RelViTNet: Relative Camera Pose Estimation Network Using Vision Transformers

Asmaa Loulou
Faculty of Engineering and Natural Sciences
Sabanci University
Istanbul, Turkey
aloulou@sabanciuniv.edu

Mustafa Unel
Faculty of Engineering and Natural Sciences
Sabanci University
Istanbul, Turkey
munel@sabanciuniv.edu

***Abstract*—Relative camera pose regressors estimate the relative pose between two cameras from two input images. A convolutional network with a multi layer perceptron head is usually trained per scene with ground truth relative poses. However, such methods are still suffering from limited accuracy and generalization. Inspired by the success of vision transformers on computer vision tasks, we propose to learn relative pose between two cameras using only vision transformer backbone with fully connected layers. The multiheaded self attention mechanism of the vision transformer allows our model to attend to the full image even from the lowest layers which further enables our model to learn the layout of the scene and focuses only the features that are relevant to our task. We evaluate our model on one outdoor and two indoor datasets. We show that our model achieves new competitive accuracies for both outdoor and indoor multi-scene relative localization benchmarks. We further compare our pose estimation results to those obtained using recent local keypoints based approaches and we show that our model outperforms these methods particularly for frames with small translation, where such methods mostly fail.**

***Index Terms*—Localization, Deep Learning, Vision Transformers**

I. INTRODUCTION

Estimating the relative pose between two cameras is a crucial task in many computer vision applications, such as structure from motion (SfM), simultaneous localization and mapping (SLAM) and visual odometry. Traditionally, this task can be accomplished by extracting and matching sparse or dense keypoints and then use the 2-D correspondences to estimate the essential matrix using 5-points or 8-point algorithms with RANSAC to reject outliers in a robust manner [1]. The performance of such methods is highly dependent on finding enough keypoints between two images and performing matching accurately. These methods fail on textureless scenes due to few correspondences and on scenes with repetitive structures or large view point changes due to noisy correspondences. Moreover, such methods calculate the translation vector up to a scale and fail when the the camera movement is purely rotational or with small translations. Recently, Sarlin et al. [2] trained an attentional graph network to perform matching between keypoints from two frames. However, their method is still dependent on finding enough keypoints in both images. Therefore, textureless and repetitive structures are still challenging. Sun et al. [3] trained a Network to perform both dense keypoints detections and matching using the self and cross attention layers in transformer. However, they still need to estimate the essential matrix using 5-points or 8-point algorithms. Thus, their translation vector is up to scale and their method fail when the essential matrix calculation fail due to pure rotation or small translations.

Estimating camera pose directly using deep learning models is shown to produce good results where feature detection methods fail [4]. Absolute pose regression using deep learning models are usually trained to predict the pose of a query image in a scene [5], [4], [6]. Relative pose regression models are trained to predict the relative pose between two input images [7], [8]. Each of these models is usually trained per scene. Deep learning methods still suffer from low accuracy and limited generalization. All the previous relative and absolute pose models used CNNs as a backbone to generate one or two global feature vectors which are used to regress the pose.

In this work, we introduce a new vision transformer (ViT) based relative camera pose regression network (RelViTNeT), motivated by the recent success of ViTs in various computer vision tasks such as image classification and segmentation [9], [10]. The self attention mechanism of ViT allows it to attend to the full image even from the lowest layers which further enables it to attend to the features of an image that are most important to a given task[10]. We propose using ViT as a backbone for our relative pose regression network without any CNN based features. We employ the plain vision transformer which was in introduced in [9] and we replace the classification head with a regression head. We finetune a pretrained ViT, which was trained as a classifier since ViTs usually require very large labeled image datasets. We evaluate our model on both 7-scenes and Cambridge Landmarks datasets which consist of multiple indoor and outdoor scenes and are commonly used for pose estimation models. We also evaluate our model on Scannet dataset which is a large indoor dataset used for training the local keypoints approaches such Superglue [2] and LoFTR [3]. We show that our model achieves better pose accuracy, particularly in challenging scenarios involving small translations. Moreover, our model estimates more accurate translation vectors in all

*This work was supported by Scientific and Technological Research Council of Turkey (TUBİTAK) under Grant 118C061.

cases. Our main contributions can be summarized as follows:

- We introduce the first completely ViT based architecture for relative camera pose regression in indoor and outdoor scenes without using any CNN based features.
- We show that our model achieves better translation estimation than the local keypoints approaches and an overall better pose estimation in scenarios involving small translations between the two input frames.

