Saliency Detection on Light Field

Nianyi Li, *Student Member, IEEE*, Jinwei Ye, *Member, IEEE*, Yu Ji, *Student Member, IEEE*, Haibin Ling, *Member, IEEE*, and Jingyi Yu, *Member, IEEE*

Abstract—Existing saliency detection approaches use images as inputs and are sensitive to foreground/background similarities, complex background textures, and occlusions. We explore the problem of using light fields as input for saliency detection. Our technique is enabled by the availability of commercial plenoptic cameras that capture the light field of a scene in a single shot. We show that the unique refocusing capability of light fields provides useful focusness, depths, and objectness cues. We further develop a new saliency detection algorithm tailored for light fields. To validate our approach, we acquire a light field database of a range of indoor and outdoor scenes and generate the ground truth saliency map. Experiments show that our saliency detection scheme can robustly handle challenging scenarios such as similar foreground and background, cluttered background, complex occlusions, etc., and achieve high accuracy and robustness.

Index Terms—Saliency detection, light field, Lytro, focus stack

1 Introduction

Salient region detection is a long standing problem in computer vision. It aims to locate pixels or regions in an image that most attract human's visual attention. Accurate and reliable saliency detection can benefit numerous tasks ranging from tracking and recognition in vision to image manipulation in graphics. For example, successful salient object detection algorithms facilitate automated image segmentation [1], more reliable object detection [2], effective image thumbnailling [3] and retargeting [4].

State-of-the-art solutions [5] have focused on integrating low-level features (pixels or superpixels) and high-level descriptors (regions or objects). However, existing solutions have many underlying assumptions, e.g., the foreground should have a different color from the background, the background should be relatively simple and smooth, the foreground is occlusion free, etc. In reality, many real images violate one or multiple assumptions as shown in Fig. 1.

By far, nearly all existing saliency detection algorithms utilize images acquired by a regular camera. In this paper, we explore the salient object detection problem by using a completely different input: the light field of a scene. A light field [8] can be essentially viewed as an array of images captured by a grid of cameras towards the scene. Commercial light field cameras can now capture reasonable quality light fields in a single shot. Lytro, for example, mounts a lenslet array in front of the sensor (as shown in Fig. 2a) to acquire a light field at a 360×360 (upsampled to $1{,}080 \times 1{,}080$) spatial resolution and 10×10 angular resolution. The Raytrix R11 camera can produce a higher spatial resolution at the cost of lower angular resolution. The multi-view nature of the light field has enabled new generations of stereo matching [9] and object segmentation algorithms [10]. In this paper, we explore how to conduct salient object detection using a light field camera.

Human vision system has the refocusing ability which can help us pay more attention to the interesting objects, since the other objects are blurred when our eye focus on certain object [11]. Due to above reason, we can easily distinguish the interesting object, i.e., the salient object, regardless the texture or the color of other objects in the scene, i.e., the background. When it comes to detect the salient object from images, several problems will be arisen if the objects in the image have similar color or texture appearance, as shown in Fig. 1.

Conceptually, the light field data can benefit saliency detection in a number of ways. First, the light field has a unique capability of post-capture refocusing [12], i.e., it can synthesize a stack of images focusing at different depths. As shown in Fig. 2b, we can always find right layers which focus on the salient object within focus stack. If we can pick out the right layers which only focus on foreground, the salient object detection problem will be equal to the focus measures algorithms. The availability of a focal stack is inline with the recently proposed "focusness" metric [13]. It is the reciprocal of blurriness and can be estimated in terms of edge scales via scale-space analysis. Second, a light field provides an approximation to scene depth and occlusions. In saliency detection, even a moderately accurate depth map can greatly help distinguish the foreground from the background. This is also inline with the "objectness" [13], i.e., a salient region should complete objects instead of cutting them into pieces.

In addition to focusness and objectness, we also exploit the recent background prior [14]. Instead of directly detecting salient regions, such algorithms aim to first find the background and then use it to prune non-salient objects.

- *N. Li, J. Ye, Y. Ji, and J. Yu are with University of Delaware, Newark, DE 19716. E-mail: {nianyi, jye, yuji, yu}@eecis.udel.edu.*
- *H. Ling is with Temple University, Philadelphia, PA 19122. E-mail: hbling@temple.edu.*

Manuscript received 19 Mar. 2015; revised 1 Aug. 2016; accepted 7 Sept. 2016. Date of publication 15 Sept. 2016; date of current version 11 July 2017. Recommended for acceptance by J. Jia.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2610425

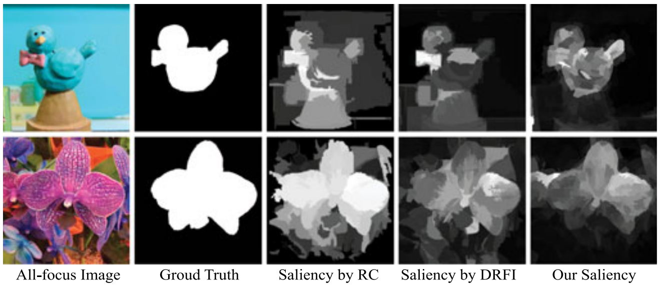


Fig. 1. Light field versus traditional saliency detection. Similar foreground and background or complex background imposes challenges on state-of-the-art algorithms (e.g., RC [6], DRFI [7]). Using light field as inputs, our saliency detection scheme is able to robustly handle these cases.

Robust background detection, however, is challenging, especially when the foreground and background have similar appearance or the background is cluttered. To resolve this problem, we utilize the *focusness and objectness* to more reliably choose the background and select the foreground saliency candidates. Specifically, we compute a *foreground likelihood score (FLS)* and a *background likelihood score (BLS)* by measuring the focusness of pixels/regions. We select the layer with the highest BLS as the background and use it to estimate the background regions. In addition, we choose regions with a high FLS as candidate salient objects. Finally, we conduct contrast-based saliency detection on the all-focus image and combine its estimation with the detected foreground saliency candidates.

For validation, we acquire a light field database of a range of indoor and outdoor scenes and generate the ground truth saliency map. We have already shared this database, i.e., Light Field Saliency Detection (LFSD) Dataset, to community online.[1] Experiments show that our saliency detection scheme can robustly handle challenging scenarios such as similar foreground and background, cluttered background, and images with multiple depth layers and with heavy occlusions, etc., and achieve high accuracy and robustness. In addition, the comparison results show that our focusness cues using light field are more effective than or equally as good as other state-of-arts depth cues.

A preliminary version of this work appeared in [15].

2 Related Work

The saliency detection literature is huge and existing solutions can be classified in terms of top-down versus bottom-up, center versus background prior, with versus without depth cue, etc. Readers can refer to [5] for a comprehensive comparisons on state-of-the-art solutions. We discuss the most relevant ones.

2.1 Top-Down versus Bottom-Up

Top-down approaches [16], [17] use visual knowledge commonly acquired through learning to detect saliency. Approaches in this category are highly effective on task-specified saliency detection, e.g., identifying human activities [18]. However, a large number of annotated images need to be used for training. Bottom-up methods do not require training and rely on low-level features such as color contrast [19], pixel/patch locations [20], histogram [21], etc., for saliency detection. Our approach falls into the category of bottom-up approaches where we add an additional class of focus-related cues.

2.2 Center versus Background Priors

Many saliency detection schemes exploit contrast cues, i.e., salient objects are expected to exhibit high contrast within certain context. Koch and Itti [19] are the first to use center-surround contrast of low level features to detect saliency. Motivated by their work, many existing approaches compute the center-surround contrast either locally or globally. Local methods compute the contrast within a small neighborhood of pixels by using color difference [22], edge orientations [23], or curvatures [24]. Global methods consider statistics of the entire image and rely on features such as power spectrum [25], color histogram [26], and element distributions [20].

Although the center-surround approaches are proven highly effective, Wei et al. [14] suggested that background priors are equally important. In fact, one can eliminate the background to significantly improve foreground detection. Yang et al. [27] observed that connectivity is an important characteristics of background and used a graph-based ranking scheme to measure patch similarities. Since most existing approaches rely on color contrast, when the foreground and background have similar color, these approaches can easily fail. Our approach resolves this issue by combining color contrast, background prior, and focusness prior w.r.t. different depth layers obtain from the light field.

