

Deep Learning for Light Field Saliency Detection

Tiantian Wang, Yongri Piao, Xiao Li, Lihe Zhang, Huchuan Lu
Dalian University of Technology, China

ti anti anwang. i ce@gmail . com, yrpi ao@dl ut. edu. cn, l hchuan@dl ut. edu. cn

Abstract

Recent research in 4D saliency detection is limited by the deficiency of a large-scale 4D light field dataset. To address this, we introduce a new dataset to assist the subsequent research in 4D light field saliency detection. To the best of our knowledge, this is to date the largest light field dataset in which the dataset provides 1465 all-focus images with human-labeled ground truth masks and the corresponding focal stacks for every light field image. To verify the effectiveness of the light field data, we first introduce a fusion framework which includes two CNN streams where the focal stacks and all-focus images serve as the input. The focal stack stream utilizes a recurrent attention mechanism to adaptively learn to integrate every slice in the focal stack, which benefits from the extracted features of the good slices. Then it is incorporated with the output map generated by the all-focus stream to make the saliency prediction. In addition, we introduce adversarial examples by adding noise intentionally into images to help train the deep network, which can improve the robustness of the proposed network. The noise is designed by users, which is imperceptible but can fool the CNNs to make the wrong prediction. Extensive experiments show the effectiveness and superiority of the proposed model on the popular evaluation metrics. The proposed method performs favorably compared with the existing 2D, 3D and 4D saliency detection methods on the proposed dataset and existing LFSD light field dataset. The code and results can be found at https://github.com/OIPLab-DUT/ICCV2019_DeepLightfield_Saliency. Moreover, to facilitate research in this field, all images we collected are shared in a ready-to-use manner.

1. Introduction

Salient object detection refers to locate and segment objects that grab human attention most, which has been a fundamental task in computer vision area for a long time.

(a) (b) (c) (d)

Figure 1. All-focus images (a) and focal stacks (b-d) generated by the Lytro Illum camera. Usually a focal stack is a collection of images (slices) and each slice indicates the different focus distance. In every slice, there are some scene points in clear focus and the other points are in blurred defocus.

The existing methods measure saliency based on the 2D [7, 18, 21, 22, 27, 32, 42, 43, 62, 63, 66, 72, 78], 3D [10, 13, 25, 33, 47, 55, 77] or 4D [38, 39, 67, 69] images. A large proportion of works lie in the first category, while only a few belong to the last two. Recently, the performance of saliency detection on 2D images has been dramatically improved, which derives from the rapid progress of deep learning techniques. Usually RGB images serve as the input to the deep networks and hierarchical features are extracted to compute saliency from the local and global perspectives. With the availability of commercial 3D sensors such as Microsoft Kinect [74], depth maps are incorporated into the deep networks on saliency detection. Dependent of RGB features, the additional depth information can describe 3D geometric information and help human in the understanding of contextual information of salient objects. Recently, the light field are becoming popular for the light field cameras can record multiple viewpoints in one single exposure. The handheld light field camera Lytro Illum [46] features a microlens array, which is composed of thousands of tiny lenses and designed to measure light from multiple directions. These information can synthesize different kinds of 2D images, including focal stacks, depth maps and all-focus images through rendering [34] and post-shot refocusing techniques [46]. See Figure 1 for example, every slice in the focal stack shows the varying focus depth levels. Focal stacks are a rich source of 3D shape information and have been used extensively for shape-from-focus

Corresponding author

and shape-from-defocus computations in computer vision [4, 23]. However, 4D data in saliency detection is limited to traditional methods that utilize handcrafted features, such as color, texture, contrast. It has less been explored because of the limited number of light field saliency data. Currently, there only exists one publicly available light field saliency dataset [39] with per-pixel ground truth, which includes 100 all-focus images, a set of focal stacks and the corresponding depth information. Without large-scale data, the scalability of algorithms is less-studied and methods that fully utilize data richness are less likely to be exploited. In light of this, it is of importance to introduce a large-scale dataset to assist the further saliency detection.

After obtaining the light field data, how to effectively incorporate them still needs to be handled. As shown in Figure 1, each focal stack slice respectively indicates that how much of an image region falls into focus, which is controlled by the depth of field. Apparently, those slices play different roles in the final saliency prediction for the reason that the refocused region in one image contains salient objects and some defocused region only represents the background. We prefer to select the salient refocused slices for measuring saliency. In this case, we expect that the proposed method can focus on the 'good' slices, which present relatively clear foreground object and blurred background noise. Hence, inspired by [80], we employ a novel slice-wise attention model, which utilizes the recurrent neural network based on the convolutional features of each slice. This model can learn to adaptively incorporate the feature of each slice to learn more effective feature representation. The output features generated by the proposed model show better performance compared to other fusion structures described in Section 3. To further assist in the training process of the network, we propose to utilize adversarial examples for the saliency detection. Adversarial examples are first introduced by Szegedy *et al.* [58], which verify that existing DNNs are vulnerable to human crafted images. That is, though there is only a small perceptual difference with correctly classified inputs, the CNNs can still be misclassified by the existing state-of-the-art classification networks [26, 57]. Motivated by their work, in this paper we demonstrate that introducing adversarial examples can help train the saliency network to some extent, which can further improve the performance for saliency detection.

Overall, our contributions are summarized below:

- We collect and annotate the first large scale light-field saliency dataset, which contains 1000 training images and 465 test ones. Each image contains one all-focus image labeled with per-pixel ground truth and a focal stack with varied refocused and defocused regions.
- We investigate several CNN fusion frameworks specifically designed for the integration of the light field data

and propose a novel framework in which an attentive recurrent CNN is utilized to integrate all focal slices. By increasing the data diversity via the adversarial examples, the robustness of the framework can be improved among varied input data.