II. Related Work

A. Visual Localization

There are many methods available to perform visual localization.

Local keypoint-based approaches depends on 2D-2D correspondences between two images in order to find the relative camera pose. These methods usually have three separate phases: Feature detection, feature description and feature matching. Traditionally, hand crafted local features like SIFT [11] and ORB [12] have shown good performances and are widely used in computer vision applications. Deep learning approaches to find keypoints like SuperPoint [13] have also shown good performances. Lately, Sarlin et al. [2] trained an attentional graph neural network (SuperGlue) in order to perform matching after the keypoints are detected. Their approach showed better results than the traditional nearest neighborhood approaches or CNN based learning. However, enough and repeatable keypoints have to be detected first either using detection algorithms like SIFT or deep learning approaches like SuperPoint. In addition, the descriptors generated by SuperGlue are not position dependent which raises many challenges in featureless environments. Sun et al. [3] solved some of these challenges by using the self and cross attention layers in transformer to find descriptors based on the two input images which enabled them to find enough matches in low-texture areas [3]. However, their method (LOFTR) only performs detection and matching, one should still solve 8-point or 5-point equation using RANSAC in order to find the relative pose between two images which might fail in cases of pure rotation or when the view scene structure is planar.

Absolute and Relative pose regression absolute pose regression was first proposed by Kendal et al. [5] and was inspired by the CNN success in multiple computer vision tasks. Usually a CNN based backbone which is originally trained for image classification on big datasets such as imagenet is retrained to regress both translation and orientation of a camera from one query image. Relative pose regression methods retrain a CNN backbone with MLP heads in order to regress the relative pose between two input images. Melekhov et al. [7] used a pretrained CNN backbone with fully-connected layers (FCs) to estimate relative pose but they did not train the model for outdoor scenes. En et al. [8] proposed to use a network with two branches: a Siamese CNN based Network regressing one pose per image and a pose inference module for computing the relative pose. More recently, Yang et al. [14] proposed to use cycle-consistent adversarial training to predict the relative camera poses of image pairs based on training over synthetic environment data. Their model consists of Siamese CNN based Network (RASNET) with fully connected layers to regress pose. While their model showed good generalization over different scenes, the accuracy is still to be improved.

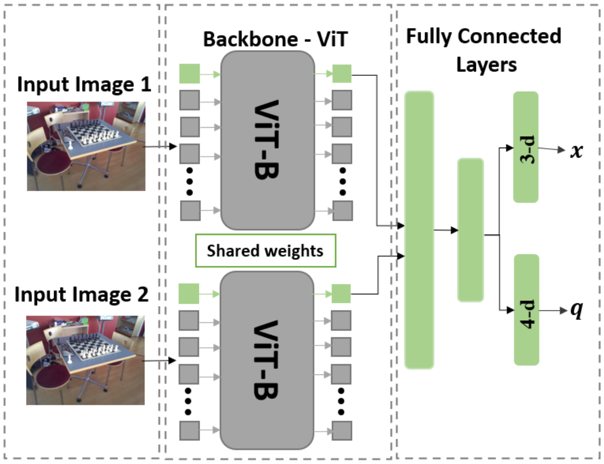


Fig. 1: **RelViTNet model architecture**: ViT backbone with 1 FC layer for Feature concatenation and 1 FC for dimension reduction. 2 more separate FC layers are used to regress 3D translation vector and 4D rotation vector.

