Pattern Recognition 60 (2016) 348–360

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Bottom-up saliency detection with sparse representation of learnt texture atoms ☆

Mai Xu [a], Lai Jiang [a], Zhaoting Ye [a], Zulin Wang [a,b,*]

[a] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
[b] Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, China

ARTICLE INFO

Article history:
Received 21 January 2016
Received in revised form
11 April 2016
Accepted 14 May 2016
Available online 21 May 2016

Keywords:
Visual attention
Saliency detection
Sparse representation
Dictionary learning

ABSTRACT

This paper proposes a saliency detection method by exploring a novel low level feature on sparse representation of learnt texture atoms (SR-LTA). The learnt texture atoms are encoded in salient and non-salient dictionaries. For salient dictionary, a formulation is proposed to learn salient texture atoms from image patches attracting extensive attention. Then, the online salient dictionary learning (OSDL) algorithm is presented to solve the proposed formulation. Similarly, the non-salient dictionary is learnt from image patches without any attention. Then, the pixel-wise SR-LTA feature is yielded based on the difference of sparse representation errors, regarding the learnt salient and non-salient dictionaries. Finally, image saliency can be predicted by linearly combining the proposed SR-LTA feature and conventional features, luminance and contrast. For the linear combination, the weights of different feature channels are determined by least square estimation on the training data. The experimental results show that our method outperforms 9 state-of-the-art methods for bottom-up saliency detection.

1. Introduction

Saliency detection refers to computing on image features to characterize the regions attracting different amounts of visual attention in a scene. Generally speaking, saliency detection is extensively studied in the context of the human visual system (HVS). Similar to the HVS, saliency detection enables machines to survive from processing a deluge of visual data. Thus, it has been widely applied in computer vision and image processing areas, such as object detection [1], object recognition [2], image retargeting [3], image quality assessment [4], and image/video compression [5].

For predicting visual attention, saliency detection can be traced back to feature integration theory [6] by Treisman and Gelade in 1980, which discussed on the possible visual features related to visual attention. To combine these features together, Koch and Ullman [7] in 1987 proposed to generate the saliency map for an input image, indicating which regions are conspicuous to attract attention in the HVS. Specifically, saliency map is a matrix with the same size as the input image, and the values of its elements range from 0 to 1. The large saliency value indicates high probability to attract human attention. Later, Itti and Koch [8] found out that the low level feature channels of intensity, color, and orientation are effective in generating the saliency map. In their method, these feature channels are decomposed for images at various scales subsampled by a Gaussian pyramid, and then conspicuity maps are constructed by center-surround responses to the decomposed feature channels. In each channel, conspicuity maps are aggregated across different scales. Finally, the saliency map can be obtained by the linear integration of conspicuity maps of all channels. Benefiting from the success of Itti's model [8], extensive saliency detection methods (e.g., [9–13]), using biological plausible features, have been proposed in the past decade.

Recently, several saliency detection methods (e.g., [14–18]) have been proposed to learn the parameters or even features from the ground-truth eye fixations[1] of training images, for saliency detection. From the perspective of parameters, Zhao and Koch [16] presented a method to learn weights associated with conspicuity maps for different feature channels, with least square fitting to fixations. This replaces the equal weight assignment in [8,19], thus improving the saliency detection accuracy. However,

☆A short version of this paper has been presented in ICCV Workshop 2015.

* Corresponding author. Tel.: +86 10 82317201.
E-mail addresses: MaiXu@buaa.edu.cn (M. Xu), wzulin@buaa.edu.cn (Z. Wang).

[1] Fixations are the points where people look during the eye tracking experiment. They are seen as the ground-truth of visual attention.

http://dx.doi.org/10.1016/j.patcog.2016.05.023
0031-3203/

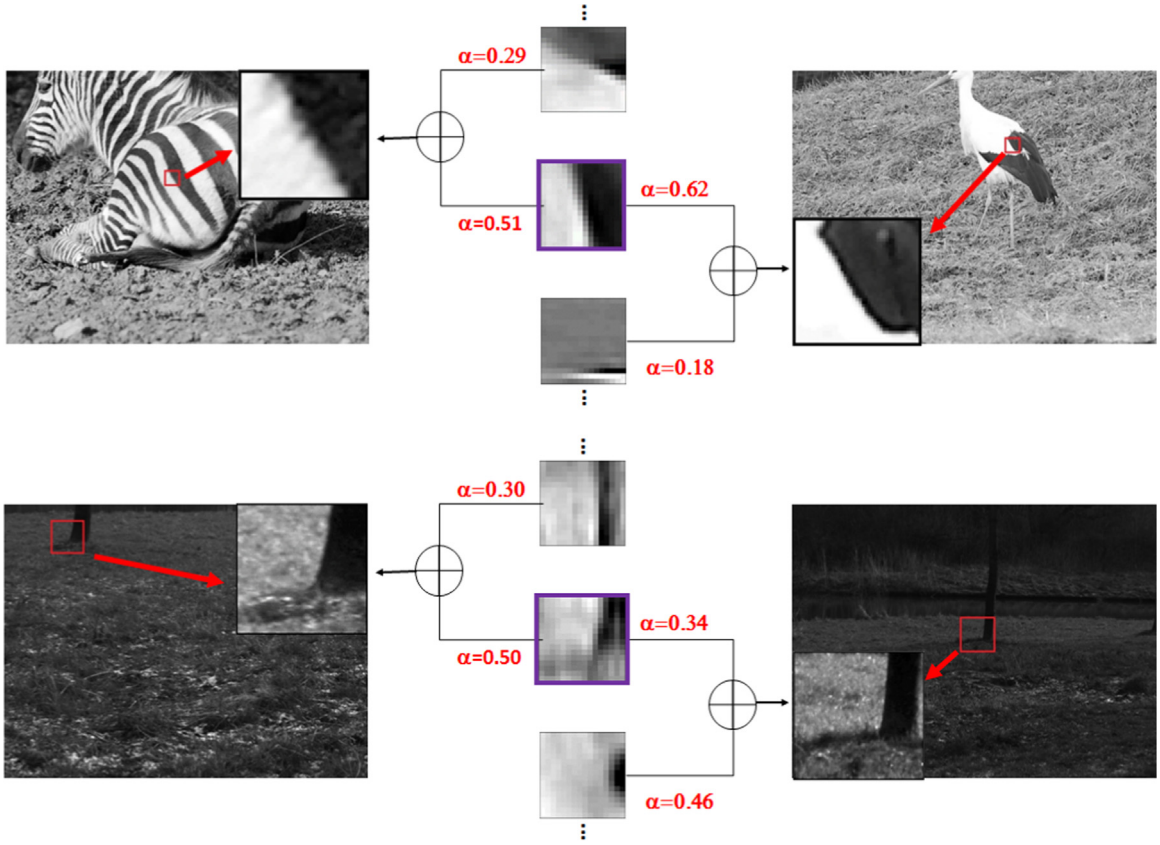

Fig. 1. An example of salient patches with similar texture patterns. The regions inside the red squares (enlarged in the corners) are salient patches, in the images of the eye tracking Kienzle database (the first row) and Doves database (the second row). Some atoms of the dictionaries, learnt from the salient regions of other training images, are shown in the middle of two images. In addition, the sparse representation coefficients α of the salient patterns regarding the learnt dictionaries are also provided. It can be seen that the salient patches across the different images may share some similar basic patterns, and these basic patterns may be learnt from the training data. Note that the patch sizes are 96×96 for DOVES and 41×41 for Kienzle et al., to ensure that the corresponding fovea degrees are around 1.5° in each database.

only few parameters can be learnt in these methods, such that the performance of these methods depends on the features of the conventional methods.

From the perspective of features, Kienzle et al. [17,18] proposed to directly learn patch patterns of salient and non-salient regions from the ground-truth eye tracking data. These patterns can be seen as low level features attracting different amount of visual attention. Specifically, two center-surround texture patches are learnt as the most relevant patterns for drawing visual attention, and two other patches are learnt as the least possible patterns for receiving eye fixations. Then, the saliency of an image patch can be detected, on the basis of the distance to the learnt texture patterns. However, the learnt four patch patterns have limited expression, since only two positive and two negative patterns are available for saliency detection.

Fig. 1 shows the possibility of learning hundreds of salient patterns (by applying the dictionary learning algorithm) for saliency detection. Accordingly, this paper proposes to learn extensive positive and negative patterns from the eye tracking data of training images, for bottom-up saliency detection. Specifically, this paper first proposes a formulation with a novel center-surround term, for learning two discriminative dictionaries. These two dictionaries contain the atoms for basic texture patterns of salient and non-salient regions, respectively. In light of online dictionary learning [20], we develop an online salient dictionary learning (OSDL) algorithm to solve the proposed formulation, and then the salient and non-salient dictionaries can be learnt from the eye tracking data of training images. Given the learnt dictionaries, a novel feature based on sparse representation of learnt texture atoms (SR-LTA) is worked out in our method. Such a feature is generally based on the errors of sparse representation regarding salient and non-salient dictionaries. Next, the saliency of an image can be predicted, via combining the SR-LTA feature with conventional luminance and contrast features. For the linear combination, the weights corresponding to each feature channel are estimated via least square fitting on the training data. Similar to other bottom-up methods [21,17,18], this paper only works on gray images with natural scenes.

In summary, the main contributions of this paper are two-folds:

- We address a novel dictionary learning formulation solved by the proposed OSDL algorithm, for generalizing salient and non-salient dictionaries from training eye tracking data.
- We propose the SR-LTA feature in light of the learnt dictionaries, together with other two conventional features (luminance and contrast), for bottom-up saliency detection of gray images.

This paper is the extended version of our conference paper [22], with some advanced work. The advances are summarized as follows. First, the related work of saliency detection is extensively reviewed, from biologically inspired and learning based aspects. Second, this paper provides technical details about the derivation of our method, e.g., the derivation of dictionary updating in our ODSL algorithm. Third, we analyze the computational time of our method, by comparing to other methods. At last, we provide more comprehensive comparison and analysis in this paper. For example, we compare our method with the latest work of [23], and show that our method still outperforms [23] in bottom-up saliency

detection. More importantly, we thoroughly analyze the performance of our method from three aspects, i.e., feature effectiveness, learning performance and robustness.

2. Related work

The existing methods on saliency detection can be classified into two categories: either biologically inspired or learning based models. In the following, we briefly review the saliency detection literatures on these two categories, respectively.

