ASD-SLAM: A Novel Adaptive-Scale Descriptor Learning for Visual SLAM

Taiyuan Ma[1] and Yafei Wang[1]and Zili Wang[2] and Xulei Liu[1] and Huimin Zhang[1]

***Abstract*— Visual Odometry and Simultaneous Localization and Mapping (SLAM) are widely used in autonomous driving. In the traditional keypoint-based visual SLAM systems, the feature matching accuracy of the front end plays a decisive role and becomes the bottleneck restricting the positioning accuracy, especially in challenging scenarios like viewpoint variation and highly repetitive scenes. Thus, increasing the discriminability and matchability of feature descriptor is of importance to improve the positioning accuracy of visual SLAM. In this paper, we proposed a novel adaptive-scale triplet loss function and apply it to triplet network to generate adaptive-scale descriptor (ASD). Based on ASD, we designed our monocular SLAM system (ASD-SLAM) which is an deep-learning enhanced system based on the state of art ORB-SLAM system. The experimental results show that ASD achieves better performance on the UBC benchmark dataset, at the same time, the ASD-SLAM system also outperforms the current popular visual SLAM frameworks on the KITTI Odometry Dataset.**

I. INTRODUCTION

Feature matching is one of the key steps for Simultaneous Localization and Mapping (SLAM), which in turn depends on the quality of descriptors. The descriptors are feature abstraction of the original pixels of the images. Effective descriptors should be able to cope with image transformation, illumination changes and so on while describing the image features. Over the past decade, researches focused keypoint descriptors based on hand-crafted solutions such as SIFT [1], SURF [2] and ORB [3]. These descriptors still play important roles in current popular visual SLAM frameworks like ORB-SLAM2[4]. Among these hand-drift descriptors, the sift descriptor has a higher matching precision, but requires too much computation. The recent rise of deep learning has created the opportunity to develop learning-based and data-driven techniques of keypoint description. According to [8], the descriptors coming from trained-CNN outperform the hand-crafted descriptors in terms of their invariance properties in patch verification tasks. Among the methods based on the CNNs for keypoint description [5-7] , [11-17], the most famous models are DeepDesc [5], L2-Net [6], CS L2-Net [6] and HardNet [7], they produce 128 or 256 dimensions unit eigenvectors like SIFT, ORB. All studies about keypoint description with trained CNNs inevitably compare their performance with hand-crafted descriptors, and come to a common conclusion that they outperform the hand-crafted descriptors in terms of their invariance properties [8]. Although these learning-based descriptors achieve good performance in patch verification tasks, they are not popular in practical applications. Especially, according to a recent research [9], in some complicated tasks, like SFM, traditional hand-crafted features (SIFT [1] and its variants [10]) still prevail over the learned ones. The main reason is that most researches did not consider the specificities of the specific applications like SLAM, SFM when designing loss functions, which made the descriptors difficult to apply to these applications. Traditionally, most researches focus on data augmentation or build more suitable datasets to increase the robust to deal with illumination and viewpoint changes in practical applications and ignore the importance of loss function. For example, most learning-based methods adopt Siamese losses [5],[11-13] and Triplet losses [6][7][14-17] aiming at reducing the distance between similar image patches and increase the distance of dissimilar ones. Generally, Triplet losses are reported to have better performance than Siamese losses according to [17]. However, Triplet losses suffer from scale uncertainty, according to [18], which is fatal to the feature matching between multiple frames of SLAM and SFM. Therefore, in this paper, In order to enable the descriptor to adapt to the feature matching of consecutive frames in SLAM, we proposed an Adaptive-Scale Triplet Loss function and apply it to Triplet Network to better solve the problem of scale uncertainty and obtain our adaptive-scale descriptor (ASD). Moreover, by replacing the front end of the traditional visual SLAM framework with ASD , we design the deep-learning enhanced SLAM system (ASD-SLAM). We separately evaluate the performance of ASD and the positioning accuracy of ASD-SLAM on the public datasets. The experimental results show that ASD achieves the better performance in patch verification tasks and the ASD-SLAM positioning results are more accurate than the influential monocular SLAM systems like ORB-SLAM, LDSO. In addition, ASD is not only applied to SLAM, but also can be extended to other similar fields like SFM. In summary, our main contributions[1] are the following:

- We proposed an adaptive-scale triplet loss function and applied it to triplet network to generate ASD which achieved state-of-art performance on the public Brown dataset.
- We design our deep-learning-enhanced SLAM system (ASD-SLAM), and obtained better results comparing to state-of-the-art visual SLAM systems like ORB-SLAM and LDSO.

[1]T. Ma, Y. Wang, X. Liu, H. Zhang are with School of Mechanical Engineering, University of Shanghai Jiao Tong, Shanghai 200240, China (corresponding author: Yafei Wang, e-mail: wyfjlu@sjtu.edu.cn).

[2]Z. Wang is with Company of Xiao Peng, Guangzhou, China

[1]https://github.com/mataiyuan/ASD-SLAM?files=1

SALM依赖特征匹配，特征匹配又依赖描述子的质量，描述子就是对图片像素的特征提取

SIFT匹配精确，但算力大

这些深度学习描述子只适用于patch verification tasks，不适用于SLAM、SFM

实际使用中在复杂任务中还是用传统设计的特征(SIFT)

主要原因是在设计损失函数的时候作者没有考虑SLAM、SFM等实际应用的特殊性，最终导致描述子难以实际使用

大多数人专注于数据增强和设计更稳定的数据集来增强鲁棒性解决光照变化和视角变化，却忽视了loss function的重要性.例如，采用Siamese losses和triplet lossed。后者一般效果更好，然后triplet有尺度不确定的缺点，对于SLAM和SFM的多帧特征匹配很致命

本文设计一种自适应尺度triplet loss function运用到triplet netwwork中更好地解决尺度不确定问题，得到ASD

设计了自适应尺度的triplet loss function，并设计了网络生成了ASD

设计了ASD-SLAM

II. RELATED WORK

Since this paper is aimming at learning suitable local descriptor which can enhance visual SLAM system, in this section we review related works with respect to the two fields that we integrate within our research, local feature descriptor learning and deep learning enhanced SLAM.

