

MCCH: A Novel Convex Hull Prior Based Solution for Saliency Detection

Xiao Lin^{a,b}, Zhi-Jie Wang^{c,d,e}, Xin Tan^a, Mei-E Fang^f, Neal N. Xiong^h, Lizhuang Ma^{a,g}

^a*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*

^b*Department of Computer Science and Engineering, Shanghai Normal University, Shanghai, China*

^c*School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China*

^d*Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China*

^e*National Engineering Laboratory for Big Data Analysis and Applications, Beijing, China*

^f*Department of Computer Science, Guangzhou University, Guangzhou, China*

^g*School of Information Science Technology, East China Normal University, Shanghai, China*

^h*Department of Mathematics and Computer Science, Northeastern State University, Oklahoma, United States*

Abstract

In the past several decades numerous papers studied the problem of salient object detection. Among them, one of the representative approaches is to use the convex hull prior to find the salient object in the image; and there are many variants, which are based on the convex hull prior. Most of these works used a *single* center to construct the convex hull center prior map. Yet, few attention has been made on the use of multiple centers. In this paper, we suggest a multi-center convex hull prior based solution for salient object detection. Our method is simple enough to be practical value. Particularly, our solution also integrates two non-trivial optimizations: one is for obtaining an enhanced global color distinction prior map, and another is for refining the preliminary saliency map. Extensive experimental results demonstrate that our solution is effective and also competitive, compared against the classic and also state-of-the-art saliency detection models.

Keywords: Salient object detection, convex hull prior, multi-center prior map, enhanced global color distinction prior map, improved Bayesian optimization framework

1. Introduction

Visual saliency [19] is one of the classical ways to find regions of interest in the image. In the past many years, abundant efforts have been made on salient object detection [20, 37, 33, 4, 30, 53, 56, 11, 60, 74, 57, 40, 46, 72]. The essence of the salient object detection is to highlight object(s) of interest in the image [33, 53]. Salient object detection is convenient for subsequent operations in image processing and other domains [56, 18, 25, 31, 61, 62, 66, 63, 64], and it has been applied to many modern computer vision tasks (e.g., *image classification* [43], *image compression* [18], *image retrieval* [67], *object location* [12, 28], *image segmentation* [41], *artificial retinal prostheses* [24], *visual attention analysis* [36], *target tracking* [3, 49]). In the literature, existing saliency detection algorithms can be generally classified into two categories: (i) the top-down approaches [33, 4, 37, 53, 5, 65, 17, 68], and (ii) the bottom-up approaches [56, 74, 57]. As pointed out in [69], the top-down approaches, including CNN-based deep learning ones, capture representative high-level features, thereby detecting salient objects of certain sizes and categories; in contrast, the bottom-up approaches have a wide range of applications because they mainly depend on some low-level visual features (e.g., color, intensity or orientation) and some prior knowledge (e.g., contrast, compactness, uniqueness or boundary). In this paper, we focus our attention on the bottom-up models.

The bottom-up approaches have many branches [57, 42, 26, 25, 56], and one of the representative approaches is the *convex hull* based approach [32, 14]. A major feature of this approach is to approximately locate the foreground seeds via points of interest; see e.g., [56, 52, 47, 32, 14, 58, 29, 73]. In these literature, most of works utilized a

Email addresses: Lin6008@126.com (Xiao Lin), wangzhij5@mail.sysu.edu.cn (Zhi-Jie Wang), fishtanx2015@gmail.com (Xin Tan), fme@gzhu.edu.cn (Mei-E Fang), xiong31@nsuok.edu (Neal N. Xiong), ma-lz@cs.sjtu.edu.cn (Lizhuang Ma)

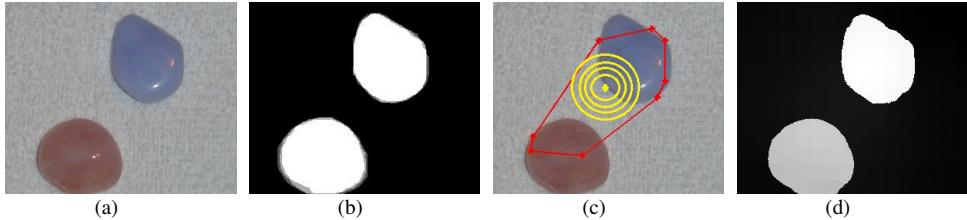


Figure 1: Example of two objects/targets in the image. From left to right: (a) input image; (b) ground-truth; (c) the convex hull and the single center; (d) the saliency map via the single center.

single center to construct the convex hull center prior map, while few efforts have been taken to use multi-centers. We observe that, in many images there are multiple targets/objects, the convex hull generated based on these targets could be the polygon with a large *span* in some direction, or the similar cases; see Fig. 1, for example. Essentially, for images in which only a single target/object exists, the above phenomenon could be also appeared, as shown in Fig. 2. The single-center convex hull based algorithms in these cases could fail to highlight the salient object effectively (see Figs. 1(d) and 2(d)), since it always assigns the high scores to the regions close to the center of the convex hull.

Inspired by the reasons above, in this paper we suggest a *Multi-Center Convex Hull prior based solution*, dubbed as MCCH, for short. The central contribution of our solution is the construction of the multi-center prior (MCP) map, which serves as the foreground prior map and is to be integrated with the background-based map in the subsequent steps. The difficulty to construct the MCP map is on how to find appropriate centers in the convex hull. To attack this challenge, we propose a *dynamic k-center algorithm* that can assist us to find appropriate centers, in terms of the number of and the locations of centers. Besides, we also present two non-trivial optimizations below.

The first optimization is used for constructing a more robust *global color distinction prior* (GCDP) map¹, which serves as the background prior map. This optimization is motivated by an insight into the existing methods for constructing the GCDP map [40]. In brief, existing methods construct the GCDP map by grouping superpixels in boundaries into three clusters and then construct three GCDP maps. Instead, we construct four different GCDP maps based on four boundaries, respectively. The intuition behind our method is that, the background often presents the local or global consistence with one or some of four boundaries. Particularly, we further improve the four GCDP maps by fully utilizing the convex hull obtained in previous steps. Later, these four (optimized) GCDP maps are to be merged into a robust GCDP map in a natural manner.

The second major optimization is used for refining the saliency result obtained by integrating the MCP and GCDP maps (a.k.a., foreground and background prior maps). Our optimization method is tailored for addressing the limitation of existing Bayesian optimization framework, improving the quality of the saliency result directly. The rationale behind our method is to assign larger weights for superpixels in the convex hull region, even if the colors in these superpixel regions are similar to that in the background, and then employ the Bayesian formula to calculate the final saliency map.

The novelty and the contributions of this paper are as follows:

- We suggest a multi-center convex hull prior based solution framework for saliency detection (Section 4).
- We present a dynamic k-center algorithm that serves as the core of the MCP map construction (Section 5). To our knowledge, this is the first work suggesting the use of MCP map based solution for saliency detection.
- We refine existing methods for constructing the GCDP map (Section 6), and present an improved Bayesian optimization framework to refine the initial saliency map (Section 7).
- We conduct extensive experiments to demonstrate the effectiveness and superiorities of our proposed model (Section 8).

In the next section, we review previous works most related to ours.

¹Sometimes, one may also call it the *global color distinction* (GCD) map. In the rest of the paper, these two terminologies are used alternatively, unless stated otherwise.

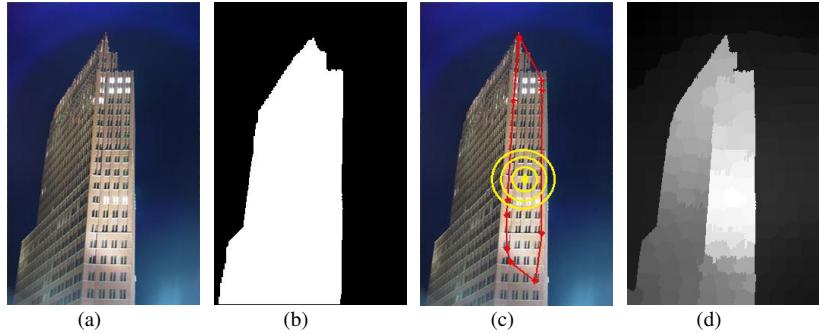


Figure 2: Example of one object/target in the image. From left to right: (a) input image; (b) ground-truth; (c) the convex hull and the single center; (d) the saliency map via the single center.

2. Related work

As mentioned before, in the literature existing saliency detection algorithms can be generally classified into two categories: (i) the top-down approaches [33, 4, 53]; and (ii) the bottom-up approaches [56, 74, 57, 40, 46]. Our work generally belongs to the second branch. In what follows, we mainly review prior works in this branch.

Regarding the bottom-up approaches, some of papers mainly consider the foreground prior, and then generate the final saliency maps based on the chosen foreground seeds (see e.g., [57, 42, 26, 25, 56]). This line of methods can be further classified into several categories below: (a) the center prior based methods, which are more likely suitable for detecting the salient object that appears at the center of the image, see e.g., [57, 46, 27]; (b) the low-ranking matrix based methods, which usually represent the input image as a low-ranking matrix, and define the salient objects as the sparse noises in a certain feature space, see e.g., [42, 38]; (c) the focusness prior based methods, which leverage the fact that a salient object is often photographed in focus to attract more attention, see e.g., [26, 21]; (d) the objectness likelihood map based methods, which utilize the probability map to locate the foreground seeds, see e.g., [25]; and (e) the convex hull based methods, which approximately locate the foreground seeds via points of interest, see e.g., [56, 52, 47]. Compared with these works (mentioned just now), our work considers both the foreground and background priors, and thus is different from theirs. Among these works, the ones most similar to ours are [56, 52, 47], since both our paper and theirs utilized the convex hull to find the foreground seeds. Note that, although these works utilized the convex hull to find the foreground seeds, (i) none of these works suggested the use of multi-center model, which is one of our major contributions; and (ii) again, these works did not utilize the background priors, which are used in our work.

It is worth noting that there are still some works (see e.g., [32, 14]) in which they consider both the foreground and background priors, and also utilize the convex hull based technique for finding the foreground seeds. Note that, they distinguish the foreground and background regions by the learning-based algorithms. Yet, our paper utilizes the color contrast to distinguish them, and thus different from theirs. One could argue that, existing works (e.g., [58, 29, 73]) used also the color contrast to differentiate the foreground and background regions. Yet, these works did not well consider the fact — the color contrast based method could restrain the foreground regions (whose colors are similar or same to some background regions). In our paper, we alleviate this limitation by suggesting targeted optimization strategies. Also, all these works mentioned before did not cover the multi-center convex hull prior. In this regard, they are also different from ours. This article is an extension of the conference version [54]. In the article, we present a comprehensive review of prior works, a more complete description on the proposed method, and a more comprehensive experimental comparison. We believe that these new added materials/elements are useful to the potential readership.

