Learning Complementary Saliency Priors for Foreground Object Segmentation in Complex Scenes

Yonghong Tian, Jia Li, Shui Yu & Tiejun Huang

International Journal of Computer Vision

ISSN 0920-5691
Volume 111
Number 2

Int J Comput Vis (2015) 111:153-170
DOI 10.1007/s11263-014-0737-1

Springer

Int J Comput Vis (2015) 111:153–170
DOI 10.1007/s11263-014-0737-1

Learning Complementary Saliency Priors for Foreground Object Segmentation in Complex Scenes

Yonghong Tian · Jia Li · Shui Yu · Tiejun Huang

Received: 31 March 2013 / Accepted: 31 May 2014 / Published online: 15 July 2014

Abstract Object segmentation is widely recognized as one of the most challenging problems in computer vision. One major problem of existing methods is that most of them are vulnerable to the cluttered background. Moreover, human intervention is often required to specify foreground/background priors, which restricts the usage of object segmentation in real-world scenario. To address these problems, we propose a novel approach to learn complementary saliency priors for foreground object segmentation in complex scenes. Different from existing saliency-based segmentation approaches, we propose to learn two complementary saliency maps that reveal the most reliable foreground and background regions. Given such priors, foreground object segmentation is formulated as a binary pixel labelling problem that can be efficiently solved using graph cuts. As such, the confident saliency priors can be utilized to extract the most salient objects and reduce the distraction of cluttered background. Extensive experiments show that our approach outperforms 16 state-of-the-art methods remarkably on three public image benchmarks.

Communicated by M. Hebert.

This work was supported in part by grants from the Chinese National Natural Science Foundation under contract No. 61035001, No. 61370113, and No. 61390515, and the Supervisor Award Funding for Excellent Doctoral Dissertation of Beijing (No. 20128000103).

Y. Tian (✉) · J. Li (✉) · T. Huang
National Engineering Laboratory for Video Technology, School of EE & CS, Peking University, Beijing 100871, China
e-mail: yhtian@pku.edu.cn

J. Li
e-mail: jia.li@pku.edu.cn

S. Yu
School of Information Technology, Deakin University, 221 Burwood HWY, Burwood 3125, VIC, Australia

Keywords Foreground object segmentation · Visual saliency · Complementary saliency map · Graph cuts

1 Introduction

Object segmentation is one of the fundamental problems in computer vision. It can be used in many applications such as visual object recognition, content-based image retrieval, and object-based video coding. However, such task is made difficult by the wide variability of the object's shape, appearance, and the complexity of the surrounding scene. Therefore, in spite of significant efforts, object segmentation still remains an open problem.

To solve this problem, a feasible solution is to incorporate priors about the object's shape, appearance and location in the segmentation process. Typically, unsupervised object segmentation methods such as (Liu et al. 2010; Borenstein and Ullman 2004; Winn and Jojic 2005; Rother et al. 2006) assume that object shape and color distribution patterns are consistent within each class and the variance of object shape, color and texture within a single object of a class is limited. These methods work well on simple scenes with unique objects but often fail to handle complex scenes with multiple foreground objects. As a consequence, some supervised approaches propose to train the segmentation model from pixel-level object masks. However, manually labelling such masks for a large dataset is very tedious in practice. In some other approaches, human intervention is used to bootstrap the segmentation process by specifying some kinds of priors (e.g., shape templates Zhao and Davis 2005, object part configuration Yu et al. 2002, connectivity Vicente et al. 2008, topology Lempitsky et al. 2009 or seed point/region Boykov and Jolly 2001; Li et al. 2005). Clearly, such human inter-

vention places restrictions on the wider applications of object segmentation on large datasets.

In recent studies, visual saliency, which serves as a selection mechanism to pop-out important contents, has been used to guide the automatic segmentation process. For example, Itti et al. (1998) combined multi-scale features to produce a saliency map and adopted a dynamical neural network to select attended areas that roughly contained the salient objects. Achanta et al. (2009) produced a frequency-tuned saliency map which was then binarized by a threshold to pop out salient regions. Hou and Zhang (2007) constructed a saliency map by analyzing the log-spectrum of an image and used a threshold to detect salient objects. Liu et al. (2007) learned a Conditional Random Field (CRF) to generate a saliency map for salient object detection. For these approaches, one fundamental assumption is that an accurate saliency map is sufficient for segmenting the whole foreground object. Unfortunately, saliency maps may become inaccurate when processing complex scenes with cluttered background. In such cases, it is often difficult to segment precise object boundaries since the priors derived from saliency maps are somehow ambiguous (Ma and Zhang 2003; Walther and Koch 2006; Liu et al. 2007; Gopalakrishnan et al. 2009).

Instead of assuming that saliency maps are always accurate, we propose to learn complementary saliency priors for foreground object segmentation. The overall framework of our approach is illustrated in Fig. 1. The whole process is divided into a learning component and a segmentation component. In the learning step, we propose to learn two mapping functions to generate two complementary saliency maps, including an *envelope map* and a *sketch map*. The envelope map always highlights a large area containing the objects while the sketch map prefers to highlight small areas inside each salient object. In the segmentation step, pixels with low envelope saliency can be regarded as background seeds while pixels with high sketch saliency can be treated as object seeds. As such, only the most confident parts of complementary saliency maps are utilized for object segmentation. This decreases the ambiguity of saliency priors in existing saliency-based segmentation methods. For the sake of simplicity, such priors extracted from complementary saliency maps are denoted as *complementary saliency priors*. Finally, foreground objects in each image are segmented using graph cuts with the learned complementary saliency priors.

To validate the effectiveness of the proposed approach, extensive experiments have been conducted on three public image benchmarks. Experimental results show that our proposed approach can adapt to various kinds of user labels (e.g., accurate object masks and bounding boxes) and gains remarkable improvements on several state-of-the-art segmentation methods (e.g., Achanta et al. 2009; Li et al. 2010a; Yu et al. 2010; Cheng et al. 2011; Jiang et al. 2011; Perazzi et al. 2012; Li et al. 2013). It can even obtain comparable results with the interactive method—Grabcut (Rother et al. 2004).

Our main contributions are summarized as follows:

1. The concept "complementary saliency priors" is first proposed for segmentation. With such priors, the most confident parts of saliency maps can be used to reduce the distraction of cluttered background, making the segmentation system work very well in complex scenes.
2. We propose a learning approach to generate complementary maps by combining various raw maps. The learning task is casted as a ML-based optimization problem which can be solved by the gradient-descent algorithm. Thus, it can be easily applied to any dataset, as long as the training samples are labeled with precise object masks or approximate bounding boxes.
3. Object segmentation is formulated as a binary pixel labelling problem and can be done using the graph cuts technique. In particular, we improve the traditional graph-cuts-based framework by incorporating complementary saliency priors. This results in a fully automatic

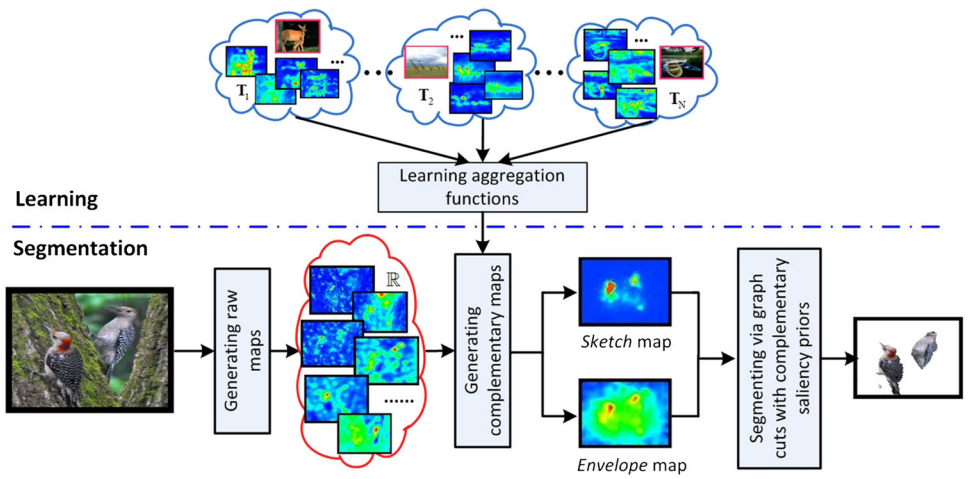

Fig. 1 System framework of our approach. Aggregation functions are first learned from training images in a supervised manner. The learned functions are then used to generate complementary saliency maps, which will be used in the graph cuts framework to assist the segmentation of foreground objects in testing images

solution for foreground object segmentation without the involvement of any human intervention.

The remainder of this paper is organized as follows: Sect. 2 briefly summarizes the related work. Section 3 formulates the concept of complementary saliency priors and Sect. 4 presents the learning algorithm. Section 5 shows object segmentation via graph cuts. Extensive experiments are presented in Sect. 6, and finally we conclude the work in Sect. 7.

2 Related Work

Roughly speaking, image segmentation can be categorized into several related but different tasks: over-segmentation, foreground object segmentation and semantic object segmentation. In this study, we mainly focus on the foreground object segmentation problem, which aims to detect and segment salient or foreground objects from an image (Zhao and Davis 2005; Liu et al. 2007). Without risk of confusion, we simply call it "object segmentation".

Depending on whether human intervention is involved, object segmentation can be divided into two categories: unsupervised and supervised. Unsupervised methods do not require training images or only require such images without manual annotation. Usually, methods in this category try to address both object class learning and object segmentation simultaneously (Liu et al. 2010; Borenstein and Ullman 2004; Winn and Jojic 2005), or co-segment a pair (set) of images (Rother et al. 2006). Moreover, the segmentation is often based on a specific assumption on the variance of object shape and color/texture (Liu et al. 2010). For example, Liu et al. (2010) and Winn and Jojic (2005) assume that the variability of color/shape in an object class is within a limited range, while Rother et al. (2006) suggests the common parts of all the segmenting images are required to have a similar shape and color/texture distribution. Due to a lack of priors about the object's shape and appearance, these unsupervised object segmentation methods are usually hard to have accurate segmentation, especially for images that contain multiple objects in a complex scene.

Comparatively, supervised methods seem to have accounted for the majority of work on object segmentation because more desirable results are obtainable. Besides manually labelling object masks in training images, shape templates (Zhao and Davis 2005) or other kinds of shape priors (e.g., object part configuration Yu et al. 2002, center-bias rectangular contour Hua et al. 2006) are also specified manually to bootstrap the segmentation process. For example, Zhao and Davis (2005) combined shape template matching with object segmentation in an iterative manner to enhance the performances on both sides. One limitation of shape-based approaches is that they can only handle specific kinds of objects given a certain set of training images. Other supervised methods require either the initial positions of objects (Li et al. 2005; Kass et al. 1988) or object/background models of visual features (e.g., color, texture) provided through human interaction (Rother et al. 2004). For example, Kass et al. (1988) introduced energy-minimizing splines (snakes) that were driven towards image features (e.g., lines, edges) to detect object contours, and Rother et al. (2004) exploited the graph cuts technique in an interactive manner for object segmentation. In practice, since an object segmentation system has to handle large quantities of images, neither labelling a number of training samples with various object classes nor manually specifying priors for bootstrapping the segmentation process are feasible.

Since visual saliency can well accord with human visual perception, and consequently serve as one type of selection mechanism for important content, it has recently been used for object segmentation in an unsupervised or supervised manner. Typically, saliency-based methods employ different visual models to compute saliency maps, and then analyze these maps to pop out salient objects. Following the pioneer work (Itti et al. 1998), Achanta et al. (2008) used low-level features (e.g., luminance, color) to determine salient regions, and their later work (Achanta et al. 2009) produced a frequency-tuned map which was then binarized by an adaptive threshold to pop out salient regions. Hou and Zhang (2007) constructed a saliency map by analyzing the log-spectrum of an image and adopted a simple threshold to detect salient objects. Ma and Zhang (2003) generated a contrast-based saliency map and extracted objects by fuzzy growing, and Cheng et al. (2011) also proposed an approach to segment images into regions and computed visual saliency using the regional contrasts. Perazzi et al. (2012) proposed to divide images into super-pixels and adopt saliency filters to detect salient objects. Similarly, Li et al. (2013) also presented an optimization framework to infer the superpixel-based saliency over each single image. In (Jiang et al. 2011), the context and shape priors were incorporated for detecting salient objects. An image was first segmented and analyzed over multiple scales and then an optimization framework was used to pop-out the salient objects. Instead, Markov random walks were performed on images by Gopalakrishnan et al. (2009) to detect salient regions. By introducing supervised learning techniques to model visual saliency, Liu et al. (2007) learned a Conditional Random Field (CRF) to combine a set of multi-scale features for salient object detection, while Mehrani and Veksler (2010) proposed an approach to refine the initial saliency-based segmentation by performing binary graph cuts optimization.