2.1. Biologically inspired saliency detection

Most saliency detection methods are biologically inspired, i.e., they are developed according to the understanding of the HVS. To be more specific, inspired by study on the eye movement deployment of the HVS on images, the computational models on image features have been extensively explored to detect saliency. The representative work on detecting image saliency is Itti's model [8], which combines center-surround features of color, intensity and orientation together. Afterwards, Koch and Ullman [24] extended the Itti's model by incorporating the proto-object inference in the saliency map produced by [8]. Benefitting from the recent success on graph theory, graph-based visual saliency (GBVS) method [10] has been proposed to model the saliency of an image, by forming and then normalizing activation maps that also depend on the low level features of color, intensity and orientation. In [10], a fully connected graph is built, which uses directed edges to represent the weights on feature dissimilarity between different locations. Then, the equilibrium distribution over map locations is treated as activation and saliency values, by defining the equivalence relation in the graphs with Markov chains. Beyond, a few advanced graph-based methods [25,26] have been proposed recently. Besides, it is intuitive that the most informative part of a scene can attract human attention. Therefore, from the perspective of information theory, some methods [9,27–29] have been proposed to measure the image entropy for saliency estimation. For example, attention based on information maximization (AIM) [9] was proposed to measure self-information entropy of visual features for detecting saliency in images. Besides, benefitting from the recent development in signal processing area, some latest signal processing tools have been incorporated in saliency detection, e.g., spectral analysis based [11,30,31], principle component analysis (PCA) based [12], and sparse representation based [32,33] methods. Most recently, some other state-of-the-art methods, e.g., adaptive whitening saliency (AWS) [34], boolean map based saliency (BMS) [13] and nonlocal center-surround reconstruction [35], have also been proposed to detect image saliency.

However, the understanding of the HVS is still in its infancy, and biologically inspired saliency detection thus has a long way to go yet. Recently, machine learning techniques have emerged as a potential way to construct visual attention model from the eye tracking data. Our method mainly focuses on learning the discriminative dictionaries from the training eye tracking data for saliency detection, rather than simply using the spatially or temporally neighboring patches as in the previous saliency detection work [32,33]. Next, we briefly overview the existing methods on learning based saliency detection.

2.2. Learning based saliency detection

The learning based saliency detection methods have emerged during the past decade. The central of these methods is learning visual attention models from eye tracking data. Here, the eye tracking data are normally obtained by using the eye tracker to record fixations of several observers on viewing specific images [15]. Generally speaking, the learning based saliency detection methods can be further divided into three categories: the learning of parameter, bottom-up feature and semantic feature. The following briefly reviews these three classes of methods.

2.2.1. Learning parameters for saliency detection

Typically, several features on determining visual attention are linearly combined with equal weights, for predicting saliency maps. Recently, some methods [14,36,15,16] have been proposed to learn such weights from eye tracking data, through optimal fitting to the ground-truth fixations. For example, the weights of conspicuity maps of three features in Itti's model [8] are equally set to 1/3. To improve the precision of saliency detection, Itti and Koch [14] proposed to learn the optimal weights of each saliency feature, with minimal square error on saliency prediction.

Afterwards, [16] extended to learn the weights of both top-down (i.e., face) and bottom-up features to enhance the accuracy of saliency detection in [8] and [19]. Moreover, Judd et al. [15] integrated a great number of low, middle and high level image features together, with their corresponding weights learnt through a linear SVM classifier. However, the above learning based methods have to work together with the biologically inspired saliency detection methods, and hence they can only make some improvement over those conventional methods, with the help of ground-truth eye fixation data.

2.2.2. Learning bottom-up features for saliency detection

Towards effective saliency detection, we may learn the relevant features from the eye tracking data [17,37,18,21,38,23], instead of studying on the HVS. To be more specific, Kienzle et al. [17,18] proposed a nonparametric bottom-up approach on learning from the eye tracking data to obtain the patches of texture patterns, corresponding to positive and negative eye fixations. As a result, two center-surround texture patches are learnt as the most relevant patterns for drawing visual attention, and two other patches are learnt as the least possible patterns for receiving fixations. Then, the saliency of an image patch can be calculated with a simple feed-forward network, which integrates the radial basis units of ℓ_2 norm distances between the current image patch and four learnt texture patterns. Later, they developed a similar approach [37] to model video saliency, which further learns the bottom-up temporal filters from the eye tracking data. In addition, a gaze-attentive fixation finding engine (GAFFE) [21] was developed to detect saliency, based on four low level image features: luminance, contrast, and bandpass outputs of luminance and contrast. In GAFFE, the bandpass filters of both luminance and contrast were learnt from the extensive eye tracking data [39], which can significantly enhance the accuracy of saliency detection.

2.2.3. Learning semantic features for saliency detection

Most recently, a few saliency detection methods [38,40,23] have been proposed to learn high level semantic features, which contain certain semantic information, such as face and object. These methods usually benefit from the great success of deep learning. For example, Huang et al. [23] proposed the saliency in context (SALICON) method to incorporate high level semantic features in saliency detection, which are learnt by deep neural networks. However, the performance of these methods heavily relies on the existence of semantic objects. In other words, they are not “good at” finding low level features attracting visual attention. Besides, these deep learning based methods require the significantly sufficient training data, which normally take great effort on collecting the eye tracking data.

This paper mainly concentrates on learning low level texture features by unitizing sparse representation and dictionary

learning, for bottom-up saliency detection of gray images. Specifically, our method learns two discriminative dictionaries (i.e., salient and non-salient dictionaries) from the eye tracking data, thus avoiding the simplicity of salient/non-salient texture patterns in [17,18]. Then, our method computes the SR-LTA feature, upon sparse presentation error of image patches with regard to the learnt salient and non-salient dictionaries. Compared with the learnt bandpass filters of [21], it has advantage in searching the center-surround patterns, which are indeed important in determining saliency maps of gray images. Finally, a parameter learning mechanism is also adopted in our method for assigning optimal weights to the feature channels of the luminance, contrast and SR-LTA.

3. Dictionary learning for salient and non-salient texture atoms

In this section, we apply the dictionary learning method to learn both salient and non-salient texture atoms to provide the low level texture feature for saliency detection. We introduce in Section 3.1 our dictionary learning formulation on generalizing both salient and non-salient texture atoms. In Section 3.2, we present a solution to the proposed dictionary learning formulation.

3.1. Dictionary leaning formulation

In sparse representation, an image patch[2] $\mathbf{x} \in \mathbb{R}^m$ can be sparsely represented by only a few texture atoms of dictionary $\mathbf{D} \in \mathbb{R}^{m\times k}$. Specifically, sparse coefficients $\boldsymbol{\alpha} \in \mathbb{R}^k$ need to be calculated for estimating image patch $\mathbf{x}$ with respect to dictionary $\mathbf{D}$. In fact, the problem of sparse representation can be formulated by

$$\min_{\boldsymbol{\alpha}} \| \mathbf{x} - \mathbf{D}\boldsymbol{\alpha} \|_2^2 \quad \text{s. t.} \quad \| \boldsymbol{\alpha} \|_0 \le L, \tag{1}$$

where L is the sparsity level of coefficients $\boldsymbol{\alpha}$. In (1), the atoms in $\mathbf{D}$ indicate the basic texture patterns for reconstructing image patches. Here, dictionary $\mathbf{D}$ needs to be learnt from training image patches $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$. This can be achieved [41,42] by

$$\min_{\mathbf{D},\mathbf{A}} \frac{1}{n} \sum_{i=1}^{n} (\| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \|_2^2 + \lambda \| \boldsymbol{\alpha}_i \|_1), \tag{2}$$

where $\mathbf{A} = \{\boldsymbol{\alpha}_i\}_{i=1}^n$ is the set of sparse representation coefficients corresponding to $\mathbf{X}$. In (2), λ is a regularization parameter, representing the tradeoff between the reconstruction error $\| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \|_2^2$ and sparsity level $\| \boldsymbol{\alpha}_i \|_1$. Next, based on (2), we concentrate on the proposed formulation on learning two dictionaries for salient texture and non-salient texture atoms, respectively. Since the center-surround patterns play an important role in attracting human visual attention [18], a novel center-surround term is incorporated in our formulation to encourage/discourage the center-surround patterns in the learnt salient/non-salient dictionary.

To be more specific, we first propose a weight function for encouraging the center-surround patterns in the learnt dictionary with salient texture atoms. In our weight function, the weight of each pixel in an atom is imposed according to its Euclidean distance to the atom's center. Assume that there are N different Euclidean distances sorted in an ascending order. In each atom, the weight for the pixels with the q-th sorted Euclidean distance can be calculated:

$$W(q) = \frac{1}{n_q} \cos\left(\frac{q}{N}\cdot\pi\right), \tag{3}$$

where n_q stands for the number of pixels with the q-th Euclidean distance. An example for weight function is shown in Fig. 2.

Then, the set of weights $W(q)$ for all pixels in an atom is represented by vector $\boldsymbol{l}^T \in \mathbb{R}^{1\times m}$. Note that m is the total number of pixels in an atom. Upon $\boldsymbol{l}^T$, the center-surround term can be designed by $\| \boldsymbol{l}^T\mathbf{D} \|_2^2$, which quantifies the degree of center-surround.

Given the center-surround term, we have the following optimization formulation to learn (salient and non-salient) texture atoms by rewriting (2):

$$\underbrace{\min_{\mathbf{D}',\mathbf{A}'} \frac{1}{n'} \sum_{\mathbf{x}_i'\in\mathbf{S}'} (\| \mathbf{x}_i' - \mathbf{D}'\boldsymbol{\alpha}_i' \|_2^2 + \lambda \| \boldsymbol{\alpha}_i' \|_1 - \eta \| \boldsymbol{l}^T\mathbf{D}' \|_2^2)}_{\text{Salient dictionary learning,}}$$

$$\underbrace{\min_{\mathbf{D}'',\mathbf{A}''} \frac{1}{n''} \sum_{\mathbf{x}_i''\in\mathbf{S}''} (\| \mathbf{x}_i'' - \mathbf{D}''\boldsymbol{\alpha}_i'' \|_2^2 + \lambda \| \boldsymbol{\alpha}_i'' \|_1 + \eta \| \boldsymbol{l}^T\mathbf{D}'' \|_2^2)}_{\text{Non-salient dictionary learning}}, \tag{4}$$

where $\mathbf{S}'$ is the training set of fixation patches[3] denoted by $\{\mathbf{x}_i'\}_{i=1}^{n'}$, and $\mathbf{S}''$ is the training set of non-fixation patches denoted by $\{\mathbf{x}_i''\}_{i=1}^{n''}$. $\{\boldsymbol{\alpha}_i'\}_{i=1}^{n'}$ and $\{\boldsymbol{\alpha}_i''\}_{i=1}^{n''}$ are sparse representation coefficients corresponding to $\{\mathbf{x}_i'\}_{i=1}^{n'}$ and $\{\mathbf{x}_i''\}_{i=1}^{n''}$, respectively. In addition, n' and n'' are the numbers of image patches in $\mathbf{S}'$ and $\mathbf{S}''$. In (4), $\mathbf{D}'$ is the dictionary with salient texture atoms, learnt from the training fixation patches, and $\mathbf{D}''$ is the non-salient dictionary generalized from training non-fixation patches. $\eta(>0)$ is a regularization parameter to control the influence of the center-surround term. Obviously, the center-surround degree is encouraged for the atoms in salient dictionary $\mathbf{D}'$, as $\| \boldsymbol{l}^T\mathbf{D}' \|_2^2$ needs to be large when $\eta > 0$. On the contrary, the center-surround degree is discouraged in non-salient dictionary $\mathbf{D}''$ by making $\| \boldsymbol{l}^T\mathbf{D}'' \|_2^2$ small.

