Frequency-Aware Self-Supervised Monocular Depth Estimation

Xingyu Chen[1] Thomas H. Li[1,2,3] Ruonan Zhang[1] Ge Li ✉[1]

[1]School of Electronic and Computer Engineering, Peking University [2]Advanced Institute of Information Technology, Peking University
[3]Information Technology R&D Innovation Center of Peking University

cxy@stu.pku.edu.cn tli@aiit.org.cn zhangrn@stu.pku.edu.cn geli@ece.pku.edu.cn

https://github.com/xingyuuchen/freq-aware-depth

Abstract

We present two versatile methods to generally enhance self-supervised monocular depth estimation (MDE) models. The high generalizability of our methods is achieved by solving the fundamental and ubiquitous problems in photometric loss function. In particular, from the perspective of spatial frequency, we first propose Ambiguity-Masking to suppress the incorrect supervision under photometric loss at specific object boundaries, the cause of which could be traced to pixel-level ambiguity. Second, we present a novel frequency-adaptive Gaussian low-pass filter, designed to robustify the photometric loss in high-frequency regions. We are the first to propose blurring images to improve depth estimators with an interpretable analysis. Both modules are lightweight, adding no parameters and no need to manually change the network structures. Experiments show that our methods provide performance boosts to a large number of existing models, including those who claimed state-of-the-art, while introducing no extra inference computation at all.

1. Introduction

Inferring the depth of each pixel in a single RGB image is a versatile tool for various fields, such as robot navigation [14], autonomous driving [30, 37] and augmented reality [24]. However, it is extremely difficult to obtain a large number of depth labels from real world, and even expensive Lidar sensors can only obtain depth information of sparse points on the image [34]. Therefore, a large number of self-supervised MDE researches have been conducted, with accuracy getting closer and closer to supervised methods. By exploiting the geometry projection constrain, the self-supervision comes from image reconstructions, requiring only known (or estimated) camera poses between different viewpoints. Though significant progress has been made, there still remains some undiscovered general problems.

First. Many works [25, 18, 13, 35, 39] concentrated on predicting clearer (sharper) depth of object boundaries. Despite their success, they mainly relied on well-designed network architectures. In this work, we show a more fundamental reason for this limitation - from input images. An interesting observation in Fig. 1b raises the question: *Does the photometric loss at the object boundaries really indicates inaccurate depth predictions?* Self-supervised training minimizes the per-pixel photometric loss based on the 2D-3D-2D reprojection [38, 12]. Every single pixel is expected to attach to one deterministic object, otherwise the depth of a *mixed* object is of no physical meaning. The pixel-level ambiguity (Fig. 1c), as it happens, manifests as making the object boundary the fused color of two different objects. These ambiguous pixels belong to no objects in the 2D-3D back-projection (see point cloud in Fig. 1d), and have no correspondence when evaluating photometric loss (on the target and synthesized images) after 3D-2D reprojection. As a result, the network always learns irrational loss from them, regardless of its predicted depths.

Second. Intuitively, for a loss function, predictions close to *gt* should have small loss, whereas predictions with large error ought to deserve harsh penalties (loss). However, photometric loss does not obey this rule in high-freq regions, as shown in Fig. 2. In such regions, a tiny deviation from *gt* receives a harsh penalty, while a large error probably has an even smaller loss than *gt*. These *unfairness* comes from high spatial frequency and the breaking of photometric consistency assumption, respectively. To reduce such *unfairness*, we present a frequency-adaptive Gaussian blur technique called Auto-Blur. It enlarges the receptive field by *radiating* photometric information of pixels when needed.

To sum up, our contributions are threefold:

1. We show the depth network suffers from irrational supervision under the photometric loss at specific boundary areas. We trace its cause to pixel-level ambiguity due to the anti-aliasing technique. Furthermore, we demonstrate the photometric loss cannot *fairly* and accurately evaluate the depth predictions in high-frequency regions.

2. To overcome these two problems, we first propose **Ambiguity-Masking** to exclude the ambiguous pixels producing irrational supervisions. Second, we present **Auto-Blur**, which pioneeringly proves blurring images could universally enhance depth estimators by reducing *unfairness* and enlarging receptive fields.

3. Our methods are highly versatile and lightweight, providing performance boosts to a large number of existing models, including those claiming SoTA, while introducing no extra inference computation at all.

Despite our superior results, the key motivation of this paper is to shed light on the problems rarely noticed by previous MDE researchers, and wish our analysis and solutions could inspire more subsequent works.

2. Related Work

2.1. Supervised Depth Estimation

Plenty recent researches have proved that deep neural networks bring remarkable improvements to MDE models. Many MDE (or stereo matching [26, 33]) methods are fully supervised, requiring the depth labels collected from RGB-D cameras or Lidar sensors. Eigen *et al.* [5] introduced a multi-scale architecture to learn coarse depth and then refined on another network. Fu *et al.* [7] changed depth regression to classification of discrete depth values. [2] further extended this idea to adaptively adjust depth bins for each input image. With direct access to depth labels, loss is formulated using the distance between predicted depth and ground truth depth (Scale-Invariant loss [21, 2], $\mathcal{L}_1$ distance [19, 33]), without relying on assumptions such as photometric consistency or static scenes. [1, 29] also computed $\mathcal{L}_1$ loss between the gradient map of predicted and *gt* depth.

2.2. Self-Supervised Depth Estimation

Self-supervised MDE transforms depth regression into image reconstruction [9, 38]. Monodepth [11] introduced the left-right consistency to alleviate depth map discontinuity. Monodepth2 [12] proposed to use *min.* reprojection loss to deal with occlusions, and auto-masking to alleviate moving objects and static cameras. In order to produce sharper depth edges, [18] leveraged the off-the-shelf fine-grained sematic segmentations, [35] designed an attention-based network to capture detailed textures. In terms of image gradient, self-supervised methods [8, 12, 27, 32] usually adopt the disparity smoothness loss [16]. [20] trained an additional 'local network' to predict depth gradients of small image patches, and then integrated them with depths from 'global network'. [22] computed photometric loss on the gradient map to deal with sudden brightness change, but it is not robust to objects with different colors but the same gradients. Most related to our Auto-Blur is Depth-Hints [32], which helped the network escape from local minima of thin structures, by using the depth proxy labels obtained from SGM stereo matching [17], while we make no use of any additional supervision and are not restricted to stereo datasets.

