# Learning Sparse Dictionaries for Saliency Detection

Karen Guo and Hwann-Tzong Chen
Department of Computer Science
National Tsing Hua University, Taiwan

***Abstract*—We present a new method of predicting the visually salient locations in an image. The basic idea is to use the sparse coding coefficients as features and find a way to reconstruct the sparse features into a saliency map. In the training phase, we use the images and the corresponding fixation values to train a feature-based dictionary for sparse coding as well as a fixation-based dictionary for converting the sparse coefficients into a saliency map. In the test phase, given a new image, we can get its sparse coding from the feature-based dictionary and then estimate the saliency map using the fixation-based dictionary. We evaluate our results on two datasets with the shuffled AUC score and show that our method is effective in deriving the saliency map from sparse coding information.**

## I. Introduction

The human visual system can process enormous visual data instantly. Many computational models try to achieve such capabilities in different ways. Among various visual mechanisms the visual saliency is a key component and can be used as a prior for other components to provide an efficient approximate solution to more difficult problems. For example, the information of visual saliency can be used for image segmentation [6], foreground-background separation [15], object detection [5], and image compression [9]. Furthermore, saliency detection has been applied to image processing tasks such as content-aware image resizing [2], decolorization [1], or photo collage [17]. The results of visual saliency detection can also be employed in designing the layout of advertisement or filling the missing object parts [18]. Computational saliency with respect to the human fixation is a central issue for these applications.

Various algorithms have been presented to produce saliency maps of images, *e.g*. [4], [8], [10], [11], [12]. Typically, most saliency detection methods attempt to find rules for combining low-level information in an image. They may be based on the study of the human visual system to figure out the rules. In addition, some learning-based methods treat the saliency detection problem as a classification or regression problem, in which training data are required to learn the mapping from image features to saliency levels.

Itti *et al*. [10] propose a bottom-up way to generate saliency maps. Their method fuses several image features, such as color, orientation, and intensity, to obtain the resulting saliency map. The Graph Based Visual Saliency (GBVS) model proposed by Harel *et al*. [8] is also a bottom-up approach that uses graph and dissimilarly measure to construct the saliency model. The idea of "Information Maximization" proposed by Bruce and Tsotsos [4] is based on the information theory, which implies that the more common a word appears, the fewer digits are needed to represent the word. Their method computes Shannon's self-information by $-\log p(x)$, where $x$ is the information of an image. We use this idea in our method to extract color and edge-orientation features.

Learning-based methods usually take the saliency detection problem as a classification or regression task. For example, Judd *et al*. [11] consider various kinds of features, including high-level features like face detection and horizon-line detection, as well as low-level features like color and orientation. The 'ground-truth' fixations are treated as labels, and a support vector machine classification model can be trained to predict if a location is salient or not according to the high- and low-level features.

Another possible way to derive the saliency map is applying graphical models. Yang *et al*. [19] present a *conditional random field* (CRF) model to describe the relation between the label (indicating the importance) and the underlying neighboring pixels. They also use the sparse coding technique to compute the pre-trained features, and then update the parameters of the CRF model and the dictionary iteratively. After learning, they can use the dictionary and the CRF model to generate the saliency map of a test image. We also use a pre-trained dictionary to compute the sparse coding of an input image, and then take the sparse coefficients as features to generate the saliency map.

The goal of this paper is to study visual saliency and to find a better way of deriving saliency information. Our method can be considered a new kind of learning-based method with a bottom-up procedure. We seek to fuse the sparse coding coefficients into a saliency map using linear mapping. A major issue with such an approach is that if we learn the mapping and the dictionary separately, the dictionary would only correspond to the image features without taking account of the fixation information. In our approach, we model the image features and the fixation information jointly. The dictionary of image features would be actively updated when a different mapping from sparse coding to fixation is learned, and the mapping would also be updated when the sparse coding is changed after the dictionary being updated. We will demonstrate that training the image feature dictionary with the fixation information would achieve better accuracy of saliency prediction.

## II. Problem Formulation

Given an image, we are interested in predicting the probability of human fixation at each pixel. We try to find a general saliency representation so that we can transform image contents represented by sparse-coding coefficients into

a saliency map. It is well known that natural images can be sparsely represented by a set of localized and oriented filters. Recent research in pattern recognition and image processing has demonstrated that sparse coding is an effective method to represent an image.

