Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging

Supplementary Material

S. Mahdi H. Miangoleh*[1] Sebastian Dille*[1] Long Mai[2] Sylvain Paris[2] Yağız Aksoy[1]

[1] Simon Fraser University [2] Adobe Research

In this supplementary document, we present (i) tables with additional error metrics as well as visual demonstration of the D^{3}R error metric we utilize in Table 1 and Figures 8 and 9, (ii) extended figures and discussions on resolution and patch selection processes in Figures 10, 11 and 12, (iii) detailed explanation for the training process of our merging network in Figure 13, (iv) our figures of the main document replicated with SGR [7] as the base network (Figures 1, 2, 3, 4, 5, 6 and 7) and (v) additional results and a discussion on the running time of our method.

A. Extended discussion on metrics

We notice a discrepancy between the apparent visual improvement gained from depth refinement techniques and numerical results based on common metrics such as RMSE, AbsRel and δ_t (percentage of pixels with $\delta = \max(\frac{z_i}{z_i^*}, \frac{z_i^*}{z_i}) > t$). Figure 8 demonstrates on an example that qualitative differences do not correlate well with these metrics.

RMSE and also more recently introduced alternatives such as perceptual similarity metric by Zhang et al. [8] are influenced by the large-scale structure of the scene but do not measure performance around fine details very well. In the main document, we describe a metric, D^{3}R, that adapts the ordinal relation metric from [7] and mixes it with the point pair selection method introduced in [9] to capture performance around high-frequency details. In their method, Zoran et al. [9] propose to use superpixel segments' centers to select point pairs. Instead of using superpixels from the RGB image we directly use the ground truth depth to generate superpixels. We then consider the ratio between the ground truth depth values at each center and its 1-neighbour centers in the superpixel segmentation and pick center pairs such that this ratio is bigger than a determined threshold so that the picked pairs represent a relatively big change in depth. This way, the selected points will specifically reflect depth accuracy around object boundaries. We refer to this adjusted metric as D^{3}R. The aforementioned threshold is empirically set to 0.1 in our experiments. Figure 9 shows an example for the created superpixel segmentation, the selected point pairs for evaluation and resulting error points for the base estimation.

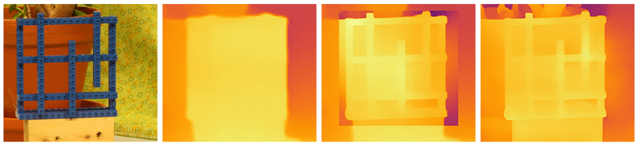

Figure 1: As extension to Figure 7, we show an example of the base estimate for a given region (second from left), a patch-estimate pasted onto the base estimate from SGR [7](second from right), and our result after merging (right).

We compare the results of our metric against RMSE in two examples in Figure 8. In both cases, D^{3}R shows a noticeable improvement for the results obtained by our method compared to the base estimation. This said, D^{3}R will only reflect accuracy around the objects boundaries and shall be used as complementary to conventional metrics.

In addition to the metrics we evaluated on in the paper, we extend and report our evaluations on $\delta_{1.25^2}$, $\delta_{1.25^3}$, AbsRel $(1/M \sum_{i=1}^{M} |z_i - z_i^*|/z_i^*)$, Log_{10} $(1/M \sum_{i=1}^{M} |log(z_i) - log(z_i^*)|)$ and SqRel $(1/M \sum_{i=1}^{M} (z_i - z_i^*)^2/z_i^*)$.

B. Extended discussion on whole-image estimation and patch selection

We want to ensure that for the whole-image estimation (Section 6.1 in the main document), individual depth cues have a distance of at most the size of the receptive field from each other. To achieve this, we calculate an edge map by applying a threshold on the RGB image gradients and rescale this edge map to different sizes, starting at the receptive field size itself and increasing it by half of the receptive field size for every step. Next, we apply a dilation kernel with the size of the receptive field on the resulting edge map. If the

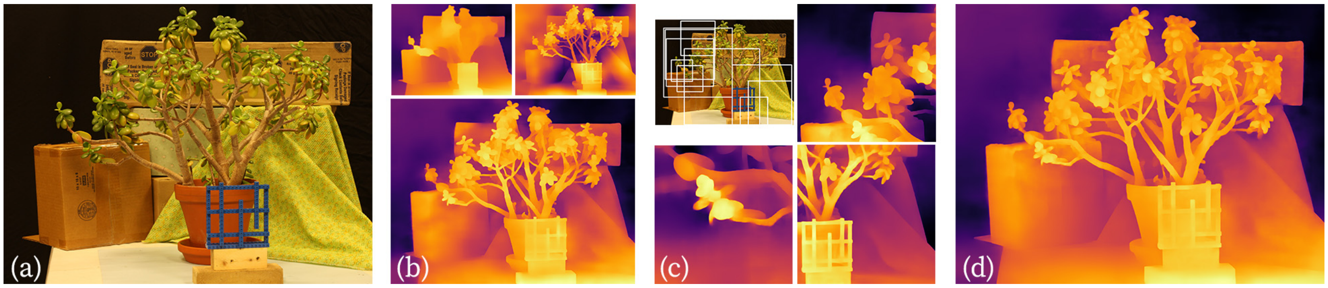

Figure 2: We show a visualization of our pipeline corresponding to Figure 2 of main paper with results obtained from SGR [7]: (b) As before, we start with feeding the image in low- and high-resolution to the network and merge them to get a base estimate. Note how merging generates a stable result although the low- and high-frequency estimates are very different from MiDaS [5]. (c) We demonstrate how the selected patches vary based on the receptive field size of the underlying network. Again, a subset of selected patches with their depth estimates is shown. (d) We merge the patch estimates into our base estimate in (b) to get our final high-resolution result.