Strictly speaking, visual saliency cannot guarantee the accuracy of object segmentation. Typically, the computation of visual saliency depends on local difference and is vulnerable to noises. Moreover, saliency maps usually have *low reso-*

lution and poorly defined borders (Achanta et al. 2009), such that they may provide ambiguous priors for object segmentation. For example, some saliency-based methods are sensitive to local sudden changes in the background (e.g., Itti et al. 1998; Achanta et al. 2009; Hou and Zhang 2007; Achanta et al. 2008), and as a consequence distracters may be treated as salient objects. This causes each segmentation result to be an *envelope*-like area containing the objects. Meanwhile, there are also other methods (e.g., Ma and Zhang 2003) that prefer only to highlight some important parts of an object (referred to as *sketch*). Although objects segmented by the two kinds of saliency-based methods may independently suffer some problems such as the inexactness of their outlines or incompleteness of their internal bodies, it is possible to obtain desirable results by integrating them in a unified framework.

Towards this end, our previous work (Yu et al. 2010) utilized saliency maps from Achanta et al. (2009) and Liu et al. (2007) to generate sketch and envelope maps, then built two KD-trees as the object-background color model, and finally used a color signature based classifier to obtain the segmentation results. Although this method performs well on the dataset from Achanta et al. (2009), it does not work well in complex scenes since the ad hoc complementary maps cannot been well generalized to different datasets. Therefore, this study extends Yu et al. (2010) by learning to generate complementary saliency maps and utilizing the graph cuts technique with complementary saliency priors for pixel labelling. We use a supervised learning but interaction-free framework for both complementary saliency prior learning and binary pixel labelling. This makes the proposed approach more applicable for large segmentation tasks while achieving an improved performance.

It should be noted that our complementary saliency prior learning algorithm is different with previous computational models (e.g., Li et al. 2010b; Judd et al. 2009; Liu et al. 2011; Borji et al. 2012) that combine a set of feature maps or raw maps to generate one saliency map. In our algorithm, the "optimal" parameter matrices should be learned so as to generate two best complementary maps that not only best fit the ground-truth but also have statistically minimal complementary energy.

3 Complementary Saliency Priors

Intuitively, saliency maps can roughly highlight conspicuous targets, while pixels with high and low saliency values are likely to belong to an object (denoted by "Obj") and background (denoted by "Bkg"), respectively. Thus *saliency prior* is used here to signify the belief that a pixel belongs to "Obj" or "Bkg" in a saliency map. Consider an image I with object O and background B. Let $\mathcal{S}$ be one of its saliency map and $s_i = \mathcal{S}(i) \in [0, 1]$ be the saliency value at pixel $i \in I$. For each pixel i, there are two possible labels {"Obj", "Bkg"} or simply $\{1, 0\}$. Then the saliency prior for i can be expressed as:

$$\hbar_i = \begin{cases} 1 - \mathcal{S}(i), & for \ \ l_i = 0, \\ \mathcal{S}(i), & for \ \ l_i = 1, \end{cases} \tag{1}$$

where l_i is the label of pixel i. Without loss of generality, the prior can be expressed as a function of two variables $\hbar_i = f(\mathcal{S}, i)$.

However, the saliency prior in (1) may be far from being exact due to inaccurate saliency map. Therefore, we propose to extract saliency priors from two complementary saliency maps which consist of an envelope map and a sketch map. As such, pixels with low saliency in the envelope map can be regarded as background seeds while pixels with high saliency in the sketch map can be treated as object seeds. To quantitatively characterize envelope and sketch maps, we introduce two types of energies for a saliency map $\mathcal{S}$, namely *envelope energy* E_{env} and *sketch energy* E_{ske}:

$$E_{env}(\mathcal{S};\ G, I) = -\sum_{i \in O} \log s_i + \sum_{i \in B} s_i, \tag{2}$$

$$E_{ske}(\mathcal{S};\ G, I) = -\sum_{i \in B} \log \widetilde{s_i} + \sum_{i \in O} \widetilde{s_i}, \tag{3}$$

where $\widetilde{s_i} = 1 - s_i$, G denotes the ground-truth of image I[1]. The first term in (2) forces a saliency map to highlight the whole object part if small envelope energy is achieved. The second term in (2) is used to ensure that the saliency map with smaller envelope energy should contain as less background pixels as possible. Note that different forms of penalty are used in the two terms since a large penalty is expected for an object pixel with saliency close to 0, while a small penalty is assigned for a background pixel with saliency close to 1. Similarly, the first term in (3) encourages a saliency map to only highlight the objects if small sketch energy is sought. The second term in (3) plays a role in punishing the map for darkening too many object pixels.

Intuitively, the ground-truth should be the ideal envelope map and sketch map simultaneously. That is, *E_{env} and E_{ske} reach their minima if and only if the saliency map $\mathcal{S}$ equals to the ground-truth G*. In this sense, we can use E_{env} (or E_{ske}) to characterize the divergence between $\mathcal{S}$ and the ground-truth G when $\mathcal{S}$ acts as an envelope map (or a sketch map).

Let $\langle \mathcal{S}^+, \mathcal{S}^- \rangle$ be an ordered pair of saliency maps of image I. To give a quantitative measure that $\langle \mathcal{S}^+, \mathcal{S}^- \rangle$ is "complementary," we define the complementary energy E_{com} between $\mathcal{S}^+$ and $\mathcal{S}^-$ as:

[1] In our implementation, we add a very small positive number to the value in every log function to avoid yielding infinity and make problems have feasible solutions.

$$E_{com}(\langle \mathcal{S}^+, \mathcal{S}^- \rangle; G, I) = E_{env}(\mathcal{S}^+;\ G, I) + E_{ske}(\mathcal{S}^-;\ G, I). \quad (4)$$

Typically, small E_{com} implies both small E_{env} and E_{ske}. Here we call $\mathcal{S}^+$ the *envelope map* and $\mathcal{S}^-$ the *sketch map*. Note that exchanging the position of $\mathcal{S}^+$ and $\mathcal{S}^-$ in (4) leads to a different value of complementary energy. For simplicity, we use $E_{com}(\mathcal{S}^+, \mathcal{S}^-)$ instead of $E_{com}(\langle \mathcal{S}^+, \mathcal{S}^- \rangle; G, I)$ in the following discussion.

Moreover, the complementary energy has the following two properties:

1. $E_{com}(\langle \mathcal{S}^+, \mathcal{S}^- \rangle) \in [0, +\infty)$, with its minima if and only if $\mathcal{S}^+ = \mathcal{S}^- = G$;
2. For two maps $\mathcal{S}^{++}$ and $\mathcal{S}^{--}$ with $E_{com}(\mathcal{S}^+, \mathcal{S}^-) \geq E_{com}(\mathcal{S}^{++}, \mathcal{S}^-)$ and $E_{com}(\mathcal{S}^+, \mathcal{S}^-) \geq E_{com}(\mathcal{S}^+, \mathcal{S}^{--})$, we have $E_{com}(\mathcal{S}^+, \mathcal{S}^-) \geq E_{com}(\mathcal{S}^{++}, \mathcal{S}^{--})$.

Following these properties, it is feasible to iteratively search an optimal or near-optimal pair of saliency maps with as lower complementary energy as possible. An illustration of the complementary energy is shown in Fig. 2. For a "good" envelope map $\mathcal{S}^{+*}$, it is important to highlight all the pixels in O with as less pixels in B as possible. The energy will increase greatly if an envelope map misses some pixels in O. While for a "good" sketch map $\mathcal{S}^{-*}$, it is important to highlight only the pixels in O. Any additional high-saliency pixels in B will be largely punished. Therefore, the following two facts can be intuitively inferred:

1. The dark pixels in any "good" envelope map $\mathcal{S}^{+*}$ (i.e., pixels with the lowest envelope values) most likely belong to B.
2. The bright pixels in any "good" sketch map $\mathcal{S}^{-*}$ (i.e., pixels with the highest sketch values) have high probabilities they will be part of O.

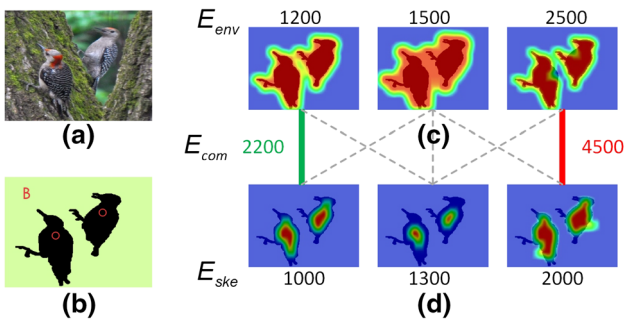

Fig. 2 Examples of complementary energy. **a** Original image, **b** ground-truth, **c** three envelope maps, and **d** three sketch maps. The map pair marked with *green* (*red*) *line* has the smallest (largest) complementary energy, leading to the most (least) confident complementary saliency priors

Here we use $\bar{E}$ to denote the regions with low envelope values, T to denote the regions with high sketch values, and X to denote the regions with high envelope values and low sketch values simultaneously. Clearly, $\bar{E} \cup X \cup T = I$ and $\bar{E} \cap X = \bar{E} \cap T = T \cap X = \emptyset$. Placing the above two facts into (1), we can derive the complementary saliency prior for pixel i:

$$\hbar_i = \begin{cases} f(\mathcal{S}^{-*},\ i), & for\ i \in T, \\ f((\mathcal{S}^{+*} + \mathcal{S}^{-*})/2,\ i), & for\ i \in X, \\ f(\mathcal{S}^{+*},\ i), & for\ i \in \bar{E}. \end{cases} \quad (5)$$

Here we conduct an experiment to validate whether smaller complementary energy contributes to better segmentation performance. Given an image, four saliency maps are generated by Achanta et al. (2009), Hou and Zhang (2007), Goferman et al. (2010), and Seo and Milanfar (2009) to produce a pair of complementary saliency maps using our learning algorithm described in the next section. The four maps are combined into five pairs, with some maps repeatedly used. For each pair, we assume one map as the envelope map and the other as the sketch map. Finally complement energies of the five pairs are computed and their corresponding segmentation results are also evaluated. From Fig. 3, we can see that with increasing complementary energy, a pair has worse "complementary" property in terms of capturing the salient object (i.e., one is less like the envelope map while the other is less like sketch map), and thus, the segmentation result is more inaccurate.

