sensors

MDPI

*Article*

# Ranking-Based Salient Object Detection and Depth Prediction for Shallow Depth-of-Field

**Ke Xian, Juewen Peng, Chao Zhang, Hao Lu and Zhiguo Cao ***

National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China; kexian@hust.edu.cn (K.X.); im.pengjw@gmail.com (J.P.); ZhangC_22@hust.edu.cn (C.Z.); hlu@hust.edu.cn (H.L.)
* Correspondence: zgcao@hust.edu.cn; Tel.: +86-027-8755-8918; Fax: +86-027-8754-3594

**Abstract:** Shallow depth-of-field (DoF), focusing on the region of interest by blurring out the rest of the image, is challenging in computer vision and computational photography. It can be achieved either by adjusting the parameters (e.g., aperture and focal length) of a single-lens reflex camera or computational techniques. In this paper, we investigate the latter one, i.e., explore a computational method to render shallow DoF. The previous methods either rely on portrait segmentation or stereo sensing, which can only be applied to portrait photos and require stereo inputs. To address these issues, we study the problem of rendering shallow DoF from an arbitrary image. In particular, we propose a method that consists of a salient object detection (SOD) module, a monocular depth prediction (MDP) module, and a DoF rendering module. The SOD module determines the focal plane, while the MDP module controls the blur degree. Specifically, we introduce a label-guided ranking loss for both salient object detection and depth prediction. For salient object detection, the label-guided ranking loss comprises two terms: (i) heterogeneous ranking loss that encourages the sampled salient pixels to be different from background pixels; (ii) homogeneous ranking loss penalizes the inconsistency of salient pixels or background pixels. For depth prediction, the label-guided ranking loss mainly relies on multilevel structural information, i.e., from low-level edge maps to high-level object instance masks. In addition, we introduce a SOD and depth-aware blur rendering method to generate shallow DoF images. Comprehensive experiments demonstrate the effectiveness of our proposed method.

**Keywords:** salient object detection; depth estimation; shallow depth-of-field; ranking loss

check for updates

**Citation:** Xian, K.; Peng, J.; Zhang, C.; Lu, H.; Cao, Z. Ranking-Based Salient Object Detection and Depth Prediction for Shallow Depth-of-Field. *Sensors* **2021**, *21*, 1815. https://doi.org/10.3390/s21051815

Academic Editor: Yitzhak Yitzhaky

Received: 25 January 2021
Accepted: 3 March 2021
Published: 5 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Breathtaking photography is all about narrative, i.e., the story the image is telling. There are numerous methods to enhance a photo to tell a story, no matter what the subject or the techniques we choose. Shallow depth-of-field (aka shallow DoF), drawing the viewers' attention to the region of interest by blurring out the rest of the image, is such a technique. With the smartphone being widely used in daily life, we always use the smartphone cameras to capture photos. However, these acquired photos are always all-in-focus due to the narrow baseline and fixed aperture. Hence, more and more attention has been paid to the shallow DoF rendering techniques [1–5] in recent years.

To render realistic shallow DoF, depth information is required. Some methods use stereo techniques to compute depth maps from stereo images [3,4] and dual-pixel data [2]. However, such methods depend on specific hardware (e.g., stereo cameras or dual-pixel sensors) to capture two views. It is challenging to predict large depth fields due to the narrow baseline. In addition to the stereo-based methods, some other studies [1,2,6] render shallow DoF effects for portrait photos. Although these methods can generate promising shallow DoF images, they cannot generalize well to other scenes. In this paper, we step further to study the problem of rendering shallow DoF effects from an unconstrained image. To this end, we propose a method that consists of a salient object detection (SOD)

module for extracting salient objects, a monocular depth prediction (MDP) module for estimating scene depth, and a DoF rendering module for generating shallow DoF images (please refer to Figure 1).




**Figure 1.** Shallow depth-of-field (DoF) rendering from an arbitrarily captured image.

The SOD module indicates the depth range covered by the salient object, in which the focal plane is selected to be within this range. Previous deep learning-based SOD methods [7–11] mainly focus on designing effective network architectures, e.g., HED-based architecture [7], hybrid upsampling operator [8], recurrent localization network [9], pixel-wise contextual attention network [10], and pyramid self-attention module [11]. In order to train these models in an end-to-end manner, the binary cross-entropy loss is widely used. However, such loss, calculated in a pixel-wise manner, cannot explicitly model neighboring relationships. Hence, the predictions often suffer from two problems: (i) the nearby pixels that have different labels (i.e., foreground and background) have the same outputs, named interclass indistinction; (ii) the pixels that share the same label have different outputs, named intraclass inconsistency. To address these issues, an option is to consider structural information. For instance, a fully connected Conditional Random Field (CRF) is always used as a postprocessing strategy to improve spatial coherence [7,12,13]. Inspired by HRWSI (Xian et al., [14]), we propose a novel label-guided ranking loss, which explicitly considers the structure information and can be trained in an end-to-end fashion. Given two points from an image, humans are always good at judging which one is more salient. Thus, a question arises: Can we model this behavior? We sample point pairs from ground-truth saliency maps and annotate which point is more salient for each pair to mimic this process. The sampled pairs can be seen as questions asked by the teacher (i.e., ground truth) and indicate the student (i.e., model) to give answers. Specifically, we define two types of the sampled pairs. One is heterogeneous pair (i.e., the sampled two points are from salient objects and background, respectively), and the other one is homogeneous pair (i.e., cosaliency and cobackground). Given these sampled point pairs, we use a ranking loss [15] to train our model. Unlike pixel-wise losses, the ranking loss only depends on the relative saliency (e.g., point A is more salient than point B or vice versa). More specifically, our label-guided ranking loss is comprised of two terms. One is a heterogeneous ranking loss, which encourages the sampled salient pixels to be different from background pixels. The other is a homogeneous ranking loss that penalizes the inconsistency of salient pixels or background pixels.

The MDP module, which estimates the depth map of an arbitrary image, is used for controlling the degree of blur. Since web stereo depth datasets (e.g., ReDWeb [15] and HRWSI [14]) can only provide depth up to a scale and shift due to the unknown camera baselines and postprocessing, training with pixel-wise losses (e.g., $\ell_1$ [5], berHu [16], and scale-invariant loss [17]) cannot generate promising predictions. Therefore, we learn MDP from such pseudodepth data by adopting a structure-guided ranking loss. Different from the loss used in SOD, this ranking loss depends on the depth ordinal relationships. For example, point A is closer than point B or vice versa. In particular, we sample point pairs according to the low-level edge maps and high-level object instance masks, leading to the generation of consistent depth maps with sharp depth discontinuities.

To obtain the final shallow DoF images, we take the all-in-focus image, the saliency map, and the depth map as inputs to the DoF rendering module. The salient map is used to determine the focal plane, while the depth map is used to adjust the blur degree. To synthesize realistic shallow DoF images, we propose a physically motivated method termed scatter-to-gather. Traditional rendering methods always use the gather and scatter operators to render shallow DoF. However, these methods [2,3,18] in practice utilize the layered depth rendering strategy that applies a blur kernel to each depth plane. To keep the refocused plane clear and enable smooth transition around depth discontinuities, our method processes each pixel one by one.

We conduct numerous experiments on SOD, MDP, and shallow DoF rendering. The experimental results demonstrate the effectiveness of our proposed method. In summary, the main contributions of this work are as follows:

- We present an automatic system consisting of a SOD module, an MDP module, and a DoF rendering module for rendering realistic shallow DoF from an arbitrary image.
- We introduce a label-guided ranking loss for SOD. It is a combination of a heterogeneous ranking loss and a homogeneous ranking loss. The heterogeneous ranking loss aims to encourage salient objects to be independent of the background, while the homogeneous ranking loss is dedicated to improving spatial coherence.
- We propose a novel rendering method to render realistic shallow DoF images.

## 2. Related Work

### *2.1. Salient Object Detection*

Traditional SOD methods are mainly based on hand-craft features and prior knowledge, including center-surrounding differences [19,20] and boundary prior knowledge [21,22]. Since a detailed survey of these methods is beyond the scope of this paper, we refer the reader to the survey paper [23] for more details. Here, we focus on the reviews of deep learning-based methods.

In recent years, deep learning-based methods [24–26] have achieved outstanding performance in visual saliency detection [27–29]. For instance, Li et al. [24] design a multilayer fully connected network to predict the saliency score of each superpixel. However, due to a large number of parameters, the fully connected layer decreases computational efficiency. To address this issue, several methods adopt a Fully Convolutional Network (FCN) to generate pixel-wise saliency maps. Liu and Han [25] propose a deep hierarchical salient network to extract both global and local information for SOD. Zhang et al. [8] integrate reformulated dropout layers and hybrid upsampling operations into an encoder-decoder network. To get detail-preserving outputs, multistream networks have been widely used in SOD. Tang and Wu [30] combine cascaded convolutional neural networks (CNN) and adversarial learning for SOD. The two-stream networks, consisting of an encoder-decoder network for global saliency estimation and a deep residual network for local refinement, are designed as a generator. To enable adversarial learning, a discriminator is then incorporated to distinguish the ground-truth saliency maps from the fake ones (i.e., predictions). Recently, contour information [9,31,32] and attention mechanism [10,11,33] have also been attempted for improving the performance of SOD models. Nevertheless, the aforementioned methods focus on network architecture designs, ignoring the explorations of the loss function. The commonly used binary cross-entropy loss, computed in a pixel-wise manner, ignores the neighboring relationships. Training with such a loss suffers from interclass indistinction and interclass inconsistency. To mitigate this issue, we propose a label-guided ranking loss that explicitly models the neighboring relationships. In addition, this operation is similar to the visual attention mechanism of primates (i.e., center-surrounding differences [19,20]).

### *2.2. Monocular Depth Prediction*

Deep learning-based MDP algorithms [14–17] have achieved outstanding performance in recent years. Eigen et al. [17] is the first to apply a multiscale CNN to MDP. Although

they use a coarse-to-fine strategy to predict depth maps, the predictions still lack details because of their low resolution. To get finer predictions, some methods [34] train CRF and CNN in a unified framework. Some other methods propose to learn depth by multitask learning, including semantic segmentation [35,36], surface normal estimation [37], and contour detection [35]. However, these methods need additional training labels. Such labels, usually manually annotated, are expensive to collect.

Apart from the aforementioned supervised methods, some researchers attempt to learn depth in a self-supervised fashion [38–40]. The basic idea behind these methods is image reconstruction. Instead of using ground-truth depth for supervision, they propose to learn depth or pose in latent space, based on which they can reconstruct the target view. Further, they use the synthesized target view and the ground-truth one to compute the reconstruction loss. Despite the significant progress made, these self-supervised methods still suffer from limitations, such as occlusions, nonrigid motion, and generalization.

The aforementioned methods are mainly trained in constrained scenes and their generalization to other scenes is not well. In other words, these methods trained on one dataset often fail to get promising predictions on a different one. To learn depth in general scenes with a single model, recent studies [15,41–44] start from constructing in-the-wild RGB-D datasets. For example, Chen et al. [41] propose the DIW dataset, which consists of about 495K natural images. However, they only provide one pair of ordinal relationships for each image, which is not enough to train an accurate MDP model. To address this issue, ReDWeb [15] and MegaDepth [42] were proposed at the same venue. The former comes from web stereo images, while the latter comes from web image sequences. Although these methods have good generalization to unconstrained scenes, their performance can be further improved, especially on depth discontinuities. Thus, Xian et al. [14] propose to guide the network towards the depth discontinuities by low-level edge maps and high-level object instance masks.

