Stereo Anywhere: Robust Zero-Shot Deep Stereo Matching Even Where Either Stereo or Mono Fail

Supplementary Material

This document reports additional material concerning CVPR paper "Stereo Anywhere: Robust Zero-Shot Deep Stereo Matching Even Where Either Stereo or Mono Fail". Specifically:

- First, we present an extended description of our proposed architecture in Sec. 3, including detailed formulations of the monocular correlation volume (Sec. 7.1), differentiable monocular scaling (Sec. 7.2), cost volume augmentation (Sec. 7.3), volume truncation (Sec. 7.4), and training supervision (Sec. 7.5).
- We then report extensive ablation studies in Sec. 8 demonstrating how our stereo matching architecture effectively generalizes across different state-of-the-art monocular depth networks (Sec. 8.1), showing consistent improvements over baseline stereo methods regardless of the specific VFM employed. Then, we show qualitatively the impact of the truncated cost volume augmentation on disparity estimation on non-Lambertian surfaces (Sec. 8.2). Furthermore, we include an analysis of runtime performance and memory consumption (Sec. 8.3) across different input resolutions and VFMs.
- Finally, we present extensive qualitative results in Sec. 9 across multiple datasets, demonstrating the effectiveness of our method in dealing with challenging scenarios such as non-Lambertian surfaces, transparent objects and textureless regions.

7. Method Overview: Additional Details

In this section, we enrich the description of Stereo Anywhere architecture.

7.1. Monocular Correlation Volume

Given the monocular depth estimations $\mathbf{M}_L \in \mathbb{R}^{1\times H\times W}$ and $\mathbf{M}_R \in \mathbb{R}^{1\times H\times W}$, we aim to estimate the normal maps $\nabla_L \in \mathbb{R}^{3\times \frac{H}{4}\times \frac{W}{4}}$ and $\nabla_R \in \mathbb{R}^{3\times \frac{H}{4}\times \frac{W}{4}}$ to construct the 3D correlation volume $\mathbf{V}_M \in \mathbb{R}^{\frac{H}{4}\times \frac{W}{4}\times \frac{W}{4}}$. We decide to use ∇_L and ∇_R instead of extracting additional features from $\mathbf{M}_L$ and $\mathbf{M}_R$ because $\mathbf{M}_L$ and $\mathbf{M}_R$ already provide high-level information. Furthermore, normal maps can handle depth inconsistencies between $\mathbf{M}_L$ and $\mathbf{M}_R$ that can occur for example when a foreground object is visible only in a single view. We downsample $\mathbf{M}_L$ and $\mathbf{M}_R$ to 1/4 – bilinear interpolation, then we estimate ∇_L and ∇_R – spatial gradient:

$$\nabla = \frac{\nabla^*}{\|\nabla^*\|}, \quad \nabla^* = \begin{bmatrix} -\frac{\partial\left(\lambda \mathbf{M}_{\frac{1}{4}}\right)}{\partial x} & -\frac{\partial\left(\lambda \mathbf{M}_{\frac{1}{4}}\right)}{\partial y} & 1 \end{bmatrix}, \quad \lambda = \frac{1}{10}\cdot\frac{W}{4} \tag{8}$$

where λ is a gain factor that is proportional to W, which permits to achieve scale-invariant normal maps.

Given the absence of texture in normal maps, $\mathbf{V}_M$ will be not ambiguous only in edges. To alleviate this problem, we segment $\mathbf{V}_M$ – the relative depth priors from $\mathbf{M}_L$ and $\mathbf{M}_R$: doing so we aim to reduce the ambiguity by forcing the matching only in similar depth regions (*e.g.*, foreground objects cannot match with background object since the correlation score is masked to zero). Considering Eq. (3), we calculate masks ${\mathcal{M}_L}^n$ and ${\mathcal{M}_R}^n$ as follows:

$$({\mathcal{M}_L}^n)_{ij} = \begin{cases} 1 \text{ if } \frac{n}{N} \le (\mathbf{M}_L)_{ij} < \frac{n+1}{N} \\ 0 \text{ otherwise} \end{cases} \qquad ({\mathcal{M}_R}^n)_{ik} = \begin{cases} 1 \text{ if } \frac{n}{N} \le (\mathbf{M}_R)_{ik} < \frac{n+1}{N} \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

To further deal with the ambiguity, we improve the 3D Convolutional Regularization model ϕ_A – an adapted version of CoEx [4] correlation volume excitation that exploits both views $\mathbf{M}_L$, $\mathbf{M}_R$:

$$({\mathbf{V}'}^s{}_M)_{fijk} = \sigma\left(({\mathbf{f}_L}^s)_{fij}\right) \odot \sigma\left(({\mathbf{f}_R}^s)_{fik}\right) \odot ({\mathbf{V}}^s{}_M)_{fijk} \tag{10}$$

where ${\mathbf{V}'}^s{}_M$ is the excited volume, $\sigma(\cdot)$ is the sigmoid function, $\odot$ is the element-wise product, ${\mathbf{V}}^s{}_M \in \mathbb{R}^{F\times \frac{H}{s}\times \frac{W}{s}\times \frac{W}{s}}$ is an intermediate correlation feature volume at scale s with F features inside module ϕ_A, ${\mathbf{f}_L}^s \in \mathbb{R}^{F\times \frac{H}{s}\times \frac{W}{s}\times 1}$ and ${\mathbf{f}_R}^s \in \mathbb{R}^{F\times \frac{H}{s}\times 1\times \frac{W}{s}}$ are shallow 2D conv-features extracted from $\mathbf{M}_L$ and $\mathbf{M}_R$ downsampled at proper scale.

7.2. Differentiable Monocular Scaling

As detailed in Sec. 3.2, volume $\mathbf{V}_M^D$ is used also to estimate the coarse disparity maps $\hat{\mathbf{D}}_L$ $\hat{\mathbf{D}}_R$, while volume $\mathbf{V}_M^C$ is utilized to estimate confidence maps $\hat{\mathbf{C}}_L$ $\hat{\mathbf{C}}_R$. $\hat{\mathbf{D}}_L$ $\hat{\mathbf{C}}_L$ and $\hat{\mathbf{D}}_R$ $\hat{\mathbf{C}}_R$ are used to scale respectively $\mathbf{M}_L$ and $\mathbf{M}_R$. As described in Eq. (4), we can estimate left disparity from a correlation volume – a softargmax operation on the last W dimension of $\mathbf{V}_M^D$ and – the relationship between left disparity and correlation. Here we report an extended version of Eq. (4) with the explicit formula for softargmax operator:

$$(\hat{\mathbf{D}}_L)_{ij} = j - \left(\text{softargmax}_L(\mathbf{V}_M^D)\right)_{ij} = j - \sum_d^{\frac{W}{4}} d \cdot \frac{e^{(\mathbf{V}_M^D)_{ijd}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^D)_{ijf}}} \tag{11}$$

At the same time, given the relationship between right disparity and correlation $d_R = k_L - k_R$ we can estimate the right disparity performing a softargmax on the first W dimension of $\mathbf{V}_M^D$:

$$(\hat{\mathbf{D}}_R)_{ik} = \left(\text{softargmax}_R(\mathbf{V}_M^D)\right)_{ik} - k = \sum_d^{\frac{W}{4}} d \cdot \frac{e^{(\mathbf{V}_M^D)_{idk}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^D)_{ifk}}} - k \tag{12}$$

Disparity maps $\hat{\mathbf{D}}_L$ $\hat{\mathbf{D}}_R$ are used in combination with confidence maps $\hat{\mathbf{C}}_L$ $\hat{\mathbf{C}}_R$ to obtain a robust scaling. We present an expanded version of the information entropy based confidence estimation (Eq. (5)), with the explicit formula for softmax operator:

$$(\hat{\mathbf{C}}_L)_{ij} = 1 + \frac{\sum_d^{\frac{W}{4}} \left(\text{softmax}_L(\mathbf{V}_M^C)\right)_{ijd} \cdot \log_2\left(\left(\text{softmax}_L(\mathbf{V}_M^C)\right)_{ijd}\right)}{\log_2(\frac{W}{4})} = 1 + \frac{\sum_d^{\frac{W}{4}} \frac{e^{(\mathbf{V}_M^C)_{ijd}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^C)_{ijf}}} \cdot \log_2\left(\frac{e^{(\mathbf{V}_M^C)_{ijd}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^C)_{ijf}}}\right)}{\log_2(\frac{W}{4})} \tag{13}$$

In the same way, we estimate right confidence map $\hat{\mathbf{C}}_R$ performing a softmax operation on the first W dimension of $\mathbf{V}_M^C$:

$$(\hat{\mathbf{C}}_R)_{ik} = 1 + \frac{\sum_d^{\frac{W}{4}} \left(\text{softmax}_R(\mathbf{V}_M^C)\right)_{idk} \cdot \log_2\left(\left(\text{softmax}_R(\mathbf{V}_M^C)\right)_{idk}\right)}{\log_2(\frac{W}{4})} = 1 + \frac{\sum_d^{\frac{W}{4}} \frac{e^{(\mathbf{V}_M^C)_{idk}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^C)_{ifk}}} \cdot \log_2\left(\frac{e^{(\mathbf{V}_M^C)_{idk}}}{\sum_f^{\frac{W}{4}} e^{(\mathbf{V}_M^C)_{ifk}}}\right)}{\log_2(\frac{W}{4})} \tag{14}$$

To improve the robustness of the scaling, we introduce a softLRC operator to classify occlusions as low-confidence pixels and consequentially mask out them from $\hat{\mathbf{C}}_L$ and $\hat{\mathbf{C}}_R$. We define the softLRC operator as follows:

$$\text{softLRC}_L(\mathbf{D}, \mathbf{D}_R) = \frac{\log\left(1 + \exp\left(T_{\text{LRC}} - |\mathbf{D}_L - \mathcal{W}_L(\mathbf{D}_L, \mathbf{D}_R)|\right)\right)}{\log(1 + \exp(T_{\text{LRC}}))} \tag{15}$$

where $T_{\text{LRC}} = 1$ is the LRC threshold and $\mathcal{W}_L(\mathbf{D}_L, \mathbf{D}_R)$ is the warping operator that uses the left disparity $\mathbf{D}_L$ to warp the right disparity $\mathbf{D}_R$ into the left view.

