ATLAS-MVSNet: Attention Layers for Feature Extraction and Cost Volume Regularization in Multi-View Stereo

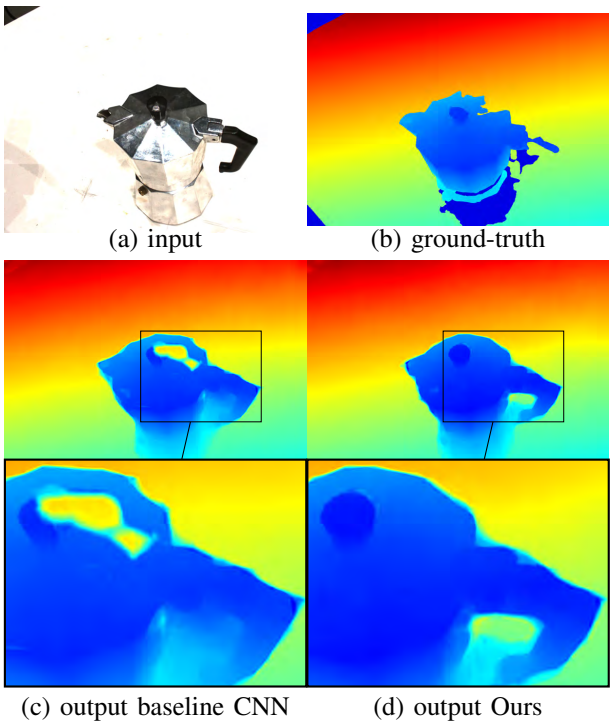
Rafael Weilharter* and Friedrich Fraundorfer*†,
*Institute of Computer Graphics and Vision, Graz University of Technology
†Remote Sensing Technology Institute, German Aerospace Center, Germany

***Abstract*—We present ATLAS-MVSNet, an end-to-end deep learning architecture relying on local attention layers for depth map inference from multi-view images. Distinct from existing works, we introduce a novel module design for neural networks, which we termed hybrid attention block, that utilizes the latest insights into attention in vision models. We are able to reap the benefits of attention in both, the carefully designed multi-stage feature extraction network and the cost volume regularization network. Our new approach displays significant improvement over its counterpart based purely on convolutions. While many state-of-the-art methods need multiple high-end GPUs in the training phase, we are able to train our network on a single consumer grade GPU. ATLAS-MVSNet exhibits excellent performance, especially in terms of accuracy, on the DTU dataset. Furthermore, ATLAS-MVSNet ranks amongst the top published methods on the online Tanks and Temples benchmark.**

I. Introduction

Multi-View Stereo (MVS) aims to reconstruct a dense 3D model of an observed scene from a series of images with their respective calibrated camera parameters alone. While traditional methods [7], [8], [25], using hand-crafted similarity metrics, have long been dominate in the field, recent deep-learning approaches are achieving superior accuracy and completeness on many MVS benchmarks [1], [17], [38]. This can be attributed to the introduction of Convolutional Neural Networks (CNNs) which are able to capture local features very well. Propelled by the computational power of modern GPUs, many of these deep-learning methods [10], [36], [37] follow a similar concept: Firstly, dense features are computed via feature extraction network. Secondly, these features are aggregated to form a cost volume utilizing the plane sweep algorithm [5]. Finally, the cost volume is regularized to estimate the final output in form of a depth map.

While these methods are able to achieve impressive results, accurate matching problems still remain in low-textured, repetitive, specular and reflective regions. A possible reason for this is that context-aware features have not been leveraged well enough yet. However, with the advent of the attention [28] mechanism, which was initially proposed for natural language processing, the computer vision community has been offered a new tool. Attention captures content-based, spatial-aware information and has already enjoyed rich success in the tasks of object detection [30] and image classification [12]. Nevertheless, these works rely on global attention layers, which attend to all spatial locations of an input and are therefore limited to a small input. To combat this issue, Ramachandran et al. [21] introduce the local self-attention layer, which extends the use of attention to larger inputs, boosting the convolutional baseline for the aforementioned tasks.



(a) input (b) ground-truth (c) output baseline CNN (d) output Ours

Fig. 1: Qualitative comparison of *scan77* on the DTU dataset. The baseline network (c), relying purely on convolutional layers, struggles with low-textured regions. Our network (d), enhancing (c) with hybrid attention blocks, is able to accurately reconstruct the problematic regions.

