Fusing Generic Objectness and Visual Saliency for Salient Object Detection

Kai-Yueh Chang[1,2] Tyng-Luh Liu[1] Hwann-Tzong Chen[2] Shang-Hong Lai[2]
[1]Institute of Information Science, Academia Sinica, Taiwan
[2]Department of Computer Science, National Tsing Hua University, Taiwan

Abstract

We present a novel computational model to explore the relatedness of objectness and saliency, each of which plays an important role in the study of visual attention. The proposed framework conceptually integrates these two concepts via constructing a graphical model to account for their relationships, and concurrently improves their estimation by iteratively optimizing a novel energy function realizing the model. Specifically, the energy function comprises the objectness, the saliency, and the interaction energy, respectively corresponding to explain their individual regularities and the mutual effects. Minimizing the energy by fixing one or the other would elegantly transform the model into solving the problem of objectness or saliency estimation, while the useful information from the other concept can be utilized through the interaction term. Experimental results on two benchmark datasets demonstrate that the proposed model can simultaneously yield a saliency map of better quality and a more meaningful objectness output for salient object detection.

1. Introduction

Visual attention relates to how humans process different information in a scene, and is primarily analyzed through investigating the allocation of eye fixations. For vision research, the attention process concerns particularly with two important concepts, namely, *saliency* and *objectness* [3, 12]. The main focuses of this paper are to investigate the close relationship between the two concepts, introduce a new formulation to effectively approximate them, and propose a useful algorithm for salient object detection.

To model where we place our eyes in a visual scene, the saliency-based approaches, *e.g.*, [5, 10, 12, 20], are considered to be the mainstream in the vision community. In such techniques, the visual saliency is typically computed in a bottom-up fashion, as is in the computational model by Itti and Koch [10] where information gathering from local image characteristics such as color, intensity, and orientation is used. Since no high-level cognitive cue is explored in the computation, visual saliency is more appropriate to account for *free-view* eye fixations. Alternatively, for *task-dependent* visual attention, additional use of relevant high-level information would more likely help better explain the fixation distribution [18, 23].

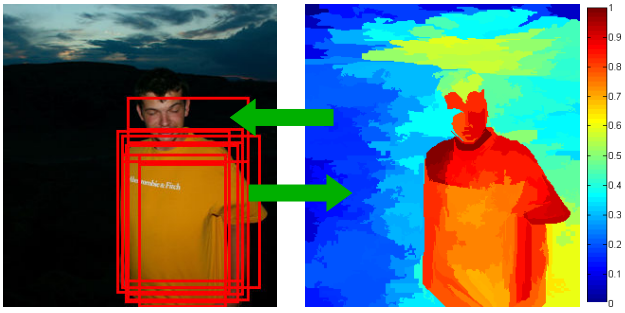
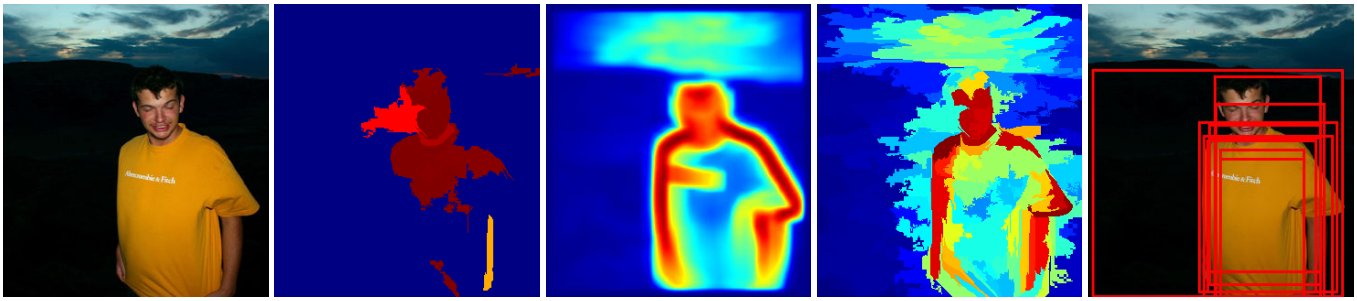


Figure 1. Mutual improvements between objectness and saliency. Left: Each red window indicates a possible object within it. (Note that we always draw the top 10 objectness windows for illustration.) Right: The estimated saliency map, where the color bar on the right shows the scales of saliency values.

It is evident that salient object detection would be more appropriately casted as a task-dependent attention model instead of a free-view one. In this case, the top-down high-level information is dictated by what kinds of objects we are to detect, or equivalently, by how the concept of objectness is defined. Our formulation adopts similar criteria of objectness described in [3], and therefore encompasses a huge set of general objects. By coupling visual saliency and generic objectness into a unified framework, the proposed approach can not only yield good performance of detecting salient objects in a scene but also concurrently improve the quality of both the saliency map and the objectness estimations. (See Figure 1 for an illustration.)