Finally, we can estimate the scale $\hat{s}$ and shift $\hat{t}$ – a differentiable weighted least-square approach. We report here the expanded form of Eq. (6):

$$\min_{\hat{s},\hat{t}} \left\| \sqrt{\hat{\mathbf{C}}_L} \odot \left[(\hat{s}\mathbf{M}_L + \hat{t}) - \hat{\mathbf{D}}_L \right] \right\|_F + \left\| \sqrt{\hat{\mathbf{C}}_R} \odot \left[(\hat{s}\mathbf{M}_R + \hat{t}) - \hat{\mathbf{D}}_R \right] \right\|_F \tag{16}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

7.3. Cost Volume Augmentations

Volume augmentations are necessary when the training set – *e.g.*, Sceneflow [61] – does not model particularly complex scenarios where a VFM could be useful, for example, when experiencing non-Lambertian surfaces. Without any augmentation of this kind, the stereo network would simply overlook the additional information from the monocular branch. As detailed in the main paper, we propose three volume augmentations and a monocular augmentation to overcome this issue. In this supplementary section, we explain the rationale behind the introduction of each augmentation:

- *Volume Rolling*: non-Lambertian surfaces such as mirrors and glasses violate the geometry constraints, leading to a high matching peak in a wrong disparity bin. This augmentation emulates this behavior by shifting some among the matching peaks to a random position: consequentially, Stereo Anywhere learns to retrieve the correct peak from the other branch.

- *Volume Noising* and *Volume Zeroing*: we introduce noise and false peaks into the correlation volume to simulate scenarios with texture-less regions, repeating patterns, and occlusions.

- *Perfect Monocular Estimation*: instead of acting inside the correlation volumes, we can substitute the prediction of the VFM with a perfect monocular map – the ground truth normalized between $[0, 1]$. This perfect prediction is noise-free and therefore the monocular branch of Stereo Anywhere will likely gain importance during the training process.

7.4. Volume Truncation

The proposed volume truncation strategy further helps Stereo Anywhere to handle mirror surfaces. Here we introduce additional details about fuzzy operators – useful to make a boolean expression differentiable – and the sigmoid curve used to truncate the volume $\mathbf{V}_S$ – the truncate mask $(\mathbf{T})_{ij} = \left[\left((\hat{\mathbf{M}}_L)_{ij} > (\hat{\mathbf{D}}_L)_{ij}\right) \wedge (\mathbf{C}_M)_{ij}\right] \vee \left[(\mathbf{C}_M)_{ij} \wedge \neg(\hat{\mathbf{C}}_L)_{ij}\right]$.

We can replace operators AND ($\wedge$), OR ($\vee$), NOT ($\neg$) and GREATER ($>$) inside $\mathbf{T}$ with the fuzzy counterparts $\text{AND}_\text{F}(A,B) = A \cdot B$, $\text{OR}_\text{F}(A,B) = A + B - A \cdot B$, $\text{NOT}_\text{F}(A) = 1 - A$ and $\text{GREATER}_\text{F}(A,B) = \sigma(A - B)$, obtaining the fuzzy truncate mask $\mathbf{T}_\text{F}$:

$$\begin{aligned}(\mathbf{T}_\text{F})_{ij} &= (\mathbf{T}_\text{F}^A)_{ij} + (\mathbf{T}_\text{F}^B)_{ij} - (\mathbf{T}_\text{F}^A)_{ij} \cdot (\mathbf{T}_\text{F}^B)_{ij} \\ (\mathbf{T}_\text{F}^A)_{ij} &= (\mathbf{C}_M)_{ij} \cdot \sigma\left((\hat{M}_L)_{ij} - (\hat{D}_L)_{ij}\right) \\ (\mathbf{T}_\text{F}^B)_{ij} &= (\mathbf{C}_M)_{ij} \cdot \left(1 - (\hat{\mathbf{C}}_L)_{ij}\right)\end{aligned} \tag{17}$$

where $\mathbf{T}_\text{F}^A$ and $\mathbf{T}_\text{F}^B$ are respectively the left section and the right section of the OR_F of mask $\mathbf{T}_\text{F}$. Next, we can apply threshold T_m to achieve the final fuzzy mask $\mathbf{T}'_\text{F}$ as follows:

$$(\mathbf{T}'_\text{F})_{ij} = \sigma\left((\mathbf{T}_\text{F})_{ij} - T_m\right) \tag{18}$$

.

Finally, we can use the fuzzy truncate mask $\mathbf{T}'_\text{F}$ and the scaled monocular map $\hat{\mathbf{M}}_L$ to generate the sigmoid-based truncation volume $\mathbf{V}_T$:

$$(\mathbf{V}_T)_{ijk} = \left(1 - (\mathbf{T}'_\text{F})_{ij}\right) + (\mathbf{T}'_\text{F})_{ij} \cdot \left[\sigma\left(j - (\hat{\mathbf{M}}_L)_{ij} - k\right) \cdot (1 - G) + G\right] \tag{19}$$

where $G = 0.9$ attenuates the impact of the truncation. The correlation volume $\mathbf{V}_S$ is truncated through an element-wise product with $\mathbf{V}_T$.

7.5. Training Supervision

We supervise the iterative module – the L1 loss with exponentially increasing weights [55]:

$$\mathcal{L}_\text{A} = \sum_{l=1}^{L} \gamma^{L-l} \|\mathbf{D}^l - \mathbf{D}_\text{Lgt}\|_1 \tag{20}$$

where L is the total number of iterations made by the update operator and $\mathbf{D}_\text{Lgt}$ is the ground truth of the left disparity map. Furthermore, we supervise the outputs $\hat{\mathbf{D}}_L$, $\hat{\mathbf{D}}_R$, $\hat{\mathbf{M}}_L$, $\hat{\mathbf{M}}_R$, $\hat{\mathbf{C}}_L$, $\hat{\mathbf{C}}_R$ of the monocular branch – respectively L1 loss and normal loss for $\hat{\mathbf{D}}_L$ $\hat{\mathbf{D}}_R$, L1 loss for $\hat{\mathbf{M}}_L$ $\hat{\mathbf{M}}_R$ and Binary Cross Entropy (BCE) loss for $\hat{\mathbf{C}}_L$ $\hat{\mathbf{C}}_R$:

$$\mathcal{L}_\text{B} = \|\hat{\mathbf{D}}_L - \mathbf{D}_\text{Lgt}\|_1 + \psi \left\|\mathbf{1} - \nabla_L \cdot \hat{\nabla}_L\right\|_1 \quad \left(\nabla_L \cdot \hat{\nabla}_L\right)_{ij} = \sum_h (\nabla_L)_{hij} \cdot (\hat{\nabla}_L)_{hij} \tag{21}$$

$$\mathcal{L}_\text{C} = \|\hat{\mathbf{D}}_R - \mathbf{D}_\text{Rgt}\|_1 + \psi \left\|\mathbf{1} - \nabla_R \cdot \hat{\nabla}_R\right\|_1 \quad \left(\nabla_R \cdot \hat{\nabla}_R\right)_{ik} = \sum_h (\nabla_L)_{hik} \cdot (\hat{\nabla}_L)_{hik} \tag{22}$$

$$\mathcal{L}_\text{D} = \|\hat{\mathbf{M}}_L - \mathbf{D}_\text{Lgt}\|_1 \quad \mathcal{L}_\text{E} = \|\hat{\mathbf{M}}_R - \mathbf{D}_\text{Rgt}\|_1 \tag{23}$$

$$\mathcal{L}_\text{F} = \text{BCE}(\hat{\mathbf{C}}_L, \mathbf{C}_\text{Lgt}), \quad (\mathbf{C}_\text{Lgt})_{ij} = \frac{\log\left(1 + \exp\left(T_\text{LRC} - \left|(\hat{\mathbf{D}}_L)_{ij} - (\mathbf{D}_\text{Lgt})_{ij}\right|\right)\right)}{\log(1 + \exp(T_\text{LRC}))} \tag{24}$$

$$\mathcal{L}_\text{G} = \text{BCE}(\hat{\mathbf{C}}_R, \mathbf{C}_\text{Rgt}), \quad (\mathbf{C}_\text{Rgt})_{ik} = \frac{\log\left(1 + \exp\left(T_\text{LRC} - \left|(\hat{\mathbf{D}}_R)_{ik} - (\mathbf{D}_\text{Rgt})_{ik}\right|\right)\right)}{\log(1 + \exp(T_\text{LRC}))} \tag{25}$$

where $\psi = 10$ is the normal loss weight, $\mathbf{D}_\text{Rgt}$ is the ground truth of the right disparity map, $\hat{\nabla}_L$ $\hat{\nabla}_R$ are the normal maps estimated respectively from $\hat{\mathbf{D}}_L$ $\hat{\mathbf{D}}_R$, $\nabla_L \cdot \hat{\nabla}_L$ and $\nabla_R \cdot \hat{\nabla}_R$ are the dot product between normal maps, and $\mathbf{C}_\text{Lgt}$ $\mathbf{C}_\text{Rgt}$ are the confidence ground truth. The final supervision loss $\mathcal{L}$ is the sum of all previous partial losses:

$$\mathcal{L} = \mathcal{L}_A + \mathcal{L}_B + \mathcal{L}_C + \mathcal{L}_D + \mathcal{L}_E + \mathcal{L}_F + \mathcal{L}_G \tag{26}$$

8. Additional Ablation Studies

In this section, we report additional studies concerning the performance of Stereo Anywhere.