In this paper, we propose ATLAS-MVSNet, a straightforward network which utilizes local Attention Layers (ATLAS) for both, the feature extraction and the 3D regularization to significantly boost the performance over vanilla CNN solutions (see Figure 1). Our main contributions can be summarized as follows:

- We introduce a multi-stage feature extraction network

with hybrid attention blocks (HABs) to extract dense features and capture important information for the later matching and depth inference tasks.
- We extend the local 2D attention layers proposed by [26] to 3D in order to be able to adopt our HABs for the 3D regularization network.
- We produce clean depth maps prior to applying any filtering technique with an end-to-end neural network that is fully trainable on a single consumer grade GPU with only 11GB of memory.
- We perform extensive evaluations to show that our ATLAS-MVSNet ranks amongst the top methods on the DTU and the more challenging Tanks and Temples (TaT) benchmarks.

II. Related Work

The computer vision community has been researching MVS methods for decades. Starting with traditional methods such as Gipuma [8] or COLMAP [25] handcrafted features were extracted and matched in order to estimate a dense 3D representation of the observed environment. Although these approaches perform well in a multitude of scenarios they require a significant amount of processing time and are not suitable for non-Lambertian, low-textured or texture-less regions, usually resulting in poor performance in terms of completeness.

In recent years, deep learning methods have accomplished significant improvements over their traditional counterparts in many vision tasks [6], [9], [11], [18], [22]. Inspired by their success, learning-based MVS methods picked up on this idea with promising performances. First approaches [13], [14] utilize a volumetric scene representation where the cost volume is built upon. However, since the memory requirements grow proportionally to scene size, only small scale reconstructions are possible. To lift this restriction, MVSNet [36] introduces a nowadays widely adapted pipeline that produces a depth map for every reference image: First off, dense features are computed for a fixed number of overlapping images. Multiple features are then mapped into one cost volume using a variance-based cost metric. The cost volume is then regularized and yields a depth estimate via regression. In this manner, memory consumption is only related to the size of the input image. Nevertheless, MVSNet is quickly brought to its limits when it comes to higher resolution images.

To address this issue, different approaches have been taken: R-MVSNet [37] adopts gated recurrent units (GRUs) to regularize the cost volume in a sequential manner, trading decreased memory requirements for increased runtime. Fast-MVSNet [40] uses a sparse-to-dense approach and first only estimates a sparse but high-resolution depth map. CasMVS-Net [10] is able to reduce the depth dimension of the cost volume by predicting the depth in a coarse-to-fine manner.

Although more recent approaches [4], [32], [34], who built upon these insights, achieve impressive results, there is still room for improvement regarding the reconstruction quality.

In the task of object detection and image classification, the attention mechanism has been successful in achieving gains by augmenting convolutional models with content-based interactions [3]. This motivated several works [19], [41] to capitalize on the new technique also in MVS. A first attempt to exploit the local attention layer proposed by [21] is performed by Yu et al. [39]. In contrast to their work, we utilize several of the latest insights into attention in vision models [21], [27] and design the hybrid attention block that we take advantage of throughout ATLAS-MVSNet.

III. Method

In this section, we introduce the detailed architecture of ATLAS-MVSNet with its novel components. We first extract features at multiple stages with decreasing resolution. Afterwards, we generate the depth at the coarsest resolution and utilize the cascading cost volume formulation [10] to predict the subsequent depth maps in a coarse-to-fine manner. An overview of our network design is shown in Figure 2.

A. Feature Extraction

For feature extraction we design a multi-stage network adopting a U-NET [23] architecture. At the beginning, we apply 4 convolutional layers, where the stride of layer 1 is set to 2. Afterwards, we pass the obtained feature map through our 2D HAB at stage 0.