4 Learning Complementary Saliency Priors

In this section, we will explore a feasible learning approach to generate complementary saliency maps by combining various raw maps generated with several predefined visual saliency models. Some main notations in this section are as follows:

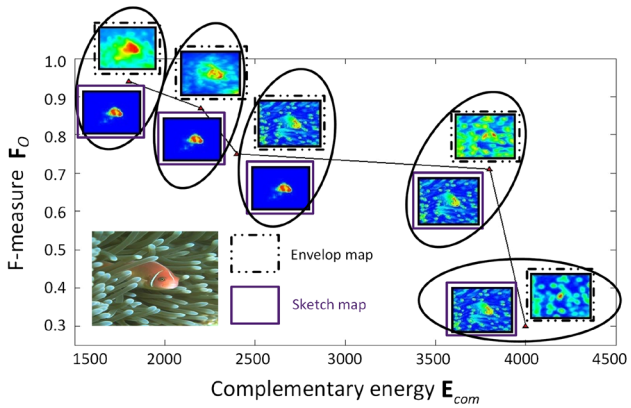

Fig. 3 The relationship between complementary energy and segmentation performance

- Let $\mathbb{T} = \{T^{(n)} = (\mathbb{R}^{(n)}, G^{(n)}, I^{(n)})\}_{1\le n\le N}$ be the training set, where $\mathbb{R}^{(n)} = \{R^{(n,k)}\}_{1\le k\le K}$ is the set of K raw maps for the nth image $I^{(n)}$, with its ground-truth $G^{(n)}$ ($g_i = 1$ if pixel $i \in I^{(n)}$ belongs to object, $g_i = 0$ if it is a background pixel).
- Let $\langle \mathcal{S}^{(n)+}, \mathcal{S}^{(n)-}\rangle$ be an ordered pair of saliency maps of image $I^{(n)}$ where $\mathcal{S}^{(n)+}$ is the envelope map while $\mathcal{S}^{(n)-}$ is the sketch map.
- Let $\boldsymbol{\phi} = \{\phi^+, \phi^-\}$ be two mapping functions with parameters $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}^+, \boldsymbol{\Theta}^-\}$ for combining raw maps to generate $\mathcal{S}^{(n)+}$ and $\mathcal{S}^{(n)-}$, i.e., $\mathcal{S}^{(n)+} = \phi^+(\mathbb{R}^{(n)})$ and $\mathcal{S}^{(n)-} = \phi^-(\mathbb{R}^{(n)})$, where $\boldsymbol{\Theta}^+ = [\theta_i^{(k)+}]_{Q\times K}$ and $\boldsymbol{\Theta}^- = [\theta_i^{(k)-}]_{Q\times K}$ are two $Q \times K$ parameter matrices, and Q is the number of blocks in an image [2]. Let $\boldsymbol{\theta}_i^+ = [\theta_i^{(k)+}]_{1\times K}$ and $\boldsymbol{\theta}_i^- = [\theta_i^{(k)-}]_{1\times K}$ be two row vectors respectively for $\boldsymbol{\Theta}^+$ and $\boldsymbol{\Theta}^-$, then $\|\boldsymbol{\theta}_i^+\|_1 = \|\boldsymbol{\theta}_i^-\|_1 = 1$.

4.1 Problem Formulation

Here our task is to learn two functions $\boldsymbol{\phi} = \{\phi^+, \phi^-\}$ from $\mathbb{T}$. This learning problem can be expressed as:

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} L(\mathbb{T}, \boldsymbol{\phi}), \tag{6}$$

where the objective function is defined as:

$$L(\mathbb{T}, \boldsymbol{\phi}) = \log P(\boldsymbol{\phi} \mid \mathbb{T}) \propto \log P(\mathbb{T} \mid \boldsymbol{\phi}), \tag{7}$$

where $P(\boldsymbol{\phi} \mid \mathbb{T})$ is the conditional probability for inferring $\boldsymbol{\phi}$, $P(\mathbb{T}|\boldsymbol{\phi})$ is the likelihood that can be interpreted as the joint-probability of different training samples given $\boldsymbol{\phi}$. Suppose $\mathbb{T} = \{T^{(n)}\}$ to be i.i.d, we have:

$$L(\mathbb{T}, \boldsymbol{\phi}) \propto \sum_{T^{(n)}\in\mathbb{T}} \log P(T^{(n)} \mid \boldsymbol{\phi}). \tag{8}$$

Then (6) turns to a typical Maximum Likelihood (ML) estimation problem.

Inspired by the Markov-Gibbs Equivalence, we model $P(T^{(n)} \mid \boldsymbol{\phi})$ using Gibbs distribution:

$$P(T^{(n)} \mid \boldsymbol{\phi}) = \frac{1}{Z(\boldsymbol{\phi}, n)} \exp\left(-E(T^{(n)}, \boldsymbol{\phi})\right), \tag{9}$$

where $E(T^{(n)}, \boldsymbol{\phi})$ is some kind of energy determined by $T^{(n)}$ and $\boldsymbol{\phi}$, and $Z(\boldsymbol{\phi}, n) = \sum_{G\in\mathcal{G}} E(T^{(n)}, \boldsymbol{\phi})$ is the partition function which ensures (9) is a probability.

Let $E(T^{(n)}, \boldsymbol{\phi}) = E_{com}(\mathcal{S}^{(n)+}, \mathcal{S}^{(n)-})$ where $\mathcal{S}^{(n)+} = \phi^+(\mathbb{R}^{(n)})$ and $\mathcal{S}^{(n)-} = \phi^-(\mathbb{R}^{(n)})$, then the learning problem can be expressed as finding an "optimal" $\boldsymbol{\phi}^*$ from $\mathbb{T}$, so that for any training image in $\mathbb{T}$, we can utilize $\boldsymbol{\phi}^*$ to generate two complementary maps with **statistically** minimal complementary energy. Following this, we first rewrite (4) as follows:

$$\begin{aligned} &E_{com}(\langle \mathcal{S}^{(n)+}, \mathcal{S}^{(n)-}\rangle; G^{(n)}, I^{(n)}) = E(T^{(n)}, \boldsymbol{\phi}) \\ &= \sum_{i\in I^{(n)}} \underbrace{\left(\widetilde{g_i}\cdot s_i^+ - g_i \cdot \log s_i^+ + g_i \cdot \widetilde{s_i}^- - \widetilde{g_i}\cdot \log \widetilde{s_i}^-\right)}_{U(g_i, i, \boldsymbol{\phi})}, \end{aligned} \tag{10}$$

where $\widetilde{g_i} = 1 - g_i$, $\widetilde{s_i}^- = 1 - s_i^-$. It can then be found that (9) is fortunately divisible:

$$P(T^{(n)} \mid \boldsymbol{\phi}) = \prod_{i\in I^{(n)}} \frac{e^{-U(g_i,i,\boldsymbol{\phi})}}{\sum_{g_i'\in\{0,1\}} e^{-U(g_i',i,\boldsymbol{\phi})}}. \tag{11}$$

Replacing $P(T^{(n)} \mid \boldsymbol{\phi})$ with (11), the objective function in (8) can be rewritten as:

$$L(\mathbb{T}, \boldsymbol{\phi}) = \sum_{T^{(n)}\in\mathbb{T}} \sum_{i\in I^{(n)}} \left(-U(g_i, i, \boldsymbol{\phi}) - \log\Big(\sum_{g_i'\in\{0,1\}} e^{-U(g_i',i,\boldsymbol{\phi})}\Big)\right). \tag{12}$$

To generate complementary saliency maps, another problem is to define mapping functions ϕ^+ and ϕ^-. Typically, raw maps generated by different methods characterize the saliency of a visual scene from various aspects. Moreover, each raw map has a certain degree of confidence about the belief of its block being salient in the envelope map $\mathcal{S}^+$ (or sketch map $\mathcal{S}^-$). In our work, such a degree of confidence is modeled as a weight and each raw map is supposed to assign different weights to different locations. Recall that *envelope map* tends to highlight all possible salient locations and *sketch map* emphasizes only the most probable salient locations, for any block $i \in I^{(n)}$, ϕ^+ and ϕ^- can be modeled as:

$$\begin{aligned} \phi^+(\mathbb{R}^{(n)}, i) &= \sum_k r_i^{(n,k)} \theta_i^{(k)+}, \\ \phi^-(\mathbb{R}^{(n)}, i) &= \prod_k \left(r_i^{(n,k)}\right)^{\theta_i^{(k)-}}, \quad \forall i \in I^{(n)}, \end{aligned} \tag{13}$$

where $r_i^{(n,k)}$ is the average saliency value at location i on $R^{(n,k)}$ (the kth map in set $\mathbb{R}^{(n)}$). Finally, bringing (10), (12) and (13) together, the optimization problem can be written as:

$$\begin{aligned} &\max_{\boldsymbol{\theta}_i^+, \boldsymbol{\theta}_i^-} \sum_{T^{(n)}\in\mathbb{T}} \sum_{i\in I^{(n)}} \left(-U(g_i^{(n)}, i, \boldsymbol{\theta}_i^+, \boldsymbol{\theta}_i^-) - \log\Big(\sum_{g_i'\in\{0,1\}} e^{-U(g_i',i,\boldsymbol{\theta}_i^+,\boldsymbol{\theta}_i^-)}\Big)\right) \\ &s.t.\ \|\boldsymbol{\theta}_i^+\|_1 = 1,\ \theta_i^{(k)+} \ge 0, \\ &\|\boldsymbol{\theta}_i^-\|_1 = 1,\ \theta_i^{(k)-} \ge 0, \\ &\forall k = 1, 2, \ldots, K,\ \ i \in I^{(n)}. \end{aligned} \tag{14}$$

[2] As in many previous works, we divide images into macro-blocks and all pixels in a block are assumed to share the same parameter. In our experiments, each block covers 4×4 pixels for an image resized to the resolution 320×240.

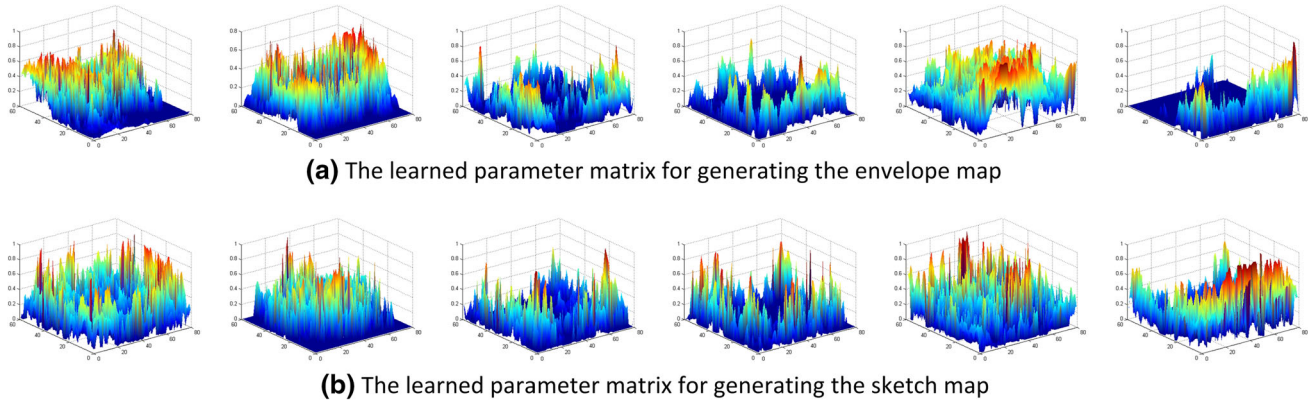

Fig. 4 Visualization of the learned parameter matrices on **MOCB**. Here each subfigure depicts weights for a raw map

Note that to avoid overfitting, the convex combination constraints are used here such that all weights for each block are non-negative and sum up to 1.

4.2 The Learning Algorithm

To solve (14), we first convert it to an unconstrained optimization problem:

$$\min_{\boldsymbol{\Theta}} F(\boldsymbol{\Theta}) = -L(\mathbb{T}, \boldsymbol{\phi}) + \sum_{i \in I^{(n)}} \left(\epsilon \cdot \sum_{k} \frac{1}{\theta_i^{(k)\pm}} + \frac{1}{\epsilon} \cdot (\|\boldsymbol{\theta}_i^{\pm}\|_1 - 1)^2 \right), \tag{15}$$

where ϵ is a pre-defined small positive number and $\theta_{\pm}$ denotes both $\theta_+$and θ_-. When all the constraints in (14) are satisfied, it is easy to see that this problem is equivalent to (14); while if any constraint is broken, $F(\boldsymbol{\Theta})$ will have a large value. To find the minima of (15), the gradient-descent algorithm (Boyd and Vandenberghe 2004) is used in our work:

$$\nabla F(\boldsymbol{\Theta}) = \left(\frac{\partial F(\boldsymbol{\Theta})}{\partial \theta_i^{(k)\pm}} \right)_{i,k}. \tag{16}$$

Since $F(\boldsymbol{\Theta})$ is a non-convex function as for the variable $\theta_i^{(k)\pm}$, only local minima of (15) can be found. In practice, we randomly initialize the parameters for a fixed number of times and find a solution for each time. We then pick the solution that yields the minimum value of (15). At each step, gradient descent operations are performed in turn for $\boldsymbol{\theta}_i^+$ and $\boldsymbol{\theta}_i^-$.