2.3 Focusness and Objectness Cue

Jiang et al. [13] proposed that objects of interest in an image are often photographed in focus. This naturally associates the focusness with that saliency. They estimated the focusness by the scale of edges using scale-space analysis. In addition, they also proposed an objectness estimation which utilized the probability of a region belongs to a complete object in some local windows to measure. Regarding our techniques versus [13], we want to emphasize that our scheme is advantageous over [13] in several ways. First, our focusness cue is extracted directly from a complete focal stack produced by the 4D light field whereas the cue has to be inferred from a single image in [13]. Therefore, our technique is more robust and reliable especially on the images that contain similar foreground/background and/or lack defocus cues. Second, the availability of light fields facilitates easier an effective extraction of location, contrast and foreground cues. These cues, in many ways, serve the similar purpose of uniqueness and objectness cues in [13] but are more robust. Third, our objectness cue is concerned as to an focus stack slice not to a certain region or pixels, which accelerates the computational speed.

2.4 Depth Cue

Recent studies on human perception [28] have shown that depth cue plays a important role in determining salient regions. However, only a handful of works incorporate depth maps into saliency models. Maki et al. [18] used depth cue to detect human motions. Their depth features are highly task-dependent and the detection is performed in

[1] http://www.eecis.udel.edu/~nianyi/LFSD.htm

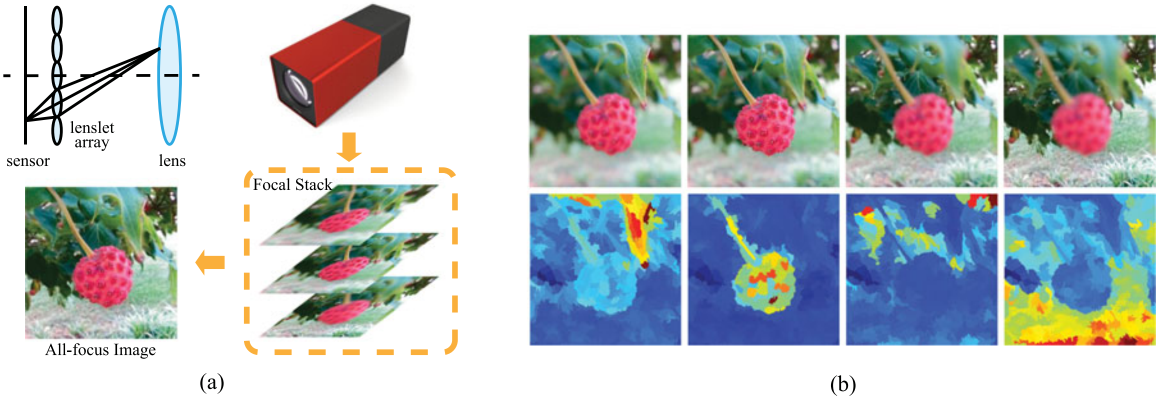


Fig. 2. (a) A Lytro light field camera can capture a light field towards the scene in a single shot. The results can be then used to synthesize a focal stack and further a all-focus image. (b) Focus stack (the first row) and its corresponding focus regions (second row).

a top-down fashion. Niu et al. [29] computed saliency based on the global disparity contrast in a pair of stereo images. Lang et al. [28] used a Kinect sensor to capture the scene depth. Ciptadi et al. [30] used 3D layouts and shape features from depth maps. Peng et al. [31] detected saliency taking account of both depth and appearance cues derived from low-level feature contrast, mid-level region grouping and high-level priors enhancement. In this paper, we exploit rich depth information embedded in the light field. Specifically, we use coarse depth information embedded in a focal stack to guide saliency detection. To achieve more accurate result, most depth cue based schemes need relatively accurate depth maps. In reality, depth estimation from images (e.g., stereo) remains challenging in both computational cost and accuracy on real scene. One can alternatively resort to active sensing (e.g., structured light or time-of-flight). However, such schemes also have their limitations such as limited depth range and interference with environment lighting. Focus stack rendering, on the other hand, is more intuitive and precise. Also, isolating different objects by depth map is more likely to break object into pieces, as we have no prior depth information of each object. Our proposed scheme aims to resemble human perception using eye: the eyes can dynamically refocus at different slices to determine saliency. This can be done by constructing focus stack using light field rendering approach. Detecting on focus stack, on the other hand, is more likely to preserve better objectness of salient object than depth map, if salient objects have narrow depth range compared with the depth range of complete scene. More detailed discussion can be found in Section 3.4.

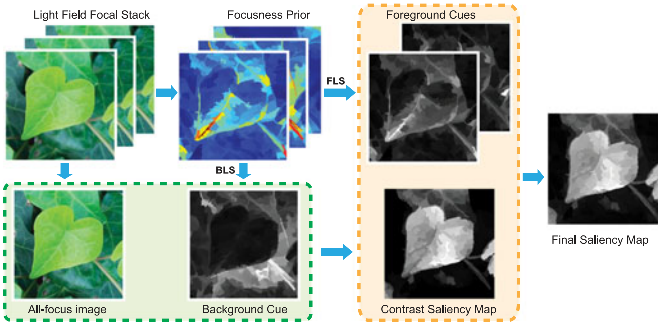


Fig. 3. Processing pipeline of our saliency detection algorithm for light fields.

3 Computing Light Field Saliency Cues

Fig. 3 shows our saliency detection approach using the light field. We first generate a focal stack and an all-focus image through light field rendering. For each image in the focal stack, we detect the in-focus regions and use them as the focusness measure. Next, we combine the focusness measure with the location prior to extract the background and the foreground salient candidates. We further couple the background prior with contrast-based saliency detection for detecting saliency candidates in the all-focus image. Finally, we use the objectness as weights for combining the saliency candidates from the all-focus image and from the focal stack as the final saliency map.

3.1 Focal Stack and All-Focus Images

A unique capability of light field is after-capture refocusing. Here we briefly reiterate its mechanism. A light field stores regularly sampled views looking towards the scene on a 2D sampling plane. These views form a 4D ray database and new views can be synthesized by querying existing rays. Given the light field of a scene, one can synthesize a Depth-of-Field (DoF) effects by selecting appropriate rays from the views and blending them, as shown in Fig. 2a. Isaksen et al. [32] proposed to render DoF by reparameterizing the rays onto the focal plane and blending them via a wide aperture filter. Ng et al. [12] proposed a similar technique in the Fourier space and the solution has been adopted in the Lytro light field camera. Using the focal stack, we can fuse an all-focus image, e.g., through photomontage [33]. We refer the readers to the comprehensive survey on light field imaging [34], [35] for more details about the refocusing algorithm.

In this paper, we use the Lytro camera as the main imaging device to acquire the light field. The Lytro camera uses an array of 360 × 360 microlenses mounted on an 11 megapixel sensor, where each microlens resembles a pinhole camera. It can produce the refocused results at a resolution of 360 × 360.

We compose an all-focus image by focus fusion using existing online-tools[2] from the focal stack so that the all-focus image has the same resolution as the focal stack. In

[2] http://code.behnam.es/python-lfp-reader/

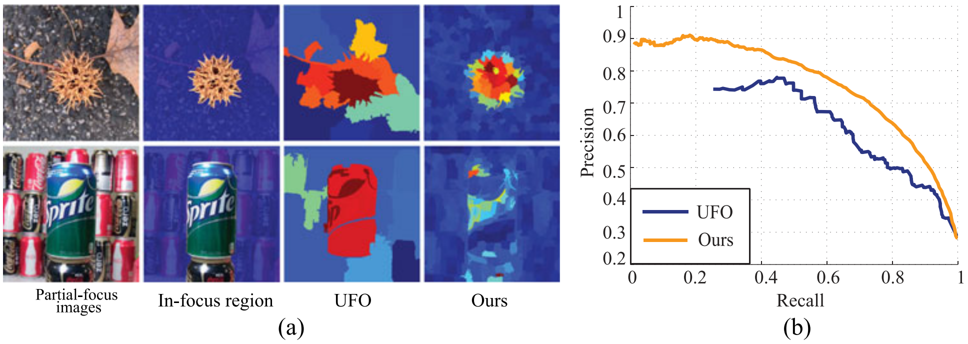


Fig. 4. Focusness detection comparison of UFO [13] versus ours. (a) Focusness detection results comparsion. (b) PRCs comparison.

addition, it is worth noting that DoF effect is not significant in Lytro focal stack due to small microlens baseline. As a result, each slice is just slightly defocused. Therefore, brute-force approaches such as applying saliency detection on each slice and then combine the results are not directly applicable since all slices will produce similar results.

