SITPOSE: A SIAMESE CONVOLUTIONAL TRANSFORMER FOR RELATIVE CAMERA POSE ESTIMATION

Kai Leng[1], *Cong Yang*[3*], *Wei Sui*[4], *Jie Liu*[1] *and Zhijun Li*[2,3*]

[1]School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
[2]School of Computer Science and Technology, Harbin Institute of Technology
[3]School of Future Science and Engineering, Soochow University
[4]Horizon Robotics

ABSTRACT

Relative Camera Pose Estimation (RCPE) aims to calculate the translation and rotation between two frames with overlapped regions, which is crucial to computer vision and robotics. This paper presents a novel siamese convolutional transformer model, SiTPose, to regress relative camera pose directly. SiTPose is distinguished in three aspects: (1) With a cross-attention feature extractor and a compact transformer encoder, extreme rotation errors ($> 150°$) are significantly reduced: from 9.7‰ with the state-of-the-art 8-Points to 1‱ on the 7Scenes dataset. (2) SiTPose is also robust to narrow-baseline cases (slight rotation angle and large translation between neighboring frames), while existing RCPE methods mainly focus on wide-baseline cases. (3) SiTPose can be flexibly extended to geometry-based vSLAM (namely SiT-SLAM) in a multi-threaded way to prevent tracking lost and scale ambiguity problems. Results on multiple datasets show that SiT-SLAM yields a marked improvement in robustness and localization accuracy in complex scenarios, e.g., RMSE error is reduced from 26.36m with the classic ORBSLAM3 method to 6.94m on the KITTI-09.

Index Terms— Relative Pose Estimation, SLAM, Camera Pose Estimation, Cross Attention

1. INTRODUCTION

Relative Camera Pose Estimation (RCPE) is a long-standing problem in computer vision and robotics, which refers to calculating the translation and rotation between two frames with overlapped regions. RCPE is also essential to visual Simultaneous Localization and Mapping (vSLAM). Notably, the tracking process of vSLAM is a series of RCPEs. In practice, the RCPE scenarios include wide-baseline and narrow-baseline cases in terms of overlapping regions and pose differences between two images. However, existing RCPE methods mainly focus on wide-baseline cases and barely cover the narrow-baseline case. Moreover, their robustness and localization accuracy are dramatically reduced in complex scenarios, including significant perspective changes, motion blur, texture-repeating, textureless regions, etc.

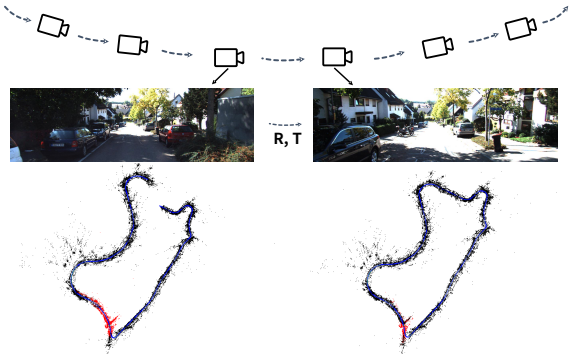

Fig. 1. Left: map of ORB-SLAM3[1]. Right: the tracking failure is fixed by our proposed SiTPose.

Traditional methods accomplished this task by extracting pixel-level interest points (e.g. [2, 3, 4, 5]) from a pair of input images, then establishing 2D-2D correspondences between keypoints and finally using 8-point[6] algorithm or less points[7] to recover the Fundamental matrix. RANSAC is very often used to eliminate wrong matches. However, these methods have limited effectiveness in handling complex environments such as low-texture, repetitive textures, exposure variations, etc. The ill performance of this technique makes point-based vSLAM prone to tracking failure in complex scenarios. To solve this problem, dual-SLAM[8] proposes a bidirectional SLAM strategy, which reduces tracking failures by a dramatic 88%. ORBSLAM3[1] makes it run for a long time by introducing orbslam-atlas[9], this way can greatly improve the robustness of SLAM, but the accuracy cannot be guaranteed. Overall, this methodology still depends on feature point matching and cannot effectively address the problem. Recently, learning-based methods have shown promising progress. For instance, RelPoseNet [11] and RPNet [12] achieve direct regress relative pose by proposing a siamese neural network upon a pose regression module. [13] maps the rotation to the probability distribution for the

*Corresponding authors.