The computation complexity of the algorithm involves four variables: N (the size of training data), Q (the block volume of an image), K (the number of raw maps) and M (the number of descent steps). For each descent step, it will take $O(NQK)$ to compute the descent or search direction. It can be assumed the search of each step length will be completed in a short constant time. Then the total time complexity is $(O(NQK) + O(1)) \times M$. Given a certain tolerance error for algorithm termination, it is proven that there is a constant upper bound of M (Boyd and Vandenberghe 2004). Therefore, the solution can be found in $O(NQK)$ time.

4.3 Discussions

In our algorithm, different blocks in each raw map are assigned to different weights such that two $Q \times K$ parameter matrices $\boldsymbol{\Theta}^+$ and $\boldsymbol{\Theta}^-$ need to be learned. So one problem is why not to learn *one weight per raw map*, instead of *one weight per block and raw map*. As mentioned above, each weight reflects a certain degree of confidence of each raw map about the belief of its block being salient in $\mathcal{S}^+$ (or $\mathcal{S}^-$). Following this, the training process is in fact a joint fitting to the ground-truth. Obviously, a single weight can hardly fit well simultaneously for thousands of blocks in a raw map. As shown in Fig. 4, the Q elements for each raw map are significantly diversified from each other. So it is impossible to use only one weight to give a good approximation for all Q elements (we will discuss this issue with experiments in Sec. 6). Further with an in-depth analysis, we find that one possible reason might be due to the center-bias prior that is statistically significant in most image datasets (as shown in Fig. 5). In this situation, some raw saliency maps are more reliable in the center of the image, others in the image boundary (e.g., as shown in Fig. 4, the 5th raw map totally shows a larger weight in the image center than the other maps), leading to location-related weighting for different raw maps. In

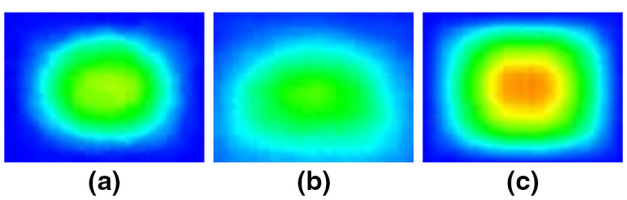

Fig. 5 Location priors for **a SOCB**, **b MOCB** and **c MSRA** datasets, each of which is constructed by counting object pixel occurrences in the corresponding training data (Gould et al. 2007)

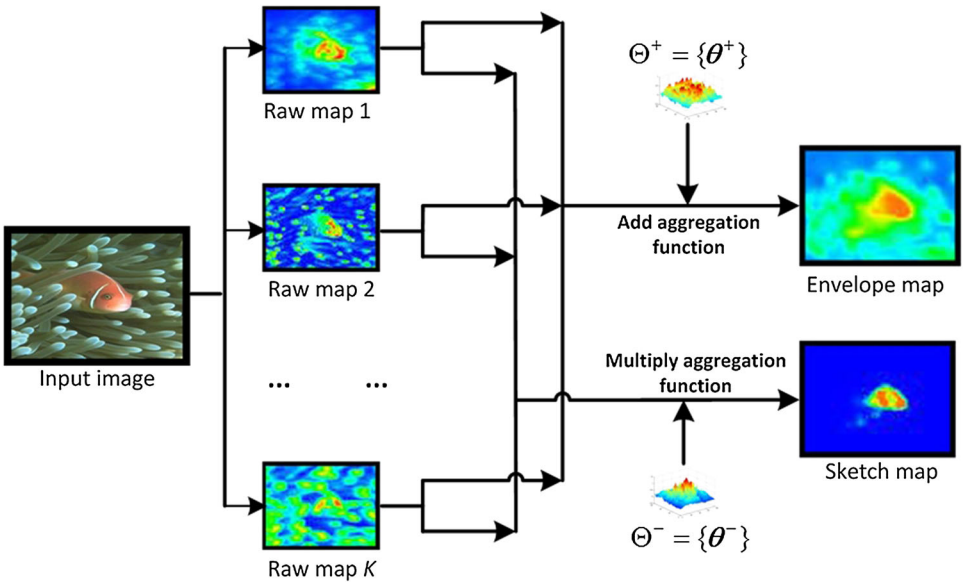

Fig. 6 An example of combining raw maps to generate complementary maps given the learned parameter matrices

the future work, we will explore a content-related weighting mechanism in the learning algorithm.

A related question is about the potential overfitting risk of our method. From (14), we can see that the parameter learning problem can be solved for each block independently, under the constraints of convex combination. So given a set of N training images, there are only $2 \times K$ parameters to fit for each block. Usually, $N \gg 2 \times K$. This reduces the overfitting risk and makes the learning process computationally efficient. It is shown in Sec. 6 that our learning algorithm can converge in less than 50 iterations on average.

Given the learned parameter matrices, the remaining problem is how to predict complementary saliency priors for a new image. As illustrated by Fig. 6, this process consists of three steps. Firstly, a given set of raw maps are produced for the input image. These maps are then combined to generate two complementary saliency maps using the mapping functions in (13). Finally, the complementary saliency priors for each pixel in the image are predicted using (5).

5 Segmentation Via Graph Cuts with Complementary Saliency Priors

Given the complementary saliency priors, we then segment foreground objects by using graph cuts. In our approach, object segmentation is modeled as a binary pixel labelling problem.

When labelling a pixel p, three factors should be considered. The first one is *prior* $\hbar_p$, indicating the label that p possibly belongs to. The second one is *visual appearance models* $\mathcal{M}$ that tell the feature distributions of objects and background. The last one is *interaction between the neighborhood* $\mathcal{N}_p$, implying the influence of pixel labels from p's neighborhood. Therefore, object segmentation can be formulated as finding a labeler $\Psi\left(p|\hbar_p, \mathcal{M}, \mathcal{N}_p\right) = l_p$, where $l_p \in \{0, 1\}$. Here we consider all these factors in a graph cuts framework (Boykov and Jolly 2001). That is, we seek a labelling assignment l that globally minimizes the following cost function:

$$C(l) = \sum_{p \in I^{(n)}} [D(l_p) + \sum_{q \in \mathcal{N}_p} V_{p,q}(l_p, l_q)], \tag{17}$$

where the first term considers the known appearance models for background and objects, and the second boundary term guarantees that two neighboring pixels are likely to have the same label. According to Kolmogorov and Zabih (2004), only *regular* costs can be used as binary costs, which must satisfy the inequality: $V_{p,q}(0, 0) + V_{p,q}(1, 1) \leq V_{p,q}(1, 0) + V_{p,q}(0, 1)$. As such, all cost functions have their global solutions.

In this study, we improve the graph cuts framework by incorporating complementary saliency priors. First, a threshold $\delta_\perp$ is used to cut the envelope map and sketch map to obtain envelope regions (denoted by E) and sketch regions (denoted by T), respectively[3]. As such, the most confident parts of complementary saliency maps (i.e., T and $\bar{E}$, where $\bar{E} = I - E$) can be utilized to train appearance models for objects and background, while less confident parts in E (i.e., $X = E - T$) are excluded from the training of appearance models. Once we have the models, the appearance likelihood $L(\mathbf{x}_p|l_p)$ can be calculated using the feature vector $\mathbf{x}_p$. Therefore, the cost $D(l_p)$ is defined as:

$$D(l_p) = -\log\left[\hbar(l_p)L(\mathbf{x}_p|l_p)\right], \tag{18}$$

[3] In our implementation, we use $\delta_\perp * avg(\mathcal{S}^+)$ and $\delta_\perp * avg(\mathcal{S}^-)$ to perform the binarization.

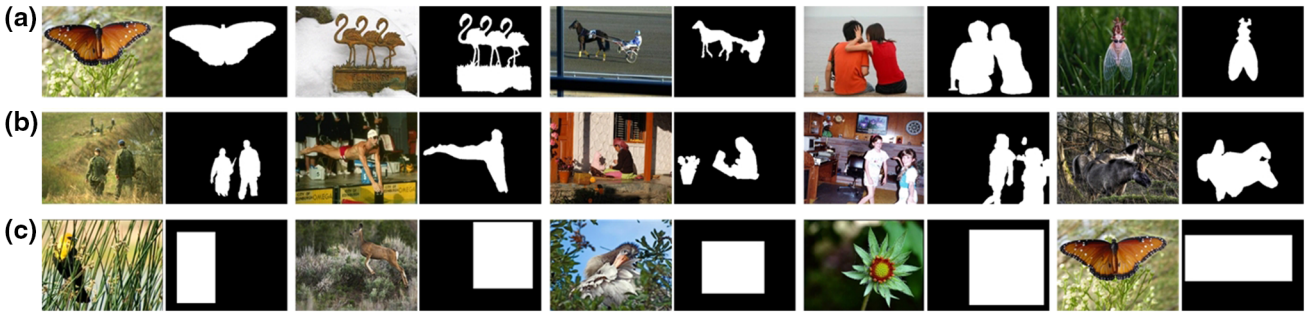

Fig. 7 Some examples from the three benchmarks. **a SOSB**, **b MOCB**, and **c MSRA**

where $\hbar(l_p)$ (calculated by (5)) is the saliency prior of pixel p if it is assigned to label l_p. For visual appearance modeling, we also use GMMs to characterize the color feature distribution of objects and background. Intuitively, such cost definition reflects the labelling belief indicated by complementary saliency priors. Specifically, for any pixel which has a low value in the envelope map, if it is assigned with label 1, a large cost will occur; for any pixel with a high saliency in the sketch map, a large cost will yield if it is labelled 0; if a pixel has a low sketch value and a high envelope value, its label is then determined by the appearance similarity. In this way, we can reduce the ambiguity in a single saliency map.

Finally, the cost $V_{p,q}(l_p, l_q)$ is defined as:

$$V_{p,q}(l_p, l_q) = \gamma [\![l_p \neq l_q]\!] \exp(-\beta ||\mathbf{x}_p - \mathbf{x}_q||^2), \tag{19}$$

where $[\![\varphi]\!]$ denotes the indicator function taking values 0 and 1. γ is a constant to adjust the weight of penalty terms. β is a constant to ensure that the exponential term switches appropriately between high and low contrast. As such, a fast max-flow algorithm (Boykov and Jolly 2001; Boykov and Kolmogorov 2004) can be used to find the global solution of (17).

6 Experiments

In this section, we conduct several experiments to show the effectiveness of our approach. Toward this end, three benchmarks are adopted in the experiments, including:

1. **SOSB**. This benchmark consists of 1,000 images labelled with exact object masks (Achanta et al. 2009). As shown in Fig. 7a, only one salient object is labelled in each image. Thus this benchmark can be used as the "baseline" to evaluate the effectiveness of various object segmentation methods.
2. **MOCB**. This benchmark contains 1,474 images labelled with precise object masks, in which 300 images are selected from Movahedi and Elder (2010) while the others are from PASCAL VOC09 (Everingham et al. 2009). As shown in Fig. 7b, multiple objects co-exist in a cluttered background in most of these images. This benchmark can be used for evaluating the robustness of object segmentation methods in complex scenes.
3. **MSRA**. This benchmark contains 5,000 images from Liu et al. (2007), while the salient objects in each image are manually labelled by several bounding boxes (as shown in Fig. 7c). This benchmark can be used to test whether our learning algorithm still works even when the training samples are approximately labelled with bounding boxes.

In the experiments, each benchmark is randomly partitioned into ten equal subsets, while four of them are used for training, one for validation and the other five subsets are used for for testing. On these benchmarks, our approach is compared with 16 state-of-the-art approaches, which can be roughly categorized into three groups, including:

▹ **Location-based Group:** This group contains 8 approaches that only output saliency maps, while each location (e.g., macro block) on a saliency map is assigned a real saliency value. Approaches in this group include Itti98 (Itti et al. 1998), Gould07 (Gould et al. 2007), Harel07 (Harel et al. 2007), Hou07 (Hou and Zhang 2007, Liu07 (Liu et al. 2007), Seo09 (Seo and Milanfar 2009), Goferman10 (Goferman et al. 2010) and Li10 (Li et al. 2010b). In the comparison, we use two thresholds[4] on the saliency maps generated by these approaches to obtain the highly reliable foreground and background pixels. The same graph cuts framework as in our approach is used to segment salient objects from these saliency maps to demonstrate the performance of the learned complementary saliency maps.