Before proceeding, we explain our notation. We denote $\{I^i\}$, $i = 1, \ldots, N$ as the focal stack synthesized from the light field and I^* the all-focus image by fusing the focused regions of $\{I^i\}$. Our goal is to compute a saliency map w.r.t. I^*. We segment each slice $\{I^i\}$ and I^* into a set of small non-overlapping regions (superpixels) using the mean-shift algorithm [36]. This segmentation helps to preserve edge consistency and maintain proper granularity. We use (x, y) index a pixel and r to index to a region.

3.2 Focusness Measure

We start with detecting the in-focus regions in each focal stack image I^i and use them as the focusness prior. In the recent focusness-based saliency detection work, Jiang et al. [13] measured focusness via edge sharpness. However, edge-based in-focus detection is only reliable when the out-of-focus regions appear severely blurred. In our case, the DoF of Lytro is not as shallow as the one in DSLR. Therefore, edges in out-of-focus regions are not as blurred as in the classical datasets, as shown in Fig. 4a. It is hence difficult to use spatial algorithms to separate the in-focus/out-of-focus regions. Our approach is to analyze the image statistics in the frequency domain. In Fig. 4, we compare the saliency detection resutls versus the focusness measure both visually and quantitatively. Specifically, we select 80 focus slices that have a clear boundary between in-focus and defocused regions. We then segment out the in-focus region. Fig. 4 illustrates that the in-focus regions are often quite different from the actual saliency maps. It is also worth noting that [13] attempts to segment the complete in-focus object whereas our algorithm handels the focusness measures at region level. Consequently, [13] is more likely to over-segement in-focus regions, i.e., it will cut into part of the out-of-focus regions, as shown in Fig. 4a. Our method, on the other hand, processes superpixels and prevents over-segmentation.

Given an $n \times n$ image I, we first transform I into frequency domain by the Discrete Cosine Transform (DCT)

$$\mathcal{D}(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} \cos\left(\frac{\pi u}{2n}(2x+1)\right) \cos\left(\frac{\pi v}{2n}(2y+1)\right) I(x, y). \tag{1}$$

Next, we compute the image's response with respect to different frequency components. We first apply a series of M bandpass filters $\{P_m\}$, $m = 1, \ldots, M$ on $\mathcal{D}(u, v)$ for decomposing the signal and then transform the decomposed results back via the inverse DCT. Recall that out-of-focus blurs will remove certain high frequency components. Therefore, only regions with a sharp focus will have high responses at all frequencies. In our implementation, we use a sliding window of 8×8 pixels and compute the variance τ_m within each patch with respect to filter P_m. To ensure reliable focusness measurements, we use the harmonic variance [37] to measure the overall variance over all M filters

$$\mathcal{F}(x, y) = \left[\frac{1}{M-1} \sum_{m=1}^{M} \frac{1}{\tau_m^2(x, y)}\right]^{-1}. \tag{2}$$

We use $\mathcal{F}(x, y)$ as the focusness measure at pixel (x, y). Under this formulation, only when the response of all filters are high, the harmonic variance $\mathcal{F}(x, y)$ will be high. Any small τ_m will result in low $\mathcal{F}$. Therefore, this formulation ensures that only local windows preserving all frequency components would be deemed as in-focus. Since both DCT and harmonic variance computations are effective, we compute $\mathcal{F}$ for every pixel in the image. Finally, to measure the focusness of a region, we simply compute the average of all pixels within a region r

$$\mathcal{F}(r) = \sum_{(x,y) \in r} \frac{\mathcal{F}(x, y)}{A_r}, \tag{3}$$

where A_r is the total number of pixels in r. We will use this region-based focusness prior $\mathcal{F}(r)$ for selecting background and saliency candidates in Sections 3.3 and 3.4. It is worth noting that more sophisticated focusness estimation techniques such as scanning through the focal volume can be used. In practice, our measure is sufficient for the task of saliency detection and is much faster. Notice that harmonic variance would fail at detecting regions with single-directed edges or with homogeneous color, as can be seen from Fig. 4. However, like the blue bottle case in Fig. 14, the wrongly suppressed regions could be correctly highlighted as salient region in the final saliency map by incorporating color contrast cue. More details are discussed in Section 4.2.

3.3 Background Selection

Next, we set out to find the background slice. Notice that the background slice is *not* equivalent to the farthest slice in the

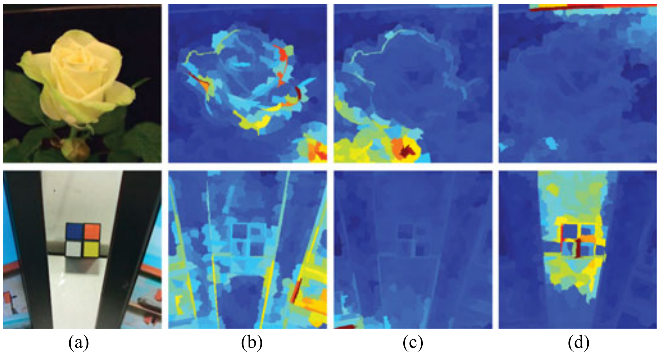


Fig. 5. Focusness detection result on focus stack. (a) All focus image. (b) Focusness map on the nearest objects. (c) Focusness map on objects at the middle of depth range. (c) Focusness map on the furthest objects.

focal stack. Recall that we synthesize the focal stack without any knowledge on scene depth range. Therefore, the farthest slice may not contain anything in focus and hence provides little cues, as shown in the first row of Fig. 5. Second, the slice that have the farthest object in focus does not necessarily translate to the background slice, like what the second example in Fig. 5 shows, the object may be isolated from majority of the background and should be treated as an outlier.

Our approach is to analyze both the distribution of the in-focus objects with respect to their locations in the image: if the majority of in-focus objects (pixels) lies near the border of the image, then they are more likely to belong to the background. Further, if the corresponding depth layer is far away, its in-focus objects are also more likely to be background. We therefore scan through all focal slices. For each slice I^i, we integrate (project) the focusness measure $\mathcal{F}$ of all pixels along the x and y axes respectively to form two 1D focusness distributions as

$$D_x = \frac{1}{\alpha}\sum_{y=1}^{h}\mathcal{F}(x,y), \quad D_y = \frac{1}{\alpha}\sum_{x=1}^{w}\mathcal{F}(x,y), \tag{4}$$

where w and h are the width and height of the image and $\alpha = \sum_x \sum_y \mathcal{F}(x,y)$ is the normalization factor.

A common assumption in saliency detection is that an salient object is more likely to lie at the central area surrounded by the background [14]. If a focal slice corresponds to the background, its D_x and D_y should be high near the endpoints but low in the middle. To quantitatively measure it, we define a "U-shaped" 1D band suppression filter

$$\mathcal{U}(x,w) = \left(\frac{1}{\sqrt{1+(x/\eta)^2}} + \frac{1}{\sqrt{1+((w-x)/\eta)^2}}\right), \tag{5}$$

where η controls the suppression bandwidth in $\mathcal{U}$ depending on the image size/resolution, i.e., a high resolution image should have a high η. The Lytro focal stack images have a uniform resolution of 360×360 and we use $\eta = 47$ in all experiments.

Finally, we scale the focusness distribution by the suppression filter to compute a Background Likelihood Score for each focal slice I^i

$$BLS(I^i) = \rho \cdot \left[\sum_{x=1}^{w} D_x^i(x)\cdot\mathcal{U}(x,w) + \sum_{y=1}^{h} D_y^i(y)\cdot\mathcal{U}(y,h)\right], \tag{6}$$

where $\rho = \exp(\frac{\lambda \cdot i}{N})$ is the weighting factor of layer i in terms of depth, N is the total number of slices in the focus stack and $\lambda = 0.3$. We choose the slice with the highest BLS as the background slice I^B. It is important to note that each focal slice has a corresponding BLS even though it is not chosen as I^B.

3.4 Objectness and Foreground Measures

Alexe et al. [38] suggested that a salient object should be complete instead of being broken into pieces and refer to this property as the objectness. Given a focal stack image I^i, we measure the objectness of its focused region using a 1D gaussian filter with mean μ and variance σ as

$$\mathcal{G}(x) = \exp\left(-\frac{x-\mu}{2\sigma^2}\right), \tag{7}$$

where μ corresponds to the centroid of the object and σ as its size. Recall that we have already computed the focusness distributions D_x or D_y. Therefore, we can directly obtain $\mu = x_p$ or y_p, that corresponds to the peak location of D_x or D_y respectively. If multiple peaks exist, we simply take their average.