Despite the close connection between saliency and objectness, existing approaches [3, 5, 12, 14, 15, 19, 20] still lack a good computational model to thoroughly explore the two concepts and their mutual effects. Only one single direction of the interactions is considered: The techniques in [5, 12, 15] exploit object detection to help saliency estimations, while those in [3, 14, 19, 20] use saliency information



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

Figure 2. Inefficiency of saliency detection and object detection. From left to right: (a) the original image, (b) the ground-truth saliency by recording eye fixations [12], (c) saliency map detected by [5], (d) region-based saliency map, and (e) the objects detected by [3].

to improve object detection or segmentation. Either way they see the insufficiencies in each of the two models pertaining to visual attention. Specifically, in most saliency detection methods, an edge pixel is often assigned a stronger saliency value since it is more distinct with respect to the whole image. (See Figure 2c.) Even when a region-based saliency detection scheme, *e.g.*, [2] is used to alleviate this effect, the resulting saliency map (*e.g.*, Figure 2d) is still quite different from the one based on the ground-truth eye fixations (*e.g.*, Figure 2b). On the other hand, in linking objectness to the allocation of eye fixations, it seems impractical to detect all the objects in a scene by simply designing specific purposed object detectors (*e.g.*, face detector) in that the number of all possible object classes is simply too large to handle. One may instead consider using a general purposed object detector [3]. However, the main drawback is that the detection result may include too many "objects" that are less likely to attract visual attention. Indeed, the detector might even not perform well due to its general purpose, as is shown in Figure 2e.

Our method begins by randomly sampling a large number of windows. Each window is assigned an objectness value and each pixel (or superpixel) a saliency value. To link objectness with saliency, we first introduce an object-level saliency for each window. This value is used to represent the saliency of the underlying object within each window. On using objectness to help estimate saliency, a top-down viewpoint is adopted so that the saliency of a pixel should be decided by the object-level saliency values of those windows enclosing it and with high objectness. On using saliency to help estimate objectness, the objectness of a window will be high if the object-level saliency can well explain most of the saliency values of the pixels it covers. To measure the object-level saliency, instead of using the conventional center-surround scheme, we make use of the shape information provided by superpixels to form a new measurement. In our experiments, the proposed model with this measurement not only performs better on the mean average precision but also on the visualization.

2. Related work

Most saliency detection methods [1, 2, 5, 7, 8, 9, 11, 15] consider only the low-level features. They basically compute the saliency values by a center-surround approach [2]. That is, if a pixel is more "different" from its surrounding area, then it is assigned a higher saliency value. According to the size of the surrounding area, these methods can be further divided into two groups. For techniques in the first group [7, 11, 15], saliency computation proceeds by comparing each pixel with its local neighborhood. While Itti *et al.* [11] compute the saliency values by the difference of Gaussians (DoG) operations, Ma and Zhang [15] just exploit the differences to the nearby pixels. In [7], Harel *et al.* further consider the appearance difference and the geometric distance of each pixel pair, and then use a graph-based method to estimate the saliency. In methods of the second group [1, 2, 5, 8, 9], the saliency estimation would require comparing each pixel with a large area or even the whole image. The technique by Hou and Zhang [8] analyzes the frequency domain of an image. It seeks the frequencies with uncommon change in magnitude, and identifies pixels with high response to these frequencies as salient. Besides using the frequency filters, they [9] also consider the filters learned from natural images. Now, the filters with low responses to a given image are uncommon and those pixels owning high response to such filters are salient. Unlike searching for the uncommon filters, Goferman *et al.* [5] directly compare each pixel with the whole image and pick out only those most similar to it. If a pixel is still dissimilar to its most similar ones, it is salient. Instead of computing so many differences, Achanta *et al.* [1, 2] use an average operation to summarize the information of the surrounding area of a pixel so that its saliency value can be determined based on the summarized information. Among the aforementioned techniques [1, 2, 5, 7, 8, 9, 11, 15], pixels are essentially processed independently. Since pixels near the edges are quite distinct with respect to those in other areas of an image, methods of this kind often assign higher saliency values to them. To address this problem, pixels are

either compared with the ones in a very low-pass filtered images to enhance the saliency of the non-edge ones [2], or considered in a region-based manner by assuming that those in the same superpixel have similar property and thus share the same saliency value.

When the top-down information is used for saliency detection, it generally requires a learning phase to incorporate the high-level knowledge into the process. Itti and Koch [10] as well as Ma and Zhang [15] both describe an abstract concept about how to include such information in the saliency computation. More practical implementations can be found in [5, 12], where (object-specific) detection results are used to generate a corresponding binary map. Goferman *et al*. [5] then directly take a max operation to combine the bottom-up and top-down results while Judd *et al*. [12] treat each map as a feature and use supervised learning to build a saliency classifier. Deviating from the conventional bottom-up concept, Moosmann *et al*. [17] define saliency from a top-down viewpoint, and treat it as an attribute to most distinguish a concept of interest from others. In this sense, finding such a saliency map can be reduced to learning a classifier, which is exactly what have been proposed by Wang and Forsyth [22], Marchesotti *et al*. [16], and Moosmann *et al*. [17].