▹ **Object-based Group:** This group contains 7 approaches that output the foreground objects, including Achanta09 (Achanta et al. 2009), Carreira10 (Li et al. 2010a), Yu10 (Yu et al. 2010), Cheng11 (Cheng et al.

[4] The two thresholds are $\delta_s * avg(\mathcal{S})$ and $\frac{1}{\delta_s} * avg(\mathcal{S})$, while $\delta_s \in (0, 1]$ is learned via experiments on the validation set, in a similar way to $\delta_\perp$ in our approach.

2011), Jiang11 (Jiang et al. 2011), Perazzi12 (Perazzi et al. 2012), and Li13 (Li et al. 2013). Note that Carreira10 is not based on visual saliency, while Achanta09 and Cheng11 output both the location-based saliency maps and salient objects for each image. In our experiments, the parameters of these approaches are manually fine-tuned to reach the best performance on every benchmark.

▹ **Interactive graph cuts**: Grabcut (Rother et al. 2004) employs the graph cuts framework for object segmentation by manually drawing a rectangle as the segmentation prior. As a result, precise "Obj-Bkg" models can be derived for segmentation.

In the experiments, these approaches will be compared with the proposed approach (denoted as **OUR**). In the comparison, we adopt precision ($\mathbf{P}_O$), recall ($\mathbf{R}_O$), and F-measure ($\mathbf{F}_O$) to evaluate the similarity between the segmentation results and the ground-truth. Precision is *the ratio of correctly segmented regions to all the segmented regions*, while recall is *the ratio of correctly segmented regions to ground-truth regions*. Then the F-measure is defined as the weighted mean of precision and recall:

$$\mathbf{F}_O = \frac{(1+\alpha)\mathbf{P}_O \times \mathbf{R}_O}{\alpha \times \mathbf{P}_O + \mathbf{R}_O}. \tag{20}$$

Here we set α=1.0 to equally address both precision and recall. Moreover, we also use overlap ($\mathbf{OLP}_O$) (Everingham et al. 2009) as an additional metric for possible comparison with related works. Here, overlap plays a similar role with F-measure in penalizing both under-segmentations and over-segmentations. In addition, we compute the improvement (on $\mathbf{F}_O$) of the proposed approach against all the other approaches, denoted as **IMP**.

6.1 Parameter Selection

In practice, the selection of raw saliency maps is the first concern of our approach. In this work, we generate eight raw maps for each image using existing visual saliency estimation methods (i.e., Itti et al. 1998; Achanta et al. 2009; Hou and Zhang 2007; Walther and Koch 2006; Goferman et al. 2010; Harel et al. 2007; Seo and Milanfar 2009; Zhang et al. 2008) since they employ different strategies to capture salient properties of a visual scene. In this experiment, Achanta09 and Harel07 were empirically used as the first two raw maps. Then we increasingly added a saliency map into the raw map set and evaluate the segmentation performance. In each step, only the saliency map with the highest $\mathbf{F}_O$ would be kept in the raw map set. This process was terminated until there was no performance improvement by adding all the remaining saliency maps in turn. As shown in Fig. 8a, the optimal K values are different for the three benchmarks: 6 for **MOCB** (i.e., Harel07, Achanta09, Hou07, Goferman10, Seo09, Itti98), 5 for **SOSB** (i.e., Harel07, Achanta09, Hou07, Goferman10, Seo09), and 4 for **MSRA** (i.e., Harel07, Achanta09, Hou07, Seo09). So the three sets of raw maps will be used in all the other experiments.

In our approach, there are three parameters in the graph cuts component, including the saliency threshold $\delta_\perp$, the term weight γ and the smoothness controller β in (19). As in Rother et al. (2004), the parameter β can be set to an adaptive value $1/(2 * aver\_dis(i))$, where $aver\_dis(i)$ is the average of the squared distances between all pairs of neighboring pixels. The parameters $\delta_\perp$ and γ are then determined via experiments on the validation set. Since the two parameters are independent of each other, an iterated parameter selection process can be used. As shown in Fig. 8b, the optimal values of $\delta_\perp$ are 0.4 for **SOSB**, 1.7 for **MOCB**, 0.9 for **MSRA**, respectively. For parameter γ, Fig. 8c shows that the F-measure maxima appears at $\gamma = 110$ for **SOSB**, $\gamma = 35$ for **MOCB** and $\gamma = 30$ for **MSRA**. Thus we set γ to 110, 35 and 30 for the three benchmarks. It should be noted that the graph cut parameters for all baseline methods are also tuned in a similar way.

Furthermore, we conduct an experiment to verify whether the learning is over-fitting or not. This experiment is carried out by varying the number of training samples. For each benchmark, we randomly divide the training set into 10 folds. In the experiment, the parameters are learned by incrementally increasing the number of training samples, and the learned parameters are then used for testing on the separate testing set. Furthermore, we also add several special cases with a very small training set (with 10, 20, 30/40 samples) to evaluate the robustness of the proposed learning algorithm. As pointed out in Sec. 3, a "good" pair of complementary saliency maps should have as low complementary energy as possible. Thus the complementary energy (denoted as $\mathbf{E}_{com}$) can be directly used to evaluate the performance of our learning algorithm given different numbers of training samples. Figure 9 shows the experimental results on the three datasets. Note that each result is obtained by averaging results of three independent runs.

From Fig. 9a–c, we can see that reducing the training data to a limited number of training samples (30 for **SOSB**, 63 for **MOCB** and 50 for **MSRA**, all less than 10% of the corresponding training dataset or 5% of the corresponding whole dataset) will not *severely* decrease the performance of our learning algorithm. When given a certain number of training samples (120 for **SOSB**, 315 for **MOCB** and 100 for **MSRA**, all less than 50% of the corresponding training dataset or 25% of the corresponding whole dataset), more training samples don't yield *remarkably different* results.

Figure 9d also shows the average training time of our learning algorithm on a 4-core PC with 3.1GHz CPU and 3G RAM. We can see that the training time costs are almost linearly correlated with the numbers of training samples.

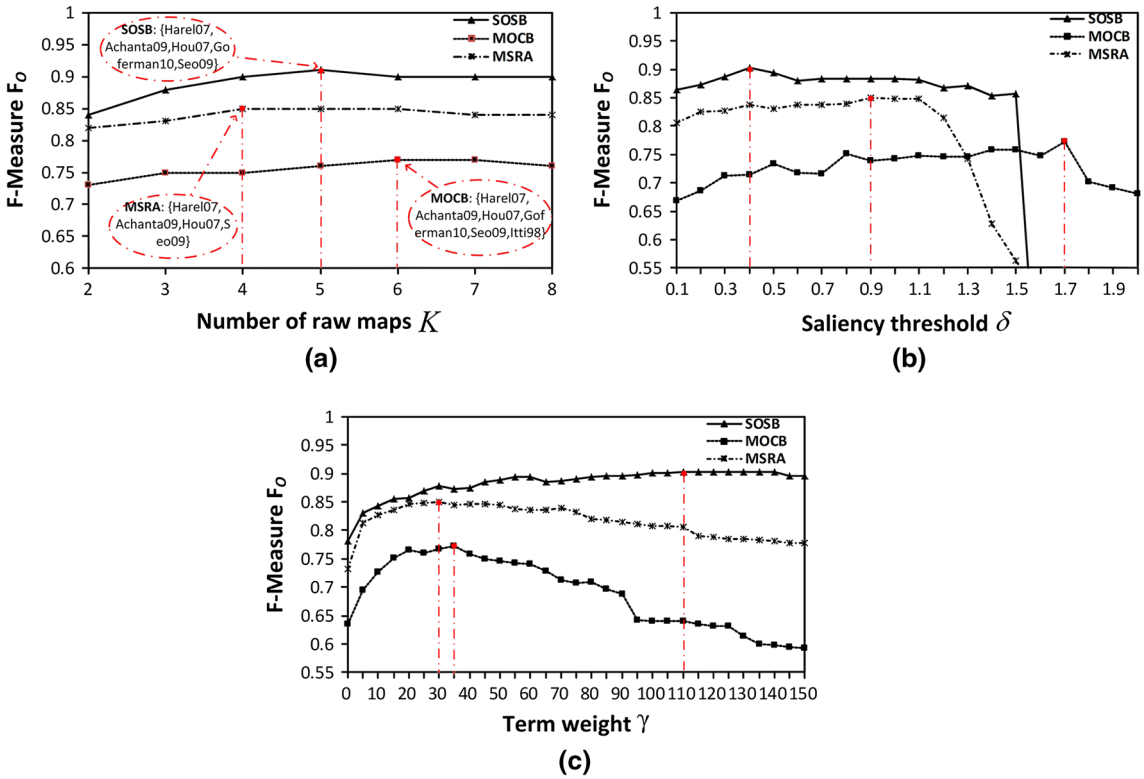

Fig. 8 Parameter selection for K, $\delta_\perp$ and γ through experiments on the validation set. **a** $\mathbf{F}_O$ by increasingly adding raw maps for different datasets; **b** $\mathbf{F}_O$ under different $\delta_\perp$ with $\gamma = 110$ for **SOSB**, $\gamma = 35$ for **MOCB**, $\gamma = 30$ for **MSRA**; **c** $\mathbf{F}_O$ under different γ with $\delta_\perp = 0.4$ for **SOSB**, $\delta_\perp = 1.7$ for **MOCB**, $\delta_\perp = 0.9$ for **MSRA**

Meanwhile, the average iterations are 20.5 for **SOSB**, 44.2 for **MOCB**, and 30.5 for **MSRA**, respectively. That is, our learning algorithm will converge in less than 50 iterations on average. All the experimental results show that our learning algorithm is computationally efficient and has a low risk of overfitting when given a certain number of training samples.

It should be noted that the average training time of our learning algorithm on **MOCB** is nearly three times more than that on **SOSB**. If without considering the training time costs, we can also use the same parameter values for both the **SOSB** and **MOCB** datasets. For example, the experimental result show that the F-score of our approach on **SOSB** can also reach 0.89 when directly using the same parameter values trained on **MOCB**. It is only slightly lower than the F-score of 0.91 shown in Table 1 when using the parameter values trained on **SOSB**. This demonstrates the good applicability of our approach in different kinds of datasets.

6.2 Performance on Object Segmentation

In this section, we conduct several experiments on three benchmarks and the main objective is to demonstrate the performance of our approach when processing various kinds of scenes (e.g., simple scenes in **SOSB** and complex scenes in **MOCB**). We will also show the robustness of our approach when trained with different kinds of user labels (i.e., accurate object masks and bounding boxes). Finally, comparisons with an interactive segmentation approach are given to further demonstrate the advantage of the proposed approach.

6.2.1 Performance on Simple Scenes

This experiment focuses on the object segmentation task on simple scenes. The performances of various methods on the **SOSB** benchmark are shown in Table 1 and some representative results are given in Fig. 10.

From Table 1 and Fig. 10, we can see that for all approaches that utilize a single saliency map for each image, the quality of saliency maps has a significant influence on the segmentation performance. Intuitively, the saliency maps of Itti98, Hou07 and Seo09 have badly defined borders and incomplete internal bodies. As shown in Fig. 10b, e and f, they perform poorly even on the simple **SOSB** dataset. Comparatively, Harel07 and Goferman10, perform much better since their saliency maps can detect the salient regions more effectively (See Fig. 10d, g). However, their saliency maps also take on some ambiguity, consequently making the segmentation results miss important parts on salient objects (e.g., Fig.