3. Preliminaries

For ease of understanding the rest of the paper, this section *reviews* some terminologies and concepts. The frequently used notations are summarized in Table 1.

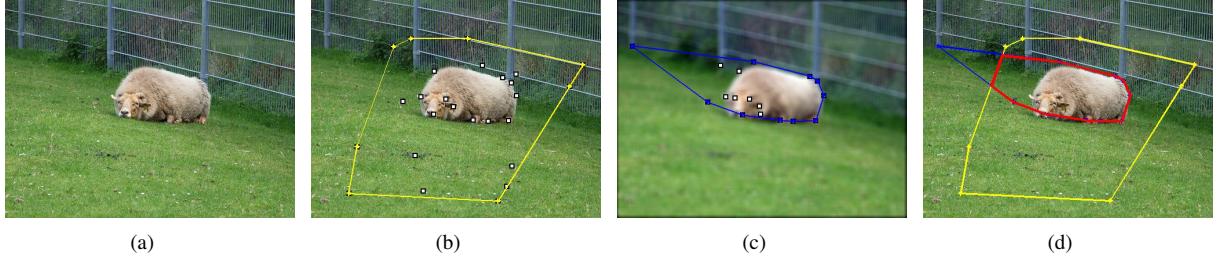


Figure 3: Example of constructing the convex hull. From left to right: (a) input image; (b) R_{prev} ; (c) R_{impr} ; (d) $R_{prev} \cap R_{impr}$.

▷ *Constructing the convex hull.* Many previous works (e.g., [56, 73]) developed the algorithms for constructing the convex hull, based on points of interest in the image. The latest method [73] is a variant of some early methods. Since our paper shall use this algorithm to construct such a convex hull, we next briefly review this algorithm. Denote by C the convex hull to be constructed. It is computed as

$$C = R_{prev} \cap R_{impr} \quad (1)$$

where \cap denotes the intersection set of two closed regions, as shown in Fig. 3(d); R_{prev} is the convex hull region constructed by early methods, whereas R_{impr} is another convex hull region that is different from R_{prev} usually. The general steps of constructing R_{prev} can be described as follows. It first uses the *color boosted Harris point operator* [48] (a.k.a., an improved version of the *Harris detector* [16]) to find points of interest in an input image. Then, it eliminates some points near to the image boundaries, and computes a region (i.e., R_{prev}) that encloses all the remaining points of interest, as shown in Fig. 3(b). Compared to R_{prev} , R_{impr} is obtained by a revision to the above steps. Specifically, it first utilizes the low-pass filter to smooth the input image, and then uses the Harris detector to find points of interest and gets the corresponding convex hull, as shown in Fig. 3(c).

▷ *Global color distinction prior map.* In existing literature, many bottom-up methods utilize the image boundaries as the background seeds. Recently, [40] proposed the *global color distinction prior* (GCDP) map to achieve more optimal background prior map. The general steps for constructing the GCDP map are as follows. It first segments the input image into a number of (e.g., N) superpixels by the *simple linear iterative clustering* (SLIC) algorithm [2], and then groups the superpixels (in the image boundaries) into $k (= 3)$ clusters. Based on these clusters, it constructs three different GCDP maps. Finally, these different GCDP maps are integrated into a single GCDP map, by summarizing them with corresponding weights.

▷ *Bayesian optimization framework.* Recently, the Bayesian formula has been used to obtain the saliency map with a high quality [56]. This optimization framework can be generally described as follows. It first estimates the

Table 1: Notations and their meanings

Notation	Description	Notation	Description
C	the convex hull	n_m	the number of superpixels in the m boundary
c	the centroid of C	\mathbb{G}_m	the m th GCDP map
r	the convex hull centroid radius	$S_{m,i}$	sp_i 's saliency value in \mathbb{G}_m
N_p	the number of points of interest in C	$w(i, j)$	the color similarity between sp_i and sp_j
η	the coverage ratio	\mathbb{G}	the final GCDP map
s	the overlap size	S_i^{bg}	sp_i 's saliency value in the final GCDP map
ι	the overlap degree	w_i	the superpixel region weight
τ_η (τ_ι)	the threshold for η (ι)	c_i	the feature vector of sp_i in CIELab color space
sp_i	the i th superpixel	$p_w(\cdot \cdot)$	the weighted observation likelihood
x_i (y_i)	the mean horizontal (vertical) coordinates of sp_i	$p(\cdot \cdot)$	the observation likelihood
S_i^{fg}	sp_i 's saliency value in the MCP map	v	the pixel in an image

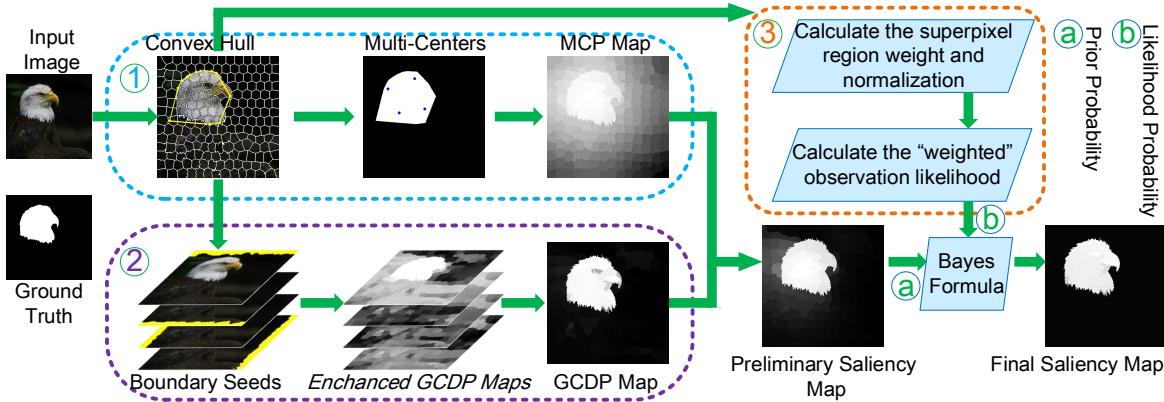


Figure 4: Architecture of our solution. It consists of three major parts: the MCP map construction (see ①), the enhanced GCDP map generation (see ②); and the improved Bayesian optimization framework (see ③).

approximate saliency region by constructing the convex hull [56]. Then, based on the estimated region, it formulates the saliency detection as a Bayesian inference problem for estimating the posterior probability at each pixel v of the image. Let $p(sal|v)$ denote the probability of predicting a pixel v be salient. It is computed as

$$p(sal|v) = \frac{p(sal)p(v|sal)}{p(sal)p(v|sal) + p(bg)p(v|bg)} \quad (2)$$

where $p(\cdot)$ denotes the probability, $p(sal)$ is the prior saliency distribution, $p(v|sal)$ is the likelihood of observations; other symbols have similar meanings.

4. Solution overview

An overall architecture of our solution is shown in Fig. 4. Generally speaking, our solution contains three major phases as follows.

▷ *Phase ①*. The main goal in this phase is to obtain the MCP map (which serves as the foreground prior map, and shall be integrated with the background prior map, in order to generate the *preliminary* saliency map). The difficulty in this phase is on how to choose k centers appropriately; we develop a *dynamic k-center algorithm* that contributes to alleviating this challenge. At a high level, the work-flow in this phase can be stated as follows. We first construct a convex hull that approximately estimates the saliency regions, and then use *Harris algorithm* [48] to find some points of interest in the convex hull region. Based on these points of interest, the clustering algorithm and some targeted strategies, we develop a *dynamic k-center algorithm* to choose a set of k centers (which will be used to construct the MCP map). We observe that, these k -centers could be not uniformly distributed in the convex hull region; to alleviate this issue, we suggest using different weighting coefficients for different centers in the final step of the MCP map construction. Notice that, regardless of multiple objects or single object in the image, our solution can work correctly and effectively.

▷ *Phase ②*. This phase is mainly to construct a more robust GCDP map (which serves as the background prior map). Essentially, this phase is an optimization on existing methods for generating the GCDP map. Our optimization can be viewed as a fusion of two ideas: (i) using separately each side of image boundaries to construct enhanced GCDP maps; and (ii) utilizing fully the convex hull (obtained before) to further optimize the enhanced GCDP maps. The main observation motivating the first idea is that, the background often presents the local or global consistence with one or some of four boundaries; while the intuition behind the second idea is that, the convex hull obtained before can enclose most or even all the foreground regions, and so assigning higher scores to the convex hull regions can weaken some noises in the foreground.

▷ *Phase ③*. The MCP and GCDP maps obtained before can be immediately integrated into the saliency map, and the existing Bayesian optimization framework can improve the quality of the generated saliency map. Yet, we realize that some superpixel regions in the foreground, whose colors are similar to that in the background, could

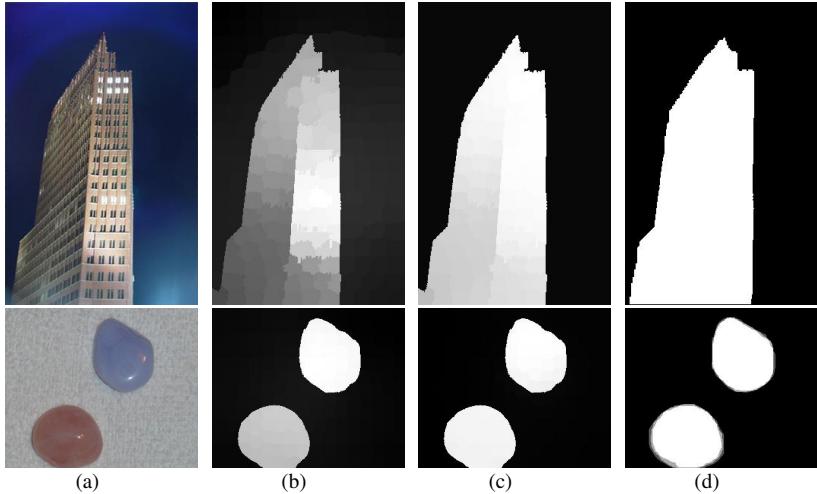


Figure 5: Our result compared with the original result. From left to right: (a) input image; (b) saliency result via single-center [58]; (c) our result; (d) ground-truth.

be restrained when one uses the existing Bayesian optimization framework. In this phase we suggest an improved Bayesian optimization framework to refine the generated saliency map (a.k.a., the *preliminary saliency map*). Overall, this framework can be viewed as a refinement to the existing framework. The rationale behind our optimization is to assign larger weights for superpixels in the convex hull region *even if* these superpixel regions' colors are similar to that in the background. Note that, although some ideas (e.g., utilizing the convex hull obtained previously) in the last two phases seem to be similar, they work at different levels: one is for the background prior map, while another works on the preliminary saliency map; and thus they are different in essence.

