Image Saliency Detection with Sparse Representation of Learnt Texture Atoms

Lai Jiang, Mai Xu*, Zhaoting Ye, Zulin Wang
School of Electronic and Information Engineering, Beihang University, Beijing, 100191, China
MaiXu@buaa.edu.cn

Abstract

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

1. Introduction

For modeling visual attention, saliency detection refers to computing on image features to characterize the regions attracting different amounts of 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 [2], object recognition [6], image retargeting [21], image quality assessment [5], and image/video compression [25].

Saliency detection can be traced back to feature integration theory [23] 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 [15] in 1987 proposed to yield the saliency map for an image, indicating which regions are conspicuous to attract attention in the HVS. Later, Itti and Koch [11] found out that the low level feature channels of intensity, color, and orientation are efficient 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 worked out by constructing 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 [11], extensive saliency detection methods (*e.g.*, [1, 3, 8, 10, 27]), using the plausible features designated by humans, have been proposed in the past decade.

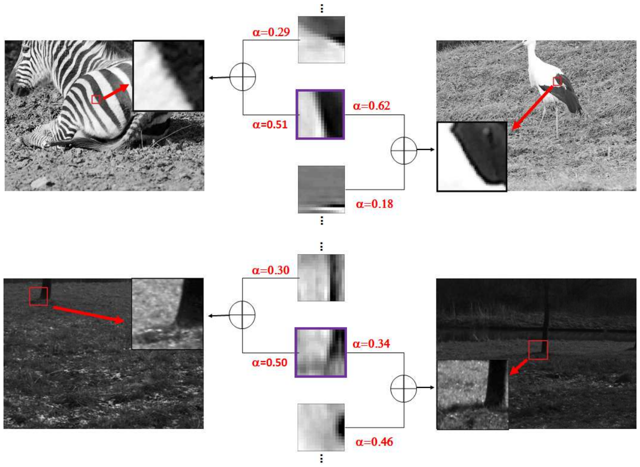


Figure 1. An example of salient patches with similar texture patterns. The regions inside the red squares (enlarged in the corners) are salient patches, in the images of the eye tracking Kienzle database (the first row) and Doves database (the second row). Some atoms of the dictionaries, learned from the salient regions of the 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 can be learned 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 about $1.5°$ in each database.

[*] Mai Xu is the corresponding author of this paper.

Recently, several saliency detection methods [12–14, 16, 19, 28] have been proposed to learn the parameters or even features from the ground-truth eye fixations over training images. From the perspective of parameters, a gaze-attentive fixation finding engine (GAFFE) [19] was developed to detect saliency, based on four low level image features: luminance, contrast, and bandpass outputs of luminance and contrast. In GAFFE, for modeling the saliency of natural images, the parameters and weights of bandpass features are learnt from the extensive eye tracking data [24]. However, the above method only focuses on learning some simple parameters, and prior features on diversifying saliency of an image still need to be exploited from the ongoing study on the HVS. From the perspective of features, Kienzle *et al.* [13, 14] proposed to directly learn patch patterns of salient and non-salient regions from the ground-truth eye tracking data. 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 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. However, the learnt patch patterns have limited expression, since only two positive and two negative patterns are available for saliency detection. Figure 1 shows the possibility of learning hundreds of salient patterns (by applying the dictionary learning algorithm) for saliency detection.

Therefore, this paper proposes to learn extensive positive and negative patterns from the eye tracking data of training images, for saliency detection. Specifically, this paper first proposes a formulation with a novel center-surround term, for learning two dictionaries which contain the atoms for basic texture patterns of salient and non-salient regions, respectively. On the basis of online dictionary learning [17], we develop the online salient dictionary learning (OSDL) algorithm to solve the proposed formulation, and then the discriminative 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, according to 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 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.

Although sparse representation of image patches has been utilized in the previous saliency detection work [20, 26], it only deals with the dictionary atoms from the spatially or temporally neighboring patches for the currently processed patch. Our method mainly focuses on learning the dictionaries from the eye tracking data of training images, rather than simply using the local image patches. Similar to other learning based methods [13, 14, 19], this paper only works on gray images with natural scenes.

The main contributions of this paper are listed as follows.