3. The Need to Consider Spatial Frequency

This section mainly describes our motivation, specifically, revealing two problems that few previous works noticed. We begin with a quick review of the universally used photometric loss in self-supervised MDE (Sec. 3.1), then we demonstrate from two aspects (Sec. 3.2 and Sec. 3.3) that the photometric loss function is not a good supervisor for guiding MDE models in some particular pixels or areas.

3.1. Appearance Based Reprojection Loss

In self-supervised MDE setting, the network predicts a dense depth image D_t given an input RGB image I_t at test time. To evaluate D_t, based on the geometry projection constraint, we generate the reconstructed image $\tilde{I}_{t+n}$ by sampling from the source images I_{t+n} taken from different viewpoints of the same scene. The loss is based on the pixel-level appearance distance between I_t and $\tilde{I}_{t+n}$. Majorities of self-supervised MDE methods [11, 27, 12, 38, 25, 36, 23] adopt the $\mathcal{L}_1 + \mathcal{L}_{ssim}$ [31] as photometric loss:

$$\mathcal{L}(I_t, \tilde{I}_{t+n}) = \frac{\alpha}{2}(1-SSIM(I_t, \tilde{I}_{t+n}))+(1-\alpha) \parallel I_t-\tilde{I}_{t+n} \parallel_1, \tag{1}$$

where $\alpha = 0.85$ by default and $SSIM$ [31] computes pixel similarity over a 3×3 window.

3.2. Does Loss at Object Boundary Make Sense?

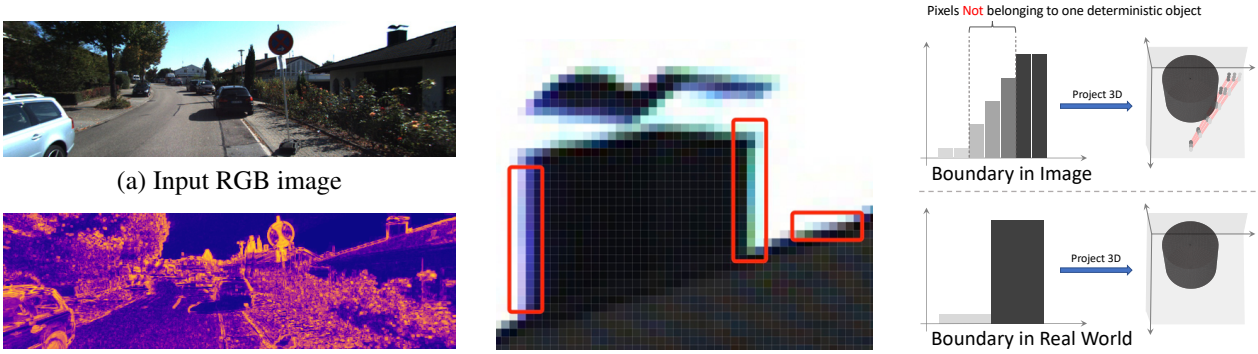
As seen from Fig. 1, when training gets to the middle part, the losses appear in two types of regions:

1. On the whole object (true-positives). Because the estimation of the object's depth (or camera motion) is inaccurate, it reprojects to another object;

2. At the object boundaries (false-positives). Such as the black chimney in the upper right corner.

So why does some loss only appear at the object boundaries, and is it reasonable? In fact, few works analyzed its cause. In order to minimize the *per-pixel* reprojection error, the network adjusts *every single pixel's* depth to make it reproject to where it is in the source view. This process works under the condition that each pixel belongs to *one* deterministic object, since we can never use *one* depth value to characterize a pixel that represents *two* different objects. However, we illustrate in Fig. 1c&d that, the anti-aliasing breaks this training condition by making the object boundary color the weighted sum of both sides' colors.

Specifically, in self-supervised MDE, pixels are first (2D-3D) back-projected to construct the 3D scene using



(a) Input RGB image

(b) Loss map from $8/20^{th}$ epoch

(c) Image patch

(d) Boundary comparison

Figure 1. **(b)** On most objects, losses appear at object boundaries. **(c)** The pixels at the boundaries are gradually changed over the junction. However, these colors are ambiguous, *i.e.*, neither from the black chimney nor the white clouds. **(d)** Object boundaries in the real world are completely mutated, where one single pixel characterizes one deterministic object. However, the ambiguous pixels each contain photometric information for two objects, whereas the network predicts at most one single depth value for them. When projecting the black chimney to 3D point clouds, the ambiguous pixels detach from their main body both spatially and photometrically, regardless of the predicted depths. Hence, no pixels in the synthesized view would match them, resulting in always-large reprojection losses.

the predicted depths, and then (3D-2D) reprojected to another viewpoint to synthesize the new image. In the 2D-3D phase, the ambiguous pixels detach from their main body, the 3D points make no physical sense as they do not represent any particular objects, neither spatially nor photometrically (Fig. 1d). After the 3D-2D phase, no correspondence from the target image could match these ambiguous colors in the synthesized image, producing large photometric loss. However, loss should only exist in the area where the depth prediction is incorrect, and these pixels produce unreasonable loss which should not be learnt by the network.

3.3. Photometric Loss is Unfair in High-Freq Area

Before delving into the problem in Fig. 2, we define what is an *absolutely fair loss function* and the *fairness degree*.