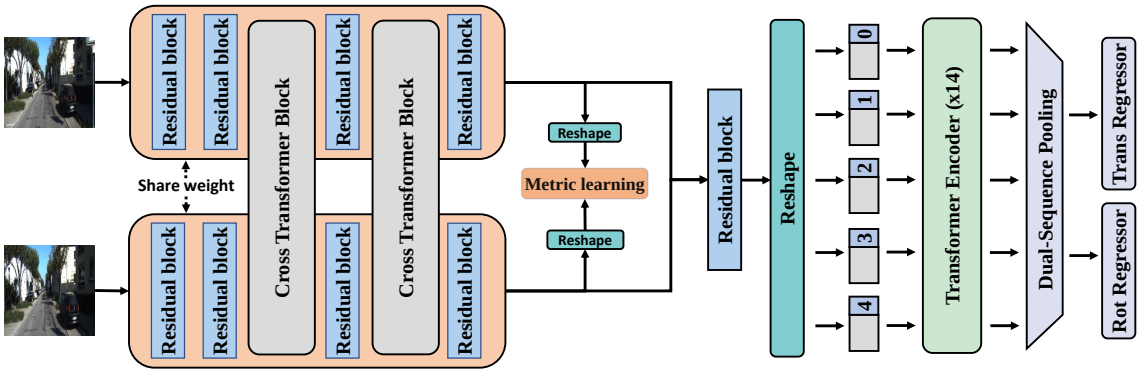

Fig. 2. SiTPose consists of three parts: a siamese feature extractor, a feature encoder based on CCT[10], and a pose regressor.

wide-baseline case. [14] introduces the 8-point algorithm as an inductive bias for relative Pose Prediction. In addition, [15] scale-consistent depth prediction by combining absolute scale and relative depth. [16, 17] using video sequences to predict camera pose. However, these methods still need to be more suitable for highly complex scenes, such as extremely wide-baselines (rotation $>$ 150 and with little overlapping regions, an example is shown in Fig. 3). An important reason is that the siamese structure adopted by the above methods lacks feature interaction in the process of feature extraction.

Wide- and Narrow-Baseline Case: The original meaning of the baseline refers to the distance between the optical centers of the two cameras in the stereo vision system. However, here we define two cases in terms of overlapping regions and pose differences between two images: wide-baseline cases and narrow-baseline cases. The wide-baseline refers to the rotation and translation between the image pairs that have large changes and the overlapping area is small. The narrow-baseline cases mean that the rotation between neighboring frames is slight but the translation varies greatly and the overlapping area is large. Unlike the wide-baseline case, the difficulty with the narrow one lies in the high similarity of the input image pairs. With the vigorous development of autonomous driving, the camera (vision-based perception) has become a mainstream sensor, and the narrow-baseline situation has become inevitable. However, existing methods are adverse to handling this situation (see Fig. 3) since they neglect the relevance between the image pairs. Their simple weight-sharing siamese network can not effectively distinguish different features of similar inputs, failing to estimate the narrow-baseline cases accurately. Based on the above observation, We propose SiTPose to directly regress the rotation and translation of a frame pair. With a siamese cross-attention feature extractor, it can effectively preserve image interactions at different scales. In addition, a modified attention-based feature encoder is added based on the siamese network to better leverage the cross information between image pair features. Finally, we introduce metric learning to guide the extractor in learning the different features of the input pair of images. By entirely using the interactive information between

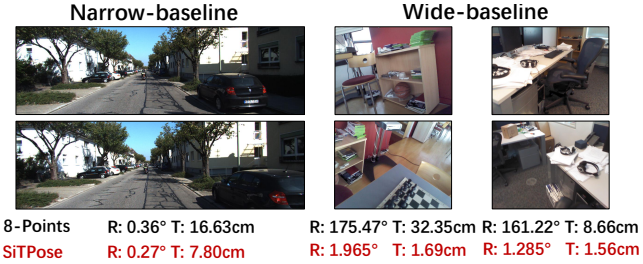

Fig. 3. Extreme cases. left: KITTI[18], right: 7scenes[19]. SiTPose has obvious advantages over SOTA 8-points[14].

image pairs, SiTPose achieves robust performance in various complex scenes.