- We address a novel dictionary learning formulation by developing the OSDL algorithm, for generalizing salient and non-salient dictionaries from training eye tracking data. The detail is to be introduced in Section 2.
- 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. For the detail, refer to Section 3.

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

Due to the unmatured progress in visual psychophysics and neurophysiology, the existing saliency detection methods have limitation on accurately predicting the particular salient regions of images. Fortunately, the past few decades have witnessed the flourish of machine learning, which has potential in generalizing the low level features on attracting human attention. 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 2.1 our dictionary learning formulation on generalizing both salient and non-salient texture atoms. In Section 2.2, we present a solution to the proposed dictionary learning formulation.

2.1. Dictionary leaning formulation

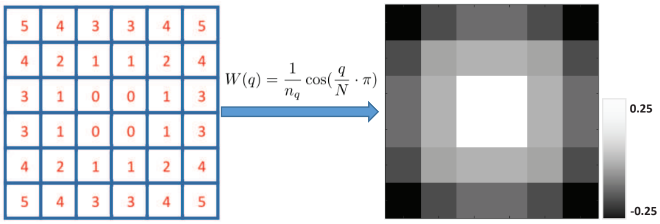
In sparse representation, an image patch[1] $\mathbf{x} \in \mathbb{R}^m$ can be sparsely represented by only a few texture atoms of dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$. Specifically, sparse coefficients $\boldsymbol{\alpha} \in \mathbb{R}^k$ need to be calculated for estimating image patch $\mathbf{x}$ with respect to dictionary $\mathbf{D}$. In fact, the problem of sparse representation can be formulated by

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

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. Towards the texture atoms, dictionary $\mathbf{D}$ needs to be learnt from training image patches $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$. This can be achieved through solving the following minimization problem [18, 22]:

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

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



5	4	3	3	4	5
4	2	1	1	2	4
3	1	0	0	1	3
3	1	0	0	1	3
4	2	1	1	2	4
5	4	3	3	4	5

Figure 2. An example of the center-surround weight function for the 6 × 6 image patches. Note that this example is only an illustration , and the real patch size is much larger. 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).

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 [13], 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. The same Euclidean distance corresponds to the same weight. 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 Euclidean distance can be calculated in the following function

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

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

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

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

where $\mathbf{S}'$ is the training set of fixation patches[2] 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. η 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 $-\eta\|\boldsymbol{l}^T\mathbf{D}'\|_2^2$ needs to be minimized. On the contrary, the center-surround degree is discouraged in non-salient dictionary $\mathbf{D}''$ by making $\eta\|\boldsymbol{l}^T\mathbf{D}''\|_2^2$ small.

2.2. 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:

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

To solve (5), the OSDL algorithm is proposed, based on online dictionary learning method [17], 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 for solving (5). Note that in our OSDL algorithm, only one randomly selected patch is input for each iteration.

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. The sparse representation is required to be conducted to work out 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 following formulation holds for estimating its sparse coefficients:

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

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

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 $\mathbf{D}' = \mathbf{D}'_T$.
```

where $\mathbf{D}'_{t-1}$ is the salient dictionary learnt from the last iteration $t-1$. In this paper, LASSO algorithm [4] is utilized for solving the sparse representation of (6).

Dictionary updating: After 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),

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

Note that $\mathbf{D}'_t$ is the salient dictionary learnt at the t-th iteration. To solve (7), we use the block-coordinate descent [17] to update each atom of the dictionary as follows,

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

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 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 matrix differentiation, (8) can be rewritten as

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

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

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

Note that 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 $\mathbf{B}'_t$. 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. The overall procedure of our OSDL algorithm is summarized in Table 1.

3. Saliency detection with the features regarding learnt texture dictionaries

Given the learnt salient and non-salient dictionaries, we can develop a novel low level feature, based on SR-LTA. More details about SR-LTA are discussed in Section 3.1. Section 3.2 presents the saliency detection method on the basis of the SR-LTA feature.

3.1. 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

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

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 Figure 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 Figure 3-(b), the conspicuity map generated by (11) is far from that of ground-truth human fixations, as the dynamic range of r is totally different from that of human fixation map. Hence, we introduce an exponential function into 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 saliency by human fixations. Here, a large value of τ is required for a more sparse distribution of conspicuity values. For example, as can seen from Figure 3-(c), $\tau = 5.6$ makes the distribution of the conspicuity values of f_1 be approaching to the ground-truth human fixation map. Accordingly, τ is set to 5.6 in our experiments in Section 4. Finally, the pixel-wise SR-LTA feature $\mathbf{f}_1$ for an image can be achieved by computing f_1 of all pixels.

3.2. Saliency detection with SR-LTA feature

Now, we focus on the saliency detection by combining the SR-LTA feature with other two features. In [11], 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 [19]. For more details about the computation on features of luminance and contrast, refer to [19]. Note that the bandpass features of luminance and contrast in [19] are not included in our method, as it is not practical to learn the bandpass parameters from the eye tracking data of testing images. 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,

$$\operatorname*{argmin}_{\boldsymbol{\omega}} \sum_s \|\mathbf{U}_s\boldsymbol{\omega} - \mathbf{v}_s\|_2 \; s.t. \; \boldsymbol{\omega} \in (0,1), \sum_{p=1}^{3} \omega_p = 1. \quad (13)$$