Our solution framework is easy-to-understand and implement, while the MCP map together with two optimizations collaboratively contributes to substantial improvements, as demonstrated in Section 8. In the subsequent sections (Sections 5 ~ 7), we discuss the main steps of our model in detail, respectively.

5. Constructing MCP map

As we mentioned in Section 1, the single-center convex hull based methods could assign the high scores to the background regions near to the center, and could restrain the saliency regions which are near to the boundary of the convex hull. This nature may lead poor results, recall Figs 1(d) and 2(d); or, see directly Fig. 5(b). To address this issue, we propose to construct the prior map using multiple centers (say k), where k is an integer. For ease of presentation, we call it *multi-center prior* (MCP) map. This technique can significantly improve the quality of saliency results, as shown in Fig. 5(c). In what follows, we present the details on how to construct the MCP map.

We first use an existing algorithm [73] to construct a convex hull, say C . This convex hull shall approximately determine the location of and also the contour of the salient object. We then use the classic and widely used *Harris algorithm* [48] to find some points of interest in the convex hull region. Assume, without loss of generality, that there are N_p points of interest (found by Harris algorithm). Next, we choose a set of k centers *dynamically*, based on these points of interest and some targeted strategies. For clarity, we call this algorithm *dynamic k-center algorithm*.

For ease of understanding the dynamic k-center algorithm, we first clarify several important concepts. Let n_l be the number of edges of C , and let c denote the centroid of C . In addition, for the i th edge l_i , we denote by $dist(l_i, c)$ the distance between the edge l_i and the centroid c . The convex hull centroid radius, denoted by r , is defined as

$$r = \operatorname{argmin}_{i \in [1, n_l]} (dist(l_i, c)) \quad (3)$$

In addition, given a set S_n of n points in the convex hull C , the coverage ratio of S_n in C , denoted by η , is defined as

Algorithm 1 *k*-center

Input: N_p points (i.e., the initial centers), the convex hull centroid radius r , the convex hull C

Output: updated N_p points // where the value N_p could be also updated

- 1: Initialize d, ϵ and τ_ι ;
 - 2: compute η and ι ; // Eqs. 4 and 7
 - 3: Let $\tau_\eta = d * \eta$;
 - 4: **while** $\iota \geq \tau_\iota$ **do**
 - 5: set $N' = \lfloor N_p / 2 \rfloor$;
 - 6: group N_p points into N' clusters;
 - 7: compute η and ι using the centers of N' clusters;
 - 8: replace N_p points with the centers of N' clusters and set $N_p = N'$;
 - 9: **return** N_p points;
-

$$\eta = \frac{\alpha(C \cap (\bigcup_{i=1}^n \odot(i, r + \epsilon)))}{\alpha(C)} \quad (4)$$

where $\alpha(\cdot)$ denotes the area of a geometry, $\odot(i, r + \epsilon)$ denotes a circle whose center is the i th point (in S_n), and is with the radius $r + \epsilon$. Note that, here ϵ is a parameter used to alleviate a too small radius. In our paper, we set ϵ as follows.

$$\epsilon = \operatorname{argmin}_{i,j \in [1, N_p], i \neq j} (\operatorname{dist}(PI_i, PI_j)) \quad (5)$$

where PI_i and PI_j denote two different points of interest.

Similarly, given a set S_n of n points in the convex hull C , the overlap region size, denoted by s , is defined as follows.

$$s = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \alpha(\odot(i, r + \epsilon) \cap \odot(j, r + \epsilon)) \quad (6)$$

where \cap denotes the intersection set of two circles. Based on s , we define the overlap degree, denoted by ι , as follows.

$$\begin{aligned} \iota &= \frac{s}{\sum_{i=1}^n \alpha(\odot(i, r + \epsilon))} \\ &= \frac{s}{n \times \alpha(\odot(i, r + \epsilon))} \end{aligned} \quad (7)$$

Let τ_η and τ_ι be two thresholds used for η and ι , respectively. The pseudocodes of the dynamic k-center algorithm are shown in Algorithm 1. Specifically, it first uses the set of N_p points of interest as the initial centers. It computes η and ι based on the above equations, and sets $\tau_\eta = d * \eta$, where $d \in [0, 1]$. In our paper, the parameters τ_ι and d are empirically chosen and set to 0.85 and 0.4, respectively. Then, it checks whether ι is less than τ_ι , where $\tau_\iota \in [0, 1]$. If so, the algorithm terminates and returns these N_p points as the centers. Otherwise, it uses k-means algorithm [15] to group these N_p points into $\lfloor \frac{N_p}{2} \rfloor$ clusters, and uses the centers of these $\lfloor \frac{N_p}{2} \rfloor$ clusters to compute “new” η and ι . This iteration terminates until one of the following two conditions is satisfied: (i) $\iota < \tau_\iota$, or (ii) $\eta < \tau_\eta$. Assume, without loss of generality, that our algorithm terminates after i times iterations, it shall return $\lfloor \frac{N_p}{2} \rfloor$ centers. In other words, $k = \lfloor \frac{N_p}{2} \rfloor$. Note that, the intuition behind the above algorithm is to select appropriate (i.e., k) points, based on points of interest, such that these k points can cover the convex hull C as much as possible (notice: each point associated with a circle), while the overlap degree of these circles should be small. Figure 6 shows an example of the iterative process of our algorithm above.

We are ready to generate the multi-center prior (MCP) map, based on the obtained k centers. One can observe that, these k centers obtained before could be not uniformly distributed in the convex hull C . To alleviate this issue, we attempt to give different weights for different centers (when we construct the MCP map). The weights are computed as follows.

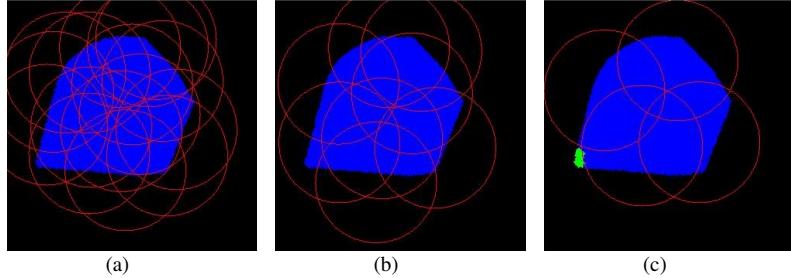


Figure 6: Example of the iterative process: (a) the overlap degree ι is larger than the threshold τ_ι , and set $\tau_\eta = d \times \eta$, where η is the coverage ratio and τ_η is the threshold used for the subsequent iterations; (b) the overlap degree ι is still larger than the threshold τ_ι , and the coverage ratio η is larger than the threshold τ_η ; (c) the overlap degree ι is smaller than the threshold τ_ι , although the coverage ratio η is still larger than the threshold τ_η ; notice that, the green part denotes it is not covered.

We first construct a $k \times k$ matrix \mathbb{M} , based on the distance of each pair of centers. That is,

$$\mathbb{M} = [dist(i, j)]_{k \times k} \quad (8)$$

where $i, j \in [1, k]$, and $dist(i, j)$ denotes the distance between the i th and j th centers. Then, for each column in \mathbb{M} , we accumulate all the values, getting a vector \mathbb{V} . That is,

$$\mathbb{V} = \left\{ \sum_{i=1}^k dist(i, 1), \sum_{i=1}^k dist(i, 2), \dots, \sum_{i=1}^k dist(i, k) \right\} \quad (9)$$

We normalize the vector \mathbb{V} , and then use the values in \mathbb{V} as the weighting coefficients of different centers, respectively. Finally, we obtain the MCP map by processing each superpixel sp_i as follows (notice: the essence of the equation below is to assign the saliency scores to each superpixel region in the image, which is similar to that in [58]).

$$S_i^{fg} = \frac{1}{k} \sum_{j=1}^k v_j \times e^{-\frac{(x_j - x_i)^2}{2\delta_x^2} - \frac{(y_j - y_i)^2}{2\delta_y^2}} \quad (10)$$

where v_j denotes the j th item in \mathbb{V} ; x_i (resp., y_i) denotes the mean horizontal (resp., vertical) coordinates of sp_i ; δ_x (resp., δ_y) denotes the horizontal (resp., vertical) variance; in our paper we set $\delta_x = \delta_y = 0.5$, unless otherwise stated.

We would like to point out that, our construction method developed above can also alleviate the issue that multiple objects are with different textures. For example, consider two objects in the image: an object is with the complex texture while another is very smooth. In this case, it is highly possible that most of points will locate on the object with complex texture, which will lead this object more highlighted, if our construction method is not used. In contrast, with the help of our construction method, this issue can be alleviated appropriately. To explain, see the example shown in Fig. 7; although the left object has more points of interest at the initial stage (cf., Fig. 7(a)), our dynamic k-center algorithm may choose only a point located on the left object as the final center (cf., Fig. 7(d)). This way, two objects can still be highlighted uniformly.

6. A more robust GCDP map

In this section, we present an optimization method that can allow us to construct a global color distinction prior (GCDP) map with a better quality. For short, we call it an *enhanced* GCDP map. Generally speaking, our optimization method can be viewed as a fusion of two ideas: (i) using separately each side of image boundaries to construct enhanced GCDP maps; and (ii) utilizing fully the convex hull (obtained before) to further optimize the GCDP maps. We next address more details.

It can be seen that existing methods (see e.g., [40]) usually group the superpixels in all boundaries into three clusters and then construct three different GCDP maps, based on these clusters. This method could incur poor results;

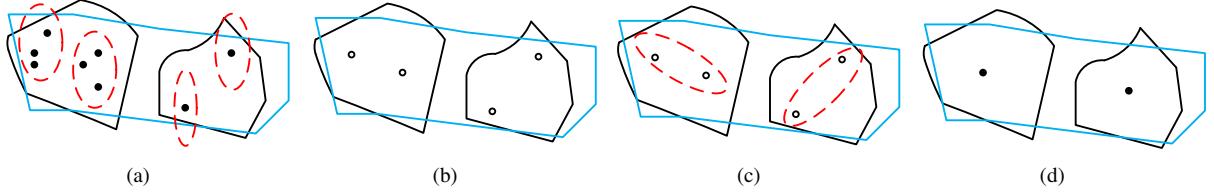


Figure 7: Illustration of two objects with different texture structures. The “left” object has more points of interest, while the “right” object has less points of interest. All these points of interest are found by Harris algorithm. The “blue” polygon denotes the convex hull constructed by existing algorithms. (a) the initial stage, here these points can be grouped into 4 clusters; (b) the centers of four clusters are looked as the new points, respectively; (c) four new points are further grouped into 2 clusters; (d) the centers of two clusters are looked as the new points, and assume that the algorithm in this step satisfies the termination condition, these two points shall be the output of our dynamic k-center algorithm.

see e.g., Fig. 8(c). Instead, in this paper we construct four different GCDP maps respectively, based on four boundaries. The intuition behind this idea is that, the background often presents the local or global consistence with one or some of four boundaries. Specifically, we do as follows.