A. Local Feature Descriptor Learning

Parallel with the long history of local feature, numerous researchers have made considerable attempts. Classical hand-craft local feature descriptors like SIFT [1], SURF [2], ORB [3] are proved to be effective in SFM, SLAM , 3D reconstruction, etc. After that, based on traditional hand-craft descriptors, combined with machine learning to generate Binboost [20], RMGD [21], PCA-SIFT [19] which are basically designed to map descriptors from high-dimensional space to low-dimensional space to improve real-time performance. However, such an operation reduces the feature expression ability of the descriptor. In recent years, lots of papers in deep-learning based descriptor are proposed. Mainly divided into two categories, one is an end-to-end network framework, and the other is based on multi-branch CNN network like Siamese and triplet networks. First, an end-to-end learning network extracts feature points and descriptors simultaneously. Lift [24] proposes a deep network framework that combines feature point detection, direction estimation, and descriptor calculation modules, using back propagation for end-to-end training. SuperPoint [22], which builds a network training dataset by artificially setting corner points, and labels the data set according to Homographic Adaptation on the dataset. LF-Net [23] embeds the entire feature extraction pipeline, and can be trained end-to-end with a small amount of images, However, LF-Net need to use image pairs with known relative pose and corresponding depth maps to generate training data, this limits the use of the network in some scenarios such as SLAM, SFM. Second, [25] first proposed the siamese network in signature verification. [12]'s goal is to directly use CNN's powerful feature expression skills to learn a common similarity function for image patches. [11] (MatchNet) proposed a joint learning patch representation of a deep network and a robust feature comparison network structure. This paper uses the fully connected layer to represent the similarity of two descriptors through the learned distance metric. Above are based on the pairwise siamese structure. However, all of above are proved to be not suitable to use fast approximate nearest neighbor algorithms for matching. [15][26] uses triplet structure and achieves better performance than siamese structure with metric learning layer. [6] proposed L2-net, and the descriptor can be matched by L2 distance while getting better performance descriptors, and L2-net uses a progressive sampling strategy which enables the network to access billions of training samples in a few epochs. Based on [6], Hardnet [7] adopted hardest negative mining strategy to select n hard-negative samples among n2-n negative samples of L2Net, and achieve state-of-the-art performance. [18] was trained by a mixed-context loss with the same architecture as L2Net, and improved the performance of descriptor. [18] firstly pay attention to the importance of scale in triplet losses. However, in [18], scale correction parameter is setted manually.

基于传统描述子，利用机器学习设计了Binboost、RMGD,PCA-SIFT，将描述子从高维空间映射到低维空间提高实时性，却降低了描述子的expression ability

深度学习描述子主要分为端到端和多分支CNN网络

LF-Net需要用图片对的相对位姿和深度图来生成训练数据，这就限制了网络在SLAM，SFM中的应用

Siamese结构不适用于快速近似最近邻匹配算法，triplet结构表现更好

L2-Net使用了sampling 策略在几个epochs中获取数百万训练样本

18首次注意到triplet loss中的尺度问题，但是尺度修正参数是手动设定的

B. Visual Slam With CNN

Deep learning is a powerful method to solve feature description and data association problems encountered in the traditional SLAM framework. Some studies abandoned directly the traditional SLAM framework, using an end-to-end network architecture [27, 28, 29]. Although the end-to-end network structure makes the entire SLAM system more integrated, in most scenarios, the effect is not as good as the traditional SLAM. SLAM based on end-to-end has a weak scene generalization ability. In order to enhance the adaptability of the SLAM system in special scenarios, the semantic SLAM has become a hot topic in current research. [31] combines semantic segmentation network with moving consistency check method to reduce the impact of dynamic objects. [32] uses the professional probability model to extract the semantic information from the scene, and combines feature-based and direct approaches to achieve positioning in highly dynamic environments. [33] learn a discriminative holistic image representation to create a dense and salient scene description, to deal the changing weather conditions, illumination and seasons. [30,34] focus on the feature extraction and matching module of SLAM. [30] apply descriptor learning to construct line segment descriptors optimized for matching tasks. This method has better adaptability in scenes with many line features, but it will fail in some scenes with few line features. [34] learns descriptor with a supervised learning strategy employing a triplet loss. However, [34] also does not take into account the scale uncertainty of triplet losses.

As can be seen from the above, triplet losses play an important role on feature descriptor learning and SLAM. However, all of researches neglect the scale uncertainty of triplet losses or did not solve the problem effectively. Different from the above methods, we implement adaptive-scale triplet loss function which ensure the triplet network can notice the scale changes of the descriptor adaptively in training to generate descriptor for SLAM. With our results, we demonstrate adaptive-scale triplet loss function achieve greate success on the UBC benchmark dataset.

深度学习善于特征描述和数据关联

端到端SLAM没有很好的场景泛化能力

语义SLAM适应动态场景，长时间定位

运用线特征做匹配的，但是线少了就不行

我们提出的自适应尺度triplet loss function保证了triplet network能自适应调整描述子的尺度变化来训练得到SLAM需要的描述子

III. METHODOLOGY

In this section, we give a brief overview how scale uncertainty of triplet losses affects feature matching of SLAM (Section III-A), how the adaptive-scale triplet loss function is applied to triplet network to generate ASD (Section III-B-E) and the framework of ASD-SLAM(Section III-F).

A. *The Influence Of Scale Uncertainty In Triplet Losses On SLAM*

尺度不确定对triplet loss的影响

Monocular SLAM systems solve pose transformation by matching the descriptors of adjacent frames, combined with visual geometry, and triangulate the depth of the feature

