_d) \tag{7}$$

The poses are obtained through the ATFVO model inference. In order to figure out the parameter l which is optimal for the VO, it can be abstracted as:

$$\mathrm{l}^* = argmax\, P(\mathcal{Y}_t\,|X_t; \mathrm{l}) \tag{8}$$

The Euclidean distance and Manhattan distance are mixed between displacement and rotation in pose $(\mathrm{a_k}, \mathrm{b_k})$ at time k and its estimated one $(\widehat{\mathrm{a_k}}, \widehat{\mathrm{b_k}})$ needs to be minimized in order to learn how to reduce l, the hyperparameters of the ATFVO. The loss function is obtained by mixed distance formula:

$$\mathrm{loss}^* = argmin \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{t} ||\widehat{\mathrm{a_k}} - \mathrm{a_k}||_2^2 + \delta|\widehat{b_k} - b_k| \tag{9}$$

where $||\cdot||$ is 2-norm, δ is a weight coefficient that balances the weights of displacement and rotation angle, and N is the

TABLE I RESULTS OF THE TESTING SEQUENCES

Seq	ORB_SLAM2		DeepVO		ATFVO	
	t(%)	*r(°)*	*t(%)*	*r(°)*	*t(%)*	*r(°)*
1	\	\	\	\	4.62	2.10
3	10.75	8.82	8.49	6.89	**6.08**	**4.27**
7	3.81	**1.67**	**2.62**	3.61	2.96	2.22
9	7.62	1.78	\	\	**6.92**	4.41
Mean	7.39	3.99	5.55	5.25	**5.15**	**3.25**

t(%): average translational RMSE drift (%) on length of 100m-800m.
r(∘): average rotational RMSE drift (∘/100m) on length of 100m-800m

number of total samples. We modify different values of δ in the experiment to get the best results. Because quaternion will increase the complexity of the whole network, Euler angles represent the orientation a in order to simplify the model.

IV. EXPERIMENTAL RESULT

In the following discussion, the proposed monocular VO approach will be estimated whether its performance is well or not on the popular KITTI VO benchmark [33]. To evaluate the proposed model, ATFVO was compared with the most advanced feature-based ORB_SLAM2 model. The experimental setups are following: NVIDIA GTX-1080Ti, CPU E5-2660, AI acceleration core and raspberry pi 3B+ are employed for hardware realization and performance comparison.

A. Dataset

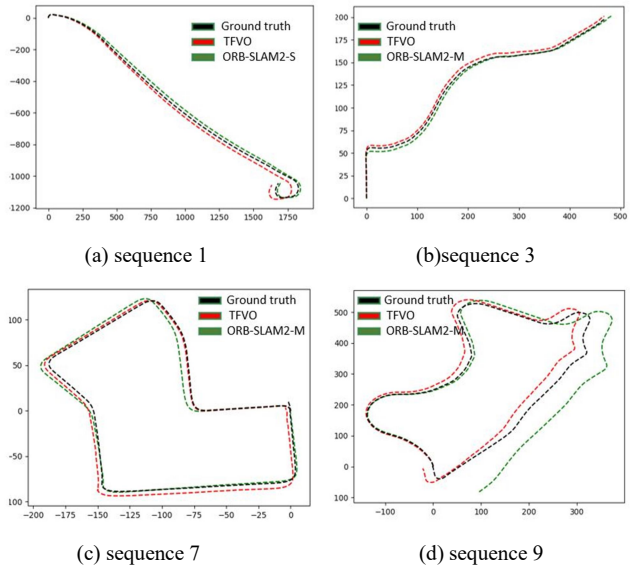
The KITTI dataset, as the largest dataset in the field of autonomous driving so far, contains 11 video sequences of known ground truth and 11 video sequences of unknown ground truth in the odometer dataset. The dataset includes videos from real driving scenes on highways and in rural areas. The ground truth matrices corresponding to the video are all quaternion rotation matrices for the initial position. Quaternion can avoid the problem of describing the gimbal lock in rotation, but our model is used to regress the parameters of rotation Euler angle, and the output of the whole network is translation and rotation in three directions.