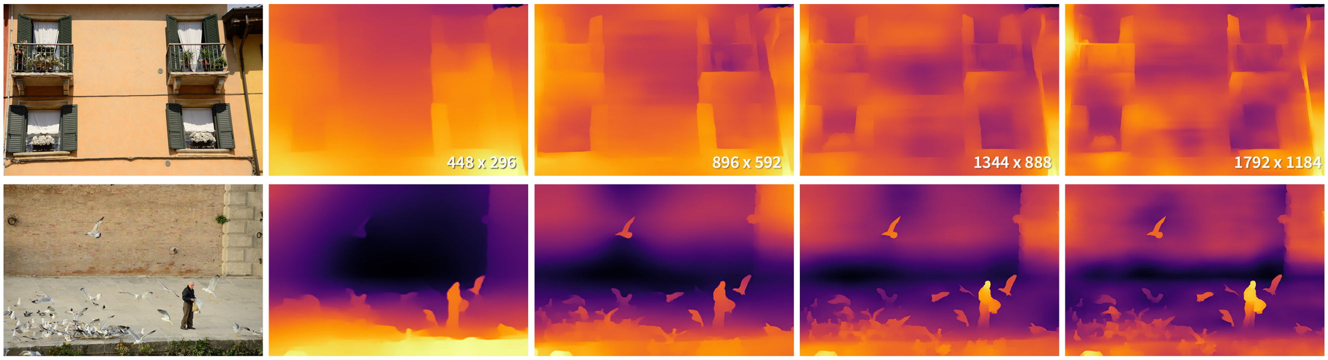

Figure 3: We complete our analysis of different input resolutions in Figure 3 of main paper with additional results from SGR [7]. At small input resolutions, the network [5] is able to estimate the overall structure of the scene successfully but often miss the details in the image, notice the missing birds in the bottom image. As the resolution gets higher, the performance around boundaries gets much better. However, the network starts losing the overall structure of the scene and generates low-frequency artifacts in the estimate. The resolution at which these artifacts start appearing depends on the distribution of contextual cues in the image.

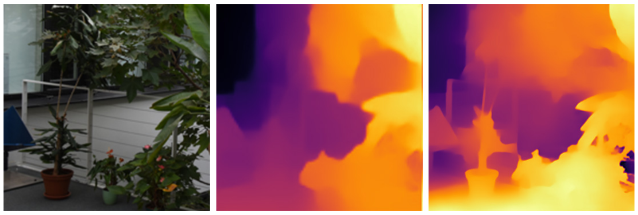

Figure 4: We complement the upsampling results from MiDaS [5]from Figure 5 of main paper here with matching results from SGR [7] The original image with resolution 192 × 192 gains additional details in the depth estimate when fed to the network after upsampling to the receptive field size of 448 × 448 (right) instead of its original resolution (middle).

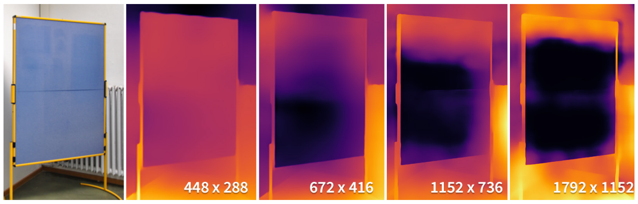

Figure 5: We show the corresponding results to the receptive field test in Figure 4 of main paper based on images from SGR [7]. As the resolution increases starting from the receptive field size of 448, the network again progressively degrades the accuracy.

dilation does not end up covering the whole image, we conclude that the depth cues are further apart than the receptive field size of the network and that we have found our maxi-

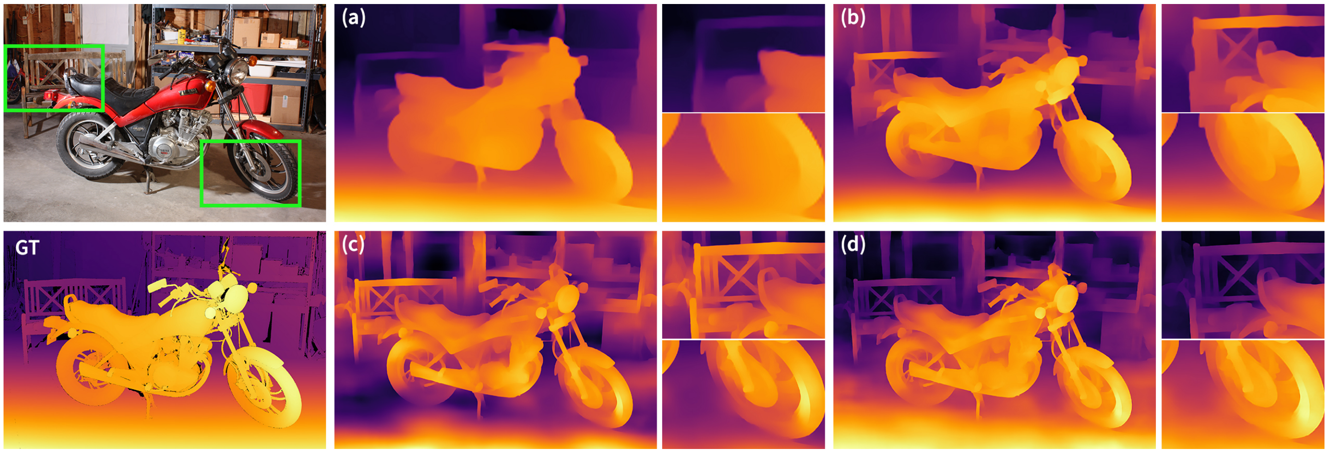

Figure 6: Corresponding to Figure 6 of main paper, we analyze the depth estimates obtained at different resolutions, (a) at the training resolution of SGR[5] at 448×448, (b) at the selected resolution with edges separated at most by 448 pixels, and (c) at a higher resolution that leaves 20% of the pixels without nearby edges. Again the increasing resolution results in sharper predictions compared to (b), but the estimates become unstable in terms of the overall structure, visible especially around the tires. (d) Our merging network is able to fuse the fine-grain details in (c) into the consistent structure in (a) to get the best of two worlds.

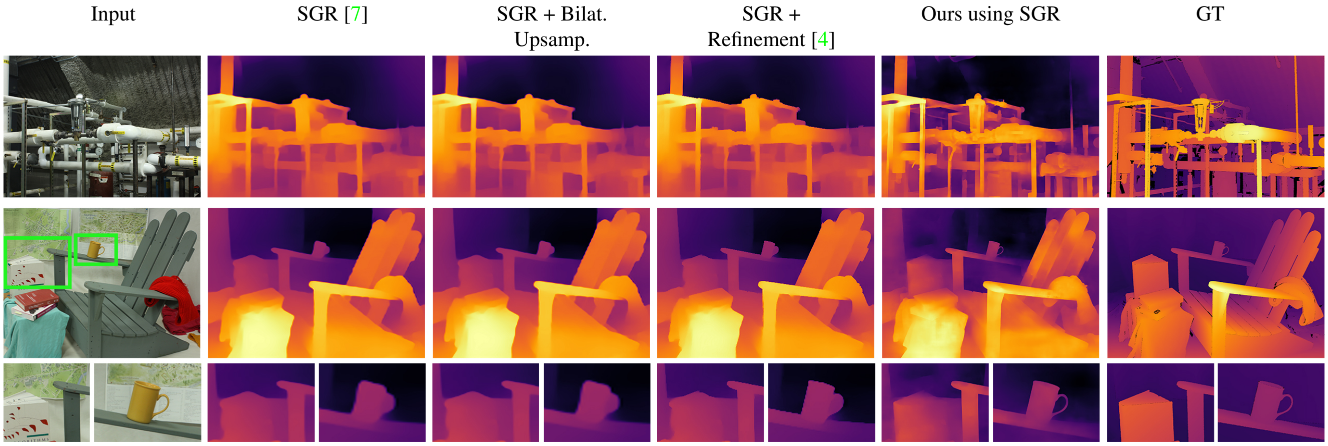

Figure 7: We extend the comparison in Figure 9 of main paper of our method with bilateral upsampling and the refinement method proposed by Niklaus et al. [4] by applying them to SGR [7] output. Refinement methods fail to add any details that does not exist in the original estimation. With our patch-based merging framework, we are able to generate sharp details in the image.

