Light Field Saliency Detection with Dual Local Graph Learning and Reciprocative Guidance

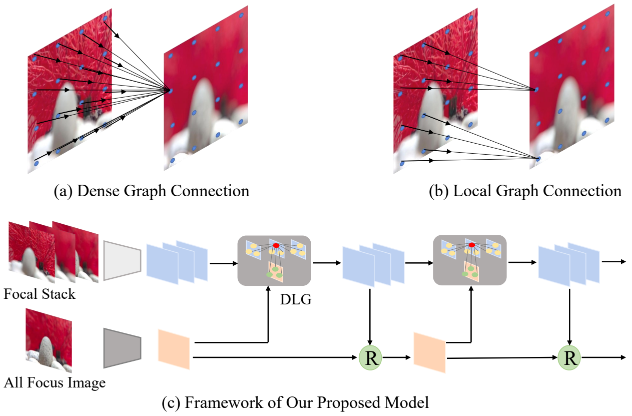
Nian Liu[1]* Wangbo Zhao[2]* Dingwen Zhang[2] Junwei Han[2]† Ling Shao[1]
[1] Inception Institute of Artificial Intelligence [2] Northwestern Polytechnical University
{liunian228, wangbo.zhao96, zhangdingwen2006yyy, junweihan2010}@gmail.com,
ling.shao@ieee.org

Abstract

The application of light field data in salient object detection is becoming increasingly popular recently. The difficulty lies in how to effectively fuse the features within the focal stack and how to cooperate them with the feature of the all-focus image. Previous methods usually fuse focal stack features via convolution or ConvLSTM, which are both less effective and ill-posed. In this paper, we model the information fusion within focal stack via graph networks. They introduce powerful context propagation from neighbouring nodes and also avoid ill-posed implementations. On the one hand, we construct local graph connections thus avoiding prohibitive computational costs of traditional graph networks. On the other hand, instead of processing the two kinds of data separately, we build a novel dual graph model to guide the focal stack fusion process using all-focus patterns. To handle the second difficulty, previous methods usually implement one-shot fusion for focal stack and all-focus features, hence lacking a thorough exploration of their supplements. We introduce a reciprocative guidance scheme and enable mutual guidance between these two kinds of information at multiple steps. As such, both kinds of features can be enhanced iteratively, finally benefiting the saliency prediction. Extensive experimental results show that the proposed models are all beneficial and we achieve significantly better results than state-of-the-art methods.

1. Introduction

Salient object detection (SOD) methods can be categorized into RGB based ones, RGB-D based ones, and the recently proposed light field based ones. By only based on static images, although RGB SOD methods [18, 26, 24, 54] have achieved excellent performance on many benchmark datasets, they still can not handle challenging and complex scenes. This is because the appearance saliency cues conveyed in RGB images are heavily constrained, especially when the foreground and background appearance are complex or similar. To solve this problem, depth information is introduced to provide supplementary cues in RGB-D SOD methods [3, 52, 31, 27]. However, it is not easy to obtain high-quality depth maps and many current RGB-D SOD benchmark datasets only have noisy depth maps. On the contrary, light field data Hence, the light field SOD problem has much potential to explore.


(a) Dense Graph Connection

(b) Local Graph Connection

(c) Framework of Our Proposed Model

Figure 1. (a) and (b) show the comparison of traditional dense graph models and our proposed local graph model. (c) illustrates the framework of our model. Ⓡ means the reciprocative unit.

Besides the focal stack images, light field data also have an all-focus image that provides the context information. Thus, light field SOD has two key points, *i.e.*, *how to effectively fuse multiple focal stack features and how to cooperate the focal stack cues with the all-focus information*. A straightforward way to solve the first problem is to concatenate focal stack features and use a convolution layer for fusion. Such a simple way can not sufficiently explore the complex interaction within different focal slices, hence may limit the model performance. It is also an ill-posed solution since convolution requires a fixed input number, hence many methods have to randomly pad the input images when

*Equal contribution.
†Corresponding author.

they are less than the pre-defined number. Adopting ConvLSTM [43] is another popular solution, where the focal stack images are processed one by one in a pre-defined sequential order using the memory mechanism. This also involves an ill-posed problem setting since there is no meaningful order among focal stack images. Furthermore, the usage of the sequential order may cause ConvLSTM to ignore the information of the focal slices that are input earlier. As for the second problem, most previous works [38] simply concatenate or sum the focal stack feature with the all-focus feature and then adopt convolutional fusion only once. Such a straightforward fusion method heavily limits the exploration of complex supplementary relations between these two kinds of information.

To solve the first problem, adopting the powerful graph neural networks (GNNs) [14, 35] is a possible way. GNNs aggregate the contextual information from neighbouring nodes and propagate it to the target node, thus can achieve effective feature fusion. At the same time, it avoids the ill-posed implementation problem since the graph connection can be built flexibly and does not depend on a sequential order. A straightforward way is to view each pixel location in the feature maps of the focal stack as a node and construct dense edge connections among all locations, as shown in Figure 1(a). However, this is impractical since light field SOD requires large feature maps for focal slices to obtain fine-grained segmentation. Hence, building a densely connected graph involves prohibitive computational costs.

To this end, we propose to build local graphs to efficiently aggregate contexts in different focal slices. We treat each image pixel location in the focal stack as nodes and build the graph only within local neighbouring nodes, as shown in Figure 1(b). As such, the context propagation within focal slices can be efficiently performed with dramatically reduced edge connections. One can further introduce multiscale local neighbours, hence incorporating larger context information with acceptable computational costs. Besides building a graph within the focal stack, we also build a focal-all graph to introduce external guidance from the all-focus feature for the fusion of focal features, thus resulting in a novel dual local graph (DLG) network.