单目SALM系统求取位姿变换是通过匹配相邻帧的描述子，运用视觉几何，以及三角化求特征点的深度

points. Fig. 1 shows the process. Each frame which observed the same feature point will calculate the descriptor of the feature point. Therefore, each feature point formed a set of matching descriptors:

$$D_{kp} = \{(k_i, X_i)\}_i^N \quad (1)$$

Each k_i is a 128-dimensional feature descriptor, X_i is the frame number. N represents the number of frames in which the feature point is observed. In the process of mapping, the depth of each feature point is obtained, and then the mappoint is formed with a descriptor k_{mp} chosen from D_{kp}. At the same time, k_{mp} will be updated continuously as new image frames are added. k_{mp} plays an important role in the subsequent SLAM process. As is Fig. 1 shown, we assume that mappoint corresponding to f_1 is mp_1 triangulated by frames(X_1, X_2, X_3), the descriptor corresponding to f_1 is k_{mp1}. We find that X_5 get the tracking of f_1 again after X_4 lost it. We assume that k_5^1 is the descriptor generated by X_5 for f_1. How to associate the descriptor k_5^1 with k_{mp1} ? Generally, we need to calculate the minimum distance $d_{(k_{mp1},k_5^1)} = \left\|k_{mp1} - k_5^1\right\|$ with nearest neighbor search, if $d_{(k_{mp1},k_5^1)} \leq a_{threshold}$, k_{mp1} and k_5^1 are successfully associated. So, $a_{threshold}$ is a key parameter. If $a_{threshold}$ is too small, we are more likely to lose tracking and large value will increase mismatching. Theoretically, the smaller D_{kp} divergence like (a) in Fig. 2, the better for SLAM. In other words, descriptors generated from the same feature point must have high robustness.

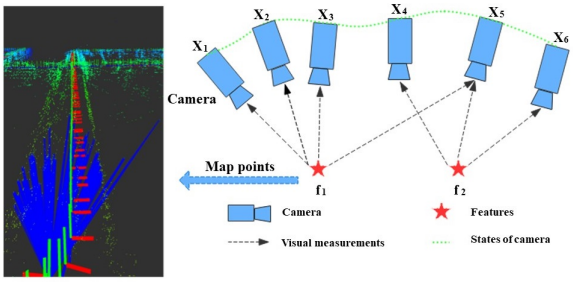


Fig. 1: An illustration of monocular SLAM problem. The feature f_1 generally observed in sucessive frames(X_1, X_2, X_3) , X_5 and is then triangulated as a map point in the map.

To generating robust descriptor, most researchers focus on learning feature descriptors with triplet losses based on triplets of patches in recent. Learning with triplets involves training from samples of the form $\{x_a, x_p, x_n\}$, x_a is called the anchor, x_p is a different sample of the same class as x_a, x_n is a sample belonging to a different class. The output of $\{x_a, x_p, x_n\}$ after propagation through the network is $\{y_a, y_p, y_n\}$. The training goal of the network is bringing x_a and x_p close in the feature space, and pushing x_a and ~~x_p~~ x_n far away. In that process, the positive distance δ_+ and the negative distance δ_- are computed:

$$\delta_+ = \|x_a - x_p\| \quad (2)$$

$$\delta_- = \|x_a - x_n\| \quad (3)$$

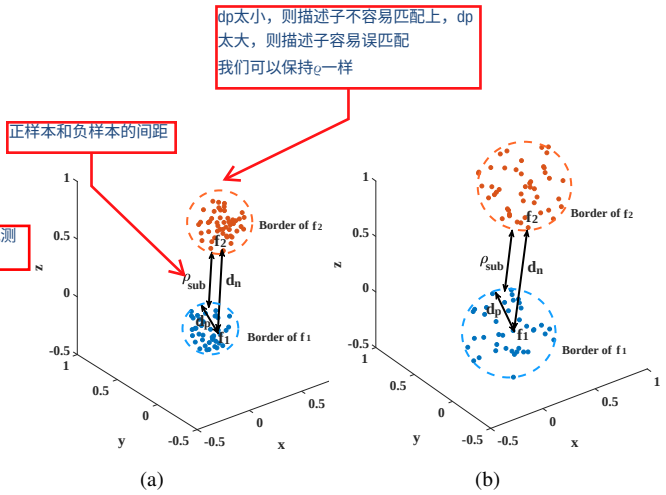


Fig. 2: The descriptors sets of feature f_1 (blue), f_2 (orange) are illustrated in a three-dimensional descriptor space. (a) is the ideal situation for SLAM, (b) has a larger divergence of two descriptors set. However, we can keep the ρ_{sub} between (a) and (b) same.

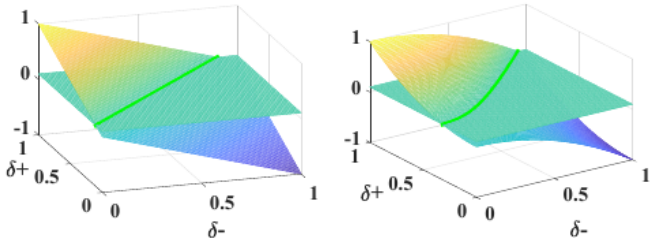


Fig. 3: An illustration of $\rho_{sub}\left(\delta_+, \delta_-\right)$ and $\rho_{sub}^2\left(\delta_+^2, \delta_-^2\right)$.

To satisfy the training goal, the following triplet losses are proposed, giving the hinge loss [7,11,12,15,16]:

$$\varphi(\delta_+, \delta_-) = \max(a + \delta_+ - \delta_-) \quad (4)$$

$$\varphi\left(\delta_+^2, \delta_-^2\right) = \max(a + \delta_+^2 - \delta_-^2) \quad (5)$$

We define $\rho_{sub}\left(\delta_+, \delta_-\right) = \delta_+ - \delta_-$, $\rho_{sub}^2\left(\delta_+^2, \delta_-^2\right) = \delta_+^2 - \delta_-^2$ to specify the performance of (4)(5). In Fig. 3 (a), we find pairs (δ_+, δ_-) fall on the intersection of the two surfaces have the same $\rho_{sub}\left(\delta_+, \delta_-\right)$. In other words, $\rho_{sub}\left(\delta_+, \delta_-\right) = \rho_{sub}\left(\delta_+ + \Delta, \delta_- + \Delta\right)$. The same principle applies to (5) in the Fig. 3(b). This is the scale uncertainty problem of triplet losses which is caused by the discontinuity of descriptor space[18].

The scale uncertainty problem of triplet losses is fatal to the feature matching in the SLAM process. For example, comparing to Fig 2 (a), in Fig 2 (b), δ_+ and δ_-increase simultaneously and $\rho_{sub}\left(\delta_+, \delta_-\right)$ keep same. Even in some cases, $\rho_{sub}\left(\delta_+, \delta_-\right) < \delta_+$, so we can't distinguish whether the sample belong to positive samples or negative samples. In this case, the mismatching of features in the SLAM process will happen.