B. Training and Testing

We designed two independent experiments. One of them is to train on sequences with a larger number of frames in 0-10, and test on sequences with a smaller number of frames. Each sequence in 0-10 is grouped into training set and testing set, training set sequence used in this model is No.00, 02, 04, 05, 08 and10. More data will improve its performance. Another experiment is to train on sequences with ground truth, then test and compare them on sequences without ground truth.

C. VO Results

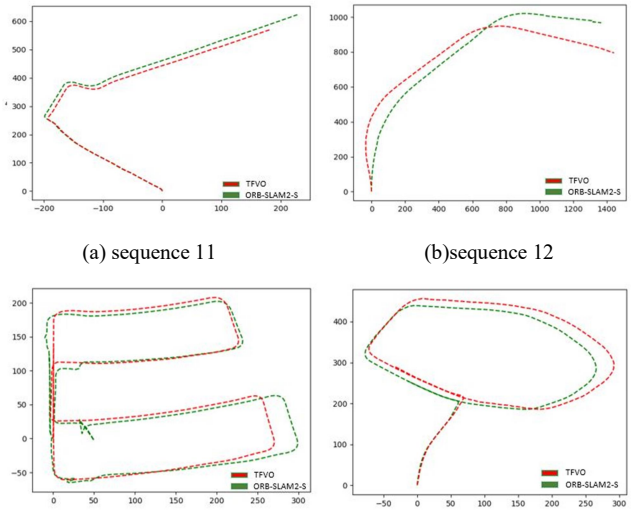
This model is trained based on the 00, 02, 04, 05, 08, 09 video data in the KITTI dataset. RMSEs is an evaluation index with high universality. In this paper, RMSEs was used as the model evaluation index to value the overall performance of the trained model in the test suite, in which we used 01,03,07,09 video data from KITTI dataset as the testing set. The inference



(a) sequence 1 (b)sequence 3

(c) sequence 7 (d) sequence 9

Figure 6. Trajectories of ATFVO and ORB-SLAM2 on Sequence 01, 03, 07 and 09.



(a) sequence 11 (b)sequence 12

(c) sequence 16 (d) sequence 18

Figure 7. Trajectory results of ATFVO and ORB-SLAM2-S using ATFVO on testing the KITTI odometry Sequence 11, 12, 16 and 18 (without ground truth).

obtained using the testing set is shown in Figure 6. According to the ground truth given in the dataset, it can be found that the prediction results consequences obtained by using this model are accurate. The test results of ATFVO with ORB_SLAM2 and DeepVO on the same test video data are shown in Table Ⅰ, among which the most important thing is the comparison of average x y z displacement drift (RMSE) and average α β γ rotational drift (RMSE). The contrast was stark. Compared with traditional ORB_SLAM2, the trajectory of ATFVO estimation is more convincing. ATFVO also has some competitive and advanced advantages for the DeepVO model according to convolutional neural network. However, due to the compression and reduction model, a slight accuracy is sacrificed compared to DeepVO.

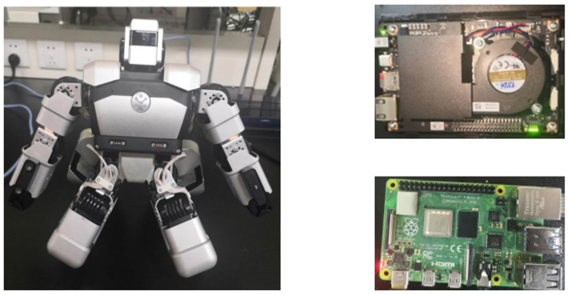


Figure 8 . (a) robot tested on (b) edge device(AI accelerator raspberry pi)

It is obvious that the ATFVO model performs very well in the first 11 video sequences, and the predicted results are also very convincing. However, to prove the adaptability of ATFVO model in more different circumstance, we decided to use the 11 video sequences that did not give ground truth in the KITTI dataset for testing. For this reason, we trained the model on the first 11 groups of data with ground truth, which increased the amount of data for training and as a result obtained a more robust model. Then, the obtained model was used to evaluate the data without ground truth. Because this part of the sequence did not give the ground truth, so we temporarily applied ORB-SLAM2 system (stereo system) to assist us in qualitative estimation, and compared it with the prediction results we obtained. The specific prediction results are shown in Figure 7. In addition, the execution times of the different systems were tested and the results are listed in Table II (A). It is not difficult to find that ATFVO has 23× speedup over the Flowdometry.