8.1. Generalization to Different VFMs

In the main paper, we assumed Depth Anything v2 [121] as the VFM fueling Stereo Anywhere, since it is the latest state-of-the-art model being published at the time of this submission. However, any VFM for monocular depth estimation would be suitable for this purpose, either current or future ones. To confirm this argument, we conducted some experiments by replacing Depth Anything v2 with other VFMs that appeared on arXiv in the last months, yet that are not been officially published. Among them, we select DepthPro [6], MoGe [109] and Lotus [33].

Table 5 shows the results achieved by Stereo Anywhere variants – different VFMs on Booster and LayeredFlow. We can appreciate how the different flavors of Stereo Anywhere always outperform the baseline stereo model [55]. In general, Depth Anything v2 remains the best choice to deal with non-Lambertian surfaces, with DepthPro allowing for small improvements on some metrics over the LayeredFlow dataset.

Model	Booster (Q)					LayeredFlow (E)			
	Error Rate (%)				Avg.	Error Rate (%)			Avg.
	> 2	> 4	> 6	> 8	(px)	> 1	> 3	> 5	(px)
Baseline [55]	17.84	13.06	10.76	9.24	3.59	89.21	79.02	71.61	19.27
Stereo Anywhere – DAv2 [121]	9.01	5.40	4.12	3.34	1.21	81.83	57.66	45.12	11.20
Stereo Anywhere – DepthPro [6]	10.53	7.02	5.79	5.13	2.40	78.76	61.11	51.04	14.43
Stereo Anywhere – MoGe [109]	9.47	5.77	4.49	3.84	1.44	84.27	68.67	58.89	16.22
Stereo Anywhere – Lotus [33]	12.44	8.71	7.58	6.98	3.21	86.04	62.75	49.47	13.98

Table 5. **Non-Lambertian Generalization of Stereo Anywhere w.r.t VFMs.** We measure the impact of different monocular depth estimation networks. Networks trained on SceneFlow [61].

Table 6 shows the results achieved by the different VFMs on the zero-shot generalization benchmark. Also in this case, we can appreciate how any Stereo Anywhere variant yields comparable accuracy, with some VFMs like Moge yielding some improvements over Depth Anything v2 on ETH3D, KITTI 2012 and 2015 at the expense of lowering the accuracy on Middlebury 2014 and 2021. Interestingly, we can observe an important drop in accuracy by using DepthPro on Middlebury 2021, due to several failures by the model itself on the scenes of this dataset.

Model	Middlebury 2014 (H)				Middlebury 2021				ETH3D				KITTI 2012				KITTI 2015			
	bad > 2			Avg.	bad > 2			Avg.	bad > 1			Avg.	bad > 3			Avg.	bad > 3			Avg.
	All	Noc	Occ	(px)	All	Noc	Occ	(px)	All	Noc	Occ	(px)	All	Noc	Occ	(px)	All	Noc	Occ	(px)
Baseline [55]	11.15	8.06	29.06	1.55	12.05	9.38	37.89	1.81	2.59	2.24	8.78	0.25	4.80	4.23	29.21	0.89	5.44	5.21	14.09	1.16
Stereo Anywhere – DAv2 [121]	6.96	4.75	20.34	0.94	7.97	5.71	29.52	1.08	1.66	1.43	5.29	0.24	3.90	3.52	21.65	0.83	3.93	3.79	11.01	0.97
Stereo Anywhere – DepthPro [6]	6.58	4.32	20.05	0.99	15.13	12.52	41.16	8.97	2.74	2.54	6.09	0.44	3.13	2.25	18.25	0.75	3.79	3.10	10.53	0.95
Stereo Anywhere – MoGe [109]	7.79	5.23	22.86	1.21	9.86	7.30	33.48	1.28	1.28	1.09	3.78	0.21	2.85	2.00	17.40	0.73	3.22	2.57	8.97	0.89
Stereo Anywhere – Lotus [33]	7.35	4.96	21.71	1.07	9.62	7.01	34.92	1.29	2.68	2.44	6.04	0.31	4.54	3.58	22.71	0.92	3.88	3.21	10.36	0.95

Table 6. **Generalization of Stereo Anywhere w.r.t VFMs.** We measure the impact of different monocular depth estimation networks. Networks trained on SceneFlow [61].

Finally, Figure 7 shows qualitative results obtained by the different variants of Stereo Anywhere, highlighting only minor differences among the different predictions.

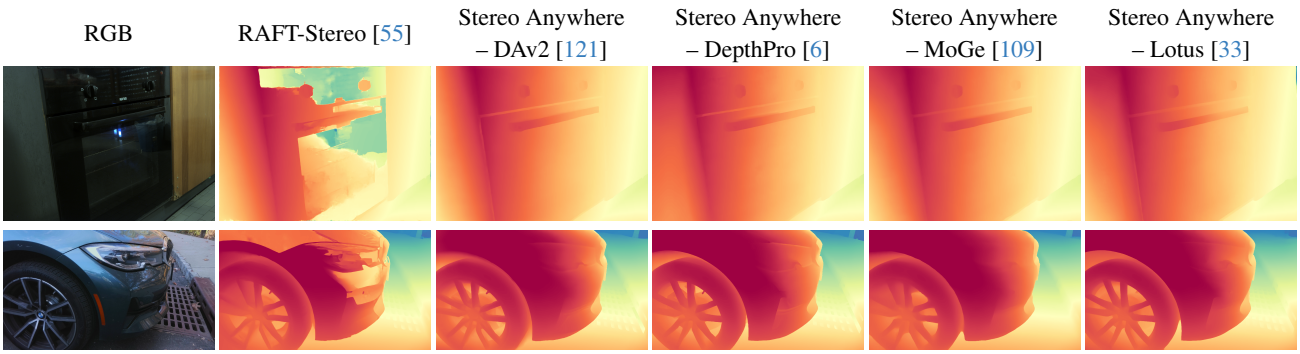

Figure 7. **Qualitative Results – Booster and LayeredFlow.** Predictions by RAFT-Stereo and Stereo Anywhere – different VFMs.

8.2. Impact of Cost Volume Truncation

Cost volume truncation is a specific augmentation we apply to improve the results in the presence of mirrors. Figure 8 shows a qualitative example of predictions by Stereo Anywhere (using Depth Anything v2) obtained by either not applying or by applying such augmentation. While Stereo Anywhere alone cannot entirely restore the surface of the mirror starting from the priors provided by the VFM, applying cost volume truncation allows for predicting a much smoother and consistent surface.

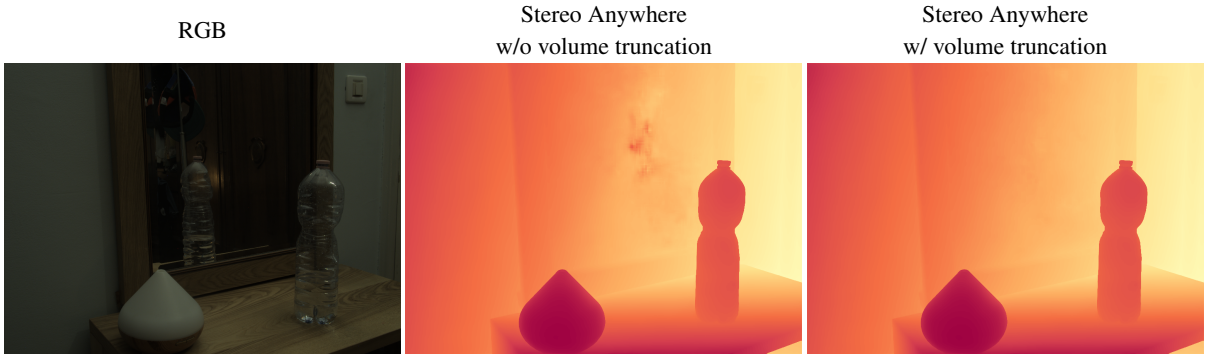

Figure 8. **Qualitative Results – Volume Truncation.** Predictions by Stereo Anywhere.

8.3. Runtime & Memory Consumption Analysis

Table 7 reports the processing time (in seconds) and memory consumption (in GB) required by Stereo Anywhere during inference, comparing it with the baseline stereo backbone, RAFT-Stereo. We measure the runtime on a single A100 GPU, repeating the experiment with three different input resolutions, specifically 256×256, 512×512, and 1024×1024, as well as by deploying the different VFMs studied before to fuel Stereo Anywhere – specifically, for each variant we report standalone runtime and memory usage by the VFM and the stereo backbone separately, as well as their sum.

Concerning runtime, Depth Anything v2 is the fastest among the VFMs, taking about 30ms to process a single image at any resolution, with Moge requiring more than $10\times$ the time for a single inference when processing 1Mpx images. The stereo backbone requires about 50% additional time compared to the baseline, RAFT-Stereo [55], because of the additional branch deployed to process the depth maps by the VFM.

For what concerns memory consumption, once again Depth Anything v2 is the most efficient among the VFMs, requiring as few as 2GB, with Moge sharing similar requirements. Our stereo backbone introduces additional memory consumption because of the second branch processing monocular cues: this overhead is negligible with 256 images, raising to about $2\times$ the memory required by RAFT-Stereo alone when dealing with 1Mpx images.