mum input resolution. As we mentioned in the main paper, we allow some percentage denoted by threshold t of these 'black areas', the areas that are not covered in the image after dilation, to happen and rely on double-estimation procedure to remove the artifacts. Table 1 and Table 3 are the extended counterpart of Table 1 of the paper that show the numerical evaluation for different t-values, with and without double-estimation procedure, on two different datasets and different base networks. We show examples of the dilated maps in Figure 10. Note that the maps are inverted to better visualize the actual gradients.

As mentioned in Section 6 in the main document, our minimum patch size is applied on this calculated resolution, thus its affected by the depth cue density of the image. Moreover, we use a scale factor to adjust the minimum patch size for images which have a low overall depth cue density but a high concentration of depth cues within few regions. Figure 11 shows how the minimum patch size is affected by the whole image resolution and how the factor adjusts the patch size when needed. Additionally, we show in Figure 12 the improvement for a patch by choosing the appropriate area size.

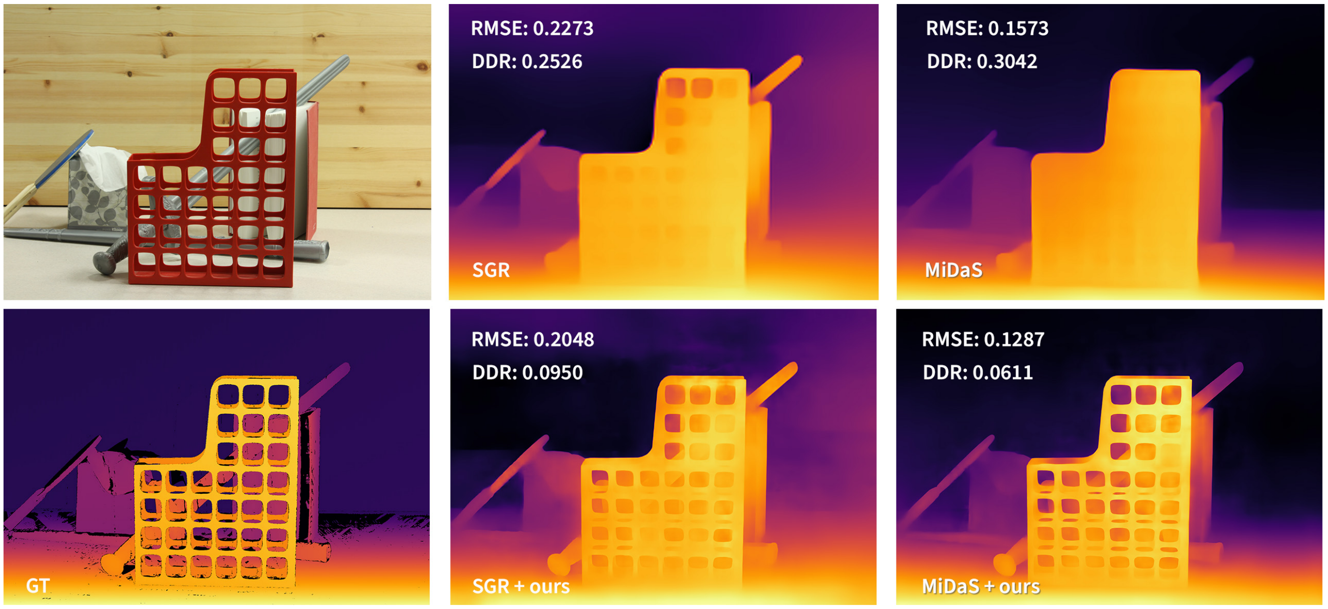

Figure 8: We show the discrepancy between visual improvement and numerical effect on RMSE against D^{3}R. We compare estimations from an RGB image (top left) against its ground truth (bottom left). Note how for the two results from the original versions of SGR [7] (top middle) and MiDaS [5] (top right) the RMSE varies by over 30% though they're visually very similar. Compared to the improvement through our pipeline (bottom row, middle and right), especially on the grid, the RMSE barely changes. The D^{3}R metric shows a clear improvement for the sharper images while showing only a small distance between the estimations from SGR [7] and MiDaS [5].

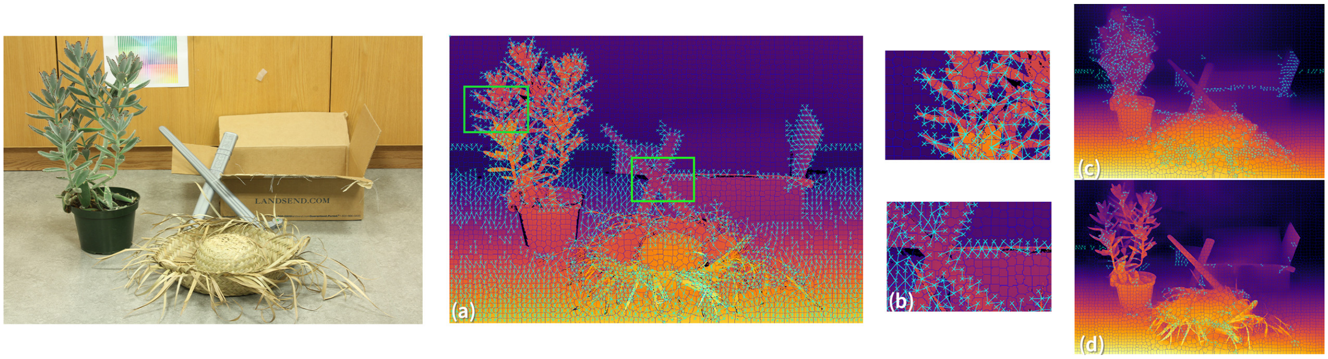

Figure 9: We show an example on the point pairs selected for our D^{3}R metric with RGB input on the left as a reference. We overlay the created superpixels in dark blue and all selected point pairs in light blue over the ground truth depth in a). In b), we show two example regions more closely for visualization. Next, we analyse the error points in light blue for the original MiDaS [5] estimation in c) and compare it with the error points (again light blue) after applying our results in d). Note how the error points especially around the leaves of the plant are drastically reduced.