### *2.3. DoF Rendering*

Realistic DoF rendering usually requires accurate depth information. Thus, some methods use RGB-D images [45–47] and stereo image pairs [3,4] to render DoF images. For example, SteReFo [3] interrelates stereo-based depth estimation and refocusing effectively. However, such methods rely on specific hardware, e.g., RGB-D sensors and stereo cameras. Therefore, some other methods [48,49] use off-the-shelf MDP methods to predict scene depth. In addition to explicitly using depth maps to render DOF images, some deep learning-based methods [18,50,51] propose to implicitly learn depth from all-in-focus and shallow DoF image pairs. Specifically, given an all-in-focus image as input, the network is optimized to render a synthetic shallow DOF image. This method, therefore, does not require ground truth supervision on depth. Unlike the aforementioned methods, some other methods [1,6] achieve DoF effects by portrait segmentation. Xu et al. [1] learn a spatially-variant RNN [52] filter to render a shallow DoF image from a portrait photo. Besides, some approaches [2,3,5] which manually select a focal plane, have also been proposed for DoF rendering. By contrast, this paper proposes a method to automatically render a shallow DoF image from an arbitrary natural image.

## 3. Method

In this section, we elaborate on our proposed method for shallow DoF rendering. As shown in Figure 2, our method consists of three modules: salient object detection, depth prediction, and DoF rendering. The rest of this section is organized as follows. Section 3.1 presents a detailed description of our ranking-based SOD module. Section 3.2 describes the ranking-based MDP module. The shallow DoF rendering module is illustrated in Section 3.3.

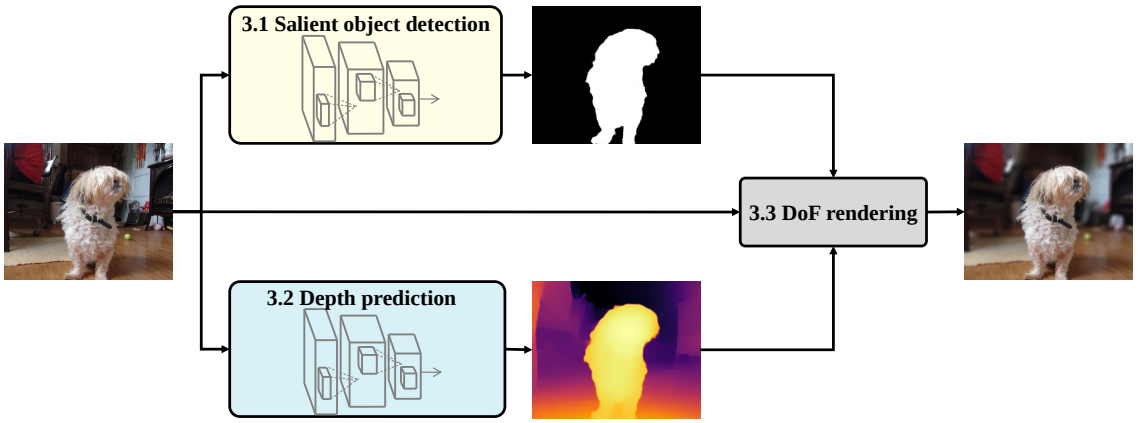



**Figure 2.** Overview of our proposed method.

### *3.1. Salient Object Detection*

#### 3.1.1. Label-Guided Ranking Loss

Instead of training with a pixel-wise loss, we propose a novel label-guided ranking loss to explore the pair-wise relations in this work explicitly. As shown in Figure 3, the pair-wise relations can be categorized into two groups: (i) heterogeneous pairs, whose labels are contrary (i.e., foreground and background); (ii) homogeneous pairs, whose labels are identical (i.e., foreground and foreground, background and background). To improve inter-class distinction and intraclass consistency, our loss function comprises a heterogeneous ranking loss and a homogeneous ranking loss. The basic idea behind the label-guided ranking loss is that we design a heterogeneous ranking loss to encourage the sampled salient pixels to be different from background pixels. Meanwhile, a homogeneous ranking loss is incorporated to penalize the inconsistency of salient pixels or background pixels.

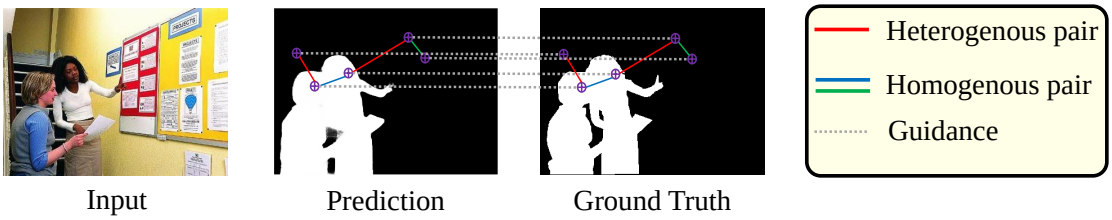



**Figure 3.** Illustration of our proposed label-guided ranking loss.

Heterogeneous ranking loss: Given a ground truth saliency map $G$, we randomly sample $N$ point pairs $(i, j)$, where $i$ and $j$ represent the first and second points' locations, respectively. For each point pair, the first point denoted as $g_i$ comes from the background, while the second point belonging to a salient region can be represented by $g_j$. Guided by the index $(i, j)$ from ground truth, we use $(p_i, p_j)$ to represent the sampled point pair from the predicted saliency map $P$. As a result, the sampled set can be represented by $Z$. Note that $N$ is image-dependent because different images have different numbers of foreground pixels. We sample N point pairs from each image, where N equals the minimum number of pixels between foreground and background.

To improve interclass distinction, we define the heterogeneous ranking loss as:

$$L_{hete} = \frac{1}{N} \sum_{(p_i, p_j) \in Z} \log(1 + \exp(\alpha(p_i - p_j))) \quad (1)$$

where $\alpha$ is a constant factor, and the term $p_i - p_j$ can be positive or negative. If this term is positive, which means $p_i$ has a greater possibility to be foreground, the loss $L_{hete}$ would

be large. In order to minimize the $L_{hete}$, the term $p_i - p_j$ should be as small as possible. Therefore, this loss encourages the predicted $p_i$ and $p_j$ to be background and foreground, respectively. Meanwhile, it enlarges the difference between $p_i$ and $p_j$.

Homogeneous ranking loss: $L_{hete}$ only measures the difference between salient objects and background, which ignores the intraclass consistency. Therefore, we supplement a homogeneous ranking loss that minimizes the intraclass difference. Instead of using a pixel-wise MSE loss, the homogeneous ranking loss explores the pair-wise relations in an explicit way. Considering that there exist two relations of the homogeneous pairs (cosaliency and cobackground) and the scales of losses are different, the homogeneous ranking loss is thus comprised of $\tilde{L}_{co-bg}$ and $\tilde{L}_{co-sal}$.

To be specific, We define the pixels sampled from background as $Z_b = \{p_i | i = 1, \ldots, N\}$ and the pixels sampled from salient objects as $Z_s = \{p_j | j = 1, \ldots, N\}$. Then we permute $Z_b$ and $Z_s$ to get $\hat{Z}_b$ and $\hat{Z}_s$. So the losses of cobackground pairs and cosaliency pairs can be calculated by:

$$\tilde{L}_{co\text{-}bg} = \frac{1}{N} \sum_{p_i \in Z_b, \hat{p}_i \in \hat{Z}_b} (p_i - \hat{p}_i)^2 \tag{2}$$

$$\tilde{L}_{co\text{-}sal} = \frac{1}{N} \sum_{p_j \in Z_s, \hat{p}_j \in \hat{Z}_s} (p_j - \hat{p}_j)^2 \tag{3}$$

where $\tilde{L}_{co\text{-}bg}$ and $\tilde{L}_{co\text{-}sal}$ measure the consistency of background and salient objects, respectively. During training, we observe $\tilde{L}_{co\text{-}sal}$ is ten times larger than $\tilde{L}_{co\text{-}bg}$. As a result, we use a hyperparameter $\sigma$ to balance the difference. Thus, the homogeneous ranking loss can be formulated as:

$$L_{homo} = \tilde{L}_{co\text{-}bg} + \sigma \tilde{L}_{co\text{-}sal} \tag{4}$$

Finally, we define the label-guided ranking loss as:

$$L_{sal} = L_{hete} + \lambda L_{homo} \tag{5}$$

where $\lambda$ is a balancing factor. The whole computational procedure of the label-guided ranking loss is summarized in Algorithm 1.

**Algorithm 1** The procedure for label-guided ranking loss

**Input:** Ground truth saliency maps $G$, predictions $P$

**Output:** Label-guided ranking loss $L$

1: Guided by salient objects from $G$, sample pixels $Z_s$ from $P$
2: Guided by background from $G$, sample pixels $Z_b$ from $P$
3: Permute $Z_s$ and $Z_b$ to get $\hat{Z}_b$ and $\hat{Z}_s$
4: Compute heterogeneous ranking loss according Equation (1)
5: Compute homogeneous ranking loss according Equation (4)
6: Output final loss $L_{sal}$ according Equation (5)

#### 3.1.2. Network Architecture

Figure 4 illustrates the schematic representation of our encoder-decoder network architecture. The whole network is based on the deep layer aggregation network structure [53], and we utilize the DLA-60 as our backbone network. In the *encoding* part, we adopt 3 convolution blocks (C1, C2, C3) and 4 hierarchical deep aggregation modules (H1, H2, H3, H4). Specifically, we set the convolutional kernel size to $7 \times 7$ in C1 and $3 \times 3$ in C1, H1, H2, H3, and H4. As shown in Figure 4, we maintain feature maps in C1 and C2 at the same resolution as the input image, and then the feature maps are downsampled via a convolution layer with a stride of 2. The hierarchical deep aggregation modules H1, H2, H3, and H4 have {1, 2, 4, 1} stages. For each stage, it contains two residual blocks and

an aggregation node. The aggregation node, used to combine and compress its inputs, can be based on any block or layer. For simplicity and efficiency, we use a single $3 \times 3$ convolution followed by batch normalization and nonlinear activation. Besides, we use skip connections and a root aggregation node to fuse the feature maps between two continuous hierarchical deep aggregation modules. For example, as shown in Figure 4, the root aggregation module combines the features generated by H1 and H2. In particular, we use a $3 \times 3$ convolution layer with a stride of 2 to downsample the feature maps from H1, then concatenate with the feature maps from H2 followed by a residual block. To expand receptive fields without losing resolution, we utilize dilated convolution in the last hierarchical deep aggregation module.