Image Size ($H \times W$)	Stereo Model Name	VFM Name	Processing Time (s)			Memory Consumption (GB)		
			VFM	Stereo	Total	VFM	Stereo	Total
256×256	**Stereo Anywhere (ours)**	DAv2 [121]	0.03	0.15	0.18	0.57	0.18	0.76
		DepthPro [6]	0.21	0.15	0.36	1.92	0.18	2.09
		MoGe [109]	0.38	0.15	0.52	0.38	0.19	0.57
		Lotus [33]	0.13	0.15	0.29	0.22	0.18	0.41
256×256	RAFT-Stereo [55]	-	-	0.10	0.10	-	0.17	0.17
512×512	**Stereo Anywhere (ours)**	DAv2 [121]	0.03	0.21	0.24	0.57	0.77	1.34
		DepthPro [6]	0.20	0.21	0.41	1.84	0.77	2.60
		MoGe [109]	0.38	0.21	0.59	0.38	0.78	1.17
		Lotus [33]	0.16	0.22	0.38	0.85	0.77	1.62
512×512	RAFT-Stereo [55]	-	-	0.14	0.14	-	0.66	0.66
1024×1024	**Stereo Anywhere (ours)**	DAv2 [121]	0.03	0.61	0.63	0.58	5.73	6.31
		DepthPro [6]	0.21	0.61	0.82	1.85	5.73	7.59
		MoGe [109]	0.38	0.60	0.98	0.42	5.77	6.19
		Lotus [33]	0.49	0.61	1.10	3.40	5.73	9.13
1024×1024	RAFT-Stereo [55]	-	-	0.36	0.36	-	2.63	2.63

Table 7. **Runtime & Memory Consumption Analysis.**

9. Qualitative Results

We conclude with additional qualitative results by Stereo Anywhere on the different datasets involved in our experiments.

Figure 9 shows two examples from the KITTI 2012 dataset (respectively, stereo pairs *000040* and *000068*). We can notice how any existing stereo model is unable to properly perceive the presence of transparent surfaces, as in correspondence of the windows on buildings and cars. On the contrary Stereo Anywhere, driven by the priors injected through the VFM, properly predicts the disparity corresponding to the transparent surfaces.

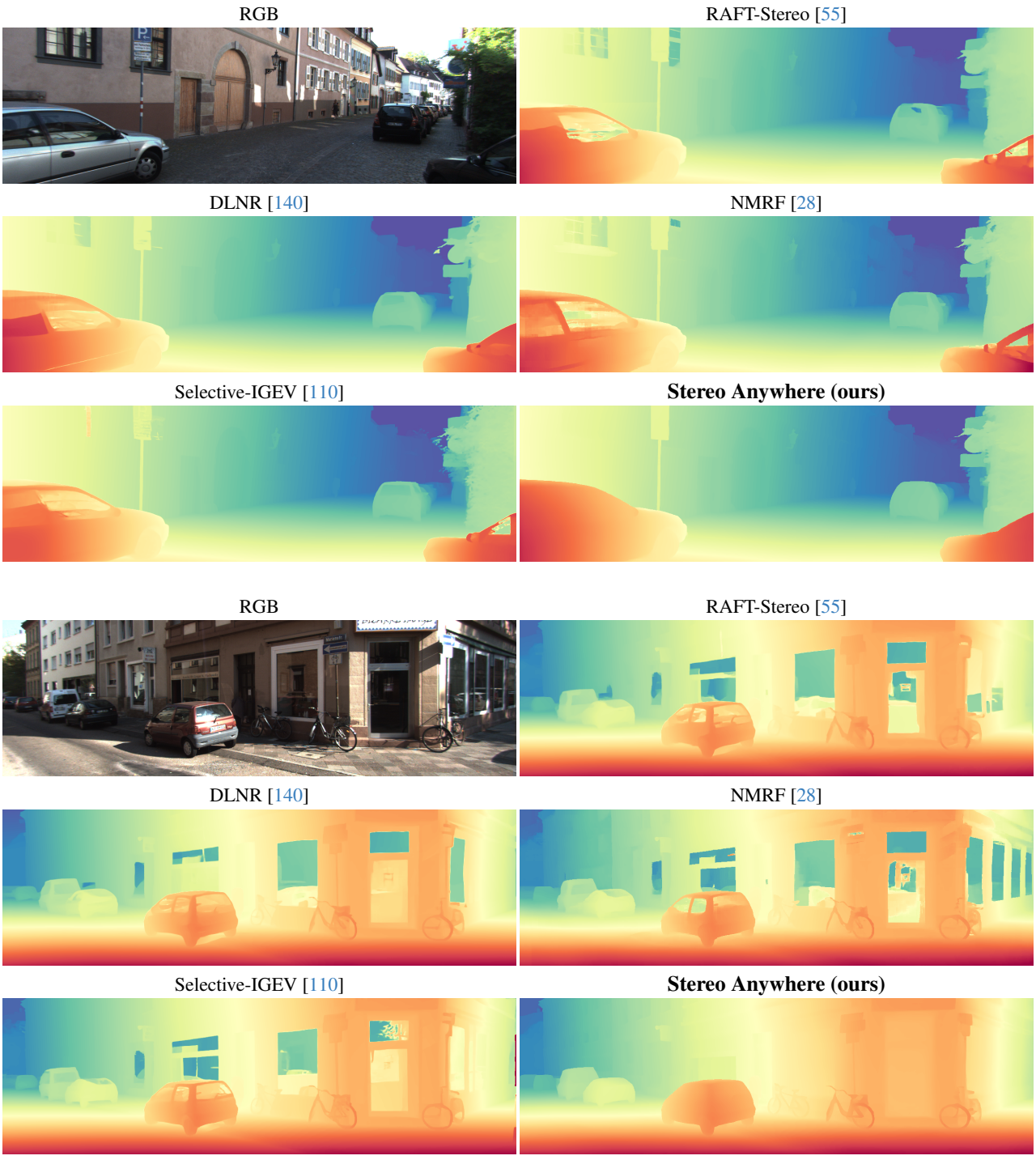

Figure 9. **Qualitative Results – KITTI 2012 (part 1).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 10 shows two further examples from KITTI 2012 (respectively, stereo pairs *000073* and *000127*). In this case, we can appreciate the much higher level of detail in the disparity maps predicted by Stereo Anywhere, with extremely thin structures in fences and gates being preserved.

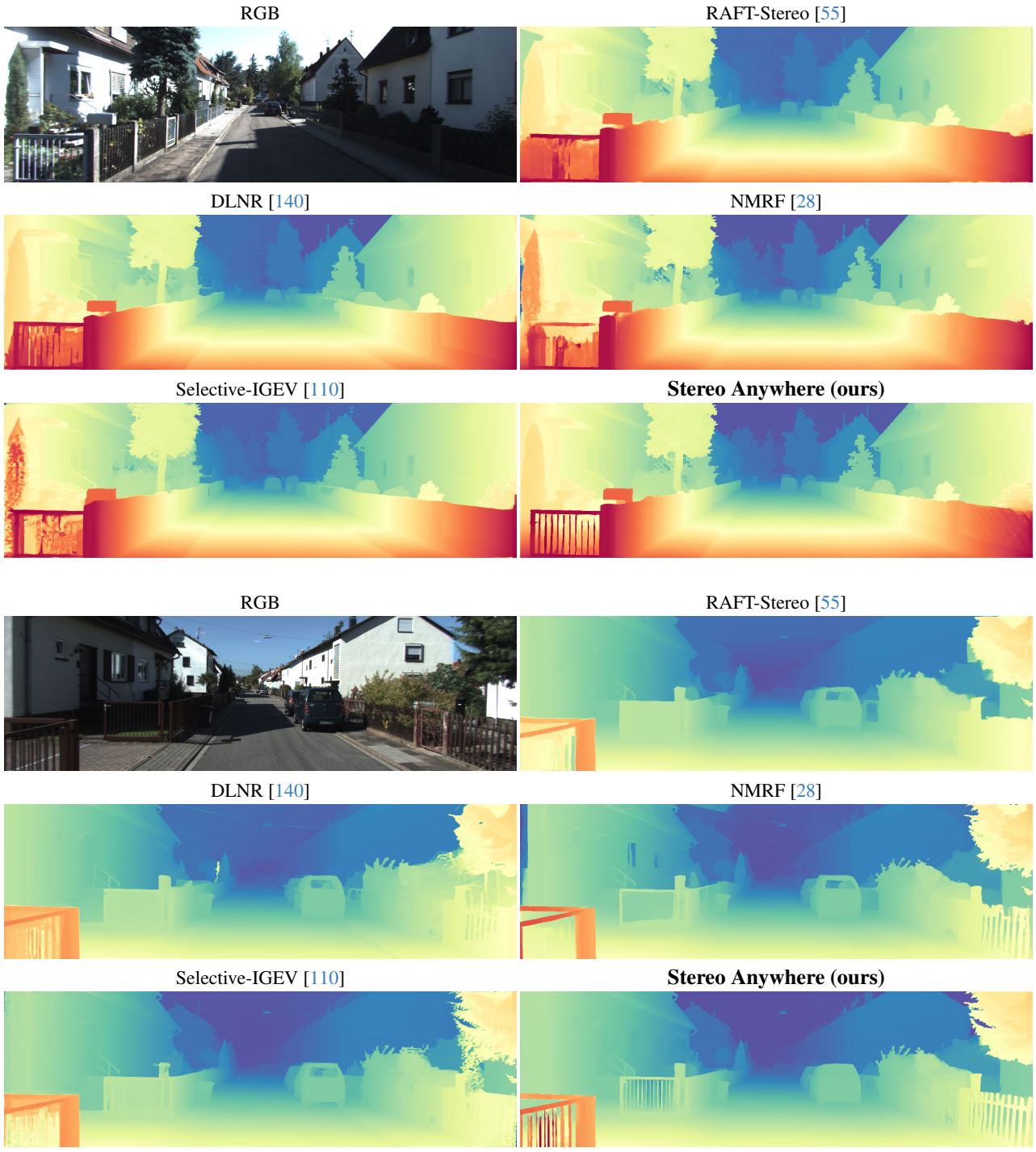

Figure 10. **Qualitative Results – KITTI 2012 (part 2).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 11 reports two stereo pairs from KITTI 2015 (respectively, *000024* and *000049*). These examples confirm the ability to recover both thin structures and transparent surfaces already appreciated in KITTI 2012.