Definition 1 (Absolutely Fair Loss Function) *Given a loss function $\mathcal{L}$ for network ψ with ground truth gt, for $\forall x_1, x_2 \in [x_{min}, x_{max}]$, if $|x_1 - gt| < |x_2 - gt|$, then $\mathcal{L}(x_1) < \mathcal{L}(x_2)$. We call $\mathcal{L}$ an absolutely fair loss function, with fairness degree = 1 as defined below.*

Definition 2 (Fairness Degree) *Given a loss function $\mathcal{L}$ for network ψ with ground truth gt, we compute its fairness degree in the range of $[x_{min}, x_{max}]$ subject to:*

$$\mathcal{D}_{fair}(\mathcal{L}, x_{min}, x_{max}) = \frac{\int_{x_{min}}^{x_{max}} \epsilon\left(\frac{\partial \mathcal{L}(x)}{\partial x}(x - gt)\right) dx}{x_{max} - x_{min}}, \tag{2}$$

where $\epsilon(\cdot)$ is the indicator function such that $\epsilon(x) = 1$ if $x > 0$, otherwise $\epsilon(x) = 0$.

To better illustrate the problem, we first look at the loss function in one of the *supervised* MDE methods [19]:

$$\mathcal{L}_{supervised} = \frac{1}{N}\sum_{n=1}^{N} \left\| d_n - \hat{d_n} \right\|_1, \tag{3}$$

where it averages $\mathcal{L}_1$ distances between the predicted depth d_n and ground truth depth $\hat{d_n}$ over all N pixels. This is an *absolutely fair loss* (Definition 1) since the network penalty is (positive) proportional to the prediction error, which always guides the network to converge towards ground truth depth. In contrast, there are two serious problems in the $\mathcal{L}_1 + \mathcal{L}_{ssim}$ photometric loss as shown in Fig. 2:

1. *A small depth estimation error leads to a large loss.* In other words, compared with ground truth, a very slight deviation can produce a large reprojection error, which harshly penalizes the network when its prediction is almost near ground truth;

2. *A large depth estimation error probably produce an even smaller loss than gt.* Due to the repeated textures in these areas, it is common to mistakenly reproject to another location with the same appearance. That is, there are too many local optimums and even false global optimum, interfering with training.

At this point, we could see that being *fair* is a basic requirement for a loss function, otherwise any neural network would be misguided. Unlike Depth-Hints [32] who used additional proxy-labels to help predictions of thin structures escape from local optimum, we focus on improving the *fairness degree* of the loss function itself.

Augmented with the proposed Auto-Blur module, we can achieve a significant improvement in the ill relationship between network penalty and prediction error. Quantitatively, the *fairness degree*, *i.e.* $\mathcal{D}_{fair}(\mathcal{L}_1 + \mathcal{L}_{ssim}, 0, 15)$,

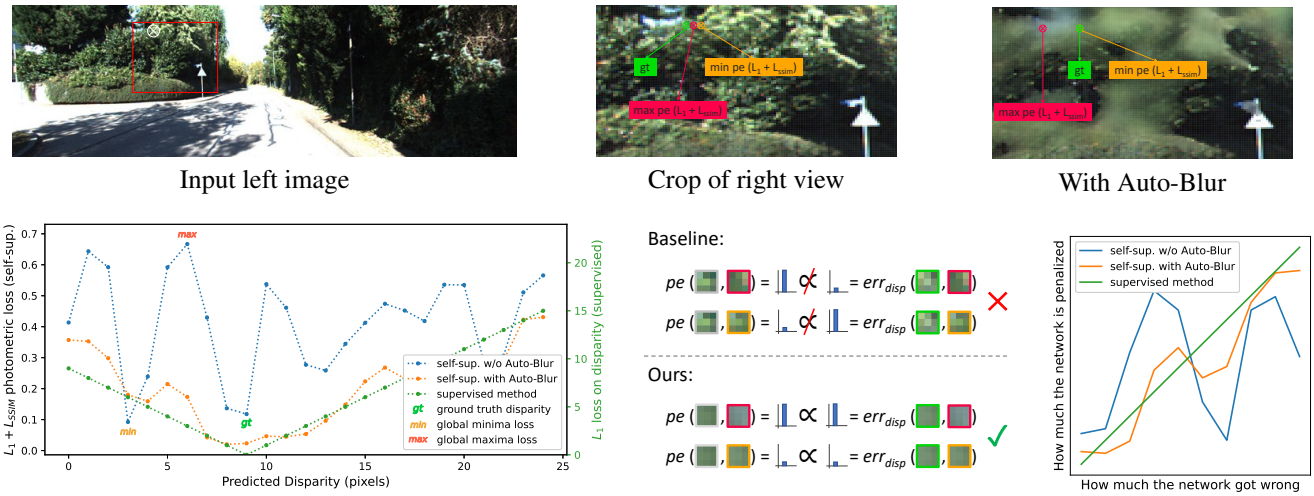


Figure 2. **Top:** A training image and its crop of the right view (stretched) with and without the proposed Auto-Blur. **Bottom:** Left is the quantitative photometric loss used in self-supervised method with/without Auto-Blur and $\mathcal{L}_1$ loss on predicted disparity (Eq. 3) used in supervised method. The middle plot ($\propto$: proportional to) shows without Auto-Blur, disparity of *max* $\mathcal{L}_1 + \mathcal{L}_{ssim}$ photometric loss is instead more accurate than that of *min* photometric loss; the photometric loss of ground truth is even larger than incorrect disparity, while self-supervised method augmented with Auto-Blur does not suffer from this misjudging. Plot on the right[2] is the qualitative analysis of the relationship between network penalty and prediction error. Supervised method exhibits the *absolutely fair* relationship. With Auto-Blur, $\mathcal{L}_1 + \mathcal{L}_{ssim}$ becomes more stable and gets closer to supervised one.

increases from $\frac{8}{15}$ to $\frac{13}{15}$. Qualitatively, $\mathcal{L}_1+\mathcal{L}_{ssim}$ no longer suffers from the false global optimum, and looks more like a 'V' curve (as the supervised method exhibits on the bottom left of Fig. 2) than before - this indicates a clearer positive proportional relationship, reducing the probability of getting stuck in the local minima. In the coming section, we show how our Auto-Blur module can relieve this problem without any semantic information.