2D Hybrid Attention Block Our HAB is constructed as a residual block that uses a hybrid combination of convolutional and local attention layers (see Figure 3). To reduce the memory requirement for the local attention layer [21], the input first passes through a convolutional layer with stride 2 followed by a group normalization [33] (GN) and ReLU layer. The realization of the local attention layer is depicted in Figure 4: Similar to a convolution, the input is a local region R of size $s \times s$ centered around the pixel of interest $\mathbf{x_{ij}}$. From R the pixel output $\mathbf{y_{ij}}$ can be calculated via *softmax* operation $\sigma(\cdot)$:

$$\mathbf{y_{ij}} = \sum_{a,b \in R} \sigma_{ab}(\mathbf{q_{ij}}^{\top} \mathbf{k_{ab}}) \mathbf{v_{ab}}, \tag{1}$$

where queries $\mathbf{q_{ij}} = W_q \mathbf{x_{ij}}$, keys $\mathbf{k_{ab}} = W_k \mathbf{x_{ab}}$ and values $\mathbf{v_{ab}} = W_v \mathbf{x_{ab}}$ are learnable linear transformations with their respective weight matrices W_q, W_k and W_v. A drawback of this formulation is that no positional information is encoded, which leads to permutation equivariance, limiting the performance for vision tasks. Hence, the relative positional embedding [26] is introduced by adding learnable parameters to the keys. The relative distance is factorized across dimensions using half of the dimension of the output channel for encoding the row direction and the other half for encoding the column direction. In practice this can be accomplished by arranging the 2D encodings as a vector $\mathbf{r_{ab}}$ resulting in:

$$\mathbf{y_{ij}} = \sum_{a,b \in R} \sigma_{ab}(\mathbf{q_{ij}}^{\top} (\mathbf{k_{ab}} + \mathbf{r_{ab}})) \mathbf{v_{ab}}. \tag{2}$$

In this manner the attention layer can be integrated into the network just like a convolutional layer. Distinct from the latter, the aggregation is done in a more entangled way with convex combinations of value vectors and the *softmax* operation.

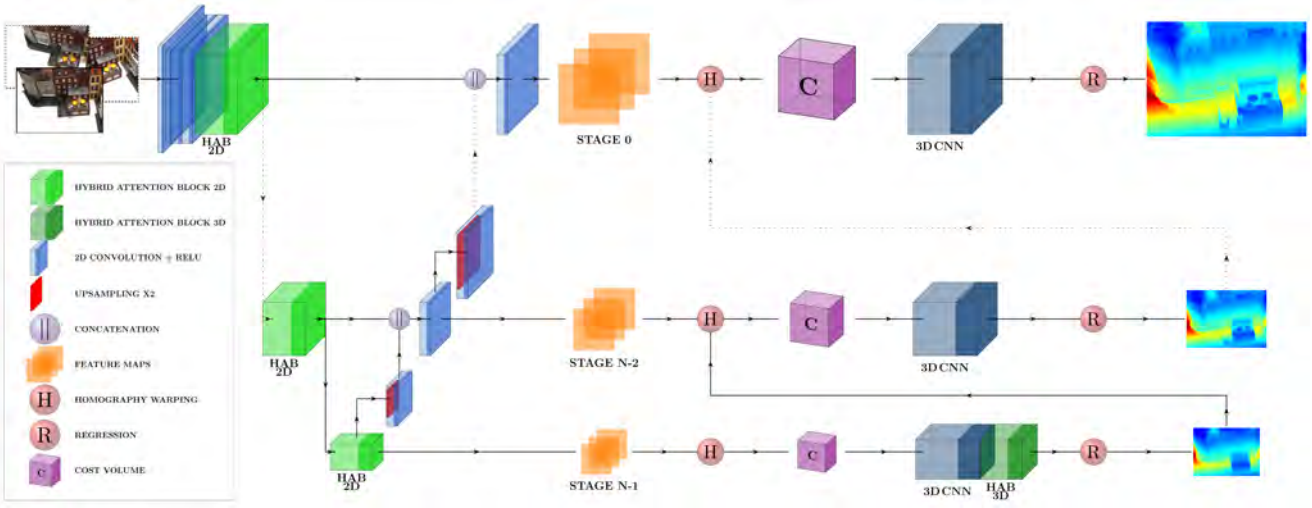


Fig. 2: Overview of the proposed ATLAS-MVSNet architecture: At first, a multi-stage feature extraction network utilizing 2D HABs is applied to a given set of images. Features at different scales are aggregated into a cost volume through homography warping. The cost volume at the coarsest scale (stage $n-1$) is regularized by a 3D CNN followed by a 3D HAB and yields a depth estimate via regression. The estimate is used to initialize the cost volumes of the subsequent stage. This process is repeated for n stages to obtain the final depth map.