- Compared with the state-of-the-art 2D, 3D and 4D methods, the proposed method performs favorably on the two light field benchmark datasets.

2. Related Works

Here we briefly introduce the related works from two aspects of saliency detection and deep fusion methods.

2.1. Saliency Detection

According to the types of the input data, existing saliency detection methods can be generally summarized into three categories [39]: 1) Saliency on RGB images; 2) Saliency on RGB-D images; 3) Saliency on Light Field images. The previous methods focus on the hand-crafted features which cannot handle the objects with complicated background. The recent deep learning can handle more complicated images, which has made a breakthrough in pixel-wise task, such as saliency, segmentation [14–16, 75] and so forth.

RGB Saliency Detection. The RGB saliency detection approaches mainly adopt handcrafted 2D visual cues, such as color, contrast and background prior [3, 8, 17, 31, 41, 50, 54, 70]. In [28], Itti *et al.* propose the local center-surround contrast of intensity, color and orientation to detect salient targets. In [65], Yang *et al.* compute the similarity of every superpixel [2, 76] with the background ones via graph-based manifold ranking. These methods are based on the idealistic assumption that image boundary regions are mostly background or the color contrast between foreground and background are high. Although they achieve promising performance on certain widely used 2D datasets [6, 30, 45], they are difficult to deal with the challenging cases where the idealistic assumption is not suitable for. To overcome this, CNNs based saliency detection have been proposed [35, 36, 60, 79] and achieve the state-of-the-art performance, which benefits from the advantages of high-level semantic information and low-level structured cues. In [60], Wang *et al.* present two different CNNs to learn the local information as well as the global contrast for saliency detection. Li *et al.* [36] propose an end-to-end deep contrast network consisting of a pixel-level fully convolutional stream and a segment-wise spatial pooling stream.

RGB-D Saliency Detection. Detecting salient objects from RGB-D images [10, 13, 33, 47, 55] attracts lots of interests due to the birth of Microsoft Kinect [5]. Depth prior is widely used in RGB-D saliency detection, which is motivated by that the objects located closest to observers always attract the most attention. Peng *et al.* [47] propose a special-

Figure 2. Different CNN architectures for the 4D inputs.

ized multi-stage RGB-D model by taking account of both depth and appearance cues. Ren *et al.* [55] present a two-stage RGB-D salient object detection framework by exploiting the effectiveness of global priors. In [33], depth prior is integrated into saliency detection models by Lang *et al.* to enhance the saliency prediction. Desingh *et al.* [13] use RGB-D saliency in conjunction with RGB saliency models through non-linear regression to measure saliency value. These methods demonstrate the depth cue plays an important role in determining the salient object. However, these methods may suffer from false positives when salient objects are situated at distant location. Recent works based on CNNs [25, 52, 56] start from different motivation by treating the depth image as the input of CNNs to compute saliency. Qu *et al.* [52] design a convolutional neural network (CNN) on RGB-D images to fuse different low-level saliency cues with hierarchical features. In [25], Han *et al.* utilize CNNs to learn the high-level representation in both RGB view and depth view and propose a multiview CNN (MV-CNN) fusion model to combine both representations. Shigematsu *et al.* [56] propose a deep CNN architecture by exploiting high-level, mid-level and low-level features.

Light Field Saliency Detection. Compared to RGB and RGB-D saliency detection methods, saliency detection for light field is still at the early exploration stage with fewer methods using light field information and no CNNs based methods are proposed because of the limited access to enough data. Li *et al.* [39] develop the first saliency detection method which utilizes the focusness and objectness simultaneously. Then, a unified saliency detection framework is presented by Li *et al.* [38] for handling heterogeneous types of input data (RGB images, RGB-D images and light field images). In [67], Zhang *et al.* introduce the depth cue in conjunction with background prior and location prior into light field saliency computation. Zhang *et al.* [69] obtain a list of saliency cues from light-field images, such as color, depth, flow and multiple viewpoints and then integrate them by using location prior as a multiplicative weighting factor. These works show that the unique refo-

cusing capability of light fields can greatly improve the accuracy of saliency detection. However, all the existing light field methods are limited to computing the hand-crafted features and omit the semantic information of salient objects which is effective in some complex scenarios.

Overall, compared with the traditional saliency methods, the CNNs based ones can improve the prediction performance by a large margin. However, existing deep salient methods are only based on RGB or RGB-D images, which ignore the important aspect of eye movement, that is, attention shifting across depth planes. The focal slices focusing on different depth in light field dataset shows the unique refocusing capability of light fields. Images in the focal stack value different in saliency prediction because various objects and background are located in different slices. To explore the interaction mechanism between light field cues, in this paper we focus on how to effectively fuse the focal stacks and all focus images in a united framework.

2.2. Deep Fusion Methods

Many existing works have performed fusion in different ways, including early [11, 61], late fusion [9, 37] and layer-wise fusion [24]. Couprie *et al.* [11] utilize a multiscale CNN for semantic segmentation. RGB image and depth image are concatenated and then transformed by a Laplacian pyramid to be fed into a CNN. In [61], saliency prior map and RGB image are concatenated to serve as the four-channel input to a recurrent fully convolutional network. In [9], a gated fusion layer is learned to combine RGB and depth prediction maps for semantic segmentation. Li *et al.* [61] propose a pixel-level fully convolutional stream and a segment-wise spatial pooling stream to produce two saliency predictions. Then those predictions are fused to produce the saliency map. In [24], Gao *et al.* learn to automatically fuse features at every convolutional layer for different tasks.