B. *Adaptive-Scale Triplet Loss*

To ensure that the matching descriptors have smaller distance than non-matching descriptors in SLAM, we need to solve the scale uncertainty problem by designing special loss function. To explain our loss clearly, we should introduce the

loss function of [18] first:

文献18的损失函数

$$\theta_{loc} = \frac{1}{2}\left(\delta_+ + \delta_-\right) \tag{6}$$

$$\theta_{mix} = \frac{1}{2}\gamma\theta_{loc} + (1-\gamma)\,\theta_{glo} \tag{7}$$

$$smax(\delta_+, \delta_-) = \exp(\delta_+)/\left(\exp(\delta_+) + \exp(\delta_-)\right) \tag{8}$$

$$\begin{aligned} L_{(\delta_+,\delta_-)} = &-\frac{1}{2\delta}\log(smax(2\delta\left(\theta_{mix} - \delta_+\right), 0)) \\ &-\frac{1}{2\delta}\log(smax(2\delta\left(\delta_- - \theta_{mix}\right), 0)) \end{aligned} \tag{9}$$

In (6) (7) (8) (9), θ_{mix} mixs the global context θ_{glo} and the local context θ_{loc}, γ represents the ratio of θ_{loc} in the mixed-context, δ is the key scale correction parameter. [18] combines θ_{glo} and θ_{loc} to take advantage of the Siamese network and triplet network. However, [18] set the scale correction parameter δ and θ_{glo} manually, which will cause several major shortcomings, firstly, as [18] descripted, an unreasonable θ_{glo} is easy to cause the network to diverge; secondly, we need set different θ_{glo} and δ in different datasets when we train the network, which increase the work for traning and can not guarantee the performance of the network at the same time.

根本解决方案是找一个多元方程，对于每一个loss值有独特解，但是这不容易。我们可以利用特殊的损失函数引导网络去注意样本的尺度变化

As are descripted in Sec A, the fundamental solution is to find a multivariate function, which has a unique solution for each loss value. However, it may be impractical to find such a loss function which satisfies the requirements of the network model, but we can guide the network to notice the scale changes of samples with special loss function. In this paper, we introduce a scale reminder,which is depicted as:

$$\xi = \frac{\delta_-}{\delta_+} \tag{10}$$

Supposing that when we keep $\rho_{sub}\left(\delta_+, \delta_-\right)$ same between (a) and (b) in Fig. 2, $\xi_a > \xi_b$. So the scale reminder ξ can reflect the changes of the scale. Based on this the scale reminder, we designed our loss function, our function consists of two parts: adaptive-scale loss, loss of correlation penalty for descriptor compactness. We will describe each loss in detail below.

Firstly, we proposed our adaptive-scale loss as follows:

我们的损失函数

$$\theta = \frac{1}{2}\left(\delta_+ + \delta_-\right) \tag{11}$$

$$\begin{aligned} T_{(\delta_+,\delta_-)} = &-\frac{1}{2\xi}\log(smax(2\xi\left(\theta_{mix} - \delta_+\right), 0)) \\ &-\frac{1}{2\xi}\log(smax(2\xi\left(\delta_- - \theta_{mix}\right), 0)) \end{aligned} \tag{12}$$

In above equations (10) (11) (12), we abandon global context, more importantly, we take ξ as the scale correction parameter which can adjust adaptively in network training. In addition, we plot our adaptive-scale loss function, loss function in [18], the common triplet loss function and their derivatives in three-dimensional space illustrated in Fig. 4. It seems unlikely that a loss function that contains δ_+ and δ_- and has a unique solution (δ_+, δ_-) at the same loss value.

只要δ->δ+，对于有这个商因子的，则对于一个loss不可能找到三个一样的值

We can assume that during the model training, when the loss value drops to a certain value L_0, theoretically, at this time, we cannot determine the effect of the model, because there are multiple solutions for the loss value L_0 which will generate different performance. However, we want the model to get a smaller δ_+ and a larger δ_-. As is known, in the process of model training, the decisive factor for guiding model training is the gradient. We only can ensure that the gradient values of each point on the green line in Fig. 4 are different, and can guide the model to converge to the desired solutions. In (b)(c), we can see that when the loss value is the same, the corresponding $diff$ is almost the same, especially (c), $diff$ value is the same throughout the whole solution space. Therefore, it is very hard for (b) and (c) to get the desired performance. In (a), the above problems does not exist, the loss function (12) is more likely to guide the model to the desired result .

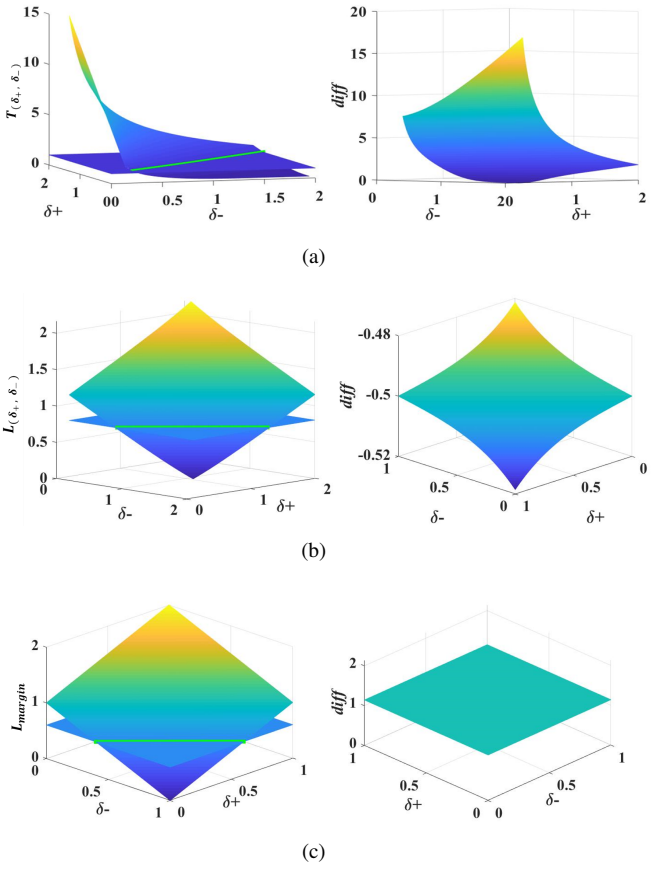


Fig. 4: An illustration of loss functions (left) and the sum of the squares of the derivative values corresponding to each loss function(right), $diff = \sqrt[2]{\frac{\partial L}{\partial \delta_+}^2 + \frac{\partial L}{\partial \delta_-}^2}$, the green line is the set of points corresponding to the same loss value, (a) and (b) are the loss functions of (12) and (9), (c) corresponds to the traditional triplet loss with a const scale, $L_{margin} = \delta_+ - \delta_- + a_{const}$.

