Revisiting the Receptive Field of Conv-GRU in DROID-SLAM

Antyanta Bangunharcana Soohyun Kim Kyung-Soo Kim
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{antabangun, soohyun, kyungsookim}@kaist.ac.kr

Abstract

This work focuses on improving the Conv-GRU-based optical flow update within a DROID-SLAM framework. Prior optical flow models typically follow a UNet or coarse-to-fine architecture in order to extract long-range cross-correlation and context cues. This helps flow estimation in the presence of large motion and challenging image regions, e.g.*, textureless regions. We propose modifications to the Conv-GRU module which follows the rationale of these prior models by integrating (Atrous) Spatial Pyramid Pooling and global self-attention into the Conv-GRU block. By enlarging the receptive field through the aforementioned modifications, the model is able to integrate information from a larger context window, thus improving the robustness even when given inputs that comprise challenging image regions. We show empirically through extensive experiments the gain in accuracy through these modifications.*

1. Introduction

Simultaneous Localization and Mapping (SLAM) through visual sensors has been a long-studied research topic with potential applications in robotics [21, 22, 26], autonomous driving [18], and AR [25]. Visual-SLAM methods generally follow a bundle-adjustment (BA) approach whose objective is to find an optimal set of poses and pixel depths that minimizes the geometric error of matched image keypoints [5, 32, 34] or the photometric error from the raw pixel intensities [1, 15, 16]. However, issues may arise when the image lacks distinct keypoints or when the photometric consistency assumption is not met.

With the advent of deep learning techniques in achieving state-of-the-art results in many vision tasks, interests begin to emerge for applying deep models in order to tackle the aforementioned issues. However, unlike many other vision tasks, the nature of the SLAM problem comprises geometric formulation, which is rather difficult for deep neural networks to model [61] compared to classical geometric modeling. Recently, researchers began taking the direction of incorporating deep learning and classical solutions together [9, 45]. In particular, DROID-SLAM [48] proposed a full SLAM solution that integrates deep learning based optical flow estimation into a dense geometric BA formulation to solve for image poses and depths.

The optical flow formulation used in DROID-SLAM is based on the RAFT model [47], which iteratively updates the flow estimates across adjacent frames using a Convolution-based GRU [8]. RAFT samples from pre-computed cross-correlation values representing pixel similarities between neighboring frames and use Conv-GRU to compute the update to its hidden states. These cross-correlation values are computed at multiple pyramid levels, allowing the model to cover a wider spatial range of optical flow and thus, improving the performance in the presence of large motion. While the cross-correlation allows the model to attend to a larger motion of the target images, the Conv-GRU update operation is only performed via a small convolution layer in a window of size 3×3, thereby not exploiting the context cues from a larger window within the query image itself. This could lead to the difficulty of the model in images that contains challenging image regions, *e.g.*, textureless regions.

The advantage of RAFT lies in its iterative updates that are performed at a high resolution, in doing so, keeping the information of smaller objects intact. In contrast, many previous deep optical models follow a UNet [13, 37] or coarse-to-fine [35, 43] architectures, which allows for aggregation of the global contexts of the source in addition to covering large motion. However, these methods display poor performance for small objects and object boundaries due to the loss of information in the coarser stages. We are motivated by these works to get the best of both worlds in order to improve the Conv-GRU updates implemented within RAFT.

In this work, we propose modifications to the Conv-GRU cell and study the impact on the DROID-SLAM performance. We modify the 3×3 convolution using (atrous) spatial pyramid pooling (SPP) to compute the GRU hidden states updates. This enables the optical flow update to integrate from a large receptive field, thereby utilizing neighboring con-

text cues. More recently, Transformers have even been used to allow the model to attend to all the pixels in the source and target image to make flow predictions [56]. Inspired by this, we also propose the use of self-attention to allow Conv-GRU updates that attend to the global features. We empirically show that the improved receptive field of the model consistently improves the DROID-SLAM algorithm.

2. Related Works

2.1. SLAM

Modern Visual odometry (VO) and SLAM have evolved from filter-based approaches [10, 30] toward optimization-based methods [16, 25, 31, 33]. The optimization typically involves a bundle-adjustment (BA) of image points and poses. In geometric BA [5, 32, 34], optimization is performed over the reprojection error of points correspondences, commonly obtained via keypoint detector-descriptor scheme [3, 28, 38]. On the other hand, the photometric BA approach directly minimizes the photometric error between adjacent images [1, 15, 16].

With the success of deep-learning in many vision tasks [7, 20, 36], research toward a deep-learning-based full SLAM system is also being actively explored. As keypoint detection has proven successful in classical geometric BA, many research focused on constructing feature detection deep networks [11, 14, 58, 59] to be combined with classical approaches. These learned keypoint features show a more robust feature correspondence matching compared to their handcrafted counterparts. The recent surge in the adoption of Transformers [50] based methods towards vision tasks [6, 12, 27] have also sparked recent research which uses Graph Neural Networks [39] and Transformers [44, 56] for correspondence search. In particular, LoFTR and COTR use Transformers to attend to self and cross features globally without explicitly extracting feature keypoints. As their approach eliminates the keypoints extraction step, matching can be performed even in images with low-texture areas by exploiting global image contexts.