4. Methodology

4.1. Self-supervised Monocular Depth Estimation

Following [38, 12], given a monocular and/or stereo video, we first train a depth network ψ_{depth} consuming a single target image I_t as input, and outputs its pixel-aligned depth map $D_t = \psi_{depth}(I_t)$. Then, except for stereo pairs whose relative camera poses are fixed, we train a pose network ψ_{pose} taking temporally adjacent frames as input, and outputs the relative camera pose $T_{t\to t+n} = \psi_{pose}(I_t, I_{t+n})$.

Suppose we have access to the camera intrinsics K, along with D_t and $T_{t\to t+n}$, we project I_t into I_{t+n} to compute the sampler $\otimes$:

$$\otimes = proj\left(D_t, T_{t\to t+n}, K\right). \tag{4}$$

The sampler $\otimes \in \mathbb{R}^{H\times W\times 2}$ (H, W represents height and width), which says 'for each pixel in I_t, where is the corresponding pixel in I_{t+n}?'. We generate the reconstructed image by sampling from I_{t+n} subject to $\otimes$:

$$\tilde{I}_{t+n} = \left\langle I_{t+n}, \otimes \right\rangle, \tag{5}$$

where $\langle\cdot,\cdot\rangle$ is the differentiable bilinear sampling operator according to [12]. The final loss functions consist of photometric loss measured by $\mathcal{L}_1 + \mathcal{L}_{ssim}$ as Eq. 1 and edge-aware smoothness loss [16].

4.2. Compute Spatial Frequency

We first calculate spatial frequencies. Following [16], for each pixel (*e.g.*, pixel at i, j), we compute differences between its adjacent pixels to represent the gradient. Specifically, we use $\mathcal{L}_2$ norm of horizontal and vertical differences:

$$\nabla_{u\pm}(i,j) = I(i,j) - I(i\pm 1, j), \tag{6}$$

$$\nabla_{v\pm}(i,j) = I(i,j) - I(i, j\pm 1), \tag{7}$$

$$\nabla_{\pm}(i,j) = \left\|\nabla_{u\pm}(i,j), \nabla_{v\pm}(i,j)\right\|_2, \tag{8}$$

$$\nabla(i,j) = \left\| \frac{\nabla_{u+}(i,j) - \nabla_{u-}(i,j)}{2}, \frac{\nabla_{v+}(i,j) - \nabla_{v-}(i,j)}{2} \right\|_2. \tag{9}$$

These spatial frequencies allow the following methods to identify their target pixels or regions. In practice, we adopt ∇_+ (Eq. 8) in Auto-Blur for simplicity; while ∇ (Eq. 9) in Ambiguity-Masking for accuracy.

4.3. Ambiguity-Masking

Extract Ambiguity in an Input Image. Given an input image I_t, we aim to exclude the pixels with ambiguous colors described in Sec. 3.2, *i.e.*, forming the ambiguity map $\mathcal{A}_t$.

[2]Data is from disparity 0∼9 and their losses in the bottom left plot.

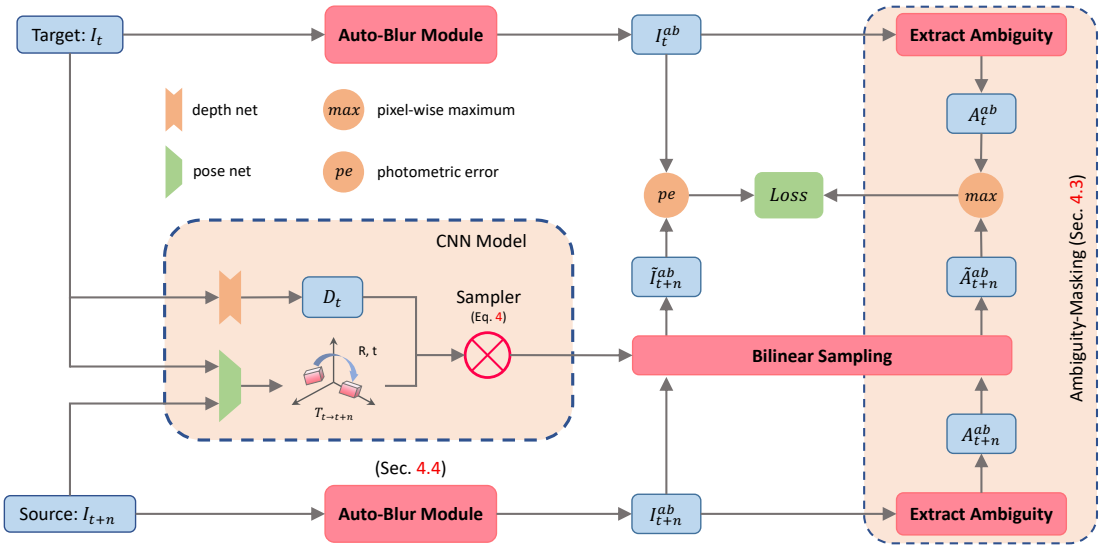


Figure 3. Overview of the proposed method. We propose two approaches to alleviate problems demonstrated in Sec. 3.2 and 3.3, respectively, namely **Ambiguity-Masking** and **Auto-Blur**. Both are highly versatile, *i.e.* orthogonal to the CNN model architectures. The input images are 'auto-blurred' adaptively, then input to the photometric loss function to increase its *fairness degree*. The Amb.-Masking extracts ambiguities both in the target and reconstructed image, eliminating irrational supervisions. Details in Sec. 4.3 and 4.4.