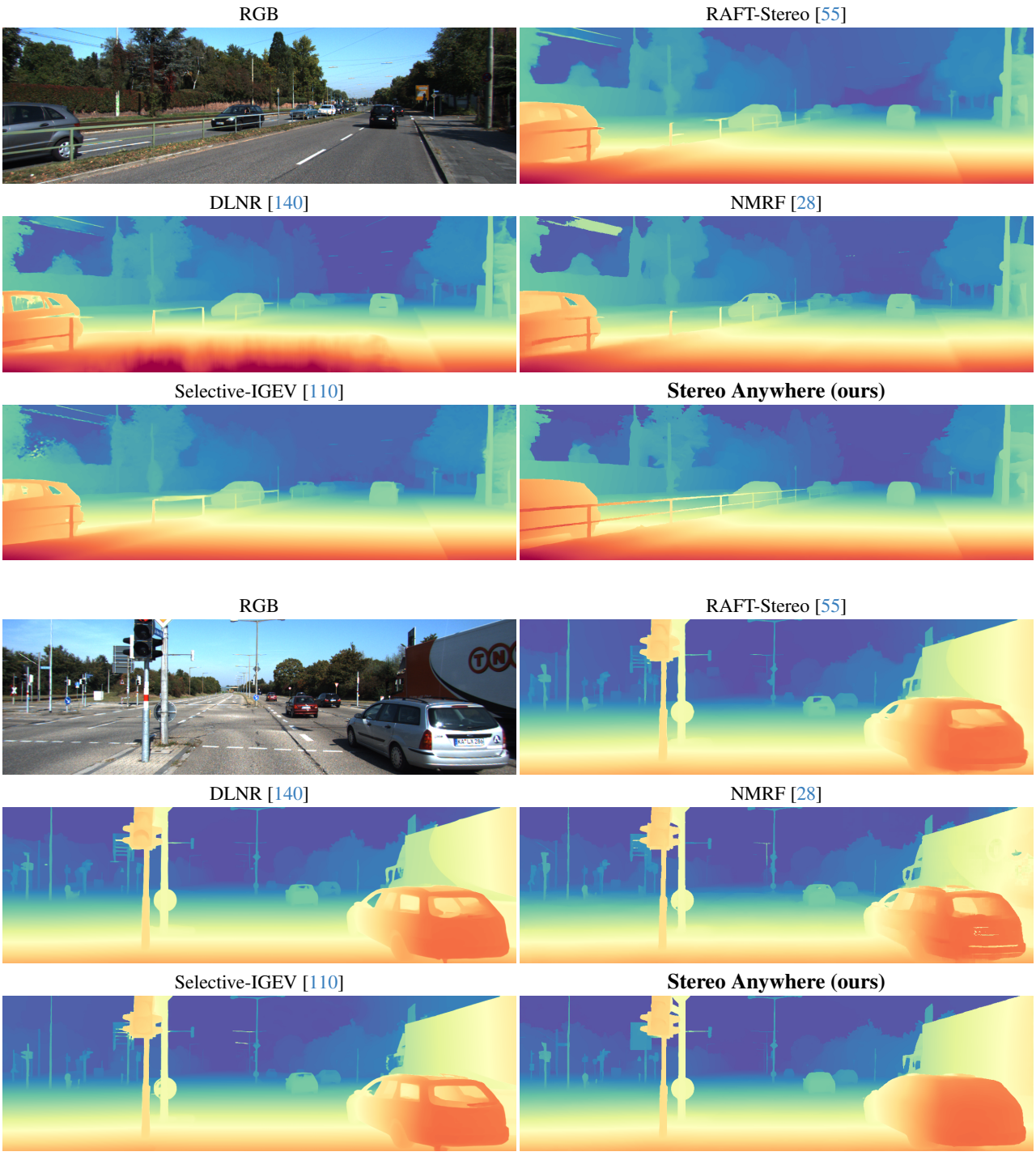

Figure 11. **Qualitative Results – KITTI 2015 (part 1).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 12 reports two additional samples from KITTI 2015 (respectively, *000093* and *000144*). These latter present both underexposed and transparent regions, respectively on the billboard and the tram in the two images. While existing stereo networks struggle at dealing with both, Stereo Anywhere exposes unprecedented robustness.

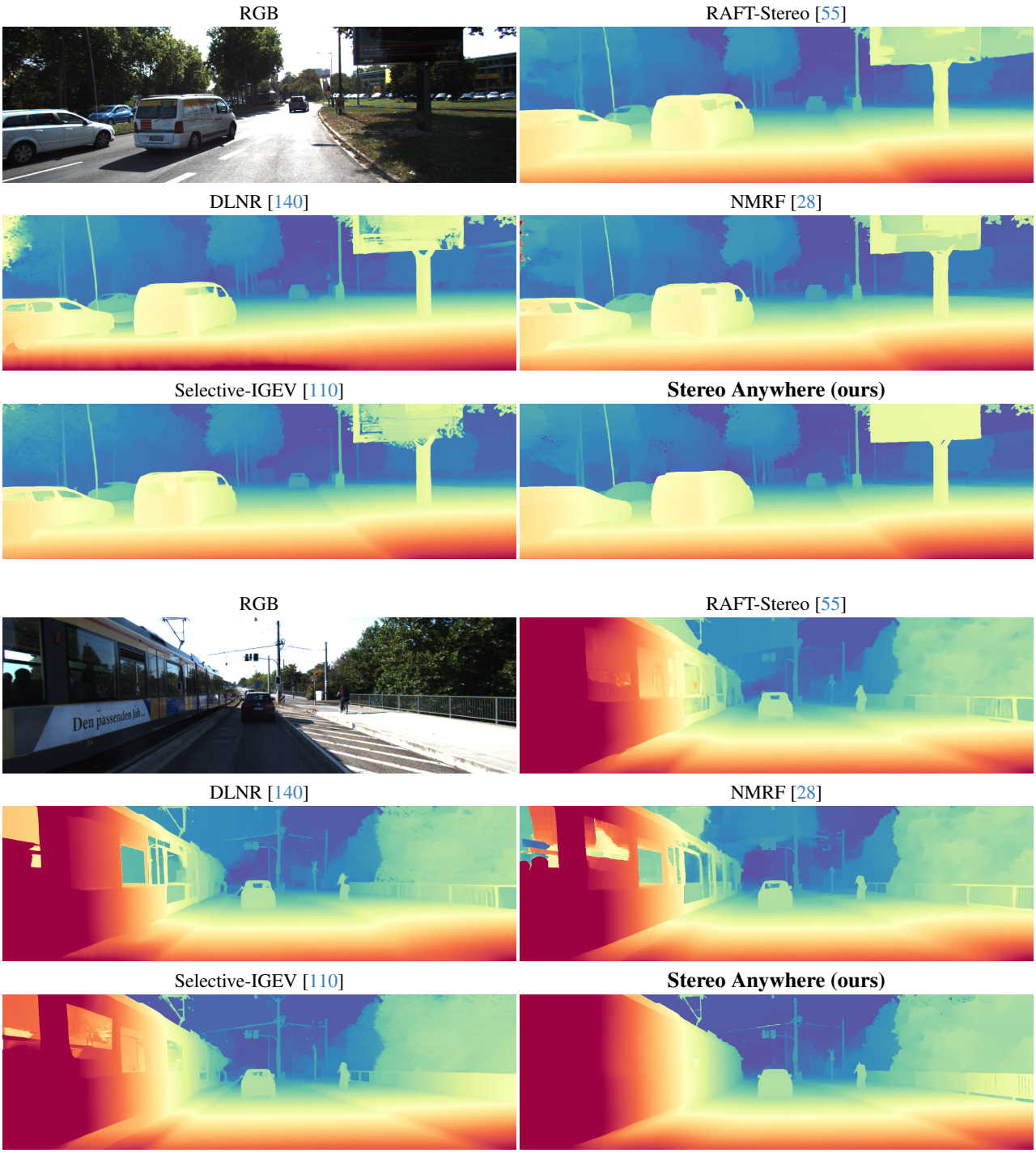

Figure 12. **Qualitative Results – KITTI 2015 (part 2).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 13 reports two image pairs from Middlebury 2014 (respectively, *Adirondack* and *Vintage*). On the former, Stereo Anywhere preserves the very thin holes on the back of the chair, while on the latter it can properly estimate the disparity for the displays, where existing methods are fooled and predict holes.