3.2. Solution to the dictionary learning formulation

As seen from (4), the dictionaries with salient and non-salient texture atoms can be learnt separately. This section only focuses on learning the salient dictionary, and we can use the similar way to obtain the non-salient dictionary. According to (4), the salient dictionary can be learnt with the following formulation:

$$\min_{\mathbf{D}',\mathbf{A}'} \frac{1}{n'} \sum_{i=1}^{n'} (\| \mathbf{x}_i' - \mathbf{D}'\boldsymbol{\alpha}_i' \|_2^2 + \lambda \| \boldsymbol{\alpha}_i' \|_1 - \eta \| \boldsymbol{l}^T\mathbf{D}' \|_2^2). \tag{5}$$

To solve (5), the OSDL algorithm is proposed, based on online dictionary learning method [20], due to its fast speed and warm restart mechanism.

Specifically, the optimization problem in (5) is normally divided into two sub-problems: sparse representation and dictionary updating. That is, once dictionary $\mathbf{D}'$ is fixed, $\mathbf{A}' = \{\boldsymbol{\alpha}_i'\}_{i=1}^k$ can be obtained through sparse representation for the first step. At the second step, given $\mathbf{A}'$, $\mathbf{D}'$ can be solved by dictionary updating. The above two steps are iterated until convergence.

Sparse representation: Assume that at the t-th iteration, $\mathbf{x}_t'$ is the image patch randomly selected from the training set of fixation patches. Sparse representation is conducted to obtain sparse coefficients $\boldsymbol{\alpha}_t'$ of $\mathbf{x}_t'$. Since $\eta \| \boldsymbol{l}^T\mathbf{D}' \|_2^2$ in (5) is independent of $\boldsymbol{\alpha}_t'$, the

[2] In this paper, the mean value of each image patch is removed to avoid the impact of pixel intensity on texture analysis.

[3] In this paper, fixation patches mean the training patches attracting several fixations, and non-fixation patches stand for the training patches attracting no fixation.

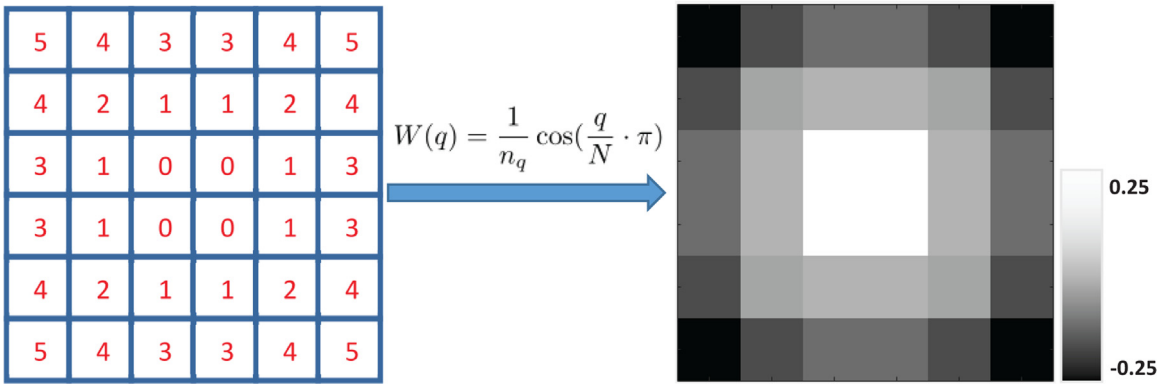

Fig. 2. An example of the center-surround weight function for the 6×6 image patches. In the left figure, the number in each grid is the value of q for the weight function in (3), indicating the q-th Euclidean distance. The right figure shows the weight of each pixel calculated by (3). Note that this example is only an illumination, and the real patch size is much larger, i.e., 96×96 for DOVES and 41×41 for Kienzle et al.

following formulation holds:

$$\boldsymbol{\alpha}_t' \triangleq \underset{\boldsymbol{\alpha}_t' \in \mathbb{R}^k}{\operatorname{argmin}} \| \mathbf{x}_t' - \mathbf{D}_{t-1}'\boldsymbol{\alpha}_t' \|_2^2 + \lambda \| \boldsymbol{\alpha}_t' \|_1, \tag{6}$$

where $\mathbf{D}_{t-1}'$ is the salient dictionary learnt at the last iteration $t-1$. In this paper, LASSO algorithm [43] is utilized for solving (6).

Dictionary updating: After the sparse representation step of the t-th iteration, sparse coefficients $\{\boldsymbol{\alpha}_i'\}_{i=1}^t$ for fixation image patches $\{\mathbf{x}_i'\}_{i=1}^t$ are obtained. Given $\boldsymbol{\alpha}_t'$, the dictionary needs to be updated at the t-th iteration with the following optimization function according to (5),

$$\mathbf{D}_t' \triangleq \underset{\mathbf{D}_t' \in \mathbb{R}^{m\times k}}{\operatorname{argmin}} \frac{1}{t}\sum_{i=1}^{t} (\| \mathbf{x}_i' - \mathbf{D}_t'\boldsymbol{\alpha}_i' \|_2^2 + \lambda \| \boldsymbol{\alpha}_i' \|_1 - \eta \| \boldsymbol{l}^T\mathbf{D}_t' \|_2^2). \tag{7}$$

$\mathbf{D}_t'$ is the salient dictionary learnt at the t-th iteration. To solve (7), we use the block-coordinate descent [20] to update each atom of the dictionary as follows,

$$\mathbf{d}_{j,t}' = \mathbf{d}_{j,t-1}' - \frac{\gamma}{t}\frac{\partial}{\partial \mathbf{d}_j'}\sum_{i=1}^{t} (\| \mathbf{x}_i' - \tilde{\mathbf{D}}_{j,t}'\boldsymbol{\alpha}_i' \|_2^2 - \eta \| \boldsymbol{l}^T\tilde{\mathbf{D}}_{j,t}' \|_2^2)_{|\mathbf{d}_{j,t-1}'}, \tag{8}$$

where

$$\tilde{\mathbf{D}}_{j,t}' = [\mathbf{d}_{1,t}', \ldots, \mathbf{d}_{j-1,t}', \mathbf{d}_j', \mathbf{d}_{j+1,t-1}', \ldots, \mathbf{d}_{k,t-1}'].$$

In (8), $\mathbf{d}_{j,t}'$ refers to the j-th atom of the dictionary at the t-th iteration, and γ is the learning rate of gradient descent. Note that dictionary $\mathbf{D}_t'$ is updated for the t-th iteration, once all atoms $\{\mathbf{d}_{j,t}'\}_{j=1}^k$ are renewed in left–right order. Note that in $\tilde{\mathbf{D}}_{j,t}'$ only $\mathbf{d}_j'$ is the variable to be updated, whereas $\{\mathbf{d}_{1,t}', \ldots, \mathbf{d}_{j-1,t}'\}$ have been updated in the current iteration and $\{\mathbf{d}_{j+1,t-1}', \ldots, \mathbf{d}_{k,t-1}'\}$ have been updated in the $(t-1)$-th iteration. According to Appendix, (8) can be rewritten as

$$\mathbf{d}_{j,t}' = \mathbf{d}_{j,t-1}' + \frac{2\gamma}{t}(\mathbf{c}_{j,t}' - \mathbf{D}_{j,t}'\mathbf{b}_{j,t}') + 2\gamma\eta\boldsymbol{l}\boldsymbol{l}^T\mathbf{d}_{j,t-1}'. \tag{9}$$

Note that, compared with $\tilde{\mathbf{D}}_{j,t}'$, $\mathbf{D}_{j,t}'$ is the matrix where the variable $\mathbf{d}_j'$ is replaced by $\mathbf{d}_{j,t-1}'$.

In (9), $\mathbf{b}_{j,t}'$ and $\mathbf{c}_{j,t}'$ are the j-th columns of $\mathbf{B}_t'$ and $\mathbf{C}_t'$, which are the matrices storing all information of sparse coefficients and image patches from the previous iterations (i.e., from iteration 1 to t). Here, $\mathbf{B}_t'$ and $\mathbf{C}_t'$ are defined as

$$\begin{aligned}\mathbf{B}_t' &= \sum_{i=1}^{t}\boldsymbol{\alpha}_i'\boldsymbol{\alpha}_i'^T = \mathbf{B}_{t-1}' + \boldsymbol{\alpha}_t'\boldsymbol{\alpha}_t'^T,\\ \mathbf{C}_t' &= \sum_{i=1}^{t}\mathbf{x}_i'\boldsymbol{\alpha}_i'^T = \mathbf{C}_{t-1}' + \mathbf{x}_t'\boldsymbol{\alpha}_t'^T.\end{aligned} \tag{10}$$

For achieving the warm restart mechanism, $2\gamma/t$ can be approximatively replaced by $1/\mathbf{B}_t'(j,j)$, where $\mathbf{B}_t'(j,j)$ is the j-th diagonal element of $\mathbf{B}_t'$. Dictionary $\mathbf{D}_t'$ is updated for the t-th iteration, once all atoms $\{\mathbf{d}_{j,t}'\}_{j=1}^k$ are renewed from left to right. The overall procedure of our OSDL algorithm is summarized in Table 1.

4. Saliency detection with the features regarding learnt texture dictionaries

4.1. The SR-LTA feature

For detecting saliency, the SR-LTA can be used as a feature channel. When calculating the SR-LTA feature for a pixel, the image patch with this pixel as the center needs to be extracted. Then, the extracted patch $\mathbf{x} \in \mathbb{R}^m$ is represented sparsely by $\mathbf{D}'$ and $\mathbf{D}''$, respectively. As such, the reconstruction errors of sparse representation regarding $\mathbf{D}'$ and $\mathbf{D}''$ are obtained. Afterwards, the difference between reconstruction errors of $\mathbf{D}''\boldsymbol{\alpha}''$ and $\mathbf{D}'\boldsymbol{\alpha}'$ for an image patch is denoted as r and computed by

Table 1
The summary of online salient dictionary learning (OSDL) algorithm.