Turning now our attention to detect or segment objects, we particularly focus on the techniques that accomplish the task via referencing a saliency map. Hou and Zhang [8] and Achanta *et al*. [2] both use an image-dependent saliency threshold to segment the objects. Liu *et al*. [14] and Rahtu *et al*. [19] instead take saliency as one of the features in the unary term of a conditional random field model. Once the parameters have been learned, single salient object detection can be achieved through inference. In addition to these approaches, Ramanathan *et al*. [20] explore the fixations and analyze their effects on the clustering-based segmentation results. Rather than detecting salient objects, Alexe *et al*. [3] aim for generic object detection. Nevertheless, saliency is still used as a cue in their naïve Bayes model.

Recent research efforts have made three databases available to the vision community. The database of Kienzle *et al*. [13] contains 200 grayscale natural images with 18065 fixation locations recorded from 14 subjects. The database of Judd *et al*. [12] consists of the fixations from 15 subjects across 1003 color images and the fixations are mostly found around objects. In the database of Ramanathan *et al*. [20], the fixations are recorded from 75 subjects and result in 758 images. Since this set contains more emotional or meaningful images, the fixations are thus strongly influenced by scene semantics. However, the fixations in these three databases are just sparse points in each image. Such information should be further processed when used in evaluating the goodness of a dense saliency map.

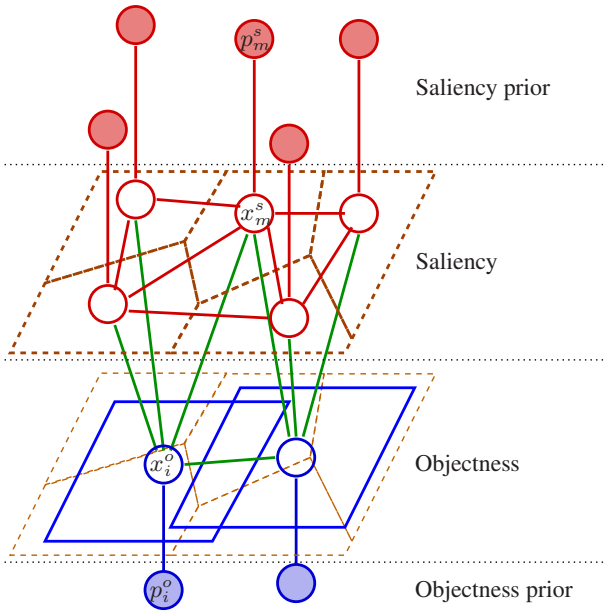


Figure 3. Our graphical model for fusing saliency and objectness. The red, blue and green edges respectively correspond to the energy appeared in F_s, F_o and Δ in (1).

3. Fusing objectness and saliency

For the computational concern, we define the saliency over superpixels yielded by over-segmenting an image [4]. Besides, representing an image by superpixels also provides the convenience of utilizing the shape information, which will be discussed in Section 3.3. Assuming that an image contains P superpixels and Q potential object windows, we are to estimate the saliency x_m^s for superpixel $m \in \{1, \ldots, P\}$ and the objectness x_i^o for window $i \in \{1, \ldots, Q\}$. For convenience, the vector notations $\mathbf{x}^s \in [0,1]^P$ and $\mathbf{x}^o \in [0,1]^Q$ will be used to respectively represent the saliency values of all superpixels and the objectness values of all windows. Estimating $\mathbf{x}^s$ and $\mathbf{x}^o$ by our method is achieved by minimizing the following energy function:

$$F(\mathbf{x}^s, \mathbf{x}^o) = F_s(\mathbf{x}^s) + F_o(\mathbf{x}^o) + \Delta(\mathbf{x}^s, \mathbf{x}^o) \qquad (1)$$

where F_s includes the energy affected only by saliency, F_o contains the energy affected only by objectness, and Δ models the interactions between saliency and objectness. Figure 3 is an illustration of the graphical model for (1).

3.1. Saliency energy

To construct the saliency energy F_s in (1), we consider a smoothness term by assuming that the saliency values of a pair of adjacent superpixels should not differ too much if their appearances are similar. In addition, if we have a prior knowledge p_m^s about the saliency of superpixel m, the

estimated saliency x_m^s also should not deviate too far away from this value. We thus define the saliency energy by

$$F_s(\mathbf{x}^s) = \sum_m (p_m^s - x_m^s)^2 + \lambda_s \sum_{m,n\in\mathcal{E}} w_{m,n}(x_m^s - x_n^s)^2 \quad (2)$$

where λ_s is the weight of the smoothness term, $\mathcal{E}$ is the set containing the pairs of adjacency superpixels, and $w_{m,n}$ is the affinity between superpixels m and n given by

$$w_{m,n} = \sum_{(k,l)\in B_{m,n}} \exp\left(-\sigma\|\mathbf{v}_k - \mathbf{v}_l\|^2\right) \quad (3)$$

where $\mathbf{v}_k$ and $\mathbf{v}_l$ are respectively the RGB values of pixels k and l, and $B_{m,n}$ contains the pairs of adjacent pixels across the boundary of superpixels m and n. The second term in (2) prefers that in minimizing F_s the saliency values x_m^s and x_n^s should be close if their corresponding superpixels m and n are similar (*i.e.*, large $w_{m,n}$).