Suppose that a local area in an image is represented as a vector $\mathbf{x}$ consisting of image features. Given a learned dictionary $\mathbf{D}$ with $n$ words of the same dimension as vector $\mathbf{x}$, we may convert the vector $\mathbf{x}$ into its sparse-coding representation $\boldsymbol{\alpha}$ by solving the following $\ell_1$-norm minimization problem:

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{2}\left\|\mathbf{D}\boldsymbol{\alpha}-\mathbf{x}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}\right\|_1 , \tag{1}$$

where $\|\cdot\|_1$ denotes the $\ell_1$-norm and $\lambda$ is a regularization parameter to decide the sparsity.

In addition to the sparse coding dictionary of image features, we include the saliency map to formulate a sparse coding learning problem described in the next section.

## III. Learning the Sparse Coding Dictionary

Given an image $\mathbf{I}^{(i)}$, we extract an $m$-dimensional feature vector $\mathbf{y}_j^{(i)}$ from an $r\times r$-pixel patch $p_j^{(i)}$ at each pixel location $j$ in $\mathbf{I}^{(i)}$. We formulate the following dictionary learning problem:

$$\begin{aligned}\underset{\mathbf{D},\boldsymbol{\alpha},\hat{D}}{\text{minimize}} \sum_i \sum_{p_j^{(i)}\in\mathbf{I}^{(i)}} &\frac{1}{2}\left\|\mathbf{D}\boldsymbol{\alpha}_j^{(i)}-\mathbf{y}_j^{(i)}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}_j^{(i)}\right\|_1 \\ &+ \frac{1}{2}\beta\left\|\hat{D}\boldsymbol{\alpha}_j^{(i)}-f_j^{(i)}\right\|_2^2 ,\end{aligned} \tag{2}$$

where $\boldsymbol{\alpha}_j^{(i)}$ is the sparse coding coefficients of $\mathbf{y}_j^{(i)}$ corresponding to $\mathbf{D}$ and $\hat{D}$, $f_j^{(i)}$ is the fixation value of pixel $j$ in image $\mathbf{I}^{(i)}$, and $\beta$ is a parameter to control the importance between features $\mathbf{y}_j^{(i)}$ and fixation $f_j^{(i)}$.

Since $\mathbf{D}$ and $\hat{D}$ share the same coefficients the $\boldsymbol{\alpha}_j^{(i)}$, we can simplify (2) as follows:

$$\begin{aligned}\underset{\mathbf{D},\boldsymbol{\alpha},\hat{D}}{\text{minimize}} \sum_i \sum_{p_j^{(i)}\in\mathbf{I}^{(i)}} &\frac{1}{2}\left\|\begin{bmatrix}\mathbf{D}\\ \beta\hat{D}\end{bmatrix}\boldsymbol{\alpha}_j^{(i)} - \begin{bmatrix}\mathbf{y}_j^{(i)}\\ \beta f_j^{(i)}\end{bmatrix}\right\|_2^2 \\ &+ \lambda\left\|\boldsymbol{\alpha}_j^{(i)}\right\|_1 .\end{aligned} \tag{3}$$

Therefore, we can use existing dictionary learning and sparse coding algorithms to solve (3). We apply the *least angle regress* (LARS) algorithm [7] implemented in the SPArse Modeling Software (SPAMS) [14] (http://spams-devel.gforge.inria.fr/) to do the sparse decomposition and generate the sparse coding $\boldsymbol{\alpha}$. In practice, we train the dictionaries with SPAMS toolkit using the sparsity mode considering both $l_1$-norm and $l_2$-norm of $\boldsymbol{\alpha}$:

$$\begin{aligned}\underset{\mathbf{D},\boldsymbol{\alpha},\hat{D}}{\text{minimize}} \sum_i \sum_{p_j^{(i)}\in\mathbf{I}^{(i)}} &\frac{1}{2}\left\|\begin{bmatrix}\mathbf{D}\\ \beta\hat{D}\end{bmatrix}\boldsymbol{\alpha}_j^{(i)} - \begin{bmatrix}\mathbf{y}_j^{(i)}\\ \beta f_j^{(i)}\end{bmatrix}\right\|_2^2 \\ &+ \lambda_1\left\|\boldsymbol{\alpha}_j^{(i)}\right\|_1 + \lambda_2\left\|\boldsymbol{\alpha}_j^{(i)}\right\|_2 .\end{aligned} \tag{4}$$

An overview of the training algorithm is illustrated in Fig. 1.