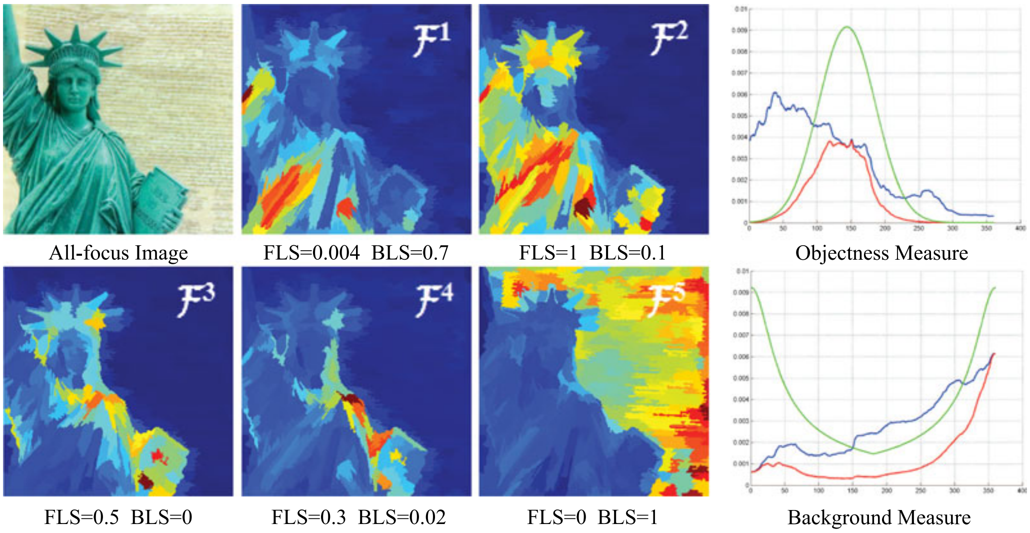
Next we estimate σ as the size of the object. If σ is too small, isolated small superpixels would be treated as an object. If σ is too large, i.e., it would treat the entire image as an object. In our implementation, we choose $\sigma = 45$, i.e., 50 percent Gaussian covers half of the D_x or D_y. We compute the objectness score (OS) for each focal slice

$$OS(I^i) = \sum_{x=1}^{w} D_x^i(x)\cdot\mathcal{G}(x,w) + \sum_{y=1}^{h} D_y^i(y)\cdot\mathcal{G}(y,h). \tag{8}$$

Conceptually, if an object in a given slice is salient, it should have a low BLS and high OS, indicating it belongs to the foreground. We therefore define a foreground likelihood score as

$$FLS(I^i) = OS(I^i)\cdot(1 - BLS(I^i)). \tag{9}$$

Same as how we select the background slice I^B, we choose the foreground slices $\{I^F\}$ as one with the higher FLS ($FLS > 0.7 \times max(FLS)$). Fig. 6 illustrates the process of finding the background and foreground slices on a sample image. Notice that salienct object can be separated into several layers, which might result in inaccurate FLS/BLS score for some focusness layers. For instance, the first layer $\mathcal{F}^1$ in Fig. 6 focuses on the salient object, but our algorithm regarded it more likely to be background layer. We would like to point out that not all slices focusing on foreground are good choices for $\{I^F\}$. In the $\mathcal{F}^1$ case, even though its highlighted regions belong to salient object, they are scattered around image boundary whereas our goal is to detect salient object as a whole. In reality, saliency objects have narrow depth range in regard to the depth range of the complete scene. This indicates that within a focal stack, there generally exists a slice where the entire salient object/region exhibits high sharpness, such as the second layer $\mathcal{F}^2$ in



All-focus Image FLS=0.004 BLS=0.7 FLS=1 BLS=0.1 Objectness Measure

FLS=0.5 BLS=0 FLS=0.3 BLS=0.02 FLS=0 BLS=1 Background Measure

Fig. 6. Separating the foreground and background using focusness cues. Left: The computed foreground likelihood score (FLS) and the background likelihood score (BLS) computed on different focal slices. Right: Examples on computing objectness measure (up) and background measure (bottom). Green curve is corresponding filter (U-shape or Gaussian); blue curve is sample D_x/D_y; red curve is the scaled distribution by the filter.

Fig. 6. Once we are able to select the correct candidate slices, i.e., slices with high FLS/BLS value, the inclusion of incorrect FLS/BLS will not greatly affect the final saliency result. In fact, inaccurate FLS will affect saliency detection only when the salient object has a large depth range in the complete scene.

4 Saliency Detection

Finally, we combine the cues obtained from the light field focal stack to detect saliency in the all-focus image I^*.

4.1 Location Cues

We first locate the background regions in I^* using the focusness measure $\mathcal{F}^B(r)$ of the estimated background slice I^B. To incorporate the location prior [20], we scale the focusness measure for each region R_r in terms of its distance to the center of the image and use it as a new background cue

$$BC(r) = \frac{1}{\gamma}[\mathcal{F}^B(r) \cdot ||\mathbf{p}_r - \mathbf{c}||^2], \tag{10}$$

where γ is a normalization factor, $\mathbf{p}_r$ is the centroid of r and $\mathbf{c}$ is image center. We further threshold the BC for determining the background regions $\{B_{r'}\}$, $r' = 1, \ldots, K$ in I^* (where K is the total number of background regions). We can then compute the Location cue as

$$LC(r) = \exp(-\beta \cdot BC(r)). \tag{11}$$

In our experiment, we use $\beta = 8$.

4.2 Contrast Cues

Once we obtain the background regions, we apply the color-contrast based saliency detection on the non-background region. For each non-background region r and background region r' in I^*, we calculate their color difference $\delta(r, r')$ w.r.t. r' as $\delta(r, r') = \max\{|red(r) - red(r')|^2, |green(r) - green(r')|^2, |blue(r) - \ blue(r')|^2\}$. To improve robustness, we use compute the harmonic variance of all $\delta(r, r')$ for r

$$HV(r) = \left[\frac{1}{K}\sum_{r'=1}^{K}\frac{1}{\delta(r, r')}\right]^{-1}. \tag{12}$$

Combining the harmonic variance of color difference HV with location cue LC, we obtain a color contrast based saliency map as

$$S_C(r) = HV(r) \cdot LC(r). \tag{13}$$

4.3 Foreground Cues

From the detected foreground salient candidates $\{I_j^F\}$, $j = 1, \ldots, L$ via focusness analysis (where L is the total number of foreground slices), we compute the foreground cues the combining the focusness maps $\mathcal{F}_j^F(r)$ and the location cue LC

$$S_F^j(r) = \mathcal{F}_j^F(r) \cdot LC(r). \tag{14}$$

4.4 Combine

Finally, We use the objectness measure as weight for combining the contrast based salience map $S_C(r)$ and foreground maps $S_F^j(r)$ as

$$S(r) = \sum_{j=1}^{L}\omega_j \cdot S_F^j(r) + \omega_C \cdot S_C(r), \tag{15}$$

where $\omega_j = OS(S_F^j)$ and $\omega_C = OS(S_C)$ are the objectness weights calculated by Eqn. (8).