The saliency prior p_m^s can be conveniently estimated by the existing bottom-up saliency detection methods. In our experiments, we use the pixel-wise saliency detection by Goferman *et al.* [5]. To obtain p_m^s, we then average all the saliency values of the pixels within superpixel m.

3.2. Objectness energy

Recall that the objectness is to be estimated for each generated window. We assume that all such windows are independent without the saliency information. Thus, given some prior knowledge p_i^o about the objectness of each window i, the objectness energy F_o in (1) can be defined as follows:

$$F_o(\mathbf{x}^o) = \lambda_o \sum_i (p_i^o - x_i^o)^2 \quad (4)$$

where λ_o is the weight of the objectness energy. To come up with a reasonable prior p_i^o, we consider the objectness framework in [3]. However, among other image features, the detector also uses the saliency cue. It implies that a direct application of such an objectness detector would be inappropriate to our formulation. We exploit the fact that the detector is formed by a naïve Bayes model where each cue is considered independently, and modify it by removing the saliency cue in all our experiments.

3.3. Interaction energy

The interaction energy Δ in (1) plays the most pivotal role in the energy function F. It is designed to model the relationships between superpixel-wise saliency and window-wise objectness. To further explore the explicit form of Δ, we need to define the following concept.

Definition 1 *Given a window i, its object-level saliency $c_i \in [0,1]$ is said to measure the degree of the difference of a specific feature distribution between the center (inside the window) and the surround (around the window) areas.*

Figure 4. An illustration of the center and the surround areas given a set of superpixels and an object window. The superpixels mostly inside the window (red) are considered as the center area (brown) while the superpixels around the center area are considered as the surround area (green).

The definition of the object-level saliency is loose, and leaves the freedom to choose what kind of the image feature to compute and how to measure the degree of the difference between two distributions. In addition, observe that superpixels often contain the shape information in that the pixels inside a superpixel are generally *consistent* and the boundary of an object usually locates on the boundaries of superpixels. We thus make use of this property in deciding the areas used in the center-surround scheme. Specifically, we define the area covered by those superpixels that fall mostly inside a given window ($\geq 80\%$ in our experiments) as the center area, and the area formed by the neighboring superpixels around the center area as surround. (See Figure 4.)

Given a window i, let $\mathbf{h}_{i,c}$ and $\mathbf{h}_{i,s}$ respectively represent the distributions of its center and surround areas. (In this work, the distribution is obtained by running K-means algorithm ($K = 20$) on a texton image [21].) We then compute the χ^2 distance between the two distributions, *i.e.*,

$$\chi^2(\mathbf{h}_{i,c}, \mathbf{h}_{i,s}) = \sum_k \frac{(\mathbf{h}_{i,c}(k) - \mathbf{h}_{i,s}(k))^2}{(\mathbf{h}_{i,c}(k) + \mathbf{h}_{i,s}(k))/2}. \quad (5)$$

Since the χ^2 distance can range from 0 to ∞, we use a sigmoid function to re-scale its value to $[0,1]$ and define the object-level saliency of window i as

$$c_i = \frac{1}{1 + \exp\left(-(\chi^2(\mathbf{h}_{i,c}, \mathbf{h}_{i,s}) - \bar{\chi}^2)\right)} \quad (6)$$

where $\bar{\chi}^2$ denotes the average χ^2 distance over all windows.

Consider now some superpixel m. It typically will be covered by multiple windows, and each of them has an object-level saliency value. Altogether, they form a top-down view about the saliency of m, which is given by

$$\tau_m^s = \frac{\sum_{\{i|m\in w_i\}} c_i x_i^o}{\sum_{\{i'|m\in w_{i'}\}} x_{i'}^o}. \quad (7)$$

Clearly, τ_m^s is a (normalized) sum of object-level saliency values weighted by their respective objectness. With (7),

we are now ready to define the interaction energy by

$$\Delta(\mathbf{x}^s, \mathbf{x}^o) = \lambda \sum_m (\tau_m^s - x_m^s)^2 \tag{8}$$

where λ is the weight of the interaction energy. To see why the interaction energy Δ defined in (8) is the main factor for concurrently improving the saliency and the objectness estimations, we first assume that the objectness information is given for all windows. Then, the interaction energy would provide the top-down information as in (7) to help the saliency computation. The justification for the other direction will be given in the next section, where the related equation is reduced to a more comprehensive form.

4. Optimization

To minimize (1), we see that the optimization problem is convex in $\mathbf{x}^s$ when $\mathbf{x}^o$ is fixed, and almost convex in $\mathbf{x}^o$ when $\mathbf{x}^s$ is fixed due to the denominator in (7). The inconvenience can be overcome with a reasonable assumption that the change in $\mathbf{x}^o$ is small between two successive iterations. It follows that $\mathbf{x}^o$ in the denominator of (7) can be replaced by the estimate at the last iteration.