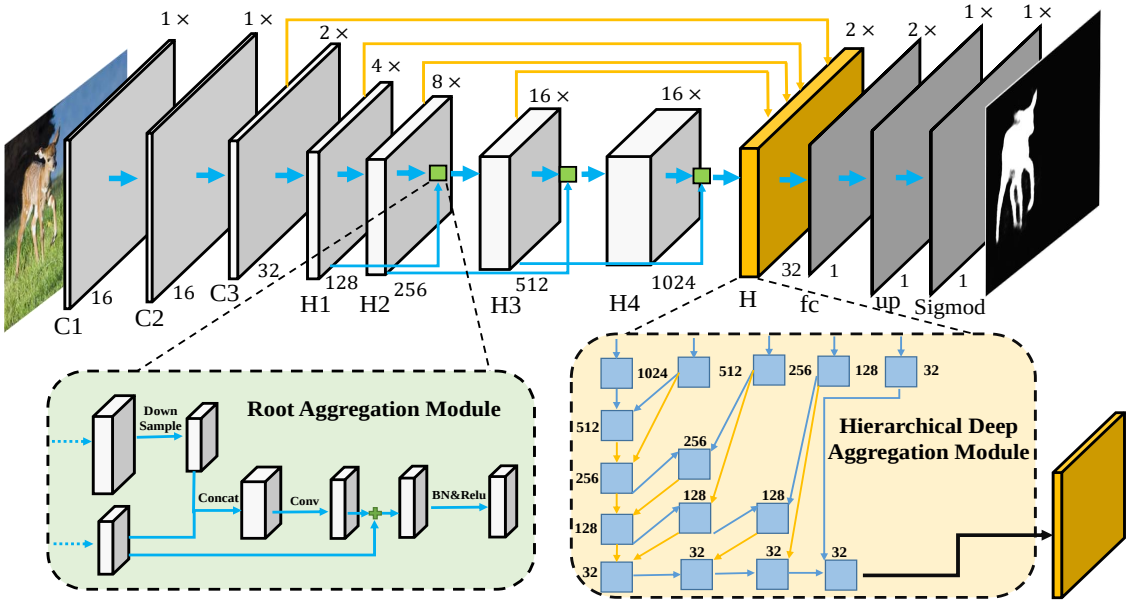



**Figure 4.** Illustrations of the proposed network architecture. In the hierarchical deep aggregation module, the orange lines indicate deconvolutional operators, and the blue blocks consist of convolution/BN/ReLU layers.

In the *decoding* part, we adopt a hierarchical deep aggregation module (H), a convolution layer (FC layer), a deconvolution layer (UP layer), and a sigmoid layer. As shown in Figure 4, we utilize the feature maps generated from (C3, H1, H2, H3, H4) in the encoding part. The whole hierarchical deep aggregation module, successively aggregating and upsampling feature maps, contains four levels. For example, at the first level of H, we fuse H4 and H3 to obtain feature maps $l1$ that have the same dimension as H3. At the following level, we fuse the $l1$ and H3 to construct $h2_1$, which has the same dimension as H2. Similarly, we combine the $h2_1$ and H2 to obtain feature maps with the same dimension as H2. Given an input image at resolution $256 \times 256 \times 3$, the hierarchical deep aggregation module generates the final feature maps at resolution $128 \times 128 \times 32$. To get our final output ($256 \times 256 \times 1$), we stack a FC layer ($1 \times 1$ kernel size), a deconvolution layer and a sigmoid layer.

The traits of our network are twofold. Firstly, in contrast to most prior works [9,10] that only aggregates features from neighboring layers, our network instead leverages the information of most former layers via skip connections, thus integrating information at different levels. Secondly, the parameters of our model have been greatly reduced via deep layer aggregation, which enables fast salient object detection.

### *3.2. Monocular Depth Prediction*

To render realistic shallow DoF effects from an arbitrary image, depth information is required. As recent MDP methods commonly use the network architecture proposed by Xian et al. [15], we also use the same one to predict depth maps. The network is mainly

comprised of an encoding backbone and a multiresolution fusion module. The encoding backbone extracts features of different resolutions and semantics. The multiresolution fusion module fuses coarse high-level semantic features with fine-grained low-level features, which enables high-resolution outputs and preserves fine details simultaneously. Since the MDP module is not a contribution of this paper, please refer to the reference [15] for more details.

We train the MDP model on HRWSI [14] dataset that consists of 20K high-quality training images. These data have unknown depth scale and shift factors, directly using pixel-wise losses (e.g., $\ell_1$, $\ell_2$, and scale-invariant loss) cannot get promising predictions [14]. Therefore, we use a ranking-based loss for training. Given a RGB image $I$, we learn a function $D = \mathcal{F}(I)$ in a supervised manner, where $D \in \mathbb{R}^{h \times w \times 1}$ is the generated depth map. The loss can be computed on a set of point pairs with ordinal annotations. In particular, for each point pair with predicted depth values $[d_0, d_1]$, the pair-wise ranking loss can be formulated as:

$$\chi(d_0, d_1) = \begin{cases} \log(1 + \exp(-\kappa(d_0 - d_1))), & \kappa \neq 0 \\ (d_0 - d_1)^2, & \kappa = 0, \end{cases} \tag{6}$$

where $\kappa$ is the ground truth ordinal label, which can be derived from a ground truth depth map:

$$\kappa = \begin{cases} +1, & d_0^*/d_1^* \geq 1 + \sigma, \\ -1, & d_0^*/d_1^* \leq \frac{1}{1+\sigma}, \\ 0, & otherwise. \end{cases} \tag{7}$$

Here, $\sigma$ is a tolerance threshold [14] that is set to 0.03 in experiments, and $d_i^*$ represents the ground truth depth value. This loss encourages the predicted $d_0$ and $d_1$ to be the same when the point pair are close in the depth space, i.e., $\kappa_i = 0$; otherwise, it would enlarge the difference between $d_0$ and $d_1$ for minimization. Then, the ranking loss of the sampled pairs can be computed by:

$$\mathcal{L}_{rank} = \frac{1}{N} \sum_i \chi(d_{i,0} - d_{i,1}). \tag{8}$$

where $N$ is the number of sampled pairs. Instead of random sampling, we follow HRWSI [14] to combine low-level edge-guided sampling and high-level object instance sampling. In this way, the networks would pay attention to the salient structure of the given image.

To encourage smoother gradient changes and sharper depth discontinuities in the predicted depth maps, we add a multiscale scale-invariant gradient matching loss. Given $R_i = D_i - D_i^*$, this loss can be defined as:

$$\mathcal{L}_{grad} = \frac{1}{M} \sum_s \sum_i (|\nabla_x {R_i}^s| + |\nabla_y R_i)^s|), \tag{9}$$

Here, $M$ and $R^s$ represent the number of valid pixels and the difference of depth maps at scale $s$, respectively. In our experiments, we use four scales.

By combining the ranking loss $\mathcal{L}_{rank}$ and the multiscale scale-invariant gradient matching loss $\mathcal{L}_{grad}$, the final loss for training the MDP model is:

$$\mathcal{L}_{depth} = \mathcal{L}_{rank} + \beta \mathcal{L}_{grad}, \tag{10}$$

Following [14], we set the $\beta$ to 0.2 in our experiments.

### *3.3. Shallow DoF*

To render realistic shallow DoF, we design a physically motivated method termed "scatter-to-gather" (S2G). The basic idea, that the light scattering can be converted to a gathering operation indirectly, is similar to [2]. However, we process each pixel one by one

instead of using the layered depth rendering strategy. Given an input image $I_a$ and a blur kernel $K$, the rendered DoF image $B$ can be computed by:

$$B = \sum_{\Delta i} I_a(i + \Delta i) K_{i+\Delta i}(-\Delta i), \tag{11}$$

Typically, the point spread function shape is circular. We thus use the disk blur kernel to synthesize realistic shallow DoF images. According to [2], the radius of kernel can be computed by:

$$r = Lf|D(p_i) - d_f|, \tag{12}$$

Here, $L$ is the aperture size, $f$ is the focal length, $D(p_i)$ is the inverse depth of pixel $p_i$, and $d_f$ is the depth of focus. Since the aperture size and focal length belong to camera factors, we use $A$ to represent the multiplication of $L$ and $f$. Note that, $A$ controls the maximum blur degree.

As illustrated in Algorithm 2, we start from an all-in-focus image $I_a$ with its predicted saliency map $S$, normalized depth map $D$, and camera factor $A$. Note that we view the depth map $D$ as an inverse depth map during the whole rendering process. We first calculate the depth of focus $d_f$ by computing the median of the depth range covered by the salient object. For each pixel $p_i$ in an all-in-focus image, we use two accumulated terms $w_{sum}$ and $c_{sum}$ to record its weight and color intensity, respectively. Then, we find the neighboring pixels of $p_i$ according to the maximum blurring radius $r$. If the blur radius $r$ is larger than the distance $l$ between two pixels, the pixel $p_j$ will cast to the pixel $p_i$, and the weight will be divided by the square of $r$. After traversing all the neighboring pixels of pixel $p_i$, we can get the color values at the location $p_i$ by dividing the accumulated weight.

**Algorithm 2** The pipeline of rendering method S2G

**Input:** all-in-focus image $I_a$, saliency map $S$, depth map $D$, camera factor $A$
**Output:** DoF image $I_d$

$d_f \leftarrow \text{CalDepthFocus}(\text{D}, \text{S})$
**for** $p_i \leftarrow \text{TraverseImage}(\text{I}_\text{a})$ **do**
  $w_{sum} \leftarrow 0$
  $c_{sum} \leftarrow 0$
  **for** $p_j \leftarrow \text{FindNeighbor}(\text{p}_\text{i})$ **do**
    $l \leftarrow \text{Dist}(\text{p}_\text{i}, \text{p}_\text{j})$
    $r \leftarrow A \cdot |D(p_j) - d_f|$
    $w \leftarrow \mathbb{1}(r - l > 0) \cdot \frac{1}{r^2}$
    $w_{sum} \leftarrow w_{sum} + w$
    $c_{sum} \leftarrow c_{sum} + w \cdot I_a(p_j)$
  **end for**
  $I_d(p_i) \leftarrow \frac{c_{sum}}{w_{sum}}$
**end for**

## 4. Experiments

### *4.1. Salient Object Detection*

Following [9,10,54], we use the DUTS-TR dataset [55] for training. We resize images to 256 × 256 with random horizontal/vertical flipping to avoid overfitting during training. We train our model using stochastic gradient descent (SGD) with an initial learning rate of 0.1, which is decayed by ×0.1 every 15 epochs. The momentum and the weight decay is set to 0.9 and 0.0005, respectively. The whole network is trained for 40 epochs with batch size 84 on two NVIDIA GTX 1080TI GPUs. We use the proposed label-guided ranking loss to train our network and set $\alpha$, $\sigma$, and $\lambda$ to 3.0, 0.1, and 1.0 in our experiments.

To evaluate the performance of our salient object detection module, we compare our method with the state-of-the-art approaches on six widely used saliency datasets: SOD [56],

ECSSD [57], PASCAL-S [58], HKU-IS [24], DUT-OMRON [59] and DUTS [55]. SOD contains 300 testing images, which are generated from the Berkeley segmentation dataset. Most images in this dataset have multiscale salient objects and complex backgrounds. ECSSD has 1000 images with various natural scenarios. PASCAL-S dataset contains 850 natural images, which are generated from the PASCAL VOC 2010 segmentation dataset. HKU-IS includes 4447 images that have multiple salient objects with low color contrast and various locations. DUT-OMRON contains 5168 challenging images that have one or more salient objects. DUTS is the largest salient object detection benchmark dataset. It consists of 10,533 training images (DUTS-TR) and 5019 testing images (DUTS-TE).