B. Vision Transformers

Transformers have become the standard for sequence modeling in natural language processing due to their simplicity and computational efficiency. Recently, and after Dosovitskiy et al. [9] introduced the vision transformer which was trained for image classification and showed its ability to scale with data, ViTs are getting more attention in the computer vision tasks such as semantic segmentation [10] and image classification [9], [10]. Numerous versions of ViTs have been introduced with different training schemes.

In this work we use the plain ViT architecture that was introduced in [9] as a backbone to generate a global feature vector for each input image. We show that ViT can be trained for multiple scenes at once and can improve the relative pose estimation pipelines.

III. Method

A. Relative Camera Pose Estimation Model

RelViTNet is trained to estimate relative camera pose directly from 2 input images using one forward pass. The output is a pose vector $p = [t, q]$, which include 3-D vector for relative camera translation between two images t and 4-D vector for relative rotation q. Similar to [5], [15], [8], [14], we represent rotation using quaternions.

B. Architecture of RelViTNet

The network architecture as shown in Figure 1 uses the plain vision transformer architecture that was introduced in [9] as a backbone for extracting a global feature vector for each input image. ViT was introduced as a classification network which performs attention on images as follows: First an input image $I \in \mathbb{R}^{H \times W \times C}$, is divided into fixed size patches which are flattened into sequences $x \in \mathbb{R}^{N \times (P^2.C)}$ where H and W are the original image size and C is the number of channels. (P, P) is the resolution of each patch and $N = HW/P^2$ is the number of patches. A trainable linear projection layer $E \in \mathbb{R}^{(P^2.C) \times D}$ is used to embed all the sequences into D dimensions as shown in Equation 2. A positional embedding $E_{pos} \in \mathbb{R}^{N+1 \times D}$ is added after the concatenation of an extra learnable sequence x_{class} to the resulted embedded sequences.

$$z_0 = [x_{class}; x_1 E; ..., x_N E] + E_{pos} \tag{2}$$

The embeddings from Equation 2 are then passed through transformer encoder which consists of alternating layers $\ell = 0, 1, 2.., L$ of multiheaded self attention (MSA) and MLP blocks with layer normalization (LN) applied before each block and residual connection after each block [9], as shown by Equations 3 and 4.

$$z'_\ell = MSA(LN(z_{\ell-1})) + z_{\ell-1} \tag{3}$$

$$z_\ell = MLP(LN(z'_\ell)) + z'_\ell \tag{4}$$

The classification head was originally attached to the state of the extra learnable sequence at the output of the transformer encoder z_L^0 (shown as a green sequare in Figure 1) as the extra learnable sequence $z_0^0 = x_{class}$ serves as an overall image representation at the output of the transformer encoder. During training, the multiheaded self attention mechanism ensures that the extra learnable sequence is learned through performing attention between all the embedded patch sequences, thereby accounting for every patch in the image. This means that the transformer encoder is able to effectively integrate information from all patches of the input image, including the extra learnable sequence. We removed the original classification head and attached a regression head instead. Our regression head consists of 4 fully connected layers. The state of the extra learnable sequence of the output of the transformer encoder is 768-D vector for each image. We concatenate the feature vectors of the two images. Thus, The first fully connected layer is 1536-D which is connected to 128-D layer to reduce the dimension and reduce overfitting. Two more separate fully connected layers (3D and 4-D) are used to output relative 3-D translation vector and 4-D rotation vector.

C. Loss function

Training translation and rotation regressors simultaneously leads to better overall performance [5]. Therefore, we used stochastic gradient descent in order to minimize both the translation and the rotation losses

$$\mathcal{L}(I) = \|\hat{x} - x\| \exp(-s_q) + s_q + \left\| \frac{\hat{q}}{\|\hat{q}\|} - q \right\| \exp(-s_x) + s_x \tag{5}$$

where $\hat{x}$ and $\hat{q}$ are the estimated translation and rotation vectors. x and q are ground truth translation and rotation vectors. We combine the two losses using the camera pose loss function suggested by Kendal et.al [5] s_q and s_x are learnable parameters controlling the balance between the rotation and the translation losses.