Assume, without loss of generality, that there are N superpixels in the image, and the number of superpixels in the m th boundary is n_m ($m \in [1, 4]$), our method constructs the m th GCDP map, denoted by \mathbb{G}_m , as follows.

$$\left\{ \begin{array}{l} \mathbb{G}_m = [S_{m,i}]_{1 \times N} \\ S_{m,i} = \frac{1}{n_m} \sum_{j=1}^{n_m} \frac{1}{w(i, j) + \beta} \end{array} \right. \quad (11a)$$

$$\left\{ \begin{array}{l} \mathbb{G}_m = [S_{m,i}]_{1 \times N} \\ S_{m,i} = \frac{1}{n_m} \sum_{j=1}^{n_m} \frac{1}{w(i, j) + \beta} \end{array} \right. \quad (11b)$$

where the element $S_{m,i}$ denotes the saliency value of superpixel i in the m th GCDP map; $w(i, j) = \exp(-\frac{\|c_i - c_j\|}{2\delta_1^2})$, it measures the color similarity of the i th and j th superpixels. In our paper the balance parameters $\delta_1 = 0.2$ and $\beta = 10$, respectively. Our method above (i.e., utilizing four boundaries) is inspired by [59], but different from theirs, since they use Manifold ranking to construct the *side-specific map*, while we use Eqs. 11a and 11b to construct the *GCDP map*.

It is not hard to see that, the saliency result based on the GCDP map above, could still not well highlight some superpixel regions, whose colors are similar to that in the image boundaries; see e.g., the center part in Fig. 8(d). To alleviate this issue, we further optimize the above map by fully utilizing the convex hull obtained previously (recall Section 5) to restrain the saliency of superpixel regions outside C and highlight the saliency of superpixel regions inside C . This optimization can be achieved by revising Eq. 11b, and can further improve the quality of the result, as shown in Fig. 8(e).

Specifically, we do as follows. For each superpixel sp_i we first compute a value, denoted by D_i , as follows.

$$D_i = \begin{cases} \varphi_1 \sum_{j=1}^{n_m} \left(1 - \frac{1}{w(i, j) + \beta}\right) & sp_i \in R_F \\ \varphi_2 \sum_{j=1}^{n_m} \frac{1}{w(i, j) + \beta} & sp_i \in R_B \end{cases} \quad (12)$$

where φ_1 and φ_2 denote two weight factors, which are set to 0.8 and 0.2; R_F (resp., R_B) denotes the regions inside (reps., outside of) the convex hull C . Then, we integrate D_i into Eq. 11b, obtaining the following enhanced version. That is,

$$S_{m,i} = \frac{1}{n_m} \left(\sum_{j=1}^{n_m} \frac{1}{e^{(-\frac{\|c_i - c_j\|}{2\delta_1^2})}} + D_i \right) \quad (13)$$

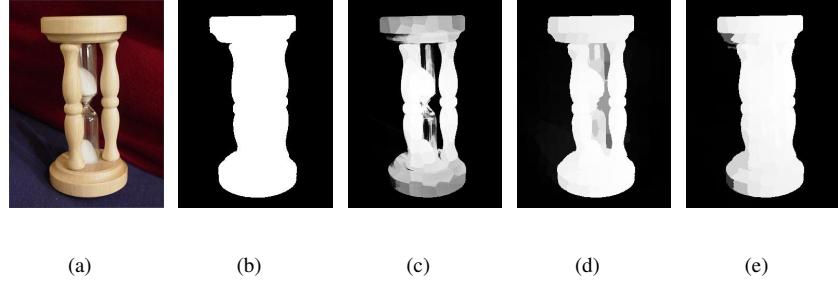


Figure 8: Our result compared with the original result: (a) input image; (b) ground truth; (c) saliency result based on the original GCDP map [40]; (d) saliency result based on our proposed model but without using idea (ii), recall the beginning of Section 6; (e) our result.

Finally, we merge all the four GCDP maps, obtaining the final GCDP map, denoted by \mathbb{G} , as follows.

$$\left\{ \begin{array}{l} \mathbb{G} = [S_i^{bg}]_{1 \times N} \\ S_i^{bg} = \prod_{m=1}^4 S_{m,i}, \quad i \in [1, N] \end{array} \right. \quad (14a)$$

$$(14b)$$

where the element S_i^{bg} denotes the saliency value of the i th superpixel in the final GCDP map.

Remark that, the method in [50] can be immediately used to integrate our MCP and GCDP maps. Specifically, for each superpixel sp_i , let S_i denote the saliency value (of the i th superpixel region) after integration. It is computed as follows.

$$S_i = S_i^{fg} \times (1 - e^{(-\lambda S_i^{bg})}) \quad (15)$$

where S_i^{fg} and S_i^{bg} have the same meanings with that in Eqs. 10 and 14a respectively, and λ is a balancing factor. Following [50], in our experiments λ is set to 6, unless otherwise stated.

7. Refining preliminary saliency map

While the integration method mentioned before can work correctly, the edges of the salient object could be not well preserved [56]. Previous works (e.g., [56, 55]) have attempted to further improve the saliency map via the *Bayesian optimization framework*. This optimization framework not only addresses the issue above, but also suppresses the background noise outside of the saliency region, to some extent. Yet, it ignores another issue: some superpixel regions (in the foreground), whose colors are similar to that in the background, could be also restrained; see e.g., Fig. 9(c). To address this issue, in this paper we suggest an improved Bayesian optimization framework, which achieves the better result, as shown in Fig. 9(d). The rationale behind our method is to assign larger weights for superpixels in C even if the colors in these superpixel regions are similar to that in the background, and then employ the Bayesian formula to compute a refined saliency map. Specifically, we do as follows.

Firstly, for each superpixel sp_i , we define the *superpixel region weight*, denoted by w_i , as follows.

$$w_i = \begin{cases} e^{-g_i}, & \text{if } sp_i \in R_F \\ e^{-(g_i+u)}, & \text{if } sp_i \in R_B \end{cases} \quad (16)$$

where $u \in (0, 2)$ is a parameter used to balance the size of the exponent item, in our experiments it is empirically set to 1; and g_i is computed as

$$\begin{aligned} g_i = & \frac{1}{N_1} \sum_{j_1=1}^{N_1} \text{Norm}(\|c_i, c_{j_1}\|) \\ & + (1 - \frac{1}{N_2} \sum_{j_2=1}^{N_2} \text{Norm}(\|c_i, c_{j_2}\|)) \end{aligned} \quad (17)$$

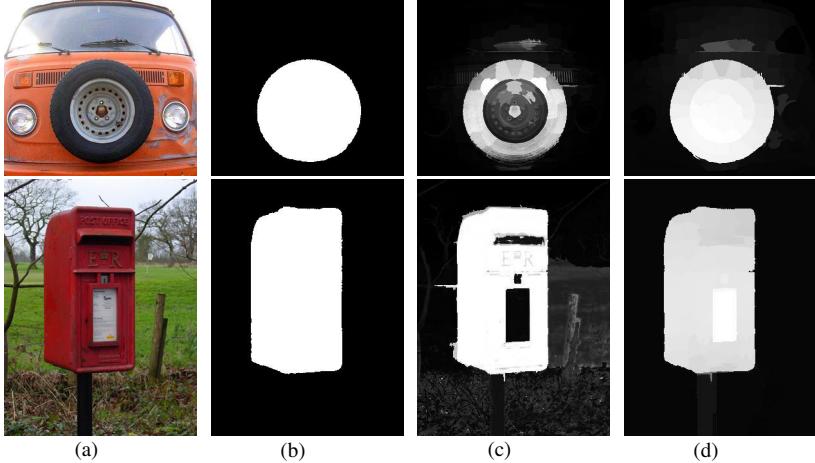


Figure 9: Our result compared with the original result: (a) input image; (b) ground truth; (c) original Bayesian framework saliency map [56]; (d) our result.

where N_1 (resp., N_2) is the number of superpixels in R_F (resp., R_B), c_i (resp., c_{j_1}) is the feature vector of a superpixel sp_i (resp., sp_{j_1}) in ‘‘CIELab color space’’, $\|c_i, c_{j_1}\|$ refers to the distance of two vectors, and $Norm(\|c_i, c_{j_1}\|)$ is obtained as follows. (Remark: $Norm(\|c_i, c_{j_2}\|)$ is obtained similarly.)

$$Norm(\|c_i, c_{j_1}\|) = \frac{\|c_i, c_{j_1}\| - \|c_i, c_{j_1}\|_{min}}{\|c_i, c_{j_1}\|_{max} - \|c_i, c_{j_1}\|_{min}} \quad (18)$$

where $\|c_i, c_{j_1}\|_{max}$ refers to $\text{argmax}_{j_1 \in [1, N_1]}(\|c_i, c_{j_1}\|)$ (i.e., the maximum one among all the $\|c_i, c_{j_1}\|$ where $j_1 \in [1, N_1]$), and $\|c_i, c_{j_1}\|_{min}$ refers to $\text{argmin}_{j_1 \in [1, N_1]}(\|c_i, c_{j_1}\|)$. The above normalization can be also viewed as a normalization of conventional Euclidean distance according to the image content. It is worth noting that, Eq. 16 essentially implies that, the region outside the convex hull shall be assigned relatively small weights. To some extent, this could suppress some background noises, improving the quality of the saliency map, from another perspective.

Second, for each superpixel region sp_i we compute the ‘‘weighted’’ observation likelihood in terms of the background and foreground, respectively. That is,

$$\begin{cases} p_w(sp_i|sal) = \frac{w_i}{\sum_{sp_j \in sal} w_j} \\ p_w(sp_i|bg) = \frac{w_i}{\sum_{sp_j \in bg} w_j} \end{cases} \quad (19a)$$

$$(19b)$$

where sal (resp., bg) denotes the foreground region R_F (resp., background region R_B); $p_w(s_i|sal)$ (resp., $p_w(s_i|bg)$) refers to the weighted observation likelihood of the superpixel sp_i in sal (resp., bg .)