In (13), $\mathbf{U}_s$ is the matrix of conspicuity maps for each training 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 [8] is applied in our method, and then the optimal weights corresponding to different feature channels can be worked out with least square fitting to the human fixations.

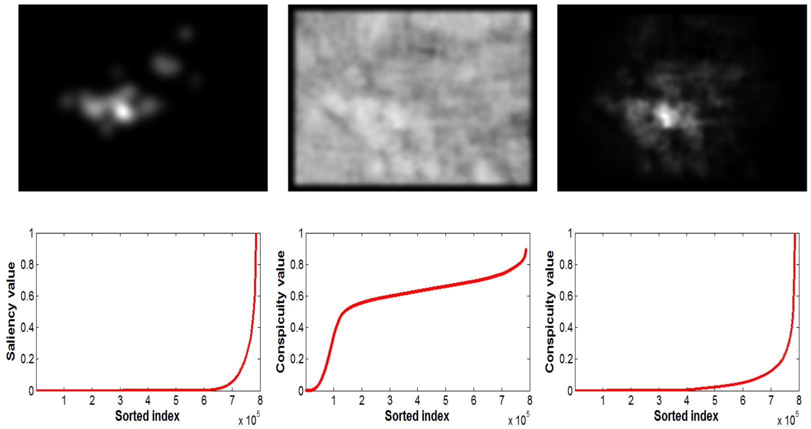
4. Experimental results

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 [24] and Kienzle *et al.* [13]. For the sake of comparison, we also provide the saliency detection results of other 8 state-of-the-art methods, including BMS [27], Itti *et al.*'s method [11], Duan *et al.*'s method [3], GAFFE [19], Hou *et al.*'s method [9], Zhao *et al.*'s method [28], Judd *et al.*'s method [12], and AWS [7]. In Section 4.1, we introduce the databases, training patches, and parameter settings in our experiments. In Section 4.2, we show the saliency detection results of our and other 8 methods. Here, the accuracy of saliency detection is evaluated using the metrics of receiver operator characteristics (ROC), area under the ROC curve (AUC), normalized scan-path saliency (NSS), and linear correlation coefficient (CC).

4.1. Experimental setup

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 [24] and Kienzle *et al.* [13], which provide the eye tracking data over gray images, were utilized for both the training and testing tasks of our experiments. Note that 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 [24] and [13].

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 for 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. Next, in our



(a) Human fixation map (b) Conspicuity map by (11) (c) Conspicuity map by f_1

Figure 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).

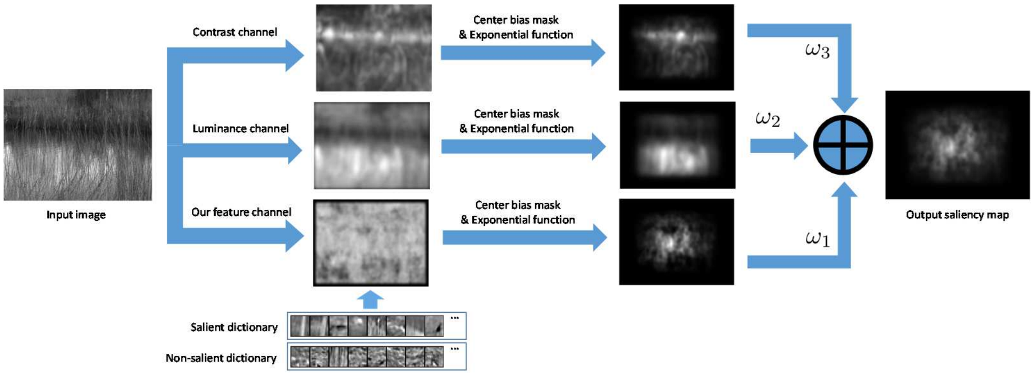


Figure 4. The procedure of our saliency detection method. The input image is processed through three channels, including our SR-LTA feature, contrast, and luminance, to obtain three conspicuity maps. Note that the conspicuity map of our SR-LTA feature channel is obtained through two learnt dictionaries. Then, a center bias mask [3] and an exponential function f_1 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 2. Details of two databases used in our experiments

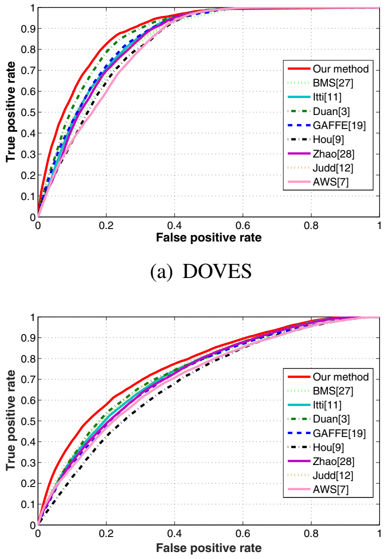
	DOVES	*Kienzle et al.*
Images	101	200
Human observers	29	14
Image size in pixel	1024 × 768	1024 × 768
Image size in visual angle	17° × 13°	36° × 27°
Total fixations	30,000+	18,000+

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 about 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 as well.

Parameters settings. All parameters related to our experiments are summarized in Table 3. For dictionary learning with our OSDL algorithm, according to the empirical settings in [17], the number of atoms k of each dictionary is set to $4 \cdot m$, and the regularization parameter λ in (4) for the tradeoff between reconstruction error and sparsity is set to $1.2/\sqrt{m}$. In addition, parameter η in (4) is tuned to 0.5 and the learning rate γ in (9) is set to 0.05 in our experiments, to make the results appropriate. For saliency detection, as verified in Section 3.1, power parameter τ in exponential function f_1 is 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 $\{0.72, 0.09, 0.19\}$.

Table 3. The parameters setting 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	0.5
	Learning rate γ	0.05
Saliency detection	Power parameter τ	5.6
	Combination weights $\{\omega_p\}_{p=1}^{3}$	$\{0.72, 0.09, 0.19\}$