The larger the color difference between the adjacent objects, the more ambiguous the pixels located in the object junction. Hence, we first compute the frequency map $\mathcal{F}_t$ as Eq. 9. Since these pixels are used to smooth the abrupt color changes at object boundary, their colors must be weighted sum of both sides' pixels (see pixels that gradually change from white to black on the sloping roof in Fig. 1c). Accordingly, we form a binary mask μ to pick the high-frequency pixels whose gradients in opposite directions have the opposite sign, either horizontally or vertically, *i.e.*,

$$\mu = \left[\nabla_{u+} \cdot \nabla_{u-} < 0 \bigvee \nabla_{v+} \cdot \nabla_{v-} < 0 \right], \tag{10}$$

where $[\cdot]$ is the Iverson bracket. Then, the initial ambiguity map $\mathcal{A}_t$ for an input I_t is computed as:

$$\mathcal{A}_t = \mu \odot \mathcal{F}_t, \tag{11}$$

where $\odot$ denotes element-wise multiplication.

Synthesize Ambiguities into a Mask. Notably, because photometric loss is based on two images, both target image I_t and reconstructed image $\tilde{I}_{t+n}$ can cause the loss to be untrustworthy. Thus, we also take $\tilde{I}_{t+n}$ into consideration.

Following [38, 12], for each I_{t+n}, we compute the sampler $\otimes_{t+n}$ using D_t, $T_{t\to t+n}$ and K subject to Eq. 4. Note that $\otimes_{t+n}$ not only contains pixel corresponding relationship used to generate the reconstructed image $\tilde{I}_{t+n}$, but also contains information of how the ambiguities of I_{t+n} affect $\tilde{I}_{t+n}$. In light of this, we bilinearly sample $\mathcal{A}_{t+n}$ to get which pixels in $\tilde{I}_{t+n}$ are from the ambiguous pixels in I_{t+n} according to $\otimes_{t+n}$:

$$\tilde{\mathcal{A}}_{t+n} = \left\langle \mathcal{A}_{t+n}, \otimes_{t+n} \right\rangle. \tag{12}$$

Then, we take the pixel-wise maximum of ambiguities in reconstructed image and target image (intuitively, a *logical or* operation):

$$\mathcal{A}_t^{max} = max\left\{\mathcal{A}_t, \tilde{\mathcal{A}}_{t+n}\right\}, \tag{13}$$

because for each pixel in $\mathcal{L}(I_t, \tilde{I}_{t+n})$, the ambiguity from either target image or reconstructed image can both cause its photometric loss to be untrustworthy. The final ambiguity mask $\mathcal{A}_t^{pe}$ is defined by:

$$\mathcal{A}_t^{pe} = \left[\, \mathcal{A}_t^{max} < \delta \,\right], \tag{14}$$

which is to be element-wise multiplied with $\mathcal{L}(I_t, \tilde{I}_{t+n})$. The pseudo-code of the overall algorithm is in Supp.

4.4. Auto-Blur

In order to improve the ill relationship between network penalty and prediction error in high-freq area, we propose Auto-Blur, an adaptive Gaussian low-pass filter in essence. To be clear, the 'auto-blurred' images are only input to the loss function but not to the network, since it is only the photometric loss being unfair and the CNN model expects as much texture as original images to predict accurate depths.

Identify Pixels in High-Frequency Area. For simplicity, in Auto-Blur we just adopt ∇_+ (Eq. 8) as the frequency map $\mathcal{F}_t$ for input image I_t. We first determine whether pixel location p is of high spatial frequency:

$$\mathcal{M}_{is\text{-}hf\text{-}pixel}(p) = \left[\, \mathcal{F}_t(p) > \lambda \,\right], \tag{15}$$

where λ is the pre-defined threshold.

Next, we apply average pooling to $\mathcal{M}_{is\text{-}hf\text{-}pixel}$ with stride set to 1, and an Iverson bracket again:

$$\mathcal{M}^{avg}_{is\text{-}hf\text{-}pixel}(p) = \frac{1}{s \times s} \sum_{q \in N_{s\times s}(p)} \mathcal{M}_{is\text{-}hf\text{-}pixel}(q), \tag{16}$$

$$\mathcal{M}_{in\text{-}hf\text{-}area}(p) = \left[\mathcal{M}^{avg}_{is\text{-}hf\text{-}pixel}(p) > \eta\%\right], \tag{17}$$

where s is the average pooling kernel size, pixel q belongs to the $s \times s$ neighbors $N_{s\times s}(p)$ of p. Intuitively, if more than $\eta\%$ of the pixels in $N_{s\times s}(p)$ are high-frequency pixels, then p is located in a high-frequency area.

Note that instead of naively averaging $\mathcal{F}_t(q)$ in $N_{s\times s}(p)$ directly, we average the Boolean elements obtained by thresholding $\mathcal{F}_t(q)$ using λ. This operation avoids misjudging *thin* object boundaries as high-frequency regions - they are just high-freq pixels themselves, but not in a high-freq *region* that is filled with high-freq pixels.

Blurring Strategy. Based on the Gaussian blurred image I^{gb}_t of I_t:

$$I^{gb}_t(p) = \sum_{q \in N(p)} w^{gb}(q) I_t(q), \tag{18}$$

$$w^{gb}(q) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\triangle x^2 + \triangle y^2}{2\sigma^2}}, \tag{19}$$

where pixel q belongs to the neighbors $N(p)$ of p, $w^{gb}(q)$ is the weight defined by Gaussian kernel, we compute the final auto-blurred image I^{ab}_t subject to:

$$I^{ab}_t(p) = w_{blur}(p) I^{gb}_t(p) + (1 - w_{blur}(p)) I_t(p), \tag{20}$$

$$w_{blur}(p) = \mathcal{M}^{avg}_{is\text{-}hf\text{-}pixel}(p) \mathcal{M}_{in\text{-}hf\text{-}area}(p), \tag{21}$$

where we let I^{ab}_t be a weighted sum of I^{gb}_t and I_t, and the more high-frequency pixels around p, the more blurry p is. On the other hand, pixels not in high-frequency areas remain unchanged.

Blurry images are often thought to degrade the performance of vision systems, instead they benefit the photometric loss, as later analyzed in Sec. 5.4.