5 Experiments

Recall that most previous approaches use a single image as input whereas our approach uses the light fields. Since a light field captures much richer information of the scene than a single image, our comparisons do not intend to show that our technique outperforms the state-of-the-art as any such comparisons would be unfair. Rather, our goal is to

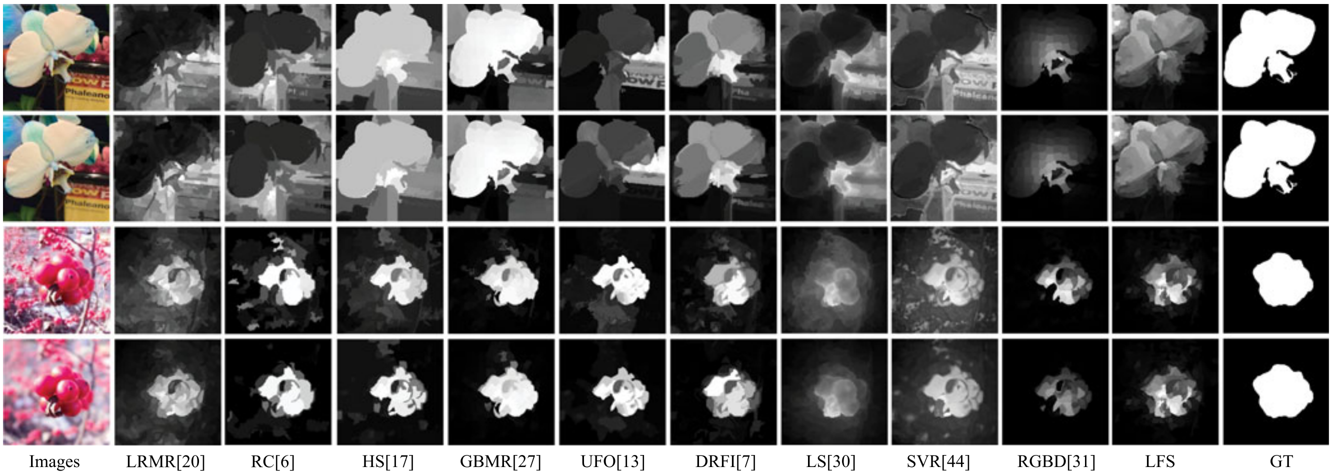


Fig. 7. Saliency results using all-focus images (the first and third rows) and partial-focus images (the second and forth rows).

show that the additional information provided by the light field can greatly improve saliency detection tasks.

5.1 Dataset

Traditional benchmark data sets [23], [39] are all single images and cannot be used to test our solution. Most online light field datasets, on the other hand, are not suitable for the purpose of saliency detection. For example, several datasets are either too simple: they only contain a single foreground object again a plain background, or too complex: foreground too cluttered. Further, most light field datasets are captured by large baseline light field cameras, to enhance the DoF effect in refocusing. Consequently, the rendered focus stacks are more likely to break salient objects into smalls pieces, which would impact the final saliency map as we discussed in Section 3.4.

We therefore first collect a dataset of 100 light fields using the Lytro light field camera. The dataset consists of 60 indoor scenes and 40 outdoor scenes.

For each data, we ask three individuals to manually segment the saliency regions from the all-focus image. The results are deemed ground truth only when all three results are consistent (i.e., they have an overlap of over 90 percent).

5.2 Evaluations on Different Superpixel Algorithms

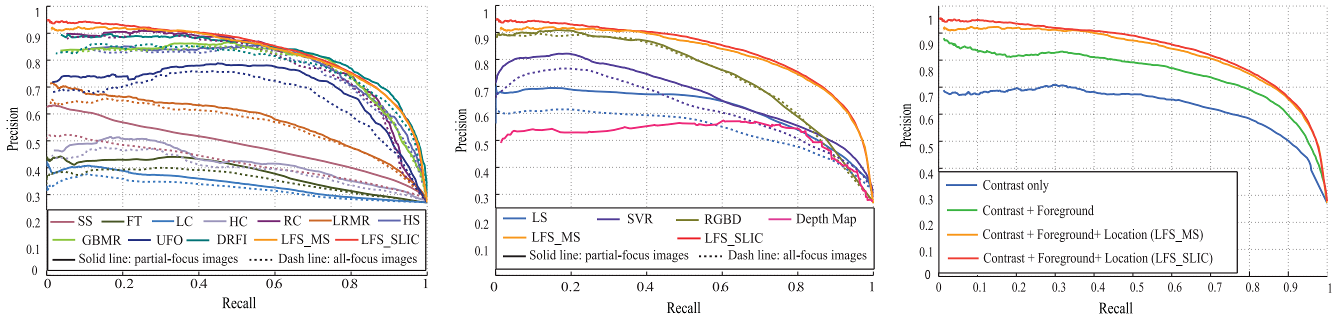
We first evaluate the impact of superpixel algorithms on our scheme. We compare the most widely used two superpixel-generating algorithms in saliency detection, i.e., Mean-Shift Clustering (MS) [36] and simple linear iterative clustering (SLIC) [40]. The rest parameters were kept the same. It is worth noting that MS would generate more regions than SLIC if they have same original superpixel number N. Consequently, we set the N of SLIC and MS to 300 and 200 respectively.

To quantitatively compare different methods, we use the canonical precision-recall curve (PRC) to evaluate the similarity between the detected saliency maps and the ground truth. Precision corresponds to the percentage of salient pixels that are correctly assigned and recall refers to the fraction of detected salient region w.r.t. the ground truth saliency. Fig. 8 shows the PRC comparison result on our light field dataset. Our experiment follows the settings in [26], i.e., we binarize the saliency map at each possible threshold within $[0, 255]$. Fig. 10 is a visual comparison between the saliency maps of different schemes. We can see that the saliency results adopting SLIC (LFS_SLIC) resemble MS (LFS_MS) whereas SLIC is about 2.5 times faster, as validated in Figs. 8 and 9.

5.3 Evaluations on Regular Images

Next, we show our light field saliency detection results and the results using a range of unsupervised schemes on regular images. These include algorithms based on spatiotemporal-cues (LC [41]), graph-based saliency (GB [42]), frequency-tuning (FT [39]), spectral residual (SS [43]), global-contrast (HC [26] and RC [6]), Low Rank Matrix Recovery (LRMR [20]), Graph-Based Manifold Ranking (GBMR [27]), focusness-based (UFO [13]), Hierarchical Saliency (HS [17]) and Discriminative Regional Feature Integration (DRFI [7]). Most these methods have open source code and we use the default parameter.

We first evaluate the performance of above methods on all-focus images. In Fig. 10, we show the saliency detection results for visual comparisons. For very challenging scenes such as the blue bird (second row), our approach produces much better results than regular image based techniques. It is important to note that all-focus image will degrade the sharpness contrast between salient object and background, which would impact the performance of algorithms based on sharpness/focusness cues. To ensure fairness, we then compare the performance on partial-focus images, i.e., the image layer focusing at a fixed depth layer. If there are several layers that focus on the same foreground object, we simply pick out the one that produces the sharpest image of the salient object. Fig. 8a provides the PRCs comparison. In Fig. 7, we show a visual comparison between the resulting saliency maps of various single-image based state-of-the-art schemes. We observe that only in cases where a partial-focus image exhibits a severely defocused background, partial-focus slice would produce better performance than an all-focus image, as shown in Fig. 7 (the blue flower scene versus the fruit scene). Notice though that the results using the complete light field still outperforms the ones using either the best partial slice or the all-focus image. This illustrates the significant advantage of using the light field as inputs for saliency detection.



(a) (b) (c)

Fig. 8. PRC comparisons on our light field dataset. (a) Results of regular image based algorithms. (b) Results of depthmap based algorithms. (c) Using different cues in our approach.

Method	FT	LC	HC	SS	RC	LRMR	HS	GBMR	DFRI	Ours_MS	Ours_SLIC
Time (ms)	8.11	3.30	231.16	102.59	918.91	12629.7	393.69	674.91	9104.18	9830.5	3755.1
Code	C++	C++	C++	Matlab	C++	Matlab	C++	Matlab&C++	Matlab	Matlab	Matlab& C++

Fig. 9. Comparison of average time taken for different saliency detection methods.