Different from their works, our method can utilize the advantage of refocused region in every slice to help saliency detection. Also, our method can pay more attention to the

useful slices by the recurrent attentive network. When applying the aforementioned different fusion mechanism to focal stacks, early fusion treats each slice (with RGB image) equally and only concatenate them across RGB channels without high-level semantic interaction among slices. Layer-wise fusion have the interaction between focal stacks and RGB image but omit the interaction between each two slices. Late fusion of focal stack extract features of each slice and then average the features to make prediction, which do not consider the importance priority of each slice. By the recurrent attention, our method can iteratively compute the weight of each slice in the focal stack and get a weighted fusion of the semantic feature of each slice, which can help improve the performance compared to the aforementioned fusion methods.

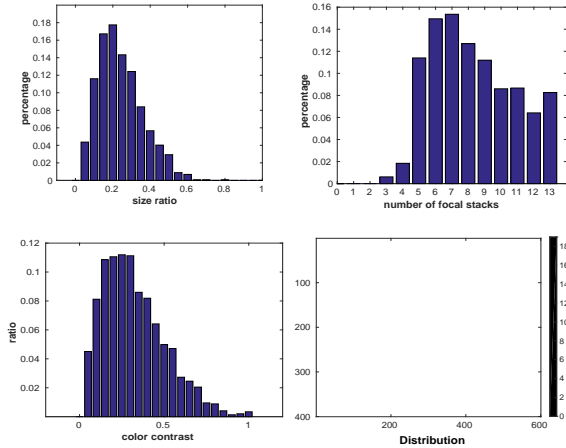


Figure 3. Statistics of the proposed light field dataset. (a), (b), (c) and (d) are located from left to right and from top to bottom, respectively. (a): the size of the salient objects. (b): the number of slices in each focal stack. (c): color contrast between the foreground and background. (d): the spatial distribution of the center of salient objects.

	Scale	Object Num.	Object Types	Object Size	Color Contrast
LFSD	100	one (mostly)	<100	0.28	0.39
HFUT-Lytro	255	multiple	<250	-	-
Ours	1465	multiple	>1000	0.22	0.30

Table 1. Comparison of the proposed dataset with two existing datasets in terms of the dataset scale, number of salient objects, type of objects, averaged size of objects and the color contrast.

3. The 4D light field dataset

The 4D light-field images are rarely available on the Internet, especially the ones including salient objects. In order to remedy the shortage of light field images, we introduce a large-scale light field dataset, which is captured with a wide range of indoor and outdoor scenes. The indoor scenes include libraries, offices, classrooms, supermarkets and so on while outdoor locations include the streets, campuses, outdoor markets and the like. Also, the images in our dataset

involve various lighting condition and different camera parameters. We utilize Lytro Desktop to convert the light field format files to jpeg images which can be handled directly and easily. We initially capture over 3000 images using the Lytro Illum camera.

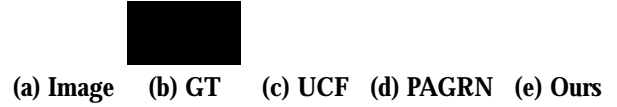


Figure 4. Example benefits of using light-field images.

Then, we discard the images which are repetitive, blurred, or contain large salient objects. After the preprocessing, we retain 1465 images to build the final dataset. Each image contains one all-focus image with the pixel-wise ground truth and one focal stack. The spatial resolution of the images is 600×400 . To obtain the pixel-wise ground truth, we manually label the images using a custom segmentation tool. We first draw the coarse boundary along the salient objects and then check the segmentation results to further refine the boundaries until we obtain final accurate annotation. Finally, the dataset are randomly divided into two parts, including 1000 training and 465 test images.

We also provide the statistics of the proposed dataset in Figure 3, which shows the object size, focal stack size, mean color contrast and location distribution of salient objects among all images. From Figure 3(a), we can see that the area percentage of all salient objects over the whole image lies in a range of $[0.05, 0.8]$ and most objects occupy less than 40% area of the image. Corresponding to every all-focus image, the number of focal slices varies from 2 to 13 (see Figure 3(b)). And most focal stacks contain more than 5 slices, which can demonstrate the diversity of image depths. The color contrast is another criterion for evaluating the challenge of an image. We compute the averaged RGB feature of all pixels inside and outside salient objects and then compute the Euclidean distance between two averaged RGB feature for each image. Figure 3(c) shows that the salient objects have low color contrast against the background and the averaged contrast is 0.30. To avoid that salient objects can be easily extracted by the center prior cue, we provide the location distribution for the center of every salient objects across the whole dataset. It can be seen from Figure 3(d) that the center of the salient objects appear at a variety of locations. Moreover, some detailed comparison of the proposed dataset and the existing two datasets are given in Table 1.

We provide some examples to demonstrate the benefits of using light-fields in Figure 4. For examples in the first and second rows, from the 2D images alone, the prediction map generate more noise on background. However, with

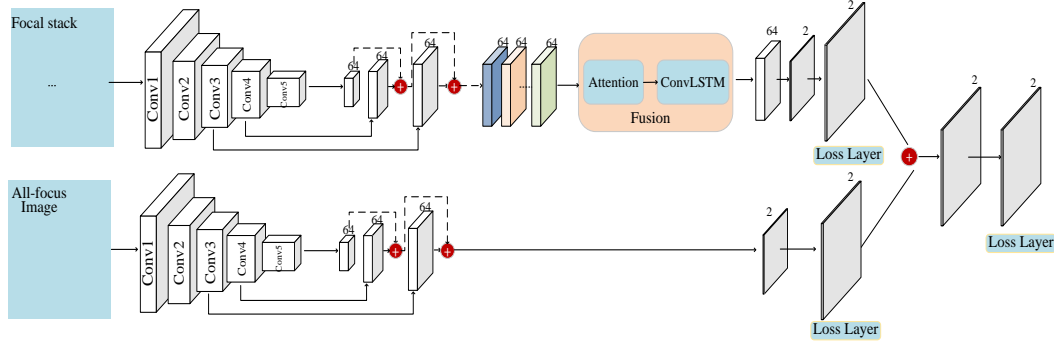


Figure 5. The overall two branch network based on the weighted late integration of the focal stack.

the aid of light field images, this prediction becomes much easier, since the object and background noise lie in different focal slice and we can choose the most relevant one to make salient objects stand out from the background.