IV. Datasets

We used the The Cambridge Landmarks dataset [5] to train and test our model for outdoor scenes. It is from an urban environment and consists of six medium-sized scenes ($\sim 900 - 5500 m^2$). Four of these scenes were used for testing and training. We used The 7-Scenes dataset [17] to train and test our model for indoor scenes. The dataset is from an office environment and includes 7 small-scale scenes ($\sim 1 - 10 m^2$). We further trained our model on Scannet dataset [18] which is a large indoor dataset from an office environment with an average floor area of ($\sim$ $22 m^2$). The dataset contains 2.5 million views in more than 1500 scans and is annotated with 3D camera poses.

V. Experiments

A. Experimental Setup

Since our model requires two input images to learn relative pose, similar to the process mentioned in [8] we sample training pairs while ensuring that each pair is from the same scene and that the two images in any pair have sufficiently overlapping field of view. Both datasets provide the absolute pose of the camera for all images. Thus, we generate the ground truth relative camera pose for each pair. From camera coordinates 1 to 2, we set R_{12} as the rotation matrix, and T_{12} as the translation vector and we calculated them as follows:

$$R_{12} = R_1^T R_2 \tag{8}$$

$$T_{12} = R_1^T (T_2 - T_1) \tag{9}$$

All images are resized to 224x224 with no random crops, normalized and mean-centered using standard deviation computed over the whole training set. The indoor network for both 7 scenes and Scannet datasets is trained with an initial learning rate of 10^{-4} which is gradually decreased 2 times after every 2 epochs while the outdoor network is trained with an initial learning rate of 10^{-5}. We used a batch size of 4 for both networks. All experiments reported in this paper were performed on an 16Gb NVIDIA GeForce RTX 3080 GPU.

B. Comparison with state of the art relative pose regressors

It is common to estimate the relative pose between an unknown query image and a known reference image with a ground truth pose [14]. Therefore, in order to be able to compare our results with PoseNet [5] and Structure-guidedNet [16] which are absolute pose regression networks, we computed the relative pose estimation accuracy as the absolute pose regression accuracy for the query image. We calculated the estimated absolute pose for the query image as follows:

$$\hat{R}_2 = R_1 \hat{R}_{rel} \tag{10}$$

TABLE I: **RelViTNet outdoor localization results for Cambridge Landmarks dataset.** We report the median rotation/translation error in degrees/meter for each method. Bold highlighting indicates better performance

Method	Kings College	Old Hospital	Shop Facade	St. Marys C.	Average[deg/m]
PoseNet[5]	5.40/1.92	5.38/**2.31**	8.08/**1.46**	8.48/2.65	6.84/2.45
RPNet[8]	3.12/1.93	4.81/2.41	7.07/1.68	**5.90**/2.29	5.22/**2.08**
RCPNet[14]	**1.72/1.80**	**3.09**/3.15	6.93/3.84	28.6/13.8	10.09/5.65
RelViTNet(ours)	2.63/1.93	3.73/2.69	**5.03**/1.94	7.80/**2.73**	**4.79**/2.32

TABLE II: **RelViTNet indoor localization results for 7Scenes dataset.** We report the median rotation/translation error in degrees/meter for each method. Bold highlighting indicates better performance

Method	chess	fire	heads	office	pumpkin	red kitchen	stairs	Average[deg/m]
PoseNet[5]	8.12/0.32	14.4/0.41	12.0/0.29	7.68/0.48	8.42/0.47	8.64/0.59	13.8/ 0.47	10.43/0.44
Structure-guidedNet[16]	8.44/0.10	11.7/0.26	13.3/0.16	6.63/0.16	5.05/0.16	6.32/0.20	9.65/0.27	8.72/0.19
RelPoseNet[15]	8.39/0.24	7.90/0.23	4.86/0.097	9.31/0.31	7.07/0.23	8.04/0.23	5.80/0.19	7.34/0.22
RCPNet[14]	**3.46/0.13**	9.45/0.31	9.87/0.15	**4.81/0.17**	**4.39**/0.22	**5.53/0.21**	7.24/0.26	6.39/0.21
RelViTNet(ours)	4.79/0.17	**6.31/0.19**	**2.67/0.10**	6.00/0.25	4.90/**0.20**	5.71/0.23	**3.77/0.18**	**4.87/0.19**