The early end-to-end deep-learning approach formulates the problem as a regression problem which takes as input a pair of images and outputs a 6-dimensional vector representing the Euler angles and translation of the relative pose between the two images [24, 49]. This approach lacks geometrical constraints, limiting the model's ability to generalize to unseen scenes compared to classical methods. Following the classical photometric BA, feature-metric optimization has been explored for localization [40, 51–53] as well as BA [45]. These methods perform optimization, which minimizes a learned feature map and displays better generalization ability. Recently, DROID-SLAM [48] performed geometric BA on point correspondences that are iteratively computed by RAFT [47] based optical flow. The iterative updates of the optical flow correspondences assist the BA in converging toward the optimal solution.

2.2. Optical Flow

Optical flow is deeply related to the keypoint correspondence search task, and we discuss its recent deep learning based development. The early models [13, 23] based themselves on the UNet [37] architecture. Typical of many correspondence search tasks [2, 29], correlation layer is also incorporated in order to extract similarity values between candidate pixel correspondences. To reduce the computational cost, follow-up works construct models that performs computation in a coarse-to-fine approach [35, 43]. This allows optical flow computation in the presence of large motion, which is then hierarchically refined to a higher resolution. Such approaches also enable the network to aggregate information from a larger effective receptive field at lower resolutions. However, the flow output of these types of models is suspect to missing details of small objects. RAFT [47] takes a different approach and uses Convolution GRU to imitate iterative optimization of the flow prediction. In this work, we take inspiration from earlier work to enlarge the Conv-GRU receptive field to integrate from larger context cues.

3. Method

3.1. Preliminaries

As this work is based on the recent **DROID-SLAM** [48] method, we briefly review the relevant components in this section. DROID-SLAM is a deep-learning based SLAM method that performs geometric bundle adjustment by using optical flow between frames that is iteratively updated. The optical flow updates are built on top of a RAFT [47] framework, which utilizes GRU blocks [8] to imitate an optimization-based update. Given a query frame and a neighboring frame of interest, feature maps of each frame are extracted by using a siamese convolutional neural network with shared weights. Full correlation volume between all pixel pairs $\mathbf{C} \in \mathbb{R}^{H \times W \times H \times W}$ is then constructed using the extracted feature maps. Next, average pooling is performed to obtain a 4-level correlation volume pyramid in order to increase the cross-frame correlation receptive field. In DROID-SLAM, this is computed between all connected frames in a frame graph.

At update step k, the current estimates for inverse pixel depths of the query frame and relative pose is used to compute the flow f_k, which gives the estimated coordinates u' of the corresponding points in the target frame. Correlation values around the neighborhood $N(u')$ of this point are then sampled from the correlation volume pyramid. The update