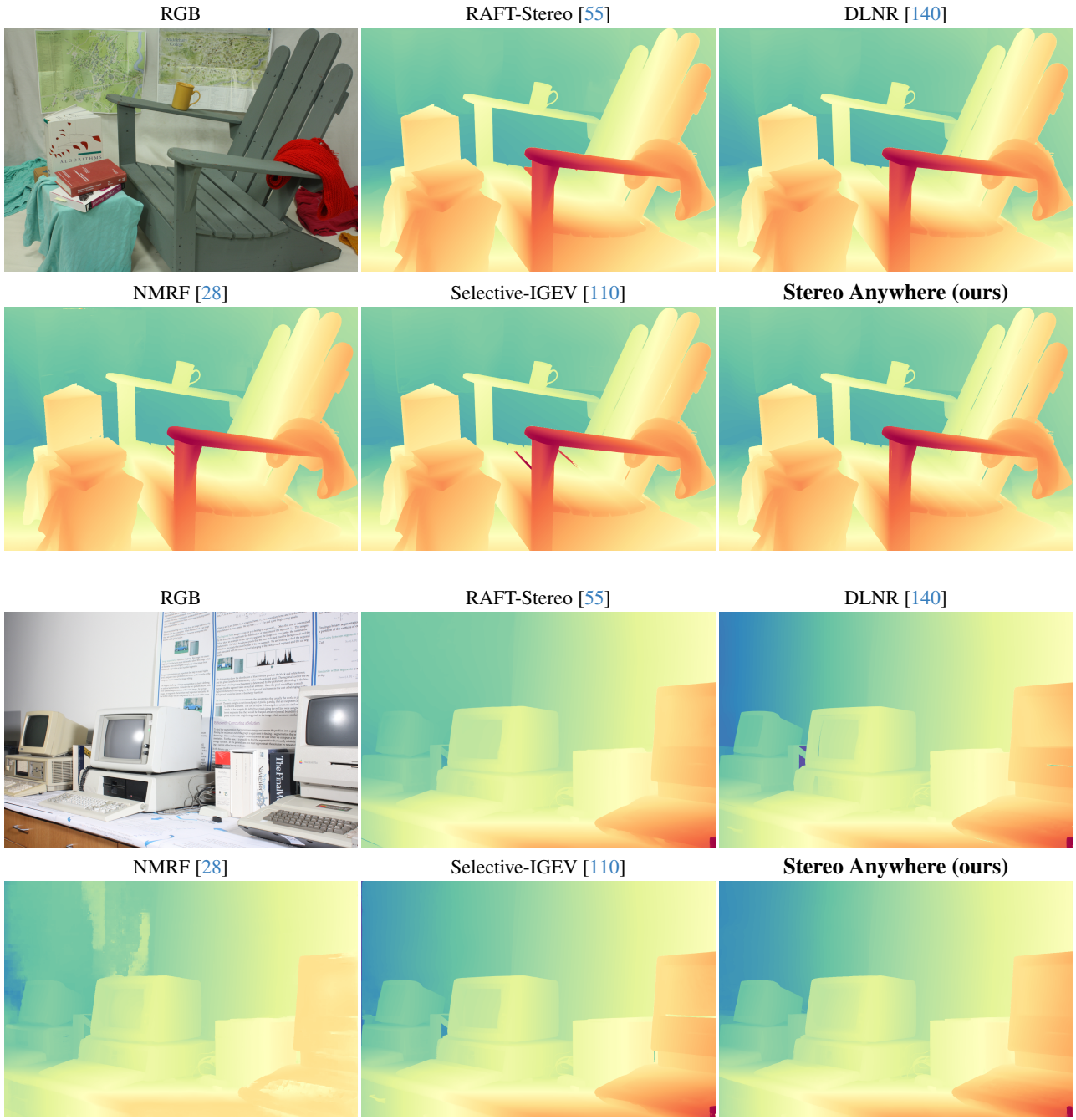

Figure 13. **Qualitative Results – Middlebury 2014.** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 14 and 15 shows the results on two samples from Middlebury 2021, peculiar for their aspect ratio (respectively, *ladder1* and *ladder2*). Although existing models perform quite well on both, they fail to preserve the skittles on the top of the scene, whereas Stereo Anywhere properly predicts their structure.

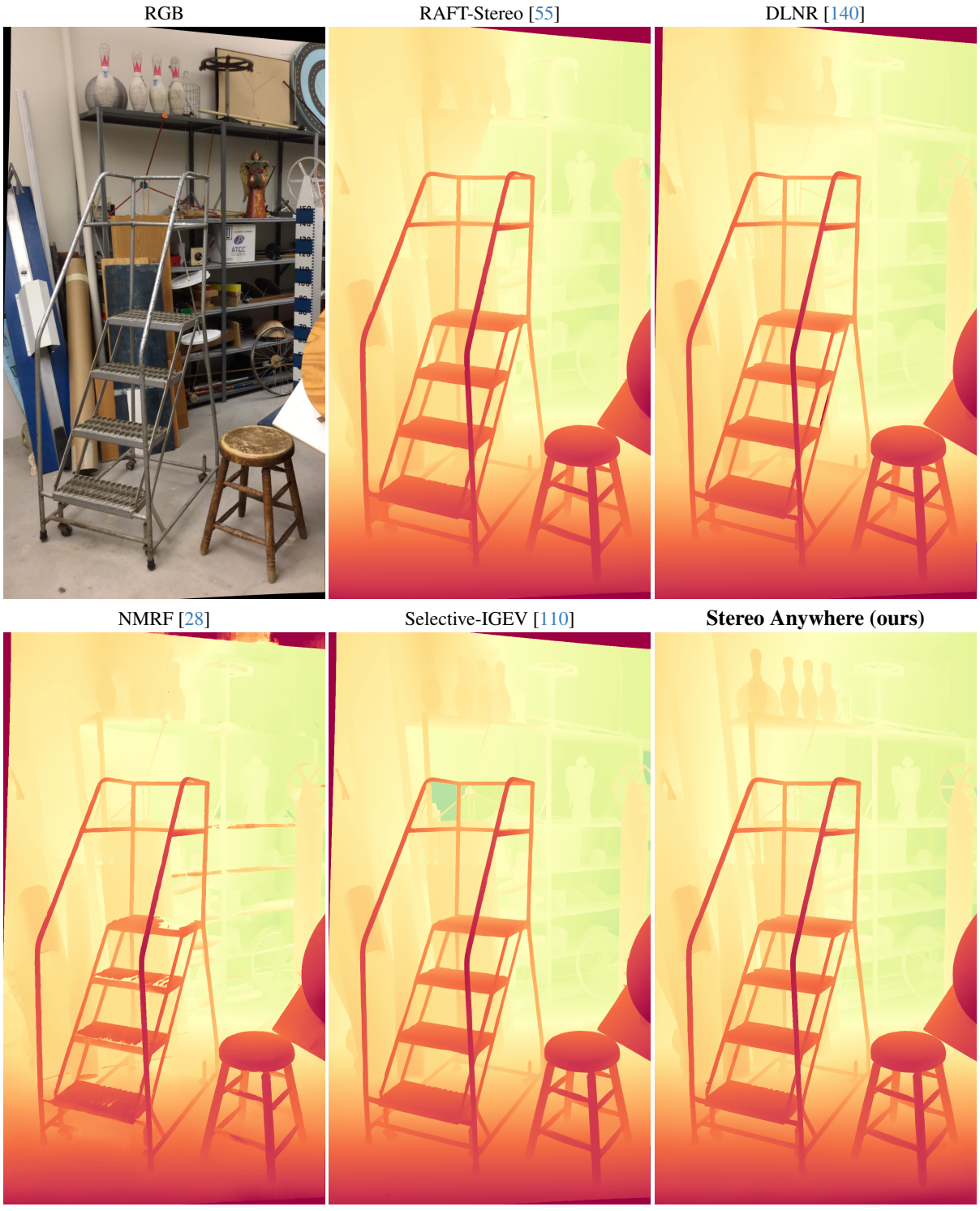

Figure 14. **Qualitative Results – Middlebury 2021 (part 1).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 15. **Qualitative Results – Middlebury 2021 (part 2).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 16 collects three outdoor images from ETH3D (respectively, *Playground1*, *Playground2* and *Playground3*). Once again, Stereo Anywhere proves its supremacy at predicting fine details such as branches and poles, while resulting more robust to challenging illumination conditions such as the sun flare in *Playground2*.