#### 4.1.1. Ablation Studies

**Comparison with baseline:** The label-guided ranking loss consists of two terms: the heterogeneous ranking term and the homogeneous ranking term. To analyze each part's contributions, we explore various configurations and evaluate the models on six datasets. We report maximum $F_\beta$-Measure, MAE, and structure-measure in Table 1. $L_{hete}$ means that we train the model with the heterogeneous ranking loss that only computes heterogeneous pairs' losses. $L_{hete} + \tilde{L}_{co-sal}$ is comprised of the heterogeneous ranking loss and the homogeneous ranking loss on cosaliency pairs. Note that the $\tilde{L}_{co-sal}$ term only computes the losses of pairs sampled from salient objects. Similarly, the $L_{hete} + \tilde{L}_{co-bg}$ term consists of the heterogeneous ranking loss and the homogeneous ranking loss on cobackground pairs. We use $L_{hete} + L_{homo}$ to represent the proposed label-guided ranking loss. As shown in Table 1, one can observe that adding $\tilde{L}_{co-sal}$ improves the performance when compared to $L_{hete}$. However, the improvements are limited, which means only considering co-salient pairs is not enough. In addition, we explore the combination of $L_{hete}$ and $\tilde{L}_{co-bg}$, which further improves the performance. Furthermore, we incorporate the $L_{hete}$ and $L_{homo}$ together to predict more accurate saliency maps.

**Comparison with other losses:** To demonstrate the effectiveness of our loss, we train the same network architecture with different loss functions. In particular, we compare our loss with four losses (Margin Ranking, MAE, MSE, and BCE). Table 2 shows the maximum $F_\beta$-Measure, MAE, and structure-measure scores on six challenging datasets. In addition to quantitative evaluations, we also show some qualitative examples in Figure 5. One can observe that our label-guided ranking loss achieves the best performance. Although we compute losses only on a sparse set of point pairs, the quantitative and qualitative results demonstrate that our model still performs better than those trained with dense per-pixel losses.

**Impact of the amounts of point pairs:** To analyze the impact of the amounts of point pairs, we sample a different number of pairs during training on DUTS-TR, and evaluate these models on DUTS-TE. Figure 6 shows the maximum $F_\beta$-Measure and MAE scores when trained with a different number of pairs. One can observe that training with more pairs improves performance. As the label-guided ranking loss uses an online sampling strategy, the diversity of samples would not be a key factor as the number of iterations increases. Besides, we did not see a significant difference in time consumption.

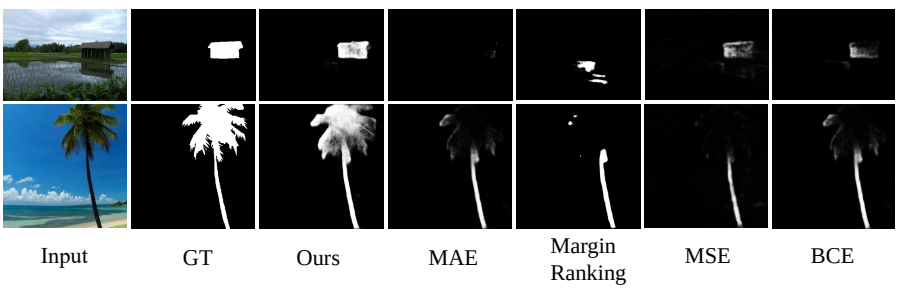



**Figure 5.** Qualitative results obtained by using different loss functions. Our results are more visually consistent with the ground-truth maps.

**Table 1.** Quantitative results of the proposed loss function with different configurations. $L_{hete}$ means only using the heterogeneous ranking loss, $L_{hete} + \tilde{L}_{co-bg}$ means adopting the heterogeneous ranking loss and the consistency loss of cobackground pairs, $L_{hete} + \tilde{L}_{co-sal}$ means adopting the heterogeneous ranking loss and the consistency loss of co-saliency pairs, and $L_{hete} + L_{homo}$ indicates utilizing the proposed label-guided ranking loss. The best performance is **boldfaced**.

| Loss | DUT-O | | | DUTS | | | HKU-IS | | | ECSSD | | | PASCALS | | | SOD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ |
| $L_{hete}$ | 0.804 | 0.063 | 0.813 | 0.857 | 0.052 | 0.840 | 0.918 | 0.039 | 0.891 | 0.931 | 0.044 | 0.904 | 0.858 | 0.071 | 0.839 | 0.856 | 0.102 | 0.803 |
| $L_{hete} + \tilde{L}_{co-sal}$ | 0.805 | 0.064 | 0.811 | 0.859 | 0.052 | 0.839 | 0.919 | 0.040 | 0.888 | 0.932 | 0.043 | 0.903 | 0.858 | 0.072 | 0.836 | 0.858 | **0.097** | **0.808** |
| $L_{hete} + \tilde{L}_{co-bg}$ | 0.813 | **0.055** | 0.833 | 0.868 | **0.045** | 0.858 | 0.925 | **0.035** | **0.907** | 0.937 | 0.042 | 0.916 | **0.864** | **0.069** | **0.851** | 0.859 | 0.108 | 0.792 |
| $L_{hete} + L_{homo}$ | **0.817** | 0.056 | **0.834** | **0.870** | **0.045** | **0.859** | **0.926** | **0.035** | **0.907** | **0.939** | **0.040** | **0.917** | 0.863 | 0.070 | 0.849 | **0.863** | 0.104 | 0.804 |

**Table 2.** Comparison of different loss functions on six datasets. Our loss achieves the best performance under the same setting. The best performance is **boldfaced.**

| Loss | DUT-O | | | DUTS | | | HKU-IS | | | ECCSD | | | PASCALS | | | SOD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ |
| MarginRank | 0.779 | 0.059 | 0.808 | 0.822 | 0.049 | 0.832 | 0.896 | 0.037 | 0.883 | 0.918 | 0.042 | 0.896 | 0.838 | 0.071 | 0.828 | 0.835 | 0.105 | 0.782 |
| MAE | 0.802 | **0.052** | 0.823 | 0.853 | 0.052 | 0.842 | 0.920 | **0.032** | 0.899 | 0.934 | **0.040** | 0.905 | 0.861 | **0.065** | 0.842 | 0.848 | 0.112 | 0.761 |
| MSE | 0.804 | 0.062 | 0.829 | 0.860 | 0.051 | 0.855 | 0.921 | 0.044 | 0.904 | 0.933 | 0.053 | 0.909 | 0.861 | 0.077 | **0.852** | 0.852 | 0.118 | 0.777 |
| BCE | 0.804 | 0.058 | 0.829 | 0.861 | 0.048 | 0.856 | 0.923 | 0.039 | 0.906 | 0.937 | 0.046 | 0.915 | 0.857 | 0.076 | 0.846 | 0.855 | 0.114 | 0.781 |
| Ours | **0.817** | 0.056 | **0.834** | **0.870** | **0.045** | **0.859** | **0.926** | 0.035 | **0.907** | **0.939** | **0.040** | **0.917** | **0.864** | 0.070 | 0.849 | **0.863** | **0.104** | **0.804** |

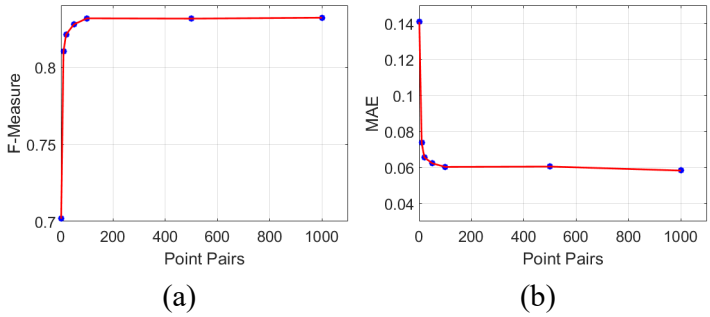



**Figure 6.** Ablation experiments on the amounts of sampled point pairs. Specifically, we sample {1, 10, 20 , 50, 100, 200, 500, 1000} point pairs per image. (**a**) F-measure; (**b**) MAE.

#### 4.1.2. Comparison with State-of-the-Arts

We compare our method against other 12 state-of-the-art algorithms, i.e., KSR [60], DCL [12], UCF [8], GBR [31], SRM [61], Amulet [62], DSS [7], DGRL [9], BdMPM [54], PiCANet [10], RAS [33], and MLMSNet [32] on six public datasets.

**Quantitative and qualitative results:** Table 3 shows the quantitative comparison in terms of maximum $F_\beta$-Measure, MAE, and structure-measure. For a fair comparison, we also use VGG-16 and Resnet-50 as our backbone model. Since DSS [7], DCL [12], and PiCANet [10] use CRF [63] to refine their predictions, we use CRF to refine saliency maps as well. The PR curves on six datasets are given in Figure 7. As shown in Table 3, our models achieve competitive or better performance when compared to other state-of-the-art methods. In Figure 8, we further show qualitative comparisons of our method against other methods. One can observe that our method can predict more accurate saliency maps that coincide with the ground truth masks. More specifically, our method can tell apart two salient object instances with similar appearances (e.g., the 5th and 6th rows) and preserve the structural consistency of a salient object (e.g., the 3th and 7th rows ). However, other methods suffer from the two problems (i.e., interclass indistinction and intraclass inconsistency), which holds our basic idea.

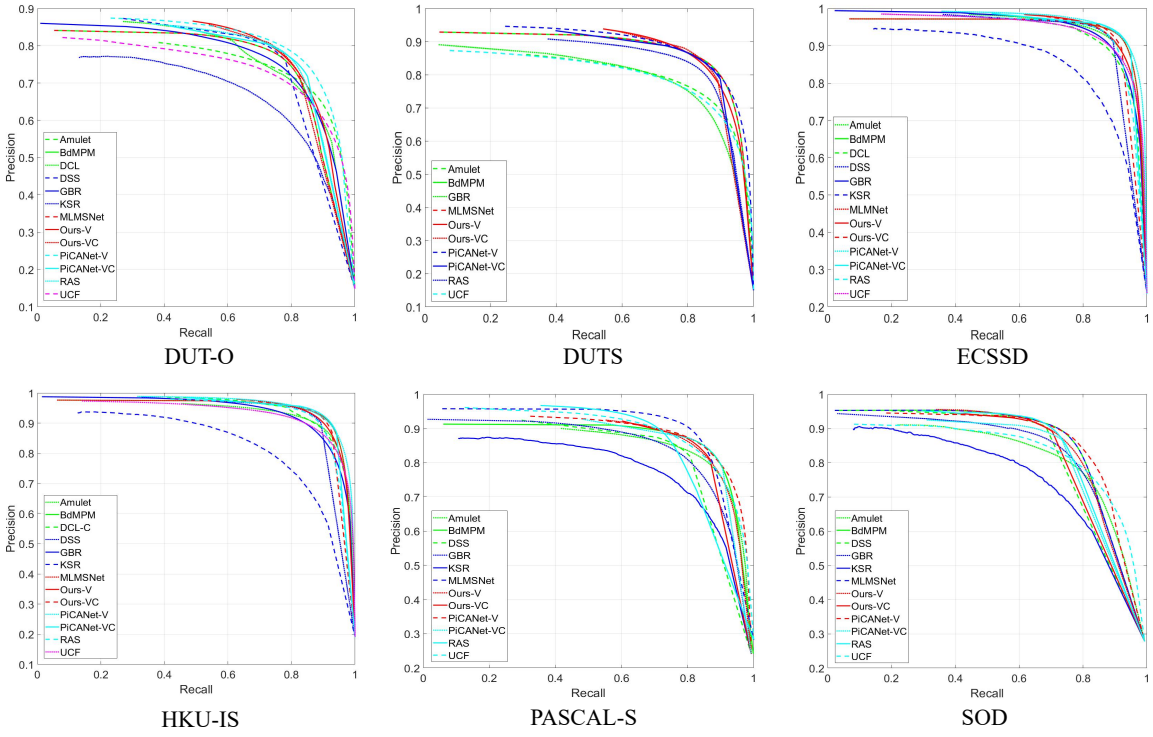



**Figure 7.** Comparison on six datasets in terms of the PR curve.

**Table 3.** Comparison of different salient object detection (SOD) methods on six datasets. The best performance is **boldfaced** and the second best is underlined. † means that the method is postprocessed by Conditional Random Field (CRF).