Then, for each pixel v in the image, we compute the observation likelihood as follows.

$$\begin{cases} p(v|sal) = \sum_{sp_i \in sal} p_w(sp_i|sal)p(v|sp_i) \end{cases} \quad (20a)$$

$$\begin{cases} p(v|bg) = \sum_{sp_i \in bg} p_w(sp_i|bg)p(v|sp_i) \end{cases} \quad (20b)$$

where $p(v|sp_i)$ is the observation likelihood of pixel v in the superpixel sp_i , $p(v|sal)$ (resp., $p(v|bg)$) is the observation likelihood of the pixel v in sal (resp., bg).

The final step is the same as that in [56, 55], it constructs the final saliency map by Bayesian formula as described in Eq. 2.

8. Experimental evaluation

This section first describes the evaluation metrics, the datasets, and the compared methods, and then covers the experimental results.

8.1. Metrics, datasets, and methodologies

In this paper, we mainly use three evaluation metrics as follows:

- *Precision-recall* (P-R) curve. Following prior works [56, 40], we obtain the P-R curve by binarizing the saliency map, using thresholds in the range [0, 255]. Here the precision value, denoted by p_v , refers to the ratio of salient pixels correctly assigned to all the pixels of extracted regions; and the recall value, denoted by r_v , refers to the percentage of detected salient pixels with regard to the ground-truth number.
- *F-measure*. It is a comprehensive evaluation, and is computed as $F_m = \frac{(1+\varphi^2) \cdot p_v \cdot r_v}{\varphi^2 \cdot p_v + r_v}$, where φ^2 is set to 0.3, as the same as that in [40].
- *Mean absolute error* (MAE). It is the average distinction between the saliency map and the ground-truth, revealing the similarity between the saliency map and ground-truth. Denote by ϵ the MAE, it can be computed as $\epsilon = \frac{1}{n_a} \sum_{i=1}^{n_a} |\mathbf{S}(p_i) - \mathbf{G}(p_i)|$, where n_a is the number of all pixels in the image, $\mathbf{S}(p_i)$ (resp., $\mathbf{G}(p_i)$) denotes the information of the i th pixel from the saliency map (resp., the ground-truth).

We evaluate the algorithms based on seven widely used datasets in saliency detection field. They are:

- *SED2* [35]. It contains 100 images, in which each image is with two targets/objects.
- *ASD* [2]. It contains 1,000 images. This dataset is relatively simple, yet is widely used in almost all methods.
- *MSRA-5000* [33]. It contains 5,000 comprehensive source images with accurate masks, and covers a lot of scenarios such as flowers, animals, etc.
- *THUS* [44]. It contains 10,000 images, labelled with pixel-wise ground truth masks.
- *ECSSD* [57]. It contains 1,000 semantically meaningful but structurally complex images.
- *THUR* [7]. It contains 6,000+ images which usually contain complex backgrounds.
- *PASCAL* [10]. It includes 1,500 images with pixel-wise ground truth masks.

We mainly compare our proposed solution with two sets of algorithms. They are:

- Various *convex hull*-based saliency detection algorithms, including XL11 [55], GR [58], LMLC [56] and MS [47].
- Other classical and/or state-of-the-art saliency detection methods, including MAP [44], RBD [74], SF [39], PCA [35], IT [19], DSR [27], RC [8], HS [57], GC [9], FT [1], CA [13], GMR [59], SVO [6] and LPS [25].

8.2. Comparing with convex hull based algorithms

This section focuses on examining the performance on the SED2 dataset, in which the images are usually with two targets².

Fig. 10 shows the comparative results of our algorithm and existing convex hull-based saliency detection algorithms. On one hand, from Fig. 10(a) we can see that, the P-R curve of our method dominates the ones of other convex hull based algorithms³. This demonstrates the effectiveness of our solution.

On the other hand, from Fig. 10(b) one can see that our MAE (resp., F-measure) value is the smallest (resp., largest) one, among all these convex hull based algorithms⁴. This further verifies the effectiveness of our proposed algorithm.

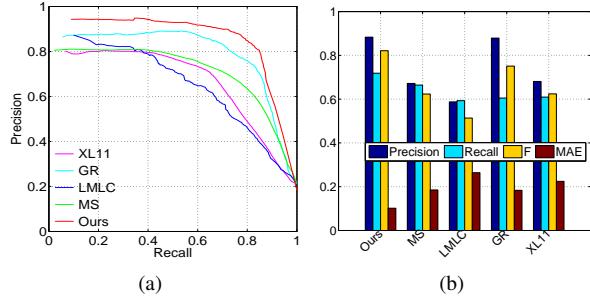


Figure 10: Comparison results of various convex-hull based methods over SED2. The curves in these figures denote the P-R curves.

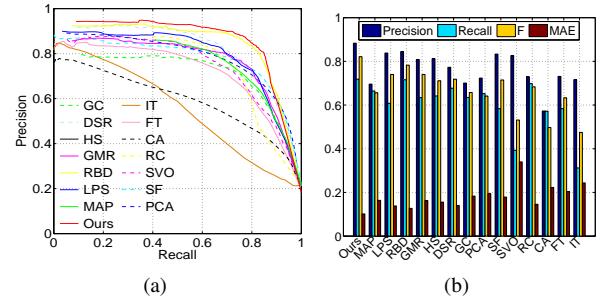


Figure 11: Comparison results of other methods and our proposed method over SED2. The curves denote the P-R curves.

8.3. Comparing with other algorithms

This section compares our algorithm with other state-of-the-art algorithms, using the dataset above (i.e., SED2). Fig. 11 shows the comparative results.

We can see from Fig. 11(a) that, the P-R curves of our algorithm is above the curves of all these competitors, although the P-R curves of some algorithms are close to ours. This essentially reflects the competitiveness of our proposed algorithm.

In addition, we can see from Fig. 11(b) that, the F-measure value of our method is the largest one among all these algorithms. This further demonstrates the competitiveness of our method. Also, it can be seen from Fig. 11(b) that, the MAE value of our algorithm is less than the ones of these competitors. This demonstrates that, (on average) the saliency results generated by our algorithm are more close to the ground truth masks.

All these results consistently show us that the proposed algorithm can perform well on the SED2 dataset, in which the images are with multiple objects. Essentially, this indirectly reflects that the MCP map is helpful for salient object detection over images with multi-targets.

8.4. Performance results on other datasets

To achieve a more comprehensive validation on our proposed solution, this section conducts extensive tests using more datasets. In what follows, we first summarize the similarities about the performance results on these several datasets, and then analyze results on different datasets, respectively.

Fig. 12 shows the results on six widely used datasets. There are three common features: (i) the P-R curve of our algorithm is above all (or almost all) the P-R curves of these competitors; (ii) the MAE of our algorithm is smaller than all (or almost all) these competitors; (iii) the F-measure of our algorithm is larger than all (or almost all) these competitors. Essentially, these evidences validate the competitiveness of our algorithm. We now examine the results on these datasets, separately.

▷ *ASD*. Besides three features mentioned before, we also note that, the P-R curve of our algorithm is highly close to that of RBD [74] (see Fig. 12(a)), and the precision and F-measure of our algorithm are superior than that of RBD (see Fig. 12(b)). Combining these facts together, one can easily understand that, as a whole, our algorithm can perform well over *simple* datasets like ASD.

▷ *ECSSD*. Figs. 12(c) and 12(d) show the results over the ECSSD dataset. Note that, the F-measure value of our algorithm is the largest one among all these algorithms. We can get that, on the whole our algorithm is pretty promising for the datasets like ECSSD, in which the images are semantically meaningful but structurally complex.

▷ *MSRA-5000*. Figs. 12(e) and 12(f) present the results over the MSRA-5000 dataset. Note that, the F-measure value of our algorithm is the largest one, and the MAE value is the smallest one. These phenomena confirm that our algorithm is very competitive for the MSRA-5000 dataset, in which different kinds of contents are included,

²Note that, convex hulls constructed from multiple targets have the high probabilities to be polygons with the large *spans*, recall Section 1.

³Remark that, when two algorithms are compared each other, the algorithm whose P-R curve is located at the above is usually more superior [46].

⁴By the large, usually the larger (resp., smaller) the F-measure (resp., MAE) value is, the better the algorithm is. More information about the relation between the algorithm performance and the MAE (or F-measure) please see prior works (e.g., [74, 40]).

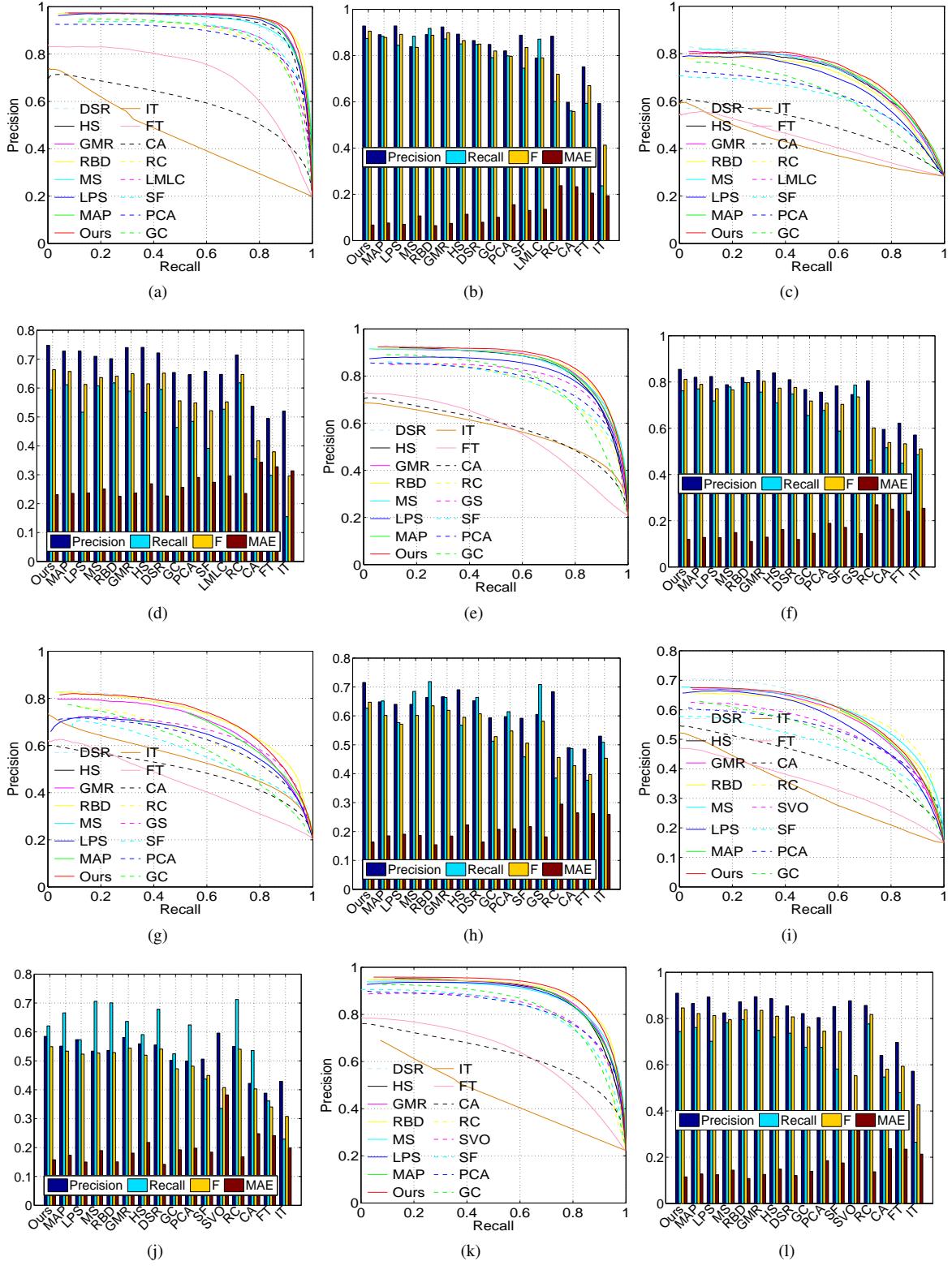


Figure 12: The results on various datasets. From left to right: (a-b) the ASD dataset; (c-d) the ECSSD dataset; (e-f) the MSRA5000 dataset; (g-h) the PASCAL dataset. (i-j) the THUR dataset; (k-l) the THUS dataset. The curves in these figures denote the P-R curves.