Secondly, to make the descriptors more differentiable, we adopt the correlation matrix which descripted in [6] to calculate loss of correlation penalty for descriptor compactness. We use the anchor samples. We define $Y_a = [y_1, \cdots, y_m, \cdots y_q]^T$ as the output of the q anchor samples after propagation through the network. The correlation matrix

让描述子更可微，采用了correlation matrix来计算描述子紧密度的correlation penalty loss。
看中descriptor compactness是为了让descriptor的所有维度的数据有更少的关联性，这样就可以用更少的维度代表更多的信息，或者同样的维度表达更多的信息

只用anchor的输出来计算compactness

$R = [r_{mn}]_{q\times q}$, $1 \le m \le q$, $1 \le n \le q$ is defined as

$$r_{mn} = \frac{(y_m - \bar{y_m})^T (y_n - \bar{y_n})}{\sqrt{(y_m - \bar{y_m})^T (y_m - \bar{y_m})}\sqrt{(y_n - \bar{y_n})^T (y_n - \bar{y_n})}} \quad (13)$$

Where $\bar{y_m}$is the mean of the m-th row in Y_a. So the loss of correlation penalty is:

$$C_{orr} = \frac{1}{2}\left(\sum_{m\neq n}(r_{mn})^2\right) \quad (14)$$

To sum up, $T_{(\delta_+,\delta_-)}$ is the adaptive-scale triplet loss ,C_{orr} is the correlation penalty loss. The total loss is $T_{(\delta_+,\delta_-)} + C_{orr}$.

C. *Adaptive-Scale Sampling*

因为kesi越小，则表示δ+和δ-月接近，则不易分类所以需要关注这些kesi小的样本

Reviewing our analysis of the reminder of scale ξ, ξ has a certain directive effect on the descriptor spatial scale problem. The smaller ξ indicates unsuitable scale, so we should focus on these samples with smaller ξ. The specific rules are as follows. First of all, a batch $Z = \{(a_i, p_i)\}_{i=1}^{N}$ of matching local patches is generated, N represents the size of a batch, a_i stand for the anchor samples and p_i are the matching of a_i. When Z pass through the network, then we get the output $Y_a = [y_{a1}, \cdots, y_{am}, \cdots, y_{aN}]^T$,$Y_p = [y_{p1}, \cdots, y_{pm}, \cdots, y_{pN}]^T$. We calculate the L2 pairwise[6] distance matrix $D = [d_{ij}]_{N\times N}$ $(1 \le i \le N, 1 \le j \le N)$.

这个跟论文[6]不太一？？？

$$d_{ij} = \sqrt{2 - 2y_{ai}2y_{pj}} \quad (15)$$

行元素与行对角商

列元素与列对角商

For each diagonal element d_{ij}, we calculate separately $\xi_r(m) = \frac{d_{im}}{d_{ii}} (1 \leqslant m \leqslant N, m \neq i)$, $\xi_r(n) = \frac{d_{ni}}{d_{ii}} (1 \leqslant n \leqslant N, n \neq i)$, and $m_0 = argmin(\xi_r(m))$, $n_0 = argmin(\xi_c(n))$, so we can define a triplet sample:

代码里面好像没有用这个公式？？？

$$Trip(i) = \begin{cases} (a_i, p_i, p_{m_0}) \text{ if } min(\xi_r(m_0)) < min(\xi_c(n_0)) \\ (p_i, a_i, a_{n_0}) \text{ otherwise} \end{cases} \quad (16)$$

Eventually, we get a batch of triplet samples $S = \{Trip(i)\}_i^N$.

D. *Model And Phased Training*

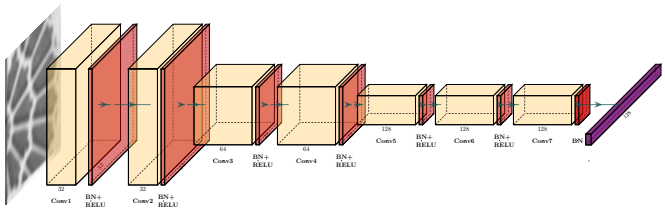


Fig. 5: An illustration of our network architecture , which is identical to L2-Net[6].

Our network architecture is shown in Fig. 5. The inputs of the model are the grayscale images which are also normalized. There are seven convolutional layers in the network. Except for the last convolutional layer, each convolutional layer is followed by ReLU non-linearity and BN (Batch normalization) operations. Before the last convolution layer, we added a drop out layer, mainly to prevent network overfitting. At the same time, we found that the dropout rate has a greater impact on the model. Here, we set the dropout rate to 0.3. The final output of the network is a normalized 128-dimensional feature vector. When it comes to the network training, we use a novel phased training strategy. Specifically, we set 10 epochs. In the first 6 epochs, we use $T_{(\delta_+,\delta_-)} + C_{orr}$ as the total loss. In the last 4 epochs, we use the sum of L_{margin} and C_{orr}. The main reason can be seen from Fig. 4 (a) that when the loss value of $T_{(\delta_+,\delta_-)}$ is close to the optimal value, the gradient decreases sharply. Therefore, in the late training period, we replace $T_{(\delta_+,\delta_-)}$ with L_{margin}. On the one hand, we took advantage of the ability of L_{margin} to speed up network training, and on the other hand, we solved the gradient disappearance problem of $T_{(\delta_+,\delta_-)}$. The network is trained with steepest gradient descent with a momentum term of 0.9 and we set the learning rate 0.1 in the beginning. From our experimental results, using two kinds of losses in different stages can produce better results.

hinge loss

用Lmargin替代Corr的原因是
1.当T接近最优值的时候，T的地图下降很快
2.Lmargin可以加速网络训练
3.解决了T的梯度消失问题

E. *Descriptor Performance Evaluation*

We evaluate our model on UBC benchmark dataset, which have three sets: Yosemite, Notredame, and Liberty with about 400k normalized 64x64 patches in each. According to standard rules, we train on one dataset and test on the other two. And our metric is the false positive rate (FPR) at point of 0.95 true positive recall. We compare popular hand-crafted and deep learning based descriptors following the same rules. Our results can be found in TABLE. I. From the data in the table, we can see that ASD outperforms other descriptors.