- **Input:** The training set of fixation patches $\mathbf{X}' = \{\mathbf{x}_i'\}_{i=1}^n$.
- **Output:** The learnt dictionary $\mathbf{D}'$ with salient textures atoms.
- Set $\mathbf{B}_0' \in \mathbb{R}^{k\times k}$ and $\mathbf{C}_0' \in \mathbb{R}^{m\times k}$ to be zero matrices.
- Initialize $\mathbf{D}_0'$ with the randomly selected fixation patches from the training set.
- **For:** $t=1$ to T
  1. Select an image patch $\mathbf{x}_t'$ randomly from training set $\mathbf{X}'$.
  2. Obtain $\boldsymbol{\alpha}_t'$ by solving (6) with LASSO.
  3. Update $\mathbf{B}_t'$ and $\mathbf{C}_t'$ as,
  $$\mathbf{B}_t' = \mathbf{B}_{t-1}' + \boldsymbol{\alpha}_t'\boldsymbol{\alpha}_t'^T,$$
  $$\mathbf{C}_t' = \mathbf{C}_{t-1}' + \mathbf{x}_t'\boldsymbol{\alpha}_t'^T.$$
  4. Update each atom of the dictionary as follows,
     - **For:** $j=1$ to k
     $$\mathbf{d}_{j,t}' = \mathbf{d}_{j,t-1}' + \frac{1}{\mathbf{B}_t'(j,j)}(\mathbf{c}_{j,t}' - \tilde{\mathbf{D}}_{j,t}'\mathbf{b}_{j,t}') + 2\gamma\eta\boldsymbol{l}\boldsymbol{l}^T\mathbf{d}_{j,t-1}'.$$
     - **End for**
  5. Obtain the salient dictionary $\mathbf{D}_t' = [\mathbf{d}_{1,t}', \ldots, \mathbf{d}_{k,t}']$ for the current iteration.
- **End for**
- **Return** learnt dictionary $\mathbf{D}' = \mathbf{D}_T'$.

$$r = \min_{\alpha''} \| \mathbf{x} - \mathbf{D}''\boldsymbol{\alpha}'' \|_2^2 - \min_{\alpha'} \| \mathbf{x} - \mathbf{D}'\boldsymbol{\alpha}' \|_2^2 \quad (11)$$

where $\boldsymbol{\alpha}''$ and $\boldsymbol{\alpha}'$ are the sparse coefficients of $\mathbf{x}$ with respect to $\mathbf{D}''$ and $\mathbf{D}'$, respectively. Note that a large value of r indicates the image patch is "close" to saliency texture atoms and "far" from non-salient texture atoms.

It has been shown in Fig. 3(a) that the human fixation map tends to be sparse, in which the saliency of major pixels is around zero. It is due to the fact that human visual attention consistently focuses on small regions. However, as can be seen from Fig. 3(b), the dynamic range of r for the conspicuity map generated by (11) is far from that of ground-truth human fixations. Hence, we introduce an exponential function into the SR-LTA feature: $f_1 = r^{\tau}$,

where f_1 is the final pixel-wise output for the SR-LTA feature channel. Moreover, τ is a parameter for adjusting the dynamic range of f_1 to cater for the real distribution of fixations. Here, a large value of τ is required for a more sparse distribution of conspicuity values. For example, as seen from Fig. 3(c), $\tau = 5.6$ makes the distribution of the conspicuity values of f_1 approaching to the ground-truth human fixation map. Accordingly, τ is set to 5.6 in our experiments in Section 5. Finally, the pixel-wise SR-LTA feature $\mathbf{f}_1$ for an image can be achieved by computing f_1 of all pixels.

4.2. Saliency detection with SR-LTA feature

Now, we focus on the saliency detection by combining the SR-LTA feature with other two features. In [8], it has been pointed out that the luminance is an important factor on attracting human attention. However, the luminance is not considered in dictionary learning for SR-LTA. Therefore, the luminance feature is included in our method. Besides, our saliency detection method also takes the contrast feature into account, the same as [21]. Specifically, luminance is a low level feature of the image, indicating how "bright" each pixel is. According to the study of the HVS, [21] defines the luminance as a weighted average of grayscale intensity for each image patch, using a circular raised cosine weighting function from the patch center. Contrast is the root-mean-square contrast within the image patch, which is also weighted by circular raised cosine function. High contrast value means that the center of the image patch is outstanding from the whole patch in intensity. In brief, luminance contains the grayscale intensity, while contrast implies the luminance difference of the image patch. For more details about the computation on features of luminance and contrast, refer to [21]. Then, final saliency map $\mathbf{S}$ can be computed by

$$\mathbf{S} = \sum_{p=1}^{3} \omega_p \mathcal{N}(\mathbf{f}_p), \quad (12)$$

where $\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3\}$ indicate three low level features: our SR-LTA feature, luminance, and contrast, with $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$ being their corresponding weights. $\mathcal{N}(\cdot)$ is the normalization operator. Note that our method can only work on the gray images, since color information is not considered.

Next, the remaining task for saliency detection on a gray image by (12) is to work out the weight of each feature channel. In fact, larger weight should be assigned to the feature channel, of which conspicuity map is more close to the human fixation map. Let $\mathbf{v}_s$ be the vectorized human fixation map of a training image. Given $\mathbf{v}_s$ of all training images, the optimal weights $\boldsymbol{\omega}$ can be obtained by solving the following ℓ_2-norm optimization with least square estimation,

$$\arg\ \min_{\omega} \sum_{s} \| \mathbf{U}_s\boldsymbol{\omega} - \mathbf{v}_s \|_2 \quad \text{s. t. } \boldsymbol{\omega} \in (0, 1),\ \sum_{p=1}^{3} \omega_p = 1. \quad (13)$$

In (13), $\mathbf{U}_s$ is the matrix of conspicuity maps for each training

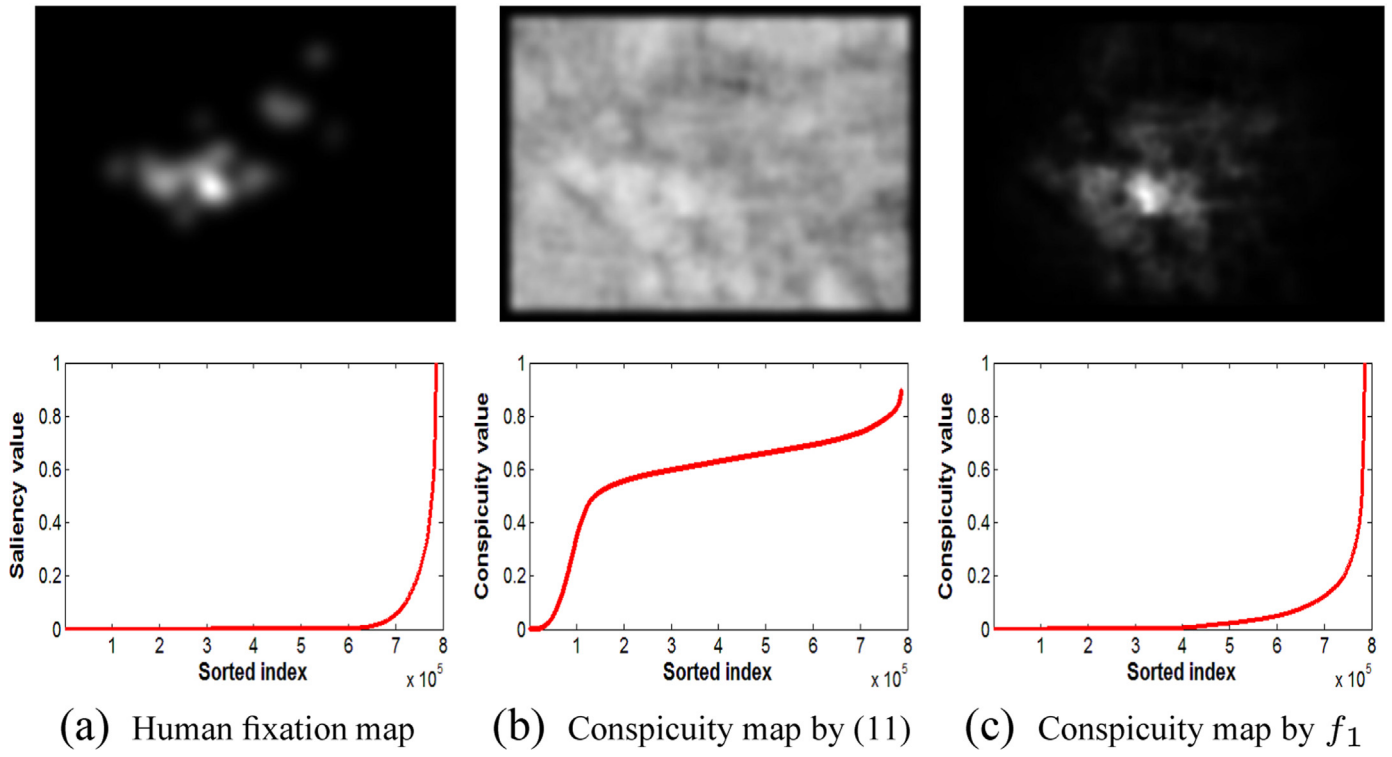

Fig. 3. An example of distributions of human fixation map, conspicuity map by (11), and conspicuity map by f_1. In the first row, (a), (b), and (c) are the maps, generated by human fixations, the sparse representation errors in (11), and the exponential function f_1 of sparse representation errors. In second row, (a), (b), and (c) are the distribution of weights in the corresponding maps, with pixels sorted in ascending order of weights in the map. Note that the exponent τ is set to be 5.6 according to the distribution of values of human fixation map in (a).

image, in which each column denotes the conspicuity map of one feature channel, among SR-LTA, luminance, and contrast features. For solving the least square estimation of (13), the disciplined convex programming approach [10] is applied in our method. Finally, the optimal weights corresponding to different feature channels can be worked out with least square fitting to the human fixations (Fig. 4).

5. Experimental results

In this section, the experimental results are presented to evaluate the proposed method for saliency detection on gray images from two eye tracking databases: DOVES [39] and Kienzle et al. [18]. In Section 5.1, we introduce the databases, training data and parameter settings in our experiments. In Section 5.2, we compare the saliency detection results of our and other 9 methods. In Section 5.3, we further analyze the performance of our method, from the aspects of feature effectiveness, learning performance and robustness.

5.1. Experimental setup

5.1.1. Database

Since this paper mainly concentrates on the low level texture feature for saliency detection, only gray natural images were tested in our experiments. Here, the databases of DOVES [39] and Kienzle et al. [18], which provide the eye tracking data over gray images, were utilized for both training and test tasks of our experiments. Note that the DOVES database includes natural images with few semantic objects, whereas the Kienzle et al. database contains the images with some semantic objects. Here, both the training and test processes were conducted for each database individually. In Table 2, we list the key properties of these two databases. For more details, refer to [39] and [18].