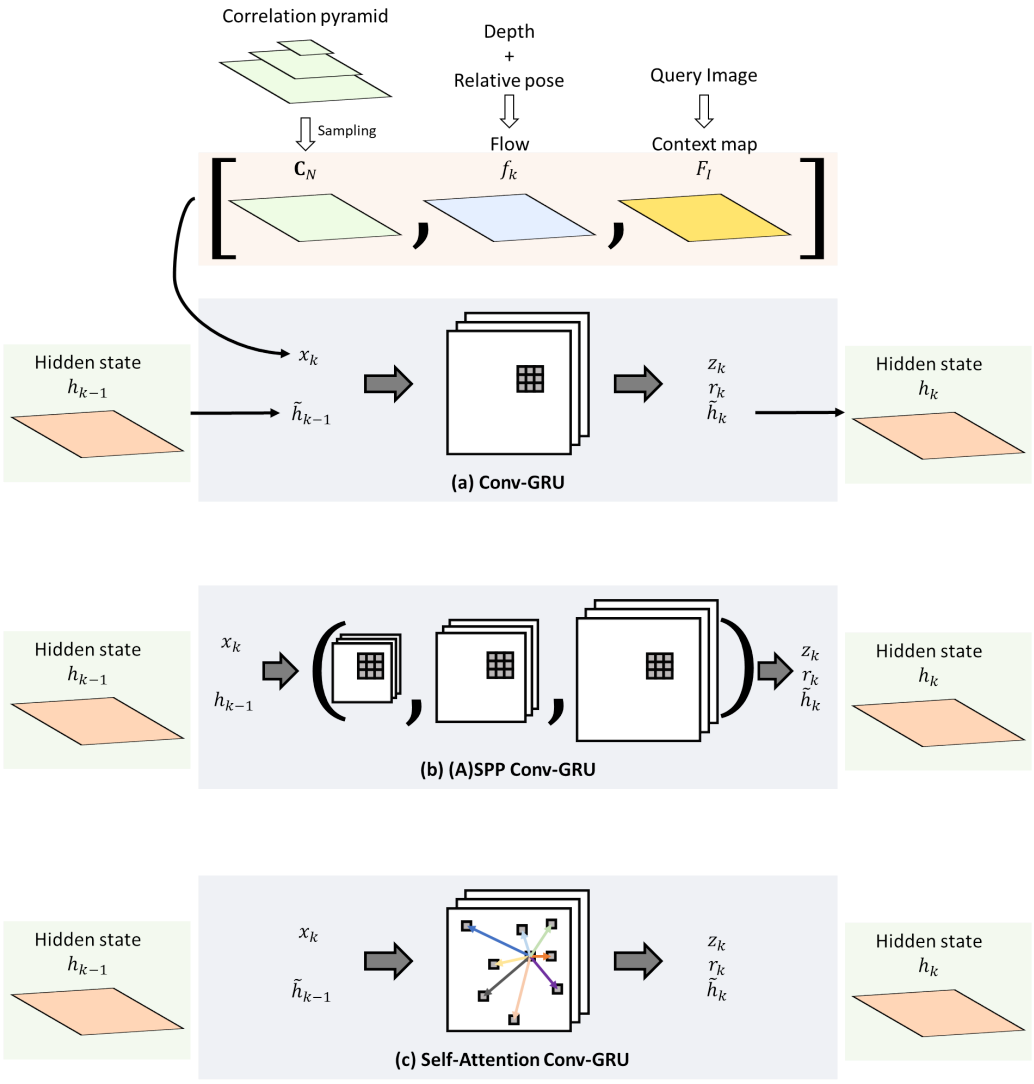


Figure 1. (a) The Conv-GRU used in DROID-SLAM. To increase the receptive field of the GRU-cell, we propose (b) (A)SPP and (c) global self-attention modification to the Conv-GRU.

input is computed using 3×3 convolution layer as

$$\begin{aligned} F_I &= g_c(I) \\ x_k &= [\text{Conv}_{3\times3}(\mathbf{C}_N, W_\mathbf{C}), \text{Conv}_{3\times3}(f_k, W_f), F_I] \end{aligned} \tag{1}$$

where F_I is a context feature map computed from the input query image. $[\cdot,\cdot]$ represents the concatenation of feature maps. The GRU update of the hidden state is computed as:

$$\begin{aligned} z_k &= \sigma(\text{Conv}_{3\times3}([h_{k-1}, x_k], W_z)) \\ r_k &= \sigma(\text{Conv}_{3\times3}([h_{k-1}, x_k], W_r)) \\ \tilde{h_k} &= \tanh(\text{Conv}_{3\times3}([r_k \odot h_{k-1}, x_k], W_h)) \\ h_k &= (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h_k} \end{aligned} \tag{2}$$

where σ is a sigmoid activation, and

$$h_0 = g_h(I) \tag{3}$$

is simply computed from the source image using CNN. The updated hidden state is used as inputs to CNN subnetworks that predict the residual flow

$$f_{k+1} = f_k + \text{Conv}_{3\times3}(h_k) \tag{4}$$

as well as the importance weights of each pixel, both of which are used as input into a geometric bundle adjustment module that optimizes for an updated depth and relative pose estimates.

As the correlation values $\mathbf{C}_N$ is obtained from 4-level pyramid volumes, this update computation utilizes a large

receptive field of cross-correlation information. However, the subsequent operation aggregates local information via a 3×3 kernel sized convolutional layers. This limits the spatial receptive field to capture the contextual cues from the query image. Fig. 1 (a) illustrates this process.

3.2. (A)SPP Conv-GRU

Context cues of the image has often been shown to benefit deep vision models in many tasks. In the optical flow problem, image contexts are especially helpful in image regions where matching values may be erroneous, such as in regions with repeated textures or textureless regions. Many deep optical flow models have a larger effective receptive field through UNet style architecture or a coarse-to-fine refinement design.

We explore the use of Spatial Pyramid Pooling (SPP) [19, 60] to increase the spatial receptive field of the Conv-GRU. Specifically, we modify the update candidate $\tilde{h_k}$ of Eq. (2) as

$$\begin{aligned} s_k &= \text{Conv}_{1\times1}([r_k \odot h_{k-1}, x_k]) \\ \tilde{h_k} &= \text{SPP}(s_k) + s_k \end{aligned} \tag{5}$$

The SPP module splits the feature map into multiple branches that resize the input into different scales (Fig. 1 (b)). We use average pooling to resize the feature maps into $(1, 1/2, 1/4, 1/8)$ the scale of the input resolution. Each feature map is then aggregated via a 3×3 convolution layer and bilinearly interpolated to the original resolution. They are then concatenated and passed into a 1×1 convolution layer. A residual connection [20] is incorporated in Eq. (5) to aid gradient flow. We also experimented with the Atrous variant [7] of SPP, wherein dilated convolution is applied to replace the above pooling and upsampling steps. Instead of resizing the input features, 4 branches of convolutional filters, each having a dilation rate of $(1, 1/2, 1/4, 1/8)$ are applied.