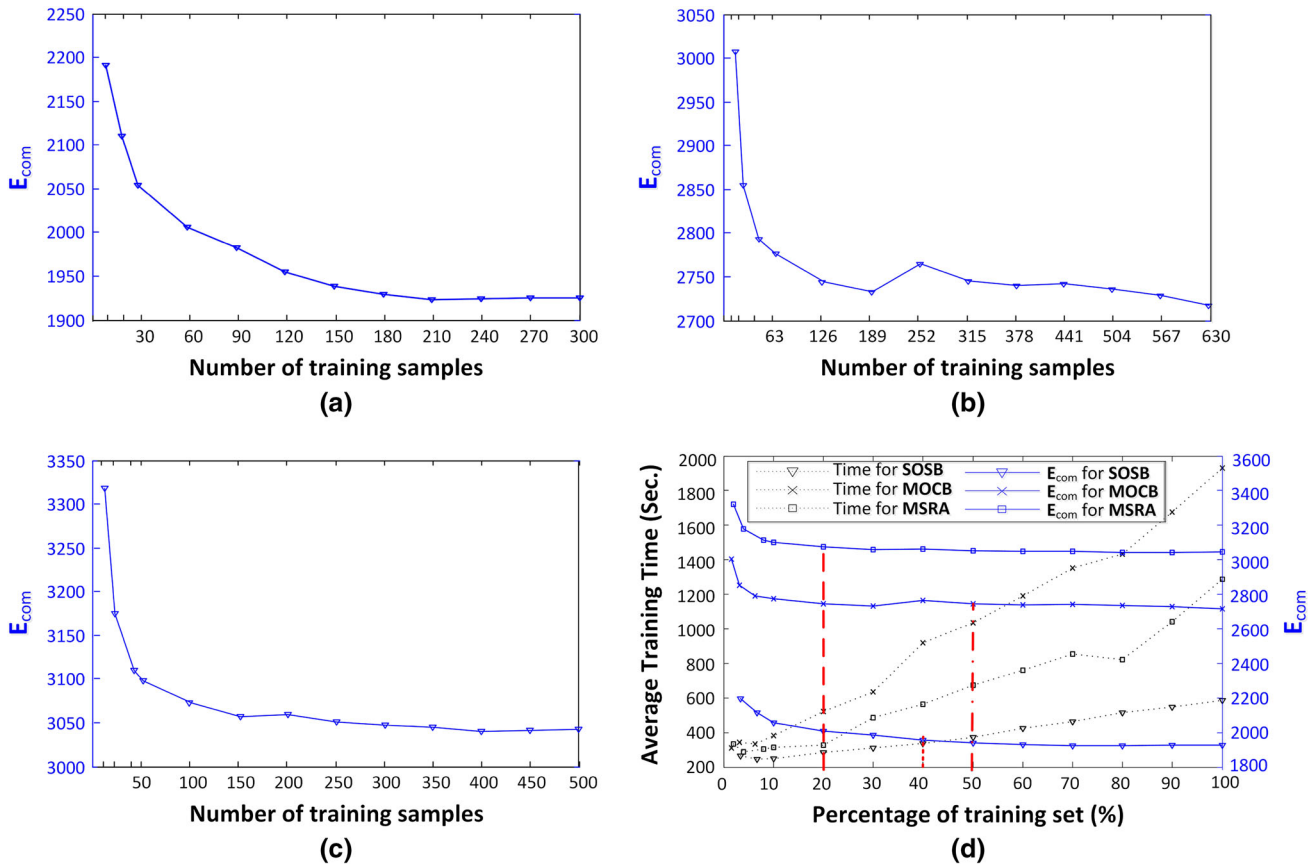

Fig. 9 The performances of the proposed learning algorithm when varying the number of training samples in dataset **a SOSB**, **b MOCB** and **c MSRA**; **d** The average training time curve on the three benchmarks

Table 1 Performance on the **SOSB** dataset

Algorithm	$\mathbf{OLP}_O$	$\mathbf{P}_O$	$\mathbf{R}_O$	$\mathbf{F}_O$	**IMP** (%)
Itti98	0.37	0.67	0.43	0.52	75.0
Gould07	0.76	0.89	0.84	0.86	5.8
Harel07	0.64	0.73	0.82	0.77	18.2
Hou07	0.35	0.72	0.38	0.50	82.0
Seo09	0.48	0.73	0.56	0.63	44.4
Goferman10	0.70	0.80	0.85	0.82	11.0
Li10	0.56	0.83	0.63	0.72	26.4
Achanta09	0.61	0.81	0.70	0.75	21.3
Carreira10	0.62	0.68	0.88	0.77	18.2
Yu10	0.74	0.83	0.88	0.86	5.8
Cheng11	0.53	0.83	0.58	0.63	44.4
Jiang11	0.71	0.87	0.78	0.82	11.1
Perazzi12	0.69	**0.90**	0.73	0.79	15.2
Li13	0.72	0.86	0.80	0.81	12.3
OUR	**0.77**	0.88	**0.93**	**0.91**	

10(d4) and (g14)), or include redundant background regions (e.g., Fig. 10(d14) and (g3)). Surprisingly, Gould07 obtains pretty good performance on **SOSB**. This is mainly because the strong photographer bias exists in **SOSB** (and its superset **MSRA**). Often, photographers tend to place the objects-of-interest near the center of their composition in order to

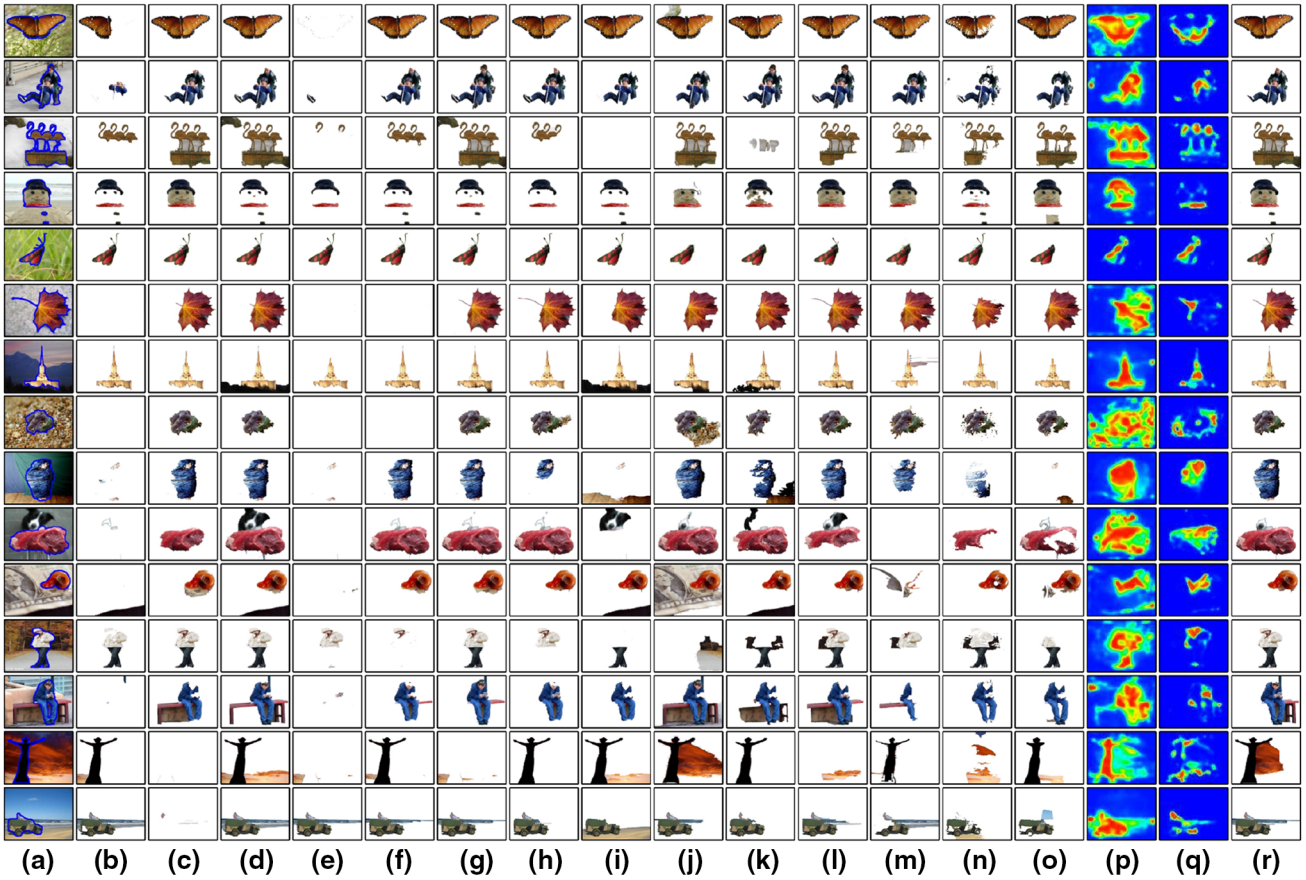

Fig. 10 Representative results on the **SOSB** dataset. **a** Original images with ground-truth; **b–o** segmented results of Itti98, Gould07, Harel07, Hou07, Seo09, Goferman10, Li10, Achanta09, Carreira10, Yu10, Cheng11, Jiang11, Perazzi12, and Li13; **p**, **q** envelope and sketch maps; and **r** segmented results of our approach

enhance their focus relative to the background (Tseng et al. 2009). However, there are also many extreme failures: in totally 30 images (about 5.8% of the test images), none of any object is detected by Gould07. Figure 10(c14) and (c15) show two examples where this bias is not satisfied.

Overall, object-based methods perform better than most of location-based methods. The proposed approach achieves the highest $\mathbf{F}_O$ of 0.91. Comparatively, Yu10's results tend to lack thin and sharp components (e.g., Fig. 10(k5), (k6)), or miss some internal bodies of salient objects (e.g., Fig. 10 (k3), (k4), (k12)). More or less, we can also find similar phenomena in the results of Cheng11, Jiang11, Perazzi12 and Li13 (e.g., Fig. 10 (l)-(o)). Instead, the proposed approach can effectively preserve elongated parts and keep the completeness of salient objects via graph cuts (e.g., Fig. 10 (r3)–(r6), (r12)). This validates our conjecture that the graph cuts framework with two complementary saliency priors can indeed improve the segmentation over that with a single saliency prior. We also noted that Carreira10 performs not so well. As shown in Fig. 10 (j), Carreira10 tends to segment objects with high recall and relatively low precision. A possible reason is that Carreira10 is originally designed to produce one segment for each individual object, rather than for foreground object segmentation.

Table 1 also presents the $\mathbf{OLP}_O$ scores of various methods. In terms of this metric, the proposed method still outperforms all other methods, though the gaps are much smaller than those measured by F-measure $\mathbf{F}_O$. Similar observations can also found in Tables 2, 3 and 4.

To further demonstrate the effectiveness of our learning algorithm, we conduct an experiment on **SOSB**. In the experiment, we simply try another three methods to generate complementary saliency maps, including: 1) VecWeight (learn one weight per raw map); 2) MinMax (select the maximum and minimum value across all raw maps to form the envelope and sketch maps, respectively) and 3) AvgWeight (assign equal weight for each raw map). We find that the F-scores of VecWeight, MinMax and AvgWeight can only reach 0.84, 0.80 and 0.81, respectively. In particular, these three methods have relatively low precision (0.80 for VecWeight, 0.76 for MinMax and 0.74 for AvgWeight). This is because their simple methods for combining raw maps (i.e., max/min values,

Table 2 Performance on the **MOCB** dataset

Algorithm	$\mathbf{OLP}_O$	$\mathbf{P}_O$	$\mathbf{R}_O$	$\mathbf{F}_O$	**IMP** (%)
Itti98	0.27	0.66	0.29	0.40	102.5
Gould07	0.55	**0.84**	0.61	0.70	15.7
Harel07	0.45	0.67	0.58	0.61	32.8
Hou07	0.29	0.68	0.31	0.42	92.9
Seo09	0.34	0.69	0.37	0.48	68.9
Goferman10	0.50	0.72	0.62	0.67	20.9
Li10	0.36	0.78	0.39	0.51	58.8
Achanta09	0.27	0.64	0.31	0.42	92.9
Carreira10	0.43	0.54	0.74	0.62	30.6
Yu10	0.52	0.73	0.63	0.68	19.1
Cheng11	0.33	0.76	0.39	0.46	76.1
Jiang11	0.43	0.77	0.50	0.61	32.8
Perazzi12	0.37	0.71	0.43	0.50	62.0
Li13	0.41	0.74	0.47	0.54	50.0
OUR	**0.66**	0.82	**0.80**	**0.81**	