5.1.2. Training patches

For each database, we divided the images into training and test sets. For the training set, 78 and 150 images were randomly chosen from DOVES and Kienzle et al. databases, respectively. The remaining 23 and 50 images in these two databases were used for the test, to evaluate the performance of saliency detection methods. Here, the three rounds of cross validation were applied in each database for the evaluation in our experiments. Next, in our method, about 5% patches with top fixation density were picked out from the training set of each database to learn the salient dictionaries. The same amount of patches, in which the fixation numbers rank bottom 5%, were picked out for learning non-salient dictionaries. To eliminate the influence of location on eye tracking data, each non-fixation patch is extracted in the same location as selected fixation patch, but from different images. It is worth pointing out that the patch sizes were 96×96 for DOVES and 41×41 for Kienzle et al., to ensure that the corresponding fovea degrees are around 1.5° in each database. In addition, all training patches from the two databases were down-sampled to be 16×16, such that the pixel number m of learnt dictionary atoms is 256.

Table 2
Details of two databases used in our experiments.

	DOVES	*Kienzle et al.*
Images	101	200
Human observers	29	14
Image size in pixel	1024×768	1024×768
Image size in visual angle	$17° \times 13°$	$36° \times 27°$
Total fixations	30,000+	18,000+

5.1.3. Parameter setting

All parameters related to our experiments are summarized in Table 3. For dictionary learning with our OSDL algorithm, according to the empirical settings in [20], the number of atoms k of each dictionary was set to 4 m, and the regularization parameter λ in (4) for the tradeoff between reconstruction error and sparsity was set to $1.2/\sqrt{m}$. In addition, parameter η in (4) was tuned to 0.05 (to be discussed in Section 5.3), and the learning rate γ in (9) was set to 0.05 in our experiments, to make the results appropriate. For saliency detection, as verified in Section 4.1, power parameter τ in exponential function f_1 was chosen to be 5.6, such that the distribution of saliency detected by our method is similar to that of human fixation map. Moreover, the weights corresponding to different features were learnt using (13) for both DOVES and Kienzle et al. databases, and the final weights are {0.72, 0.09, 0.19}.

5.2. Comparison

From now on, we present the saliency detection results of our method, compared with other 9 state-of-the-art methods, including BMS [13], Itti et al.'s method [8], Duan et al.'s method [12], GAFFE [21], Hou et al.'s method [30], Zhao et al.'s method [16], Judd et al.'s method [15], AWS [34], and SALICON [23]. The same center bias mask [12] was employed in our and all other methods, since it has been pointed out [15] that the center bias is able to make saliency detection more precise according to the HVS. Here, the accuracy of saliency detection is evaluated using the metrics of

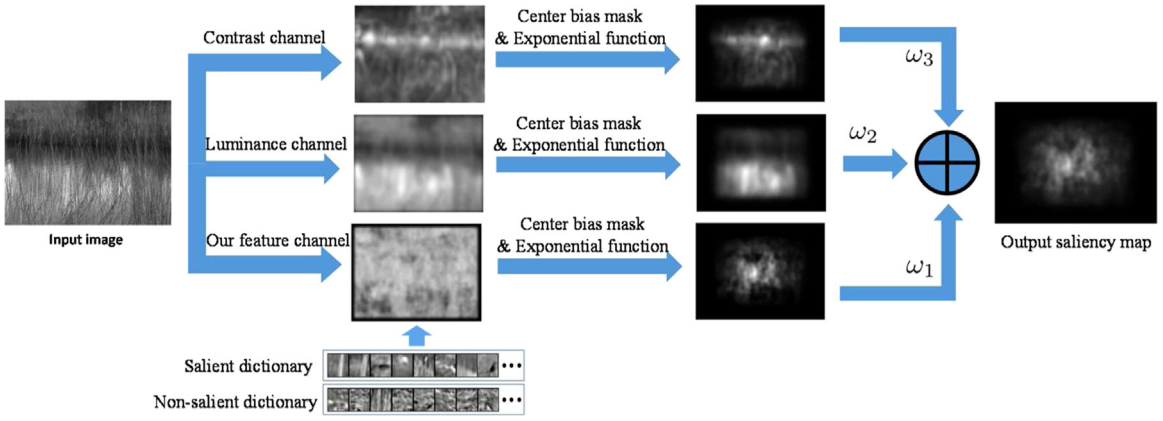

Fig. 4. Procedure of our saliency detection method. The input image is processed through three channels, including our SR-LTA feature, contrast, and luminance, to obtain three conspicuity maps. Note that the conspicuity map of our SR-LTA feature channel is obtained through two learnt dictionaries. Then, a center bias mask [12] and an exponential function are processed on these maps to make saliency detection more reasonable. The weighted sum of those three maps makes up the final saliency map.

Table 3
The setting of parameters for our method.

Dictionary learning	Dictionary atom size m	256 pixels
	Atoms number in a dictionary k	1024
	Regularization parameter λ	0.075
	Regularization parameter η	0.05
	Learning rate γ	0.05
Saliency detection	Power parameter τ	5.6
	Combination weights $\{\omega_p\}_{p=1}^{3}$	{0.72, 0.09, 0.19}

receiver operator characteristics (ROC), area under the ROC curve (AUC), normalized scan-path saliency (NSS), linear correlation coefficient (CC), and chi-square distance (χ^2 distance). Besides, the computational cost is also compared in this section.

5.2.1. ROC curves and AUC

As a metric of detection accuracy, the ROC curve [44] is plotted as false positive rate (FPR) versus true positive rate (TPR) at various detection thresholds. Here, ROC curve is applied to show how well the detected saliency map predicts human fixations. Specifically, in a saliency map each pixel is assigned with a saliency value, ranging from 0 to 1. If the saliency value of a pixel is greater than a predefined threshold, it is seen as the predicted positive sample. Otherwise, it is seen as the negative sample. By varying the threshold from 1 to 0, the ROC curves can be plotted with different pairs of FPR and TPR, which both increase from 0 to 1. Note that FPR indicates the proportion of incorrectly predicted fixations among all ground-truth non-fixations, and TPR means the ratio of correctly predicted fixations among all ground-truth fixations. Therefore, a large TPR with small FPR implies more accurate saliency detection. Furthermore, AUC defines the area under the ROC curve, which is calculated to quantify the ROC curve. Obviously, a larger value of AUC means a better result for saliency detection.

In Fig. 5, we show the ROC curves of saliency detection by our and other 9 methods, averaged over all test images, for each database. Meanwhile, Table 4 tabulates the AUC results of our and other 9 methods. It can be seen from Fig. 5 and Table 4 that in comparison with all 9 methods, our method offers the better AUC results on detecting saliency of test images from the DOVES database, which rarely contains semantic objects. However, the performance of our method is inferior to the latest SALICON [23] for the Kienzle et al. database, where some semantic objects are included in the images. The superior performance of [23] is mainly due to the top-down semantic features learnt by the deep neural networks. In a word, our method provides effective low level features for bottom-up saliency detection.

5.2.2. NSS and CC

Next, we move to the comparison of NSS and CC metrics for a more comprehensive evaluation. For evaluating the accuracy of saliency detection, NSS is computed to quantify the relevance between fixation locations and saliency prediction. To be more specific, the NSS score [45] is averaged over the normalized saliency value of all human fixations:

$$\mathrm{NSS} = \frac{1}{\sigma_s m}\sum_{k=1}^{m}\left(\mathbf{S}(x_k, y_k) - \mu_s\right), \tag{14}$$

where μ_s and σ_s are the mean and the standard deviation of the saliency map, respectively. $\mathbf{S}(x_k, y_k)$ indicates the saliency value at fixation (x_k, y_k), and m is the total number of human fixations. CC is another popular metric to measure the linear correlation between human fixation map $\mathbf{H}(x, y)$ (convolved with Gaussian window) and saliency map $\mathbf{S}(x, y)$, where (x,y) is the location of each pixel in the image. It can be calculated by

$$\mathrm{CC} = \frac{\sum_{x,y}\left(\mathbf{H}(x, y) - \mu_h\right)\cdot\left(\mathbf{S}(x, y) - \mu_s\right)}{\sqrt{\sigma_h^2\sigma_s^2}}, \tag{15}$$

where μ_h and σ_h represent the mean and standard deviation of the human fixation map $\mathbf{H}$, while μ_s and σ_s denote the mean and standard deviation of predicted saliency map $\mathbf{S}$. Note that a larger value of NSS or CC indicates more accurate saliency detection.

The NSS and CC results, averaged over all test images of each database, are also listed in Table 4. Again, it can be found in this table that our method is significantly superior to all other 8 methods (excluding the SALICON [23]), in terms of both NSS and CC metrics. Compared with [23], our method is much better in the DOVE database, while it performs a little worse in the Kienzle et al. database. It implies that our method is capable of bottom-up saliency detection due to the learnt low level features. By contrast, [23] is able to provide high level features for top-down saliency detection. However, the standard deviations of [23] are rather high, since it does not perform well for images with few semantic object in the Kienzle et al. database. Such a reason is to be verified in the following subjective evaluation. It is worth pointing out that the high level features of [23] are learnt from hundreds of images (from exterior databases) with the deep neural networks.

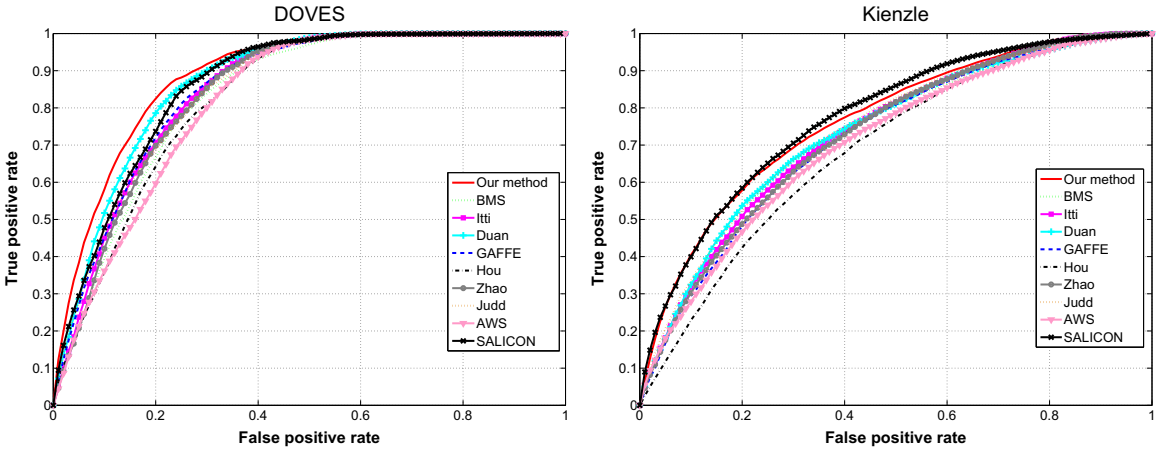

Fig. 5. The ROC curves of saliency detection by our and other 9 methods over two databases, respectively.