3.3. Self-attention Conv-GRU

Inspired by the success of recent correspondence search works by using Transformers to attend to global features [44, 56], we investigate attention-based updates of the Conv-GRU. We are only interested in applying self-attention to improve the effective receptive field within the query image itself. The candidate update value can be computed as

$$\begin{aligned} s_k &= \text{Conv}_{1\times1}([r_k \odot h_{k-1}, x_k] + pe) \\ \tilde{h_k} &= \text{Att}(s_k) + s_k \end{aligned} \tag{6}$$

where the self-attention layer Att is computed as

$$\begin{aligned} Q = \text{MLP}(x), K = \text{MLP}(x), V = \text{MLP}(x) \\ \text{Att}(x) = \text{MLP}(\text{softmax}(QK^{\text{T}})V) \end{aligned} \tag{7}$$

Name	Conv-GRU SPP	ASPP	Self-Attention	Strided
DROID				
Strided				✓
SPP	✓			✓
ASPP		✓		✓
Self-Att			✓	✓

Table 1. Model Configurations

The self-attention mechanism enables the network to assign weights to each pixel on the image according to the query Q and key K (Fig. 1 (c)). Typically, multiple heads of the above self-attention operations are performed in practice. In addition, we add positional encoding term pe into the update input in Eq. (6). The self-attention operation treats the data as an unordered sequence, so a positional encoding is necessary to give position cues to the network. In our implementation, we follow previous works and use a learned positional encoding [12]. Nevertheless, note that this learned positional encoding is specific to the image size. To use this positional encoding during test time, the input image has to be resized to match the training image size.

3.4. Strided implementation

The proposed (A)SPP and self-attention modifications into the Conv-GRU block increase the memory consumed by the model. In our experiments, we use the proposed modules with a strided implementation of the Conv-GRU cell to mitigate the issue of memory requirements.

We modify the input to the GRU cell x_k (Eq. (1)) using convolutions with stride $s = 2$

$$\begin{aligned} F_{I_{down}} &= g_{c_{down}}(I) = \text{Conv}^{s=2}_{3\times3}(g_c(I)) \\ x_k &= [\text{Conv}^{s=2}_{3\times3}([\mathbf{C}_N, W_{\mathbf{C}}), \text{Conv}^{s=2}_{3\times3}([f_k, W_f), F_{I_{down}}] \end{aligned} \tag{8}$$

We now have the input to the GRU cell x_k at a lower scale. The following GRU updates then follow Eq. (2), with the hidden state in Eq. (2) computed at a downsampled resolution as well:

$$h_{0_{down}} = \text{Conv}^{s=2}_{3\times3}(g_h(I)) \tag{9}$$

This allows the update operations with a reduced overall memory requirement of the system while also increasing the effective receptive field. As the hidden states are now represented at a lower resolution, we upsample the hidden states via bilinear interpolation followed by a 1×1 convolution prior to the residual flow and importance weight computations.

4. Experiments

4.1. Experiment Configuration

We are interested in empirically evaluating the effectiveness of the discussed Conv-GRU modifications toward the

Validation set	ATE				
	DROID	Strided	SPP	ASPP	Self-Att
abandonedfactory/Easy/P011	2.32	0.13	0.34	**0.11**	0.13
abandonedfactory/Hard/P011	1.45	0.22	7.30	**0.07**	0.24
abandonedfactory_night/Easy/P013	**0.02**	0.46	1.38	0.64	0.04
abandonedfactory_night/Hard/P014	0.30	**0.10**	1.96	1.19	0.17
amusement/Easy/P008	0.16	0.16	**0.08**	0.32	0.29
amusement/Hard/P007	**0.06**	**0.06**	0.14	0.19	0.07
carwelding/Easy/P007	**0.03**	0.05	0.04	0.04	**0.03**
endofworld/Easy/P009	0.06	0.32	**0.04**	0.05	0.24
gascola/Easy/P008	0.29	0.45	**0.21**	0.22	0.64
gascola/Hard/P009	0.43	0.51	**0.39**	0.68	0.80
hospital/Easy/P036	**0.01**	0.03	0.32	0.02	**0.01**
hospital/Hard/P049	0.04	0.01	0.02	0.01	0.01
japanesealley/Easy/P007	**0.02**	0.06	**0.02**	0.04	0.03
japanesealley/Hard/P005	0.01	0.01	0.01	0.01	0.02
neighborhood/Easy/P021	**0.20**	0.47	2.18	0.49	0.66
neighborhood/Hard/P017	0.06	0.05	0.27	0.04	**0.03**
ocean/Easy/P013	0.19	0.17	0.16	**0.10**	0.11
ocean/Hard/P009	1.56	0.67	**0.49**	0.61	0.74
office2/Easy/P011	0.02	0.02	**0.01**	0.02	0.02
office2/Hard/P010	0.09	0.16	0.05	0.05	0.05
office/Hard/P007	**0.00**	0.01	0.01	0.01	0.01
oldtown/Easy/P007	0.17	1.60	0.25	0.32	0.30
oldtown/Hard/P008	17.26	4.62	9.93	3.20	**3.13**
seasidetown/Easy/P009	0.11	0.09	0.12	0.08	**0.07**
seasonsforest/Easy/P011	0.35	0.39	**0.25**	0.38	0.32
seasonsforest/Hard/P006	0.29	0.13	0.19	**0.12**	0.24
seasonsforest_winter/Easy/P009	0.26	**0.08**	0.50	0.49	0.41
seasonsforest_winter/Hard/P018	3.49	0.84	1.57	**0.42**	1.00
soulcity/Easy/P012	0.32	0.15	**0.06**	0.13	0.15
soulcity/Hard/P009	0.29	0.22	0.17	**0.15**	0.43
westerndesert/Easy/P013	**0.23**	0.33	0.31	0.50	0.52
westerndesert/Hard/P007	0.38	0.45	0.25	0.27	**0.16**
Avg	0.95	0.41	0.91	**0.34**	0.35