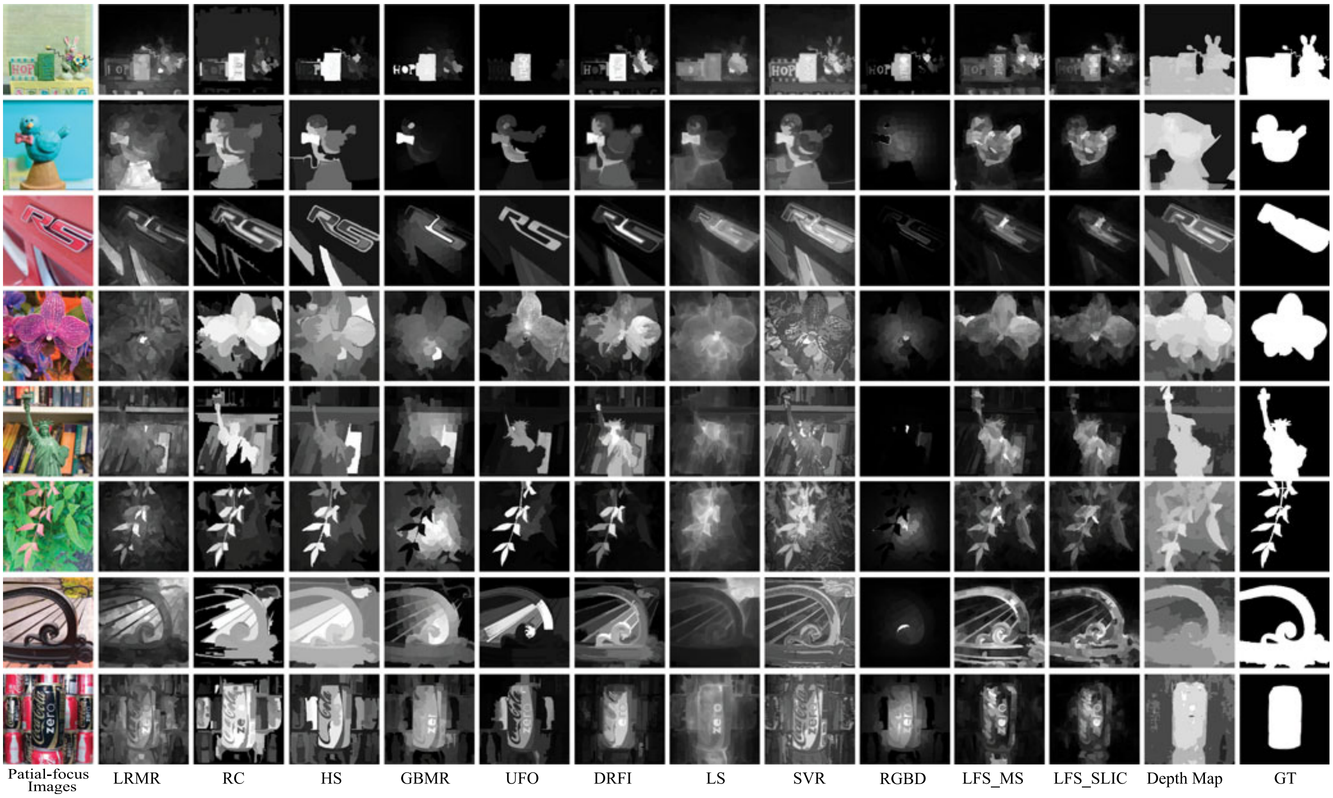


Fig. 10. Visual comparisons of different saliency detection algorithms versus ours on our light field dataset.

We would like to point it out that the PRCs are less smooth than they appear in traditional saliency works. This is due to the small amount of data in our dataset (100 light field sets versus 1,000 images in classical benchmarks), although the curves still provide useful insights on the performance. Also note that a large number of scenes in our light field dataset is highly challenging to previous techniques, i.e., many have complex background or similar foreground and background. Fig. 10 shows sample in-focus images of these difficult scenes. We observe that recently proposed RC [6], HS [17] and DRFI [7] can still achieve reasonable performance. This is partially due to the background prior refinement and color space smoothing methods used in RC, the multi-scale features used in HS and the supervised feature vector mapping approach used in DRFI. Results using our technique produces the highest precision in the entire recall range. This illustrates the importance of focusness and objectness prior provided by the light field.

Fig. 9 evaluates the running time of each methods. We implemented all methods with open source code and list their average running time for one scene. Notice that even though our algorithm needs processing much larger data (about 10 times) than others, the average processing time is

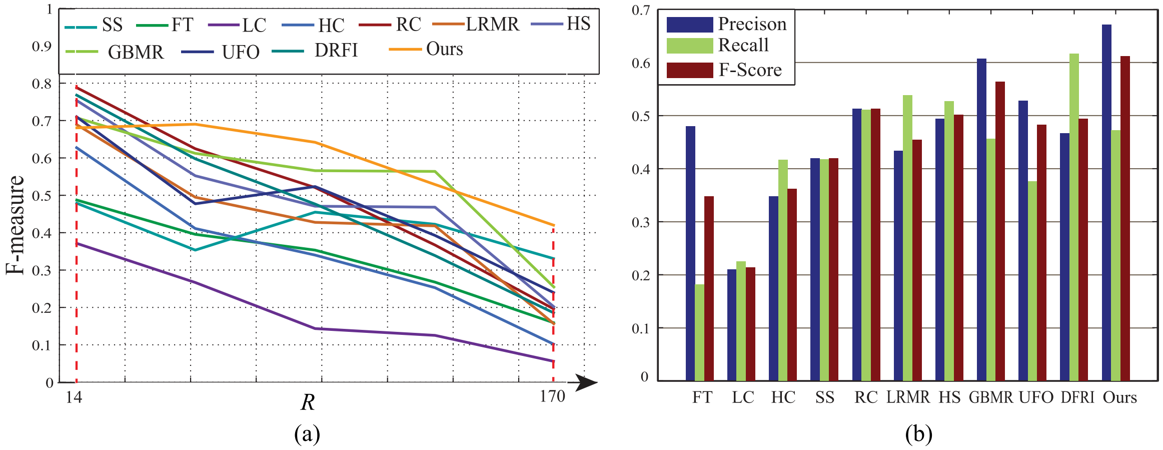


Fig. 11. (a) Performance comparisons of F-score regarding $\mathcal{R}$. (b) Average precision, recall and F-score on 50 testing light fields.

still comparable to those regular-image-based techniques using the same programming platform.

5.4 Evaluations on Images with Depth Information

We further choose three recent proposed depthmap-based methods, i.e. SVR [44], LS [30] and RGBD [31], to compare their performances with our model. The depth maps of LFSD are generated directly from Lytro desktop. Fig. 8b shows the PRC comparisons among above mentioned algorithms, which illustrates that the focusness cues utilized in our technique are equally or more useful than depth cue. Fig. 10 shows the visual comparisons on several LFSD images. We observe that LS and SVR may produce low precision results, since they treat depth cues independently for saliency detection, while ignoring the strong complementarity between appearance and depth cues and utilize depth cues as an independent image channel for saliency detection. It is important to note that directly using depth maps as saliency cues is not reliable. For example, simple thresholding on the depth maps would produce large errors on images in row 3, 4, and 6 of Fig. 10 where both salient and non-salient objects lie at the similar depth. In fact, in Fig. 8b, we have plotted the PRC performance by using merely depth maps as saliency cues and the results show that it is inferior to depth-based approaches.

We also evaluate their performance on both all-focus and partial-focus images. It is noteworthy that all-focus images also degrade the performance of those depthmap-based techniques. This is because that all these three methods incorporate depth saliency with regular saliency models to obtain the final saliency maps. Moreover, the larger the regular saliency features weigh, the more evident improvement will show.

5.5 The Effect of Camera-to-Object Distance

Recall that a Lytro camera has small baseline. In order to enlarge the infocusing and defocusing contrast between foreground and background, most of the salient objects in our dataset are placed near the Lytro camera. When the foreground object is faraway from the camera, the change of depth-of-field when switching the focus from the foreground to the background would be less significant. To test whether our algorithm is robust to the camera-to-object distance d, we capture 50 more light fields where salient objects are located at diverse d. Notice that the maximum d making the object notable is proportional to the object size. In our experiment, instead of exploring the connection of performance versus d, we analyze the relevance of performance versus objects' depth-to-size ratio $\mathcal{R}$

$$\mathcal{R} = \frac{d \cdot max(depth\ range)}{Height(Object) \cdot Width(Object)}. \tag{16}$$

Typically, the range of $\mathcal{R}$ in our testing set is between $[14, 170]$. To plot the performance versus $\mathcal{R}$ curve, we divide the 50 light fields into 5 subsets according to their $\mathcal{R}$ value. Each set contains about 10 light fields. Here, we adopt the F-score methodology

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}. \tag{17}$$

It is concluded in [23] that precision rate is more essential than recall in attention detection. Accordingly, we choose $\beta^2 = 0.3$ to weigh precision more than recall. For each light field set, we calculate its average precision and recall rate. The average F_β is derived by Eqn. (17). Fig. 11a shows the $F_\beta - \mathcal{R}$ curves of different methods and Fig. 11b presents the comparison of average precision, recall and F-score on this 50 light fields set. Fig. 12 provides the visual comparison of different methods when changing d. We can tell that as $\mathcal{R}$ goes larger, the performance of most algorithms decrease. This is because that when the object lies far away from the camera, in the slice where the object is in-focus, the background appears nearly focused as well. Such a scenario resembles the classical all-focus saliency detection case where the usefulness of most focusness cue is reduced. Due to the effectiveness of our focusness detection algorithm, our method shows the best robustness in $F_\beta - \mathcal{R}$ curves and also achieves the highest average F-score.