C. Extended discussion on the merging network training procedure

The goal for our merging network is to seamlessly merge the complementary information coming from the low-resolution and high-resolution depth estimations. A proxy for high-resolution estimation to be used as target in training is the whole-image estimation at a higher resolution. As mentioned in the paper, we use 672*672 as the high resolution to generate these proxy depth maps. To have a smooth training, we discard estimations with artifacts from the training data. We use a guided filtering on the patch estimation with the proxy ground truth as the guide to ensure the proxy ground truth and the patch estimations have the same amount of fine-grained details. Furthermore, to make sure the low resolution estimation training depth data and

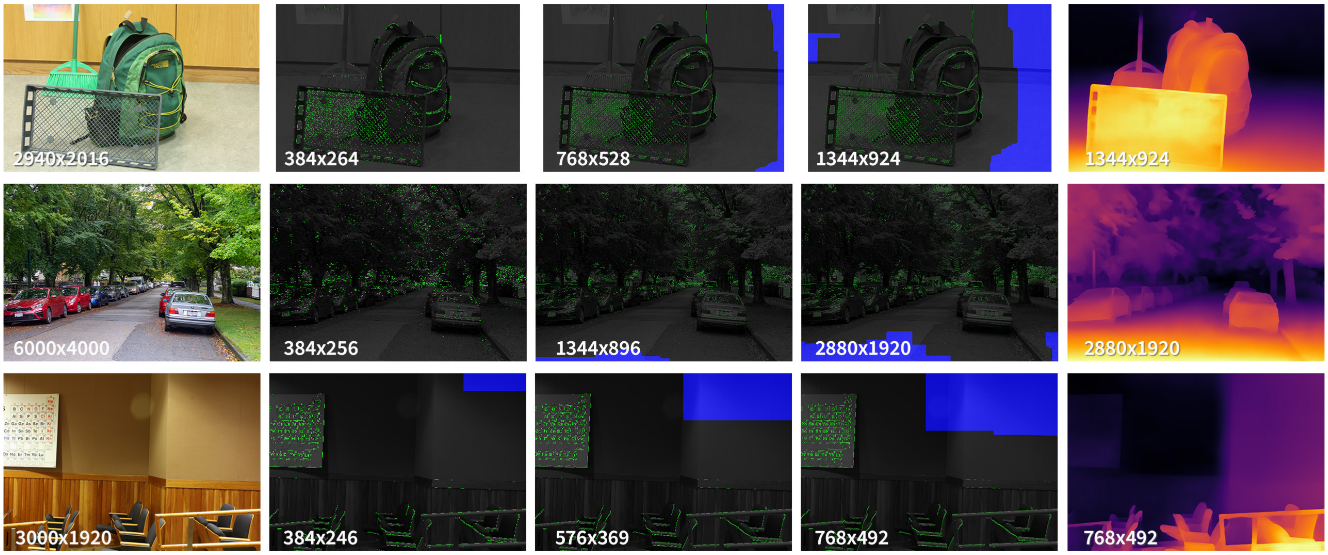

Figure 10: We show dilated gradient maps at different resolutions. The image gradients are displayed in green, blue are the dilated regions without gradients. Notice how the resolution of the whole image estimation (right) increases the more gradients are in the image.

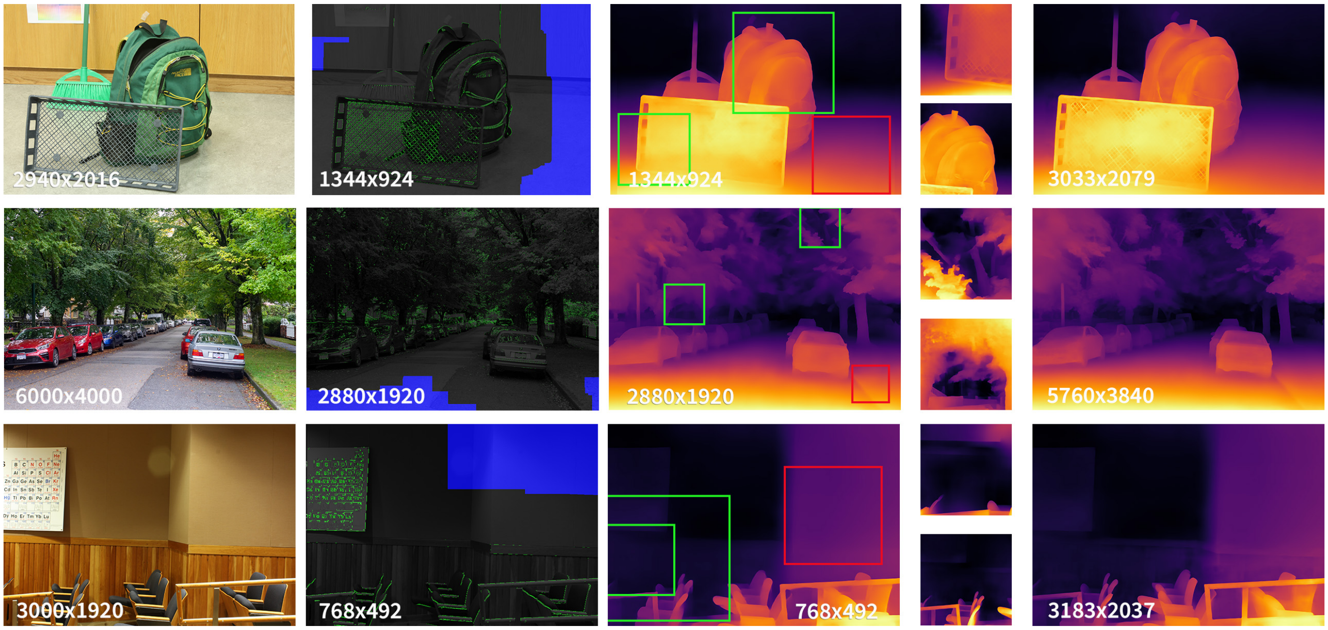

Figure 11: We show the effect of our scale factor on the patch size with example patches. An image with a low amount of gradients in the first row and a sparse gradient map (second column) results in a low-res double estimation (third column). The receptive field size (red rectangle) takes a large portion of the image. The patch size (green rectangles) is close to the size of the receptive field. An image with a high amount of gradients in the second row and a dense gradient map results in a high-res double estimation with the minimum patch size (green rectangles) and the receptive field size (red rectangle) only taking a small part of the image. Lastly, an image with a high amount of gradients located at the bottom in the third row still results in a sparse gradient map. With our factor, the minimum patch size (green rectangles) gets adjusted to a smaller resolution than the receptive field size (red rectangle). We also show the effective resolutions of the whole image estimations and the our final result. Note that the the final resolution of the third image gets increased by our scale factor to be larger than the first image although the whole image estimation is smaller. This is due to the high gradient concentration in the lower part of the third image.