In detail, to solve $\mathbf{x}^s$ when $\mathbf{x}^o$ is fixed, we minimize

$$\|\mathbf{p}^s - \mathbf{x}^s\|^2 + \lambda_s \mathbf{x}^{s\mathrm{T}} L \mathbf{x}^s + \lambda \|\boldsymbol{\tau}^s - \mathbf{x}^s\|^2 \tag{9}$$

where $\mathbf{p}^s$ and $\boldsymbol{\tau}^s$ are the vector forms of p_m^s and τ_m^s, respectively. $L = D - W$ is the Laplacian matrix, where W is an affinity matrix with $W(m,n) = w_{m,n}$ and D is a diagonal matrix with $D(m,m) = \sum_n w_{m,n}$. In the form of (9), we have a closed-form solution

$$\mathbf{x}^s = (\lambda_s L + (1+\lambda) I)^{-1} \cdot (\mathbf{p}^s + \lambda \boldsymbol{\tau}^s) \tag{10}$$

where I is the identity matrix. Note that $(\lambda_s L + (1+\lambda)I)^{-1}$ needs to be computed only once since it is fixed throughout the iterations.

To solve $\mathbf{x}^o$ when $\mathbf{x}^s$ is fixed, we first replace $x_{i'}^o$ in the denominator of (7) with $\tilde{x}_{i'}^o$, the estimate in the last iteration. Then we minimize

$$\lambda_o \|\mathbf{p}^o - \mathbf{x}^o\|^2 + \lambda \|C\mathbf{x}^o - \mathbf{x}^s\|^2 \tag{11}$$

where $\mathbf{p}^o$ is the vector form of p_i^o and C is defined by

$$C(m,i) = \frac{c_i}{\sum_{\{i'|m \in w_i\}} \tilde{x}_{i'}^o} \times \delta[m \in w_i] \tag{12}$$

where $\delta[\cdot]$ is the indicator function. From the second term of (11), we can see how saliency helps estimate objectness. Given $\mathbf{x}^s$, it would yield large objectness values to the windows, whose object-level saliency well accounts for the corresponding $\mathbf{x}^s$ (*i.e.*, the saliency values of the superpixels covered by a window). We use the `cvx` toolbox [6] to solve (11). The optimization proceeds by iteratively solving (9) and (11) until the energy in (1) can not be further reduced.

5. Experimental results

Two datasets are used in our experiments. For evaluating objectness, we choose the set $\mathcal{B}$ by Liu *et al*. [14], which contains 5000 images, each of which has object bounding boxes labeled with higher agreement. For each image, we directly average the bounding boxes labeled by 9 subjects as the ground-truth window w_g. We then randomly sample 10000 windows $\{w_i\}_{i=1}^{10000}$ and decide, with respect to w_g, their ground-truth (objectness) label g_i^o by

$$g_i^o = \begin{cases} 1, & \text{if } \frac{|w_i \cap w_g|}{|w_i \cup w_g|} \geq T_o, \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

where T_o is the threshold to decide the degree of consistency to w_g for a window to be considered as positive. In our experiments, for the sake of comparison, methods dealing with objectness use the same 10000 windows in each image. The parameters λ_s, λ_o and λ in (2), (4) and (8) are respectively set to fixed values, namely, $\frac{1}{64}$, $\frac{1}{40}$ and 16 across all images. Note that the large weight of the interaction energy confirms its importance in our model.

For comparing saliency, we use the dataset by Judd *et al*. [12] since it contains more "object" images than the other two [13, 20]. From the set, we further choose 373 images that each contains more obvious objects. Pertaining to generating the ground-truth saliency maps, we consider two reasonable implementations. The first is the same as what Judd *et al*. adopt in [12] to make the maps by applying Gaussian smoothing to the fixation data. The second is motivated by that since the fixations in each image are sparse, their surrounding area subject to similar appearance is likely to receive the same degree of attention. It follows that for each superpixel m of size a_m, we can define the degree of attention d_m to be the number of fixations inside it divided by a_m. An example is shown in Figure 2b. To decide which superpixels or pixels are salient, we follow a similar scheme in [2], and set the threshold as

$$T_s = 2 \times \frac{\sum_m d_m a_m}{\sum_{m'} a_{m'}}. \tag{14}$$

Note that the ground-truth saliency map is still in pixel level, as the superpixel information is used just for deciding which pixels should receive the same degree of attention.

Our method is run on a PC with Intel i7 CPU @ 2.8 GHz. It takes about 30 seconds per image to solve (1) with Matlab and the `cvx` toolbox [6]. Now, for each window i, we have its optimal objectness value x_i^o as well as the object-level saliency c_i. The product of these two quantities, $c_i \times x_i^o$, can then be used for the salient object detection. In addition, we have for each superpixel m its optimal saliency value x_m^s. By letting all pixels in superpixel m have the same saliency value x_m^s, we can obtain a saliency map in pixel level.