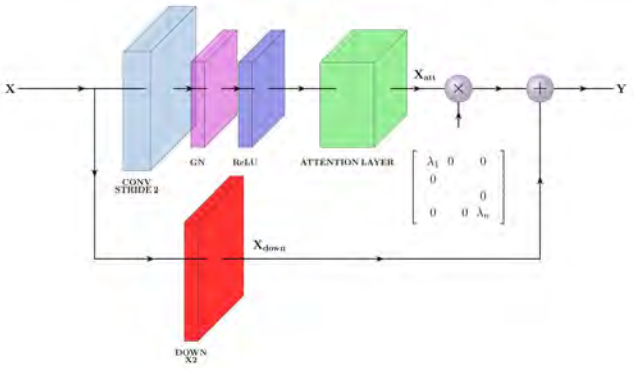


Fig. 3: Hybrid attention block overview: We adopt a residual block architecture where the input first passes through a convolutional layer with stride 2 followed by a GN and ReLU layer. We then apply a local attention layer with LayerScale.

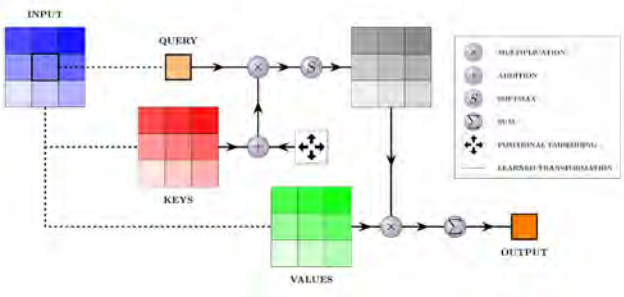


Fig. 4: Local attention layer details for a spatial extend of $s = 3$: In contrast to a convolutional layer with 1 transformation, we learn 3 distinct transformations for query, keys and values.

As a normalization strategy we apply the LayerScale as proposed by [27]. Formally, this is done by multiplying a diagonal matrix to the output $\mathbf{X_{att}}$ after the attention layer:

$$\mathbf{Y} = diag(\lambda_1, ..., \lambda_n) \times \mathbf{X_{att}} + \mathbf{X_{down}}, \tag{3}$$

where Y is the final output of the HAB and $\mathbf{X_{down}}$ is the down sampled input. The parameters λ_1 to λ_n are learnable weights.

The last output at the lowest scale yields the coarsest feature map. For the following stages we upscale the previous HAB output by a factor of 2 and concatenate the features with the current stage HAB output. An additional convolutional layer is applied after concatenation.

B. Cost Volume Construction

Following previous architectures [10], [35], [37], we construct the cost volume by warping the obtained feature maps into fronto-parallel planes in the reference camera frustum. The warping is defined by the homography:

$$H_i(d) = K_i \cdot R_i \cdot \left(I - \frac{(\mathbf{t_0} - \mathbf{t_i}) \cdot \mathbf{n_0}^\top}{d} \right) \cdot R_0^\top \cdot K_0^\top, \tag{4}$$

where $H_i(d)$ is the homography between the i^{th} feature map and the reference feature map at depth d. The parameters $K_i, R_i, \mathbf{t_i}$ refer to the camera intrinsics and extrinsics with index 0 indicating the reference view, $\mathbf{n_0}$ is the principle axis of the reference camera and I is the identity matrix. For the aggregation of multiple feature volumes to one cost volume, the variance-based cost metric is employed to accommodate an arbitrary number of input feature volumes.

We need to perform these operations in each stage of our network to obtain a cost volume of the corresponding scale.

However, as the GPU memory requirements would increase cubically for every stage we follow [10] and only cover the full depth range in the coarsest stage which has the smallest cost volume resolution. The ensuing cost volumes can then be built upon a narrower depth range based on the previous prediction using the cascading cost volume formulation. This allows us to use a fine plane interval while keeping the memory consumption in check.

C. Cost Volume Regularization

As stated previously, we predict the depth maps in a coarse-to-fine pattern. To get a depth map from a cost volume, we pass it through a 3D regularization network and regress the depth via *soft argmin* operation [16]:

$$\textit{soft argmin} := \sum_{d=1}^{d_{max}} d \times \sigma(-c_d), \tag{5}$$

where d_{max} is the maximum depth value, c_d is the predicted cost and $\sigma(\cdot)$ is again the *softmax* operation.

