Supplementary Material for "Cascaded Transposed Long-range Convolutions for Monocular Depth Estimation"

Go Irie, Daiki Ikami, Takahito Kawanishi, and Kunio Kashino

NTT Corporation, Kanagawa 243-0124, Japan
goirie@ieee.org go.irie.nv@hco.ntt.co.jp

Abstract. This paper is the supplementary material for the paper entitled "Cascaded Transposed Long-range Convolutions for Monocular Depth Estimation." In this supplementary material, we report additional analysis of our Cascaded Transposed Long-range Convolutions (CTLCs), specifically the impact of kernel size and feature extraction backbone networks on performance and qualitative results.

1 Additional Analysis of Kernel Size

In Table 1 of the main paper, we report our analysis of the impact of kernel shapes on depth estimation performance. The analysis is limited to the kernels with (approximately) the same number of parameters as $(5 \times 5) \rightarrow (5 \times 5)$. We here report additional results for other patterns with different number of parameters such as $(1 \times 5) \rightarrow (5 \times 1)$ and $(1 \times 9) \rightarrow (9 \times 1)$.

The results are shown in Table 1. We observed a similar trend to the results presented in our main paper; the longer the kernel shapes, the better the performance. An additional observation from this analysis is that $(1\times5) \rightarrow (5\times1)$ (resp. $(1\times9) \rightarrow (9\times1)$) is slightly better than $(3\times3) \rightarrow (3\times3)$ (resp. $(5\times5) \rightarrow (5\times5)$) that has more number of parameters. These results highlight the validity of our idea of CTLC. Note that $(1 \times 25) \rightarrow (25 \times 1)$ is still the best. This is because (1×9) or (1×5) is a downgrade version of (1×25) and cannot capture local vertical-horizontal correlations as (3×9) can. We tried possible combinations of these kernels (e.g., $(1 \times 5) \rightarrow (5 \times 1)$ and $(1 \times 9) \rightarrow (9 \times 1)$) for making the final CTLC block, but none surpassed our original version shown in Fig. 5 of the main paper.

2 Additional Results on KITTI

In Table 4 of the main paper, we compare our method with the state-of-the-art methods on KITTI. In this comparison, we use DenseNet-161 as our feature extraction backbone network. In this supplementary material, we report the performance of our method when it is coupled with the ResNet-50 feature extraction backbone network.

Backbone	Kernel Shape	higher is better			lower is better			
		δ_1	δ_2	δ_3	AbsRel	SqRel	RMSE	$\text{RMSE}_{\log}$
DenseNet-161	$(5 \times 1) \rightarrow (1 \times 5)$	0.849	0.975	0.995	0.126	0.077	0.430	0.157
	$(1 \times 5) \rightarrow (5 \times 1)$	0.845	0.975	0.995	0.126	0.075	0.434	0.158
	$(3 \times 3) \rightarrow (3 \times 3)$	0.844	0.974	0.995	0.128	0.078	0.437	0.160
	$(9 \times 1) \rightarrow (1 \times 9)$	0.857	0.977	**0.996**	0.121	0.072	0.425	0.153
	$(1 \times 9) \rightarrow (9 \times 1)$	0.852	0.976	0.995	0.123	0.075	0.431	0.156
	$(5 \times 5) \rightarrow (5 \times 5)$	0.850	0.975	0.995	0.126	0.075	0.431	0.156
	$(25 \times 1) \rightarrow (1 \times 25)$	0.867	0.978	0.995	0.119	0.070	0.412	0.148
	$(1 \times 25) \rightarrow (25 \times 1)$	**0.871**	**0.979**	0.995	**0.116**	**0.068**	**0.409**	**0.147**
ResNet-50	$(5 \times 1) \rightarrow (1 \times 5)$	0.781	0.955	0.990	0.156	0.109	0.514	0.193
	$(1 \times 5) \rightarrow (5 \times 1)$	0.783	0.956	0.991	0.155	0.108	0.513	0.192
	$(3 \times 3) \rightarrow (3 \times 3)$	0.773	0.952	0.988	0.164	0.117	0.523	0.197
	$(9 \times 1) \rightarrow (1 \times 9)$	0.818	0.965	0.992	0.143	0.094	0.471	0.175
	$(1 \times 9) \rightarrow (9 \times 1)$	0.814	0.966	0.993	0.141	0.092	0.477	0.176
	$(5 \times 5) \rightarrow (5 \times 5)$	0.812	0.965	0.992	0.143	0.094	0.478	0.177
	$(25 \times 1) \rightarrow (1 \times 25)$	0.863	0.977	0.994	0.120	0.073	0.418	0.150
	$(1 \times 25) \rightarrow (25 \times 1)$	**0.867**	**0.980**	**0.995**	**0.117**	**0.069**	**0.413**	**0.148**

Table 1: **Additional results on impact of kernel sizes and shapes.** The scores are evaluated on NYU Depth V2 dataset. The best scores for each metric are shown in bold.

The results are shown in Table 2. Although there are slight performance differences between Ours w/ ResNet-50 and Ours w/ DenseNet-161, we found that Ours w/ ResNet50 can still outperform all the baselines listed in Table 4 of the main paper. These results demonstrate that our method has excellent performance on KITTI, regardless of the feature extraction backbone network.