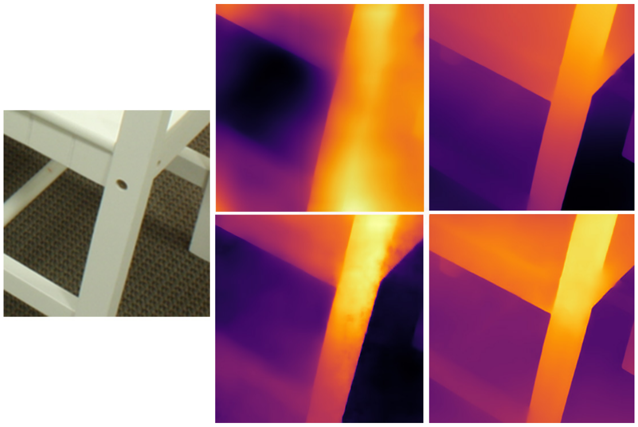

Figure 12: We show the effect of choosing the correct area size for a patch. For the patch on the left, the direct estimation (middle) gives a result with reduced sharpness towards the borders and estimation artifacts. Notice the blurred bit on the bottom left and the dot-like artifacts on the leg. If we increase the area size by 60% (right), the bottom bit is sharp and the artifacts are reduced.

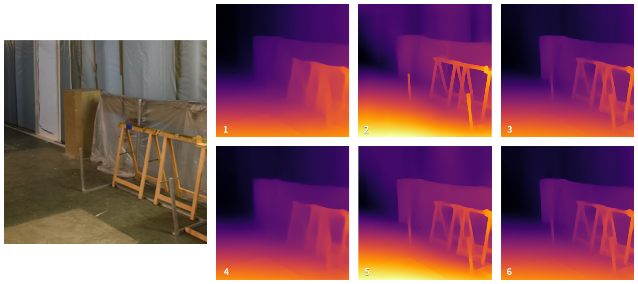

Figure 13: We show examples of the training data that we generate for the merging network. First row (left to right): 1 - low resolution depth patch from low resolution whole-image estimation 2 - high resolution depth estimated at a high resolution from a patch of the rgb image 3 - proxy ground truth patch from high resolution whole-image estimation. Second row: training data after proper guided filtering (left to right): 4 - low resolution input, 5 - high resolution patch estimation input, 6 - ground truth proxy.

the proxy depth share the same absolute value, we blur the proxy image with the guidance of the low resolution depth image to obtain the low resolution depth. We use guided filter instead of a simple Gaussian blur to ensure that the training data has the same characteristics as the low resolution estimations. Figure 13 shows depth images we use to generate the training data alongside with the training images we generate by filtering the depth maps as described.

D. Extended discussion on our results

Running time

Our method consists of two major steps: 1. Finding the optimal whole-image resolution and performing the double-estimation procedure to obtain the base depth estimation and 2. selecting patches followed by a double-estimation procedure on each patch and merging them to the base estimation.

The first step contains searching for the whole-image resolution, two forward paths on the depth estimation network and one forward path on the merge network for the double-estimation procedure. This step takes on average $\sim$5 seconds on high-gradient-density images from the RAISE [1] dataset. The search process is the most time-consuming part here. Discarding the whole-image search process and fixing a number for the whole-image resolution can decrease the run time to less than a second per image independent of the image content. The second step is highly dependant to the image content because of the context dependency of our patch selection method. For each selected patch, we have two forward paths on the depth estimation network and one forward path on the merge network for the double-estimation procedure plus one forward path on the merge network to merge the patch estimation to the base depth. For the complete pipeline, it can happen that the running time for an image on average is increased to $\sim$20 seconds per image on the same dataset.

As we show in Table 2 and 4, our method outperforms the baseline and other state of the art refinement methods with only using double estimation. Quantitative performance alongside the good running time property makes the double-estimate procedure on its own an adequate method to generate higher resolution depth estimations.

Additional evaluation results

We present the extension of Table 1 of the original paper on Middleburry2014 [6] and Ibims-1 [2] in Tables 2 and 4 respectively. For both base networks, our approach excels in both D^{3}R as well as the ORD (original ordinal relation metric [7]) over the raw estimations and other refinement methods. Our method is able to halve the D^{3}R error for estimations on MiDaS and still reduces estimation errors from SGR by roughly 40%. Interestingly, for MiDaS [5] the result already improves if an optimized input size for global estimation is selected. This confirms our assumption that the context density heavily influences the estimation quality. Our method also performs comparably in terms of RMSE, $\delta_{1.25}$, $\delta_{1.25^2}$, $\delta_{1.25^3}$, Log_{10}, SqRel and AbsRel which favor low-frequency global regions.

Table 1: Extension of Table 2 in the main paper by results based on SGR [7] and additional metrics: Whole image estimation performance with changing resolution and double estimation on the Middlebury dataset [6]. Lower is better.