Succinctly, the main contributions are as follows: 1) We propose a siamese framework, SiTPose, with a cross-attention feature extractor and a compact transformer encoder for RCPE, which yields highly robust and accurate to both wide- and narrow-baseline cases. 2) Benefiting from SiTPose's excellent prediction ability for extreme cases, we extended SiTPose to geometry-based Monocular SLAM and proposed SiT-SLAM, which addresses the problems of traditional vSLAM that are prone to tracking lost and scale ambiguity.

2. SITPOSE

2.1. Architecture

Our goal is to predict the rotation $R \in \mathrm{SO}(3)$ and translation $\mathbf{T} \in \mathbf{R}^3$ between two images with overlapping regions. This mission requires robust and accurate estimation of two related images under various conditions, such as wide-baseline cases and narrow-baseline cases. Therefore, it not only requires the model to be able to find the relationship between two images with low similarity but also to be qualified to distinguish and exploit the differences between two similar images.

The SiTPose architecture is illustrated in Fig. 2. Our model consists of three modules: a cross-attention feature extractor, a compact feature encoder, and a pose regressor.

Cross-attention feature extractor adopts a siamese net-

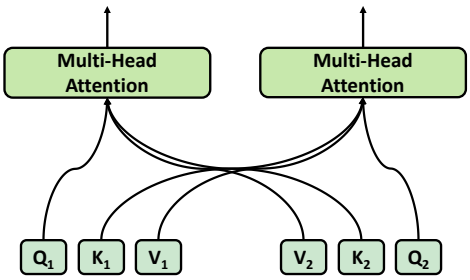

Fig. 4. Cross multi-head attention

work structure, utilizing the first seven layers of ResNet34 as the backbone to preserve sufficient feature information. To better extract the relevance between two images in the feature extraction process, we design a Cross Transformer Block (CTB). Similar to ViT[20], it consists of Layernorm, MLP blocks, and an attention layer. The main difference is that we proposed a cross multi-head attention (CMHA) layer to replace the traditional multi-head attention (MHA) layer [21]. We insert CTB as residuals at different convolutional levels to better interact with the two branches of the siamese network.

Compared with MHA, CMHA needs to accept two queries Q, two Keys K and two values V from two features as input (see Fig. 4) and output two refined features for the siamese backbone. It can be formulated as follows:

$$\begin{aligned} \boldsymbol{A_1} &= \text{softmax}\left(\mathbf{Q_{cur}}{\mathbf{K_{ref}}}^\top/\sqrt{D_h}\right)\mathbf{V_{ref}} \\ \boldsymbol{A_2} &= \text{softmax}\left(\mathbf{Q_{ref}}{\mathbf{K_{cur}}}^\top/\sqrt{D_h}\right)\mathbf{V_{cur}} \end{aligned} \quad . \tag{1}$$

where the subscripts cur and ref represent the two image features after convolution and D_h is the dimension of each head. After reshaping, $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ can be sent to the siamese network as residuals.

A Residual block (consisting of 3×3 convolutions, batch normalization, and ReLU activation) is added after the siamese structure for feature fusion. Meanwhile, we introduced metric learning at the end of the siamese network to guide the extractor to better extract different features.

In terms of **compact feature encoder** design, followed by CCT[20], we use standard learnable 1D position embeddings and a 14-layer transformer encoder (consisting of multi-headed self-attention layer, MLP layer, and apples LayerNorm before every block).

The output sequence of the transformer encoder contains relevant information across different parts of the input patches and preserving this information can improve performance. We proposed dual-SeqPool based on CCT. It fully preserves the relevant information of the original sequence and the encoded sequence. This module can be viewed as a learnable weight distribution method that transforms the output of the transformer encoder from $\mathbf{R}^{b\times n\times d}$ to $\mathbf{R}^{b\times 1\times d}$:

$$\begin{aligned} \mathbf{x}'_{in} &= \text{softmax}\left(\mathrm{f}\left(\mathbf{x}_{in}\right)^T\right) \in \mathbf{R}^{b\times 1\times n} \\ \mathbf{x}'_{out} &= \text{softmax}\left(\mathrm{g}\left(\mathbf{x}_{out}\right)^T\right) \in \mathbf{R}^{b\times 1\times n} \end{aligned} \quad . \tag{2}$$

where $x_{in} \in \mathbf{R}^{b\times n\times d}$ is the output of the feature extractor after reshaping, and $x_{out} \in \mathbf{R}^{b\times n\times d}$ is the output of the 14-layer transformer encoder. $\mathbf{f}$ and $\mathbf{g}$ are both linear layers which transfer $\mathbf{R}^{b\times n\times d}$ to $\mathbf{R}^{b\times n\times 1}$. Assigning these two weights to the output can be formulated as follows:

$$\mathbf{x} = \alpha\cdot\mathbf{x}'_{in}\mathbf{x}_{out} + (1-\alpha)\cdot\mathbf{x}'_{out}\mathbf{x}_{out} \in \mathbf{R}^{b\times 1\times d} \quad . \tag{3}$$

where $\mathbf{x}'_{in}\mathbf{x}_{out}$ and $\mathbf{x}'_{out}\mathbf{x}_{out}$ can be viewed as a weighted assignment of spatial order importance from the input and output of the transformer encoder, and α is a learnable coefficient that was initially set to 0.1.

The last part is the **pose regression** module, it predicts rotation (in quaternion form) and translation (in real units) through two fully connected layers.

2.2. Loss

Siamese network is usually applied to learn the similarity of two images, whereas we use it to extract the relevance and difference of relative images in our mission. In the vehicle case, narrow-baseline image pairs are often highly similar (as shown in Fig. 3 left). This situation generates similar feature extraction results from the two images by the siamese network and leads to a decrease in prediction accuracy. Inspired by [22], we design a loss function based on the contrastive loss to alleviate this problem. We use the pose difference as the margin to guide the feature extractor to increase the distance between the two groups of features as much as possible:

$$\begin{aligned} dif &= \|(q^*_{\text{ref}}\times q_{\text{cur}} - q_{\text{cur}})\|_2 + \|(t_{\text{ref}} - t_{\text{cur}})\|_2 \\ dis &= \|(f_{\text{ref}} - f_{\text{cur}})\|_2 \end{aligned} \quad . \tag{4}$$

where q_{ref}, q_{cur}, t_{ref} and t_{cur} are the rotation (in quaternion) and translation true values of the input two images, q^*_{ref} is the conjugate quaternion of q_{ref}. f_{ref} and f_{cur} are the outputs of the feature extractor flattened into 1D vector. dis is the Euclidean distance between f_{ref} and f_{cur}. Based on dis and dif, the loss function can be formulated as follows:

$$loss_{ML} = \frac{1}{N}\sum_{n=1}^{N}\max\{(\alpha\cdot dif - dis), 0\} \quad . \tag{5}$$

where α is a positive constant to balance dis and dif, and we set it to 16 and 4.5 for outdoor and indoor scenes empirically.

Our training process is divided into three steps. First, we use l2 MSELoss for training. Then, we use l1 Geodesic Loss for retraining after MSEloss stabilizes, where the Geodesic Loss is the magnitude of the vector from prediction to ground truth pose. Finally, we introduce $loss_{ML}$ to further improve the performance.

2.3. Implementation

Our source code will be released at https://github.com/cong-yang/SiTPose. The backbone (ResNet-34) was pre-trained using the ImageNet dataset and the rest weights of SiTPose are initialized by Xavier[23]. The backbone is then truncated into three parts as the feature extractors of three branches.

Each branch generates a feature map with a size of $512 \times 7 \times 7$. Inspired by the architecture of co-attention[24], we insert CTB as a residual at each truncation. After fusion, the feature extractor finally generates a $384 \times 4 \times 4$ feature map. Our feature encoder is implemented based on CCT-14. We use a single RTX 3090 with 24G GPU memory for training. The model is implemented in PyTorch, and lietorch was extended for backpropagation of Geodesic loss. We choose the Adam optimizer to train the network, and the weight decay is 10^{-5}. We train the network with a learning rate of 10^{-3} and 10^{-5} for MSE Loss and Geodesic Loss, respectively. The batch size is set to 64 and the whole training process takes around 36 hours in practice.

2.4. SiTPose with vSLAM

vSLAM is an incremental system, and RCPE can be naturally integrated into vSLAM as a part. We fuse SiTPose with traditional monocular vSLAM to solve two problems: 1. Tracking failures are prone to occur in complex environments. 2. monocular vSLAM scale ambiguity. We integrate SiTPose in a multi-threaded way on the basis of ORBSLAM3[1].