5. Experiments

5.1. Implementation Details

We set $\delta = 0.3$ to extract ambiguous pixels. We also make use of negative exponential function as an alternative to Eq. 14, *i.e.* $\mathcal{A}^{pe}_t = e^{-\gamma \mathcal{A}^{max}_t}$, where $\gamma = 3$. In Auto-Blur, we set $\lambda = 0.2$ to determine whether a pixel is of high frequency, η is set to 60 and s is set to 9 so that if more than 60% of a pixel's 9×9 neighbors are of high frequency, then it is regarded as 'in high frequency region'. Our methods support plug and play, so other settings just remain exactly unchanged when embedded into a new baseline, with no more than 10% additional training time and no extra inference time at all.

5.2. Quantitative Results

Rather than simply comparing our results to previous SoTA, we run experiments on large numbers of existing models, and compare the results with and w/o our methods in each one of them. We show our methods lead to superior results, which not only proves that the newly revealed problems are general and ubiquitous, but also makes it possible for future researches to overcome these obstacles.

KITTI dataset [10] consists of calibrated stereo videos captured from a car driving on the streets in Germany. The depth evaluation is done on the Lidar point cloud, with all seven of the standard metrics [6]. We use the Eigen split of

Method	PP	Data	Extra time	AbsRel	SqRel	RMSE	RMSE log	δ_1	δ_2	δ_3
Monodepth2 no pt [12]	✗	S	-	0.130	1.144	5.485	0.232	0.831	0.932	0.968
+ Ours	✗	S	**+ 0ms**	**0.127**	**1.086**	**5.406**	**0.224**	**0.832**	**0.937**	**0.971**
Monodepth2 M [12]	✗	M	-	0.115	0.903	4.863	0.193	0.877	0.959	0.981
+ Ours	✗	M	**+ 0ms**	**0.112**	**0.834**	**4.746**	**0.189**	**0.880**	**0.961**	**0.982**
Zhou *et al.* [38]	✗	M	-	0.183	1.595	6.709	0.270	0.734	0.902	0.959
+ Ours	✗	M	**+ 0ms**	**0.142**	**1.547**	**5.433**	**0.224**	**0.840**	**0.944**	**0.974**
WaveletMonodepth [27]	✗	S	-	0.109	**0.845**	4.800	0.196	0.870	0.956	**0.980**
+ Ours	✗	S	**+ 0ms**	**0.108**	0.862	**4.786**	**0.194**	**0.875**	**0.957**	**0.980**
Monodepth2 S [12]	✗	S	-	0.109	0.873	4.960	0.209	0.864	0.948	0.975
+ Ours	✗	S	**+ 0ms**	**0.107**	**0.835**	**4.850**	**0.201**	**0.865**	**0.951**	**0.978**
FSRE-Depth [18]	✗	M	-	**0.105**	0.722	4.547	0.182	**0.886**	**0.964**	**0.984**
+ Ours	✗	M	**+ 0ms**	**0.105**	**0.711**	**4.452**	**0.181**	**0.886**	**0.964**	**0.984**
Monodepth2 MS [12]	✗	MS	-	**0.106**	0.818	4.750	0.196	0.874	0.957	0.979
+ Ours	✗	MS	**+ 0ms**	**0.106**	**0.797**	**4.672**	**0.187**	**0.887**	**0.961**	**0.982**
CADepth [35]	✗	S	-	0.107	0.849	4.885	0.204	0.869	0.951	0.976
+ Ours	✗	S	**+ 0ms**	**0.106**	**0.823**	**4.835**	**0.201**	**0.870**	**0.953**	**0.977**
Depth-Hints [32]	✗	S	-	0.109	0.845	4.800	0.196	0.870	0.956	0.980
+ Ours	✗	S	**+ 0ms**	**0.105**	**0.811**	**4.695**	**0.192**	**0.875**	**0.958**	**0.981**

Table 1. **Comparison of existing models with and without our methods on KITTI Eigen split [6].** The *Data* column specifies the training data type: S - stereo images, M - monocular video and MS - stereo video. All models are trained with 192×640 images and Resnet18 [15] as backbone. All results are not **P**ost-**P**rocessed [11]. Metrics are error metrics ↓ and accuracy metrics ↑. Models augmented with our methods achieve better scores on almost all metrics, generally. No extra inference computation is needed at all.

Method	Pre-trained	Auto-Blur	Amb.-Masking	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
Base no pt				0.132	1.044	5.142	0.210	0.845	0.948	0.977
Base	✓			0.115	0.903	4.863	0.193	0.877	0.959	0.981
Base + Auto-Blur	✓	✓		0.113	0.858	4.837	0.192	0.876	0.959	0.981
Base + Amb.-Mask	✓		✓	0.113	0.871	4.785	0.191	**0.880**	0.960	**0.982**
Our full model	✓	✓	✓	**0.112**	**0.834**	**4.746**	**0.189**	**0.880**	**0.961**	**0.982**

Table 2. **Ablation on Eigen split [6].** The baseline model with none of our contributions and without ImageNet [4] pretraining performs poorly. With any of our two approaches, the performance gets improved, and our full model performs the best. All models are trained with 192×640 monocular videos and Resnet18 [15] as backbone, with results not post-processed [11].

KITTI [6] and evaluate with Garg's crop [9], with standard cap of 80m [11]. Results are reported in Tab. 1, showing that we help a large number of existing models to achieve better performance.

Other Datasets. To fully justify our benefits, we also carried out experiments on CityScapes [3] and NYUv2 [28]. Again, our methods consistently bring significant improvements to the existing model. Note that CityScapes even witnesses more significant improvements than KITTI. In order to further validate our generalizability, we also report results of training on KITTI but evaluating on CS and NYUv2.

Method	Train / Test	AbsR	SqR	$\log_{10}$	RMSE	δ_1
MD2 [12]	CS / CS	0.129	1.569	–	6.876	0.849
+ Ours*		**0.125**	**1.356**	–	**6.618**	**0.856**
MD2 [12]	KITTI / CS	0.163	1.883	–	8.967	0.757
+ Ours		**0.160**	**1.854**	–	**8.954**	**0.764**
MD2 [12]	KITTI / NYUv2	0.399	0.679	0.159	1.227	0.420
+ Ours		**0.370**	**0.610**	**0.142**	**1.133**	**0.459**

* Only δ needs to be fine-tuned to 0.4 in CityScapes to reach the best performance.