一个训练，两个测试

TABLE I: Patch Verification Performance On UBC Phototour ,Our Best Results Are In BOLD , Methods With Suffix"+" Are Trained With Data Augmentation.

Suquences	Train / Test	NOT	YOS	LIB	YOS	LIB	NOT
		LIB		NOT		YOS	
SIFT [1]		29.84		22.53		27.29	
MatchNet [11]		6.9	10.77	3.87	5.67	10.88	8.39
T-Feat [15]		7.22	9.53	3.12	3.83	7.82	7.08
[14]		4.55	7.40	2.01	2.52	4.75	4.38
L2Net+ [6]		2.36	4.7	0.72	1.29	2.57	1.71
HardNet+ [7]		2.28	3.25	0.57	0.96	2.13	2.22
CS-L2Net+ [6]		1.71	3.87	0.56	1.09	2.07	1.30
[18]		1.79	2.96	0.68	1.02	2.51	1.64
ASD		**1.43**	**2.60**	**0.53**	**0.79**	**1.99**	1.70

F. *Framework of ASD-SLAM*

Unlike those end-to-end SLAM methods, we still use the traditional SLAM framework. In the field of monocular vision, ORB-SLAM has the best effect in practical applications. So, we use our ASD to replace the front end of the traditional ORB-SLAM system to design our ASD-SLAM system framework. The overview of ASD-SLAM is shown in Fig. 6. At the front end of the system, we detect FAST corners, and get a certain number of image patches around the position of each corner in the image, then, send each

image patch to our deep learning network to get the corresponding feature descriptor ASD for every single frame. At the beginning, the system performs monocular initialization through the data association of descriptors between adjacent frames under epipolar constraints. In the tracking thread, we use a uniform motion model to assist the matching of feature descriptors and generate suitable keyframes at the same time. After getting the keyframe, the 3D positions of the feature points are traingulated by matching with the previous keyframe to build a local map. For each valid 3D point in the local map, a descriptor is maintained for the matching of consecutive frames, loopclosing, and the relocalisation process. In addition, once a loop is detected, the system will perform sim3 optimization, and then perform global pose optimization to reduce the cumulative error of the entire process. For loopclosing process, we adopt visual vocabulary method. We extracts a big set of ASD descriptors from training sets offline. Then, we train the vocabulary on DBOW.

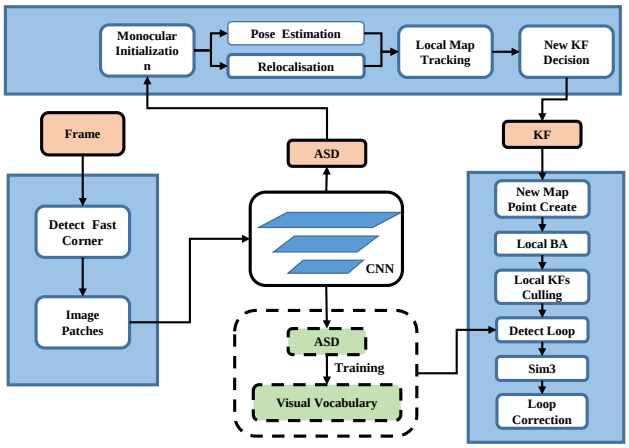


Fig. 6: The system overview of our ASD-SLAM.

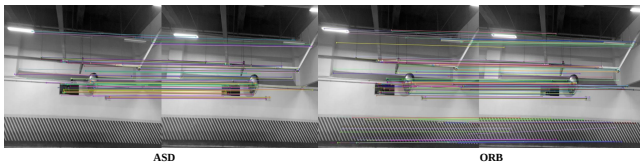
IV. EXPERIMENTAL RESULTS

In this section, we will present experiment results to demonstrate the effectiveness of ASD in practical SLAM systems and our ASD-SLAM has a better performance than traditional SLAM systems. Our experiment is divided into two parts: matching performance in special scenarios, evaluation using The KITTI odometry dataset.

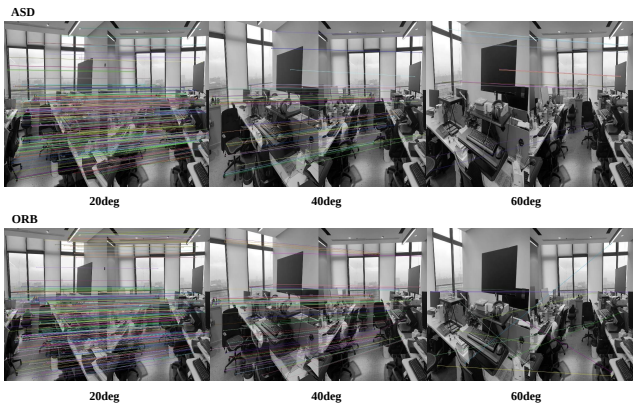
A. Matching Performance In Special Scenarios

Firstly, we compare the matching performance of our descriptors ASD to ORB which is used in ORB-SLAM system in a high repetition scenario like Fig. 7(a). In Fig. 7(a), traditional visual SLAM systems often fail because of high mismatches caused by highly repetitive lines on the wall like ORB in (a) even if we can do RANSAC. However, with our ASD descriptors, we can easily eliminate mismatches with RANSAC. The reason is that there are far more correct matches than mismatches with ASD. Secondly, we test the matching performance of descriptors from different perspectives. We set three different perspectives, 20 degrees, 40 degrees, 60 degrees. In Fig. 7(a), we can see that when the perspectives are 20 degrees, 40 degrees, ASD has more matches than ORB. When the perspectives are 60 degrees, ORB has no right matches, however, all of matches in ASD are right. From the above, it can be seen that ASD descriptors are highly robust in dealing with some challenging SLAM scenarios.