Our 3D regularization networks consists of 5 blocks of two 3D convolutional layers with a residual connection followed by a 3D HAB.

3D Hybrid Attention Block The design principle is the same as depicted in Figure 3 but without the down sampling as we want to keep the cost volume at a constant scale. We can extend the local 2D attention layers in Figure 4 by expanding the weight matrices W_q, W_k and W_v into the third dimension. In order to extend the positional encoding to 3D, we need to add another vector of learnable parameters for the depth direction. This implies that we now factorize across 3 dimensions, each encoding embedded in $\frac{1}{3}$ of the output channel dimension. Again, as was the case with the 2D HAB, the intention of 3D HAB is to capture positionally relevant context information.

We employ our 3D HAB only at the coarsest stage for the following 2 reasons: 1) Unfortunately, the HAB comes at the cost of increased GPU memory consumption as we have to learn a distinct transformation for each, query, key and value. This leads to an exponential increase of GPU memory requirements. 2) It is most critical to get a correct depth estimate in the coarsest stage, which covers the full depth range, as this prediction will get propagated through the other stages.

In this manner, we are able to produce a high quality depth map as direct output of our network (see Figure 5).

D. Loss Function

ATLAS-MVSNet with n stages produces $n-1$ intermediate outputs and 1 final depth prediction. We apply a multi-scale loss over all outputs by calculating the mean absolute difference between ground truth and predicted depth map in every stage:

$$l = \sum_{k=0}^{n-1} \lambda_k \cdot \|D_{k,gt} - D_{k,pred}\|_1, \tag{6}$$

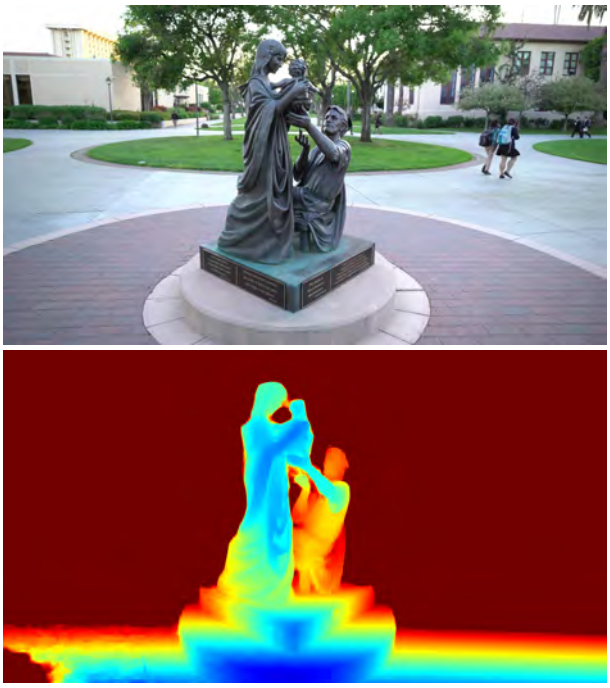


Fig. 5: Example depth map output of our network. We are able to produce a high quality depth map even before applying any filtering technique.

where λ_k is the loss weight which we reduce by a factor of $\frac{1}{2}$ in every stage in order to account for the different scale levels.

IV. Implementation

We set the number of stages in our final network to 5. The reasoning behind this decision is twofold: 1) Memory efficiency: As only the coarsest stage needs to cover the full depth range of the image, we can set a lower number of depth hypothesis in subsequent stages resulting in a smaller cost volume size. 2) Result accuracy: Empirically we found that 5 stages yield the better results over 4 stages (see Section V-C).

From the coarsest stage 4 to the finest stage 0 we set the number of depth hypothesis to 32, 8, 8, 8 and 4 respectively. Throughout the network we normalize each layer with GN.

A. Training

We train ATLAS-MVSNet on the DTU dataset [1] and on the BlendedMVS dataset [38]. In case of the DTU dataset, data is captured under the same lab conditions for every scene and ground truth is only available in form of laser point clouds. Therefore we utilize the depth maps provided by MVSNet [36] which are generated from the point clouds via screened Poisson surface reconstruction [15]. In order to introduce more variety into our training data, we also include the BlendedMVS dataset. The dataset contains various scenes including cities, architectures, sculptures and small objects. It has been shown in [38] that this additional data can help to improve the network performance and generalization on more complex scenes.