Table 3 Performance when trained by bounding boxes

Algorithm	Evaluation with bounding boxes					Evaluation with object masks				
	$\mathbf{OLP}_O$	$\mathbf{P}_O$	$\mathbf{R}_O$	$\mathbf{F}_O$	**IMP** (%)	$\mathbf{OLP}_O$	$\mathbf{P}_O$	$\mathbf{R}_O$	$\mathbf{F}_O$	**IMP** (%)
Itti98	0.51	0.68	0.72	0.70	20.0	0.33	0.64	0.36	0.46	95.7
Gould07	0.69	0.86	0.79	0.82	2.4	0.58	0.83	0.93	0.88	2.3
Harel07	0.56	0.65	0.83	0.73	15.1	0.60	0.72	0.79	0.75	20.0
Hou07	0.53	0.73	0.71	0.72	16.7	0.33	0.69	0.36	0.47	91.5
Liu07	–	0.82	0.81	0.81	3.7	–	–	–	–	–
Seo09	0.56	0.74	0.72	0.73	15.1	0.41	0.70	0.46	0.55	63.6
Goferman10	0.63	0.77	0.80	0.78	7.7	0.70	0.80	0.85	0.82	9.8
Li10	0.50	**0.88**	0.32	0.47	78.7	0.58	0.81	0.64	0.72	25.0
Achanta09	0.52	0.70	0.71	0.71	18.3	0.58	0.76	0.68	0.72	25.0
Carreira10	0.57	0.67	0.82	0.74	13.5	0.62	0.68	0.88	0.76	18.4
Yu10	0.61	0.79	0.72	0.76	10.5	0.73	0.83	0.88	0.85	5.9
Cheng11	0.55	0.72	0.77	0.74	13.5	0.53	0.82	0.59	0.63	42.9
Jiang11	0.66	0.80	0.80	0.80	5.0	0.72	0.88	0.79	0.83	8.4
Perazzi12	0.64	0.70	**0.90**	0.79	6.3	0.69	**0.90**	0.73	0.79	13.9
Li13	0.66	0.78	0.82	0.80	5.0	0.73	0.86	0.80	0.81	11.1
OUR	**0.73**	0.84	0.84	**0.84**		**0.78**	0.86	**0.95**	**0.90**	

a single weight per map, or arithmetically averaging) cannot guarantee the exactness of the generated complementary maps. As a result, some redundant regions in the background are included in the segmentation results. This also implicitly validates the effectiveness of our learning algorithm in generating complementary saliency maps.

6.2.2 Performance on Complex Scenes

On the **MOCB** dataset, multiple salient objects co-exist in a complex scene, and sometimes the contrast between objects

Table 4 Performance of Grabcut with different δ

		$\mathbf{OLP}_O$	$\mathbf{P}_O$	$\mathbf{R}_O$	$\mathbf{F}_O$
Grabcut	$\delta = 0$	0.87	0.94	0.95	0.95
	$\delta = 0.1$	0.85	0.92	0.96	0.94
	$\delta = 0.2$	0.84	0.90	0.96	0.93
	$\delta = 0.3$	0.80	0.86	0.96	0.91
	$\delta = 0.4$	0.77	0.82	0.96	0.89
	$\delta = 0.5$	0.73	0.78	0.96	0.86
OUR		0.77	0.88	0.93	0.91

and the background is low. Thus, this experiment is to evaluate the robustness of the proposed approach in real-world segmentation tasks. Experimental results are given in Table 2, and some representative results are illustrated in Fig. 11.

From Table 2, we can see that our approach also outperforms all the other methods on **MOCB**. This fact reveals that the proposed approach not only works really well on simple datasets, but also keeps a relatively higher robustness on complex scenes. We also find that the performances of all the methods decrease significantly from simple scenes to complex scenes, with reasons varying for different methods. Itti98, Seo09 and Goferman10 rely on the center-surround difference of visual features. However, this strategy is hardly reliable when the background is cluttered (e.g., Fig. 11b, f, g). Hou07 uses the spectral residuals to pop out the salient objects, while in complex scenes, it is difficult to accurately locate residuals (e.g., Fig. 11e). Gould07 performs much worse on **MOCB** than on **SOSB** since objects in complex scenes exhibit much larger variability in pose and location (e.g., Fig. 11c). Li10 learns the task-related "stimulus-saliency" mapping functions and various fusion strategies for different scenes. However, when applied to the complex scenes with large intra-class variance (e.g., Fig. 11 (h4) vs. (h12)), it is difficult to find an optimal solution that can fit well for all scenes in a category. Achanta09 computes the difference between a pixel's feature and the average feature on the Gaussian-blurred image. When colors are mixed, the results are undesirable (e.g., Fig. 11i). The reason for Yu10's decrease is because it uses ad hoc complementary saliency maps which cannot directly adapt to complex datasets. As shown in Fig. 11k, most segmented objects miss some internal bodies. Similarly, the other four salient object-based approaches, including Cheng11, Jiang11, Perazzi12, and Li13, also perform poorly on **MOCB** (e.g., Fig. 11 (l-o)).

Note that the performance of the proposed approach also significantly decreases, mainly owing to the fact that some raw maps are inaccurate and even misleading. Despite this, our learning process makes it possible to integrate these inaccurate raw maps to generate two more reliable complementary maps. Given the two maps, our approach can segment objects with complex structures (e.g., Fig. 11 (r7), (r10)). Even when the contrast between objects and background is low, our approach still yields a satisfying result (e.g., Fig. 11 (r1–r3), (r6), (r8)). However, it should be acknowledged that the object segmentation task on complex scenes remains very challenging. Thus there is still a much room to improve the performance of the proposed method.

6.2.3 Performance when Trained by Bounding Boxes

In practice, manually labelling images with precise object masks is very tedious. Thus, this experiment is designed to show whether roughly annotated samples are applicable for training if precisely-labelling samples are not available. Although the training samples are labelled with approximate bounding boxes, we also wish our learning algorithm remain unchanged and the final segmentation outputs still in the form of object masks. We can quantitatively compare our results with others in the forms of boxes and object masks. When using boxes, a rectangle with the smallest area is simply drawn to enclose at least 98% of object pixels in each of the segmented results. In this experiment, we use the **MSRA** dataset since it offers bounding boxes enclosing the salient objects as the ground-truth. The **SOSB** dataset, a subset of **MSRA**, is also used for performance evaluation in terms of object masks.

Table 3 presents the performances of various methods, respectively evaluated by boxes or object masks. Here the performance of Liu07 is directly cited from Liu et al. (2007). We notice that the output form of bounding boxes narrows the performance gaps between different methods. Nevertheless, our approach still outperforms the others, with an F-measure of 0.84 (evaluated by boxes) or 0.90 (evaluated by object masks). As shown in Fig. 12, our results are very satisfying.

Interestingly, when trained by bounding boxes, all saliency-based methods can obtain comparable or slightly lower results with those trained by object masks. Clearly, this should be mainly attributed to the generalization of the graph cuts framework. However, the robustness of our saliency learning algorithm is also an important factor. Clearly, other forms of ground-truth such as ellipses can also be directly applied to our approach as training data.

6.2.4 Comparisons with Grabcut

In the last experiment, we want to verify whether our approach can obtain comparable results with the popular interactive method, Grabcut (Rother et al. 2004). This experiment is performed on **SOSB** because each image in this dataset has both forms of ground-truth data. Usually, Grabcut needs the user to drag a rectangle around an object to bootstrap the segmentation. However, given the same image, different people may draw different rectangles and the precision of these rectangles is crucial to segmentation. For a fair comparison, we simulate six interactive cases by forming rectangles of different sizes based on the bounding-box ground-truth data, where the size difference δ is set to [0, 0.5]. We evaluate the performances of Grabcut in different cases. The results are listed in Table 4, and some representative results are shown in Fig. 13.

Table 4 shows that our approach can obtain comparable results in terms of F-measure with Grabcut when $\delta = 0.3$. The recalls of Grabcut in all cases are very high since each interactive rectangle can almost enclose all parts of an object (see Fig. 13b). However, with larger δ, more redun-

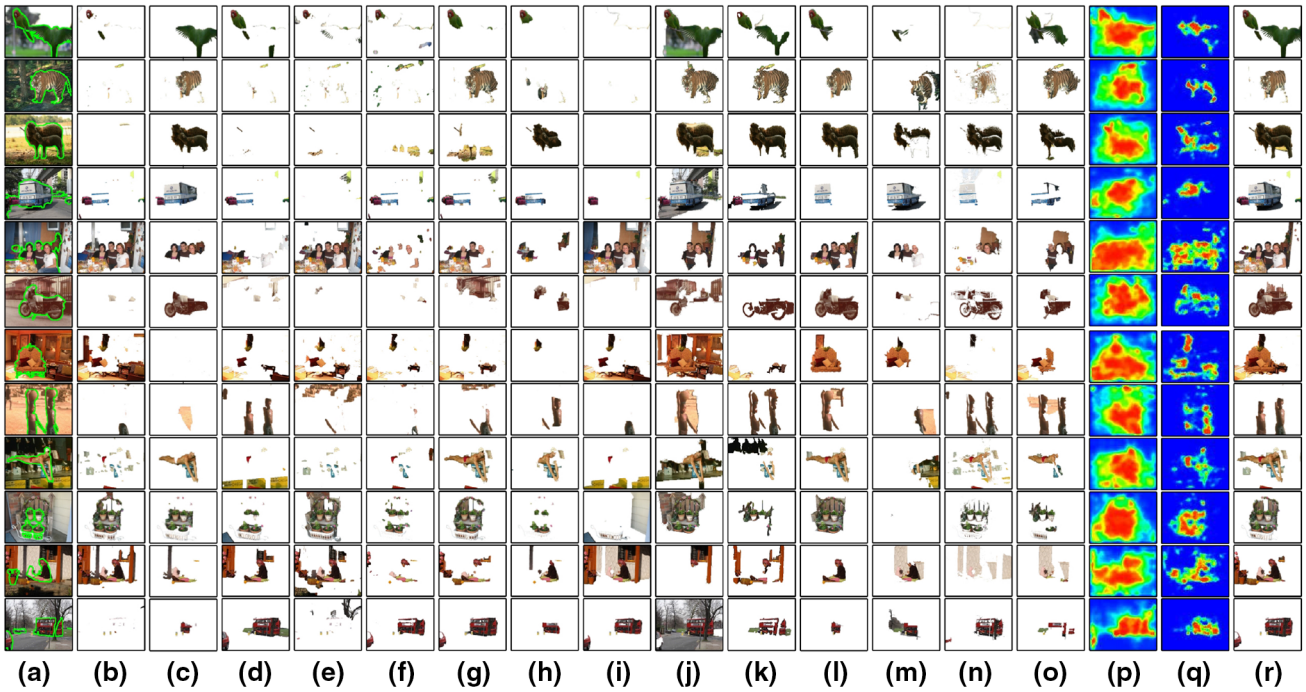

Fig. 11 Representative results on the **MOCB** dataset. **a** Original images with ground-truth; **b**–**o** Segmented results of Itti98, Gould07, Harel07, Hou07, Seo09, Goferman10, Li10, Achanta09, Carreira10, Yu10, Cheng11, Jiang11, Perazzi12, and Li13; **p**, **q** Envelope and sketch maps; and **r** Segmented results of our approach

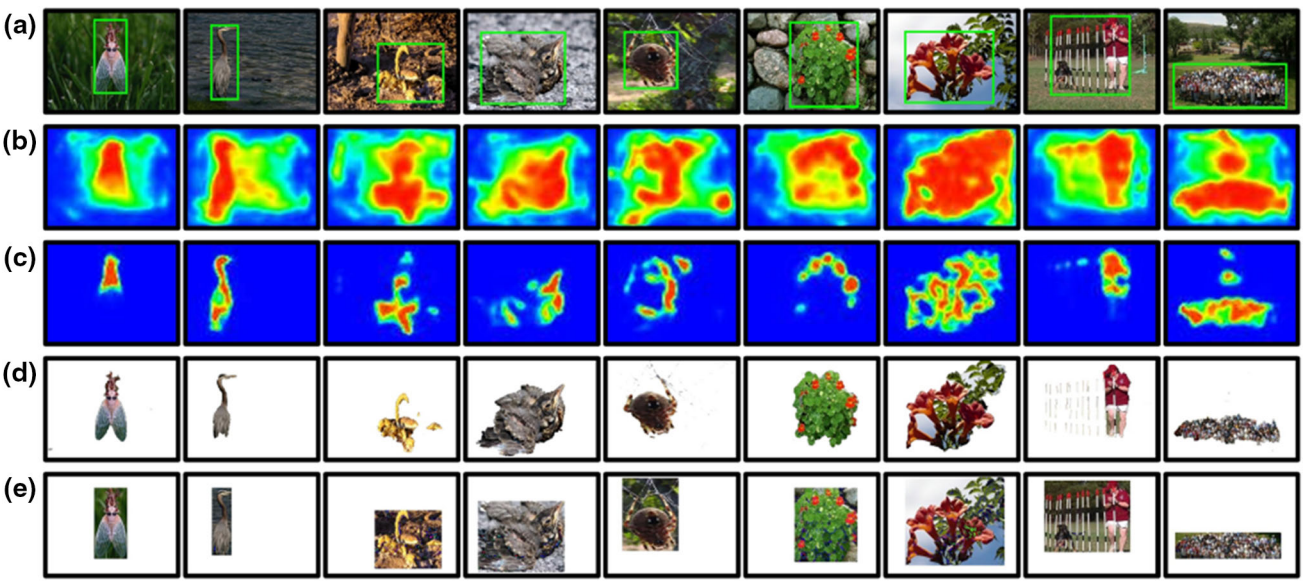

Fig. 12 Representative results of the proposed approach on the **MSRA** dataset when trained by bounding boxes. **a** Original images with ground-truth; **b** Envelope maps; **c** Sketch maps; **d** Segmentation results in object masks; **e** Segmentation results in bounding boxes

dant background will also be included in the segmented results, leading to a rapid decrease of Grabcut's segmentation precision.