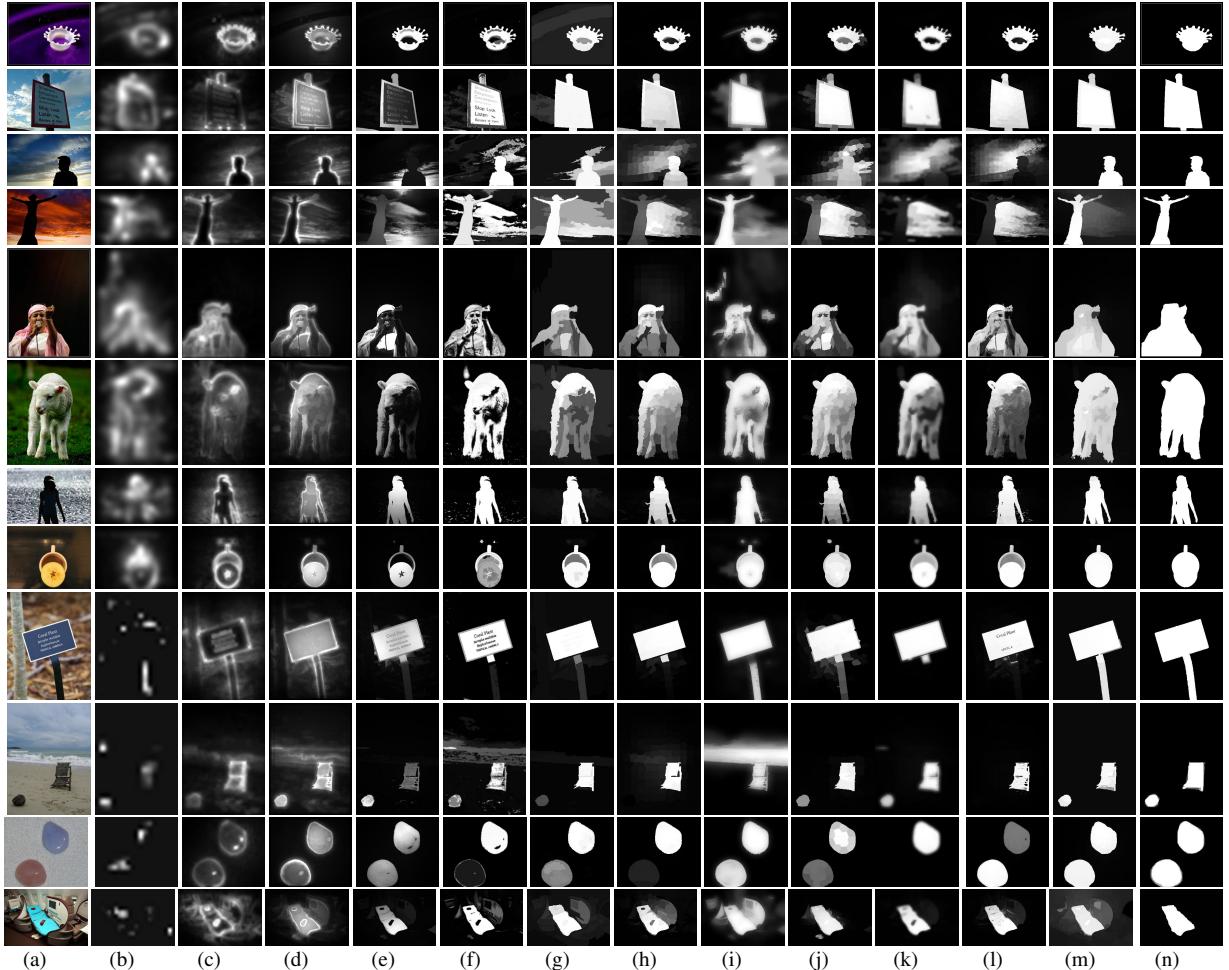


Figure 13: The saliency maps obtained based on different methods: (a) input images; (b) IT; (c) CA; (d) PCA; (e) SF; (f) GC; (g) HS; (h) GMR; (i) MS; (j) RBD; (k) MAP; (l) LPS; (m) Ours; and (n) GT (ground-truth).

the background is usually large while the salient object is relatively compact, and the color contrast between the foreground and background is small.

▷ *PASCAL*. Figs. 12(g) and 12(h) cover the results over the PASCAL dataset. One can see that, the F-measure value of our algorithm is the largest one among these algorithms. This verifies the effectiveness of our algorithm, since the F-measure evaluates the performance of algorithms in a more comprehensive way [40, 74, 46]. In other words, these results demonstrate that our algorithm can perform well for the datasets like PASCAL, in which the images are with complicate backgrounds and the objects/targets in the images have complex shapes.

▷ *THUR*. For the THUR dataset, one can observe that the F-measure value of our algorithm is the highest one among all these algorithms (see Fig. 12(j)). This shows that the proposed solution performs well on the THUR dataset, to some extent. However, we have to point out that, our algorithm is slightly inferior to DSR on this dataset (i.e., THUR), especially when the recall value is larger than 0.6 (see Fig. 12(i)). The reason could be that the THUR database contains many images with some unique features (e.g., serious background noise), rendering us hard to get relatively precise convex hull and thus weakening the performance of our method. This fact demonstrates that there is still some improvement space for us to further enhance the proposed algorithm. In the future, we are ready to develop other new techniques (or optimization strategies) to further improve the performance. We leave this interesting research topic as our future work.

▷ *THUS*. Figs. 12(k) and 12(l) show the results over this dataset. On one hand, the F-measure value of our

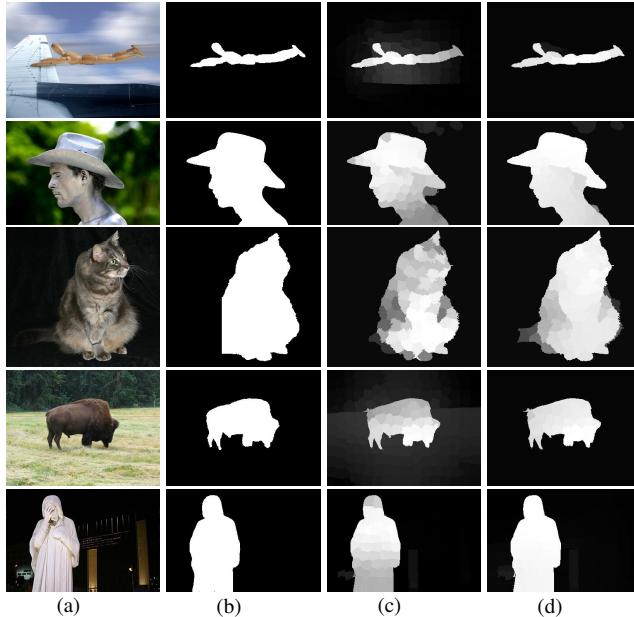


Figure 14: From left to right: (a) input image; (b) ground-truth; (c) without using the MCP map; (d) our solution.

algorithm is the highest among all these algorithms. On the other hand, one can observe that, in terms of MAE, our algorithm is inferior than RBD; yet, our MAE value is very close to that of RBD. Note that, the MAE value is only one of the quantitative evaluations (let alone qualitative evaluation). Overall, all these facts show that our algorithm performs well on the THUS dataset, which is a large-scale benchmark dataset, and each image in the dataset has an unambiguous salient object.

8.5. Saliency maps obtained based on various methodologies

To evaluate our algorithm qualitatively, this section covers representative saliency maps obtained based on various methods. The results are shown in Fig. 13. Note that, in the saliency maps the bright pixel-regions indicate that they are assigned with high saliency values.

It is not hard to understand that, these competitors have their unique merits. For example, the CA approach uses global clues to estimate the saliency regions; it can generate high saliency values on the boundaries of saliency regions (e.g., the 3rd row). The PCA method uses colors and high-level cues to identify the unique pattern; it is helpful to detect the whole saliency regions (e.g., the 9th row). The SF method uses the contrast-based saliency measure; it is helpful to differentiate the foreground and background (e.g., the 7th row). The GC method employs color and spatial contrasts; it benefits to finding small salient objects (e.g., the 10th row). The HS method uses a hierarchical model to find salient objects; it can work well for images with complex structures (e.g., the 3rd row). The GMR model uses the graph-based manifold ranking to process the similarity of the image elements; it can efficiently highlight saliency regions (e.g., the 8th row). The MS method uses multi-scale analysis and integrates the Bayesian framework to improve the quality of saliency maps (e.g., the 7th row). The RBD method uses a principled optimization framework to integrate multiple low-level cues; it can obtain clean and uniform saliency maps (e.g., the 6th row). The MAP algorithm exploits the relationship between the saliency detection and the Markov absorption probability; it can perform well for images with complex patterns (e.g., the 6th row). The LPS method achieves saliency detection via label propagation; it can detect salient objects even from low contrast foreground and cluttered background (e.g., the 2nd row).

Clearly, these methods mentioned above have also their limitations. For example, some foreground regions similar to the backgrounds are mistaken as the backgrounds for most competitors (see e.g., the 1st and 8th rows); the saliency regions are only partially detected for some competitors (e.g., the 9th row). In summary, they cannot perform well for all kinds of images. In this regard, our method has also the limitation (e.g., see the 4th and 12th rows, the background noises are not suppressed sufficiently).