Table 2. ATE RMSE on the TartanAir validation set. **Bold**: Best, Red: Worst.

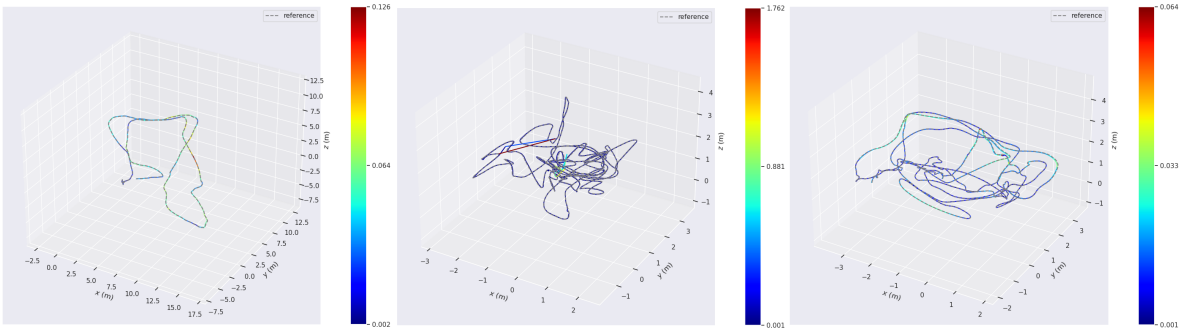
	MH000	MH001	MH002	MH003	MH004	MH005	MH006	MH007	Avg
ORB-SLAM [31]	1.30	0.04	2.37	2.45	X	X	21.47	2.73	-
DeepV2D [46]	6.15	2.12	4.54	3.89	2.71	11.55	5.53	3.76	5.03
TartanVO [54]	4.88	0.26	2.00	0.94	1.07	3.19	1.00	2.04	1.92
DROID-SLAM [48]	0.08	0.05	0.04	0.02	0.01	1.31	0.30	0.07	0.24
DROID	0.06	0.17	0.05	0.05	1.93	1.37	0.38	0.16	0.52
Stride	0.14	0.05	0.06	**0.02**	1.78	0.93	0.37	0.26	0.45
SPP	0.41	0.11	**0.02**	0.05	**0.19**	1.57	**0.29**	1.30	0.49
ASPP	**0.12**	**0.04**	0.03	**0.02**	2.22	2.64	0.37	**0.15**	0.70
Self-Att	0.35	0.05	0.06	0.04	1.29	**0.56**	0.48	0.21	**0.38**

Table 3. ATE RMSE on the TartanAir test set. **Bold**: Best, Red: Worst.

performance of DROID-SLAM. We re-train the original DROID-SLAM model and train a set of models that have the proposed changes integrated. We perform experiments on five with the configurations as shown in Table 1. We also carefully designed the discussed modifications with minimal change to the number of layers and number of parameters to make for a fair comparison. Note that due to memory limitations of the hardware, all the proposed modifications are incorporated within a strided implementation. In our experiments, we observe the memory consumption of the original DROID-SLAM during training exceeds ∼23GB, whereas the strided implementation only consumes roughly

		MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg
Deep	DeepFactors [9]	1.587	1.479	3.139	5.331	4.002	1.520	0.679	0.900	0.876	1.905	1.021	2.040
	DeepV2D [46]	0.739	1.144	0.752	1.492	1.567	0.981	0.801	1.570	0.290	2.202	2.743	1.298
	TartanVO [54]	0.639	0.325	0.550	1.153	1.021	0.447	0.389	0.622	0.433	0.749	1.152	0.680
	D3VO & DSO [57]	-	-	0.08	-	0.09	-	-	0.11	-	0.05	0.19	-
Classical	ORB-SLAM [31]	0.071	0.067	0.071	0.082	0.060	0.015	0.020	X	0.021	0.018	X	-
	DSO [15]	0.046	0.046	0.172	3.810	0.110	0.089	0.107	0.903	0.044	0.132	1.152	0.601
	SVO [17]	0.100	0.120	0.410	0.430	0.300	0.070	0.210	X	0.110	0.110	1.080	-
	DSM [62]	0.039	0.036	0.055	0.057	0.067	0.095	0.059	0.076	0.056	0.057	0.784	0.126
	ORB-SLAM3 [5]	0.016	0.027	0.028	0.138	0.072	0.033	0.015	0.033	0.023	0.029	X	-
	DROID-SLAM [48]	0.013	0.014	0.022	0.043	0.043	0.037	0.012	0.020	0.017	0.013	0.014	0.022
	DROID	0.027	0.014	0.023	0.048	4.189	0.036	0.018	0.085	0.016	0.011	0.032	0.409
	Strided	0.016	0.013	**0.022**	0.051	0.044	**0.034**	**0.011**	0.070	0.016	**0.009**	0.015	**0.027**
	SPP	0.014	0.013	0.024	**0.045**	0.046	0.036	0.014	0.086	**0.014**	0.037	0.042	0.034
	ASPP	0.019	0.013	0.024	0.047	**0.041**	0.035	0.012	**0.036**	**0.014**	0.089	**0.014**	0.031
	Self-Att	**0.013**	0.013	0.023	0.047	0.046	0.035	0.012	0.066	0.015	0.019	**0.014**	0.028

Table 4. ATE RMSE on the EuRoC datasets. **Bold**: Best, Red: Worst.



(a) (b) (c)

Figure 2. Estimated trajectory of self-attention based model and ATE against the ground truth reference in EuRoC MH_05_difficult, VH_01_difficult, and VH_02_difficult image sets.

∼19GB of memory per batch. Additionally, we perform experiments with only a single attention head for the self-attention Conv-GRU as the memory consumption restricts us from implementing multi-head attention.

We conducted training based on the publicly available official DROID-SLAM [48] codes. Training is done with 2 RTX-3090 GPUs with a batch size of 2, each with 7 adjacent frames. The images are resized to 384×512 resolution. 15 update iterations is performed during training. We follow the DROID-SLAM system at test time using our trained model weights.

Training is performed on the monocular images from the synthetic TartanAir dataset [55], which is split into training and validation split. The models are trained for $120k$ iterations. Each model takes approximately 2.5 days to train. Using the trained models, we conduct evaluations on the TartanAir validation and official test sets, EuRoC [4], and TUM-RGBD [42] datasets. The RMSE of the Absolute Trajectory Error (ATE) [42] is used to evaluate the accuracy of the computed trajectory. Evaluation is performed on the full camera trajectory.

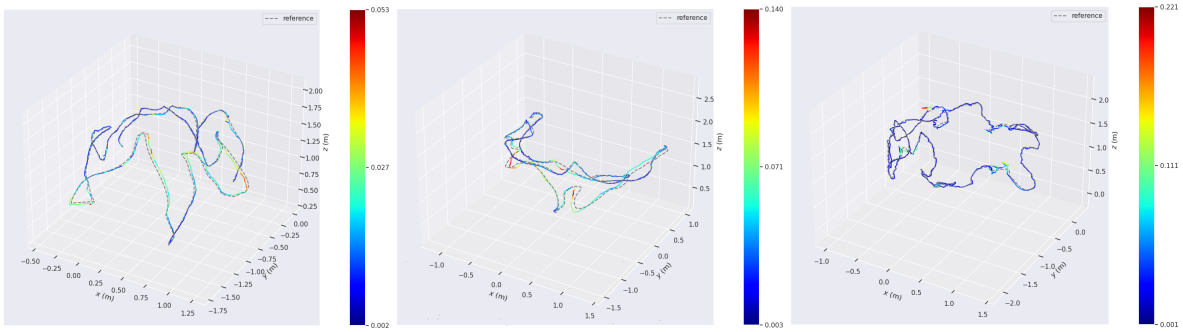
4.2. TartanAir [55]

TartanAir is a synthetic SLAM dataset collected using the AirSim [41] interface, containing challenging scenes with dynamic objects as well as light and weather variations.

For each model, we conduct evaluations on the validation set after epoch $50k, 100k$, and $120k$ training iterations. We select the best-performing model and present the ATE in Table 2. The TartanAir validation set covers a wide range of scenarios and difficulties. We mark the best performing values in each set with **bold** and the worst-performing ones with red. Interestingly, through these tests, we found the strided implementation of the DROID-SLAM to produce a more consistent performance across the different scenarios within the validation set when compared to the original implementation. The ATE of the original DROID fluctuates and occasionally shows a significant error. In addition, the inclusion of (A)SPP and self-attention into the model also improves the robustness in many scenarios, displaying the best overall average ATE.