4. CNN structures

In this section, we concentrate on the problem-how to make effective use of the captured 4D light field images, including the all-focus images and focal stacks. We propose a two-stream trainable convolutional neural network architecture to address the problem of salient object detection. The proposed model can be trained end-to-end. We first present the basic structure in Section 4.1 and then introduce the proposed framework in Section 4.2. Besides the proposed method, we also describe four alternative fusion structures which one may put forward straightforwardly based on the rich light field data. Though those architectures can also address the proposed task, we analyze the difference among those methods and emphasize the advantages of the proposed one. Finally, we introduce the adversarial examples and integrate them with the original images to help fine-tune the proposed network in Section 4.3.

4.1. Basic Structure

As shown in the Figure 2(b-d), each of the two-stream CNN is composed of the basic structure (a), which is a FCN-based network. For the two-stream network, we feed the focal stacks into the first stream and treat all-focus images as the input of the second stream. Both streams are based on the VGG19 network. We just retain the first 16 convolutional layers and remove the last max pooling, two fully connected and softmax loss layers which are designed for the task of classification. There are five convolutional blocks in VGG19 and given an input image, the output feature maps of each convolutional block are represented as f^1, f^2, \dots, f^5 . The feature maps f^4 and f^5 has the smallest spatial resolution which are the 1/16 of the input image. To construct the first stream, we connect three convolutional layers behind f^3, f^4, f^5 to reduce the dimension of the corresponding output feature maps to 64, which are denoted by

f_r^3, f_r^4, f_r^5 . Then we utilize a FCN-like structure where the features f_r^4, f_r^5 are upsampled by using the bilinear interpolation and using element-wise addition operation on the upsampled f_r^4 and f_r^5 with f_r^3 to produce the final feature representation. The output feature map is 1/4 of the input spatial resolution and the feature dimension is 64. The second stream employs the similar architecture as the first one and the only difference is that we connect one convolutional layer with 2 kernels of 3×3 after the output 64×64 feature map to generate the prediction map. The first channel is the background mask while the second one denotes the salient mask. The spatial size of the prediction map is same as the output feature of the first stream. The detailed framework of the two streams can be found in Figure 5.

4.2. Different Fusion Structures

As shown in Figure 2, there are several different ways to perform the integration of these two data streams, ranging from the early fusion to the later one.

Early integration of light field data. As shown in Figure 2(b), we only utilize the second stream and concatenate each slice in the focal stack and all-focus image across their RGB channels before feeding them into the second stream. This can be realized by changing the input channel of the first convolutional layer of VGG19 while leaving the rest layers unchanged. However, each focal stack slice and all-focus image are only dealt with through the low-level color features, which definitely lose the high-level semantic relationship among those images.

Layer-wise integration of light field data. For every convolutional block of VGG19, we extract the feature maps of both streams indicated in Figure 2(c). Then we integrate both features through element-wise addition. The output features serve as the input to the next convolutional layer for both streams. This can exploit the hierarchical feature correlation of each focal stack and all-focus image. However, the input are the concatenated RGB features of the focal slices, which omit the interactive relevance between each two slices.

Late integration of focal stack. See Figure 2(d), we

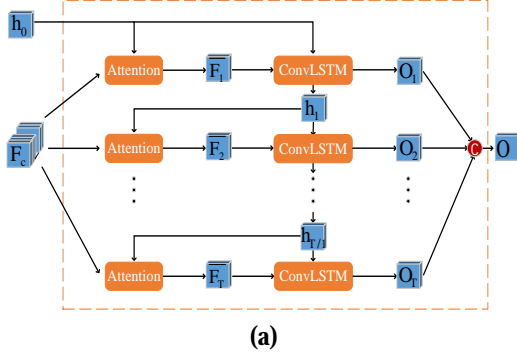


Figure 6. The structure of the recurrent attention network is shown in (a) and (b) denotes the attention subnet.

feedforward each slice independently into the first stream and average each output feature map directly. This method combines the information extracted from hierarchical features of the CNN. However, this just treat the features of each slice equally and does not fully consider the role of each slice plays in the salient prediction.

Weighted late integration of focal stack. Our proposed fusion process is based on the model of Figure 2(d), as shown in Figure 5. We focus on the integration of the diversified focal slices with an recurrent attention model in Figure 6. Specifically, an attention subnet is utilized to learn the importance of each slice in a focal stack and a recurrent ConvLSTM [64] is employed to learn the integrated feature representation.