To tackle the second key point in light field SOD, we propose a novel reciprocative guidance architecture, as shown in Figure 1(c). It introduces multi-step guidance between the all-focus image feature and the focal stack features. In each step, the former is first used to guide the fusion of the latter, and then the fused feature is used to update the former. We perform such a process in a reciprocative fashion, where mutual guidance can be conducted recurrently. Finally, the two kinds of features can be improved with more discriminability, benefiting the final SOD decision.

Our main contributions can be summarised as:

- We propose a new GNN model named dual local graph to enable effective context propagation in focal stack features under the guidance of the all-focus feature and also avoid high computational costs.
- We propose a novel reciprocative guidance scheme to make the focal stack and the all-focus features guide and promote each other at multiple steps, thus gradually improving the saliency detection performance.
- Extensive experiments illustrate the effectiveness of our method. It surpasses other light field methods by a large margin. Moreover, with much less training data, our method also shows competitive or better performance compared with RGB-D or RGB based SOD models.

2. Related Work

2.1. Light Field SOD

Although the usage of CNNs has improved RGB SOD and RGB-D SOD by a large margin [39, 56], there are still lots of challenges in the SOD task, especially when the visual scenes are complex. Hence, several works have tried to leverage the focal cues in light field data to perform SOD. [21] was the first work to explore SOD with light field data, and constructed the first benchmark dataset. After that, the background prior [46], weighted sparse coding [20], and light field flow [47] are widely used for this new task. More details about traditional methods can be found in [12].

When it comes to the deep learning era, several deep-learning methods have promoted the light field SOD performance significantly. Zhang *et al*. [50] inputted the feature maps of focal slices and the all-focus image into a ConvLSTM [43] to fuse them one by one. This scheme has the ill-posed implementation problem. On the other hand, their method only fuses focal stack and all-focus features once. Wang *et al*. [38] and Piao *et al*. [33] both fused the features from different focal slices using varying attention weights, which are inferred at multiple time steps in a ConvLSTM. As such, they performed feature fusion within focal slices several times. However, [38] conducted focal stack and all-focus feature fusion only once, while [33] did not perform such a fusion. They adopted knowledge distillation [16] to improve the representation ability of the all-focus branch.

Different from previous works, our proposed DLG network enables efficient context fusion among all focal slice images. We also introduce the guidance from the all-focus feature into the focal stack fusion process, and our proposed reciprocative architecture introduces mutual guidance between the two kinds of features multiple times. These two points have never been explored by previous works.

2.2. Grapth Neural Network

Graph neural networks (GNNs) were proposed by [14] and developed by [35] to model data structures in graph

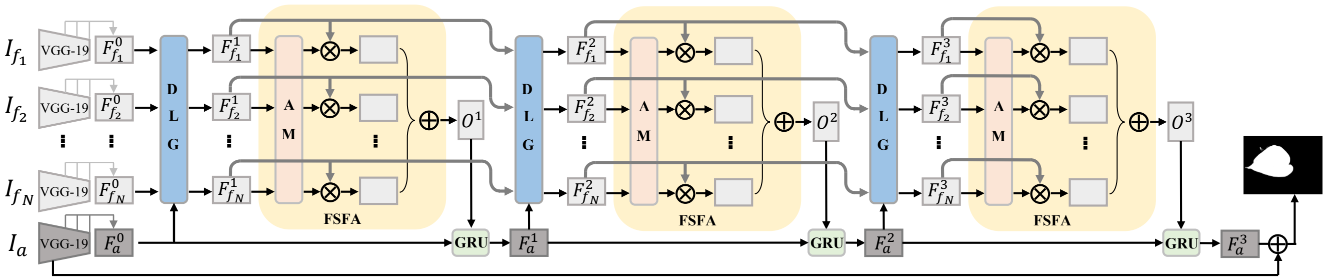


Figure 2. Overview of our proposed model. DLG: the proposed dual local graph; FSFA: focal stack feature aggregation; AM: used to generate the attention matrix A in (11); $\otimes$: element-wise multiplicationl; $\oplus$: element-wise addition; GRU: ConvGRU [5]. Due to space limitation, we only illustrate the reciprocative guidance process at three time steps.

domains. Since GNNs can model the relationships among nodes, they have been applied in many fields, such as molecular biology [13], natural language processing [2], knowledge graph [15], and disease classification [34].

Recently, GNNs have also been widely explored in the field of computer vision. Wang *et al*. [41] adopted a graph convolutional network to build the spatial-temporal relationships for action recognition. For dense prediction tasks, Luo *et al*. [28] used GNNs to construct graphs among feature maps and learn cross-modality and cross-scale reasoning simultaneously for RGB-D SOD. In [40], Wang *et al*. proposed an attentive GNN to learn the semantic and appearance relationships among several video frames for video object segmentation. Zhang *et al*. [48] adopted a graph convolutional network to jointly implement both intra-saliency detection and inter-image correspondence for co-saliency detection. Both the latter two works constructed densely-connected pixel-pixel graphs, which are computationally expensive and lack scalability, especially for the light field data that can have more than 10 focal stack images. On the contrary, we propose a novel graph architecture with local pixel-pixel connections for light field SOD. We also introduce dilated neighbouring connections to incorporate large contexts with computational efficiency. Furthermore, we build two graphs to simultaneously propagate context interaction among focal stack images and incorporate the guidance from the all-focus image.

2.3. Reciprocative Models

Reciprocative or recurrent models, including RNN, LSTM [17], and GRU [6], process temporal sequences using internal state or memory, and progressively update their states. There are also many other saliency detection works or other related tasks using reciprocative models. AGNN [40] and CAS-GNN [28] used reciprocative models as the node updating function in GNNs to update graph node embeddings. DMRA [31], DLSD [38], ERNet [33], and MoLF [50] combined attention models with reciprocative models to refine a given feature or a set of features. R3Net[7] and RFCN [37] recurrently fused a saliency map with CNN features or the input image to refine the saliency map. Different from them, we use reciprocative models to iteratively update two kinds of features, *i.e*., the focal stack feature and the all-focus feature, where the interactions between them are considered to introduce mutual guidance.