Table 4
The saliency detection results on test images of two databases.

Metrics	*DOVES*				*Kienzle et al.*				Time
	AUC	NSS	CC	χ^2	AUC	NSS	CC	χ^2	cost(s)
Our	**0.886** ± .028	**1.961** ± .365	**0.582** ± .086	**0.571** ± .057	0.766 ± .065	1.187 ± .456	0.488 ± .139	0.602 ± .057	0.24
SR-LTA	0.875 ± **.027**	1.815 ± .358	0.575 ± .089	0.580 ± .054	0.765 ± .064	1.199 ± .512	0.491 ± .146	0.618 ± .061	–
BMS	0.834 ± .057	1.274 ± .374	0.383 ± .112	0.692 ± .070	0.728 ± .093	0.887 ± .471	0.364 ± .170	0.679 ± .057	0.46
Itti	0.850 ± .036	1.331 ± **.234**	0.414 ± **.077**	0.702 ± **.051**	0.734 ± .068	0.865 ± **.277**	0.364 ± **.110**	0.688 ± **.052**	**0.16**
Duan	0.870 ± .043	1.463 ± .272	0.448 ± .093	0.684 ± .057	0.735 ± .080	0.899 ± .346	0.387 ± .142	0.682 ± .056	1.53
GAFFE	0.852 ± .050	1.404 ± .313	0.432 ± .102	0.689 ± .060	0.721 ± .076	0.831 ± .333	0.357 ± .139	0.683 ± .056	7.65
Hou	0.828 ± .061	1.213 ± .343	0.382 ± .124	0.704 ± .058	0.690 ± .095	0.630 ± .388	0.286 ± .179	0.698 ± .062	0.31
Zhao	0.843 ± .052	1.308 ± .385	0.407 ± .123	0.700 ± **.051**	0.727 ± **.062**	0.860 ± .288	0.359 ± .113	0.685 ± **.052**	0.29
Judd	0.849 ± .058	1.438 ± .415	0.439 ± .133	0.683 ± .064	0.741 ± .082	0.981 ± .491	0.399 ± .143	0.670 ± .053	13.34
AWS	0.822 ± .034	1.183 ± .248	0.363 ± .079	0.705 ± .066	0.709 ± .092	0.825 ± .512	0.337 ± .174	0.690 ± .057	4.37
SALICON	0.863 ± .042	1.613 ± .489	0.484 ± .120	0.684 ± .057	**0.775** ± .075	**1.327** ± 1.139	**0.510** ± .170	**0.599** ± .070	–

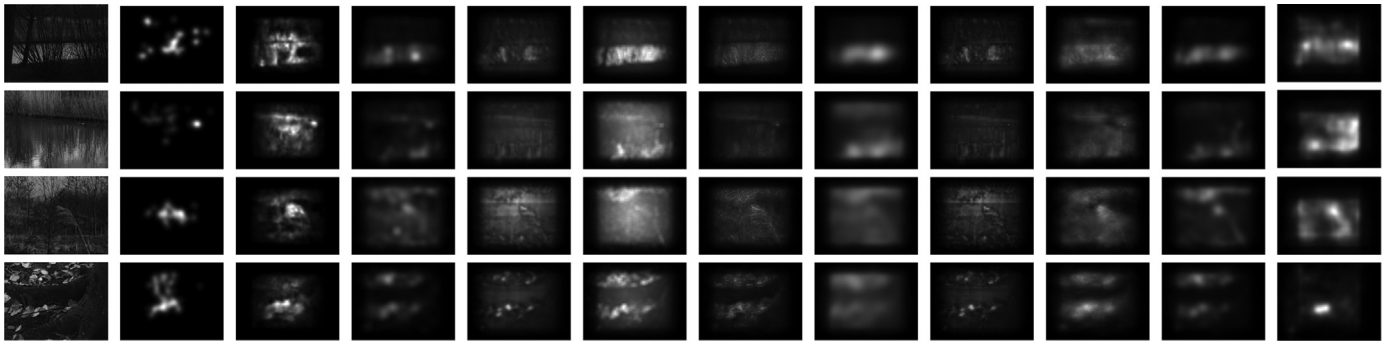

Fig. 6. Saliency maps of four test images from DOVES database, output by our and other 9 methods as well human fixation map. From left to right: Input images, human fixation maps, our, BMS, Itti, Duan, GAFFE, Hou, Zhao, Judd, AWS and SALICON methods.

5.2.3. χ^2 distance

χ^2 distance statistically measures the difference between the expected distribution and observed distribution, via the χ^2 test. We see normalized human fixation map $\tilde{\mathbf{H}}(x, y)$ as the expected distribution, and saliency map $\tilde{\mathbf{S}}(x, y)$ as the observed distribution. Then, the χ^2 distance is defined as

$$\chi^2 = \frac{1}{2}\sum_{x,y}\frac{\left(\tilde{\mathbf{H}}(x, y) - \tilde{\mathbf{S}}(x, y)\right)^2}{\tilde{\mathbf{H}}(x, y) + \tilde{\mathbf{S}}(x, y)}. \tag{16}$$

Obviously, less χ^2 distance means better saliency detection. We tabulate in Table 4 the χ^2 distances of our and other methods. Again, our method generally performs the best for bottom-up saliency detection.

5.2.4. Saliency maps

At last, we show in Figs. 6 and 7 the saliency maps of several randomly selected test images, detected by our and other 9 methods as well as the human fixation map. From these figures, we can see that in comparison with other methods, our method is capable of well locating the saliency regions, much closer to the maps of human fixations. We can further see from Fig. 7 that [23] yields almost perfect saliency maps for images with semantic objects (e.g., the second and last images). Nevertheless, it fails to generate accurate saliency maps for images without any semantic object (e.g., the first and seventh images). On the other hand, the subjective results here show the great performance of our method in bottom-up saliency detection.

5.2.5. Computational time

The average running time (seconds per image) of our and other saliency detection methods is listed in Table 4. Note that the codes of all methods were run on Matlab 2012b at a computer with Intel Core i7-4770 CPU@3.4 GHz and 16 GB RAM. As seen from Table 4, the computational time of our saliency detection method ranks second among all methods, except SALICON.[4] This verifies that our method performs well in time efficiency.

5.3. Performance analysis

5.3.1. Analysis on feature effectiveness

Our method is based on a novel low level feature SR-LTA, and it is important to evaluate its benefit to the improvement of saliency detection accuracy. To this end, in our method we set the weight of the SR-LTA channel to one, and the weights of other channels to zero. Then, we report the results in the second row of Table 4. As seen from this table, the accuracy of saliency detection by the SR-LTA feature is better than other methods, in terms of AUC, NSS, and CC. More interestingly, it even slightly outperforms our method in AUC and CC values for the Kienzle database. Besides, the saliency detection accuracy of each single feature is also shown in Fig. 9 for further comparison. We can find that the proposed SR-LTA feature performs much better than two conventional features, i.e., luminance and contrast. Thus, the effectiveness of the proposed SR-LTA feature can be verified.

The center-surround term is newly added in our formulation (4) for the SR-LTA feature. Thereby, it is worth analyzing the impact of the proposed center-surround term on saliency detection.

[4] There is only a demo of SALICON on the website [46] but without any source code. Thus, the computational time of SALICON is not included in the comparison.

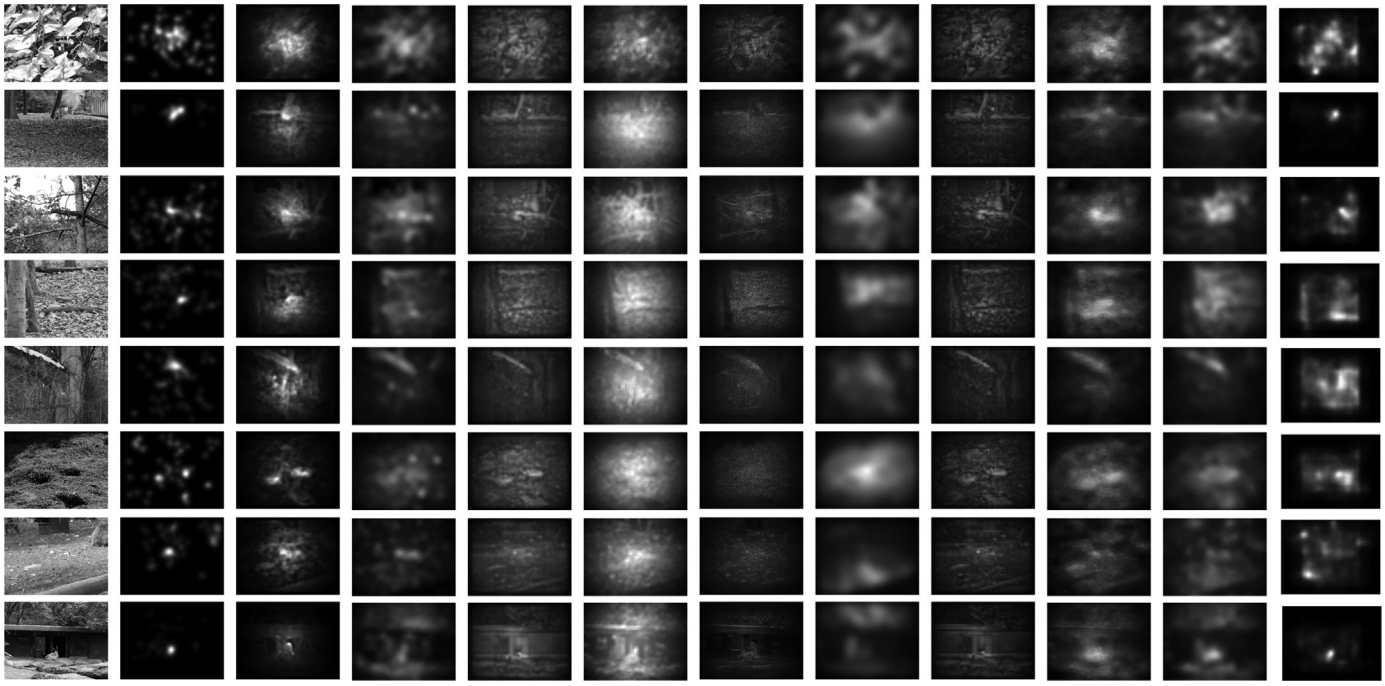

Fig. 7. Saliency maps of eight test images from Kienzle et al. database, output by our and other 9 methods as well human fixations. From left to right: Input images, human fixation maps, our, BMS, Itti, Duan, GAFFE, Hou, Zhao, Judd, AWS and SALICON methods.