## IV. Computing Local and Global Features

To generate meaningful saliency maps, we consider both local and global features in our model. Local features characterize the distinctiveness of a patch in comparison with its neighbors, while global features represent the rarity of a patch with respect to all patches in the image. We need to normalize the values of all features before training. The features used in our method are described as follows.

### A. Dense SIFT

SIFT [13] is widely used in computer vision for computing local features. The SIFT descriptor calculates the statistics of gradient orientations in a local region. In out model we use the dense-SIFT program in the VLFeat toolkit [16]. Dense-SIFT generates SIFT features at a pixel with given window size and step length. We compute the SIFT features in a $10\times 10$ window and sample points with step length equal to 5 pixels. Since the dimension of original SIFT features is high, we use PCA to reduced its dimension to speed up the evaluation. We make the $\ell_2$-norm of the SIFT feature of each patch to be 1 so that it would not dominate the whole optimization problem.

### B. Global Color Distribution

Color histogram is a common feature in visual saliency research. We propose a global color feature that represents the color uniqueness of a patch within the whole image. This idea is inspired by the global saliency score proposed by Borji and Itti [3]. We quantize the RGB space into bins, and each pixel is cast into one of the bins according to its color. We compute the color distribution with respect to the color bins, and use the color distribution to derive the global color feature $\phi_j^{(i)}$ of each pixel $j$ in image $I^{(i)}$ by the following formula:

$$\phi_j^{(i)} = -\sum_{k\in\mathcal{N}_j^{(i)}} \log P(c_k) , \tag{5}$$

where $\mathcal{N}_j^{(i)}$ is the neighborhood of pixel $j$ in image $\mathbf{I}^{(i)}$, and $P(c_k)$ is the likelihood of observing color $c_k$ in the image according to the color distribution. In our experiments, we quantize each channel into 8 levels and thus we have totally 512 bins in the RGB space.

### C. Global Gabor Filter Response

The global Gabor filter response is obtained in a similar way as the global color distribution. We apply the Gabor filters of different orientations and different scales to each image. We then quantize the response values of each Gabor filter and transform them into a probability distribution. With

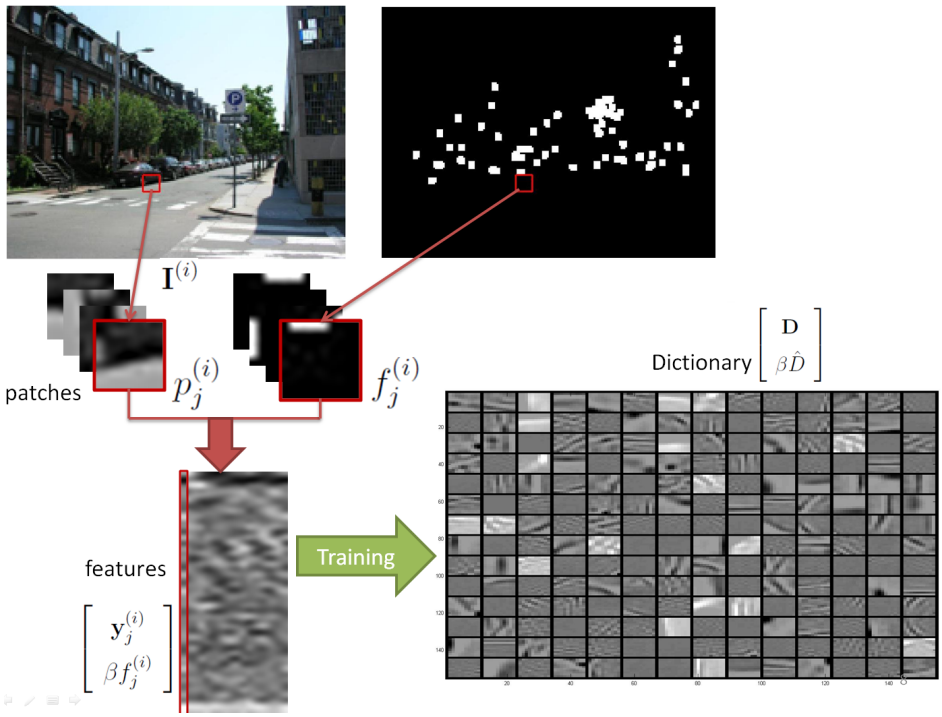


Fig. 1. The workflow of dictionary training

the distributions derived from different Gabor filters, we can generate global Gabor filter response $\psi_j^{(i)}$ of each pixel $j$ in image $\mathbf{I}^{(i)}$:

$$\psi_j^{(i)} = -\sum_{k\in\mathcal{N}_j^{(i)}}\sum_{g_t^{(i)}} \log P(g_{t,k}^{(i)})\,, \tag{6}$$

where $g_t^{(i)}$ is the $t$th Gabor filter response of image $\mathbf{I}^{(i)}$, and $P(g_{t,k}^{(i)}))$ is the probability for pixel $k$ having the Gobar filter response $g_t^{(i)}$. In our experiments, we use the Gabor filter toolkit implemented by Petkov and Wieling (http://matlabserver.cs.rug.nl/) to generate 3-scale filters with $\{8, 8, 4\}$ orientations.