In some cases, our approach *will* identify some background regions as objects, since they may be indeed visually "salient" (e.g., Fig 13 (i6)). Nevertheless, with the help of interactive rectangles to confine the possible range of objects, Grabcut can avoid such regions being included in the results (e.g., Fig. 13 (c6)–(f6)). In other cases, if some parts of an object are not identified as "salient" by all of the used raw maps, the learned complementary maps cannot perfectly highlight this object, consequently leading to an incomplete segmentation (e.g., Fig 13 (i5)). In fact, these cases are difficult for nearly all automatic saliency-based methods.

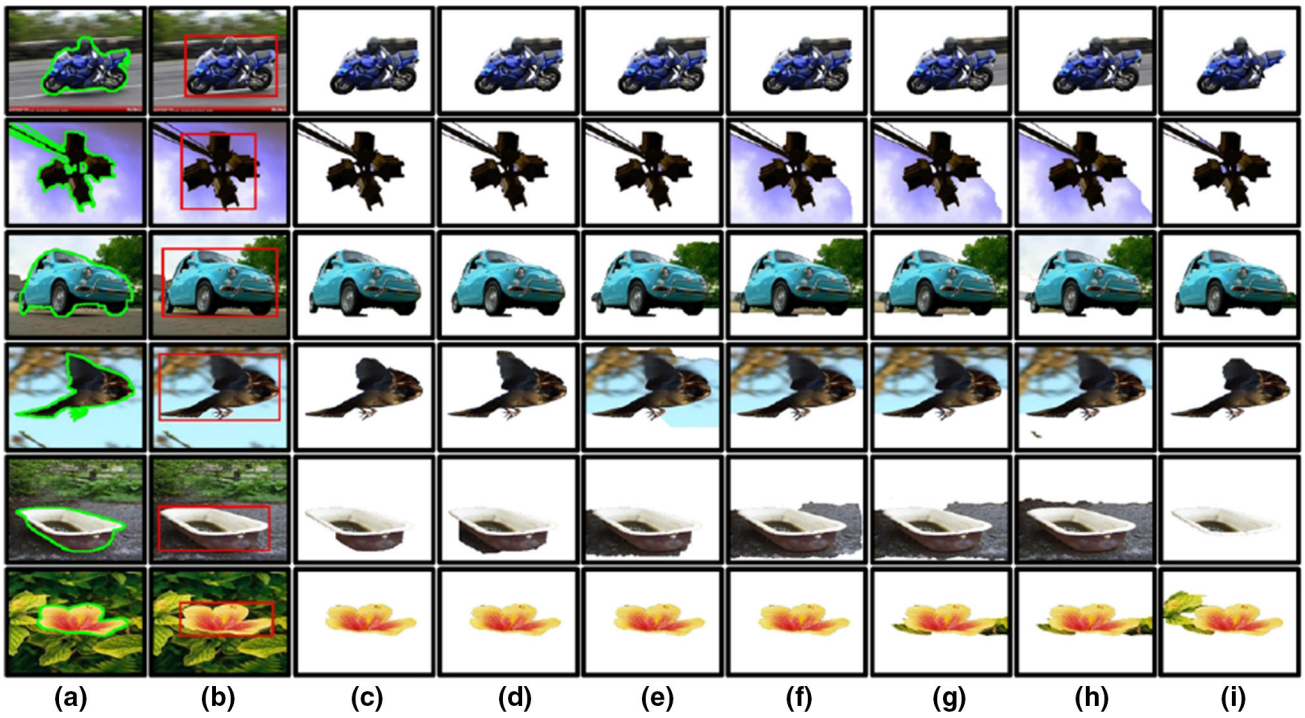

Fig. 13 Some representative results when compared with Grabcut. **a** Original images with object-mask ground-truth; **b** "Precise" interactive rectangles for Grabcut; **c**–**h** Grabcut's results given different sizes of interactive rectangles, respectively with $\delta = 0, 0.1, 0.2, 0.3, 0.4$, and 0.5; **i** Segmented results of the proposed approach

7 Conclusion

In this paper, we propose a novel automatic approach for foreground object segmentation based on complementary saliency priors. We learn to generate complementary maps according to the Maximum Likelihood estimate, making saliency priors general and robust for the purpose of foreground object segmentation. We then solve the segmentation task using graph cuts with complementary saliency priors. Experimental results show that our approach outperforms several state-of-the-art methods and can even obtain comparable results with the popular interactive method Grabcut.

In future work, we will extend the graph cuts framework by incorporating more visual features (e.g., shape and texture). This will lead to better visual appearance models for objects and for the background, and consequently make the proposed approach more robust in real-world segmentation tasks.

References

Achanta, R., Estrada, F., Wils, P., & Susstrunk, S. (2008). Salient region detection and segmentation. In *IEEE international conference on computer vision*, pp 66–75.

Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009). Frequency-tuned salient region detection. In *IEEE conference on computer vision and pattern recognition*, pp. 1597–1604.

Borenstein, E., & Ullman, S. (2004). Learning to segment. In *The 8th European conference on computer vision*, pp. 315–328.

Borji, A., Sihite, D. N., & Itti, L. (2012). Salient object detection: A benchmark. In *European conference on computer vision 2012, Part II, LNCS 7573*, pp. 414–429.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Boykov, Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE international conference on computer vision*, pp. 105–112.

Boykov, Y., & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(11), 1124–1137.

Cheng, M. M., Zhang, G. X., Mitra, N. J., Huang, X., & Hu, S. M. (2011). Global contrast based salient region detection. In *IEEE conference on computer vision and pattern recognition*, pp. 409–416.

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J., & Zisserman, A. (2009). The pascal visual object classes challenge 2009 (voc2009) results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

Goferman, S., Manor, L. Z., & Tal, A. (2010). Context-aware saliency detection. In *IEEE conference on computer vision and pattern recognition*, pp. 2376–2383.

Gopalakrishnan, V., Hu, Y., & Rajan, D. (2009). Random walks on graphs to model saliency in images. In *IEEE conference on computer vision and pattern recognition,*, pp. 1698–1705.

Gould, S., Rodgers, J., Cohen, D., Elidan, G., & Koller, D. (2007). Multi-class segmentation with relative location prior. *Ineternational Journal of Computer Vision*, *80*(3), 300–316.

Harel, J., Koch, C., & Perona, P. (2007). Graph-based visual saliency. In *Neural information processing systems*, *19*, 545–552.

Hou, X., & Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE conference on computer vision and pattern recognition*.

Hua, G., Liu, Z., Zhang, Z., & Wu, Y. (2006). Iterative local-global energy minimization for automatic extraction of objects of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1701–1706.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254–1259.

Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N., & Li, S. (2011). Automatic salient object segmentation based on context and shape prior. In *British machine vision conference*, pp. 1–12.

Judd, T., Ehinger, K., Durand, F., & Torralba, A. (2009). Learning to predict where humans look. In *International conference on computer vision*, pp. 2106–2113.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331.

Kolmogorov, V., & Zabih, R. (2004). What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(2), 147–159.

Lempitsky, V., Kohli, P., Rother, C., & Sharp, T. (2009). Image segmentation with a bounding box prior. In *IEEE international conference on computer vision*, pp. 277–284.

Li, C., Xu, C., Gui, C., & Fox, M. D. (2005). Level set evolution without re-initialization: A new variational formulation. In *IEEE conference on computer vision and pattern recognition*, pp. 430–436.

Li, F., Carreira, J., & Sminchisescu, C. (2010a). Object recognition as ranking holistic figure-ground hypotheses. In *IEEE conference computer vision and pattern recognition*, pp. 1712–1719.

Li, J., Tian, Y., Huang, T., & Gao, W. (2010b). Probabilistic multi-task learning for visual saliency estimation in video. *International Journal of Computer Vision*, *90*(2), 150–165.

Li, J., Tian, Y., Duan, L., & Huang, T. (2013). Estimating visual saliency through single image optimization. *IEEE Signal Processing Letters*, *20*(9), 845–848.

Liu, G., Lin, Z., Yu, Y., & Tang, X. (2010). Unsupervised object segmentation with a hybrid graph model (hgm). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(4), 910–924.

Liu, T., Sun, J., Zheng, N., Tang, X., & Shum, H. (2007). Learning to detect a salient object. In *IEEE conference on computer vision and pattern recognition*, pp. 1–8.

Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., et al. (2011). Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(2), 353–367.

Ma, Y., & Zhang, H. (2003). Contrast-based image attention analysis by using fuzzy growing. In *The 11th ACM international conference on multimedia*, pp. 374–381.

Mehrani, P., & Veksler, O. (2010). Saliency segmentation based on learning and graph cut refinement. In *British machine vision conference*, pp. 1–12.

Movahedi, V., & Elder, J. H. (2010). Design and perceptual validation of performance measures for salient object segmentation. In *IEEE workshop on perceptual organization in computer vision*.

Perazzi, F., Krahenbuhl, P., Pritch, Y., & Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *IEEE conference on computer vision and pattern recognition*, pp. 733–740.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut—Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, *23*(3), 309–314.

Rother, C., Minka, T., Blake, A., & Kolmogorov, V. (2006). Cosegmentation of image pairs by histogram matchingincorporating a global constraint into mrfs. In *IEEE conference on computer vision and pattern recognition*, pp. 993–1000.

Seo, H. J., & Milanfar, P. (2009). Static and space-time visual saliency detection by self-resemblance. *Journal of Vision, 9*(12):article 15, 1–27.

Tseng, P., Carmi, R., Cameron, L., Munoz, D. P., & Itti, L. (2009). Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of Vision*, *9*(7), 1–16.

Vicente, S., Kolmogorov, V., & Rother, C. (2008). Graph cut based image segmentation with connectivity priors. In *IEEE conference on computer vision and pattern recognition*, pp. 1–8.

Walther, D., & Koch, C. (2006). Modeling attention to salient pro-to-object. *Neural Networks*, *19*, 1395–1407.

Winn, J., & Jojic, N. (2005). Locus: Learning object classes with unsupervised segmentation. In *IEEE international conference on computer vision*, pp. 756–763.

Yu, H., Li, J., Tian, Y., & Huang, T. (2010). Automatic interesting object extraction from images using complementary saliency maps. In *2010 ACM international conference on multimedia*, pp. 891–894.

Yu, S., Gross, R., & Shi, J. (2002). Concurrent object recognition and segmentation by graph partitioning. In *Neural information processing systems*, *14*, 1383–1390.

Zhang, L., Tong, M. H., Marks, T. K., Shan, H., & Cottrell, G. W. (2008). Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, *8*(7), 1–20.

Zhao, L., & Davis, L. (2005). Closely coupled object detection and segmentation. In *IEEE international conference on computer vision*, pp. 454–461.