3 Qualitative Results

Our final CTLC block has a parallel structure with four branches with long-range kernels of different shapes and combinations (see Fig. 5 of the main paper). In the main paper, we quantitatively analyze the benefits of this parallelization. In this supplementary material, we demonstrate its qualitative benefits.

Fig. 1 shows qualitative results of our final CTLC block. The depth maps estimated by our final CTLC block have the advantages of both when not parallelized. While maintaining the high estimation accuracy as $(1 \times 25) \rightarrow (25 \times 1)$, it can represent roundish corners as $(1 \times 9) \rightarrow (9 \times 1)$. Compared to $(1 \times 9) \rightarrow (9 \times 1)$ and $(1 \times 25) \rightarrow (25 \times 1)$, the final CTLC block can recover more precise depth boundaries. These observations emphasize the qualitative advantages of the final CTLC block.

Method	cap	higher is better			lower is better			
		δ_1	δ_2	δ_3	AbsRel	SqRel	RMSE	$\text{RMSE}_{\log}$
Ours w/ ResNet-50 (raw)	80m	0.938	0.989	**0.998**	0.071	0.317	3.040	0.112
Ours w/ ResNet-50 (GT)	80m	0.947	**0.992**	**0.998**	0.065	0.272	**2.929**	0.103
Ours w/ DenseNet-161 (raw)	80m	0.896	0.972	0.990	0.093	0.519	3.856	0.155
Ours w/ DenseNet-161 (GT)	80m	**0.951**	**0.992**	**0.998**	**0.064**	**0.271**	2.945	**0.101**
Ours w/ ResNet-50 (raw)	50m	0.947	0.992	0.998	0.072	0.232	2.224	0.107
Ours w/ ResNet-50 (GT)	50m	0.952	0.993	0.998	0.068	0.213	2.177	0.101
Ours w/ DenseNet-161 (raw)	50m	0.911	0.975	0.991	0.086	0.399	2.933	0.145
Ours w/ DenseNet-161 (GT)	50m	**0.956**	**0.994**	**0.999**	**0.065**	**0.199**	**2.141**	**0.098**

Table 2: **Performance with different feature extraction backbone networks on KITTI.** "cap" gives the maximum depth used for evaluation. "(raw)" and "(GT)" mean that the models are trained with raw depth maps and post-processed ground truth depth maps, respectively. The best scores for each metric are shown in bold.

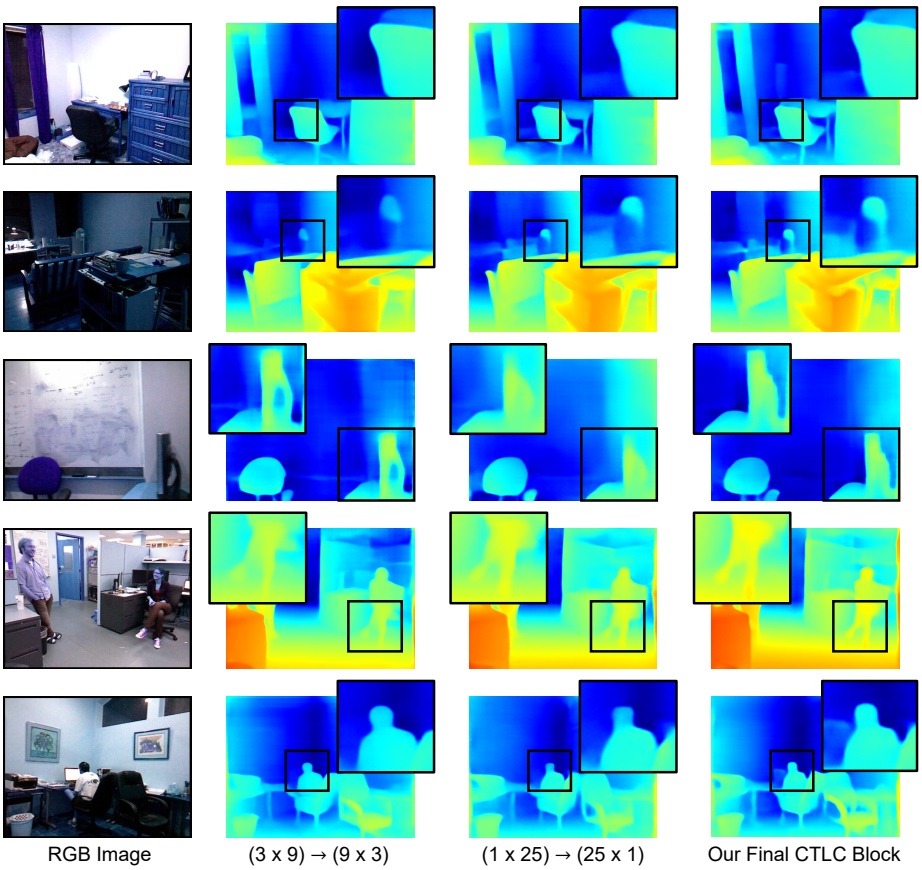

Fig. 1: **Qualitative results.** Examples of the estimated depth maps on NYU Depth V2 are shown. DenseNet-161 is used for the feature extraction backbone.