传统的ORB描述子在高重复场景时会有较多误匹配，导致RANSAC也无法纠正
ASD可以轻易利用RANSAC去除误匹配
ASD也能在大角度变化下正常工作



(a)



(b)

Fig. 7: Matching performance of ASD in special scenarios. (a) shows the matching performance of ASD and ORB in a high repetition scenario. (b) identifies the matching performance of ASD and ORB under different perspective changes. We set three different perspectives, 20 degrees, 40 degrees, 60 degrees.

B. Evaluation Using The KITTI Odometry Dataset

To verify the effectiveness of our ASD-SLAM system, we perform performance evaluations of localization and mapping on the KITTI odometry dataset. Firstly, we compare ASD-SLAM with LDSO and ORB-SLAM2 and performed sim3 trajectory alignment to the ground truth to computed the RMSE position error over the aligned trajectory on all the sequences of the KITTI odometry dataset. The RMSE results are shown in TABLE II with the best results highlighted in bold. At the same time, we use the method proposed by[35] to calculate the relative translation error statistics on sequences (seq. 00, 05, 07), and the results are shown in Fig. 9. Our method achieves comparable accuracy comparing to LDSO and ORB-SLAM2. In addition, in sequences 09, both LDSO and ORB-SLAM2 can not get loop closure in our tests, however, our ASD-SLAM can get loop closure normally. The reconstructed map of seq. 00 is shown in Fig. 10.

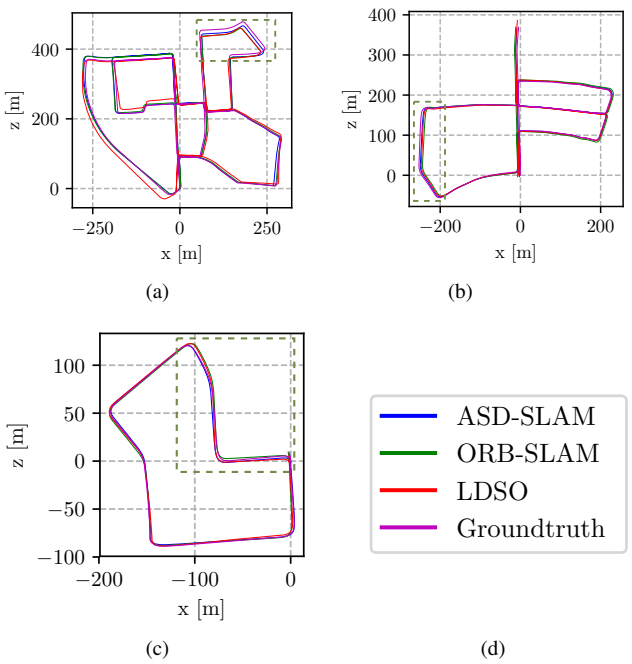


Fig. 8: An illustration of aligned trajectories of KITTI sequence 00, 05 and 07. As can be seen from the green box in the figure, the trajectories estimated by ASD-SLAM are closer to ground truth.

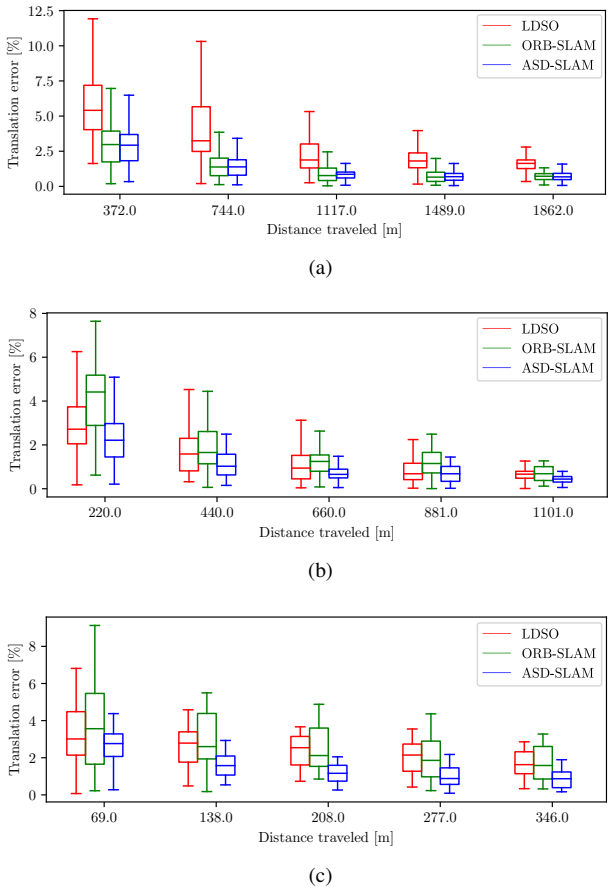


Fig. 9: Boxplot summarizing the relative translation error statistics with ASD-SLAM, ORB-SLAM and LDSO over sequences 00, 05 and 07. We divide the whole trajectory into five segments to calculate the relative translation error statistics by different ratios 0.1, 0.2, 0.3, 0.4, 0.5.

TABLE II: The ATEs Of ORB-SLAM, LDSO And ASD-SLAM On All Of KITTI Odometry Dataset Sequences.

Suquences	ORB-SLAM	LDSO	ASD-SLAM
00	8.18	13.80	**6.04**
01	416.11	**12.14**	208.16
02	24.62	22.60	**22.48**
03	1.35	2.92	**1.07**
04	1.16	1.15	**0.84**
05	6.09	4.58	**3.40**
06	10.26	13.23	**7.76**
07	2.80	2.23	**1.59**
08	56.52	128.68	**52.40**
09	53.35	76.37	**7.17**
10	8.65	17.08	**7.15**

前两个没能回环，ASD成功回环

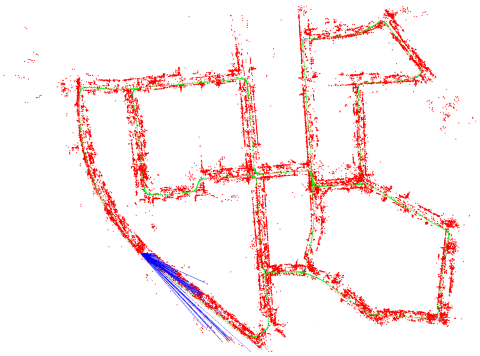


Fig. 10: An illustration of reconstructed map of seq. 00. In this map, the green line is the trajectory and the blue lines represent the observations in each keyframe.