$$\hat{T}_2 = R_1 \hat{T}_{rel} + T_1 \tag{11}$$

where $\hat{R}_{rel}$ is the relative rotation matrix that resulted from the estimated rotation quaternion and $\hat{T}_{rel}$ is the estimated relative translation vector. R_1 and T_1 are the known rotation and translation of the reference image.

1) Outdoor Results: In addition to PoseNet, we compare our model to the available relative pose regression networks which are trained depending on pairs of images and ground truth poses with no depth information and that uses only two input images at inference time. We compare our outdoor results to PoseNet, RPNet and RCPNet. We report the average of median position and orientation errors in Table I. RelViTNet was trained across all scenes. RCPNet was trained across 3 scenes while St.Marys Church was not seen during training. PoseNet and RPNet are trained per scene. Test pairs were sampled from unseen sequences similar to [8]. RelViTNet exhibits an average increase of 52.53% and 58.94% for rotation and translation compared to RCPNet, an average increase of 3.10% and 25.59% for rotation and translation compared to RCPNet on three scenes (Kings., Old., Shop.), an average increase of 8.24% for rotation with a decrease of 11.50% for translation compared to RPNet, and an average increase of 29.30% and 5.31% for rotation and translation compared to PoseNet.

2) Indoor Results: Indoor results are shown in Table II. We report the average of median position and orientation errors. PoseNet and structure-guidedNet are absolute pose networks and are trained for each scene separately. RCPNet was trained across-scenes for five scenes (Chess, Fire, Heads, Pumpkin, and Stairs), while individually trained for Office and Red Kitchen. RelViTNet and RelPoseNet were trained across all scenes. All test pairs were sampled from unseen sequences in each scene with 30 frames difference between the two images in each pair. As shown even with 30 frames difference RelViTNet performs better than all others with an average 23.79% and 9.52% increase for rotation and translation compared to RCPNet, an average of 33.65% and 13.64% increase for rotation and translation compared to RelPoseNet, an average 44.15% increase for rotation compared to structure-guidedNet, and an average 53.31% and 56.82% increase for rotation and translation compared to PoseNet. Our results shows good generalization on the indoor dataset.

C. Comparison with state of the art Local keypoints based approaches

We compare our model to two local keypoint approaches: SuperGlue [2] and LoFTR [3]. SuperGlue uses a trained graph attention network (GAT) for keypoint matching and employs Superpoint [13] to detect keypoints. LoFTR [3] is a detector-free model, performing both detection and matching using cross and self-attention layers of the transformer. Similar to ViT, both models use the multiheaded self attention mechanism. However, the attention in ViT is being performed between all the patches of the image while the attention in such methods is performed between keypoints' descriptors. We compare our model performance to these approaches. We report the AUC of the rotation error for three thresholds (5°, 10°, 20°) where the rotation error is the angular error between the estimated rotation matrix and the ground truth. We also report the AUC of the translation error for three thresholds (0.1m, 1m, 10m) where the translation error is magnitude of the absolute difference between the estimated translation vector and the ground truth.

We first chose three test sequences from Scannet dataset. Theses sequences were not seen during training and consists of 1300 to 1950 frames. There is small translation between each two consecutive frames as the images where captured at 30 Hz [18]. Thus, we used our model to estimate the relative pose between each two consecutive frames in each sequence. We compare our model to both SuperGlue and LoFTR. As Table III shows, RelViTNeT outperforms other methods for both translation and rotation and for all sequences. Moreover, both superglue and LoFTR have multiple failures on such cases due to the small translation between frames and the use of 5-point or 8-point algorithm with RANSAC to estimate the

TABLE III: **RelViTNet localization results for Scannet dataset- Full sequences.** The AUC of the pose error in percentage is reported. Bold highlighting indicates better performance