## V. Datasets

In this work, we use two datasets to evaluate our method.

- MIT dataset [11]:
  This dataset contains 1,003 images of various sizes. The images are collected from Flickr and LabelMe, and the fixation data are obtained from the eye tracking results of 15 subjects. The dataset is available at http://people.csail.mit.edu/tjudd/WherePeopleLook/.
- Toronto dataset [4]:
  This dataset contains 120 images of a size of $511 \times 681$ pixels, capturing indoor and outdoor scenes. The fixation data are collected from 20 subjects. The dataset can be downloaded from http://www-sop.inria.fr/members/Neil.Bruce/.

With the two datasets, we are able to train the dictionaries using one dataset and perform the tests on the other dataset. The experimental results shown in the following sections are all obtained under this protocol.

## VI. Evaluation

We use *shuffled AUC* [20] instead of AUC (Area Under Curve) [4] to evaluate the performance because the AUC method would favor central Gaussian bias. The AUC is calculated by treating the saliency map as a classifier. It takes the fixations as the positive data and randomly choose the negative data from the rest areas of the image. By setting different thresholds to the saliency map, we obtain different performances of the classifier associated with the saliency map. The resulting false positive and true positive rates yield an ROC (Receiver Operating Characteristic) curve, and we may compute the area under curve as a measure of the overall quality of the saliency map. Instead of uniformly choosing negative data from all locations in the image, shuffled AUC introduces the *shuffled map* as a source for generating negative data. The shuffled map is created by piling up all the fixation maps of the training data except the positive one. (See Fig. 2 for examples of shuffled maps.) The evaluation method is implemented by Borji and Itti [3], and the MATLAB code can be obtained from https://sites.google.com/site/saliencyevaluation/evaluation-measures.

### A. Predicting Saliency

After we obtain the feature-based dictionary $\mathbf{D}$ and the fixation-based dictionary $\hat{D}$, we may use them to predict the saliency map of a new input image. Given a test image $\mathbf{I}$, we first extract the features $\mathbf{y}$. We use LARS decomposition

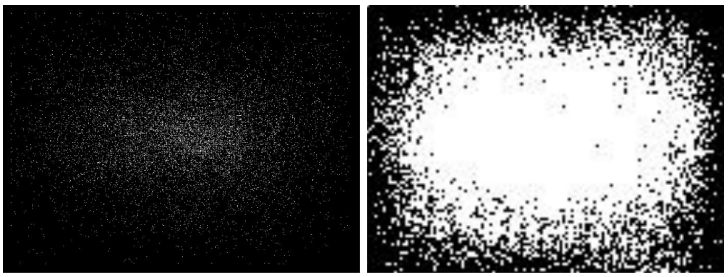


Fig. 2. The shuffled maps of the two datasets.

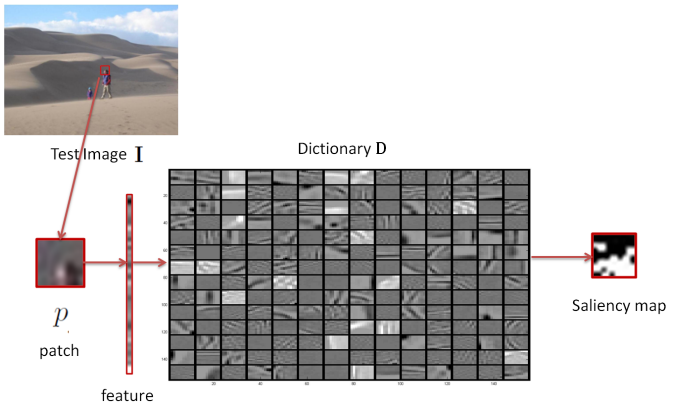


Fig. 3. The workflow of estimating the saliency map of a test image: The features extracted from the image are transformed into sparse coding by the dictionary. Then by fusing the sparse coding, we can obtain the predicted saliency map.