3. Proposed model

In Figure 2, we illustrate the overview of the proposed model. First, we use two encoders to extract features from the all-focus image and the corresponding focal slices, respectively. Then, we input them into the proposed DLG model to propagate contextual features among focal slices, which are further aggregated by the focal stack feature aggregation model. With the proposed reciprocative guidance scheme, focal stack and the all-focus features can be fused with each other several times, hence being improved progressively. Finally, the fused feature is fused with a low-level feature to predict the final saliency map.

3.1. General GNNs

GNNs have powerful capability to propagate contexts from neighbouring nodes for graph-structured data. Given a specific GNN model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ represents the set of nodes and $e_{i,j} \in \mathcal{E}$ represents the edge from v_j to v_i. Each node v_i has a corresponding node embedding as its inital state h_i^0. We use $\mathcal{N}_i$ to represent the set of neighbouring nodes of v_i. GNNs first aggregate contextual information from $\mathcal{N}_i$ to update the state of v_i with a learned message passing function M, which has specific formulations in different kinds of GNNs. The general formulation of the message passing process for v_i at step k can be written as:

$$m_i^k = M([h_1^{k-1}, h_2^{k-1}, ...h_j^{k-1}], [e_{i,1}, e_{i,2}, ...e_{i,j},]), \quad (1)$$

where each $v_j \in \mathcal{N}_i$. After message passing, a state update function U can be learned to update the state of v_i based on the aggregated message, which can be defined as:

$$h_i^{k+1} = U(h_i^k, m_i^k). \quad (2)$$

Finally, after K steps updating, a readout function can be applied to h_i^K to get the final output.

3.2. Feature Encoders

For the problem of light field SOD, we have an all-focus image I_a and its corresponding focal stack I_f with N focal slices $\{I_{f_1}, I_{f_2}, ...I_{f_N}\}$, which have different focused regions. Before defining nodes and their embeddings in the graph, we first use encoder networks to extract image features. As shown in Figure 2, I_a and I_f are first inputted into two unshared encoders to extract all-focus image features and focal stack features. Similar to previous works [33, 50], we adopt the VGG-19 [36] network without the last pooling layer and fully connected layers as the backbone of our encoder. We obtain high-level features from the last three convolutional stages. Then, we fuse them in an top-down manner [23] and obtain the fused multiscale features $F_a \in \mathbb{R}^{1\times C\times H\times W}$ and $F_f \in \mathbb{R}^{N\times C\times H\times W}$ at the $1/4$ scale, where $F_f = \{F_{f_1}, F_{f_2}, ...F_{f_N}\}$ represents the feature set of N focal slices, W, H, and C denote the width, height, and channel number of the feature maps, respectively.

3.3. Dual Local Graph

We use graph models to fuse the focal stack features F_f by propagating contexts within the focal slices and also under the guidance from the all-focus feature F_a. The latter can provide external guidance for the feature update of F_f.

Directly constructing a densely connected graph among F_f and F_a, which is the case in [40, 48], requires $(N+1)WH \times (N+1)WH$ edge connections. This scheme is computationally prohibitive for the message passing process when the feature maps have large spatial sizes. The reason for [40, 48] to use densely connected graphs is that the target objects in video object segmentation and co-saliency detection are usually located in different spatial locations. Thus, global context is needed. However, for light field SOD, each all-focus image and its corresponding focal stack images are spatially aligned. Hence, using local context solely is enough. Therefore, in this paper we propose a novel DLG model that only constructs edge connections within local neighbouring nodes for light field SOD. We design two subgraphs, named the focal-focal graph and the focal-all graph, to propagate contextual information from focal slices to focal slices and from the all-focus image to focal slices, respectively. The whole process can be defined as:

$$F'_f = DLG(F_f, F_a), \tag{3}$$

where $F'_f \in \mathbb{R}^{N\times C\times H\times W}$ represents the updated feature after context aggregation.

Defination of the surrounding area: Before the introduction of the proposed graph network, we first define a surrounding area of a location in a feature map. Given the pixel location (w, h) in a feature map, we have a sampling window with a size of $k \times k$ and dilation d, centering at (w, h). Then, we can view all sampled locations except the central one, *i.e.*, the location (w, h) itself, as its surrounding area, as shown as the blue dots in Figure 3. The surrounding area defines the context in a local region and can be used to construct local graph connections.

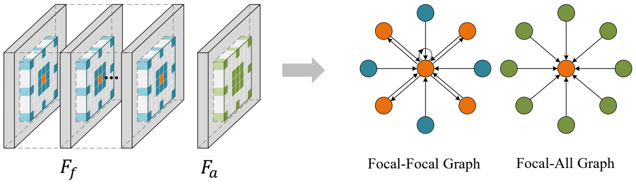


Figure 3. Structure of the Focal-Focal Graph and the Focal-All Graph. For each spatial location (w, h) in F_f, we have N target nodes (orange). For their corresponding surrounding area, we have $N \times (k \times k - 1)$ nodes (blue). In F_a, we regard the same location (w, h) and its surrounding area as the guidance context and obtain $k \times k$ nodes (green). The connections between one target node and its neighbors in two graphs are shown on the right.

Focal-Focal Graph: First, we build a graph $\mathcal{G}^f_{w,h} = (\mathcal{V}^f_{w,h}, \mathcal{E}^f_{w,h})$ for each spatial location (w, h) only in focal features. To ease the presentation, we omit the subscript below. Given the extracted focal stack feature map F_f, we can view it as having N points from N focal slices with C channels for each spatial location. To be specific, for the location (w, h), we have N target nodes with a C-dimensional embedding, which can be defined as $\mathcal{V}_T$. In the surrounding area of (w, h), we also have $N \times (k \times k - 1)$ nodes with a C-dimensional embedding, where we use $\mathcal{V}_S$ to represent the set of these nodes. Here we have $\mathcal{V}_T \cup \mathcal{V}_S = \mathcal{V}^f$.