5.1. Evaluation criteria

A number of saliency detection methods [2, 5, 8, 11, 12] have been included for comparison. Each algorithm has been tested by resizing all the images into a set of predetermined dimensions, and the case that yields the best performance is listed in the second column of Table 1. Note that each detection result is obtained by first re-scaling the saliency map back to its original size, and then evaluated in pixel level. Since the number of the positive (salient) points is relatively small in the ground-truth data, the average precision (AP) would be more suitable for comparing the performances. The average precision is the area under the recall-precision curve. Thus, the negative points ranked below all positive ones will not be considered. Assume that we have a ranked list $\mathbf{r}$ and the ground-truth label list $\mathbf{g}$, the average precision is then given by

$$\left(\sum_i \frac{\sum_{j=1}^{i} \mathbf{g}(\mathbf{r}(j))}{i} \times \mathbf{g}(\mathbf{r}(i))\right) \Big/ \left(\sum_{i'} \mathbf{g}(\mathbf{r}(i'))\right). \tag{15}$$

Analogously, average precision is used in evaluating the performances of objectness estimation in that the number of positive windows is also relatively small. (See the second column of Table 2.) For all images, we compute the mean average precision (mAP) as the overall performance.

5.2. Saliency detections

Table 1 reports the saliency detection results in mean average precision by ours and other methods [2, 5, 8, 12, 11]. Our model is implemented in two versions, Ours-Rect and Ours-SP. They differ in how the center-surround areas are decided for computing the window-wise object-level saliency. The former uses a conventional center-surround layout based on two rectangles, while the latter adopts the superpixel-based scheme described in Section 3.3. Note that [12] uses a supervised learning scheme while the others only process each image directly. Our method achieves better saliency detection results among the learning-free. In Figure 5, examples of saliency maps by the various techniques are provided for visualizing the detection quality. In rows 1, 2, 4 and 5 of Figure 5, the saliency of the edge part by [5, 8, 11, 12] is overemphasized, while the saliency inside the objects by ours is visually more reasonable. Comparing with Ours-Rect, the saliency detected by Ours-SP is more conspicuous in the object boundary. (See rows 1, 2, 4 and 5 of Figure 5.) We also show the salient objects detected (*i.e.*, based on the product $c_i \times x_i^o$) by Ours-SP. The detected windows fit the objects well and are able to recover multiple objects.(See rows 1, 2, 3 and 6 of Figure 5.) Take, for example, the row 1 of Figure 5: The proposed method detects not only a person but also his face. In the last row of Figure 5, we show an example of less satisfactory results of saliency detections in a more challenging situation.

Method	Size	mAP-Gaussian	mAP-SP
[11]	200×200	0.2692	0.2481
[2]	Whole image	0.2007	0.1963
[8]	32×32	0.2931	0.2782
[5]	100×100	0.3885	0.3697
[12]	200×200	**0.4536**	0.4176
Ours-Rect	Whole image	0.3934	0.4177
Ours-SP	Whole image	0.4076	**0.4284**

Table 1. Saliency detection results with respect to two ground-truth settings of saliency maps derived by smoothing the fixation maps with a Gaussian filter (mAP-Gaussian) or superpixels (mAP-SP).

5.3. Objectness estimations

For the objectness estimation, we compare our method with the generic object detector proposed by Alexe *et al*. [3]. The results in mean average precision are listed in Table 2. Since we have used a modified version of [3] to obtain the objectness prior $\mathbf{p}^o$ in (4) and (11), the results by their model without using the saliency cue are also provided in the fourth column of Table 2. This way it is convenient to see the improvements by our model. The experiments are carried out with respect to different values of the threshold T_o, which directly controls the hardness of fitting a ground-truth bounding box. From Table 2, it can be concluded that whether the object-level saliency information is used or not, the proposed method significantly improves the results in the fourth column (denoted as [3]\Saliency) and also those by Alexe *et al*. [3]. It is worth mentioning that our method may detect meaningful but not salient objects, *e.g*. the land in row 2 and columns 3-4 of Figure 6. Overall, the object detection results by our method are generally better in visualization than those by Alexe *et al*. [3]. It can also detect multiple objects (*e.g*., row 5). Besides, the salient objects detected by Ours-SP often better align with the ground truth (*e.g*., rows 3 and 4). The last row of Figure 6 shows a failed example, caused by the distinct center of the flower and the ambiguity between the flower and the background.

6. Conclusion

Objectness and saliency, the two concepts are somewhat contrasted to each other in that the former is essentially cognitive-based, and the latter is simply image-based. Our formulation emphasizes the use of generic objectness to avoid the fusion being dominated by strong high-level object information, and meanwhile to promote their interactions. Our experimental results have shown that the proposed energy minimization can simultaneously improve the quality of saliency and objectness estimations. In addition, it also yields the information of object-level saliency, with which the objectness estimation can be extended to salient object detection. Our future work would further explore the vision applications of the object-level saliency.