Base		Size	ORD [7]	D^{3}R	RMSE	$\delta > 1.25$	$\delta > 1.25^2$	$\delta > 1.25^3$	Log_{10}	Absrel	Sqrel
MiDaS [5]	Single	Fixed (384)	0.3840	0.3343	0.1708	0.7649	0.5392	0.3587	0.2810	0.7242	0.1042
		Fixed (768)	0.3718	0.2176	0.1527	0.7454	0.5029	0.3200	0.2587	0.7801	0.1014
		Fixed (1152)	0.4266	0.1876	0.1654	0.7403	0.5434	0.3827	0.2812	0.9255	0.1339
		Fixed (1536)	0.4787	0.1892	0.1864	0.7933	0.6082	0.4475	0.3146	1.1855	0.1713
		Context-adaptive ($\mathcal{R}_0$)	0.3554	0.2504	**0.1481**	**0.7161**	**0.4751**	**0.3067**	**0.2471**	0.7283	**0.0916**
		Context-adaptive ($\mathcal{R}_{10}$)	0.4579	0.1971	0.1836	0.7882	0.5908	0.4146	0.3045	1.1725	0.1691
		Context-adaptive ($\mathcal{R}_{20}$)	0.5054	0.1995	0.1980	0.8035	0.6303	0.4686	0.3334	1.3799	0.2006
	Double	Context-adaptive ($\mathcal{R}_0$)	0.3617	0.2585	0.1643	0.7491	0.5241	0.3555	0.2717	0.7016	0.1085
		Context-adaptive ($\mathcal{R}_{10}$)	**0.3491**	0.1838	0.1570	0.7307	0.5046	0.3309	0.2579	0.6581	0.0961
		Context-adaptive ($\mathcal{R}_{20}$)	0.3496	**0.1709**	0.1563	0.7364	0.5086	0.3316	0.2576	0.6540	0.0936
		Context-adaptive ($\mathcal{R}_{30}$)	0.3521	0.1715	0.1567	0.7459	0.5116	0.3337	0.2588	**0.6501**	0.0928
SGR [7]	Single	Fixed (448)	0.4087	0.3889	0.2123	0.7989	0.6089	0.4711	0.3572	0.6804	0.1196
		Fixed (896)	0.4327	0.2968	0.1987	0.7940	0.6005	**0.4414**	0.3372	0.7661	**0.1015**
		Fixed (1344)	0.5044	0.2882	0.2280	0.8271	0.6617	0.5087	0.3829	0.9694	0.1427
		Fixed (1792)	0.5754	0.3018	0.2479	0.8441	0.7000	0.5600	0.4112	1.0438	0.1594
		Context-adaptive ($\mathcal{R}_0$)	0.4312	0.3131	0.1999	**0.7841**	**0.5840**	0.4424	0.3337	0.7243	0.1148
		Context-adaptive ($\mathcal{R}_{10}$)	0.5610	0.3058	0.2368	0.8377	0.6840	0.5434	0.3983	1.0633	0.1477
		Context-adaptive ($\mathcal{R}_{20}$)	0.5974	0.3227	0.2447	0.8465	0.6861	0.5434	0.4034	1.1855	0.1797
	Double	Context-adaptive ($\mathcal{R}_0$)	0.3890	0.3066	0.2034	0.7860	0.5933	0.4528	0.3419	0.6585	0.1247
		Context-adaptive ($\mathcal{R}_{10}$)	**0.3903**	0.2592	0.2000	0.7949	0.6002	0.4533	0.3352	0.6724	0.1217
		Context-adaptive ($\mathcal{R}_{20}$)	0.3944	**0.2540**	0.1983	0.7931	0.5966	0.4488	**0.3309**	**0.6655**	0.1136
		Context-adaptive ($\mathcal{R}_{30}$)	0.3941	0.2569	**0.1980**	0.7964	0.5973	0.4499	0.3311	0.6695	0.1167

Table 2: Quantitative evaluation and comparison with state-of-the-art refinement methods using MiDaS [5] and SGR [7] as base networks on Middleburry2014[6], as extension of Table 1 in the main paper. Lower is better.

base	Method	ORD [7]	D^{3}R	RMSE	$\delta > 1.25$	$\delta > 1.25^2$	$\delta > 1.25^3$	Log_{10}	Absrel	Sqrel
MiDaS [5]	MiDaS	0.3840	0.3343	0.1708	0.7649	0.5392	0.3587	0.2810	0.7242	0.1042
	Refinement-Bilateral	0.3806	0.3366	0.1707	0.7627	0.5396	0.3614	0.2809	0.7287	0.1071
	Refinement-with [4]	0.3826	0.3377	0.1704	0.7622	0.5399	0.3594	0.2797	0.7305	0.1066
	Single-estimation ($\mathcal{R}_0$)	0.3554	0.2504	**0.1481**	**0.7161**	**0.4751**	**0.3067**	**0.2471**	0.7283	**0.0916**
	Double-estimation ($\mathcal{R}_{20}$)	0.3496	0.1709	0.1563	0.7364	0.5086	0.3316	0.2576	**0.6540**	0.0936
	OURS	**0.3467**	**0.1578**	0.1557	0.7406	0.5111	0.3377	0.2586	0.6676	0.1014
SGR [7]	SGR	0.4087	0.3889	0.2123	0.7989	0.6089	0.4711	0.3572	0.6804	0.1196
	Refinement-Bilateral	0.4078	0.3904	0.2122	0.7990	0.6080	0.4714	0.3563	0.6825	0.1227
	Refinement-with [4]	0.4081	0.3880	0.2115	0.7993	0.6054	0.4667	0.3533	0.6712	0.1194
	Single-estimation ($\mathcal{R}_0$)	0.4312	0.3131	0.1999	**0.7841**	**0.5840**	**0.4424**	0.3337	0.7243	0.1148
	Double-estimation ($\mathcal{R}_{20}$)	0.3944	0.2540	0.1983	0.7931	0.5966	0.4488	0.3309	**0.6655**	**0.1136**
	OURS	**0.3879**	**0.2324**	**0.1973**	0.7891	0.5911	0.4470	**0.3302**	0.6705	0.1200

We also notice that our relative improvement on D^{3}R for Ibims-1 [2] is less significant, though with a reduction of $> 12\%$ for MiDaS [5] estimations still visible. This is explainable if we take the original image resolution into account. For a dataset like Ibims-1 with a spatial resolution of 640x480 pixels, the original MiDaS training resolution

Table 3: Extension of Table 2 in the main paper by results based on SGR [7] and additional metrics: Whole image estimation performance with changing resolution and double estimation on the Ibims1 dataset [2]. Lower is better.