After that, we define edges to link these nodes. We follow two rules: **1)** The nodes in $\mathcal{V}_T$ are our modeling targets. Hence they are linked to each other, including themselves. **2)** The nodes in $\mathcal{V}_S$ serves as local contexts for the target nodes. Therefore, they are linked to each node in $\mathcal{V}_T$. Except for these edges, there are no other connections in the graph. The two kinds of edges constitute $\mathcal{E}^f$.

Now, we need to define edge embeddings. For simplicity, we use u and v to represent a target node and one of its neighbours, respectively, *i.e.*, $u \in \mathcal{V}_T$ and $v \in \mathcal{V}^f$. Their states (features) can be written as h_u and h_v. The edge embedding $e^f_{u,v}$ represents the relation from v to u. As the two nodes are both from the focal stack feature map, we use inner product to compute the edge embeddings as:

$$e^f_{u,v} = \theta^f(h_u)^\top \phi^f(h_v), \tag{4}$$

where $\theta^f(*)$ and $\phi^f(*)$ are two linear transformation functions with learnable parameters. They have the same output dimensions and can be implemented by fully connected layers. As a result, the computed $e^f_{u,v}$ is a scalar.

Focal-All Graph: For the target of using the all-focus feature to guide the updating of focal features, we also build a graph $\mathcal{G}^a = (\mathcal{V}^a, \mathcal{E}^a)$ for focal and all-focus image features together at each spatial location (w, h). Again we omit the subscript (w,h) for easy presentation. For location (w,h) in F_f, we have the same target node set $\mathcal{V}_T$. Then, we regard the same location (w,h) in F_a and its surrounding area as the guidance context, and obtain a set of $k \times k$ nodes $\mathcal{V}'_S$. Here $\mathcal{V}_T \cup \mathcal{V}'_S = \mathcal{V}^a$.

We connect all nodes in $\mathcal{V}'_S$ to each node in $\mathcal{V}_T$ to incorporate the guidance context for all target nodes. Here we use u and q to represent a target node and one of its neighbours, respectively, *i.e.*, $u \in \mathcal{V}_T$ and $q \in \mathcal{V}'_S$. Similarly, their states are denoted by h_u and h_q. Since the two nodes are from two different feature spaces, we use a linear transformation to build the edge embedding from q to u, which can be defined as:

$$e^a_{u,q} = \psi([\theta^a(h_u), \phi^a(h_q)]), \tag{5}$$

where $[,]$ denotes the concatenation operation, $\theta^a(*)$ and $\phi^a(*)$ represent two linear transformation functions. The last linear function ψ projects the input to a scalar.

Message passing: After getting the embedding for each node and edge, we can define the formulation of the message passing process now. For the target node u, we respectively define the message passing in the Focal-Focal Graph and the Focal-All Graph as:

$$m^f_u = \sum_{v \in \mathcal{V}^f} \alpha^f_{u,v} g^f(h_v), \tag{6}$$

$$m^a_u = \sum_{q \in \mathcal{V}'_S} \alpha^a_{u,q} g^a(h_q), \tag{7}$$

where $g^f(*)$ and $g^a(*)$ are two linear transformation functions in the two Graphs, and α^*_* can be computed by the Softmax normalization:

$$\alpha^f_{u,v} = \frac{exp(e^f_{u,v})}{\sum_{j \in \mathcal{V}^f} exp(e^f_{u,j})}, \tag{8}$$

$$\alpha^a_{u,q} = \frac{exp(e^a_{u,q})}{\sum_{j \in \mathcal{V}'_S} exp(e^a_{u,j})}. \tag{9}$$

From (6) and (7), we can aggregate the contexts and guidance from neighbour nodes by considering the relationships inferred in (4) and (5).

Node Updating: After achieving the messages from neighbours in the two subgraphs, we can update the state of u by:

$$h'_u = \varphi^f(m^f_u) + \varphi^a(m^a_u) + h_u \tag{10}$$

where $\varphi^f(*)$ and $\varphi^a(*)$ are two linear transformation functions in the Focal-Focal Graph and the Focal-All Graph, respectively, to transform the messages to the original node embedding space.

By adopting the proposed local graph model, the computational complexity of modeling the context propagation in light field images is reduced from $O(((N+1)(HW))^2C)$ to $O(NHWC(N+1)k^2)$. Considering $k^2 \ll HW$, our model shows significant efficiency.

Multiscale surrounding area: With the introduced surrounding area, the proposed two graph networks can aggregate information in a local region, which can reduce computational costs dramatically. However, only using one sampling window is sensitive to scale variations. Motivated by ASPP [4], we combine multiple sampling windows with different dilation rates to incorporate multiscale and larger contexts, as shown in Figure 3, in which we use two 3×3 sampling windows with dilation rates of 1 and 3, respectively.

3.4. Focal Stack Feature Aggregation

After updating each node embedding h'_u in the DLG model, we can obtain the final output focal stack feature F'_f in (3). The features of different focal slices have communicated with each other and received guidance from the all-focus feature. Now we can tell useful and useless features among them and aggregate N feature maps into one. First, we use a 1×1 convolutional layer to reduce the channel number of F'_f from C to 1. Then, the Softmax normalization function is used along the first dimension to obtain an attention matrix $A \in \mathbb{R}^{N \times 1 \times H \times W}$, where the N-dimensional attention weights at each location encode the usefulness of each focal slice at this location. The final aggregated focal stack feature can be obtained by:

$$\begin{aligned} O &= FSFA(F'_f), \\ &= \sum_{i=1}^{N} A^i \odot F'_{f_i}, \end{aligned} \tag{11}$$

where $O \in \mathbb{R}^{C \times H \times W}$, $\odot$ is the element-wise multiplication, A^i and F'_{f_i} mean the attention map and the feature map for the i^{th} focal slice, respectively.