提出一种尺度自适应损失函数和基于scale reminder的sampling（采样）策略 来生成更好的局部特征描述子（ASD），并设计了一。ASD个SLAM系统叫ASD-SLAM。因为不需要手动设定尺度修正参数，所以训练更容易。ASD对于视角变化大和高重复场景有优势

In this paper, we have presented a novel adaptive-scale loss function and sampling strategy based our scale reminder to generate better local feature descriptors (ASD) and design a SLAM system named ASD-SLAM based on our descriptor ASD at the same time. Firstly, the proposed loss function take the ratio between negative distance and positive distance as the scale reminder to make the network to overcome the scale uncertainty problem. Without setting the scale correction parameter manually, it is more easy to train the network. Comparing to other methods, ASD descriptor achieve state-of-the-art performance in the tasks of patch verification on the public dataset, at the same time, ASD descriptor can handle the challenging scenes like viewpoint variation and highly repetitive scenes. Secondly, we replaced the front-end feature extraction and feature matching of ORB-SLAM2 with ASD and designed the ASD-SLAM system. Our experiment results show that ASD-SLAM outperforms the traditional visual SLAM systems on KITTI Odometry Dataset,which proves ASD descriptors also have outstanding advantages than traditional hand-crafted descriptors in complex practical applications.

ACKNOWLEDGMENT

This work is supported by The National Key Research and Development Program of China under Project of 2017YFB0102503 and the National Natural Science Foundation of China under Project of 51605285.

REFERENCES

[1] Lowe D G . Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.

[2] Bay H , Ess A , Tuytelaars T , et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision and Image Understanding, 2008, 110(3):346-359.

[3] Rublee E , Rabaud V , Konolige K , et al. ORB: An efficient alternative to SIFT or SURF[C]// 2011 International Conference on Computer Vision. IEEE, 2012.

[4] Mur-Artal R , Montiel J M M , Tardos J D . ORB-SLAM: a Versatile and Accurate Monocular SLAM System[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.

[5] Simo-Serra E , Trulls E , Ferraz L , et al. Discriminative Learning of Deep Convolutional Feature Point Descriptors[C]// 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2016.

[6] Tian Y , Fan B , Wu F . L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[7] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in Advances in Neural Information Processing Systems, 2017, pp. 4826–4837.

[8] Z. Dai, X. Huang, W. Chen, L. He and H. Zhang, "A Comparison of CNN-Based and Hand-Crafted Keypoint Descriptors," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 2399-2404.

[9] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative evaluation of hand-crafted and learned local features," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 6959–6968.

[10] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A comparative study of sift and its variants," Measurement science review, vol. 13, no. 3, pp. 122–131, 2013.

[11] Xufeng Han, T. Leung, Y. Jia, R. Sukthankar and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3279-3286.

[12] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 4353-4361.

[13] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua and F. Moreno-Noguer, "Discriminative Learning of Deep Convolutional Feature Point Descriptors," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 118-126.

[14] Balntas V , Johns E , Tang L , et al. PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors[J]. 2016.

[15] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. Proceedings of the British Machine Vision Conference (BMVC), 2016. 1, 2, 7, 8

[16] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823.

[17] E. Hoffer and N. Ailon. Deep metric learning using Triplet network. International Workshop on Similarity-Based Pattern Recognition, 2015. 2, 3, 10

[18] M. Keller, Z. Chen, F. Maffra, P. Schmuck and M. Chli, "Learning Deep Descriptors with Scale-Aware Triplet Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2762-2770.

[19] Yan Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Washington, DC, USA, 2004, pp. II-II.

[20] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2874–2881, 2013.

[21] Y. Gao, W. Huang and Y. Qiao, "Local Multi-Grouped Binary Descriptor With Ring-Based Pooling Configuration and Optimization," in IEEE Transactions on Image Processing, vol. 24, no. 12, pp. 4820-4833, Dec. 2015.

[22] D. DeTone, T. Malisiewicz and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 337-33712.

[23] Ono Y , Trulls E , Fua P , et al. LF-Net: Learning Local Features from Images[J]. 2018.

[24] Yi K M , Trulls E , Lepetit V , et al. LIFT: Learned Invariant Feature Transform[J]. 2016.

[25] Bromley J , Guyon I , Lecun Y , et al. Signature Verification Using a Siamese Time Delay Neural Network[C]// Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]. Morgan Kaufmann Publishers Inc. 1993.

[26] E. Hoffer and N. Ailon. Deep metric learning using Triplet network. International Workshop on Similarity-Based Pat- tern Recognition, 2015. 2, 3, 10

[27] S. Wang, R. Clark, H. Wen and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 2043-2050.

[28] T. Zhou, M. Brown, N. Snavely and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6612-6619.

[29] C. Zhao, L. Sun, P. Purkait, T. Duckett and R. Stolkin, "Learning Monocular Visual Odometry with Dense 3D Mapping from Dense 3D Flow," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 6864-6871.

[30] A. Vakhitov and V. Lempitsky, "Learnable Line Segment Descriptor for Visual SLAM," in IEEE Access, vol. 7, pp. 39923-39934, 2019.

[31] C. Yu et al., "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 1168-1174.

[32] N. Brasch, A. Bozic, J. Lallemand and F. Tombari, "Semantic Monocular SLAM for Highly Dynamic Environments," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 393-400.

[33] T. Naseer, G. L. Oliveira, T. Brox and W. Burgard, "Semantics-aware visual localization under challenging perceptual conditions," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 2614-2620.

[34] A. Loquercio, M. Dymczyk, B. Zeisl, S. Lynen, I. Gilitschenski and R. Siegwart, "Efficient descriptor learning for large scale localization," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3170-3177.

[35] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 7244-7251.

[36] X. Gao, R. Wang, N. Demmel and D. Cremers, "LDSO: Direct Sparse Odometry with Loop Closure," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, 2018, pp. 2198-2204.