Sequence (#frames)	Methods	Rotation estimation AUC			Translation estimation AUC			#Failures	Time (min)
		@5°	@10°	@20°	@0.1m	@1m	@10m		
1 (1390)	LoFTR	27.93	30.01	31.06	0.0	0.28	28.91	943	22.38
	SuperGlue	**73.60**	79.13	81.91	0.0	0.67	76.34	212	11.38
	RelViTNeT	72.60	**86.27**	**93.13**	**55.42**	**95.54**	**99.55**	**0**	**3.58**
2 (1945)	LoFTR	25.03	27.19	28.24	0.0	0.17	26.39	1375	30.73
	SuperGlue	69.60	74.24	76.56	0.0	0.45	71.02	411	16.45
	RelViTNeT	**71.26**	**85.59**	**92.92**	**54.48**	**95.43**	**99.54**	**0**	**3.9**
3 (1930)	LoFTR	35.69	39.88	40.40	0.00	0.22	37.79	1120	30.31
	SuperGlue	60.91	63.33	64.54	0.0	0.33	59.25	660	16.18
	RelViTNeT	**61.66**	**80.01**	**89.99**	**52.01**	**95.19**	**99.52**	**0**	**3.85**

TABLE IV: **RelViTNet localization results for Scannet dataset- 1500 test set.** The AUC of the pose error in percentage is reported. Bold highlighting indicates better performance

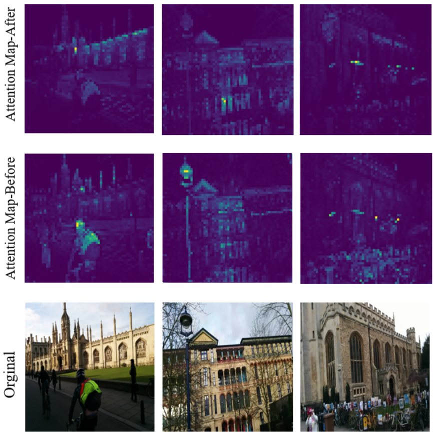
Methods	Rotation estimation AUC			Translation estimation AUC			#Failures
	@5°	@10°	@20°	@0.1m	@1m	@10m	
LoFTR	16.17	30.38	47.15	0.17	15.70	89.94	653
SuperGlue	**36.3**	**54.29**	**69.21**	0.92	33.57	89.29	47
RelViTNeT	23.38	27.53	38.38	**24.08**	**54.12**	**93.39**	**0**

pose from keypoint matches. In addition to having no failure cases, RelViTNeT is also much faster than both Superglue and LoFTR as it only needs one forward pass to estimate the pose without the need of any further optimization algorithms.

We also tested RelViTNeT on a test set of 1500 pairs sampled from multiple test sequences similar to the approach followed to test both SuperGlue [2] and LoFTR [3]. These test pairs are not necessarily consecutive frames. Table IV shows our results. While SuperGlue performs better on rotation estimation, RelVitNeT outperforms both methods on translation estimation. In addition, RelViTNeT outperforms LoFTR for the 5° rotation threshold.

D. Attention Maps Visualization and Interpretation

ViTs generate attention maps revealing how they focus on relevant image areas. Caron et al. [10] showed that after training, these maps explicitly depict scene layout and object boundaries. We visualize attention maps from the last block's self-attention modules in ViT backbones. Figure 2 shows the attention maps of the ViT backbone before and after training for the outdoor scenes. The attention maps before training are generated using pretrained ViT from [10] on a classification task. After retraining the model to regress pose, Figure 2 shows that the model learns the overall layout of the scene and is able to attend to the distinctive features of each scene. The model focuses on the main buildings that are specific to each scene and ignores the moving or temporary objects such as pedestrians. Figure 3 show the attention maps for the indoor scenes, more attention is placed on the distinctive features of each scene. For example, Figure 3 shows that model focuses mostly on the fire extinguisher for the fire scene and around the chess board area for the chess scene. Moreover, less attention is placed on the objects which are similar in more than one scene such as screens or white walls or floors. The overall layout of the scene can be inferred from the attention maps.



a) King's C. b) Old Hospital c) St Marys C.