For tracking failure, ORBSLAM3 adopts the following strategy: the first step is relocation. Until relocation failure, a new submap will be created as an active map, and the previous map will be reserved as a nonactive map to wait for subsequent loop-closing detection to merge related submaps. However, in vehicle cases, this strategy usually fails, resulting in wasted frame information during relocation. In response to the above problems, we integrate SiTPose into ORBSLAM3 to solve the tracking failure problem.

Our processing flow is shown in Fig. 5. The starting points are where the tracking starts, and the breaking points are where the tracking lost occurs. Under our strategy, when tracking lost occurs, our model will quickly create a second submap by 2D-2D matching while relocating. When the number of keyframes in the second submap reaches 3, we deliver the last keyframe (KF1) of the previous submap and the first keyframe (KF2) of the current submap to the SiTPose thread for estimating rotation and translation. When the current submap is stable (the number of keyframes reaches 14), We set the pose of the KF2 :

$$T_{KF2} = T_1 T_2, \quad T_1, T_2 \in SE(3) \tag{6}$$

where T_1 is the pose of KF1, and T_2 is the result predicted by SiTPose. We subsequently propagate this modification to the remaining 13 frames of the second submap and merge the two submaps using the predicted pose from SiTPose.

To complete the fusion of the two submaps, a problem that has to be solved is the scale consistency of the two maps. The monocular feature point method (such as ORBSLAM3) usually uses the method of setting the median value of the triangulated restored spatial point to 1 for scale normalization. However, this method lacks a constant scale standard, which not only fails to restore the true scale but also causes the failure of map merging due to the inconsistency of the scale for

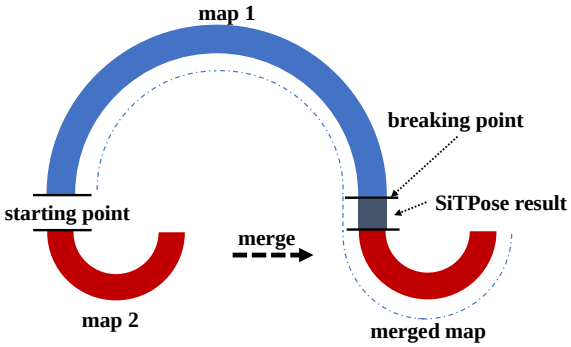

Fig. 5. SiT-SLAM process.

multiple submaps. We use SiTPose as the annotation for scale recovery during initialization and scale the median depth of the spatial point to the prediction result of SiTPose. This operation is shown as follows:

$$medianDepth = \frac{1}{\text{med}(z_1, z_2, ..., z_n)} \times \frac{D_{pre}}{D_{run}} \quad . \tag{7}$$

where $\text{med}(z_1, z_2, ..., z_n)$ is the median value on the z-axis of the spatial point recovered by triangulation, D_{pre} is the value of translation predicted by SiTPose, and D_{run} is the corresponding value of ORBSLAM3 at running time with D_{pre}. Finally, we integrate the above operations into ORBSLAM3, which greatly improves its ability to address tracking failures in the case of complex scenarios. The visualization results are shown in Fig. 6.

3. EXPERIMENTS

3.1. Dataset

We generate image pairs dataset on two two benchmark datasets: the 7-Scenes[19] and the KITTI[18] datasets. All images are resized to 224×224.

7-Scenes is a collection of tracked RGB-D camera frames of seven indoor environments. All scenes were recorded from a handheld Kinect RGB-D camera at 640 × 480 resolution. We use the same image pairs generation strategy as [25]. The generated dataset contains a large number of wide-baseline situations, and the mean values of rotation and translation are 29.7260° and 1.0468m, respectively.

KITTI is a large-scale outdoor autonomous driving dataset. Contains real image data collected from scenes such as urban areas, rural areas, and highways. It is the most authoritative and popular dataset in the field of visual SLAM. We generate image pairs for the narrow-baseline case on KITTI. Specifically, the interval between the image pairs we generate is 1-5 frames. In the generated dataset, the mean values of rotation and translation are 1.7947° and 2.8631m, respectively.