| Methods | DUT-O | | | DUTS | | | HKU-IS | | | ECSSD | | | PASCALS | | | SOD | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ | $F_\beta \uparrow$ | $MAE \downarrow$ | $S \uparrow$ |
| **VGG-16 backbone** | | | | | | | | | | | | | | | | | | |
| KSR | 0.678 | 0.131 | 0.708 | – | – | – | 0.792 | 0.120 | 0.729 | 0.829 | 0.132 | 0.763 | 0.762 | 0.154 | 0.716 | 0.741 | 0.197 | 0.633 |
| DCL † | 0.757 | 0.086 | 0.771 | – | – | – | 0.907 | 0.055 | 0.877 | 0.901 | 0.075 | 0.868 | – | – | – | – | – | – |
| UCF | 0.730 | 0.120 | 0.760 | 0.772 | 0.112 | 0.777 | 0.888 | 0.062 | 0.875 | 0.903 | 0.069 | 0.883 | 0.848 | 0.115 | 0.806 | 0.805 | 0.148 | 0.763 |
| GBR | 0.759 | 0.073 | 0.806 | 0.774 | 0.073 | 0.798 | 0.891 | 0.057 | 0.877 | 0.909 | 0.066 | 0.887 | 0.821 | 0.107 | 0.807 | 0.819 | 0.130 | 0.762 |
| Amulet | 0.743 | 0.098 | 0.781 | 0.777 | 0.085 | 0.796 | 0.897 | 0.051 | 0.886 | 0.915 | 0.059 | 0.894 | 0.828 | 0.100 | 0.818 | 0.795 | 0.144 | 0.755 |
| DSS † | 0.781 | 0.063 | 0.790 | – | – | – | 0.916 | 0.040 | 0.878 | 0.921 | 0.052 | 0.882 | 0.831 | 0.093 | 0.799 | 0.843 | 0.122 | 0.746 |
| BdMPM | 0.774 | 0.064 | 0.809 | 0.851 | 0.049 | 0.851 | 0.921 | 0.039 | 0.907 | 0.928 | 0.045 | 0.911 | 0.855 | 0.074 | 0.844 | 0.852 | 0.106 | 0.790 |
| PiCANet | 0.794 | 0.068 | 0.826 | 0.851 | 0.054 | 0.851 | 0.921 | 0.042 | 0.906 | 0.931 | 0.047 | 0.914 | 0.856 | 0.078 | 0.848 | 0.850 | 0.101 | 0.793 |
| PiCANet † | 0.784 | 0.059 | 0.815 | 0.850 | 0.045 | 0.839 | 0.925 | 0.031 | 0.904 | 0.933 | 0.036 | 0.910 | 0.856 | 0.069 | 0.841 | 0.834 | 0.095 | 0.776 |
| RAS | 0.786 | 0.062 | 0.814 | 0.831 | 0.059 | 0.828 | 0.913 | 0.045 | 0.887 | 0.921 | 0.056 | 0.893 | 0.850 | 0.101 | 0.735 | 0.847 | 0.123 | 0.767 |
| MLMSNet | 0.774 | 0.064 | 0.809 | 0.851 | 0.049 | 0.851 | 0.921 | 0.039 | 0.907 | 0.928 | 0.045 | 0.911 | **0.881** | 0.074 | 0.794 | 0.852 | 0.106 | 0.790 |
| Ours | 0.787 | 0.063 | 0.808 | 0.853 | 0.049 | 0.845 | 0.917 | 0.039 | 0.897 | 0.922 | 0.052 | 0.895 | 0.850 | 0.076 | 0.836 | 0.841 | 0.120 | 0.771 |
| Ours † | 0.789 | 0.056 | 0.807 | 0.858 | 0.043 | 0.843 | 0.924 | 0.032 | 0.899 | 0.928 | 0.046 | 0.893 | 0.855 | 0.071 | 0.831 | 0.840 | 0.117 | 0.762 |
| **Resnet-50 backbone** | | | | | | | | | | | | | | | | | | |
| SRM | 0.769 | 0.069 | 0.798 | 0.827 | 0.059 | 0.824 | 0.906 | 0.046 | 0.887 | 0.917 | 0.054 | 0.895 | 0.838 | 0.084 | 0.834 | 0.840 | 0.126 | 0.745 |
| DGRL | 0.779 | 0.063 | 0.810 | 0.834 | 0.051 | 0.836 | 0.914 | 0.037 | 0.897 | 0.925 | 0.043 | 0.906 | 0.848 | 0.074 | 0.839 | 0.844 | 0.104 | 0.777 |
| PiCANet | 0.803 | 0.065 | 0.832 | 0.860 | 0.050 | 0.859 | 0.919 | 0.043 | 0.904 | 0.935 | 0.047 | 0.917 | 0.857 | 0.075 | **0.854** | 0.853 | 0.103 | 0.793 |
| PiCANet † | 0.804 | 0.054 | 0.826 | 0.866 | 0.040 | 0.849 | 0.927 | 0.031 | 0.905 | 0.940 | 0.035 | 0.916 | 0.859 | **0.064** | 0.846 | 0.851 | **0.094** | 0.780 |
| Ours | 0.809 | 0.057 | 0.827 | 0.866 | 0.046 | 0.856 | 0.925 | 0.036 | 0.905 | 0.936 | 0.042 | 0.913 | 0.862 | 0.071 | 0.846 | 0.853 | 0.103 | 0.791 |
| Ours † | 0.810 | 0.051 | 0.827 | 0.871 | 0.040 | 0.855 | 0.932 | 0.029 | 0.908 | 0.942 | 0.035 | 0.913 | 0.862 | 0.066 | 0.841 | 0.857 | 0.106 | 0.783 |
| **DLA-60 backbone** | | | | | | | | | | | | | | | | | | |
| Ours | **0.817** | 0.056 | 0.834 | 0.870 | 0.045 | 0.859 | 0.926 | 0.035 | 0.907 | 0.939 | 0.040 | 0.917 | 0.863 | 0.070 | 0.849 | 0.863 | 0.104 | **0.804** |
| Ours † | **0.817** | **0.050** | **0.835** | **0.874** | **0.038** | **0.860** | **0.933** | **0.028** | **0.911** | **0.945** | **0.033** | **0.918** | 0.863 | 0.065 | 0.845 | **0.865** | 0.100 | 0.794 |

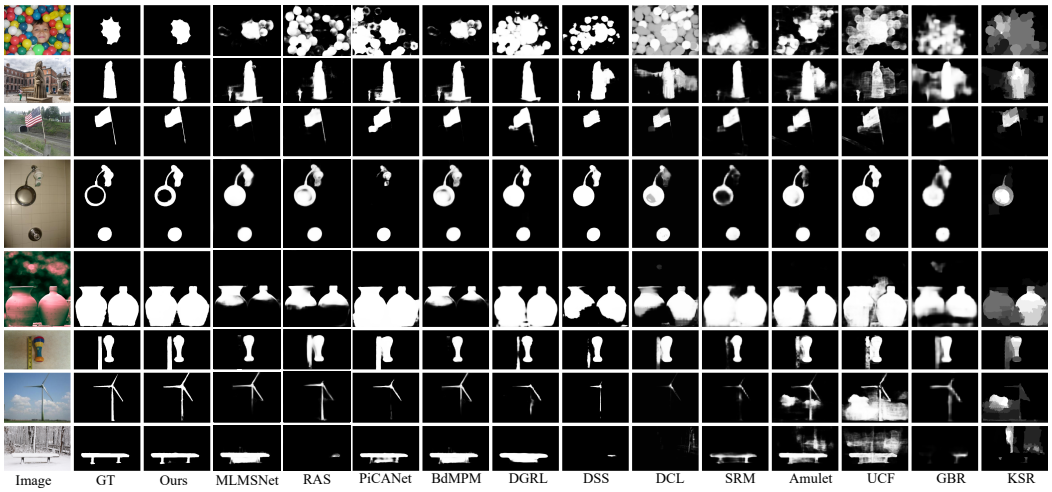



**Figure 8.** Qualitative comparison of saliency maps. We compare our model with MLMSNet [32], RAS [33], PiCANet [10], BdMPM [54], DGRL [9], DSS [7], DCL [12], SRM [61], Amulet [62], UCF [8], GBR [31], and KSR [60].

### *4.2. Monocular Depth Prediction*

Our MDP model, based on a ResNet101-based encoder-decoder architecture [15], is trained on the HRWSI dataset [14]. In order to evaluate the performance of the MDP module, we compare it against other methods on six RGB-D datasets, including NYUDv2 [64], Ibims [65], TUM [66], KITTI [67], Sintel [68], and DIODE [69]. Note that these datasets were unseen during training. The NYUDv2 dataset, consisting of 654 indoor RGB-D image pairs, is captured by a Kinect depth sensor in indoor scenes. Ibims is a high-quality RGB-D dataset specially designed for testing MDP methods. It contains 100 indoor RGB-D pairs with a deficient noise level, sharp depth transitions, no occlusions, and high depth ranges. TUM is also an indoor RGB-D dataset, which mainly focuses on moving people. Particularly, there are 11 image sequences with 1815 images for testing. In addition to testing on indoor scenes, we also test methods on outdoor datasets. KITTI is the widely used outdoor dataset used for testing MDP methods. In our experiments, we use the split (697 images) provided by Eigen et al. [17] for evaluation. Moreover, we also evaluate methods on Sintel, a synthetic RGB-D dataset with accurate ground truth depth maps. This dataset is comprised of 1064 images derived from an open-source 3D animated film. Additionally, we test MDP methods on the official test set (771 images) of DIODE, which contains both indoor and outdoor scenes.

**Quantitative and qualitative results:** In Table 4, we compare our MDP model with 7 state-of-the-art methods, including DIW [41], DL [5], RW [15], MD [42], Y3D [44], MC [43], and HRWSI [14]. For the definition of the metrics, please refer to [14]. As shown in Table 4, one can observe that our MDP model outperforms other methods, exhibiting good generalization performance. Despite being trained with less data when compared to DIW, MD, Y3D, and MC, our MDP model still exhibits better generalization performance. The reasons may lie in the quality of training data as well as the structure-guided ranking loss. The HRWSI dataset has diverse training samples with high-quality ground truth depth data. The structure-guided ranking loss guides the model toward the regions that better characterize the structure of the image.

We further show some qualitative comparisons in Figure 9. Our MDP model can get more accurate predictions, which has more consistent depth with sharper depth discontinuities.