T_o	Positive windows	[3]	[3]\Saliency	Objectness $\{x_i^o\}$		Salient objectness $\{c_i \times x_i^o\}$	
				Ours-Rect	Ours-SP	Ours-Rect	Ours-SP
0.5	31.56%	0.5082	0.4938	0.5114	**0.5224**	0.5120	**0.5358**
0.6	15.07%	0.3353	0.3292	0.3435	**0.3567**	0.3420	**0.3934**
0.7	5.11%	0.1797	0.1806	0.1877	**0.1998**	0.1847	**0.2579**
0.8	1.01%	0.0658	0.0685	0.0698	**0.0770**	0.0696	**0.1383**
0.9	0.06%	0.0127	0.0130	0.0131	**0.0149**	0.0148	**0.0442**

Table 2. Results of objectness estimations in mean average precision (mAP).

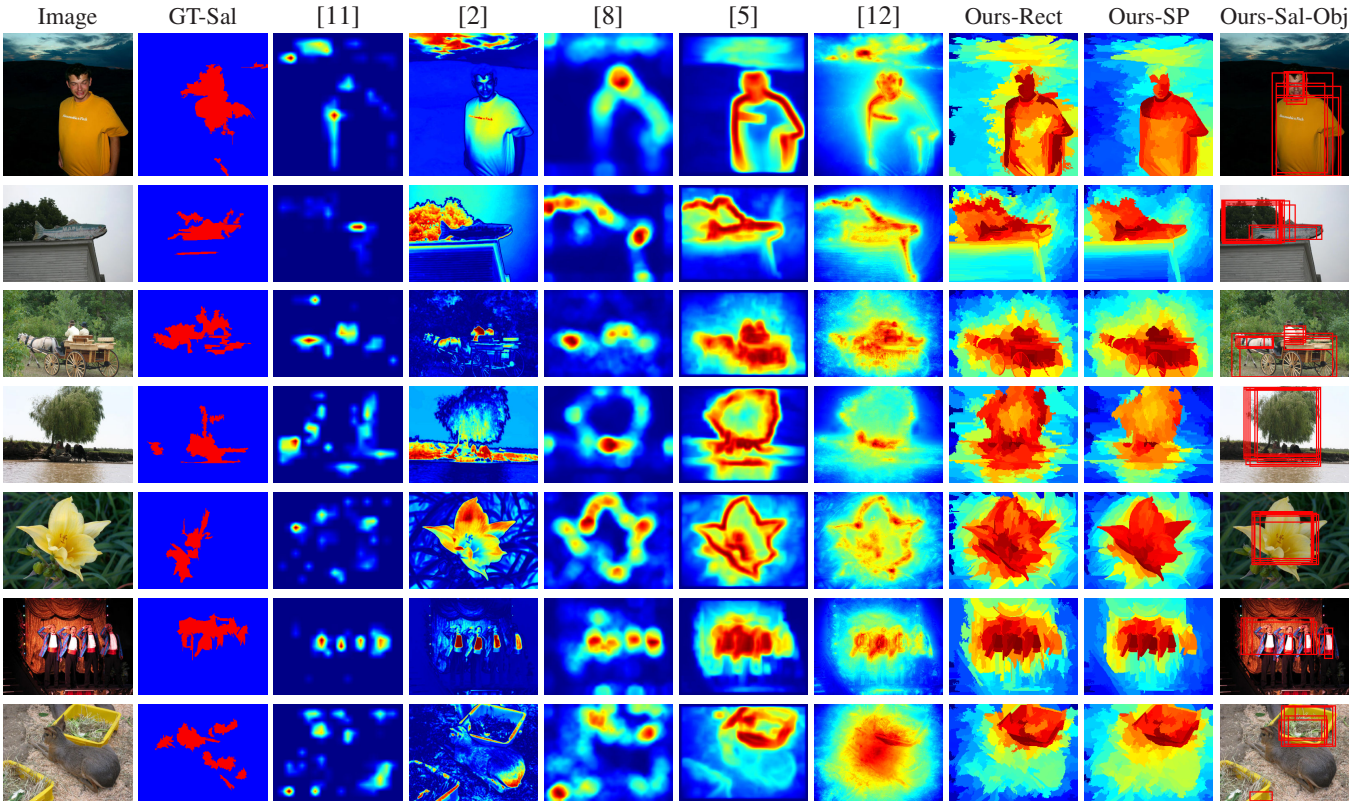


Figure 5. Examples of saliency detection results by different methods are shown in columns 3 through 9, while the salient objects detected by Ours-SP are given in the last column. The last row shows an example of failed detections. (GT-Sal denotes ground-truth saliency.)

Acknowledgements

We want to thank the anonymous reviewers for their valuable suggestions. This work is supported in part by NSC grants 97-2221-E-001-019-MY3 and 99-2221-E-001-011-MY3.

References

[1] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk. Salient region detection and segmentation. In *ICVS*, pages 66–75, 2008.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

[4] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, September 2004.

[5] S. Goferman, L. Zelnik Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383, 2010.

[6] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, Oct. 2010.

[7] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS 19*, pages 545–552, 2007.

[8] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[9] X. Hou and L. Zhang. Dynamic visual attention: searching for coding length increments. In *NIPS 21*, pages 681–688, 2009.

[10] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Review Neuroscience*, 2(3):194–203, March 2001.