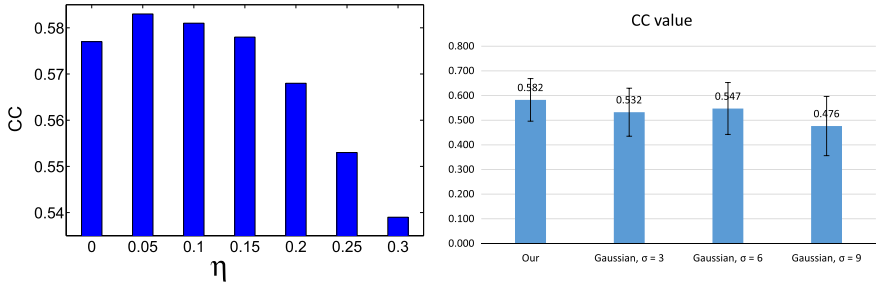

Fig. 8. Left: Saliency detection accuracy of our method alongside increased η. Right: Saliency detection accuracy of our methods using different weight functions. The accuracy of saliency detection is measured by CC averaged over the DOVES database.

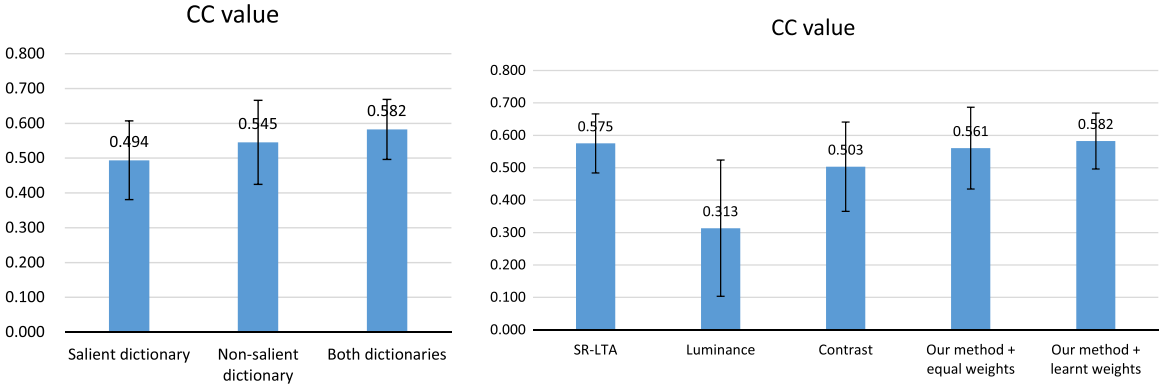

Fig. 9. Left: Saliency detection accuracy of our method using only salient or non-salient dictionary against using both dictionaries. Right: Performance of our method with learnt or equal weights, as well as three single features. Note that the accuracy of saliency detection is measured by CC averaged over the DOVES database.

We can see from (4) that parameter η controls the trade-off between the center-surround term and sparse representation term. Fig. 8 demonstrates the performance of our method at various η. As observed from this figure, $\eta=0.05$ makes the saliency detection accuracy highest. This figure thus guides us that parameter η should be set to 0.05, and it also implies the positive effect of the center-surround term on improving saliency detection accuracy in our method. In addition, the center-surround term is based on the

Table 5
The saliency detection accuracy of test images with different noise levels and different intensity offsets on DOVES database, using our saliency detection method.

Noise levels (σ_n)	0	10	20	30	50	Intensity offsets (Δ_I)	−20%	−10%	0	10%	20%
CC	0.582	0.577	0.564	0.563	0.542	CC	0.582	0.583	0.582	0.582	0.583

weight function proposed in (3). Hence, we further validate in Fig. 8 the effectiveness of the proposed center-surround weight function, by comparing to other Gaussian functions with different standard deviations σ. We can see from this figure that the proposed weight function is effective in modeling the center-surround term for bottom-up saliency detection, when compared with Gaussian weight functions.

5.3.2. Analysis on learning performance

Now, we focus on the learning performance of our method from two aspects. First, our OSDL algorithm includes the learning of both salient and non-salient dictionaries in formulation (11). Accordingly, Fig. 9 shows the performance of our method with single salient/non-salient dictionary and with both two dictionaries. We can see from this figure that the combination of salient and non-salient dictionaries performs better than each single dictionary for saliency detection. Second, our method learns to integrate all features together. Thus, we investigate the effectiveness of learnt weights obtained by (13) in feature integration. To this end, we plot in Fig. 9 the saliency detection accuracy (in term of CC) of our method with learnt weights and with equal weights. We can see from this figure that CC increases from 0.561 to 0.582, when using the learnt weights instead of equal weights in our method. Thus, the effectiveness of learnt weights can be validated.

5.3.3. Analysis on robustness

Finally, we analyze the robustness of our method to image noise and intensity offsets. Here, we follow the basic idea of [47–49] to investigate the impact of image noise and intensity offsets on saliency detection accuracy of our method. The results are reported in Table 5. We can see from this table that the CC results of our saliency detection method decrease along with increased noise. However, even when the Gaussian noise is large ($\sigma_n = 50$), our method has higher CC value (0.542) than other saliency detection methods (the best one is 0.484 as seen in Table 4). Moreover, CC values are almost unchanged in our method, once the illumination varies in test images. In a word, our method is robust to image noise and intensity offsets. The investigation on the robustness of our method to other image effects, e.g., geometric transformations, is a promising future work.

6. Conclusions

In this paper, we have proposed a learning based method with a novel feature called SR-LTA, to predict saliency of gray images. In the proposed method, an optimization formulation with a center-surround term was proposed, for learning both salient and non-salient dictionaries from the training fixation and non-fixation patches. Then, the OSDL algorithm was developed to solve the proposed optimization formulation, in light of online dictionary learning. Here, the two learnt dictionaries are discriminative to classify the salient and non-salient regions. Thus, the SR-LTA feature can be computed in our method, upon the difference between sparse representation errors with respect to the salient and non-salient dictionaries. At last, the saliency map of an input gray image can be generated, via linearly combining conspicuity maps of the proposed SR-LTA feature and two other conventional features (luminance and contrast). For the linear combination, the weight of each feature channel is determined by the least square fitting on training data. Experimental results show that our method advances the state-of-the-art bottom-up saliency detection.

Our work in the current form only focuses on saliency detection of gray images. Thus, the future work should incorporate the SR-LTA feature of our method into saliency detection of color images. Moreover, the SR-LTA feature proposed in our method can be seen as a bottom-up feature for saliency detection. Indeed, an ideal saliency detection system, like the one of the HVS, requires the combination of both bottom-up and top-down information flow. Thus, the protocols, for integrating bottom-up and top-down processes in saliency detection, show a promising research trend in future.

Conflict of interest

None declared.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC), China projects under Grants 61573037, 61202139, and 61471022, and the China 973 Program under Grant 2013CB329006. This work was also supported by Fok Ying Tung Education Foundation under grant 151061.

Appendix

Derivation from (8) to (9) in Section 3.2.

At first, the ℓ_2 norm of (8) can be rewritten in the form of matrix trace denoted as $\mathrm{tr}(\cdot)$. That is

$$\begin{aligned}
&\sum_{i=1}^{t} (\| \mathbf{x}_i' - \tilde{\mathbf{D}}'_{j,t}\boldsymbol{\alpha}_i' \|_2^2 - \eta \| \boldsymbol{l}^T \tilde{\mathbf{D}}'_{j,t} \|_2^2) \\
&= \sum_{i=1}^{t} \left(\mathrm{tr}(\mathbf{x}_i' - \tilde{\mathbf{D}}'_{j,t}\boldsymbol{\alpha}_i')(\mathbf{x}_i' - \tilde{\mathbf{D}}'_{j,t}\boldsymbol{\alpha}_i')^T - \mathrm{tr}(\eta \boldsymbol{l}^T \tilde{\mathbf{D}}'_{j,t} \tilde{\mathbf{D}}'^T_{j,t} \boldsymbol{l}) \right) \\
&= \mathrm{tr}\left(\tilde{\mathbf{D}}'_{j,t} \sum_{i=1}^{t} \boldsymbol{\alpha}_i' \boldsymbol{\alpha}_i'^T \tilde{\mathbf{D}}'^T_{j,t} \right) - \mathrm{tr}\left(2\tilde{\mathbf{D}}'^T_{j,t} \sum_{i=1}^{t} \mathbf{x}_i' \boldsymbol{\alpha}_i'^T \right) + \mathrm{tr}\left(\sum_{i=1}^{t} \mathbf{x}_i' \mathbf{x}_i'^T \right) \\
&\quad - \mathrm{tr}(t\eta \boldsymbol{l}^T \tilde{\mathbf{D}}'_{j,t} \tilde{\mathbf{D}}'^T_{j,t} \boldsymbol{l}),
\end{aligned} \tag{17}$$

where

$$\tilde{\mathbf{D}}'_{j,t} = [\mathbf{d}'_{1,t}, \ldots, \mathbf{d}'_{j-1,t}, \mathbf{d}'_j, \mathbf{d}'_{j+1,t-1}, \ldots, \mathbf{d}'_{k,t-1}].$$

Note that only the j-th atom $\mathbf{d}'_j$ in $\tilde{\mathbf{D}}'_{j,t}$ is variable.

As $\mathbf{d}'_j$ is independent of $\mathbf{x}_i'$, the derivative of (17) can be written as

$$\frac{\partial}{\partial \mathbf{d}'_j}\sum_{i=1}^{t}(\| \mathbf{x}'_i - \tilde{\mathbf{D}}'_{j,t}\boldsymbol{\alpha}'_i \|_2^2 - \eta \| \boldsymbol{l}^T\tilde{\mathbf{D}}'_{j,t} \|_2^2)$$
$$= \frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(\tilde{\mathbf{D}}'_{j,t}\mathbf{B}'_t\tilde{\mathbf{D}}'^T_{j,t}) - \frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(2\tilde{\mathbf{D}}'^T_{j,t}\mathbf{C}'_t)$$
$$- \frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(t\eta\boldsymbol{l}^T\tilde{\mathbf{D}}'_{j,t}\tilde{\mathbf{D}}'^T_{j,t}\boldsymbol{l}), \tag{18}$$

where $\mathbf{B}'_t$ and $\mathbf{C}'_t$ have been defined as $\sum_{i=1}^{t}\boldsymbol{\alpha}'_i\boldsymbol{\alpha}'^T_i$ and $\sum_{i=1}^{t}\mathbf{x}'_i\boldsymbol{\alpha}'^T_i$, respectively.