We use Adam optimizer [2] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and initialize the learning rate with 0.001. The learning rate is reduced by a factor of 0.5 during training at epochs 10, 12 and 14. We fix the number of input images to 3 with an image resolution of 1600×1152 and train for a total of 18 epochs. Our network is trainable end-to-end on a single consumer grade GPU with 11GB memory (e.g. Nvidia GeForce GTX 1080 Ti, Nvidia GeForce RTX 2080 Ti).

B. Point Cloud Fusion

As our network processes the final cost volume at $\frac{1}{4}$ of the input resolution, we upscale the depth map by a factor of 2 before projecting every pixel into 3D space to obtain a denser point cloud. However, some pixels might contain inaccurate or wrong depth predictions due to occlusions or uncertainties. Since the same 3D point can be observed from multiple views, we can filter these inaccuracies by checking the geometric consistency. This can be done by projecting a reference pixel p_{ref} through its depth d_{ref} to pixel p_i in a different view and then back-project p_i through d_i to obtain p_{proj}. The depth is now 2 view consistent if it satisfies:

$$\|p_{ref} - p_{proj}\| < \tau, \tag{7}$$

where τ is a threshold value for the back-projection error. To adapt for the typical case of multiple views, we use the dynamic consistency checking (DCC) strategy [34] and adjust τ dynamically. In its essence, this strategy deems an estimated depth value as accurate and reliable if it has a very low back-projection error in a few views or a certain consensus in the majority of views.

V. Experiments

We evaluate ATLAS-MVSNet on the well known DTU [1] and Tanks and Temples [17] benchmarks. We use the full resolution images and set the number of input images to 5.

A. Evaluation on the DTU Dataset

We evaluate our final model trained on the DTU dataset in Table I, where our method compares favorably with other state-of-the-art methods. *Accuracy* reflects the absolute average distance for every generated 3D point from the ground truth point cloud, whereas *completeness* expresses the integrity of the reconstruction i.e. the absolute average distance for every ground truth 3D point from the generated point cloud. Note, that there is a trade-off between these measurements, which is dependent on the fusion parameter τ. We find that our *overall* score increases when we opt for high *accuracy*. To the best of our knowledge, we are the first method to beat the traditional approach of Gipuma [8] in *accuracy*.

B. Evaluation on Tanks and Temples

The TaT intermediate benchmark consists of outdoor scenes captured in a more complex and real setting with varying depth ranges. We follow recent practices [20], [31] and train our model on the BlendedMVS dataset in order to achieve better generalization. The quantitative evaluation of our method can be found in Table II, where our approach ranks amongst the top published methods, while keeping runtime and GPU memory requirements low. Furthermore we compare our method qualitatively to EPP-MVSNet [20], ranked highest in Table II, in Figure 6. We can see that our method produces a low number of 3D point outliers and generates visually appealing results.

TABLE I: Quantitative results on the DTU test dataset. All scores are in mm and represent the mean average distance (lower is better). Best results are shown in bold and the runner-ups are underlined.

Method	Acc.	Comp.	Overall
Gipuma [8]	<u>0.283</u>	0.873	0.578
COLMAP [24], [25]	0.400	0.664	0.532
MVSNet [36]	0.396	0.527	0.462
R-MVSNet [37]	0.383	0.452	0.417
CasMVSNet [10]	0.346	0.351	0.348
PatchmatchNet [29]	0.427	**0.277**	0.352
D^2HC-RMVSNET [34]	0.395	0.378	0.386
EPP-MVSNet [20]	0.413	<u>0.296</u>	0.355
AA-RMVSNet [31]	0.376	0.339	0.357
AACVP-MVSNet [39]	0.357	0.326	0.341
AttMVS [19]	0.383	0.329	0.356
LANet [41]	0.320	0.349	<u>0.335</u>
Ours	**0.278**	0.377	**0.327**

C. Ablation Study

We perform a number of ablation studies on the DTU dataset to find the ideal hyperparameters for our network and to validate its novel components in Table III. We train for 16 epochs on the training set and use simple 3-view consistent filtering, as proposed by [36], with $\tau = 0.25$. In order to be able to cover the whole depth range of the scene within our memory budget, we need at least 4 stages. However, in our experiments we found that setting the number of stages to $n = 5$ tends to yield better results.