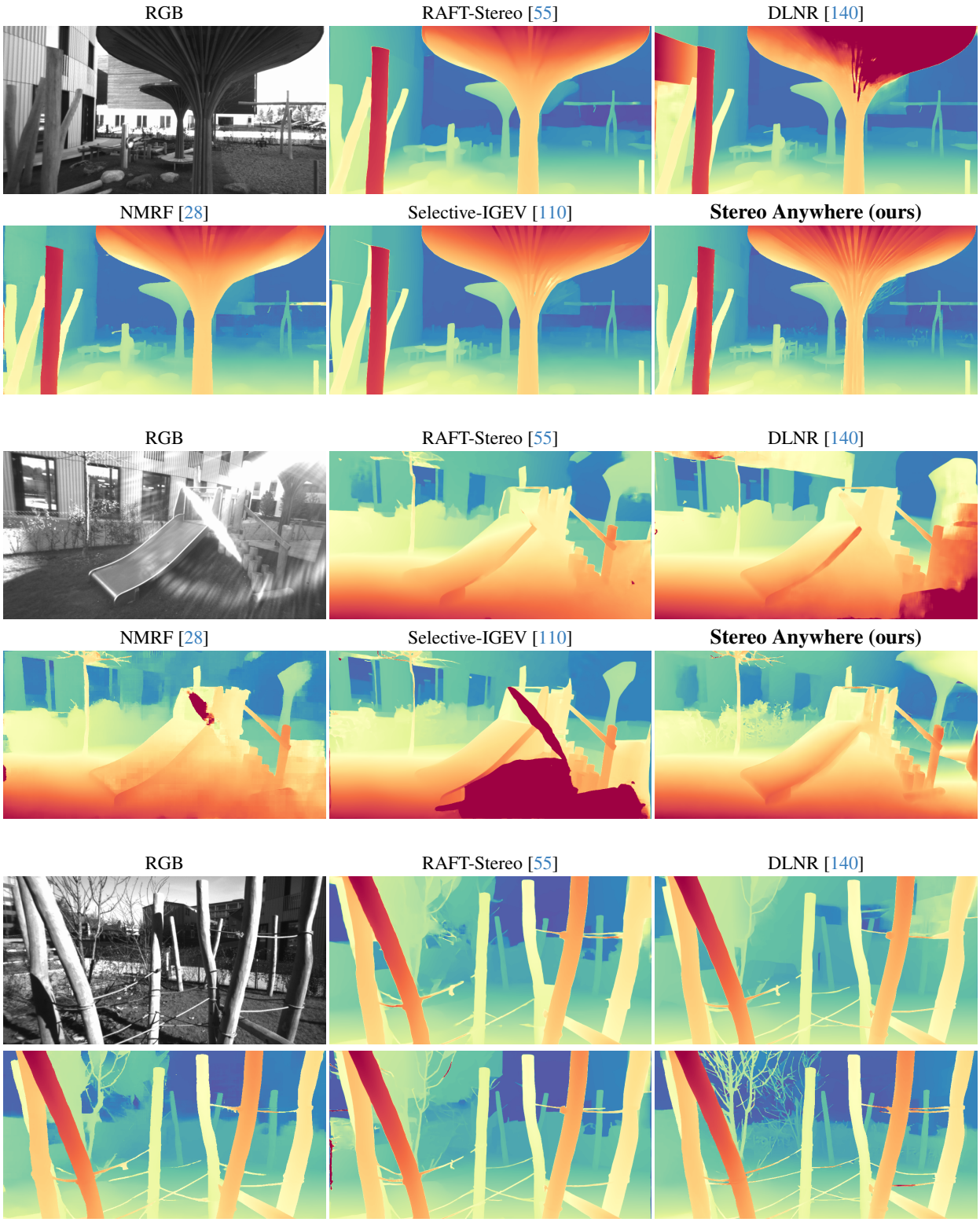

Figure 16. **Qualitative Results – ETH3D.** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 17 and 18 report four examples from the Booster dataset, confirming how Stereo Anywhere can exploit the strong priors provided by the VFM to properly perceive the glass surface on the window in the former image, as well as challenging, untextured black surfaces of the computer, the TV and the displays appearing in the remaining samples.

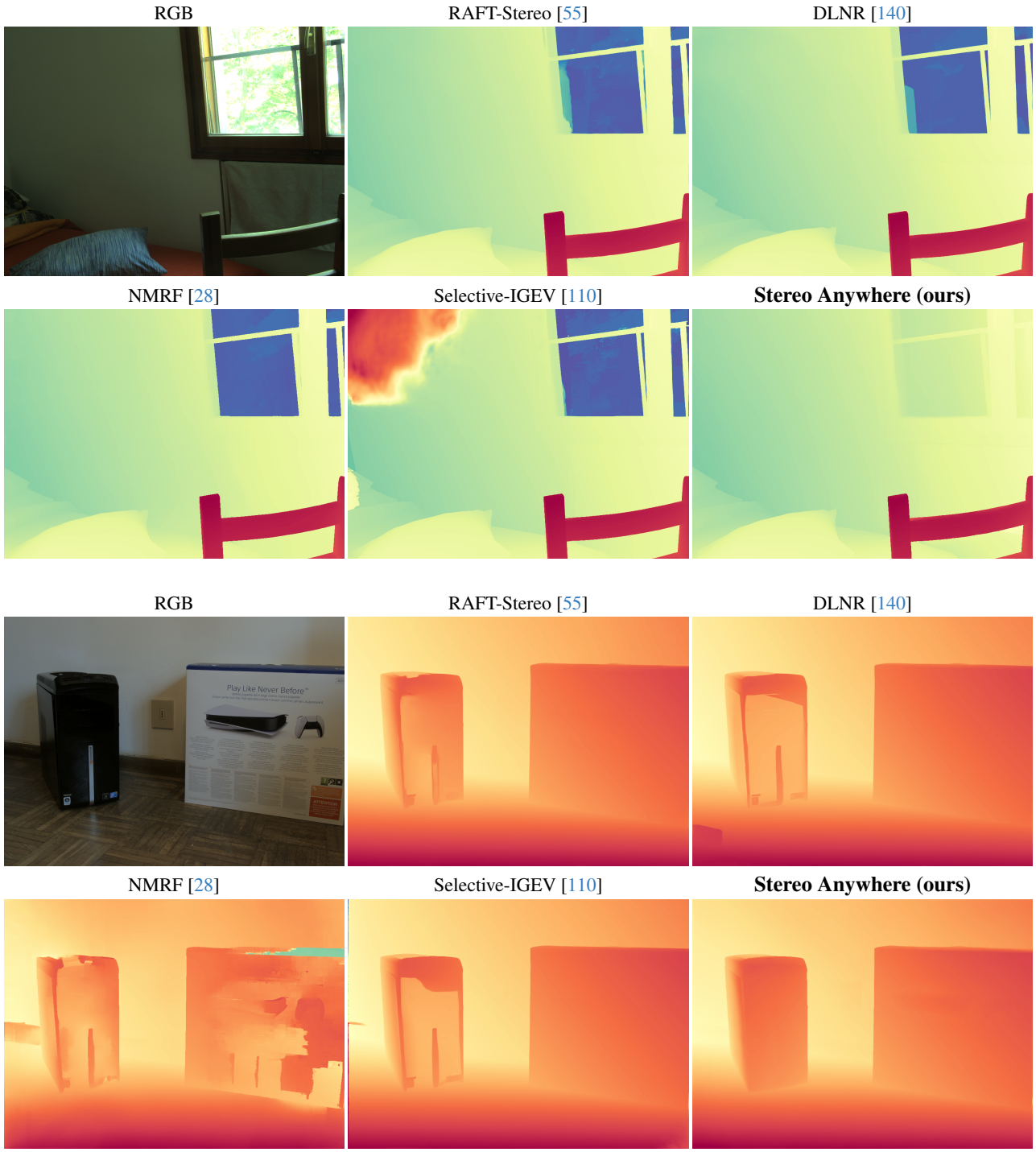

Figure 17. **Qualitative Results – Booster (part 1).** Predictions by state-of-the-art models and Stereo Anywhere.

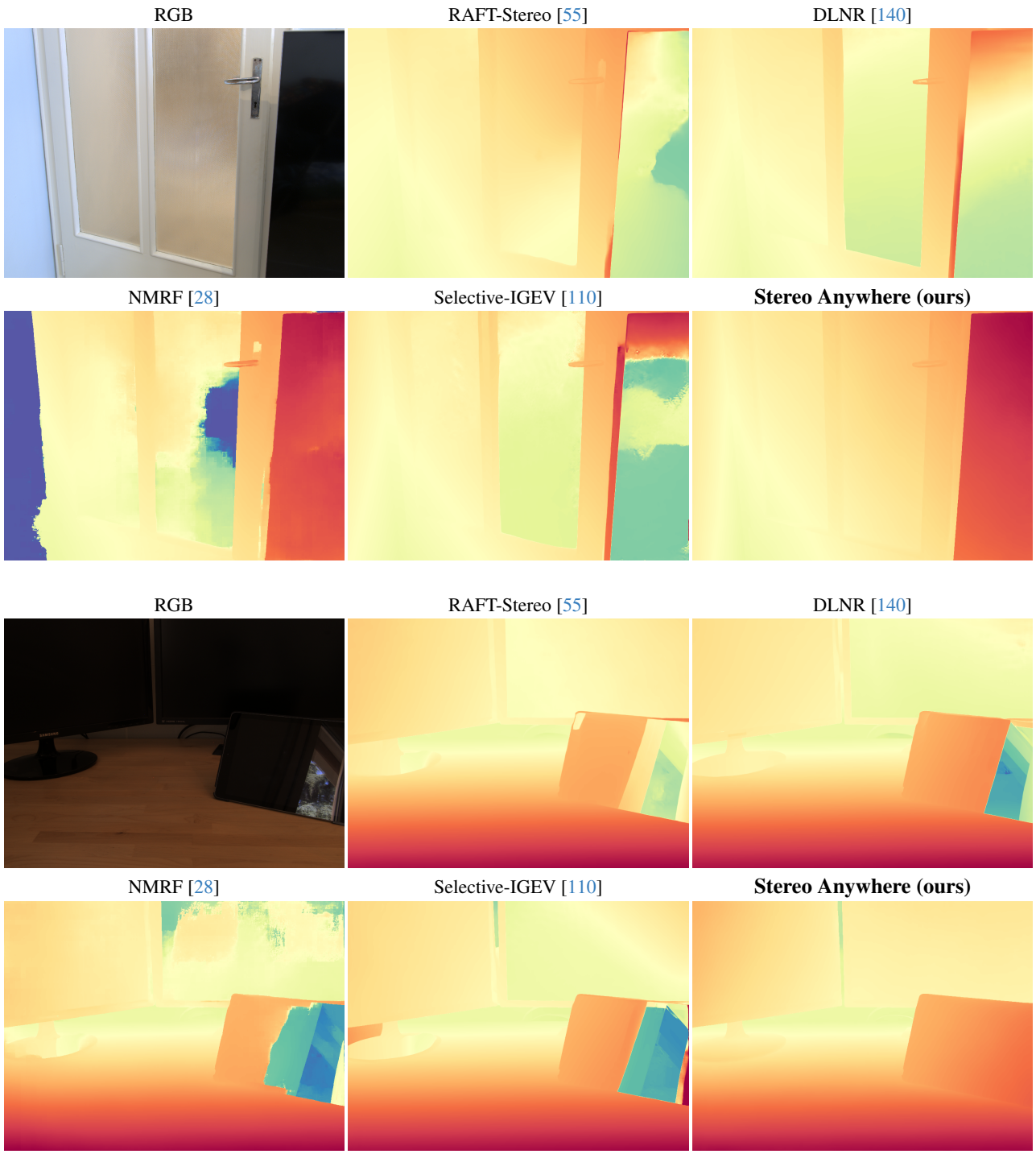

Figure 18. **Qualitative Results – Booster (part 2).** Predictions by state-of-the-art models and Stereo Anywhere.