According to the rules of derivative of matrix traces [50], we can obtain

$$\frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(\tilde{\mathbf{D}}'_{j,t}\mathbf{B}'_t\tilde{\mathbf{D}}'^T_{j,t}) = 2\tilde{\mathbf{D}}'_{j,t}\mathbf{b}'_{j,t}, \tag{19}$$

$$\frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(2\tilde{\mathbf{D}}'^T_{j,t}\mathbf{C}'_t) = 2\mathbf{c}'_{j,t}, \tag{20}$$

and

$$\frac{\partial}{\partial \mathbf{d}'_j}\mathrm{tr}(t\eta\boldsymbol{l}^T\tilde{\mathbf{D}}'_{j,t}\tilde{\mathbf{D}}'^T_{j,t}\boldsymbol{l}) = 2t\eta\boldsymbol{l}\boldsymbol{l}^T\mathbf{d}'_j. \tag{21}$$

In above equations, $\mathbf{b}'_{j,t}$ and $\mathbf{c}'_{j,t}$ are the j-th columns of $\mathbf{B}'_t$ and $\mathbf{C}'_t$.

Based on (19)–(21), the following holds:

$$\frac{\partial}{\partial \mathbf{d}'_j}\sum_{i=1}^{t}(\| \mathbf{x}'_i - \tilde{\mathbf{D}}'_{j,t}\boldsymbol{\alpha}'_i \|_2^2 - \eta \| \boldsymbol{l}^T\tilde{\mathbf{D}}'_{j,t} \|_2^2)$$
$$= 2\tilde{\mathbf{D}}'_{j,t}\mathbf{b}'_{j,t} - 2\mathbf{c}'_{j,t} - 2t\eta\boldsymbol{l}\boldsymbol{l}^T\mathbf{d}'_j. \tag{22}$$

Consequently, (8) can be rewritten by (9). This completes the derivation from (8) to (9) in Section 3.2.

References

[1] L. Huo, L. Jiao, S. Wang, S. Yang, Object-level saliency detection with color attributes, Pattern Recognit. 49 (2016) 162–173.
[2] D. Gao, S. Han, N. Vasconcelos, Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (6) (2009) 989–1005.
[3] M. Rubinstein, D. Gutierrez, O. Sorkine, A. Shamir, A comparative study of image retargeting, ACM Trans. Graph. 29 (6) (2010) 160:01–10.
[4] U. Engelke, H. Kaprykowsky, H. Zepernick, P. Ndjiki-Nya, Visual attention in quality assessment, IEEE Signal Process. Mag. 28 (6) (2011) 50–59.
[5] M. Xu, X. Deng, S. Li, Z. Wang, Region-of-interest based conversational hevc coding with hierarchical perception model of face, IEEE J. Sel. Top. Signal Process. 8 (3) (2014) 475–489.
[6] A.M. Treisman, G. Gelade, A feature-integration theory of attention, Cognit. Psychol. 12 (1) (1980) 97–136.
[7] C. Koch, S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry, Matters Intell. 188 (1987) 115–141.
[8] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[9] N. Bruce, J. Tsotsos, Saliency based on information maximization, in: Advances in neural information processing systems (NIPS), 2005, pp. 155–162.
[10] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Advances in neural information processing systems (NIPS), 2006, pp. 545–552.
[11] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, in: Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[12] L. Duan, C. Wu, J. Miao, L. Qing, Y. Fu, Visual saliency detection by spatially weighted dissimilarity, in: Computer Vision and Pattern Recognition (CVPR), 2011, pp. 473–480.
[13] J. Zhang, S. Sclaroff, Saliency detection: a boolean map approach, in: International Conference on Computer Vision (ICCV), 2013, pp. 153–160.
[14] L. Itti, C. Koch, Feature combination strategies for saliency-based visual attention systems, J. Electron. Imaging 10 (1) (2001) 161–169.
[15] T. Judd, K. Ehinger, F. Durand, A. Torralba, Learning to predict where humans look, in: International Conference on Computer Vision (ICCV), 2009, pp. 2106–2113.
[16] Q. Zhao, C. Koch, Learning a saliency map using fixated locations in natural scenes, J. Vision. 11 (3) (2011) 9.
[17] W. Kienzle, F.A. Wichmann, M.O. Franz, B. Schölkopf, A nonparametric approach to bottom-up visual saliency, in: Advances in Neural Information Processing Systems (NIPS), 2007, pp. 689–696.
[18] W. Kienzle, M.O. Franz, B. Schölkopf, F.A. Wichmann, Center-surround patterns emerge as optimal predictors for human saccade targets, J. Vision 9 (5) (2009) 7.
[19] M. Cerf, J. Harel, W. Einhäuser, C. Koch, Predicting human gaze using low-level saliency combined with face detection, in: In Advances in neural information processing systems (NIPS), 2008, pp. 241–248.
[20] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, in: International Conference on Machine Learning (ICML), 2009, pp. 689–696.
[21] U. Rajashekar, I. Van Der Linde, A.C. Bovik, L.K. Cormack, Gaffe: a gaze-attentive fixation finding engine, IEEE Trans. Image Process. 17 (4) (2008) 564–573.
[22] L. Jiang, M. Xu, Z. Ye, Z. Wang, Image saliency detection with sparse representation of learnt texture atoms, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, pp. 54–62.
[23] X. Huang, C. Shen, X. Boix, Q. Zhao, Salicon: reducing the semantic gap in saliency prediction by adapting deep neural networks, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 262–270.
[24] D. Walther, C. Koch, Modeling attention to salient proto-objects, Neural Netw. 19 (9) (2006) 1395–1407.
[25] D. Pang, A. Kimura, T. Takeuchi, J. Yamato, K. Kashino, A stochastic model of selective visual attention with a dynamic bayesian network, in: Proceedings of the International Conference on Multimedia and Expo (ICME), 2008, pp. 1073–1076.
[26] T. Avraham, M. Lindenbaum, Esaliency (extended saliency): meaningful attention using stochastic image modeling, IEEE Trans. Pattern Anal. Mach. Intell. 32 (4) (2010) 693–708.
[27] Y. Li, Y. Zhou, J. Yan, Z. Niu, J. Yang, Visual saliency based on conditional entropy, in: Asian Conference on Computer Vision (ACCV), 2009, pp. 246–257.
[28] W. Wang, Y. Wang, Q. Huang, W. Gao, Measuring visual saliency by site entropy rate, in: Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2368–2375.
[29] W. Hou, X. Gao, D. Tao, X. Li, Visual saliency detection using information divergence, Pattern Recognit. 46 (10) (2013) 2658–2669.
[30] X. Hou, J. Harel, C. Koch, Image signature: highlighting sparse salient regions, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 194–201.
[31] J. Li, M.D. Levine, X. An, X. Xu, H. He, Visual saliency based on scale-space analysis in the frequency domain, IEEE Trans. Pattern Anal. Mach. Intell. 35 (4) (2013) 996–1010.
[32] J. Yan, M. Zhu, H. Liu, Y. Liu, Visual saliency detection via sparsity pursuit, IEEE Signal Process. Lett. 17 (8) (2010) 739–742.
[33] Z. Ren, S. Gao, L.-T. Chia, D. Rajan, Regularized feature reconstruction for spatio-temporal saliency detection, IEEE Trans. Image Process. 22 (8) (2013) 3120–3132.
[34] A. Garcia-Diaz, V. Leborán, X.R. Fdez-Vidal, X.M. Pardo, On the relationship between optical variability, visual saliency, and eye fixations: a computational approach, J. Vision 12 (6) (2012) 17.
[35] C. Xia, F. Qi, G. Shi, P. Wang, Nonlocal center-surround reconstruction-based bottom-up saliency estimation, Pattern Recognit. 48 (4) (2015) 1337–1348.
[36] V. Navalpakkam, L. Itti, Search goal tunes visual features optimally, Neuron 53 (4) (2007) 605–617.
[37] W. Kienzle, B. Schölkopf, F.A. Wichmann, M.O. Franz, How to find interesting locations in video: a spatiotemporal interest point detector learned from human eye movements, in: Proceedings of the International Conference on Pattern Recognition (ICPR), 2007, pp. 405–414.
[38] M. Kümmerer, L. Theis, M. Bethge, Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet, ICLR workshop, 2014, pp. 1–8.
[39] I. Van Der Linde, U. Rajashekar, A.C. Bovik, L.K. Cormack, Doves: a database of visual eye movements, Spat. Vision 22 (2) (2009) 161–177.
[40] S.S. Kruthiventi, K. Ayush, R.V. Babu, Deepfix: a fully convolutional neural network for predicting human eye fixations, arXiv preprint arxiv:1510.02927.
[41] R. Rubinstein, A.M. Bruckstein, M. Elad, Dictionaries for sparse representation modeling, Proc. IEEE 98 (6) (2010) 1045–1057.
[42] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
[43] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.
[44] C.E. Metz, Basic principles of roc analysis, in: Seminars in nuclear medicine, Vol. 8, Elsevier, 1978, pp. 283–298.
[45] A. Borji, L. Itti, State-of-the-art in visual attention modeling, Pattern Anal. Mach. Intell. IEEE Trans. 35 (1) (2013) 185–207.
[46] Salicon demo, ⟨http://salicon.net/demo/⟩.
[47] B.-K. Bao, G. Zhu, J. Shen, S. Yan, Robust image analysis with sparse representation on quantized visual features, Image Process. IEEE Trans. 22 (3) (2013) 860–871.
[48] R. Maani, S. Kalra, Y.-H. Yang, Robust edge aware descriptor for image matching, in: Computer Vision–ACCV 2014, Springer, 2014, pp. 553–568.
[49] N. Anantrasirichai, J. Burn, D.R. Bull, Robust texture features based on undecimated dual-tree complex wavelets and local magnitude binary patterns, in: Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015, pp. 3957–3961.
[50] J.R. Magnus, H. Neudecker, Matrix Differential Calculus, Wiley, New York, 1988.

Mai Xu received the B.S. degree from Beihang University in 2003, the M.S. degree from TsinghuaUniversity in 2006 and the Ph.D. degree from Imperial College London in 2010. From 2010 to 2012, he was working as a research fellow at the Electrical Engineering Department, Tsinghua University. Since Jan. 2013, he has been with Beihang University as an Associate Professor. His research interests mainly include visual communication and image processing. He has published more than 40 technical papers in international journals and conference proceedings.

Lai Jiang received her B.S. degree in electronic engineering from Beihang University, Beijing, China, in June 2015. He is currently a postgraduate student of Beihang University. His research interests include saliency detection and computer vision.

Zhaoting Ye is now undergraduate student of Beihang University. His research interests include image processing.

Zulin Wang received the B.S. and M.S. degrees in electronic engineering from Beihang University, in 1986 and 1989, respectively. He also received his Ph.D. degree at the same university in 2000. He is currently the dean of school of electronic and information engineering, at Beihang University, Beijing, China. His research interests include image processing, video coding, high-speed signal processing, electromagnetic countermeasure,complex object test, and satellite communications technology. He is author or co-author of over 100 papers and holds 6 patents, as well as published 2 books in these fields. He has undertaken approximately 30 projects related to image/video coding, wireless communication, etc. Now he has taught “image signal processing” course to undergraduates and “digital signal processing” course to postgraduates for nearly one decade. He is also the expert of China 863 program and the independent director of China Electronic Limited by Share Ltd.