Compared to the baseline without any attention, we can see an improvement of the overall score from 0.342 to 0.337 when applying the 2D HAB and to 0.336 when applying 2D and 3D HAB. Our findings conclude that adding the 3D HAB in addition to the 2D HAB will improve the result, while increasing the number of attention heads will worsen it. Usually multiple attention heads are used to learn multiple distinct representations of the input by partitioning pixel features into groups [21]. Commonly, this leads to an increase in performance of the network when using a large channel size. However, as our network only uses a small channel size of 36, this is not the case. Finally, we see another slight improvement in the overall score when applying DCC with a rather strict threshold, thus trading completeness for accuracy.

VI. Conclusion

We have presented ATLAS-MVSNet, a deep learning architecture that utilizes local attention to achieve superior 3D reconstruction accuracy. In particular, we have proposed the hybrid attention block design for 2D, based on the self-attention layer and extended its application to 3D. In experiments we have shown that utilizing HABs instead of a vanilla residual

TABLE II: Results on the Tanks and Temples intermediate dataset of state-of-the-art MVS and our method. Precision and recall is combined as *f-score* (higher is better). Best results are shown in bold and the runner-ups are underlined.

Method	Mean	Family	Francis	Horse	LH	M60	Panther	PG	Train	Time(ms)	Mem.(GB)
COLMAP [24], [25]	42.41	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04	-	-
MVSNet [36]	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69	-	15.3
R-MVSNet [37]	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38	-	6.7
CasMVSNet [10]	56.42	76.36	58.45	46.20	55.53	56.11	54.02	58.17	46.56	792.2	9.5
PatchmatchNet [29]	53.15	66.99	52.64	43.24	54.87	52.87	49.54	54.21	50.81	505.0	2.9
D^2HC-RMVSNET [34]	59.20	74.69	56.04	49.42	60.08	59.81	59.61	60.04	53.92	-	-
EPP-MVSNet [20]	**61.68**	77.86	60.54	**52.96**	62.33	61.69	**60.34**	62.44	55.30	555.2	8.2
AA-RMVSNet [31]	61.51	77.77	59.53	51.53	64.02	**64.05**	59.47	60.85	54.90	-	-
AACVP-MVSNet [39]	58.39	**78.71**	57.85	50.34	52.76	59.73	54.81	57.98	54.94	-	-
AttMVS [19]	60.05	73.90	**62.58**	44.08	**64.88**	56.08	59.39	63.42	**56.06**	-	-
LANet [41]	55.70	76.24	54.32	49.85	54.03	56.08	50.82	53.71	50.57	-	-
Ours	60.71	77.62	61.94	49.55	61.63	60.04	58.69	**63.58**	52.59	**394.5**	**2.1**