Here, we utilize $I = \{I_n | I_n \in \mathbb{R}^{W \times H \times C}\}_{n=1}^N$ to denote the input focal stack I with N slices. W and H represents the width and height of the slice, respectively. C is the dimension of the image. The focal stack stream accepts I as the input and output the feature map of each slice, denoted by $F = \{F_n | F_n \in \mathbb{R}^{W_s \times H_s \times C_s}\}_{n=1}^N$. We show the detailed framework of the recurrent attention model in Figure 6. At each time step t , the attention subnet adopts the N features and the hidden state h_{t-1} of ConvLSTM as the input and the output \bar{F}_t is a weighted average of the input features. The attention subnet is described in the right of Figure 6. We connect one convolutional layer behind F_c and h_{t-1} to reduce the dimension of the feature maps to 64 for efficient computation. Then an element-wise addition is operated on both features and a global average pooling layer is adopted to aggregate the spatial information of each position. Next, we use a convolutional layer with N kernels of 1×1 to predict the weight $w_{t,n}$ of each feature map in F . $w_{t,n}$ is spatially normalized with the softmax operation via $\bar{w}_{t,n} = \exp(w_{t,n}) / \sum_{n=1}^N \exp(w_{t,n})$. The output map of the attention subnet is calculated by

$$\bar{F}_t = \sum_{n=1}^N \bar{w}_{t,n} F_n. \quad (1)$$

The attention subnet can focus on the focal slices which

contribute much to the saliency detection. To further exploit the spatial relevance among the attentive features, we next feed the \bar{F}_t into the ConvLSTM network. At each time step t , the ConvLSTM uses the hidden state h_{t-1} from previous time step and attentive feature \bar{F}_t as input and generate the output o_t , which is described by the following formulations,

$$\begin{aligned} i_t &= \text{sigmoid}(W_{xi} \bar{F}_t + W_{hi} h_{t-1} + W_{ci} C_{t-1} + b_i) \\ f_t &= \text{sigmoid}(W_{xf} \bar{F}_t + W_{hf} h_{t-1} + W_{cf} C_{t-1} + b_f) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} \bar{F}_t + W_{hc} h_{t-1} + b_c) \\ o_t &= \text{sigmoid}(W_{xo} \bar{F}_t + W_{ho} h_{t-1} + W_{co} C_{t-1} + b_o) \\ H_t &= o_t \odot \tanh(C_t), \end{aligned} \quad (2)$$

where C_t denotes the cell output at time step t , H_1, \dots, H_t is the hidden state and i_t, f_t, o_t represents the gates. i_t is the input gate which determines if the information will be accumulated to the cell state C_t . f_t is the forget gate, which decides what information can be thrown from the cell state C_t . Input information will be accumulated to the cell C_t if the input gate i_t is on and the past cell state C_{t-1} will be forgotten if the forget gate f_t is on. The final state H_t will accumulate the latest cell output C_t if the output gate o_t is on. All b means the bias of the convolutional layer. The symbol \odot denotes the convolution operation, \odot denotes the Hadamard product (element-wise product) and $\sigma(\cdot)$ means sigmoid function.

The final feature representation of the focal stack is the concatenation of the feature maps of its output o_t at each time step $t \in \{1, 2, \dots, T\}$. After obtaining the integrated features of each slice, we connect one convolutional layer with 64 channels and another one layer with two channels to make the saliency prediction of the first stream. Finally, the prediction map generated by the first stream will combine with the one generated by the second stream to make the final prediction.

4.3. Adversarial Examples

Adversarial perturbations cause a neural network to change its original prediction when added to the original

Datasets		MST	BSCA	DCL	DHS	DSS	Amulet	UCF	PAGRNet	PiCANet	R ³ Net	DFRGBD	RGBD	ACSD	DILF	LFS	WSC	Ours
LFSD	maxF	0.704	0.795	0.780	0.856	0.768	0.863	0.865	0.840	0.867	0.876	0.841	0.841	0.780	0.849	0.779	0.786	0.863
	MAE	0.209	0.205	0.161	0.115	0.178	0.093	0.143	0.132	0.111	0.098	0.180	0.197	0.218	0.153	0.239	0.168	0.093
	S-m	0.646	0.725	0.742	0.803	0.678	0.801	0.808	0.766	0.822	0.811	0.732	0.650	0.681	0.801	0.655	0.700	0.826
	E-m	0.720	0.766	0.784	0.844	0.865	0.847	0.844	0.791	0.847	0.852	0.737	0.650	0.675	0.845	0.625	0.770	0.877
Ours	maxF	0.545	0.642	0.716	0.816	0.735	0.782	0.789	0.849	0.851	0.761	0.722	0.570	0.262	-	0.439	-	0.868
	MAE	0.210	0.215	0.156	0.095	0.132	0.070	0.153	0.084	0.089	0.114	0.163	0.202	0.337	-	0.259	-	0.070
	S-m	0.594	0.66	0.710	0.803	0.714	0.777	0.770	0.810	0.838	0.733	0.687	0.5	0.357	-	0.517	-	0.852
	E-m	0.717	0.742	0.781	0.865	0.784	0.843	0.828	0.841	0.872	0.808	0.684	0.432	0.545	-	0.545	-	0.905

Table 2. Quantitative comparison of maximum F-measure, MAE, S-measure, E-measure scores on two datasets. The color in red and blue represent the best and second scores.

input I . By training using the original images and the adversarial examples, the whole network of CNNs can avoid overfitting to some extent and can still predict exactly when facing the perturbation from different source. To generate the adversarial examples, we utilize the formulation which is described below,

$$I^{\text{adv}} = I + \alpha \cdot \text{sign}(-\nabla_{\mathbf{I}} J(f(I; \theta), S)), \quad (3)$$

which is motivated by increasing the cross-entropy of the network on the input image I and ground truth mask S . The symbol $f(\cdot)$ denotes the neural network parametrized by θ . α is set to the constant 0.1 which can control the magnitude of the noise. $\nabla_{\mathbf{I}} J(\cdot, \cdot)$ is the gradient of the models loss function with respect to the input image I .

We show visual examples with and without adopting adversarial examples in Figure 7. It can be seen that by introducing the adversarial examples to help training process, the predicted map can prohibit the background noise better.



Figure 7. Examples with and without adversarial examples.