(a) DOVES

(b) Kienzle *et al.*

Figure 5. The ROC curves of saliency detection by our and other 8 methods over two databases, respectively.

4.2. Saliency detection results

In this subsection, we present the saliency detection results of our method, compared with other 8 state-of-the-art methods. In our method, the salient and non-salient dictionaries were learnt with the OSDL algorithm from the training images for each database, respectively. Then, the saliency maps of test images of each database were detected upon the SR-LTA with respect to the dictionaries of the corresponding database, using the method of Section 3.2. Moreover, the same center bias mask was employed in our and all other methods, since it has been pointed out [27] that the center bias is able to make saliency detection more precise according to the HVS.

First, we show in Figure 5 the ROC curves of saliency detection by our and other 8 methods, averaged over all test images, for each database. As can be observed from this figure, our method outperforms all other methods for both DOVES and Kienzle *et al.* databases. We further quantify the ROC performance of our and other 8 methods via AUC metric. Table 4 tabulates the AUC results of our and other 8 methods. It can be seen from this table that our method offers the better AUC results on detecting saliency of test images, for both DOVES and Kienzle *et al.* databases.

Second, 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, and CC measures the strength of a linear relationship between human fixations and saliency maps. Note that a large 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 eight methods, in terms of both NSS and CC metrics. Specifically, our method enjoys at least 0.306 and 0.106 improvement in NSS and CC, respectively.

Since our method is based on a novel low level feature SR-LTA, it is important to evaluate its benefit to the improvement of saliency detection accuracy. In fact, both our and GAFFE [19] methods employ the features of luminance and contrast, whereas our method includes the SR-LTA feature instead of the bandpass features in [19]. Thus, the improvement by the proposed SR-LTA feature on saliency detection can be determined, via comparing the saliency detection accuracy between our and GAFFE methods. As seen from Table 4, our method achieves 0.042 increment in AUC over GAFFE. Beyond, our method offers 0.420 and 0.137 enhancement in NSS and CC metrics, respectively, when compared with GAFFE. This verifies the effectiveness of the SR-LTA feature.

It is interesting to investigate the results of only applying the SR-LTA channel to saliency detection. 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 Table 4 (the second row for SR-LTA). 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. However, the overall performance of our method is superior to such a single feature, indicating the positive effect of luminance and contrast features.

At last, we show in Figures 6 and 7 the saliency maps of several randomly selected test images, detected by our and other 8 methods as well as the human fixations. 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. The subjective results here, together with all above objective results, illustrate that our method performs much better than other 8 state-of-the-art methods on saliency detection.

5. Conclusions