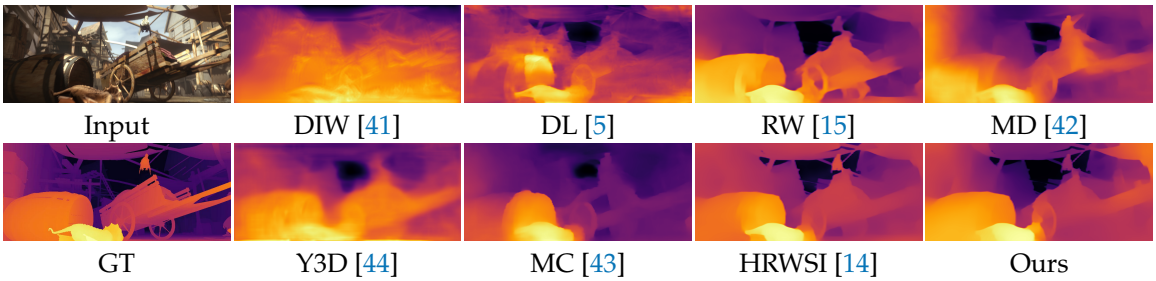
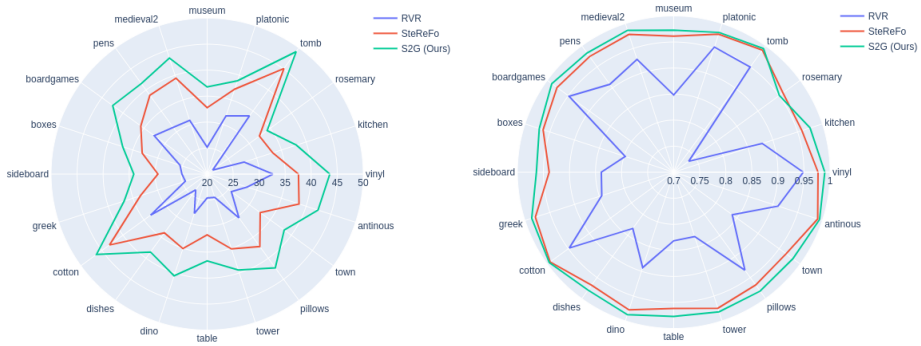



**Figure 9.** Qualitative results of different MDP methods.

### *4.3. Shallow DoF*

We conduct experiments on 4D Light Field (4DLF) dataset [70], that consists of 20 photorealistic scenes. For each scene, it provides an all-in-focus image, a disparity map, as well as $9 \times 9$ light fields. We use the light field refocusing method [71] to synthesize the DoF images as ground truth. In particular, each image is refocused at five disparity planes, i.e., $-1.5$ px, $-0.75$ px, 0 px, 0.75 px, and 1.5 px. To verify the effectiveness of our DoF rendering module, we implement two DoF rendering methods (i.e., RVR [18] and SteReFo [3]).

Table 5 reports the quantitative results of these methods in terms of PSNR and SSIM, and Figure 10 summarizes all scores computed on the three methods and 20 scenes. One can observe that our proposed method outperforms other methods by a large margin. In Figure 11, we further show the qualitative results of these methods. Given an all-in-focus image, these DoF images are generated by focusing on the plane of disparity zero and blurring out the rest of the given image. As shown in Figure 11, the left red box highlights the details of the focused area, while the right one shows the blurred background. One can observe: (i) RVR is prone to generating halo artifacts along the boundaries; (ii) SteReFo tends to generate blurred pixels at the focused area; (iii) Our method, by contrast, can keep the focused area clear and blur out the rest of the given image. To find out why RVR and SteReFo fail to generate promising predictions, we revisit their definitions and implementations. RVR is the iterative rendering without weight normalization that leads to halo artifacts along the boundaries. For SteReFo, although it can get a smooth transition of the refocus plane by assigning each pixel to multiple depth layers, such operation causes blur at the focused plane at the same time.




(**a**) PSNR (**b**) SSIM

**Figure 10.** The radar charts summarize the scores on different scenes.

**Table 4.** Cross-dataset evaluation on six RGB-D datasets. The best performance is **boldfaced.**

| Methods | Ibims | | | TUM | | | Sintel | | | NYUDv2 | | | KITTI | | | DIODE | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ | *Ord* ↓ | $\delta >$ **1.25** ↓ | *Rel* ↓ |
| DIW [41] | 46.97 | 39.30 | 0.232 | 39.62 | 37.42 | 0.270 | 43.50 | 56.21 | 0.405 | 37.33 | 36.85 | 0.210 | 29.92 | 51.45 | 0.306 | 45.40 | 42.25 | 0.307 |
| DL [5] | 40.92 | 34.75 | 0.211 | 31.62 | 25.26 | 0.205 | 36.63 | 48.20 | 0.407 | 31.67 | 32.71 | 0.196 | 25.40 | 45.32 | 0.271 | 43.77 | 40.04 | 0.311 |
| RW [15] | 33.13 | 30.46 | 0.220 | 30.07 | 25.16 | 0.200 | 31.12 | 45.46 | 0.410 | 26.76 | 28.86 | 0.178 | 16.40 | 31.32 | 0.207 | 39.42 | 38.27 | 0.320 |
| MD [42] | 36.82 | 31.31 | 0.200 | 31.88 | 26.86 | 0.226 | 38.07 | 53.56 | 0.422 | 27.84 | 29.69 | 0.182 | 17.50 | 36.32 | 0.238 | 39.07 | 39.03 | 0.323 |
| Y3D [44] | 31.73 | 26.02 | 0.174 | 30.37 | 26.36 | 0.230 | 33.88 | 47.50 | 0.329 | 26.39 | 23.13 | 0.153 | 15.08 | 30.20 | 0.185 | 35.57 | 36.48 | **0.276** |
| MC [43] | 31.30 | 21.53 | 0.152 | 26.22 | 26.06 | 0.204 | 37.49 | 44.85 | 0.476 | 25.48 | 23.70 | 0.159 | 22.46 | 48.02 | 0.280 | 40.85 | 39.29 | 0.337 |
| HRWSI [14] | 27.23 | 23.09 | 0.170 | **25.67** | **19.41** | **0.194** | 30.70 | 44.84 | 0.402 | 23.21 | 23.50 | 0.157 | 14.01 | 25.40 | 0.179 | 33.11 | 34.44 | 0.301 |
| Ours | **25.05** | **20.40** | **0.156** | 25.93 | 20.11 | 0.199 | **29.29** | **44.15** | **0.390** | **22.46** | **22.45** | **0.152** | **13.70** | **24.86** | **0.178** | **32.37** | **33.96** | 0.310 |

**Table 5.** Quantitative results of different DoF rendering methods on the 4DLF dataset [70]. The best performance is **boldfaced.**

| Focused Disparity | −1.5 | | −0.75 | | 0 | | 0.75 | | 1.5 | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** | **PSNR** | **SSIM** |
| RVR [18] | 25.82 | 0.85 | 27.69 | 0.88 | 29.46 | 0.90 | 29.14 | 0.90 | 27.94 | 0.87 | 28.01 | 0.88 |
| SteReFo [3] | 33.91 | 0.97 | 35.67 | 0.97 | 37.24 | 0.97 | 36.78 | 0.97 | 35.34 | 0.97 | 35.79 | 0.97 |
| S2G (Ours) | **39.56** | **0.98** | **40.51** | **0.98** | **39.47** | **0.98** | **39.71** | **0.98** | **40.87** | **0.98** | **40.03** | **0.98** |

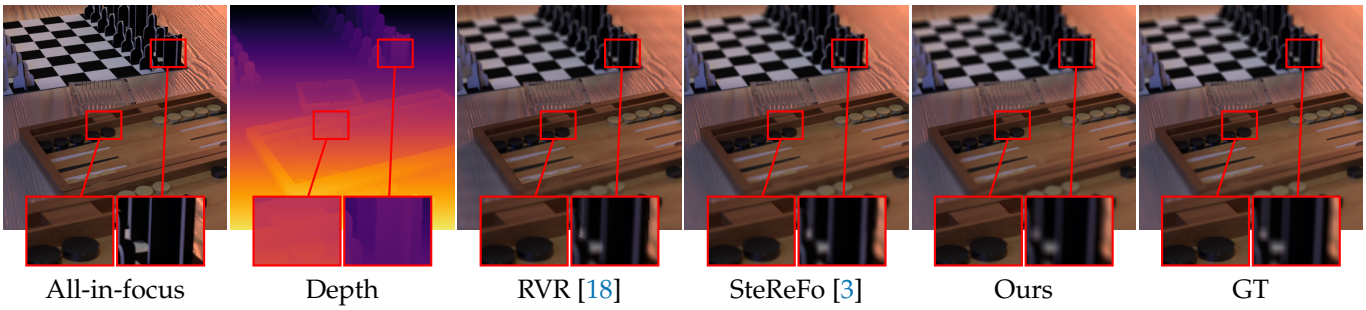



**Figure 11.** Qualitative results of different DoF rendering methods on the 4DLF dataset. Best viewed zoomed in on-screen.

We also conduct experiments on the NJU2K [72] dataset, split into a validation set of 121 images and a test set of 364 images. We first choose the size of the blur kernel according to the performance on the validation set and then test the compared methods on the test set. Table 6 reports the PSNR and SSIM metrics of different rendering methods. The *Ours (w/o depth)* indicates a variant of our method, which blurs the background according to the salient object masks. The results imply the importance of our MDP module. Note that we did not report the performance of our method without the SOD module because our method first needs to know where to focus. The SOD module, detecting salient objects in the image, is used to determine the focal plane. In addition to the quantitative results, we also show the qualitative comparisons in Figure 12 and the visual results of different components in Figure 13. One can see: (i) RVR produces images with strong artifacts around the boundaries; (ii) SteReFo synthesizes reasonable shallow DoF images, but it tends to blur the in-focus objects; (iii) The generalization of deep learning methods is limited. For example, PyNet sometimes focuses on the background by blurring the foreground objects as it is trained on EBB [50] with no knowledge of the focal plane on NJU2K; Instead, DL can generate plausible shallow DoF images due to the use of our SOD module. (iv) The transition of boundaries is too sharp in *Ours (w/o depth)*; (v) Our predictions, by contrast, are clearer at the refocused plane and are more accurate around the boundaries. We also report the running time of our system on the NJU2K dataset. We ran our system with a NVIDIA GTX 1080Ti GPU and an E5-2650 V4 CPU for measuring the running time. The average time consuming for each image is listed as follows. The SOD module takes 0.018 s, the MDP module takes 0.121 s, and the DOF rendering module takes 0.141 s. Therefore, the whole system takes 0.28 s for rendering a shallow DoF image.

**Table 6.** Quantitative results of different rendering methods on the NJU2K [72] dataset.

| | RVR [18] | SteReFo [3] | DL [5] | PyNet [50] | Ours (w/o Depth) | Ours |
|---|---|---|---|---|---|---|
| PSNR | 30.82 | 32.06 | 29.56 | 24.69 | 31.40 | **33.04** |
| SSIM | 0.89 | 0.90 | 0.86 | 0.84 | 0.89 | **0.91** |

To further verify the effectiveness of the overall framework, we compare with other state-of-the-art methods on the EBB [50] dataset. This dataset provides 4694 shallow/wide DoF image pairs captured by a Canon 7D DSLR with 50 mm f/1.8 lenses. We create a training set of 3694 images for training deep learning models, a validation set of 500 images for model selection, and a test set of 500 images for model evaluation. Table 7 shows the quantitative results of different methods. One can observe that our method achieves the best performance in terms of PSNR. PyNet, trained to map the narrow-aperture images into shallow DoF photos in an end-to-end manner, achieves the best SSIM result. In addition to the quantitative comparisons, we also present the qualitative results in Figure 14. In general, our method can synthesize realistic shallow DoF images on the EBB dataset.