5. Experiments

5.1. Training Details

All networks are implemented using the publicly available Pytorch toolbox with two Nvidia 1080-Ti GPUs. We employ the general data augmentation schemes, including the flipping, cropping and rotating operations. Specifically, we use the horizontal-flipping and vertical-flipping and crop out the most top, bottom, left, right and middle 9/10 image. We also rotate all images with the angles of 90°, 180°, 270°. In sum, we increase the training set by 11 times including the original image. We set the momentum and weight decay to 0.99 and 0.0005 respectively. The learning rate of the initial VGG19 and other layers is fixed to 10^{-10} and 10^{-8} ,

respectively. We utilize the official VGG19 Pytorch model to initialize the whole network. All the weight parameters for convolutional layers not in the VGG19 network are initialized by a normal distribution with zero mean and 10^{-2} variance, while the biases are initialized with constant zero. The minibatch size is set to 4. We resize all training images and testing ones to 256×256 . We first train the whole network in Figure 5 with the augmented images, and after this we utilize Equation 3 with the ground truth masks to generate adversarial examples. Then we finetune the whole network by integrating the original images with the adversarial ones until it converges.

5.2. Datasets

To evaluate the performance of the proposed approach, we conduct experiments on LFSD [39] dataset as well as the proposed dataset.

LFSD contains 100 light field images captured by the Lytro light field camera, including 60 indoor and 40 outdoor scenes. This is first light field dataset designed for saliency detection. Most of the scenes contain only one salient object with high contrast with the background.

The proposed dataset contains 1000 training images and 465 test images. This is a more challenging dataset with the following characteristics: lower contrast between the salient objects and the background, more small-scale salient objects, multiple disconnected salient objects and various light conditions such as the dark light or strong light.

5.3. Evaluation Metrics

We adopt precision-recall (PR) curves, F-measure [1], mean absolute error (MAE) scores [49], Structure-measure (S-measure) [19] and Enhanced-alignment measure (E-measure) [20] to verify the effectiveness of our proposed algorithm.

5.4. Comparison with the state-of-the-art

Here we provide the quantitative comparison of 16 state-of-the-art salient object detection methods, including the 2D, 3D and 4D algorithms, i.e., BSCA [51], MST [59], DCL [37], DHS [43], DSS [27], Amulet [71], UCF [72], PAGRN [73], PiCANet [44], R³Net [12], DFRGBD [53], RGBD [48], ACSD [29], LFS [40], WSC [38] and

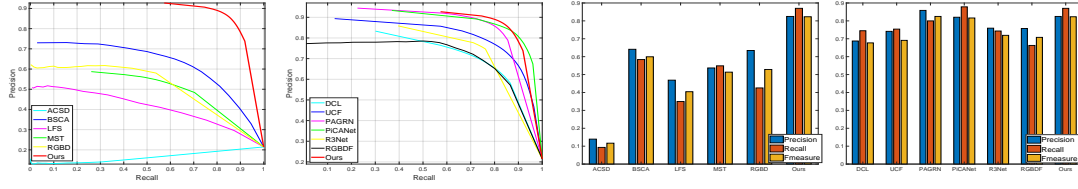


Figure 8. Comparison of several state-of-the-art methods on our dataset. The first and second columns show the P-R curves and the last two columns show the F-measure scores.

DILF [68]. We either use the saliency maps provided by the authors or run the available codes by using the default parameters set by the authors.

For quantitative evaluation, we list the MAE, F-measure, S-measure and E-measure scores in Table 2. It can be seen that the proposed method performs favorably against the 2D, 3D and 4D algorithms across two datasets in terms of four metrics. We then further provide the P-R curves in Figure 8, which can also demonstrate the effectiveness of the proposed method among all algorithms.

Qualitative results of the proposed method, other state-of-the-art methods and more results can be found in the supplementary material.

5.5. Ablation Analysis

To verify the advantages of the proposed method, we provide the experimental results in terms of Max F-measure and MAE scores in Table 3 for different variants. First, compared with the framework without using the focal stacks (a), most methods produce better results which demonstrates that the introduction of focal stacks can supply supplementary information for saliency detection. Second, we can see the performance of different models (b-d) varies in terms of different integration mechanism, which shows that how to perform integration between all-focus images and focal stacks are of vital importance. Some example samples can be found in Figure 9. We find that the proposed method can highlight salient objects uniformly and produce sharp boundaries.

We also provide the results of our structure without employing ConvLSTM (+att). Compared to (d) where the output features of each focal stack are operated by an element-wise addition and the utilization of both ConvLSTM and attention modules, the attention mechanism without LSTM decreases the performance, which demonstrates the interaction between LSTM and attention mechanism are important for the saliency detection. By utilizing the ConvLSTM, our method can learn more effective feature representation with the recurrent mechanism. Also, compared to the model without adversarial examples (+att+LSTM), our model performs better which derives from the advantages that adversarial examples in assisting training the deep networks.

*	LFSD		Our dataset	
	MAE	maxF	MAE	maxF
(a)	0.133	0.817	0.104	0.819
(b)	0.125	0.825	0.086	0.847
(c)	0.126	0.831	0.104	0.823
(d)	0.132	0.830	0.110	0.822
+ att	0.236	0.825	0.227	0.819
+ att + LSTM	0.100	0.851	0.072	0.863
Ours	0.093	0.863	0.070	0.868

Table 3. Ablation study of various structures. (a)-(d) represent the structures in Figure 2. The symbol ‘+’ means we gradually add modules on the late fusion of focal stacks (d). ‘att’ denotes the attention mechanism.



Image GT (a) (b) (c) (d) Ours
Figure 9. Visual comparison among different networks.