In this paper, we have proposed a saliency detection method with a novel feature called SR-LTA, to predict the saliency maps of gray images. In the proposed method, an optimization formulation with a novel center-surround term was proposed, for learning both salient and non-salient dic-

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

Metrics	*DOVES*			*Kienzle et al.*			*Overall*		
	AUC	NSS	CC	AUC	NSS	CC	AUC	NSS	CC
Our method	**0.886**(0.028)	**1.961**(0.365)	**0.582**(0.086)	**0.766**(0.065)	1.187(0.456)	0.488(0.139)	**0.804**(0.056)	**1.431**(0.429)	**0.518**(0.125)
SR-LTA	0.875(**0.027**)	1.815(0.358)	0.575(0.089)	0.765(0.064)	**1.199**(0.512)	**0.491**(0.146)	0.799(**0.052**)	1.393(0.463)	**0.518**(0.128)
BMS [27]	0.834(0.057)	1.274(0.374)	0.383(0.112)	0.728(0.093)	0.887(0.471)	0.364(0.170)	0.761(0.084)	1.009(0.442)	0.370(0.154)
Itti [11]	0.850(0.036)	1.331(**0.234**)	0.414(**0.077**)	0.734(0.068)	0.865(**0.277**)	0.364(**0.110**)	0.770(0.060)	1.012(**0.264**)	0.379(**0.101**)
Duan [3]	0.870(0.043)	1.463(0.272)	0.448(0.093)	0.735(0.080)	0.899(0.346)	0.387(0.142)	0.777(0.071)	1.077(0.324)	0.406(0.129)
GAFFE [19]	0.852(0.050)	1.404(0.313)	0.432(0.102)	0.721(0.076)	0.831(0.333)	0.357(0.139)	0.762(0.069)	1.011(0.327)	0.381(0.129)
Hou [9]	0.828(0.061)	1.213(0.343)	0.382(0.124)	0.690(0.095)	0.630(0.388)	0.286(0.179)	0.733(0.086)	0.814(0.375)	0.317(0.164)
Zhao [28]	0.843(0.052)	1.308(0.385)	0.407(0.123)	0.727(**0.062**)	0.860(0.288)	0.359(0.113)	0.764(0.059)	1.001(0.322)	0.374(0.116)
Judd [12]	0.849(0.058)	1.438(0.415)	0.439(0.133)	0.741(0.082)	0.981(0.491)	0.399(0.143)	0.775(0.075)	1.125(0.468)	0.412(0.140)
AWS [7]	0.822(0.034)	1.183(0.248)	0.363(0.079)	0.709(0.092)	0.825(0.512)	0.337(0.174)	0.745(0.079)	0.938(0.446)	0.346(0.150)

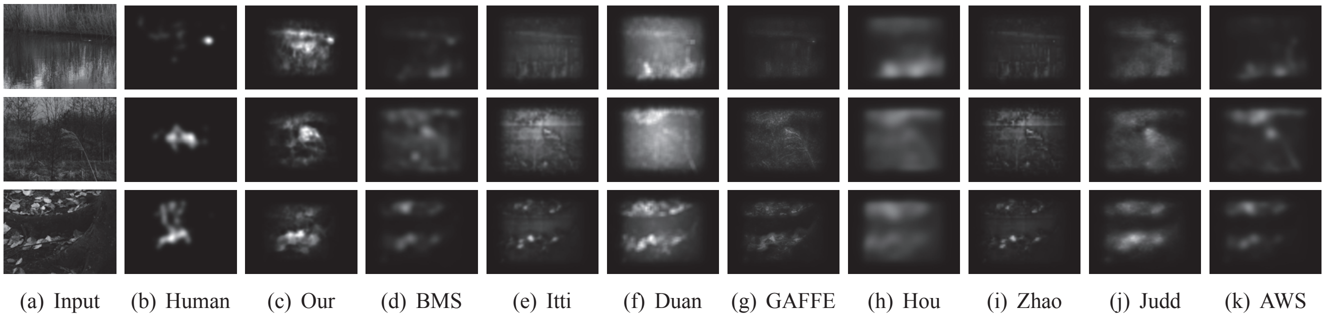

(a) Input (b) Human (c) Our (d) BMS (e) Itti (f) Duan (g) GAFFE (h) Hou (i) Zhao (j) Judd (k) AWS

Figure 6. Saliency maps of three test images from DOVES database, yielded by our and other 8 methods as well human fixations.

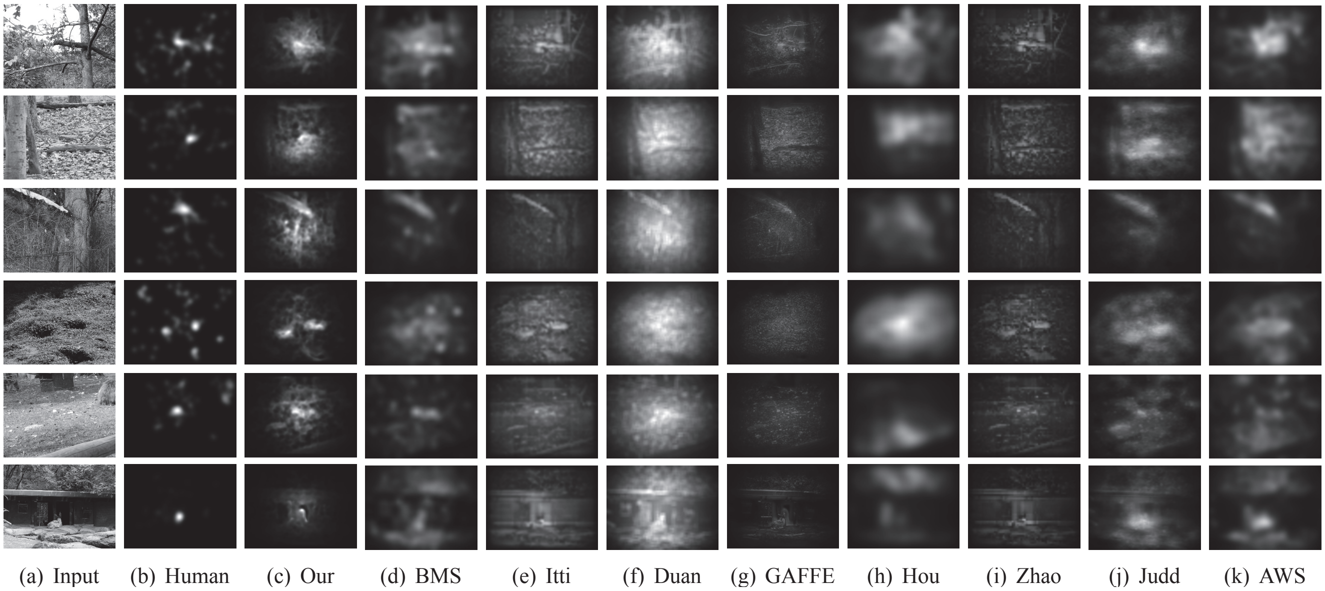

(a) Input (b) Human (c) Our (d) BMS (e) Itti (f) Duan (g) GAFFE (h) Hou (i) Zhao (j) Judd (k) AWS

Figure 7. Saliency maps of six test images from Kienzle *et al.* database, yielded by our and other 8 methods as well human fixations.

tionaries from the training fixation and non-fixation patches. Beyond, the OSDL algorithm was developed to solve the proposed formulation for learning dictionaries, in light of online dictionary learning. Then, the SR-LTA feature can be obtained, upon the difference of sparse representation errors regarding the learnt salient and non-salient dictionaries. At last, the saliency map of an input gray image can be generated, via combining the conspicuity maps of the proposed SR-LTA feature with those of other two low level features (luminance and contrast), in which the weight of each feature channel is determined via the least square fitting on training data. Compared with other 8 state-of-the-art saliency detection methods, our method performs significantly better on two database: DOVES and Kienzle *et al.*, in terms of ROC, AUC, NSS, and CC.

Acknowledgement. This work was supported by the NSFC projects under Grants 61573037, 61202139, and 61471022.

References

[1] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *Advances in neural information processing systems (NIPS), 2005*. 1

[2] N. J. Butko and J. R. Movellan. Optimal scanning for faster object detection. In *Computer Vision and Pattern Recognition (CVPR), 2009*. 1