Fig. 2: **Attention Maps for different scenes in the Cambridge Landmarks dataset.**

E. Ablation study

We conducted some further experiments in order to study how different backbones and global feature vectors may influence the results. We studied 4 different backbones: two convolutional based (RESNET) and two ViT based architectures. We altered the input size of the first fully connected layer in the regression head as different backbones have different sized feature vectors. All the other parts of the regression head are kept constant as shown in Figure 1 and the training was performed as mentioned in the experimental setup section. All the models we used were pretrained on imagenet dataset. As Table V shows, ViT based architectures greatly outperform

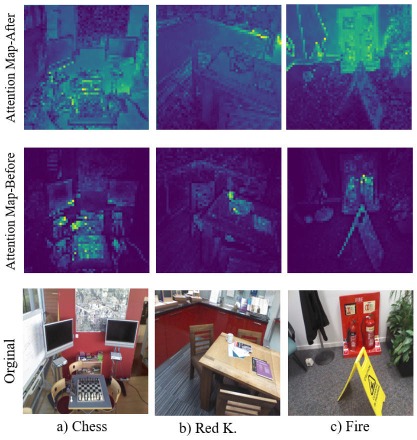


Fig. 3: **Attention Maps for different scenes in the 7 Scene dataset.**

TABLE V: Training the Network with different backbones. We report the median position/orientation error in meter/degrees for each method. Bold highlighting indicates better performance

Backbone	FC input	Rotation[degree]	Translation[m]
ViT-S	384	5.43	2.46
ViT-B	**768**	**4.79**	**2.32**
RESNET-18	512	8.49	3.93
RESNET-50	2048	11.32	5.11

RESNET models on both rotation and translation. ViT-B achieves slightly better results than its smaller version ViT-S. We argue that this is due to the attention mechanism that is used to aggregate image information through the layers of ViT and its ability to focus mainly on the relevant features to our task.

VI. Conclusions

In this work we introduced a new relative camera pose regression network using only vision transformer as a backbone with no CNN. We showed that the multiheaded self attention mechanism of the transformer encoder allows the network to attend to the parts of the image which are most important for pose estimation. Our network was able to improve rotation accuracy by 8.24% but reduces translation accuracy by 11.50% in outdoor scenes, while in indoor scenes, it improves rotation and translation accuracy by 23.79% and 9.52%, respectively. Moreover, we showed that our model outperforms local keypoint approaches when used on challenging cases with small translations and has no failure cases even when used on full sequences with up to 1900 frames. Our model is much faster than these approaches as it takes only up to 3.9 minutes to perform pose estimation between consecutive frames on a sequence with 1945 frames while the other methods takes at least 16.5 minutes long. This is due to the fact that our model uses no further optimization method and require one forward pass to estimate the relative pose. As future work, we will investigate how to improve the model in order perform more accurate relative pose estimations with fast motion and large rotations.

References

[1] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[2] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4937–4946.

[3] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "Loftr: Detector-free local feature matching with transformers," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8918–8927.

[4] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using lstms for structured feature correlation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 627–637.

[5] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[6] V. Balntas, S. Li, and V. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 751–767.

[7] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, 2017.

[8] S. En, A. Lechervy, and F. Jurie, "Rpnet: An end-to-end network for relative camera pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[10] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9630–9640.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–33 712.

[14] C. Yang, Y. Liu, and A. Zell, "Relative camera pose estimation using synthetic data with domain adaptation via cycle-consistent adversarial networks," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 4, pp. 1–17, 2021.

[15] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 920–929.

[16] Q. Li, R. Cao, K. Liu, Z. Li, J. Zhu, Z. Bao, X. Fang, Q. Li, X. Huang, and G. Qiu, "Structure-guided camera localization for indoor environments," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 202, pp. 219–229, 2023.

[17] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2013.

[18] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5828–5839.