6. Conclusion

In this paper, we first introduce a large-scale light field dataset to address the deficiency problem of light field data. Our dataset contains 1465 images, which is much larger than all the previous datasets. Then we focus on the problem of how to introduce the light field data into deep learning and how to effectively combine the light field data (all-focus images and focal stacks). We utilize a recurrent attention network to fuse each slice in the focal stack. The attention network can focus on the most informative features of each slice and generate a weighted fusion of them. The recurrent network (ConvLSTM) is employed to effectively learn the feature representation relying on the spatial relation among the focal slices. To further increase the robustness of the proposed method, we introduce adversarial examples to serve as the input to help train the network. Extensively quantitative and qualitative evaluations show the promising results of the proposed method over the existing 2D, 3D and 4D images.

7. Acknowledgements

This work was supported by the Dalian Science and Technology Innovation Foundation (2019J12GX039), National Natural Science Foundation of China (61605022, 61876202 and U1708263) and the Fundamental Research Funds for the Central Universities (DUT19JC58).

References

- [1] R. Achanta, S.S. Hemami, F.J. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009. **7**
- [2] R. Achanta, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. Technical report, EPFL, Tech. Rep. 149300, 2010. **2**
- [3] Radhakrishna Achanta and Sabine Ssstrunk. Saliency detection using maximum symmetric surround. In *ICIP*, pages 2653–2656, 2010. **2**
- [4] Naoki Asada, Hisanaga Fujiwara, and Takashi Matsuyama. Edge and depth from focus. *IJCV*, (2):153–163, 1998. **2**
- [5] David Braue. Kinect for xbox. *APC*, 2011. **2**
- [6] Neil D. B. Bruce and John K. Tsotsos. Saliency based on information maximization. In *NIPS*, pages 155–162, 2005. **2**
- [7] M.M. Cheng, G.X. Zhang, N.J. Mitra, X. Huang, and S.M. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. **1**
- [8] Ming Ming Cheng, Guo Xin Zhang, N. J. Mitra, Xiaolei Huang, and Shi Min Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011. **2**
- [9] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Localitysensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, volume 3, 2017. **3**
- [10] Arridhana Ciptadi, Tucker Hermans, and James Rehg. An in depth view of saliency. In *BMVC*, pages 112.1–112.11, 2013. **1, 2**
- [11] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv*, 2013. **3**
- [12] Zijun Deng, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Jing Qin, Guoqiang Han, and Pheng-Ann Heng. R3net: Recurrent residual refinement network for saliency detection. In *IJCAI*, pages 684–690. AAAI Press, 2018. **7**
- [13] Karthik Desingh, Krishna K Madhava, Deepu Rajan, and C. V. Jawahar. Depth really matters: Improving visual salient region detection with depth. In *BMVC*, pages 98.1–98.11, 2013. **1, 2**
- [14] Henghui Ding, Xudong Jiang, Ai Qun Liu, Nadia Magnenat Thalmann, and Gang Wang. Boundary-aware feature propagation for scene segmentation. In *ICCV*, 2019. **2**
- [15] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *CVPR*, pages 2393–2402, 2018. **2**
- [16] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *CVPR*, pages 8885–8894, 2019. **2**
- [17] Lijuan Duan, Chunpeng Wu, Jun Miao, and Laiyun Qing. Visual saliency detection by spatially weighted dissimilarity. In *CVPR*, pages 473–480, 2011. **2**
- [18] Deng-Ping Fan, Ming-Ming Cheng, Jiang-Jiang Liu, Shang-Hua Gao, Qibin Hou, and Ali Borji. Salient objects in clutter: Bringing salient object detection to the foreground. In *ECCV*, pages 186–202, 2018. **1**
- [19] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. **7**
- [20] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv*, 2018. **7**
- [21] Deng-Ping Fan, Zheng Lin, Jia-Xing Zhao, Yun Liu, Zhao Zhang, Qibin Hou, Menglong Zhu, and Ming-Ming Cheng. Rethinking rgb-d salient object detection: Models, datasets, and large-scale benchmarks. *arXiv*, 2019. **1**
- [22] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019. **1**
- [23] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE TPAMI*, 27(3):406–417, 2005. **2**
- [24] Yuan Gao, Qi She, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layer-wise feature fusing in multi-task cnn by neural discriminative dimensionality reduction. *arXiv*, 2018. **3**
- [25] Junwei Han, Hao Chen, Nian Liu, Chenggang Yan, and Xuelong Li. Cnns-based rgb-d saliency detection via cross-view transfer and multiview fusion. *IEEE Transactions on Cybernetics*, PP(99):1–13, 2017. **1, 3**
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. **2**
- [27] Qibin Hou, Ming-Ming Cheng, Xiao-Wei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *CVPR*, 2017. **1, 7**
- [28] Laurent Itti, Christof Koch, and Ernst Niebur. *A model of saliency-based visual attention for rapid scene analysis*. IEEE Computer Society, 1998. **2**
- [29] Ran Ju, Yang Liu, Tongwei Ren, Ling Ge, and Gangshan Wu. Depth-aware salient object detection using anisotropic center-surround difference. *Signal Processing: Image Communication*, 38:115–126, 2015. **7**
- [30] Tilke Judd, Frdo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. 2012. **2**
- [31] Jiwhan Kim, Dongyoon Han, Yu Wing Tai, and Junmo Kim. Salient region detection via high-dimensional color transform. In *CVPR*, pages 883–890, 2014. **2**
- [32] Yuqiu Kong, Lijun Wang, Xiuping Liu, Huchuan Lu, and Xiang Ruan. Pattern mining saliency. In *ECCV*, pages 583–598, 2016. **1**
- [33] Congyan Lang, Tam V. Nguyen, Harish Katti, Karthik Yada, Mohan Kankanhalli, and Shuicheng Yan. Depth matters: influence of depth cues on visual saliency. In *ECCV*, pages 101–115, 2012. **1, 2**
- [34] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996. **1**
- [35] Guanbin Li and Yizhou Yu. Visual saliency based on multi-scale deep features. In *CVPR*, pages 5455–5463, 2015.