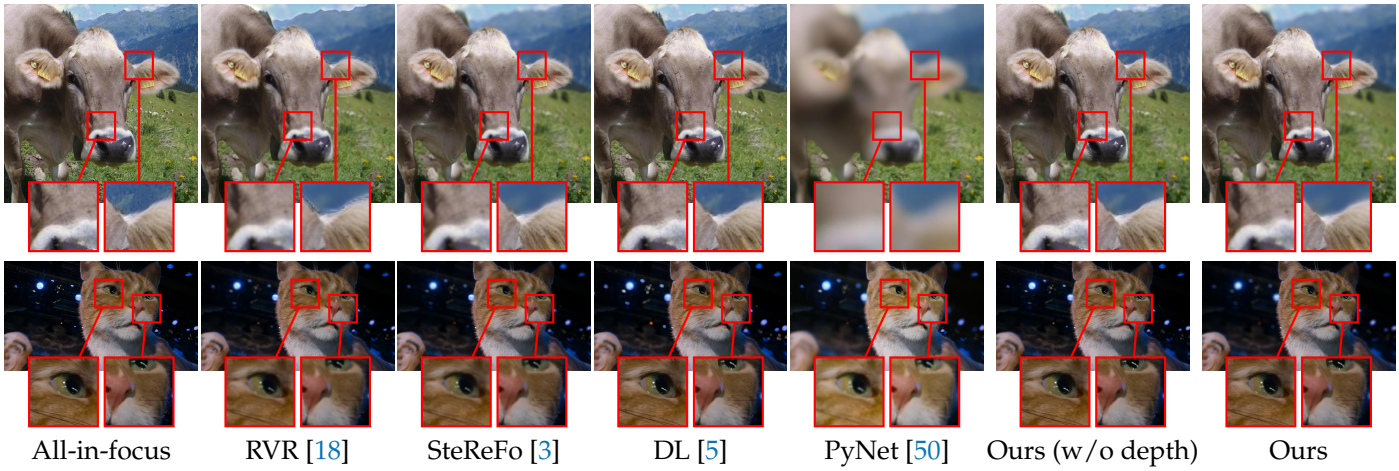



**Figure 12.** Qualitative results on the NJU2K dataset. Best viewed zoomed in on-screen.

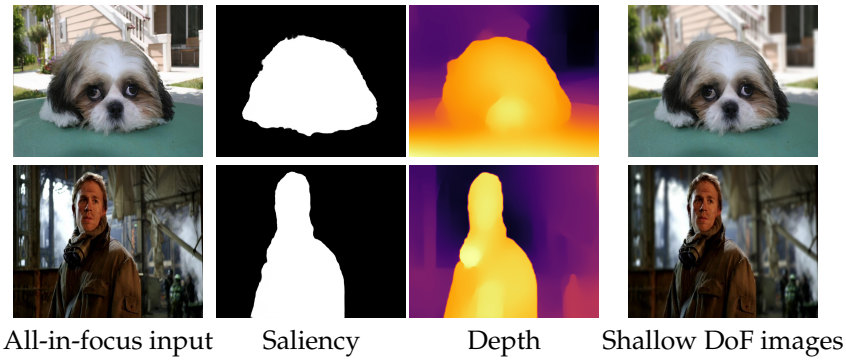



**Figure 13.** Some visual examples of our predicted saliency maps, depth maps, and shallow DoF images.

**Table 7.** Quantitative results of different methods on the EBB [50] dataset.

| | RVR [18] | SteReFo [3] | DL [5] | PyNet [50] | Ours |
|---|---|---|---|---|---|
| PSNR | 22.82 | 23.54 | 23.63 | 23.37 | **23.79** |
| SSIM | 0.82 | 0.85 | 0.86 | **0.87** | 0.86 |

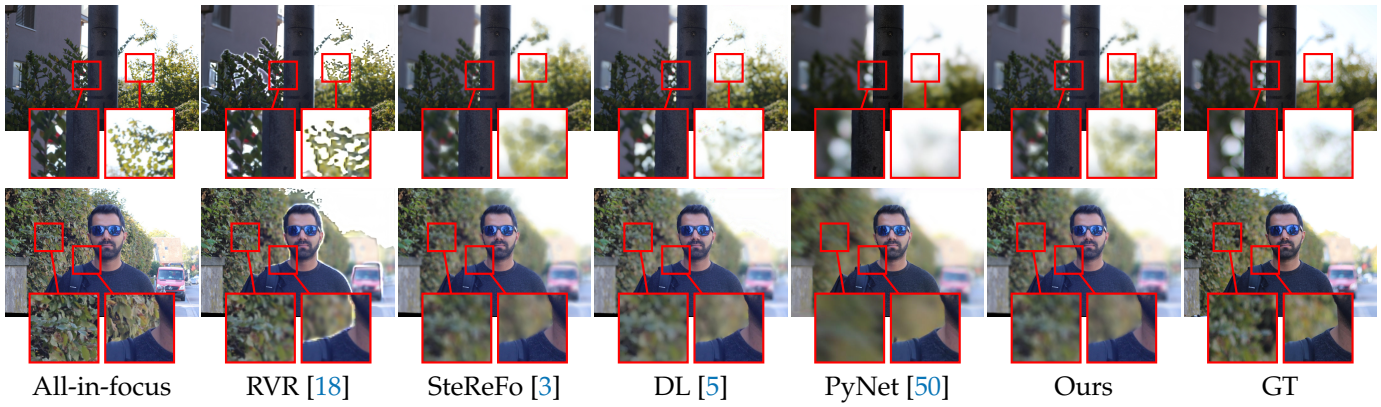



**Figure 14.** Qualitative results on the EBB dataset. Best viewed zoomed in on-screen.

**Ablation studies:** To study the impact of the SOD module and the MDP module, we further conduct ablation studies on the NJU2K dataset. In particular, we replace our SOD module with PICANet [10], DSS [7] and a randomly selected focal plane, respectively. From

Table 8, one can observe that the performance of the saliency-based methods is close. It makes sense because our current rendering method does not rely too much on the accuracy of the SOD module. We use the SOD module to guide the focal plane. Nevertheless, that does not mean the accuracy of the SOD module is not important. If its performance is too poor, the rendering results will certainly be affected (see the *Random* in Table 8). Furthermore, we replace our MDP module with other two methods (i.e., MD [42] and DL [5]) to study the impact of the MDP module. From Table 8, one can observe that with the increase of the MDP module, the rendering performance is also improved. This indicates that the more accurate the MDP model, the better the rendering results. Figure 15 demonstrates the qualitative comparisons.

**Table 8.** Ablation study on the impact of different components.

| | Random | PICANet [10] | DSS [7] | MD [42] | DL [5] | Ours |
|---|---|---|---|---|---|---|
| PSNR | 29.77 | **33.04** | 32.95 | 31.84 | 32.23 | **33.04** |
| SSIM | 0.85 | **0.91** | **0.91** | 0.89 | 0.90 | **0.91** |

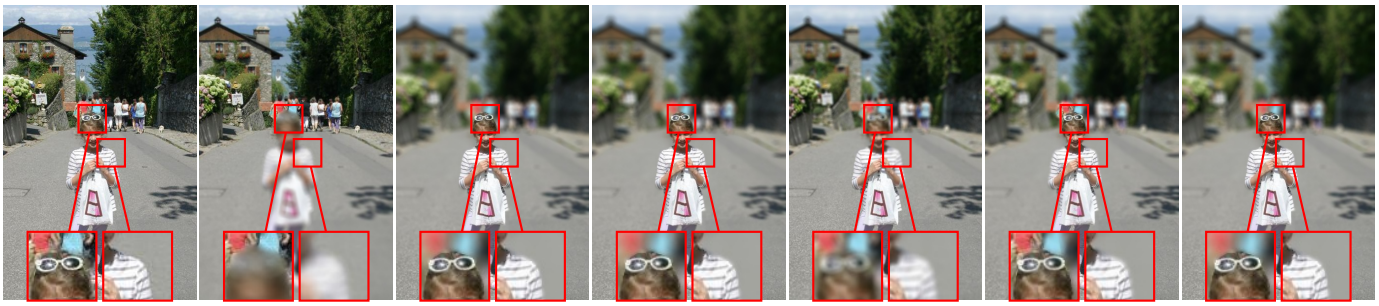



**Figure 15.** Impact of the SOD module and the MDP module. Best viewed zoomed in on-screen.

## 5. Conclusions

This paper presents an automatic shallow DoF system consisting of a SOD module, an MDP module, and a DoF rendering module. The SOD module is used to determine the refocused depth, and the MDP module is used to control the degree of blur. We show that explicit modeling of the pairwise relations benefits both SOD and MDP. In particular, we propose a label-guided ranking loss for SOD. The loss comprises a heterogeneous ranking term that improves the interclass distinction and a homogeneous ranking term that enhances the intraclass consistency. To synthesize realistic shallow DoF images, we further propose an S2G method. By combing the SOD module, the MDP module, and the DoF rendering module, our system can generate realistic shallow DoF images. Besides, our method, capable of adjusting the focal plane and blur degree, is flexible in real-world applications. By changing the point spread function and the size of the blur kernel, our method can control the shape and visual quality of the defocused area. Although our system is able to generate realistic shallow DoF from an arbitrary image, it depends too much on the quality of the predicted depth. In the future, we plan to further improve the quality of monocular depth prediction.