Table 3. **Generalization to other datasets.** Images size: NYUv2 - 256×320; **C**ity**S**capes - 416×128 (with preprocessing from [38]). One more metric log_{10} is reported in NYUv2.

5.3. Ablation Study

We validate our components in Tab. 2 and then analyse our design decisions in detail.

Can the loss function average out the irrational loss? Obviously, the proportion of the ambiguous pixels is low compared to the whole image. So, does our Amb.-Masking technique really matter? We made statistics for the ambiguous pixels over 10^3 batches and report the *mean* value of each metric in Tab. 4. Although the number percentage of the ambiguous pixels is not high, the key point is that they each have large irrational loss. As a result, $\sim$20% of the final photometric loss comes from these unreasonable pixels, which is actually not a low proportion (almost doubles from the number proportion).

	Number %	Photometric Loss	Loss Value %
Ambiguous pixels	10.66%	0.2415	19.43%
Other pixels	89.34%	0.1195	80.57%

Table 4. Statistical mean values of the ambiguous pixels obtained from 10^3 training batches.

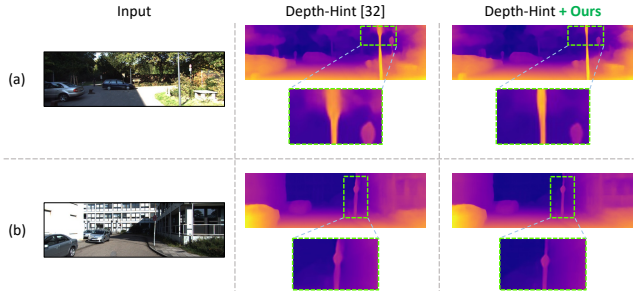


Figure 4. **Qualitative comparisons**. **(a)** Auto-Blur enlarges receptive fields (Sec. 5.4), helping the pole distinguish from the high-freq background. **(b)** Amb.-Mask helps to exclude ambiguous pixels whose depths are neither from the pole nor the building.

How Auto-Blur cooperate with image pyramid loss? Previous works [12, 18, 32] adopt an image pyramid to evaluate the photometric loss, where a low resolution image is similar to smoothing input images as ours. So, based on image pyramid loss, how can our Auto-Blur still help? (*i*) We are texture-specific. We adaptively (Eq. 20&21) smooth the high-freq regions which could confuse the photometric loss and keep the original low-freq regions, whereas the pyramid roughly smooths the whole image, which further weakens the supervision signal in texture-less regions that is already weak. (*ii*) We enlarge the receptive field by making each pixel attach the photometric information of its surroundings, thus no information loses. While downsampling, *e.g.*, could directly erase a two-pixel wide pole.

Hyper-params decisions. We study the hyper-params in Tab. 5. For threshold λ, $\downarrow \lambda$ would wrongly smooth the texture-less regions, as the already-weak supervision signal on them will be further weakened. $\uparrow \lambda$ would miss some pixels in high-freq regions which could confuse the photometric loss as illustrated in Fig. 2. For kernel size s in Auto-Blur, if $\downarrow s$, the receptive field could not be effectively enlarged when measuring pixel similarity. If $\uparrow s$, the central pixel's contribution (its own characteristic color) is reduced since the Gaussian distribution gets 'shorter' and 'wider'. See Supp. for ablations of all hyper-params.

5.4. Interpretable Analysis of Auto-Blur

Based on the OpenCV result of a specific case in Fig. 5, we give clear explanations on the effectiveness of the pro-