3.5. Reciprocative Guidance

Although the aggregated focal stack feature O can be directly fused with the all-focus feature F_a for predicting the saliency map, we argue that the single-phase fusion scheme can not effectively mine complex interactions and supplements between the two kinds of features, which are crucial for light field SOD. Hence, we propose a reciprocative guidance scheme to make the two kinds of features promote each other for multiple steps.

Here, to avoid confusion, we redefine the outputs from the encoders as $F^0_f = \{F^0_{f1}, F^0_{f2}, ...F^0_{fN}\}$ and F^0_a, where the superscripts represent the initial reciprocative step. Then, we define the proposed reciprocative guidance process as:

$$F_f^{t+1} = DLG(F_f^t, F_a^t), \tag{12}$$

$$O^{t+1} = FSFA(F_f^{t+1}), \tag{13}$$

$$F_a^{t+1} = ConvGRU(O^{t+1}, F_a^t), \tag{14}$$

where $t \in [0, T-1]$, $ConvGRU$ is the convolutional gated recurrent unit model [5].

In each step, the all-focus feature F_a^t is first used to guide the feature fusion of the focal features F_f^t. After the graph model and the feature aggregation, the aggregated focal stack feature O^{t+1} is further used to enhance F_a^t for saliency detection via ConvGRU. At last, ConvGRU can effectively fuse the two kinds of features, *i.e.*, O^{t+1} and F_a^t, in all reciprocative steps using the memory mechanism. As the reciprocative process goes on, the two features can be improved step by step under the guidance of each other, thus benefiting the final saliency detection. On the other hand, as the reciprocative process goes on, the context propagation in the focal-focal graph can be performed multiple times, hence also enhancing the feature fusion within F_f^t.

3.6. Saliency Prediction and Loss Function

Since it has been proved that low-level features can benefit the recovery of object details, we also leverage the low-level all-focus feature to perform saliency map refinement after the reciprocative guidance process. Specifically, we use a skip-connection to incorporate the all-focus feature from the first stage of the encoder VGG network and sum it with the upsampled F_a^T. Then, we perform feature fusion at the 1/2 scale via three 3×3 convolutional layers with ReLU activation functions. After that, another 3×3 convolutional layer with the Sigmoid activation function is used to obtain the final saliency map, as shown in Figure 2.

After each reciprocative process, we can obtain an enhanced feature F_a^t. In order to guide our model to gradually enhance the image features, we add a 1×1 convolutional layer with a sigmoid active function on F_a^t to predict a saliency map. Then we employ the binary cross-entropy loss to supervise the training of the t-th reciprocative step. Finally, the overall loss is the summation of each loss at each step.

4. Experiments

4.1. Datasets

Our experiments are conducted on three public light field benchmark datasets: LFSD [21], HFUT [47], and DUTLF-FS [38]. DUTLF-FS is the largest dataset that contains 1462 light field images and is split into 1000 and 462 images for training and testing, respectively. HFUT and LFSD are relatively small, containing only 255 and 100 samples, respectively. Each sample includes an all-focus image, several focal slices, and the corresponding ground-truth saliency map.

4.2. Evaluation Metrics

We follow many previous works to adopt the maximum F-measure (F_β) [1], S-measure (S_α) [8], the maximum E-measure (E_ϕ) [9], and the Mean Absolute Error (MAE) to evaluate the performance of different models in a comprehensive way.

4.3. Implementation Details

We design two sampling windows with size $k = 3$ and dilation rates $d = 1, 3$ in DLG, and set the reciprocative step number T as 5 based on experiments. For a fair comparison, we use the same training set with [33], which includes the training set of DUTLF-FS and 100 samples selected from HFUT. We also augment the training data with random flipping, cropping, and rotation. We use Adam [19] as the optimization algorithm and set the learning rate to 1e-4. The minibatch size is set to 1 and our network is trained for 200,000 steps. The learning rate is multiplied by 0.1 at the 150,000 and 180,000 steps, respectively. In both training and testing, we resize images to 256×256 for easy implementation. The proposed method is implemented using the Pytorch toolbox [30] and all experiments are conducted on one RTX 2080Ti GPU. The inference time of our model averaged on all three datasets is only 0.07s per image. Our code is publicly available at: `https://github.com/wangbo-zhao/2021ICCV-DLGLRG`.

4.4. Comparison with State-of-the-art Methods

Quantitative Comparison: For comprehensive comparisons, we compare our method with 19 state-of-the-art models, including six RGB SOD methods: LDF [42], ITSD [55], MINet [29], EGNet [53], PoolNet [24] and PiCANet [25], six RGB-D SOD models: BBS [10], SSF [51], S2MA [27], ATSA [49], JLDCF [11] and UCNet [44], and seven light field SOD methods: ERNet [33], MAC [45], MoLF [50], DLSD [32], LFS [22], WSC [20], and DILF [46].

As shown in Table 1, our method can achieve the best performance on DUTLF-FS and LFSD compared with all RGB, RBG-D, and light field methods. When it comes to the HFUT, our method can surpass other methods in terms of MAE, but performs worse in terms of the other three metrics. We argue that this is because many images in HFUT have uncommon SOD annotations, such as numbers and texts, which are rarely related to focus information. Thus, our model is not good at handling them.

It is noteworthy that on DUTLF-FS and LFSD, our method significantly outperforms ERNet [33], MoLF [50] and DLSD [32], which all use ConvLSTM models. This result demonstrates the superiority of our proposed reciprocative scheme. We also note that with only 1100 training samples, our method can achieve better performance on DUTLF-FS and LFSD than most deep RGB-D and RGB

Table 1. Benchmarking results. ↑ & ↓ denote larger and smaller is better, respectively. The best scores in each group are indicated in **blue**, and the best scores in all groups are indicated in **red**. * means we do not compare with this model due to incompatible training/test split.