**Author Contributions:** K.X. principal investigator, designed the project, data acquisition, performed analysis, and wrote the manuscript. J.P. and C.Z. contributed to data acquisition and diagnosis of the cases. H.L. revised the paper. Z.C. supervised the project and approved final submission. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China (Grant No. U1913602 and 61876211).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets analyzed in this study are available through their respective respository pages. The salient object detection datasets can be found here: (SOD [56] at https://www.elderlab.yorku.ca/resources/salient-objects-dataset-sod/), (ECSSD [57] at http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html), (PASCAL-S [58] at http://cbs.ic.gatech.edu/salobj/), (HKU-IS [24] at https://i.cs.hku.hk/~yzyu/research/deep_saliency.html), (DUT-OMRON [59] at http://saliencydetection.net/dut-omron/), and (DUTS [55] at http://saliencydetection.net/duts/). The monoculuar depth predition datasets can be found here: (HRWSI [14] at https://kexianhust.github.io/Structure-Guided-Ranking-Loss/), (NYUDv2 [64] at https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html), (Ibims [65] at https://www.bgu.tum.de/lmf/ibims1/), (TUM [66] at https://vision.in.tum.de/data/datasets/rgbd-dataset/download), (KITTI [67] at http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction), (Sintel [68] at http://sintel.is.tue.mpg.de/downloads), and (DIODE [69] at https://diode-dataset.org/). The DoF related datasets can be found here: (4DLF [70] at https://lightfield-analysis.uni-konstanz.de/), (NJU2K [72] at https://drive.google.com/file/d/1R1O2dWr6HqpTOiDn6hZxUWTesOSJteQo/view), and (EBB [50] at http://people.ee.ethz.ch/~ihnatova/pynet-bokeh.html), all accessed on 5 March 2021.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Xu, X.; Sun, D.; Liu, S.; Ren, W.; Zhang, Y.J.; Yang, M.H.; Sun, J. Rendering portraitures from monocular camera and beyond. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 35–50.
2. Wadhwa, N.; Garg, R.; Jacobs, D.E.; Feldman, B.E.; Kanazawa, N.; Carroll, R.; Movshovitz-Attias, Y.; Barron, J.T.; Pritch, Y.; Levoy, M. Synthetic depth-of-field with a single-camera mobile phone. *ACM Trans. Gr.* **2018**, *37*, 1–13. [CrossRef]
3. Busam, B.; Hog, M.; McDonagh, S.; Slabaugh, G. SteReFo: Efficient Image Refocusing with Stereo Vision. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27 October–2 November 2019.
4. Barron, J.T.; Adams, A.; Shih, Y.; Hernández, C. Fast bilateral-space stereo for synthetic defocus. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4466–4474.
5. Wang, L.; Shen, X.; Zhang, J.; Wang, O.; Hsieh, C.Y.; Kong, S.; Lu, H. Deeplens: Shallow depth of field from a single image. *ACM Trans. Gr.* **2018**, *37*, 1–11. [CrossRef]
6. Shen, X.; Hertzmann, A.; Jia, J.; Paris, S.; Price, B.; Shechtman, E.; Sachs, I. Automatic portrait segmentation for image stylization. In *Comput Graph Forum*; Wiley Online Library: Hoboken, NJ, USA, 2016, Volume 35; pp. 93–102.
7. Hou, Q.; Cheng, M.M.; Hu, X.; Borji, A.; Tu, Z.; Torr, P.H.S. Deeply Supervised Salient Object Detection With Short Connections. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3203–3212.
8. Zhang, P.; Wang, D.; Lu, H.; Wang, H.; Yin, B. Learning Uncertain Convolutional Features for Accurate Saliency Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 212–221.
9. Wang, T.; Zhang, L.; Wang, S.; Lu, H.; Yang, G.; Ruan, X.; Borji, A. Detect Globally, Refine Locally: A Novel Approach to Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 July 2018; pp. 3127–3135.
10. Liu, N.; Han, J.; Yang, M.H. PiCANet: Learning Pixel-Wise Contextual Attention for Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3089–3098.
11. Ren, G.; Dai, T.; Barmpoutis, P.; Stathaki, T. Salient Object Detection Combining a Self-Attention Module and a Feature Pyramid Network. *Electronics* **2020**, *9*, 1702. [CrossRef]
12. Li, G.; Yu, Y. Deep Contrast Learning for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 478–487.
13. Liu, N.; Han, J. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Trans. Image Proc.* **2018**, *27*, 3264–3274. [CrossRef]
14. Xian, K.; Zhang, J.; Wang, O.; Mai, L.; Lin, Z.; Cao, Z. Structure-Guided Ranking Loss for Single Image Depth Prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 611–620.
15. Xian, K.; Shen, C.; Cao, Z.; Lu, H.; Xiao, Y.; Li, R.; Luo, Z. Monocular Relative Depth Perception With Web Stereo Data Supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 311–320.
16. Laina, I.; Rupprecht, C.; Belagiannis, V.; Tombari, F.; Navab, N. Deeper Depth Prediction with Fully Convolutional Residual Networks. In Proceedings of the IEEE International Conference on 3D Vision, Stanford, CA, USA, 25–28 October 2016; pp. 239–248.

17. Eigen, D.; Puhrsch, C.; Fergus, R. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27, pp. 2366–2374.
18. Zhang, X.; Matzen, K.; Nguyen, V.; Yao, D.; Zhang, K.; Ng, R. Synthetic defocus and look-ahead autofocus for casual videography. *ACM Trans. Gr.* **2019**, *38*, 1–16.
19. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [CrossRef]
20. Klein, D.A.; Frintrop, S. Center-surround divergence of feature statistics for salient object detection. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2214–2219.
21. Zhu, W.; Liang, S.; Wei, Y.; Sun, J. Saliency Optimization from Robust Background Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2814–2821.
22. Wei, Y.; Wen, F.; Zhu, W.; Sun, J. Geodesic saliency using background priors. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 29–42.
23. Borji, A.; Cheng, M.M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Trans. Image Proc.* **2015**, *24*, 5706–5722. [CrossRef] [PubMed]
24. Li, G.; Yu, Y. Visual Saliency Based on Multiscale Deep Features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5455–5463.
25. Liu, N.; Han, J. DHSNet: Deep Hierarchical Saliency Network for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 678–686.
26. Feng, W.; Li, X.; Gao, G.; Chen, X.; Liu, Q. Multi-Scale Global Contrast CNN for Salient Object Detection. *Sensors* **2020**, *20*, 2656. [CrossRef]
27. Cong, R.; Lei, J.; Fu, H.; Cheng, M.; Lin, W.; Huang, Q. Review of Visual Saliency Detection With Comprehensive Information. *IEEE Trans. Circ. Syst. Video Technol.* **2019**, *29*, 2941–2959. [CrossRef]
28. Wang, W.; Shen, J.; Xie, J.; Cheng, M.M.; Ling, H.; Borji, A. Revisiting Video Saliency Prediction in the Deep Learning Era. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 220–237. [CrossRef] [PubMed]
29. Han, J.; Zhang, D.; Cheng, G.; Liu, N.; Xu, D. Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. *IEEE Signal Proc. Mag.* **2018**, *35*, 84–100. [CrossRef]
30. Tang, Y.; Wu, X. Salient Object Detection Using Cascaded Convolutional Neural Networks and Adversarial Learning. *IEEE Trans. Multimed.* **2019**, *21*, 2237–2247. [CrossRef]
31. Tan, X.; Zhu, H.; Shao, Z.; Hou, X.; Hao, Y.; Ma, L. Saliency Detection by Deep Network with Boundary Refinement and Global Context. In Proceedings of the IEEE International Conference on Multimedia and Expo, San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
32. Wu, R.; Feng, M.; Guan, W.; Wang, D.; Lu, H.; Ding, E. A Mutual Learning Method for Salient Object Detection with intertwined Multi-Supervision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8150–8159.
33. Chen, S.; Tan, X.; Wang, B.; Hu, X. Reverse Attention for Salient Object Detection. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 234–250.
34. Liu, F.; Shen, C.; Lin, G.; Reid, I. Learning Depth from Single Monocular Images Using Deep Convolutional Neural Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2024–2039. [CrossRef] [PubMed]
35. Xu, D.; Ouyang, W.; Wang, X.; Sebe, N. PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 675–684.
36. Zhang, Z.; Cui, Z.; Xu, C.; Jie, Z.; Li, X.; Yang, J. Joint task-recursive learning for semantic segmentation and depth estimation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 235–251.
37. Yin, W.; Liu, Y.; Shen, C.; Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 5684–5693.
38. Godard, C.; Mac Aodha, O.; Brostow, G.J. Unsupervised Monocular Depth Estimation with Left-Right Consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 270–279.
39. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1851–1858.
40. Godard, C.; Mac Aodha, O.; Firman, M.; Brostow, G.J. Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3828–3838.
41. Chen, W.; Fu, Z.; Yang, D.; Deng, J. Single-Image Depth Perception in the Wild. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems, NIPS 2016, Barcelona, Spain, 5–10 December 2016; pp. 730–738.
42. Li, Z.; Snavely, N. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2041–2050.
43. Li, Z.; Dekel, T.; Cole, F.; Tucker, R.; Snavely, N.; Liu, C.; Freeman, W.T. Learning the Depths of Moving People by Watching Frozen People. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4521–4530.

44. Chen, W.; Qian, S.; Deng, J. Learning single-image depth from videos using quality assessment networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5604–5613.
45. Huhle, B.; Schairer, T.; Jenke, P.; Straßer, W. Realistic depth blur for images with range data. In *Workshop on Dynamic 3D Imaging*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 84–95.
46. Zhou, T.; Chen, J.X.; Pullen, M. Accurate depth of field simulation in real time. In *Computer Graphics Forum*; Blackwell Publishing Ltd.: Oxford, UK, 2007; Volume 26, pp. 15–23.
47. Yang, Y.; Lin, H.; Yu, Z.; Paris, S.; Yu, J. Virtual DSLR: High quality dynamic depth-of-field synthesis on mobile platforms. *Electron. Imaging* **2016**, *2016*, 1–9. [CrossRef]
48. Purohit, K.; Suin, M.; Kandula, P.; Ambasamudram, R. Depth-Guided Dense Dynamic Filtering Network for Bokeh Effect Rendering. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, Seoul, Korea, 27–28 October 2019; pp. 3417–3426.
49. Dutta, S. Depth-aware Blending of Smoothed Images for Bokeh Effect Generation. *arXiv* **2020**, arXiv:2005.14214.
50. Ignatov, A.; Patel, J.; Timofte, R. Rendering Natural Camera Bokeh Effect with Deep Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 1676–1686.
51. Lee, J.; Lee, S.; Cho, S.; Lee, S. Deep Defocus Map Estimation Using Domain Adaptation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12214–12222.
52. Liu, S.; Pan, J.; Yang, M.H. Learning recursive filters for low-level vision via a hybrid neural network. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 560–576.
53. Yu, F.; Wang, D.; Shelhamer, E.; Darrell, T. Deep Layer Aggregation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2403–2412.
54. Zhang, L.; Dai, J.; Lu, H.; He, Y.; Wang, G. A Bi-Directional Message Passing Model for Salient Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1741–1750.
55. Zhao, R.; Ouyang, W.; Li, H.; Wang, X. Saliency Detection by Multi-Context Deep Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1265–1274.
56. Movahedi, V.; Elder, J.H. Design and perceptual validation of performance measures for salient object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA 13–18 June 2010; pp. 49–56.
57. Yan, Q.; Xu, L.; Shi, J.; Jia, J. Hierarchical Saliency Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 1155–1162.
58. Li, Y.; Hou, X.; Koch, C.; Rehg, J.M.; Yuille, A.L. The Secrets of Salient Object Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 280–287.
59. Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; Yang, M.H. Saliency Detection via Graph-Based Manifold Ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 3166–3173.
60. Wang, T.; Zhang, L.; Lu, H.; Sun, C.; Qi, J. Kernelized subspace ranking for saliency detection. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 450–466.
61. Wang, T.; Borji, A.; Zhang, L.; Zhang, P.; Lu, H. A Stagewise Refinement Model for Detecting Salient Objects in Images. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4019–4028.
62. Zhang, P.; Wang, D.; Lu, H.; Wang, H.; Ruan, X. Amulet: Aggregating Multi-Level Convolutional Features for Salient Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 202–211.
63. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In Proceedings of the Advances in Neural Information Processing Systems 24, Granada, Spain, 12–17 December 2011; pp. 109–117.
64. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 746–760.
65. Koch, T.; Liebel, L.; Fraundorfer, F.; Körner, M. Evaluation of CNN-Based Single-Image Depth Estimation Methods. In Proceedings of the European Conference on Computer Vision Workshop, Munich, Germany, 8–14 September 2018; pp. 331–348.
66. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580.
67. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *The Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
68. Butler, D.J.; Wulff, J.; Stanley, G.B.; Black, M.J. A naturalistic open source movie for optical flow evaluation. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 611–625.
69. Vasiljevic, I.; Kolkin, N.; Zhang, S.; Luo, R.; Wang, H.; Dai, F.Z.; Daniele, A.F.; Mostajabi, M.; Basart, S.; Walter, M.R.; et al. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *arxiv* **2019**, arxiv:1908.00463.
70. Honauer, K.; Johannsen, O.; Kondermann, D.; Goldluecke, B. A dataset and evaluation methodology for depth estimation on 4d light fields. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; pp. 19–34.
71. Wang, Y.; Yang, J.; Guo, Y.; Xiao, C.; An, W. Selective light field refocusing for camera arrays using bokeh rendering and superresolution. *IEEE Signal Proc. Lett.* **2018**, *26*, 204–208. [CrossRef]

72. Ju, R.; Ge, L.; Geng, W.; Ren, T.; Wu, G. Depth saliency based on anisotropic center-surround difference. In Proceedings of the IEEE International Conference on Image Processing, Paris, France, 27–30 October 2014; pp. 1115–1119.