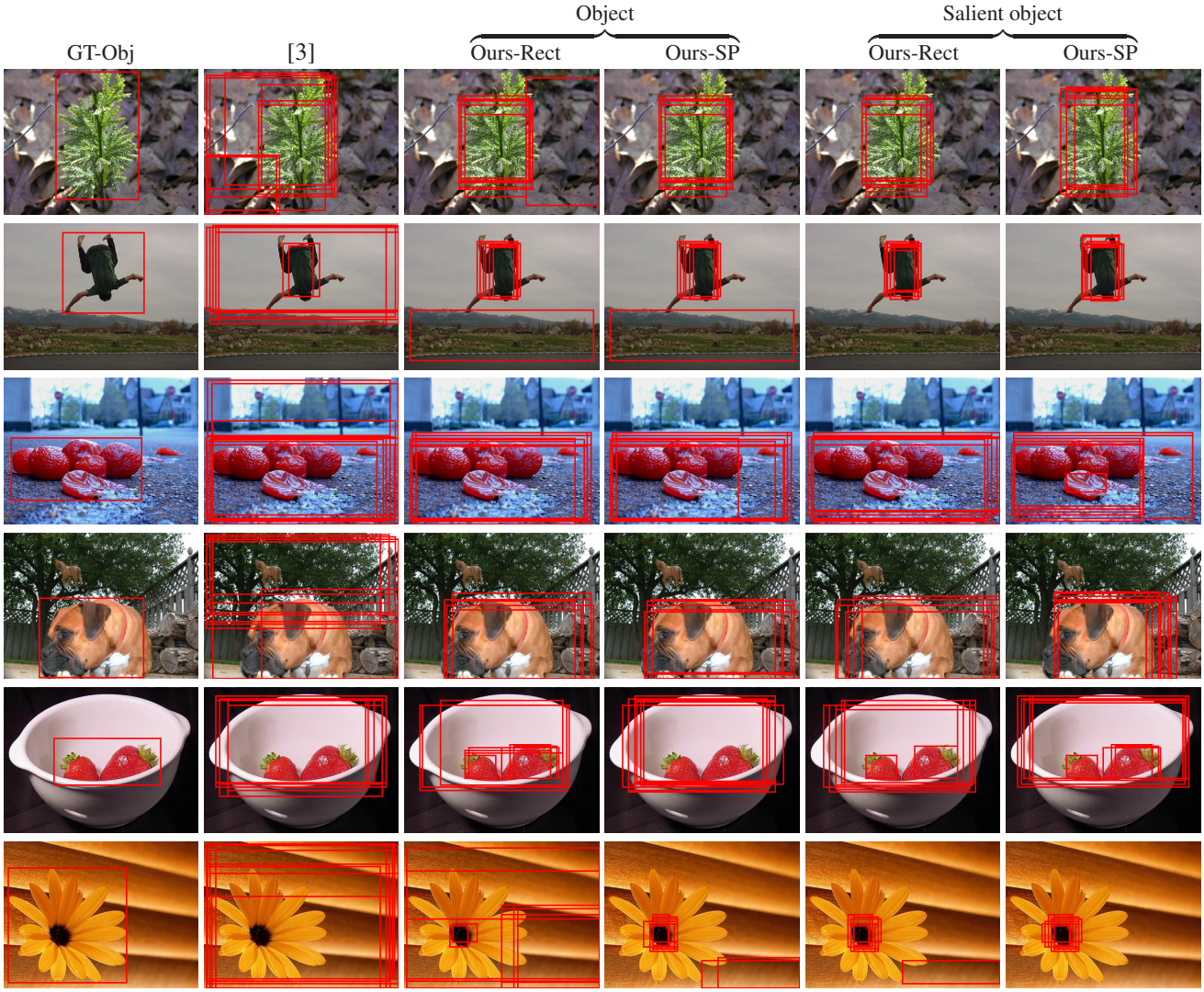


Figure 6. Object and salient object detection results (GT-Obj: ground-truth objectness). The last row shows a failed example.

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, November 1998.

[12] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2009.

[13] W. Kienzle, F. Wichmann, B. Schölkopf, and M. Franz. A nonparametric approach to bottom-up visual saliency. In *NIPS 19*, pages 689–696, 2007.

[14] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to detect a salient object. In *CVPR*, 2007.

[15] Y. Ma and H. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACMMM*, pages 374–381, 2003.

[16] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *ICCV*, 2009.

[17] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *PAMI*, 30(9):1632–1646, September 2008.

[18] A. Nuthmann and J. Henderson. Object-based attentional selection in scene viewing. *Journal of Vision*, 10(8), 2010.

[19] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä. Segmenting salient objects from images and videos. In *ECCV*, pages 366–379, 2010.

[20] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T. Chua. An eye fixation database for saliency detection in images. In *ECCV*, pages 30–43, 2010.

[21] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-2):61–81, April 2005.

[22] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, pages 537–544, 2009.

[23] A. Yarbus. *Eye movement and vision*. Plenum Press, New York, 1967.