Figure 19 showcases three images from the LayeredFlow dataset, highlighting once again the inability of the state-of-the-art networks to model even small, transparent surfaces as those in the doors from the first and second samples, conversely to Stereo Anywhere which can properly identify their presence. Finally, the third sample further highlights the high level of detail in Stereo Anywhere predictions once again.

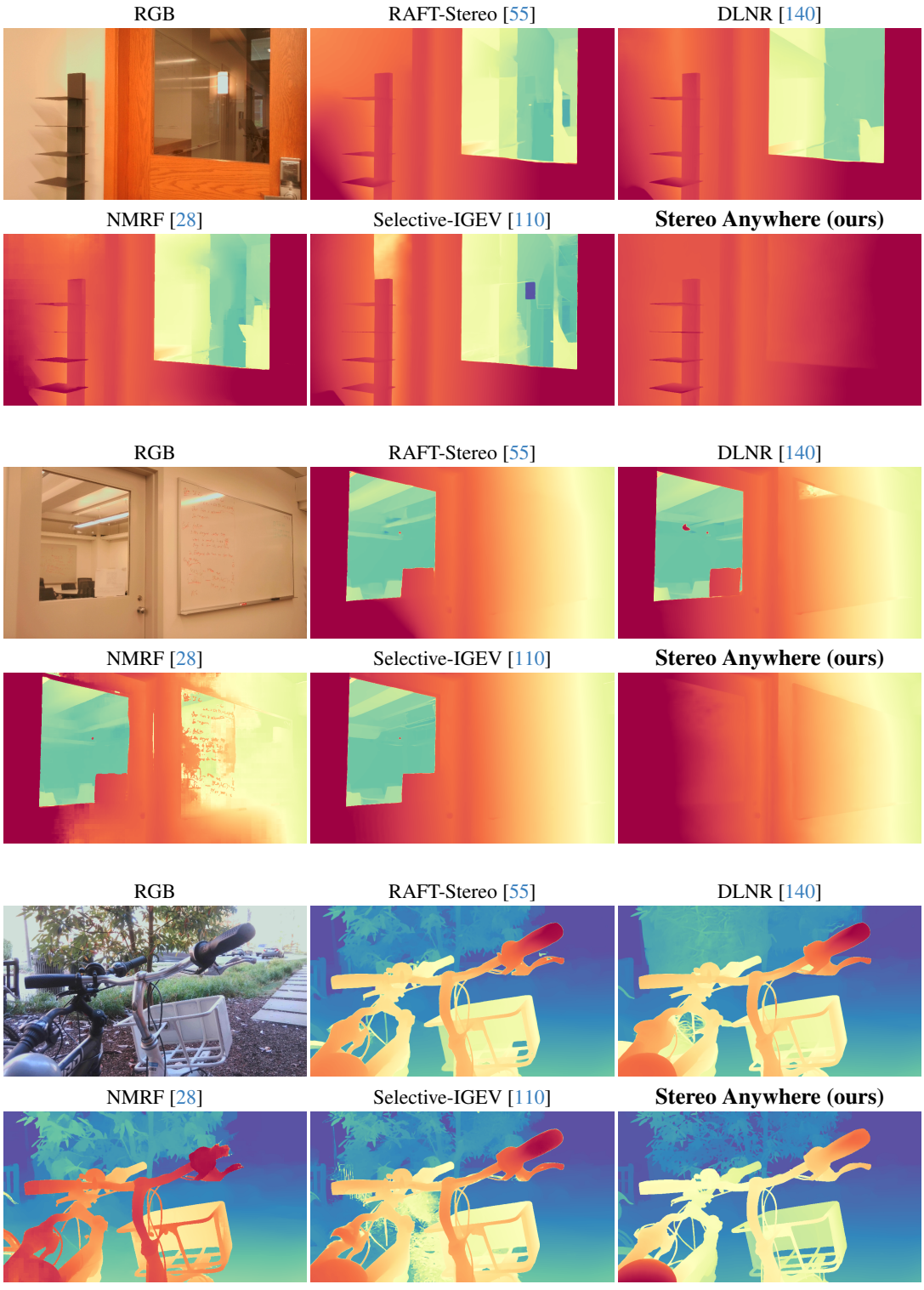

Figure 19. **Qualitative Results – LayeredFlow.** Predictions by state-of-the-art models and Stereo Anywhere.

To conclude, Figure 20 collects three scenes from our novel MonoTrap dataset. In this case, we report predictions by both state-of-the-art monocular and stereo models, as well as by Stereo Anywhere. The perspective illusions fooling monocular methods, unsurprisingly, do not affect stereo networks, which however are inaccurate near the left border of the image (first sample) or in the absence of texture (second sample). Stereo Anywhere effectively combines the strength of both worlds, while being not affected by any of their weaknesses.

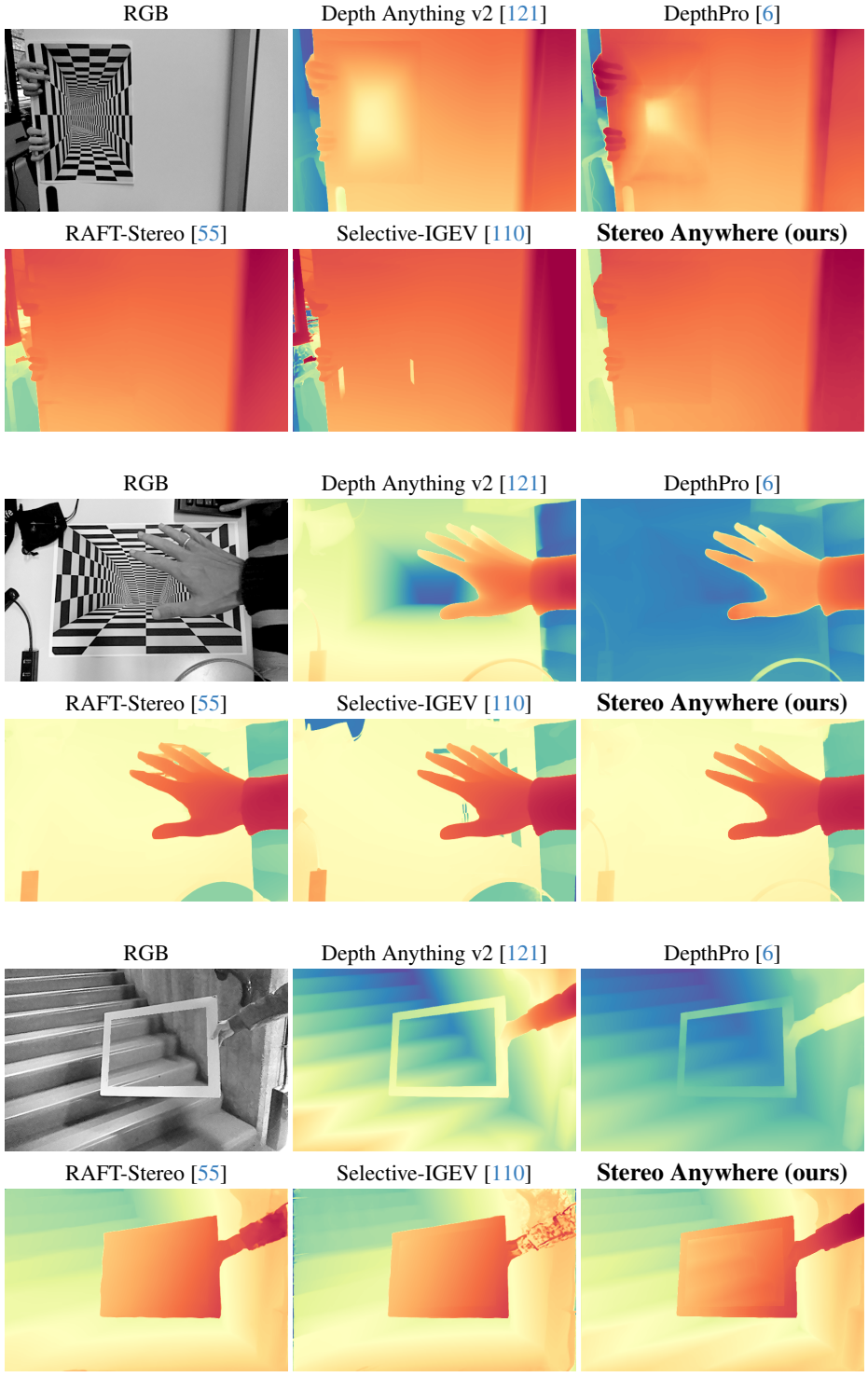

Figure 20. **Qualitative Results – MonoTrap.** Predictions by state-of-the-art models and Stereo Anywhere.