	360	desk	desk2	floor	plant	room	rpy	teddy	xyz	Avg
ORB-SLAM2 [32]	X	0.071	X	0.023	X	X	X	X	0.010	-
ORB-SLAM3 [5]	X	0.017	0.210	X	0.034	X	X	X	0.009	-
DeepV2D [46]	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
DeepFactors [9]	0.159	0.170	0.253	0.169	0.305	0.364	0.043	0.601	0.035	0.233
DROID-SLAM [48]	0.111	0.018	0.042	0.021	0.016	0.049	0.026	0.048	0.012	0.038
DROID	**0.081**	0.019	**0.029**	0.021	0.019	0.052	0.022	0.031	0.010	**0.032**
Strided	0.175	**0.018**	0.037	0.024	0.024	0.058	0.023	0.045	0.010	0.046
SPP	0.177	0.019	0.030	0.057	0.135	**0.048**	**0.021**	**0.029**	**0.009**	0.058
ASPP	0.153	**0.018**	0.034	0.021	0.020	0.617	0.024	0.042	0.010	0.104
Self-Att	0.156	0.019	0.031	0.021	**0.018**	0.053	0.023	0.038	0.010	0.041

Table 5. ATE RMSE of monocular methods on the TUM-RGBD datasets. **Bold**: Best, Red: Worst.



(a) (b) (c)

Figure 3. Estimated trajectory of self-attention based model and ATE against the ground truth reference in TUM-RGBD plant, room, and teddy image sets.

We present the ATE of the models in the 'hard' subset of the official test split of the TartanAir dataset in Table 3. Additionally, we also show the ATE of recent SoTA models as well as the official DROID-SLAM (trained for 250k iterations) for comparison. Here, we observe the self-attention based DROID-SLAM to yield the lowest overall error. Although the self-attention based model doesn't show the best performance in many sequences, we found some of the other models to struggle in 'MH004' and 'MH005' sequence and displays huge error. This indicates that global self-attention improves the robustness of the model under challenging scenarios.

4.3. EuRoC [4]

We test the models in other datasets to evaluate the cross-domain capability of the models. EuRoC is a real-world dataset collected by micro-aerial-vehicle (MAV) capturing industrial environments. Table 4 presents the ATE values on the EuRoC sub-sequences. Again, we also present the other SoTA models and the original DROID-SLAM results as a comparison. In this dataset, we didn't see improvement in a having larger receptive field over the base strided implementation. Fig. 2 shows the plotted trajectory of the self-attention based model in the difficult EuRoC scenarios.

4.4. TUM-RGBD [42]

TUM-RGBD is a real-world dataset collected by hand-held cameras that capture indoor environments. Although the data contains RGB-D images, we conduct evaluations in monocular settings. Table 5 presents the ATE values on the TUM-RGBD subsequences. In this data, we found the original DROID-SLAM to perform best compared to the modified models. This could be due to the heavier motion blur and rolling shutter artifacts that are present in this data. In such a scenario, the original DROID model can utilize more precise image data compared to the strided implementation. However, we again observe that the global self-attention based model improves upon the standard strided implementation. We present the plotted trajectory of the self-attention based model in selected scenarios of the TUM-RGBD dataset.

4.5. Discussion

In most of the datasets, we observe self-attention based Conv-GRU to provide the most robust performance. However, we also note that our training is performed with a small batch size of 2 due to resources limitation, whereas the original DROID-SLAM conducted their training with a batch size of 4. The smaller batch size can potentially lead to noisier optimization during training, which may result in inconsistent

models. Additionally, the model can be trained for longer to account for this, just as the original DROID-SLAM was trained for 250k iterations.

Overall, we can observe consistent improvements in the proposed Conv-GRU variants over the standard Conv-GRU. A satisfactory level of accuracy was also obtained with the strided implementation along with the reduction in memory consumption, potentially enabling the model to be deployed in a smaller embedded platform in the future. We display visualization outputs in Fig. 4. Further experiments could be conducted to integrate a small-sized UNet inside Conv-GRU. This enables the neural network to obtain the benefit of both the UNet-based architecture of previous optical flow models, as well as the high-resolution iterative updates of the Conv-GRU.

While this work focused on the effect of the proposed modifications on the DROID-SLAM model, further analysis of its impact on the RAFT performance in the optical flow task should also be conducted.

5. Conclusions

In this work, we propose spatial pyramid pooling and global self-attention to be integrated into the Conv-GRU update block of DROID-SLAM. These modifications allow the model to aggregate context cues from a larger effective receptive field, allowing the model to infer optical flow updates even in challenging image regions. We showed empirically in our experiments the improvement in accuracy of the DROID-SLAM model across multiple datasets. We also noted potential directions to improve the Conv-GRU module further and hope that this work will be valuable to future works.

References

[1] Hatem Alismail, Michael Kaess, Brett Browning, and Simon Lucey. Direct visual odometry in low light using binary descriptors. *IEEE Robotics and Automation Letters*, 2(2):444–451, 2016.