- [36] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016. **2**
- [37] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016. **3, 7**
- [38] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. In *CVPR*, June 2015. **1, 3, 7**
- [39] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *CVPR*, June 2014. **1, 2, 3, 7**
- [40] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *CVPR*, pages 2806–2813, 2014. **7**
- [41] Xiaohui Li, Huchuan Lu, Lihe Zhang, and Ruan Xiang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013. **2**
- [42] Xi Li, Liming Zhao, Lina Wei, Ming-Hsuan Yang, Fei Wu, Yueting Zhuang, Haibin Ling, and Jingdong Wang. Deep-saliency: Multi-task deep neural network model for salient object detection. *IEEE TIP*, 25(8):3919–3930, 2016. **1**
- [43] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016. **1, 7**
- [44] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018. **7**
- [45] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung Yeung Shum. Learning to detect a salient object. *IEEE TPAMI*, 33(2):353–367, 2010. **2**
- [46] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11):1–11, 2005. **1**
- [47] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgb-d salient object detection: A benchmark and algorithms. In *ECCV*, pages 92–109. Springer, 2014. **1, 2**
- [48] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgb-d salient object detection: a benchmark and algorithms. In *ECCV*, pages 92–109. Springer, 2014. **7**
- [49] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012. **7**
- [50] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015. **2**
- [51] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015. **7**
- [52] Liangqiong Qu, Shengfeng He, Jiawei Zhang, Jiandong Tian, Yandong Tang, and Qingxiong Yang. Rgb-d salient object detection via deep fusion. *IEEE TIP*, 26(5):2274–2285, 2016. **3**
- [53] Liangqiong Qu, Shengfeng He, Jiawei Zhang, Jiandong Tian, Yandong Tang, and Qingxiong Yang. Rgb-d salient object detection via deep fusion. *IEEE TIP*, 26(5):2274–2285, 2017. **7**
- [54] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, pages 366–379. Springer, 2010. **2**
- [55] Jianqiang Ren, Xiaojin Gong, Lu Yu, Wenhui Zhou, and Michael Ying Yang. Exploiting global priors for rgb-d saliency detection. In *CVPRW*, pages 25–32, 2015. **1, 2, 3**
- [56] Riku Shigematsu, David Feng, Shaodi You, and Nick Barnes. Learning rgb-d salient object detection using background enclosure, depth contrast, and top-down features. pages 2749–2757, 2017. **3**
- [57] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. **2**
- [58] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv*, 2013. **2**
- [59] Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien. Real-time salient object detection with a minimum spanning tree. In *CVPR*, pages 2334–2342, 2016. **7**
- [60] L. Wang, H. Lu, X. Ruan, and M. H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015. **2**
- [61] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016. **3**
- [62] Tiantian Wang, Ali Borji, Lihe Zhang, Pingping Zhang, and Huchuan Lu. A stagewise refinement model for detecting salient objects in images. In *ICCV*, pages 4019–4028, 2017. **1**
- [63] Tiantian Wang, Lihe Zhang, Huchuan Lu, Chong Sun, and Jinqing Qi. Kernelized subspace ranking for saliency detection. In *ECCV*, pages 450–466, 2016. **1**
- [64] Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015. **6**
- [65] Chuan Yang, Lihe Zhang, Huchuan Lu, Ruan Xiang, and Ming Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013. **2**
- [66] Miao Zhang, Xiao Li, Huchuan Lu, Yongri Piao, Zhengkun Rong. Deep light-field-driven saliency detection from a single view. In *IJCAI*, 2019. **1**
- [67] Jun Zhang, Meng Wang, Jun Gao, Yi Wang, Xudong Zhang, and Xindong Wu. Saliency detection with a deeper investigation of light field. In *IJCAI*, pages 2212–2218, 2015. **1**
- [68] Jun Zhang, Meng Wang, Jun Gao, Yi Wang, Xudong Zhang, and Xindong Wu. Saliency detection with a deeper investigation of light field. In *IJCAI*, pages 2212–2218, 2015. **8**
- [69] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. Saliency detection on light field: A multi-cue approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(3):32, 2017. **1, 3**

- [70] Lihe Zhang, Chuan Yang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Ranking saliency. *IEEE TPAMI*, 39(9):1892–1904, 2016. [2](#)
- [71] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017. [7](#)
- [72] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency detection. *arXiv*, 2017. [1](#), [7](#)
- [73] Xiaoning Zhang, Tiantian Wang, Jinqing Qi, Huchuan Lu, and Gang Wang. Progressive attention guided recurrent network for salient object detection. In *CVPR*, pages 714–722, 2018. [7](#)
- [74] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MM*, 19(2):4–10, 2012. [1](#)
- [75] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. [2](#)
- [76] Jiaxing Zhao, Ren Bo, Qibin Hou, and Ming-Ming Cheng. Flic: Fast linear iterative clustering with active search. In *AAAI*, 2018. [2](#)
- [77] Jiaxing Zhao, Yang Cao, Deng-Ping Fan, Xuan-Yi Li, Le Zhang, and Ming-Ming Cheng. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *CVPR*, 2019. [1](#)
- [78] Jiaxing Zhao, Jiangjiang Liu, Dengping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnnet:edge guidance network for salient object detection. In *ICCV*, 2019. [1](#)
- [79] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *CVPR*, pages 1265–1274, 2015. [2](#)
- [80] Zhen Zhou, Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *CVPR*, pages 6776–6785. IEEE, 2017. [2](#)