Table 2: Ours vs. baselines (bsl1: without using the MCP map; bsl2: without using the enhanced GCDP map; bsl3: without using improved Bayesian framework)

Metric Method \ Metric	F	MAE	Precision	Recall
BSL1	0.8611	0.0727	0.8604	0.9347
Ours	0.8917	0.0436	0.8786	0.9843
BSL2	0.9872	0.0458	0.9952	0.9620
Ours	0.9952	0.0304	0.9957	0.9939
BSL3	0.9021	0.0584	0.8697	0.9752
Ours	0.9245	0.0439	0.9113	0.9737

Yet, we would like to emphasize that, the overall performance of our method is still competitive. For example, for images in which some pixels' colors in the foreground are extremely similar to that in the background, our model can favourably highlight these pixels, as shown in the 1st, 2nd, 8th and 9th rows. Also, one can observe that, for images in which the objects' colors are black and the background patterns are sophisticated, most of methods cannot highlight the salient object efficiently, while our model can accurately highlight the salient object, as shown in the 3rd, 4th and 7th rows. In addition, for images in which the color contrast between the foreground and the background is large while the object is relatively dim, most of methods cannot highlight the salient object uniformly, our model achieves relatively good effects, as shown in the 5th and 6th rows. Last but not least, for images containing multiple objects, some competitors highlight only one of the objects, others could fail to detect the entire salient objects, while our model not only accurately detects all the objects but also highlights them properly, achieving satisfactory saliency maps, as shown in the 10th and 11th rows.

In summary, all these evidences mentioned above consistently show that our model is effective and also competitive, compared with the classic and state-of-the-art saliency detection models.

8.6. Effectiveness of proposed techniques/strategies

In this section, we verify the effectiveness of the proposed model, from another perspective. Recall previous sections, we present three major techniques and/or optimization strategies: (i) a multi-center prior (MCP) map; (ii) a more robust global color distinction prior (GCDP) map; and (iii) an improved Bayesian optimization framework. To evaluate the effectiveness of each technique, we replace it with the traditional technique, and then compare it against our proposed solution.

▷ *Multi-center prior map.* For ease of validating the effectiveness of this multi-center prior map, we compare our proposed solution with the following baseline method (known as BSL1): we use the method in [73] to construct the convex hull, and then use the algorithm in [58] to construct the single-center prior map; the rest of steps are the same as our proposed model. Fig. 14 reports the saliency results generated by this baseline method and our proposed solution (notice: Table 2 summarizes the quantitative results of representative images, for reference). It can be seen that, although the baseline method employs other two optimizations (i.e., the enhanced GCDP map and the improved Bayesian optimization framework), the saliency maps generated by this baseline method is obviously inferior to that of our proposed solution. Specifically, the baseline method highlights some parts of the saliency region, while other parts in the saliency region are not well highlighted. In contrast, our proposed solution highlights the saliency region uniformly, compared with the baseline method. Moreover, one can observe that, for the baseline method, some background regions that are near to the center of the “highlighted” saliency region are also highlighted (see e.g., the first and fourth rows in Fig. 14). Yet, our method overcomes this limitation. This essentially demonstrates the effectiveness of our MCP map that assigns different weights for different centers, avoiding over-highlighting some parts.

▷ *Enhanced global color distinction prior map.* In this set of experiments, we compare our method with the following baseline method (known as BSL2): it generates the multi-center prior map using our proposed technique, and then uses existing algorithms [40] to construct the GCDP map; the rest of steps are the same as our proposed method. Fig. 15 shows the comparison results (Table 2 summarizes the quantitative results, for reference). It is not

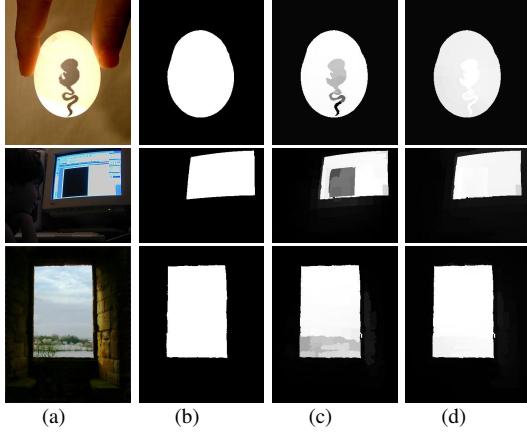


Figure 15: From left to right: (a) input image; (b) ground-truth; (c) without using the enhanced GCDP map; (d) our solution.

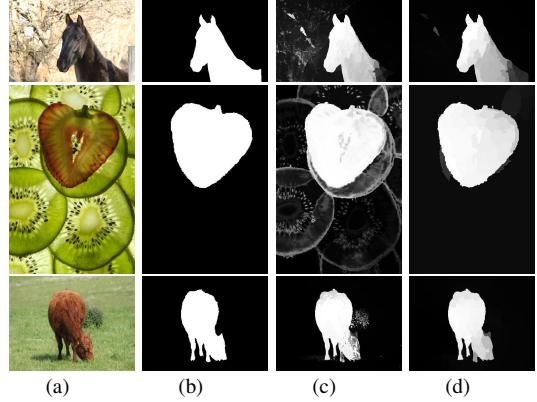


Figure 16: From left to right: (a) input image; (b) ground-truth; (c) without using the improved Bayesian framework; (d) our solution.

Table 3: Comparison of running time

Method	IT	CA	PCA	SF	GC	HS	GMR	MS	RBD	LPS	Ours
Time	0.2151	32.3432	3.2575	0.1642	0.0316	0.4072	0.4113	2.6534	0.1713	1.0762	0.8224
Code	M	M+C	M+C	C	C	C	M	M	M	M	M+C

hard to see that, for the saliency maps generated by the baseline method, some parts in the saliency region are still dark (see e.g., the second row in Fig. 15(c)). Essentially, one can see (cf., Fig. 15(a)) that, the colors in these parts are either similar to the background, or significantly different from other parts in the saliency regions, incurring that the baseline method cannot highlight these parts. In contrast, our proposed model can favourably highlight these parts, demonstrating the effectiveness of the enhanced GCDP map.

▷ *Improved Bayesian optimization framework.* We compare our proposed solution with the following baseline method (known as BSL3): it uses the method in [73] to estimate the approximate saliency region, and employs the Bayesian optimization method in [56] to obtain the final saliency result. We remark that, other steps (e.g., computing MCP map and enhanced GCDP map) are same to our proposed solution. Fig. 16 depicts the saliency results obtained by these two methods (Table 2 summarizes the quantitative results, for reference). From this figure we can see that, compared with the saliency maps generated by this baseline method, the saliency maps generated by our solution further suppress the noises located in the saliency regions. Also, one can observe that, for some images with complex backgrounds (e.g., the second row in Fig. 16), the baseline method cannot well suppress the background noises, while our solution achieves relatively satisfactory results. This implies that the improved Bayesian optimization framework can bring us an extra benefit — further suppressing background noises, as mentioned in Section 7.

8.7. Running time

We compare the average running time of our algorithm against that of representative algorithms and state-of-the-art algorithms. In our test, all experiments are conducted on a PC with 3.50 GHz CPU and 32 GB RAM. The dataset used in our test is ASD [2]. Table 3 covers the comparison results, in which “M” means “Matlab”, and “C” means “C/C++”; the time unit is in seconds, and the average running time refers to the time used to process each single image. It can be seen that our algorithm consumes relatively less time (i.e., no more than 1 sec). Combining the superiorities validated in previous sections, on the whole our proposed solution is competitive.

8.8. Discussion

As we mentioned earlier, our method belongs to bottom-up models [56, 74, 57, 40, 46]. In the field of saliency detection, there is a common sense, as pointed out in [69], “the bottom-up model has a wide range of applications

Table 4: Other Results

Method	ELD	RFCN	DCL	UCF	GBR	NLDF	Amulet	Ours
F	0.771	0.832	0.810	0.818	0.824	0.831	0.763	0.658
MAE	0.121	0.118	0.115	0.116	0.107	0.099	0.098	0.167

because it mainly depends on some low-level visual features (e.g., color, intensity or orientation) and some prior knowledge (e.g., contrast, compactness, uniqueness or boundary). On the contrary, top-down models, including CNN-based deep learning ones, capture representative high-level features, thereby detecting salient objects of certain sizes and categories. In general, their performance is better than that of bottom-up models. However, top-down methods often need the time-consuming training process.”

We also basically agree the above common sense. For example, Table 4 summarizes the F-measure values and MAE values on the PASCAL dataset, with regard to several deep learning based methods, including ELD [22], RFCN [51], DCL [23], UCF [71], GBR [45], NLDF [34], Amulet [70]. From this table it can be seen that this line of methods obtain higher F-measure values and lower MAE values, which are better than our results. In addition, the infer time of these deep learning based solution is pretty small (e.g., NLDF consumes only about 0.081 seconds for a single image on average, Amulet consumes only about 0.062 seconds for a single image on average). On the other hand, we also observe that most of deep learning based methods take a long time to train their models (e.g., GBR [45] consumes 12 *hours* on an NVIDIA GTX-1080Ti GPU and an Intel E5-2630 CPU processor, DCL [23] consumes 25 *hours* on an NVIDIA Titan Black GPU and a 3.4GHz Intel processor, UCF [71] consumes 23 *hours* on an NVIDIA Titan X GPU and an i7-4790 CPU, NLDF consumes about 9 hours on NVIDIA Tian X GPU, Amulet consumes about 16 hours on a NVIDIA Titan X GPU and an i7-4790 CPU). These phenomena essentially further validate the common sense mentioned above.

9. Conclusions

In this paper, we proposed a new convex hull prior based saliency detection method. The central idea of our method is to dynamically select k centers to construct a multi-center prior map. Besides, our solution also integrates two non-trivial optimizations that advance existing techniques. We conducted extensive experiments to validate the effectiveness of our proposed solution. Experimental results demonstrate that our proposed solution is competitive and attractive, compared against many classical and/or state-of-the-art saliency detection algorithms. In the future, we would like to adapt our solution as well as other state-of-the-art techniques to specific applications such as visualization navigation and volume visualization.

Acknowledgements

We thank very much the editors and anonymous reviewers for their efforts in evaluating our manuscript. This work was partially supported by the National Key R&D Program of China (2018YFC0810204, 2018YFB1004400), and the NSFC (61472245, 61472453, 61502220, 61572326, 61775139, 61872242, U1401256, U1501252, U1611264, U1711261, and U1711262), and Shanghai Science and Technology Innovation Action Plan Project (16111107502, 17511107203), and the program for tackling key problems in Henan Science and Technology (172102310636).