TABLE I
SHUFFLED AUC SCORES OF DIFFERENT METHODS AND DIFFERENT FEATURE COMBINATIONS.

| | Center | GBVS [8] | Itti *et al.* [10] | Gray Only | SIFT Only | Ours All |
|---|---|---|---|---|---|---|
| MIT | 0.5420 | 0.6582 | 0.6735 | 0.5746 | 0.5723 | 0.6796 |
| Toronto | 0.5094 | 0.6292 | 0.6557 | 0.5669 | 0.5730 | 0.6940 |

TABLE II
THE SHUFFLED AUC SCORES WITH DIFFERENT $\beta$ VALUES.

| | $\beta = 1$ | $\beta = 5$ | $\beta = 10$ | $\beta = 23$ | $\beta = 46$ |
|---|---|---|---|---|---|
| MIT | 0.6235 | n/a | n/a | n/a | n/a |
| Toronto | 0.6422 | 0.6526 | 0.6233 | 0.6287 | 0.6108 |

| | $\beta = 0.005$ | $\beta = 0.01$ | $\beta = 0.1$ | $\beta = 0.25$ | $\beta = 0.5$ |
|---|---|---|---|---|---|
| MIT | n/a | 0.6796 | n/a | n/a | 0.6706 |
| Toronto | 0.6906 | 0.6940 | 0.6913 | 0.6886 | 0.6898 |

to find the sparse coding coefficients $\boldsymbol{\alpha}$ with respect to the feature-based dictionary $\mathbf{D}$. With the sparse coding coefficients $\boldsymbol{\alpha}$, we can reconstruct the saliency map by multiplying the pre-trained fixation-based dicttionary $\hat{D}$ and the sparse coding coefficients $\boldsymbol{\alpha}$ to approximate the saliency map. The workflow of testing is shown in Fig. 3.

### B. Features Selection

We include three types of features in our method. In Table I we show the results of using different feature combinations. We also compare our method with the approaches of Harel *et al*. [8] and Itti *et al*. [10]. 'Center' means taking the simple Gaussian blob as a saliency map. From the table, we can see that using only the intensity information ('Gray Only') or SIFT features ('SIFT Only') cannot achieve comparable scores to those generated from the combined features. In addition, our method achieve higher scores than both the GBVS method proposed by Harel *et al*. [8] and the method of Itti *et al*. [10].

### C. Different Weights between Features and Fixation Values

The correlation between features and fixation is controlled by the parameter $\beta$. We evaluate the effects of changing the value of $\beta$ and show their corresponding shuffled AUC scores in Table II.

The results shown in Table II suggest that smaller $\beta$ values might yield higher shuffled AUC scores. What is interesting is that if we intuitively consider an equitable training preference for both feature-based and fixation-based dictionaries, the weight of fixation should be higher. However, no increase in the shuffled AUC score is observed when we increase the value of $\beta$. A possible explanation to this situation is that during the testing phase we compute the sparse coding coefficients only from the image features because we do not have the fixation information during testing. Therefore, emphasizing too much on fixation during training would result in larger errors on features, and it would make the feature-based dictionary lose its generalization power. Fig. 4 shows some examples of the resulting saliency maps using different $\beta$ values.

### D. Comparison: Training Dictionaries Separately

We are also interested in the difference between training the feature-based dictionary $\mathbf{D}$ and the fixation-based dictionary $\hat{D}$ separately or jointly. We compare the shuffled AUC scores of these two methodologies and examine whether our assumption that training the two dictionaries jointly would be better is true or not. The comparison is shown in Table III. We can see that training the dictionaries $\mathbf{D}$ and $\hat{D}$ separately performs worse than our proposed method in which the dictionaries are trained jointly. Although the difference is small, the result still demonstrates that training with image features and fixations