			HFUT [47]				DUTLF-FS [38]				LFSD [21]			
	Methods	Years	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	$MAE \downarrow$
Light Field	Ours	-	0.766	0.697	0.839	**0.071**	**0.928**	**0.936**	**0.959**	**0.031**	**0.867**	**0.870**	**0.906**	**0.069**
	ERNet [33]	2020	**0.778**	**0.722**	**0.841**	0.082	0.899	0.908	0.949	0.039	0.832	0.850	0.886	0.082
	MAC [45]	2020	0.731	0.667	0.797	0.107	0.804	0.792	0.863	0.102	0.782	0.776	0.832	0.127
	MoLF [50]	2019	0.742	0.662	0.812	0.094	0.887	0.903	0.939	0.051	0.835	0.834	0.888	0.089
	DLSD [32]	2019	0.711	0.624	0.784	0.111	*	*	*	*	0.786	0.784	0.859	0.117
	LFS [22]	2017	0.565	0.427	0.637	0.221	0.585	0.533	0.711	0.227	0.681	0.744	0.809	0.205
	WSC [20]	2015	0.613	0.508	0.695	0.154	0.657	0.621	0.789	0.149	0.700	0.743	0.787	0.151
	DILF [46]	2015	0.675	0.595	0.750	0.144	0.654	0.585	0.757	0.165	0.811	0.811	0.861	0.136
RGB-D	BBS [10]	2020	0.751	0.676	0.801	**0.073**	0.865	0.852	0.900	0.066	**0.864**	0.858	0.900	0.072
	SSF [51]	2020	0.725	0.647	0.816	0.090	0.879	0.887	0.922	0.050	0.859	**0.868**	0.901	**0.067**
	S2MA [27]	2020	0.729	0.650	0.777	0.112	0.787	0.754	0.839	0.102	0.837	0.835	0.873	0.094
	ATSA [49]	2020	0.772	**0.729**	0.833	0.084	**0.901**	**0.915**	**0.941**	**0.041**	0.858	0.866	**0.902**	0.068
	JLDCF [11]	2020	**0.789**	0.727	**0.844**	0.075	0.877	0.878	0.925	0.058	0.862	0.867	**0.902**	0.070
	UCNet [44]	2020	0.748	0.677	0.804	0.090	0.831	0.816	0.876	0.081	0.858	0.859	0.898	0.072
RGB	LDF [42]	2020	0.780	0.708	0.804	0.093	0.873	0.861	0.898	0.061	0.821	0.803	0.843	0.096
	ITSD [55]	2020	**0.805**	**0.759**	**0.839**	0.089	**0.899**	**0.899**	**0.930**	0.052	0.847	0.840	0.879	0.088
	MINet [29]	2020	0.792	0.720	0.816	**0.086**	0.890	0.882	0.916	**0.050**	0.834	0.828	0.861	0.091
	EGNet [53]	2019	0.769	0.676	0.796	0.092	0.886	0.868	0.910	0.053	0.843	0.821	0.872	0.083
	PoolNet [24]	2019	0.769	0.676	0.794	0.091	0.883	0.859	0.911	0.051	**0.858**	**0.848**	**0.894**	**0.074**
	PiCANet [25]	2018	0.783	0.715	0.816	0.107	0.876	0.865	0.907	0.072	0.832	0.834	0.866	0.103

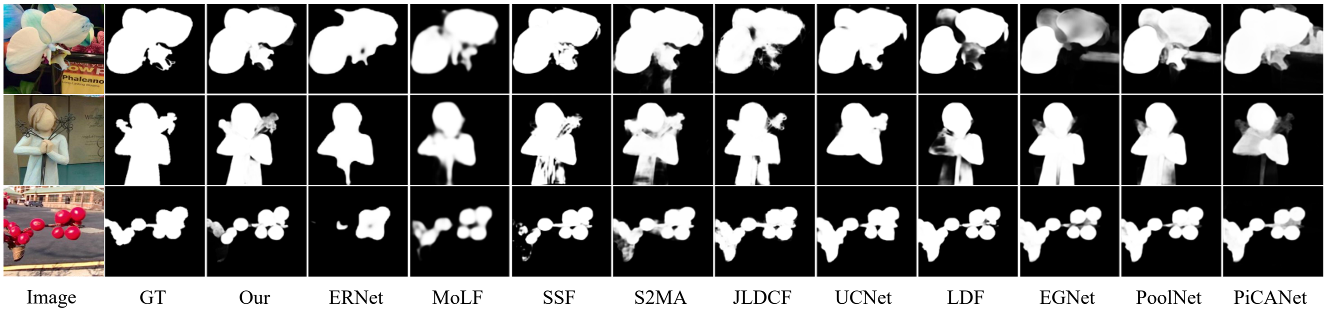


Figure 4. Visual comparison of the saliency maps between our model and state-of-the-art methods.

SOD methods, which are usually trained on much more images. This indicates that our method can effectively explore the information conveyed in light field data.

Qualitative Comparison: In Figure 4, we visualize some representative saliency map comparison cases. We can find that, compared with other SOTA methods, our model can not only more accurately localize salient objects, but also more precisely recover object details.

4.5. Ablation Study

In this section, we conduct ablation experiments on the largest DUTLF-FS dataset to thoroughly analyze our proposed model.

Effectiveness of Different Model Components. We first verify the effectiveness of our different model components in Table 2. For fair comparisons, we keep our feature encoders in Section 3.2 unchanged and try different decoder architectures to fuse the focal stack feature F_f and the all-focus feature F_a.

We first report the results of two baseline models of fusing F_f and F_a using naive concatenation and LSTM, respectively. For the first one, we follow many previous methods to randomly replicate focal slices in each focal stack to 12 images, then $F_f \in \mathbb{R}^{12\times C\times H\times W}$. Next, we concatenate F_f with F_a and use convolution to fuse these 13 feature maps. We denote this strategy as "Enc-concat". For the second one, we use a ConvLSTM to directly fuse the $N+1$ feature maps in F_f and F_a, which is denoted as "Enc-lstm". We find that using LSTM performs better for fusing the two kinds of features.