Base		Size	ORD [7]	D^3R	RMSE	$\delta > 1.25$	$\delta > 1.25^2$	$\delta > 1.25^3$	Log_{10}	Absrel	Sqrel
MiDaS [5]	Single	Fixed (384)	0.4002	0.3698	0.1596	0.6345	**0.4127**	**0.2712**	**0.2473**	**2.0325**	0.7903
		Fixed (768)	0.4657	**0.3165**	0.1700	0.6658	0.4468	0.3019	0.2637	2.1768	0.7954
		Fixed (1152)	0.5876	0.3330	0.1917	0.7153	0.4983	0.3486	0.2930	2.5442	0.9339
		Fixed (1536)	0.6694	0.4005	0.2101	0.7325	0.5225	0.3767	0.3146	2.9826	1.2006
		Context-adaptive ($\mathcal{R}_0$)	0.4504	0.3269	0.1687	**0.6633**	0.4442	0.2896	0.2589	2.1160	**0.7857**
		Context-adaptive ($\mathcal{R}_{10}$)	0.5868	0.3762	0.1928	0.7059	0.4820	0.3354	0.2917	2.7206	1.0936
		Context-adaptive ($\mathcal{R}_{20}$)	0.6249	0.3944	0.2025	0.7165	0.5067	0.3620	0.3046	2.8297	1.1471
	Double	Context-adaptive ($\mathcal{R}_0$)	**0.3853**	0.3321	0.1600	0.6366	0.4138	0.2738	0.2475	2.0352	0.7996
		Context-adaptive ($\mathcal{R}_{10}$)	0.4152	0.3258	**0.1596**	0.6378	0.4135	0.2725	0.2475	2.0399	0.7938
		Context-adaptive ($\mathcal{R}_{20}$)	0.4112	0.3272	0.1597	0.6386	0.4139	0.2716	0.2474	2.0449	0.7943
		Context-adaptive ($\mathcal{R}_{30}$)	0.4020	0.3325	0.1602	0.6407	0.4157	0.2727	0.2479	2.0482	0.7964
SGR [7]	Single	Fixed (448)	0.5555	0.4736	**0.1956**	0.7513	0.5411	0.3808	0.3056	**1.9813**	**0.7680**
		Fixed (896)	0.7224	0.4941	0.2263	0.7978	0.6163	0.4568	0.3476	2.4066	0.9271
		Fixed (1344)	0.8208	0.5850	0.2456	0.8272	0.6608	0.5045	0.3762	2.7507	1.0965
		Fixed (1792)	0.8863	0.6573	0.2602	0.8413	0.6857	0.5370	0.3983	3.0460	1.3221
		Context-adaptive ($\mathcal{R}_0$)	0.6343	0.4901	0.2146	0.7856	0.5961	0.4366	0.3351	2.1914	0.7978
		Context-adaptive ($\mathcal{R}_{10}$)	0.7820	0.5929	0.2412	0.8255	0.6590	0.5078	0.3762	2.6508	1.0119
		Context-adaptive ($\mathcal{R}_{20}$)	0.8540	0.6244	0.2509	0.8315	0.6714	0.5214	0.3894	2.8449	1.1453
	Double	Context-adaptive ($\mathcal{R}_0$)	**0.5405**	**0.4502**	0.1969	**0.7457**	0.5367	**0.3804**	0.3050	2.0008	0.8088
		Context-adaptive ($\mathcal{R}_{10}$)	0.5591	0.4681	0.1971	0.7477	0.5372	0.3832	0.3052	2.0205	0.8050
		Context-adaptive ($\mathcal{R}_{20}$)	0.5591	0.4829	0.1967	0.7473	**0.5352**	0.3821	0.3047	2.0285	0.8034
		Context-adaptive ($\mathcal{R}_{30}$)	0.5643	0.4847	0.1967	0.7476	0.5362	0.3824	**0.3041**	2.0301	0.8014

Table 4: Quantitative evaluation and comparison with state-of-the-art refinement methods using MiDaS [5] and SGR [7] as base networks on Ibims1[2], as extension of Table 1 in the main paper. Lower is better.

base	Method	ORD [7]	D^3R	RMSE	$\delta > 1.25$	$\delta > 1.25^2$	$\delta > 1.25^3$	Log_{10}	Absrel	Sqrel
MiDaS [5]	MiDaS	0.4002	0.3698	**0.1596**	**0.6345**	0.4127	0.2712	0.2473	2.0325	0.7903
	Refinement-Bilateral	0.3982	0.3768	0.1596	0.6350	**0.4120**	**0.2705**	**0.2471**	**2.0238**	0.7875
	Refinement-with [4]	0.4006	0.3761	0.1600	0.6351	0.4132	0.2716	0.2478	2.0519	0.7986
	Single-estimation ($\mathcal{R}_0$)	0.4504	0.3269	0.1687	0.6633	0.4442	0.2896	0.2589	2.1160	**0.7857**
	Double-estimation ($\mathcal{R}_{20}$)	0.4112	0.3272	0.1597	0.6386	0.4139	0.2716	0.2474	2.0449	0.7943
	OURS	**0.3938**	**0.3222**	0.1598	0.6390	0.4144	0.2718	0.2477	2.0510	0.8007
SGR [7]	SGR	0.5555	0.4736	0.1956	0.7513	0.5411	0.3808	0.3056	1.9813	**0.7680**
	Refinement-Bilateral	0.5551	0.4750	0.1956	0.7501	0.5416	0.3822	0.3062	**1.9754**	0.7709
	Refinement-with [4]	0.5488	0.4780	**0.1953**	0.7482	0.5388	**0.3789**	**0.3041**	1.9953	0.7751
	Single-estimation ($\mathcal{R}_0$)	0.6343	0.4901	0.2146	0.7856	0.5961	0.4366	0.3351	2.1914	0.7978
	Double-estimation ($\mathcal{R}_{20}$)	0.5591	0.4829	0.1967	0.7473	0.5352	0.3821	0.3047	2.0285	0.8034
	OURS	**0.5538**	**0.4671**	0.1965	**0.7460**	**0.5340**	0.3796	0.3039	2.0247	0.8055

is already relatively close to the optimal size for estimation. This leaves less room for improvement through input optimization. We conclude that our method is creating more significant results the higher the original image resolution is. Figure 14 shows corresponding results from high-resolution images from the RAISE [1] dataset.