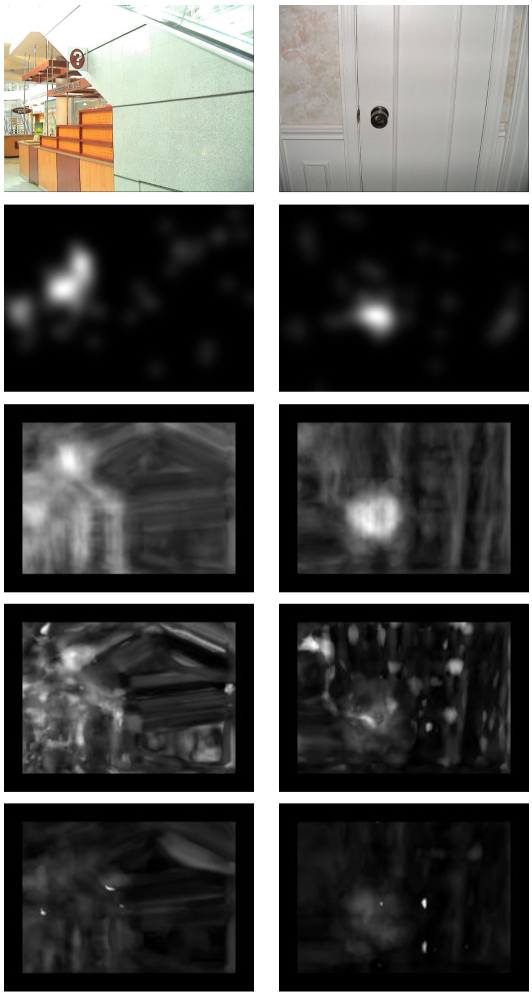

Fig. 4. Two sets of results obtained by setting different $\beta$ values. From top to bottom in each column: the original image, the 'ground-truth' saliency map derived from the fixation data, the resulting saliency map with $\beta = 0.01$, the resulting saliency map with $\beta = 1$, the resulting saliency map with $\beta = 46$.

together can preserve more information.

TABLE III
THE RESULTS OF TRAINING THE DICTIONARIES $\mathbf{D}$ AND $\hat{D}$ JOINTLY OR SEPARATELY.

| | Separately trained | Jointly trained, $\beta = 0.5$ |
|---|---|---|
| MIT | 0.6790 | 0.6880 |
| Toronto | 0.6885 | 0.6940 |

## VII. DISCUSSIONS

This paper presented a new method of estimating the saliency map by training dictionaries of image features and fixations. Through the experiments, we find that our method performs comparably well as previous saliency detection methods. Nevertheless, several issues are worth further investigations. First, the features using in our method are too specific. Although we have considered different types of features with global and local properties, there exist some other types of features that might be useful. For example, the position of each pixel can be used to indicate if the saliency values are stronger at some specific areas.

Another issue that we need to address is the run-time of our program. We use LARS decomposition in the SPAMS toolkits to train dictionaries and to extract sparse coding coefficients, and the whole process takes a lot of time to run through all the images in the dataset. It would take about one minute to obtain the sparse coding of one image under our current implementation. We may try to find other efficient decomposition tools to improve the run-time in the future.

In sum, we have proposed a new learning-based method for saliency detection and have shown that it achieves good performance in predicting the location of fixation. Future research may focus on new feature extraction and sparse decomposition methods to efficiently and accurately reproduce the human fixation.

**Acknowledgment:** This research was supported in part by Novatek grant 100F2242EA.

## REFERENCES

[1] C. O. Ancuti, C. Ancuti, and P. Bekaert. Enhancing by saliency-guided decolorization. In *CVPR*, pages 257–264, 2011.
[2] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3):10, 2007.
[3] A. Borji and L. Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, 2012.
[4] N. D. B. Bruce and J. K. Tsotsos. Saliency based on information maximization. In *NIPS*, 2005.
[5] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai. Fusing generic objectness and visual saliency for salient object detection. In *ICCV*, pages 914–921, 2011.
[6] K.-Y. Chang, T.-L. Liu, and S.-H. Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, pages 2129–2136, 2011.
[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. . least angle regression. *Ann. Stat.*, 32(2), 2004.
[8] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.
[9] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.
[10] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.
[11] T. Judd, K. A. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2009.
[12] Y. Li, Y. Zhou, L. Xu, X. Yang, and J. Yang. Incremental sparse saliency detection. In *ICIP*, pages 3093–3096, 2009.
[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.
[15] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.
[16] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.
[17] J. Wang, L. Quan, J. Sun, X. Tang, and H.-Y. Shum. Picture collage. In *CVPR (1)*, pages 347–354, 2006.
[18] M. Wang, J. Konrad, P. Ishwar, K. Jing, and H. A. Rowley. Image saliency: From intrinsic to extrinsic context. In *CVPR*, pages 417–424, 2011.
[19] J. Yang and M.-H. Yang. Top-down visual saliency via joint crf and dictionary learning. In *CVPR*, 2012.
[20] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8:32–32, 2008.