Table 2. Quantitative results of using different feature fusion strategies. "Enc" means our feature encoders in Section 3.2, "R" denotes our proposed reciprocative guidance scheme, and "r" means using the low-level all-focus feature to refine the saliency map. **Blue** indicates the best performance.

Settings	DUTLF-FS			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	MAE ↓
Enc-concat	0.891	0.898	0.934	0.062
Enc-lstm	0.900	0.909	0.940	0.047
Enc-DLG	0.907	0.911	0.944	0.044
Enc-DLG-R	0.923	0.932	0.957	0.035
Enc-DLG-R-r	**0.928**	**0.936**	**0.959**	**0.031**

Next, we progressively adopt our proposed DLG model, the reciprocative guidance scheme, and the refinement decoder using the low-level all-focus feature. These three

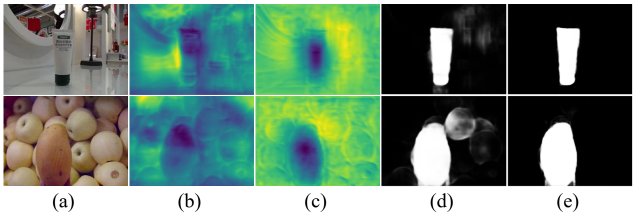


Figure 5. Comparison of models with and without using the proposed DLG model in terms of both feature maps and saliency maps. (a): Images. (b): Feature maps w/o using DLG. (c): Feature maps w using DLG. (d): Saliency maps w/o using DLG. (e): Saliency maps w using DLG.

models are denoted as "Enc-DLG", "Enc-DLG-R", and "Enc-DLG-R-r", respectively. From Table 2, we can see that the three models can progressively improve the light field SOD performance, finally outperforming the two baseline models by a large margin. Using the DLG model achieves better results than using naive concatenation and LSTM, and also avoids their ill-posed implementation problem. We also try to use a densely connected graph network to fuse the $N+1$ feature maps, but only to obtain the out of memory error. This result proves the efficiency of our DLG model. Furthermore, we find that the reciprocative guidance scheme brings the largest model improvement, clearly demonstrating its powerful capacity. We believe this strategy can also benefit future light field SOD research a lot.

We also show the comparison of the feature maps and saliency maps with and without using the DLG model in Figure 5. We can see that by using DLG, the feature maps can filter out distractions in backgrounds and focus more on the salient objects, hence resulting in better SOD results.

DLG Settings. Since we build multiscale local neighbours in DLG to introduce larger contexts with acceptable computational costs, we also explore different multiscale settings in DLG in Table 3. Specifically, we test different settings of the sampling window size k and the dilation rates d in the "Enc-DLG-R" model. We start from the naive setting with 1×1 sampling window. From Table 3, we can find that, when we use more and larger windows, the performance of our model can be gradually improved. However, the performance is saturated when using two 3×3 sampling windows with dilation rates of 1 and 3. Further using one more window with $d = 5$ only brings little improvement. Considering the computational costs, we choose $k = 3$ and $d = 1, 3$ as our final setting.

To verify the effectiveness of simultaneously using the focal-focal graph $\mathcal{G}^f$ and the focal-all graph $\mathcal{G}^a$, we try to use them separately and report the results in the last two rows in Table 3. We find that using them separately will degrade the model performance, hence verifying the necessity of our proposed dual graph scheme.

Reciprocative Steps. We conduct experiments to choose the optimal reciprocative step number T in Table 4. Note that, when $T = 1$, the model downgrades to the "Enc-DLG" model. We find that, as we increase T from 1 to 5, the performance can be improved progressively. When $T > 5$, we observe that the performance has been saturated and the model will exceed the GPU memory. Hence, we take $T = 5$ as the final setting for our reciprocative guidance scheme.

Table 3. Comparison of using different DLG Settings.

Settings				DUTLF-FS			
k	d	$\mathcal{G}^f$	$\mathcal{G}^a$	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	MAE $\downarrow$
1	1	✓	✓	0.914	0.919	0.946	0.042
3	1	✓	✓	0.915	0.919	0.950	0.038
3	1,3	✓	✓	0.923	0.932	0.957	**0.035**
3	1,3,5	✓	✓	**0.924**	**0.933**	**0.960**	**0.035**
3	1,3	✓	✗	0.917	0.929	0.953	0.038
3	1,3	✗	✓	0.919	0.927	0.952	0.037

Table 4. Comparison of using different reciprocative step numbers.

T	DUTLF-FS			
	$S_\alpha \uparrow$	$F_\beta \uparrow$	$E_\phi \uparrow$	MAE $\downarrow$
1	0.907	0.911	0.944	0.044
3	0.917	0.924	0.952	0.040
5	**0.923**	**0.932**	**0.957**	**0.035**

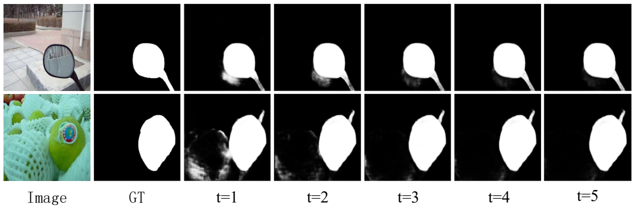


Figure 6. Visualization of the saliency maps at different reciprocative steps.

We also visualize two representative samples to show the improvements of the saliency maps obtained at different reciprocative steps in Figure 6. We can find that, along with the reciprocative guidance learning, false-positive highlights can be gradually suppressed and the SOD results can be steadily improved.

5. Conclusion

In this paper, we propose a novel dual local graph neural network and a reciprocative guidance architecture for light field SOD. Our DLG model efficiently aggregates contexts in focal stack images under the guidance of the all-focus image. The reciprocative guidance scheme introduces iterative guidance between the two kinds of features, making them promote each other at multiple steps. Experimental results show that our method achieves superior performance over state-of-the-art RGB, RGB-D, and light field based SOD methods on most datasets.