[3] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu. Visual saliency detection by spatially weighted dissimilarity. In *Computer Vision and Pattern Recognition (CVPR), 2011*. 1, 5, 6, 8

[4] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 4

[5] U. Engelke, H. Kaprykowsky, H. Zepernick, and P. Ndjiki-Nya. Visual attention in quality assessment. *Signal Processing Magazine, IEEE*, 28(6):50–59, 2011. 1

[6] D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):989–1005, 2009. 1

[7] A. Garcia-Diaz, V. Leborán, X. R. Fdez-Vidal, and X. M. Pardo. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17, 2012. 5, 8

[8] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems (NIPS), 2006*. 1, 5

[9] X. Hou, J. Harel, and C. Koch. Image signature: Highlighting sparse salient regions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):194–201, 2012. 5, 8

[10] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition (CVPR), 2007*. 1

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. 1, 5, 8

[12] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *International Conference on Computer Vision (ICCV), 2009*. 2, 5, 8

[13] W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann. Center-surround patterns emerge as optimal predictors for human saccade targets. *Journal of Vision*, 9(5):7, 2009. 2, 3, 5

[14] W. Kienzle, F. A. Wichmann, M. O. Franz, and B. Schölkopf. A nonparametric approach to bottom-up visual saliency. In *Advances in Neural Information Processing Systems (NIPS), 2007*. 2

[15] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of Intelligence, 1987*. 1

[16] M. Kümmerer, L. Theis, and M. Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *arXiv preprint arXiv:1411.1045*, 2014. 2

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *International Conference on Machine Learning(ACM), 2009*. 2, 3, 4, 6

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition (CVPR), 2008*. 2

[19] U. Rajashekar, I. Van Der Linde, A. C. Bovik, and L. K. Cormack. Gaffe: A gaze-attentive fixation finding engine. *Image Processing, IEEE Transactions on*, 17(4):564–573, 2008. 2, 5, 7, 8

[20] Z. Ren, S. Gao, L.-T. Chia, and D. Rajan. Regularized feature reconstruction for spatio-temporal saliency detection. *Image Processing, IEEE Transactions on*, 22(8):3120–3132, 2013. 2

[21] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. In *ACM transactions on graphics (TOG), 2010*. 1

[22] R. Rubinstein, A. M. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010. 2

[23] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980. 1

[24] I. Van Der Linde, U. Rajashekar, A. Bovik, and L. Cormack. Doves: a database of visual eye movements. *Spatial vision*, 22(2):161, 2009. 2, 5

[25] M. Xu, X. Deng, S. Li, and Z. Wang. Region-of-interest based conversational hevc coding with hierarchical perception model of face. *IEEE Journal of Selected Topics on Signal Processing*, 8(3), 2014. 1

[26] J. Yan, M. Zhu, H. Liu, and Y. Liu. Visual saliency detection via sparsity pursuit. *Signal Processing Letters, IEEE*, 17(8):739–742, 2010. 2

[27] J. Zhang and S. Sclaroff. Saliency detection: a boolean map approach. In *International Conference on Computer Vision (ICCV), 2013*. 1, 5, 7, 8

[28] Q. Zhao and C. Koch. Learning a saliency map using fixated locations in natural scenes. *Journal of vision*, 11(3):9, 2011. 2, 5, 8