5.6 The Effect of Parameters

For all the experiments described above, the parameters were kept fixed, i.e., no user fine-tuning was done. To test the robustness of our algorithm to the parameters and to analyze their effect, we repeated the experiments, while varying η from Eqn. (5), σ from Eqn. (7), N (the number of

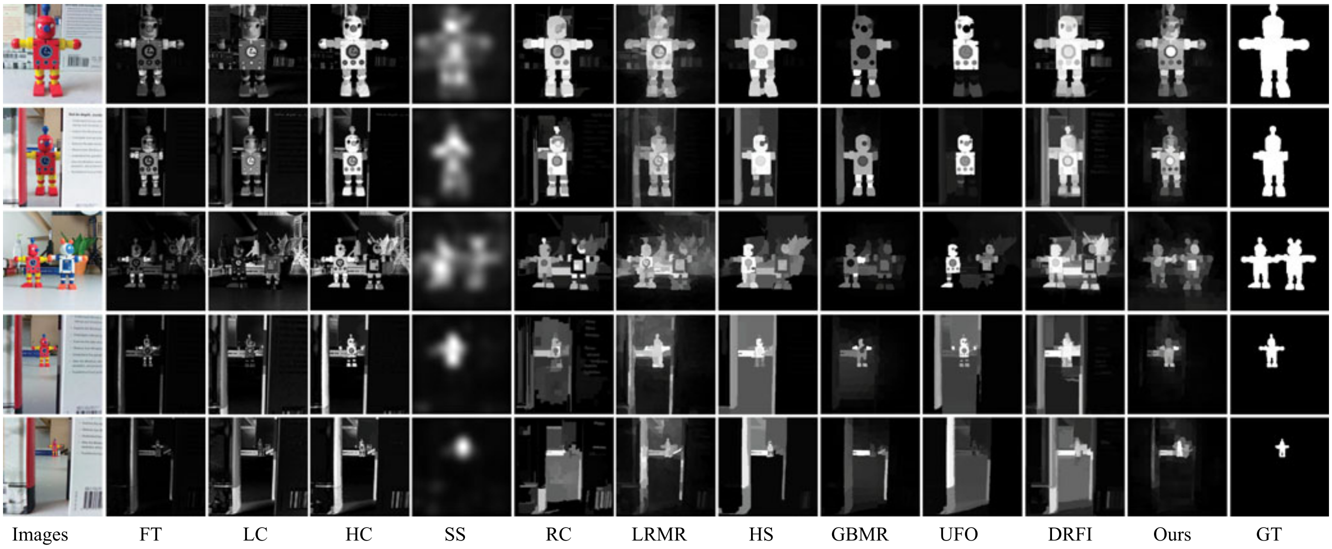


Fig. 12. Saliency maps of red robot at different $\mathcal{R}$. From top to down, $\mathcal{R} = 14, 53, 92, 131, 170$.

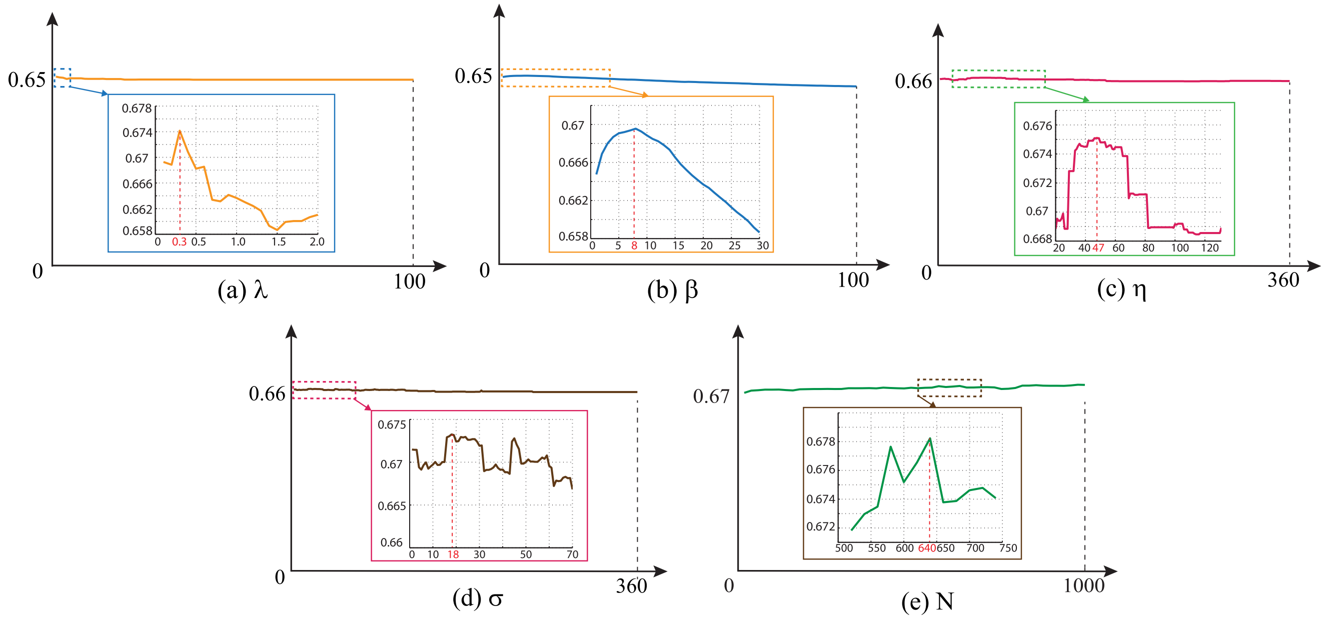


Fig. 13. F-score curvers when varying λ, β, η, σ, N.

superpixels), β from Eqn. (11), and λ from Eqn. (6). To quantitatively show these impact, we follow the F-score methodologies described in Section 5.5 to evaluate the accuracy of the detected saliency when varying η, σ, N, β and λ, as shown in Fig. 13.

Parameter η controls U-shape filter while parameter σ controls the shape of Gaussian. Since we have normalized both U-shape and Gaussian filter to a very small scale, those two parameters causes barely modifications when changing values. N denotes the number of superpxiels. It can be observed that our algorithm is very robust as well, due to the performance is dominantly affected by the objectness of superpixels and less by the number of superpixels. Certainly when the number of superpixels is too small, the salient and non-salient regions will be merged and the performance of our approach will be inferior. It is noticeable that the above three parameters are varying among different ranges. η should vary between 0 and $min(w, h)$, where w and h are the width and height of the image. Recall that the Gaussian filter has relative large response between $[\mu - \sigma, \mu + \sigma]$, during which the salient objects should be located. Therefore, we normally keep σ between 0 and $min(w, h)$. As for N, we range it from 20 to 1,000.

Parameter β effects the highlight extend of regions located at the center of images and parameters λ controls the probability of picking the back layer as background slice. β and λ of exponential functions, on the other hand, have much smaller ranges, i.e., [0, 100], to prevent from out of memory issues. Unlike η, σ and N, changing the values of β and λ has specific impact on our final results, even though slightly.

Fig. 13b reveals that when β goes large, the performance of our approach will degrade. Notice that low recall occurs when highlighted regions in saliency maps are all of high value. In our case, the larger β is, the higher values will be assigned to the central regions by the location cue, i.e., LC.

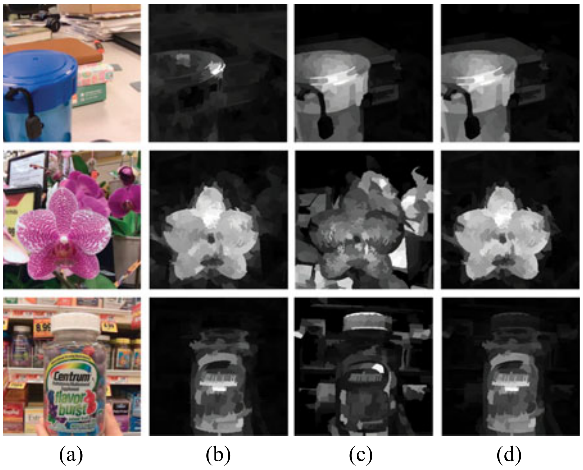


Fig. 14. Saliency detection using different cues. (a) All-focus images; (b) Detected saliency using focusness cues; (c) Detected saliency using color contrast. (d) Saliency results by combining (b) and (c).