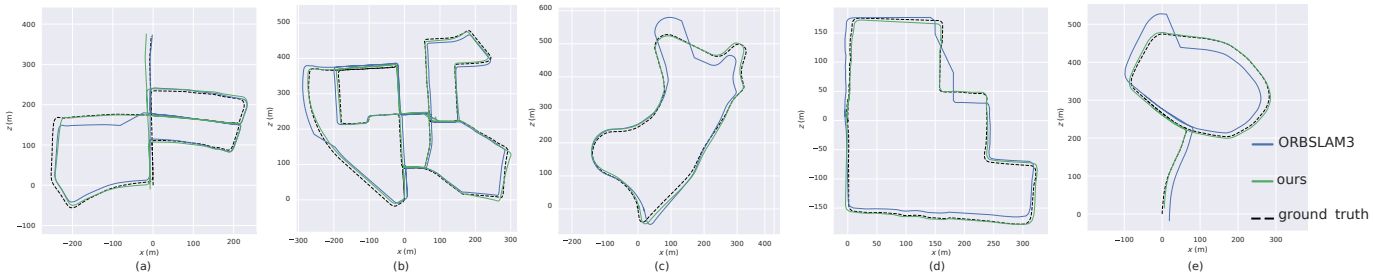

Fig. 6. The visual results of ORBSLAM3 and ORBSLAM3 with SiTPose on sequences 05, 00, 09, 15, and 18 of KITTI are presented in (a), (b), (c), (d), and (e), respectively. We set a breaking point on each sequence, based on the criterion that the number of matched feature points is insufficient.

3.2. Comparison and Ablation study

We compare SiTPose with four baselines. First, we compare with the classical methods RPNet[12] and RelPoseNet[11], these two methods both employ a simple Siamese network structure. The second is Planeformers[26], a sparse reconstruction method, it uses RCPE as a subtask, and we take its RCPE part out for comparison. The last one is the latest work [14], a ViT-based RCPE network using 8-point algorithm as inductive bias. We show the average translation and rotation errors on the KITTI and 7Scenes datasets (as shown in Table 1). Our method is far ahead of other baselines, especially in translation errors (on the 7Scenes dataset, it is less than 1 cm). We found that SiTPose's superior prediction accuracy mainly depends on its ability to handle extreme cases. Like the example shown in Fig. 3, there are 101 rotation errors greater than 150° in the 8-point[14] prediction results, while SiTPose has only one on the 7Scenes dataset.

Table 1. **Translation and Rotation Performance on** KITTI and 7Scenes, our method is the best among the five baselines.

	7Scenes		KITTI	
	R(deg)	T(cm)	R(deg)	T(cm)
RelPoseNet [11]	3.81	7.26	1.71	124.51
RPNet [12]	4.79	8.51	1.80	127.00
PlaneFormers [26]	1.81	5.06	0.51	13.11
8-points [14]	1.56	4.30	0.45	11.13
ours	**0.92**	**0.77**	**0.35**	**6.25**

We performed ablation experiments to prove the effectiveness of each module of our model. We use CNN Pose Regressor (a modified ResNet34 with the first seven layers extended as a siamese network) as a starting point for ablation experiments. All modules are introduced in Sec 2. The ablation study results are presented in Table 2. It can be seen that: (1) the performance increases when more components are used for RCPE, indicating the contribution of each part. (2) The geodesic loss based on Lietorch will guide the model to achieve better prediction accuracy. In addition, we provide the inference time of each module of SiTPose (see Table 3).

Table 2. **Ablations study on** KITTI and 7Scenes. Average translation error (°) and rotation error (cm) is reported.

	7Scenes		KITTI	
	R(deg)	T(cm)	R(deg)	T(cm)
CNN Pose regressor	1.42	1.46	1.31	12.93
+ Feature Encoder	1.26	0.98	0.76	8.16
+ CTB	1.17	0.94	0.76	7.64
Transfer to GeoLoss	0.94	0.77	0.38	6.26
+ Metric Learning	**0.92**	**0.77**	**0.35**	**6.25**

Table 3. Inference time of each module of SiTPose, reported from 1000 samples on RTX3090.

	SiTPose	Feature extractor		Feature	Pose
		CTB	resnet34	Encoder	Regressor
time(s)	12.82	1.27	7.33	4.06	0.05

3.3. SLAM with SiTPose

To demonstrate the superiority of monocular SLAM integrated with SiTPose in the face of complex situations, we compared SiT-SLAM with ORBSLAM3 on i5-12500. We selected a breakpoint on each sequence based on the number of successful feature point matches on the KITTI dataset. Quantitative analysis is shown in Table 4, and the visualization results are shown in Fig. 6. It can be seen that the advantage of SiT-SLAM lies in the recovery ability to tracking lost.