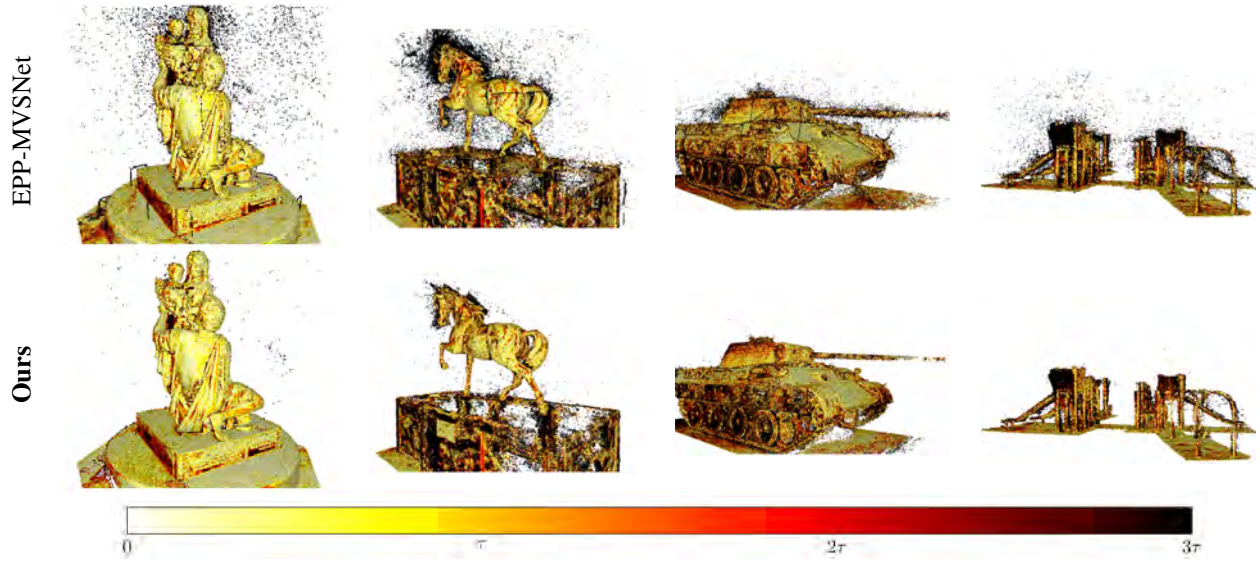


Fig. 6: Qualitative comparison between EPP-MVSNet and our method on the TaT benchmark. The color indicates points within a certain threshold distance τ to the ground truth. We can see that our method, despite being lower in *f-score*, produces far less outliers and therefore a visually more appealing result.

TABLE III: Ablations on the DTU dataset. H indicates the number of heads in the attention layers, 2D and 3D indicates the use of the respective HABs, DCC the use of dynamic consistency checking.

Planes	2D	3D	H	DCC	Acc.	Comp.	Overall
32/8/8/8/4	-	-	-	-	0.334	0.349	0.342
16/8/8/4	✓	✓	1	-	0.330	0.349	0.340
16/16/8/8/4	✓	-	2	-	0.338	0.355	0.346
16/16/8/8/4	✓	✓	2	-	0.335	0.348	0.342
32/8/8/8/4	✓	✓	2	-	0.333	0.348	0.341
32/8/8/8/4	✓	-	1	-	0.328	0.346	0.337
32/8/8/8/4	✓	✓	1	-	0.328	**0.344**	0.336
32/8/8/8/4	✓	✓	1	✓	**0.287**	0.381	**0.334**

blocks can boost the performance of a network qualitatively and quantitatively. ATLAS-MVSNet demonstrates great results on public benchmarks and outperforms most other state-of-the-art methods, while keeping GPU memory requirements low. This stems from the fact that our feature extraction network encodes features to $\frac{1}{4}$ of the input resolution and hence the regularization network only has to process cost volumes of reduced size. A limitation of our work is the memory requirement of the 3D HAB in the training phase, which makes its application infeasible in finer stages of the network. We might tackle this issue in future work in order to be able to use the HAB also for larger inputs.

REFERENCES

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3286–3295, 2019.

[4] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2524–2534, 2020.

[5] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. IEEE, 1996.

[6] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11592–11601, 2020.

[7] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[8] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25(361-369):1–2, 2016.

[9] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[10] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *arXiv preprint arXiv:1810.12348*, 2018.

[13] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017.

[14] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017.

[15] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[16] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[17] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] Keyang Luo, Tao Guan, Lili Ju, Yuesong Wang, Zhuo Chen, and Yawei Luo. Attention-aware multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1590–1599, 2020.

[20] Xinjun Ma, Yue Gong, Qirui Wang, Jingwei Huang, Lei Chen, and Fan Yu. Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5732–5740, 2021.

[21] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] 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.

[24] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[27] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[29] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14194–14203, 2021.

[30] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[31] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6187–6196, 2021.

[32] Rafael Weilharter and Friedrich Fraundorfer. Highres-mvsnet: A fast multi-view stereo network for dense 3d reconstruction from high-resolution images. *IEEE Access*, 9:11306–11315, 2021.

[33] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[34] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision*, pages 674–689. Springer, 2020.

[35] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020.

[36] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[37] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[38] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020.

[39] Anzhu Yu, Wenyue Guo, Bing Liu, Xin Chen, Xin Wang, Xuefeng Cao, and Bingchuan Jiang. Attention aware cost volume pyramid based multi-view stereo network for 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:448–460, 2021.

[40] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1949–1958, 2020.

[41] Xudong Zhang, Yutao Hu, Haochen Wang, Xianbin Cao, and Baochang Zhang. Long-range attention network for multi-view stereo. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3782–3791, 2021.