References

- [1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.
- [2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurlien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.
- [3] M. M. Naushad Ali, M. Abdullah-Al-Wadud, and Seok-Lyong Lee. Multiple object tracking with partial occlusion handling using salient feature points. *Inf. Sci.*, 278:448–465, 2014.
- [4] Ali Borji, Dicky N Sihite, and Laurent Itti. Probabilistic learning of task-specific visual attention. In *CVPR*, pages 470–477, 2012.

- [5] Feilong Cao, Yuehua Liu, and Dianhui Wang. Efficient saliency detection using convolutional neural networks with feature selection. *Inf. Sci.*, 456:34–49, 2018.
- [6] Kai-Yueh Chang, Tyng-Luh Liu, Hwann-Tzong Chen, and Shang-Hong Lai. Fusing generic objectness and visual saliency for salient object detection. In *ICCV*, pages 914–921, 2011.
- [7] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, and Shi-Min Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, 30(4):443–453, 2014.
- [8] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, 2015.
- [9] Ming-Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook. Efficient salient region detection with soft image abstraction. In *ICCV*, pages 1529–1536, 2013.
- [10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [11] Yuming Fang, Weisi Lin, Zhijun Fang, Zhenzhong Chen, Chia-Wen Lin, and Chenwei Deng. Visual acuity inspired saliency detection by using sparse features. *Inf. Sci.*, 309:1–10, 2015.
- [12] Dashan Gao, Sunhyoung Han, and Nuno Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):989–1005, 2009.
- [13] Stas Goferman, Lih Zelnik-Manor, and Ayallet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.
- [14] Chen Gong, Dacheng Tao, Wei Liu, Stephen J Maybank, Meng Fang, Keren Fu, and Jie Yang. Saliency propagation from simple to difficult. In *CVPR*, pages 2531–2539, 2015.
- [15] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques, Third Edition*. Morgan Kaufmann, Massachusetts, 2011.
- [16] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, pages 10–5244, 1988.
- [17] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. Deeply supervised salient object detection with short connections. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5300–5309, 2017.
- [18] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.
- [19] Laurent Itti, Christof Koch, Ernst Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
- [20] Muwei Jian, Kin-Man Lam, Junyu Dong, and Linlin Shen. Visual-patch-attention-aware saliency detection. *IEEE Transactions on Cybernetics*, 45(8):1575–1586, 2015.
- [21] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *ICCV*, pages 1976–1983, 2013.
- [22] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features. In *CVPR*, pages 660–668, 2016.
- [23] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016.
- [24] Heng Li, Tingting Han, Jing Wang, Zhuofan Lu, Xiaofei Cao, Yao Chen, Liming Li, Chuangqing Zhou, and Xinyu Chai. A real-time image optimization strategy based on global saliency detection for artificial retinal prostheses. *Inf. Sci.*, 415:1–18, 2017.
- [25] Hongyang Li, Huchuan Lu, Zhe Lin, Xiaohui Shen, and Brian Price. Inner and inter label propagation: salient object detection in the wild. *IEEE Transactions on Image Processing*, 24(10):3176–3186, 2015.
- [26] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *CVPR*, pages 2806–2813, 2014.
- [27] Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, pages 2976–2983, 2013.
- [28] Yansheng Li, Yongjun Zhang, Jin-Gang Yu, Yihua Tan, Jinwen Tian, and Jiayi Ma. A novel spatio-temporal saliency approach for robust dim moving target detection from airborne infrared image sequences. *Inf. Sci.*, 369:548–563, 2016.
- [29] Yijun Li, Keren Fu, Lei Zhou, Yu Qiao, and Jie Yang. Saliency detection via foreground rendering and background exclusion. In *ICIP*, pages 3263–3267, 2014.
- [30] Ming Liang and Xiaolin Hu. Feature selection in supervised saliency prediction. *IEEE Transactions on Cybernetics*, 45(5):900–912, 2015.
- [31] Zhen Liang, Binggang Xu, Zheru Chi, and David Dagan Feng. Relative saliency model over multiple images with an application to yarn surface evaluation. *IEEE Transactions on Cybernetics*, 44(8):1249–1258, 2014.
- [32] Risheng Liu, Junjie Cao, Zhouchen Lin, and Shiguang Shan. Adaptive partial differential equation learning for visual saliency detection. In *CVPR*, pages 3866–3873, 2014.
- [33] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaolu Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):353–367, 2011.
- [34] Zhiming Luo, Akshaya Kumar Mishra, Andrew Achkar, Justin A. Eichel, Shaozi Li, and Pierre-Marc Jodoin. Non-local deep features for salient object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6593–6601, 2017.
- [35] Ran Margolin, Ayallet Tal, and Lih Zelnik-Manor. What makes a patch distinct? In *CVPR*, pages 1139–1146, 2013.
- [36] Xiongkuo Min, Guangtao Zhai, Ke Gu, Jing Liu, Shiqi Wang, Xinfeng Zhang, and Xiaokang Yang. Visual attention analysis and prediction on human faces. *Inf. Sci.*, 420:417–430, 2017.
- [37] Kang Han Oh, Myungeun Lee, Yura Lee, and Soo-Hyung Kim. Salient object detection using recursive regional feature clustering. *Inf. Sci.*, 387:1–18, 2017.
- [38] Houwen Peng, Bing Li, Rongrong Ji, Weiming Hu, Weihua Xiong, Congyan Lang, et al. Salient object detection via low-rank and structured sparse matrix decomposition. In *AAAI*, pages 796–802, 2013.

- [39] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.
- [40] Yao Qin, Huchuan Lu, Yiqun Xu, and He Wang. Saliency detection via cellular automata. In *CVPR*, pages 110–119, 2015.
- [41] Esa Rahtu, Juho Kannala, Mikko Salo, and Janne Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, pages 366–379, 2010.
- [42] Xiaohui Shen and Ying Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860, 2012.
- [43] Christian Siagian and Laurent Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007.
- [44] Jingang Sun, Huchuan Lu, and Xiuping Liu. Saliency region detection based on markov absorption probabilities. *IEEE Transactions on Image Processing*, 24(5):1639–1649, 2015.
- [45] Xin Tan, Hengliang Zhu, Zhiwen Shao, Xiaonan Hou, Yangyang Hao, and Lizhuang Ma. Saliency detection by deep network with boundary refinement and global context. In *ICME*, pages 1–6, 2018.
- [46] Na Tong, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Salient object detection via bootstrap learning. In *CVPR*, pages 1884–1892, 2015.
- [47] Na Tong, Huchuan Lu, Lihe Zhang, and Xiang Ruan. Saliency detection with multi-scale superpixels. *IEEE Signal processing letters*, 21(9):1035–1039, 2014.
- [48] Joost Van de Weijer, Theo Gevers, and Andrew D Bagdanov. Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):150–156, 2006.
- [49] Fanglin Wang, Yi Zhen, Bineng Zhong, and Rongrong Ji. Robust infrared target tracking based on particle filter with embedded saliency detection. *Inf. Sci.*, 301:215–226, 2015.
- [50] Jianpeng Wang, Huchuan Lu, Xiaohui Li, Na Tong, and Wei Liu. Saliency detection via background and foreground seeds election. *Neurocomputing*, 152:359–368, 2015.
- [51] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.
- [52] Weining Wang, Dong Cai, Xiangmin Xu, and Alan Wee-Chung Liew. Visual saliency detection based on region descriptors and prior knowledge. *Signal Processing: Image Communication*, 29(3):424–433, 2014.
- [53] Wenguan Wang, Jianbing Shen, Ling Shao, and Fatih Porikli. Correspondence driven saliency transfer. *IEEE Transactions on Image Processing*, 25(11):5025–5034, 2016.
- [54] Zhi-Jie Wang, Lizhuang Ma, Xiao Lin, and Hui Zhong. Saliency detection via multi-center convex hull prior. In *The 43rd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, 2018.
- [55] Yulin Xie and Huchuan Lu. Visual saliency detection based on bayesian model. In *ICIP*, pages 645–648, 2011.
- [56] Yulin Xie, Huchuan Lu, and Ming-Hsuan Yang. Bayesian saliency via low and mid level cues. *IEEE Transactions on Image Processing*, 22(5):1689–1698, 2013.
- [57] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.
- [58] Chuan Yang, Lihe Zhang, and Huchuan Lu. Graph-regularized saliency detection with convex-hull-based center prior. *IEEE Signal Processing Letters*, 20(7):637–640, 2013.
- [59] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.
- [60] Jin-Gan Yu, Ji Zhao, Jinwen Tian, and Yihua Tan. Maximal entropy random walk for region-based visual saliency. *IEEE Transactions on Cybernetics*, 44(9):1661–1672, 2014.
- [61] Jun Yu, Zhenzhong Kuang, Baopeng Zhang, Wei Zhang, Dan Lin, and Jianping Fan. Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing. *IEEE Trans. Information Forensics and Security*, 13(5):1317–1332, 2018.
- [62] Jun Yu, Baopeng Zhang, Zhengzhong Kuang, Dan Lin, and Jianping Fan. iprivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Trans. Information Forensics and Security*, 12(5):1005–1016, 2017.
- [63] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Trans. Neural Netw. Learning Syst.*, 29(12):5947–5959, 2018.
- [64] Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 1114–1120, 2018.
- [65] Dingwen Zhang, Junwei Han, and Yu Zhang. Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 4068–4076, 2017.
- [66] Jian Zhang, Jun Yu, and Dacheng Tao. Local deep-feature alignment for unsupervised dimension reduction. *IEEE Trans. Image Processing*, 27(5):2420–2432, 2018.
- [67] Jing Zhang, Sheng-Wei Feng, Da Li, Yong-Wei Gao, Zhihua Chen, and Yubo Yuan. Image retrieval using the extended salient region. *Inf. Sci.*, 399:154–182, 2017.
- [68] Jing Zhang, Tong Zhang, Yuchao Dai, Mehrtash Harandi, and Richard I. Hartley. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 9029–9038, 2018.
- [69] Lihe Zhang, Jianwu Ai, Bowen Jiang, Huchuan Lu, and Xiukui Li. Saliency detection via absorbing markov chain with learnt transition probability. *IEEE Transactions on Image Processing*, 27(2):987–998, 2018.
- [70] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 202–211, 2017.
- [71] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Baocai Yin. Learning uncertain convolutional features for accurate saliency

- detection. In *ICCV*, pages 212–221, 2017.
- [72] Yongdong Zhang, Zhendong Mao, Jintao Li, and Qi Tian. Salient region detection for complex background images using integrated features. *Inf. Sci.*, 281:586–600, 2014.
- [73] Hengliang Zhu, Bin Sheng, Xiao Lin, Yangyang Hao, and Lizhuang Ma. Foreground object sensing for saliency detection. In *ICMR*, pages 111–118, 2016.
- [74] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014.