Table 4. **SiT-SLAM and ORBSLAM3** on KITTI dataset. We use RMSE and median error as criteria. The ground truth of sequence 10-18 are obtained by stereo ORBSLAM3.

Seq	SiT-SLAM		ORBSLAM3	
	median(m)	RMSE(m)	median(m)	RMSE(m)
00	**6.52**	**7.51**	10.47	12.13
05	8.56	**7.56**	**8.25**	12.76
09	**6.10**	**6.94**	20.39	26.36
13	**5.38**	**6.46**	13.71	14.34
15	**4.47**	**4.88**	8.69	14.44
16	**7.15**	**7.90**	8.03	10.34
18	**4.90**	**5.69**	9.39	10.58

4. CONCLUSION AND FUTURE WORKS

In this paper, we introduce SiTPose, a novel approach that exhibits comparable robustness in both narrow- and wide-baseline scenarios and significantly enhances accuracy compared to previous methods. The superior performance of SiTPose highlights the significance of interactive features for image pairs. Besides, we demonstrate the successful integration of learning-based RCPE with traditional vSLAM in SiT-SLAM, which greatly improved the robustness of monocular vSLAM in the face of complex situations. Future research avenue can be directed towards enhancing the speed of SiTPose and investigating better fusion strategies to improve its performance in challenging scenarios.

5. ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (Grant Number: 62072137) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant Number: 22KJB520008).

6. REFERENCES

[1] Carlos Campos et al., "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] David G Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.

[3] Herbert Bay et al., "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[4] Ethan Rublee et al., "Orb: An efficient alternative to sift or surf," in *ICCV*, 2011, pp. 2564–2571.

[5] Daniel DeTone et al., "Superpoint: Self-supervised interest point detection and description," in *CVPRW*, 2018, pp. 224–236.

[6] Richard I Hartley, "In defense of the point algorithm," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[7] Yaqing Ding et al., "Homography-based minimal-case relative pose estimation with known gravity direction," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 196–210, 2020.

[8] Huajian Huang et al., "Dual-slam: A framework for robust single camera navigation," in *IROS*, 2020, pp. 4942–4949.

[9] Richard Elvira et al., "Orbslam-atlas: a robust and accurate multi-map system," in *IROS*, 2019, pp. 6253–6259.

[10] Ali Hassani et al., "Escaping the big data paradigm with compact transformers," *arXiv preprint arXiv:2104.05704*, 2021.

[11] Zakaria Laskar et al., "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *ICCVW*, 2017, pp. 929–938.

[12] Sovann En et al., "Rpnet: An end-to-end network for relative camera pose estimation," in *ECCVW*, 2018, pp. 1–8.

[13] Kefan Chen et al., "Wide-baseline relative camera pose estimation with directional learning," in *CVPR*, 2021, pp. 3258–3268.

[14] Chris Rockwell et al., "The 8-point algorithm as an inductive bias for relative pose prediction by vits," in *3DV*, 2022, pp. 1–26.

[15] Zhiqiang Yan et al., "Desnet: Decomposed scale-consistent network for unsupervised depth completion," *arXiv preprint arXiv:2211.10994*, 2022.

[16] Tinghui Zhou et al., "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017, pp. 1851–1858.

[17] Ruonan Li et al., "Amgb: Trajectory prediction using attention-based mechanism gcn-bilstm in iov," *Pattern Recognition Letters*, 2023.

[18] Andreas Geiger et al., "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012, pp. 3354–3361.

[19] Jamie Shotton et al., "Scene coordinate regression forests for camera relocalization in rgb-d images," in *CVPR*, 2013, pp. 2930–2937.

[20] Alexey Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

[21] Ashish Vaswani et al., "Attention is all you need," *NIPS*, vol. 30, pp. 5998–6008, 2017.

[22] Qing Li et al., "Relative geometry-aware siamese neural network for 6dof camera relocalization," *Neurocomputing*, vol. 426, pp. 134–146, 2021.

[23] Xavier Glorot et al., "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[24] Jiasen Lu et al., "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *NIPS*, vol. 32, 2019.

[25] Iaroslav Melekhov et al., "Relative camera pose estimation using convolutional neural networks," in *ACIVS*. Springer, 2017, pp. 675–687.

[26] Samir Agarwala et al., "Planeformers: From sparse view planes to 3d reconstruction," in *ECCV*. Springer, 2022, pp. 192–209.