[2] Antyanta Bangunharcana, Jae Won Cho, Seokju Lee, In So Kweon, Kyung-Soo Kim, and Soohyun Kim. Correlate-and-excite: Real-time stereo matching via guided cost volume excitation. *arXiv preprint arXiv:2108.05773*, 2021.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[4] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[5] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–

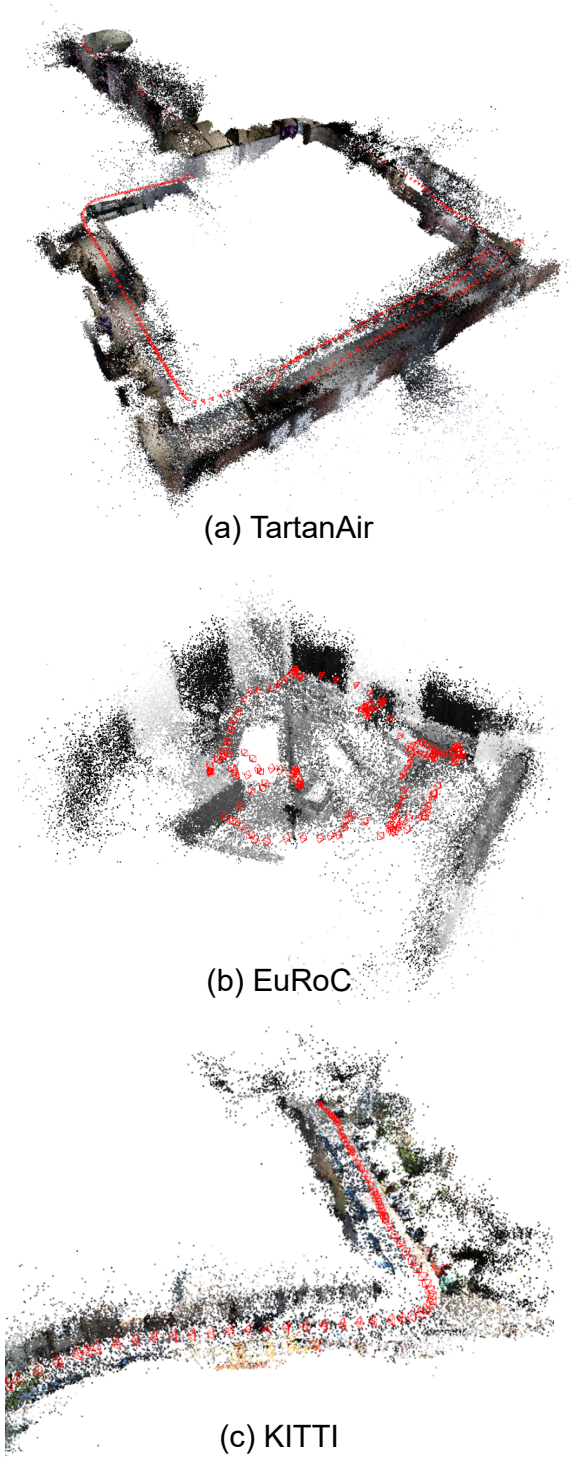


Figure 4. Visualization of the SLAM outputs.

1890, 2021.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[8] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[9] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5(2):721–728, 2020.

[10] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[14] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[15] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[16] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[17] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Sungchul Hong, Antyanta Bangunharcana, Jae-Min Park, Minseong Choi, and Hyu-Soung Shin. Visual slam-based robotic mapping method for planetary construction. *Sensors*, 21(22):7715, 2021.

[22] Sungchul Hong, Pranjay Shyam, Antyanta Bangunharcana, and Hyuseoung Shin. Robotic mapping approach under illumination-variant environments at planetary construction sites. *Remote Sensing*, 14(4):1027, 2022.

[23] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[24] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

[25] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007.

[26] Mathieu Labbé and François Michaud. Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2):416–446, 2019.

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[28] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[29] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[30] Anastasios I Mourikis, Stergios I Roumeliotis, et al. A multi-state constraint kalman filter for vision-aided inertial navigation. In *ICRA*, volume 2, page 6, 2007.

[31] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[32] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[33] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011.

[34] Taihú Pire, Thomas Fischer, Gastón Castro, Pablo De Cristóforis, Javier Civera, and Julio Jacobo Berlles. S-ptam: Stereo parallel tracking and mapping. *Robotics and Autonomous Systems*, 93:27–42, 2017.

[35] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[38] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[39] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[40] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3247–3257, 2021.

[41] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, pages 621–635. Springer, 2018.

[42] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[44] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[45] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018.

[46] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018.

[47] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020.

[48] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems*, 34, 2021.

[49] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[51] Lukas Von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. Gn-net: The gauss-newton loss for multi-weather relocalization. *IEEE Robotics and Automation Letters*, 5(2):890–897, 2020.

[52] Lukas Von Stumberg, Patrick Wenzel, Nan Yang, and Daniel Cremers. Lm-reloc: Levenberg-marquardt based direct visual relocalization. In *2020 International Conference on 3D Vision (3DV)*, pages 968–977. IEEE, 2020.

[53] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[54] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. *arXiv preprint arXiv:2011.00359*, 2020.

[55] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020.

[56] Yutong Xie, Jianpeng Zhang, Chunhua Shen, and Yong Xia. Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 171–180. Springer, 2021.

[57] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[58] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European conference on computer vision*, pages 467–483. Springer, 2016.

[59] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 2666–2674, 2018.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[61] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018.

[62] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4):1363–1370, 2020.