D. Edge Devices Results

To verify the speedup and lightweight of our proposed VO, we implemented it on an edge device with the high efficient and programmable, AI chip, raspberry pi 4 and raspberry pi 3B+ as shown in Figure 8. The AI chip contains a digital vision pre-processing module, an Arm CPU, two AI cores, an on-chip buffer with the capacity of 8 MB, and some driver interfaces.

Due to complicate network architecture, the optical flow feature extraction part and the LSTM prediction part are the most time-consuming parts in ATFVO. AI acceleration core acts a pivotal part in speeding up the spatio-temporal feature extraction part. In addition, tensor compression is utilized to accelerate the A-T-LSTM prediction process. The overall implementation flow applied to the AI core based edge device can be divided into 5 steps. Firstly, the Ubuntu for ARM 16.04 -64-bit firmware is made on the AI core based edge device as the operating system. Secondly, all of the dependent libraries such as OpenCV, Python, Tensorflow, Keras and device development kit are compiled in the operating system. Then, solve the incompatibility of these libraries. After the running environment is configured, the inference code would be built on the board. Finally, ATFVO is achieved on the AI core based edge device.

Besides the outstanding accuracy of VO, the storage size and inference speed are also remarkable compared with the existing approaches as shown in TABLE Ⅲ. The storage size

TABLE II(A) EVLUATE ON PC

NAME	Optical flow calculation (s/frame)	RGB conversion (s/frame)	Odometry Calculation (s/frame)	Total execution (s/frame)	Speed up
VISO2-M	14.812	0.210	0.063	15.085	558
SVR-VO	14.812	0.210	0.112	15.134	560
P-CNN	14.812	0.210	0.041	15.063	557
Flowdometry	0.271	0	0.362	0.633	23
ATFVO	0.02	0	0.007	0.027	—

Timing information is evaluated for each system.

TABLE II(B) EVLUATE ON EDGE DEVICE

NAME	Raspberry pi 3B+	Raspberry pi 4B	AI Core Chip
ATFVO	0.178s	0.093s	0.019s

We put the ATFVO excluding the optical flow network on the edge device for deployment, and tested the time required for single pose estimation. The running time of the ORB-SLAM2-M on raspberry pi 3B+ is 0.576s for each frame.

TABLE Ⅲ PARAMETERS OF DIFFERENT METHODS

NAME	*Method*	*Parameter*
DeepVO	CNN+LSTM	510.16M
Flowdometry	CNN+FC	260.31M
ATFVO	CNN+A-T-LSTM	89.4M

ATFVO model parameter size is superior to other deep learning algorithms, especially when deployed on edge devices

of T-LSTM model is reduced by 210.9× after tensor compression. Furthermore, it also shows superiority against other advanced approaches. Table II(B) shows the time comsuming between different edge systems, it is found that ATFVO can also have high speed over the devices. We try to deploy a similar deep learning odometer like deepvo on the same edge device, but it can't execute because of the limited computing and storage resources of the edge device.

V. CONCLUSION

This paper proposes a new light deep learning framework model ATFVO that is mainly used for the estimation and calculation of camera poses in the limited resource environment with only monocular camera and edge devices with limited computing power. Primarily, CNN and A-T-LSTM form ATFVO. After comparing the performance of ATFVO with other advanced algorithms on the KITTI dataset, it is not difficult to find that ATFVO algorithm has strongly competitive performance in the accuracy of pose estimation and illumination invariance, and the model has advantages when deployed on edge devices in terms of size and accuracy.

We plan to concentrate on deep learning to replace other slam modules, and use tensor compression and quantization techniques to compress neural networks for the future work. Finally, form a complete slam system that can be deployed on edge devices.