F-score, therefore, will be decreased consequently. When λ is large (> 2), the performance will degrade slightly (about 0.03). This is because that large λ value will enforce picking the furthest layer as background slice, which may fail at cases we discussed in Section 3.3.

We also compare the saliency components obtained using different cues, i.e., color contrast, location and focusness cues. Fig. 8c shows the PRC comparisons using individual versus combined cues. The plot illustrates that each cue has its unique contribution to saliency detection, although in some cases, an image can be dominated by a specific cue as shown in Fig. 14. In the first row, color contrast provides most valuable cues and the estimated saliency from it resembles the final one. This is mainly because the blue mug lacks texture and hence is not robustly detected as the foreground object to provide focusness cues. In contrast, in the flower scene in the second row, the color contrast result treats both the foreground flower and the background clutter as saliency. The focusness cue, however, manages to correct the errors by removing the background. In the last example, the color contrast result misses the foreground bottle and the focusness cue manages to add it back.

5.7 Limitations

The performance of our algorithm is largely dependent on the quality of the acquired light field. Lytro, however, has a much narrow Field-of-View than regular cameras. Therefore, objects in our light fields generally appear "bigger" than in other benchmarks. With emerging interest on light field camera designs, we expect next-generation models to overcome this limitation. There are also alternative approaches to use the light field for saliency detection. For example, one can potentially first construct a depth map using stereo matching. However, the quality of stereo matching depends largely on scene composition. Nevertheless, even a low quality depth map may provide useful cues comparable to the focusness cue. Furthermore, it is also possible to first conduct saliency detection on the all-focus image and then use the results to improve the quality and speed of light field stereo matching.

6 Conclusion

We have presented a saliency detection algorithm tailored for light fields. We believe this is the first light field saliency detection scheme. The key advantage of using a light field instead of a single image is that it provides both focusness and depth cues. In recent works [13], [29], these new cues have shown great success in improving accuracy and robustness in saliency detection. Our solution echoes these observations and also provides an alternative and more robust method to extract these cues through the analysis of light fields. Experiments show that our technique can handle many challenging scenarios that cast problems on traditional single-image-based algorithms. Another contribution of our work is the construction of the light field saliency dataset which consists of the raw light field data, the synthesized focal stacks and all-focus images, and the ground truth saliency maps. Our immediate future work is to build a much larger and comprehensive database and share it with the community.

Acknowledgments

This research is supported by National Science Foundation Grant IIS-1218156. Jingyi Yu is the corresponding author.

References

[1] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": Interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, 2004, pp. 309–314.

[2] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 989–1005, Jun. 2009.

[3] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, "Automatic thumbnail cropping and its effectiveness," in *Proc. 16th Annu. ACM Symp. User Interface Softw. Technol.*, 2003, pp. 95–104.

[4] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, 2008.

[5] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5706–5722, Dec. 2015.

[6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 569–582, Mar. 2015.

[7] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 2083–2090.

[8] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. 28th Annu. Conf. Comput. Graph. Interactive Techn.*, 2001, pp. 425–432.

[9] C. Kim, A. Hornung, S. Heinzle, W. Matusik, and M. Gross, "Multi-perspective stereoscopy from light fields," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 190:1–190:10, Dec. 2011.

[10] S. Wanner, C. Straehle, and B. Goldluecke, "Globally consistent multi-label assignment on the ray space of 4D light fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1011–1018.

[11] L. Itti, "Quantitative modelling of perceptual salience at human eye position," *Visual Cognition*, vol. 14, no. 4-8, pp. 959–984, 2006.

[12] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Apr. 2005. [Online]. Available: http://graphics.stanford.edu/papers/lfcamera/

[13] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1976–1983.

[14] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 29–42.

[15] N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu, "Saliency detection on light field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2806–2813.

[16] L. Mai, Y. Niu, and F. Liu, "Saliency aggregation: A data-driven approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1131–1138.

[17] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1155–1162.

[18] A. Maki, P. Nordlund, and J.-O. Eklundh, "Attentional scene segmentation: Integrating depth and motion," *Comput. Vis. Image Understanding*, vol. 78, no. 3, pp. 351–373, 2000.

[19] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[20] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 853–860.

[21] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, "Visual saliency detection by spatially weighted dissimilarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 473–480.

[22] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Proc. Advances Neural Inf. Process. Syst.*, 2006, pp. 155–162.

[23] T. Liu, et al., "Learning to detect a salient object," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 353–367, Feb. 2011.

[24] R. Valenti, N. Sebe, and T. Gevers, "Image saliency by isocentric curvedness and color," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 2185–2192.

[25] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[26] M. Cheng, G. Zhang, N. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 409–416.

[27] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3166–3173.

[28] C. Lang, T. V. Nguyen, H. Katti, K. Yadati, M. Kankanhalli, and S. Yan, "Depth matters: Influence of depth cues on visual saliency," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 101–115.

[29] Y. Niu, Y. Geng, X. Li, and F. Liu, "Leveraging stereopsis for saliency analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 454–461.

[30] A. Ciptadi, T. Hermans, and J. M. Rehg, "An in depth view of saliency," in *Proc. British Mach. Vis. Conf.*, 2013, pp. 112.1–112.11.

[31] H. Peng, B. Li, W. Xiong, W. Hu, and R. Ji, "RGBD salient object detection: A benchmark and algorithms," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 92–109.

[32] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Techn.*, 2000, pp. 297–306.

[33] A. Agarwala, et al., "Interactive digital photomontage," in *Proc. ACM SIGGRAPH*, 2004, pp. 294–302.

[34] M. Levoy, "Light fields and computational imaging," *IEEE Comput.*, vol. 39, no. 8, pp. 46–55, Aug. 2006.

[35] J. Ye and J. Yu, "Ray geometry in non-pinhole cameras: A survey," *Visual Comput.*, vol. 30, pp. 93–112, 2013.

[36] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[37] F. Li and F. Porikli, "Harmonic variance: A novel measure for in-focus segmentation," in *Proc. British Mach. Vis. Conf.*, 2013, pp. 33.1–33.11.

[38] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[39] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1597–1604.

[40] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[41] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," *Proc. 14th ACM Int. Conf. Multimedia*, 2006.

[42] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. Advances Neural Inf. Process. Syst.*, 2006, pp. 545–552.

[43] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 194–201, Jan. 2012.

[44] K. Desingh, K. M. Krishna, D. Rajan, and C. Jawahar, "Depth really matters: Improving visual salient region detection with depth," in *Proc. British Mach. Vis. Conf.*, 2013, pp. 98.1–98.11.

Nianyi Li received the BE degree from the Department of Electrical Engineering, Huazhong University of Science and Technology, in 2013. She is now working toward the PhD degree in the Department of Computer and Information Sciences, University of Delaware. Her research interests include saliency detection, object recognition, super resolution, scene reconstruction, light field rendering, and non-pinhole camera models. She is a student member of the IEEE.

Jinwei Ye received the PhD degree from the University of Delaware, in 2013. After that she spent one year working as a postdoctoral researcher in the US Army Research Laboratory. She is currently a senior scientist in the Innovation Center of Canon U.S.A., Inc., San Jose, California. Her research interests include invisible object acquisition and reconstruction, image processing, ray geometry, and non-pinhole camera models. She is a member of the IEEE.

Yu Ji received the bachelor's degree from Huazhong University of Science and Technology, in 2009, and the MSc degree from Nanyang Technological University, in 2011. He joined the Department of Computer and Information Science, University of Delaware, in 2011, where he is currently working toward the PhD degree under the supervision of Prof. Jingyi Yu. His research interests include computational photography, computer vision, and computer graphics. He is a student member of the IEEE.

Haibin Ling received the PhD degree from the University of Maryland, College Park, in 2006. From 2006 to 2007, he worked as a postdoctoral scientist with the University of California, Los Angeles. After that, he joined Siemens Corporate Research as a research scientist. He joined Temple University, in 2008, where he is now an associate professor. His research interests include computer vision, augmented reality, medical image analysis, and human computer interaction. He is a member of the IEEE.

Jingyi Yu received the BS degree from Caltech, in 2000, and the MS and PhD degrees in EECS from MIT, in 2005. He is a professor in the Computer and Information Science Department, University of Delaware. His research interests span a range of topics in computer graphics, computer vision, and image processing, including computational photography, medical imaging, nonconventional optics and camera design, tracking and surveillance, and graphics hardware. He is a member of the IEEE.