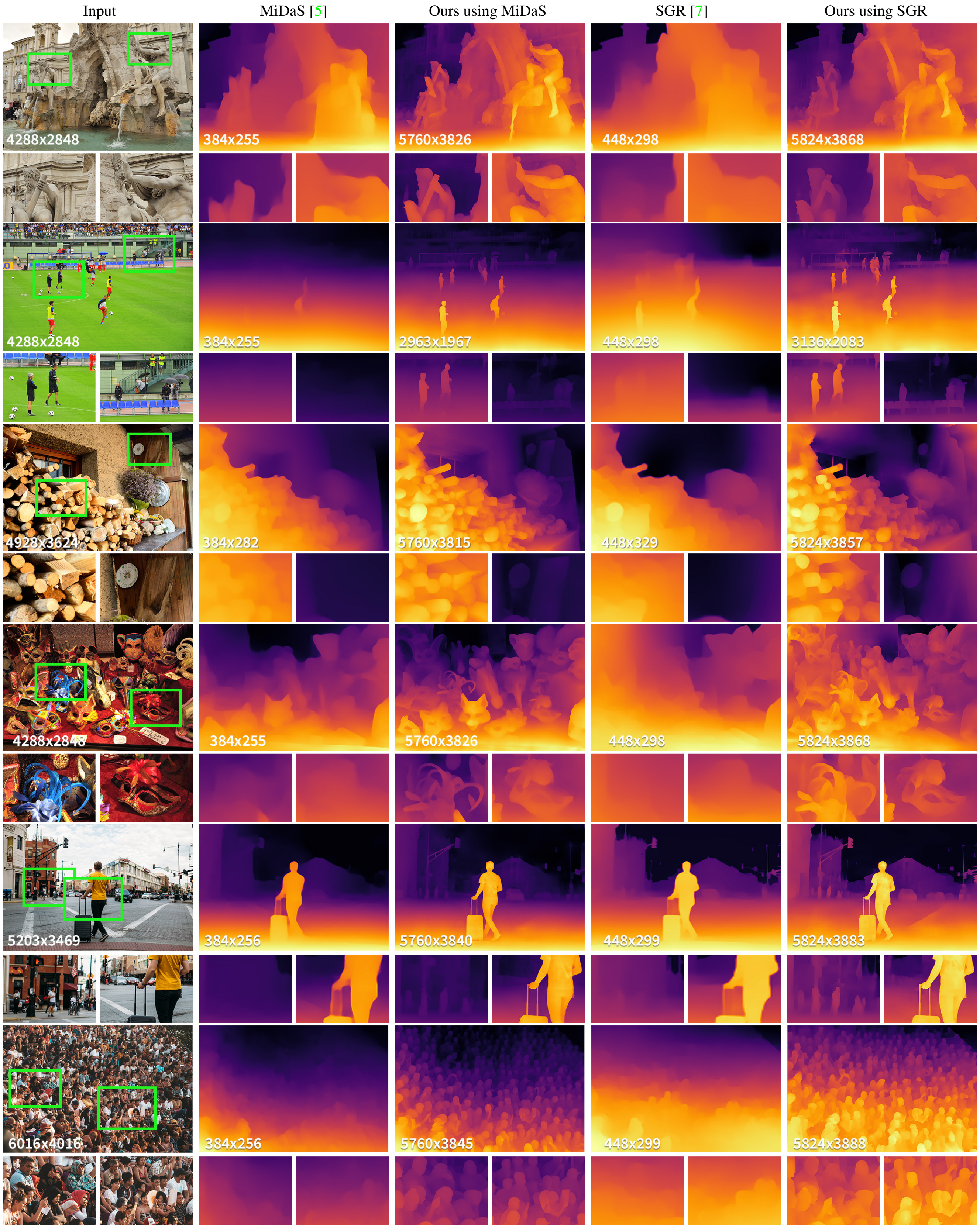

Figure 14: We present additional results to Figure 8 using MiDaS [5] and the Structure-Guided Ranking Loss method [7] compared to the original methods run at their default size with effective resolutions for every image.

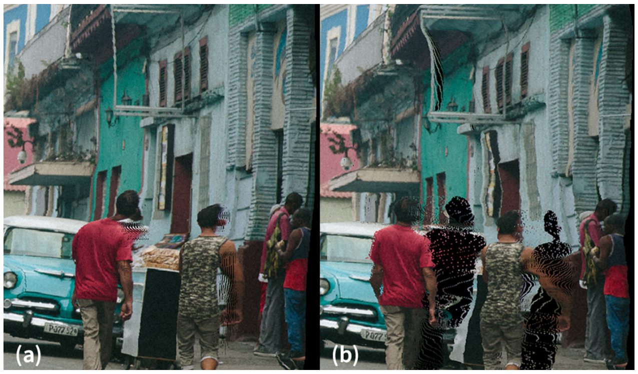

Figure 15: The patch-based procedure is causing a distortion of the scene geometry. This causes straight edges like the walls in b) to appear bumpy compared to the low-resolution full image estimation in a).

Geometric accuracy

Monocular depth estimation networks utilize depth cues to predict the depth relations within the scene. They can determine if one object is closer than another, but fail in establishing the absolute distance to the camera. Our method is limited by the same constrains. Additionally, every selected patch is treated by the underlying depth estimation network as a unique image. This causes the depth range of the patch estimation to differ from the corresponding area within the full image estimation, both regarding the limits as well as the distribution. Our merging network can reduce this difference greatly, but is not able to eliminate it completely. This results in visible distortions of the scene geometry as shown in Figure 15. Especially straight edges appear bumpy.

Sensitivity to input image noise

During the evaluation we noticed that our method can not generate reasonable estimations on the NYU [3] dataset. The image resolution for NYU is not high, but comparable to Ibims-1 [2] for which we achieve competitive results. The main difference between these datasets is the quality of the RGB images in terms of presence of noise and sharpness of the images. We notice that the presence of the noise highly effects the performance of the underlying base depth estimation network in generating higher resolution estimates. As a result, our method also fails to generate reasonable estimations. In order to examine the effect of noise on the underlying base depth estimation network we added noise to the Ibims-1 [2] dataset RGB images. Table 5 shows the effect of adding noise to the Ibims-1 [2] dataset.

Table 5: Effect of adding noise the RGB images on the resulted depth estimations using Ibims-1 [2] dataset.

	Noise	ORD [7]	D^{3}R	RMSE	$\delta > 1.25$
Double estimation ($\mathcal{R}_{20}$)	None	**0.4112**	**0.3272**	**0.1597**	**0.6386**
	var=0.001	0.4529	0.3830	0.1634	0.6484
	var=0.002	0.4455	0.4021	0.1637	0.6515
OURS	None	**0.3938**	**0.3222**	**0.1598**	**0.6390**
	var=0.001	0.4264	0.3813	0.1638	0.6464
	var=0.002	0.4430	0.3984	0.1631	0.6470

References

[1] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proc. ACM Multimedia Systems Conference*, 2015.

[2] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of CNN-Based Single-Image Depth Estimation Methods. In *Proc. ECCV Workshops*, 2018.

[3] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proc. ECCV*, 2012.

[4] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3D Ken Burns effect from a single image. *ACM Trans. Graph.*, 38(6):1–15, 2019.

[5] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[6] D. Scharstein, H. Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Proc. GCPR*, 2014.

[7] Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. Structure-guided ranking loss for single image depth prediction. In *Proc. CVPR*, 2020.

[8] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. CVPR*, 2018.

[9] Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. Learning ordinal relationships for mid-level vision. In *Proc. ICCV*, 2015.