REFERENCES

[1] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2004 CVPR 2004, vol. 1, no. C, pp. 652–659, 2004.

[2] a J. Davison, "SLAM with a Single Camera," SLAMCML Work. ICRA 2002, 2002.

[3] M. Li, B. Hong, Z. Cai, and R. Luo, "Novel Rao-Blackwellized particle filter for mobile robot SLAM using monocular vision," Int. J. Intell, vol. 1, no. 1, pp. 1021–1027, 2006.

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. V enu-gopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolu-tional networks for visual recognition and description," IEEE Trans-actions on Pattern Analysis and Machine Intelligence, to appear.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, 2015

[6] C. Zach, T. Pock, H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," Dagm,2007, 4713(5): 214-223

[7] D. G. Lowe. "Distinctive image features from scale-invariant keypoints," International Journalof Computer Vision, 2004, 60(2): 91-110

[8] H. Bay, T. Tuytelaars, L. Van Gool., "Surf: Speeded up robust features," Computer Vision –ECCV 2006, Berlin, Heidelberg, 2006, 404-417

[9] G. Klein, D. Murray, "Parallel tracking and mapping for small ar workspaces," Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed andAugmented Reality, 2007,1-10

[10] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," Proc. IEEE Int. Conf. Comput. Vis., pp. 1449–1456, 2013.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM : Dense Tracking and Mapping in Real-Time" pp. 2320–2327, 2011.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," Proc. - IEEE Int. Conf. Robot. Autom., 2014.

[13] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Transactions on Pattern Anal-ysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[14] J. Engel, V . Koltun, and D. Cremers, "Direct sparse odometry," in arXiv: 1607.02565, July 2016.

[15] T. A. Ciarfuglia, G. Costante, P. V aligi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," Robotics and Au-to nomous Systems, vol. 62, no. 12, pp. 1717–1730, 2014.

[16] R. Roberts, H. Nguyen, N. Krishnamurthi, et al, "Memory-based learning for visual odometry," 2008 IEEE International Conference on Robotics and Automation, 2008, 47-52

[17] E. Rosten, T. Drummond, "Machine learning for high-speed corner detection," European conference on computer vision, 2006, 430-443

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

[19] K. Konda and R. Memisevic, "Learning Visual Odometry with a Convolutional Network," Int. Conf. Comput. Vis. Theory Appl., pp. 486–490, 2015.

[20] P. Muller, A. Savakis, " Flowdometry: An optical flow and deep learning based approach to visual odometry[C]," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 2017,624-631

[21] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. HighAccuracy， "Optical Flow Estimation Based on a Theory forWarping," ECCV 2004, volume 3024,pp 25 – 36. 2004.

[22] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards End to End Visual Odometry with Deep Recurrent Convolutional Neural Networks," IEEE Int. Conf. Robot. Autom, pp. 2043–2050, 2017.

[23] J. Jiao, J. Jiao, Y. Mo, et al, "Magicvo: End-to-end monocular visual odometry through deep bi-directional recurrent convolutional neural network," arXiv: Computer Vision and Pattern Recognition, 2018, :

[24] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison, "Gvnn: Neural network library for geometric computer vision," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9915 LNCS, pp. 67–82, 2016.

[25] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," In CVPR, 2017

[26] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8934–8943.

[27] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V . Golkov, P. V a Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 2758–2766.

[28] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, " Flownet 2.0: Evolution of optical flow estimation with deep networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2462–2470.

[29] S. K. Sønderby, C. K. Sønderby, L. Maaløe, andO. Winther, "Recurrent Spatial Transformer Networks," pp. 1–9, 2015.

[30] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in Proceedings of International Conference on Machine Learning (ICML), vol. 14, 2014, pp. 1764–1772.

[31] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. V etrov, "Tensorizing neural networks," in Advances in neural information processing systems, 2015, pp. 442–450.

[32] R. Christoph and F. A. Pinz, "Spatiotemporal residual networks for video action recognition," Advances in Neural Information Processing Systems, pp. 3468–3476, 2016.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2012