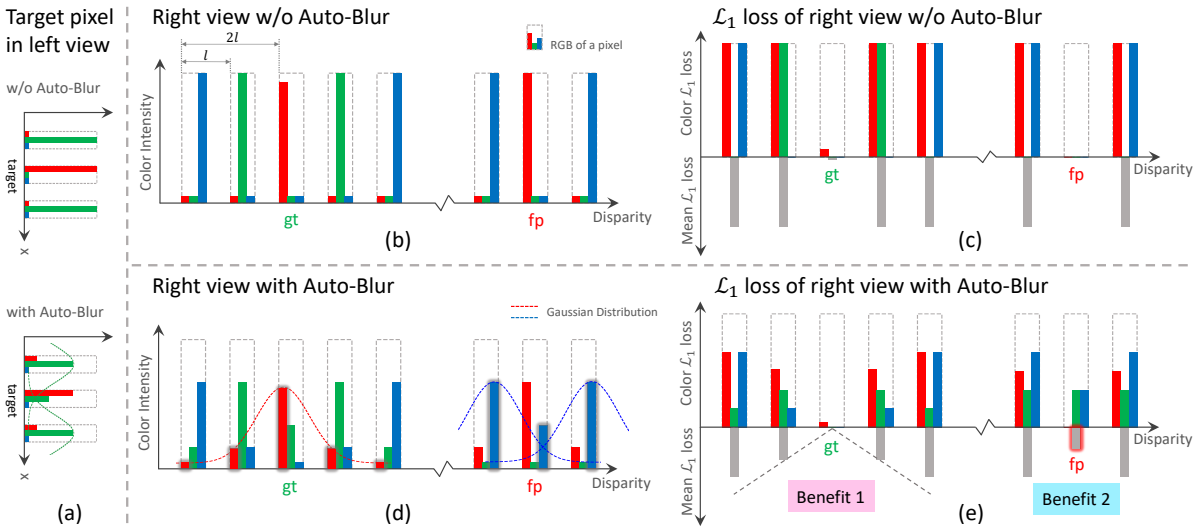


Figure 5. **An example of why Auto-Blur works. (a)** The color intensity of the target pixel (and its neighbours) in left view with and without Auto-Blur. **(b)** Original right view. The red *gt* pixel is located in a high-frequency area filled with R, G, B pixels, and there is a target-like (red) pixel in the distance, *i.e* the *false-positive*. **(c)** Losses for all estimated disparities are the same except *gt* and *fp*. Notably, *fp* shows a smaller loss than *gt*. **(d)** Under the action of Gaussian kernel, all pixels affect the surroundings and are affected by surroundings. Note that all color channels follow Gaussian distribution independently, the red and blue are highlighted just to illustrate benefit 1 and 2. **(e)** In contrast to (c), **Benefit 1:** The augmented $\mathcal{L}_1$ loss becomes *absolutely fair* as defined in Definition 1; **Benefit 2:** The loss of *fp* gets increased so that it can no longer deceive the network (detailed analysis in Sec. 5.4). All values are calculated by OpenCV library, with Gaussian kernel size set to $4l+1$, σ set to *s.t.* $f_{gaussian}(0) = \frac{2}{3}, f_{gaussian}(l) = \frac{1}{6}, f_{gaussian}(2l) \approx 0$ and borders are zero-padding.

λ	AbsR	SqR	δ_1	s	AbsR	SqR	δ_1
0.15	0.113	0.844	0.879	7	**0.112**	0.836	0.878
0.20	**0.112**	**0.834**	**0.880**	9	**0.112**	**0.834**	**0.880**
0.25	0.113	0.881	0.877	11	0.113	0.868	0.877

Table 5. **Ablations on hyper-params.** Full results in Supp.

posed Auto-Blur from two aspects.

A Fair Judge. As benefit 1 in Fig. 5e shows, being an *absolutely fair loss* in a certain range as defined in Definition 1, $\mathcal{L}_1 + \mathcal{L}_{ssim}$ photometric loss can fairly and accurately assess the network predictions. While in baseline, no matter how much the predicted disparity deviates from *gt*, loss remains unchanged. The role of Auto-Blur is to 'radiate' *gt*'s characteristic color (red) to the surroundings (Fig. 5d left). Moreover, this radiation is inversely proportional to the distance, so that as the distance between the predicted disparity and *gt* gets smaller, the photometric information of *gt* gets stronger, informing the network the current disparity is getting closer to *gt*. Thus, within a certain range, this strategy can make the penalty on the network increase proportionally with the degree of prediction error.

Expose False-Positive. When the assumption of photometric consistency breaks down, some false-positive (*fp*) pixels may look more like the target pixel than *gt*, thereby fooling the network. (*E.g.* in Fig. 4a, the pole and background's depths mixed together, since some white background pixels' color might incorrectly match the white pole). As benefit 2 in Fig. 5e shows, Auto-Blur helps to increase penalty/loss of *fp*, thus preventing it from matching the target pixel. The benefit of Gaussian kernel's 'color radiation' is to increase receptive fields when measuring pixel similarity, while keeping the original CNN kernel size. Concretely, the *fp* is radiated by surrounding blue (Fig. 5d right), whereas *gt* is radiated by surrounding green (Fig. 5a bottom), thus making a difference. Loss of *fp* therefore gets increased, preventing it from deceiving the network.

6. Conclusion

We examine the photometric loss of self-supervised MDE from the spatial frequency domain, and reveal two general issues rarely noticed by previous MDE researchers. We draw a conclusion that the pixel-level ambiguity in the object junctions of input images is a more fundamental reason that hinders sharper depth edges. Furthermore, we demonstrate that the photometric loss function cannot *fairly* assess the predictions in high-freq regions, and could sometimes produce false global optimums. Interestingly, we prove blurring the input images could reduce such *unfairness* and efficiently enlarge receptive fields. Our approaches are highly lightweight and versatile. A large number of existing models get performance boosts from our methods, while no extra inference computation is needed at all.

Acknowledgement. This work was supported by National Natural Science Foundation of China (No. 62172021).

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014.

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[8] Huachen Gao, Xiaoyu Liu, Meixia Qu, and Shijie Huang. Pdanet: Self-supervised monocular depth estimation using perceptual and data augmentation consistency. *Applied Sciences*, 11(12):5383, 2021.

[9] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016.

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

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[13] Juan Luis GonzalezBello and Munchurl Kim. Forget about the lidar: Self-supervised depth estimators with med probability volumes. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12626–12637. Curran Associates, Inc., 2020.

[14] Brent Griffin, Victoria Florence, and Jason Corso. Video object segmentation-based visual servo control and object depth estimation on a mobile robot. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1647–1657, 2020.

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

[16] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2360–2367, 2013.

[17] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[18] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12642–12652, 2021.

[19] 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.

[20] Youngjung Kim, Hyungjoo Jung, Dongbo Min, and Kwanghoon Sohn. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8):4131–4144, 2018.

[21] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

[22] Rui Li, Xiantuo He, Yu Zhu, Xianjun Li, Jinqiu Sun, and Yanning Zhang. Enhancing self-supervised monocular depth estimation via incorporating robust constraints. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3108–3117, 2020.

[23] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2624–2641, 2019.

[24] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020.

[25] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. HR-Depth: High resolution self-supervised monocular depth estimation. *arXiv preprint arXiv:2012.07356*, 6, 2020.

[26] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity,

optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[27] Michaël Ramamonjisoa, Michael Firman, Jamie Watson, Vincent Lepetit, and Daniyar Turmukhambetov. Single image depth prediction with wavelet decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11089–11098, 2021.

[28] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[29] Minsoo Song, Seokjae Lim, and Wonjun Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE transactions on circuits and systems for video technology*, 31(11):4381–4393, 2021.

[30] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019.

[31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[32] Jamie Watson, Michael Firman, Gabriel J Brostow, and Daniyar Turmukhambetov. Self-supervised monocular depth hints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2162–2171, 2019.

[33] Haofei Xu and Juyong Zhang. AANet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2811–2820, 2019.

[35] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 464–473. IEEE, 2021.

[36] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 817–833, 2018.

[37] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019.

[38] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[39] Shengjie Zhu, Garrick Brazil, and Xiaoming Liu. The edge of depth: Explicit constraints between segmentation and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13116–13125, 2020.