Acknowledgments: This work was supported in part by the National Key R&D Program of China under Grant 2020AAA0105701, the National Science Foundation of China under Grant 62027813, 62036005, U20B2065, U20B2068.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009.

[2] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. *arXiv preprint arXiv:1806.09835*, 2018.

[3] Hao Chen and Youfu Li. Progressively complementarity-aware fusion network for rgb-d salient object detection. In *CVPR*, pages 3051–3060, 2018.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[7] 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 *AAAI*, pages 684–690. AAAI Press, 2018.

[8] 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.

[9] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018.

[10] Deng-Ping Fan, Yingjie Zhai, Ali Borji, Jufeng Yang, and Ling Shao. Bbs-net: Rgb-d salient object detection with a bifurcated backbone strategy network. In *ECCV*, pages 275–292. Springer, 2020.

[11] Keren Fu, Deng-Ping Fan, Ge-Peng Ji, and Qijun Zhao. Jl-dcf: Joint learning and densely-cooperative fusion framework for rgb-d salient object detection. In *CVPR*, pages 3052–3062, 2020.

[12] Keren Fu, Yao Jiang, Ge-Peng Ji, Tao Zhou, Qijun Zhao, and Deng-Ping Fan. Light field salient object detection: A review and benchmark. *arXiv preprint arXiv:2010.04968*, 2020.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[14] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *IJCNN*, volume 2, pages 729–734. IEEE, 2005.

[15] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674*, 2017.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[18] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *CVPR*, pages 3203–3212, 2017.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. In *CVPR*, pages 5216–5223, 2015.

[21] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *CVPR*, pages 2806–2813, 2014.

[22] N Li, J Ye, Y Ji, H Ling, and J Yu. Saliency detection on light field. *TPAMI*, 39(8):1605–1616, 2017.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[24] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, pages 3917–3926, 2019.

[25] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *CVPR*, pages 3089–3098, 2018.

[26] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Pixel-wise contextual attention learning for accurate saliency detection. *TIP*, 2020.

[27] Nian Liu, Ni Zhang, and Junwei Han. Learning selective self-mutual attention for rgb-d saliency detection. In *CVPR*, pages 13756–13765, 2020.

[28] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, Hong Cheng, and Siwei Lyu. Cascade graph neural networks for rgb-d salient object detection. In *ECCV*, pages 346–364. Springer, 2020.

[29] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, pages 9413–9422, 2020.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NeurIPS Workshop*, 2017.

[31] Yongri Piao, Wei Ji, Jingjing Li, Miao Zhang, and Huchuan Lu. Depth-induced multi-scale recurrent attention network for saliency detection. In *ICCV*, pages 7254–7263, 2019.

[32] Yongri Piao, Zhengkun Rong, Miao Zhang, Xiao Li, and Huchuan Lu. Deep light-field-driven saliency detection from a single view. In *IJCAI*, pages 904–911, 2019.

[33] Yongri Piao, Zhengkun Rong, Miao Zhang, and Huchuan Lu. Exploit and replace: An asymmetrical two-stream architecture for versatile light field saliency detection. In *AAAI*, pages 11865–11873, 2020.

[34] Sungmin Rhee, Seokjun Seo, and Sun Kim. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. *arXiv preprint arXiv:1711.05859*, 2017.

[35] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[37] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841. Springer, 2016.

[38] Tiantian Wang, Yongri Piao, Xiao Li, Lihe Zhang, and Huchuan Lu. Deep learning for light field saliency detection. In *ICCV*, pages 8838–8848, 2019.

[39] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *TPAMI*, 2021.

[40] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, pages 9236–9245, 2019.

[41] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pages 399–417, 2018.

[42] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *CVPR*, pages 13025–13034, 2020.

[43] 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 *NeurIPS*, pages 802–810, 2015.

[44] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Tong Zhang, and Nick Barnes. Uc-net: uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In *CVPR*, pages 8582–8591, 2020.

[45] Jun Zhang, Yamei Liu, Shengping Zhang, Ronald Poppe, and Meng Wang. Light field saliency detection with deep convolutional networks. *TIP*, 29:4421–4434, 2020.

[46] 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.

[47] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. Saliency detection on light field: A multicue approach. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 13(3):1–22, 2017.

[48] Kaihua Zhang, Tengpeng Li, Shiwen Shen, Bo Liu, Jin Chen, and Qingshan Liu. Adaptive graph convolutional network with attention graph clustering for co-saliency detection. In *CVPR*, pages 9050–9059, 2020.

[49] Miao Zhang, Sun Xiao Fei, Jie Liu, Shuang Xu, Yongri Piao, and Huchuan Lu. Asymmetric two-stream architecture for accurate rgb-d saliency detection. In *ECCV*, 2020.

[50] Miao Zhang, Jingjing Li, Ji Wei, Yongri Piao, and Huchuan Lu. Memory-oriented decoder for light field salient object detection. In *NeurIPS*, pages 898–908, 2019.

[51] Miao Zhang, Weisong Ren, Yongri Piao, Zhengkun Rong, and Huchuan Lu. Select, supplement and focus for rgb-d saliency detection. In *CVPR*, pages 3472–3481, 2020.

[52] Jia-Xing Zhao, Yang Cao, Deng-Ping Fan, Ming-Ming Cheng, Xuan-Yi Li, and Le Zhang. Contrast prior and fluid pyramid integration for rgbd salient object detection. In *CVPR*, pages 3927–3936, 2019.

[53] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, pages 8779–8788, 2019.

[54] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51. Springer, 2020.

[55] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *CVPR*, pages 9141–9150, 2020.

[56] Tao Zhou, Deng-Ping Fan, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Rgb-d salient object detection: A survey. *Computational Visual Media*, pages 1–33, 2021.