A computational visual saliency model based on statistics and machine learning

Ru-Je Lin

Wei-Song Lin

Department of Electrical Engineering,
National Taiwan University, Taiwan

Identifying the type of stimuli that attracts human visual attention has been an appealing topic for scientists for many years. In particular, marking the salient regions in images is useful for both psychologists and many computer vision applications. In this paper, we propose a computational approach for producing saliency maps using statistics and machine learning methods. Based on four assumptions, three properties (Feature-Prior, Position-Prior, and Feature-Distribution) can be derived and combined by a simple intersection operation to obtain a saliency map. These properties are implemented by a similarity computation, support vector regression (SVR) technique, statistical analysis of training samples, and information theory using low-level features. This technique is able to learn the preferences of human visual behavior while simultaneously considering feature uniqueness. Experimental results show that our approach performs better in predicting human visual attention regions than 12 other models in two test databases.

Introduction

Selective visual attention is a mechanism that helps humans select a relevant region in a scene. It organizes the vast amounts of external stimuli, extracts important information efficiently, and compensates for the limited human capability for visual processing. While psychologists and physiologists are interested in human visual attention behavior and anatomical evidence to support attention theory, computer scientists are concentrating on building computational models of visual attention that implement visual saliency in computers or machines. Computational visual attention has many applications for computer vision tasks, such as robot localization (Shubina & Tsotsos, 2010; Siagian & Itti, 2009), object tracking (G. Zhang, Yuan, Zheng, Sheng, & Liu, 2010), image/video compression (Guo & Zhang, 2010; Itti, 2004), object detection (Frintrop, 2006; Liu et al., 2011), image thumbnailing (Le Meur, Le Callet, Barba, & Thoreau, 2006; Marchesotti, Cifarelli, & Csurka, 2009), and implementation of smart cameras (Casares, Velipasalar, & Pinto, 2010).

However, it is not easy to simulate human visual behavior perfectly by machine. Attention is an abstractive concept, and it needs objective metrics for evaluation. Judging the results of experiments by intuitive observation is not precise because different people might focus on different regions of the same scene. To solve this issue, eye tracker equipment that can record human eye fixation, saccades, and gazes are routinely used. Investigations of human eye movement data provide more objective evaluations of computational attention models.

Most existing computational visual attention/saliency models process pixel relationships in an image/video according to human visual system behaviors. They compute some features such as region contrast, block similarity, symmetry, entropy, and spatial frequency, which are believed to be relevant to human attention, and attempt to analyze their rules to construct saliency maps. However, each of these features can represent only an isolated part of human visual behavior; sometimes these behaviors interact and simultaneously affect the results. Thus, it is a complex problem to construct perfect saliency maps by using only isolated features and linearly combining them. In addition, some known human biases, such as center bias and the border effect, have also influenced experimental results (L. Zhang, Tong, Marks, Shan, & Cottrell, 2008). Motivated by this, our aim is to choose only a few concepts that encompass comprehensive human visual behaviors, clarify the interactions among them, and develop a method for implementing the visual saliency
model. According to a derivation using a Bayesian theory, the concepts of visual saliency can be implemented by certain low-level properties and combined by simple mathematical operations. The experimental results show that the method is very useful and performs well.

The structure of this manuscript is as follows: The Previous works section provides a brief description and discussion of several existing computational visual attention models. The Assumption and Bayesian formulation section describes the assumptions, derivations, and relationships of salience concepts. The Learning-based saliency detection section illustrates the implementation details of the model. The Evaluation section evaluates our approach using two famous databases, the Toronto database (N. Bruce & Tsotsos, 2005) and Li 2013 database (Li, Levine, An, Xu, & He, 2013), and comparing the performance of 12 state-of-the-art models. The Discussion section presents a general discussion of evaluation results. The Conclusion section states the conclusion and outlines future work.

### Previous works

Most early visual attention/saliency models are inspired by the Feature Integration Theory, proposed by Treisman and Gelade (1980). Koch and Ullman (1985) proposed the winner-take-all (WTA) mechanism in combining features. Later, Itti, Koch, and Niebur (1998) proposed a model using a center-surround mechanism and hierarchical structure to predict salient regions. Many extended works are based on their model and attempt to improve upon it. Walther and Koch (2006) conducted an implementation called the Saliency Toolbox (STB), which posited proto-objects in saliency region detection. Harel, Koch, and Perona (2006) proposed the Graph-Based Visual Saliency (GBVS) model, similar to the Itti and Koch model (Itti et al., 1998) but with improved performance in the activation and normalization/combination stage that used a Markovian approach to describe dissimilarity and concentration mass regions. Following the Koch and Ullman model, Le Meur et al. (2006) proposed a bottom-up coherent computational approach, which used contrast sensitivity, perceptual decomposition, visual masking, and center-surround interaction techniques. It extracted features in Krauskopf’s color space and implemented saliency in three phases: visibility, perceptual grouping, and perception. Gao, Mahadevan, and Vasconcelos (2008) advanced discussed the center-surround hypothesis in their visual saliency model. Their study also proposed three applications of computational visual saliency: prediction of eye fixation on nature scenes, discriminant saliency on motion fields, and discriminant saliency in dynamic scenes. Constructed in the architecture of the Itti and Koch model, Erdem and Erdem (2013) published an improved model that contributed feature integration using region covariance. In general, the center-surround and WTA mechanisms importantly influenced the development of visual attention/saliency models, and inspired many later studies.

Many researchers applied the concept of probability to implement saliency in recent years. Itti and Baldi (2005) proposed a Bayesian definition of surprise to capture salient regions in dynamic scenes by measuring the difference between posterior and prior beliefs of observers. N. Bruce and Tsotsos (2005) proposed the Attention based on Information Maximization (AIM) model to implement saliency using joint likelihood and Shannon’s self-information. They also used independent component analysis (ICA) to build some pre-set basis functions for extracting features. Avraham and Lindenbaum (2010) proposed Esaliency, a stochastic model, to estimate the probability of interest in an image. They roughly segmented the image first and used a graphical model approximation in global considerations to determine which parts are more salient. Boiman and Irani (2007) proposed a general method for detecting irregularities in image and video. They developed a graph-based Bayesian inference algorithm in their model and proposed five applications of the system, including detecting unusual image configurations and suspicious behavior. However, occlusion and memory complexity are the two main limitations of the method. L. Zhang et al. (2008) proposed the Saliency Using Natural Statistics (SUN) model, which also used a Bayesian framework to describe saliency. The difference-of-Gaussian (DoG) filter and linear ICA filter is used to implement the model. L. Zhang et al. (2008) discussed the center bias and edge effects in detail, and considered implementing them in their model. Seo and Milanfar (2009a, 2009b) proposed a non-parametric method, the Saliency Detection by Self-Resemblance (SDSR) model, to detect saliency. This method computes self-resemblance of a local regression kernel obtained from the likeness of a pixel and its surroundings. Liu et al. (2011) proposed an approach to separate salient objects from their backgrounds by combining image features using a conditional random field. The features they used are described at local, regional, and global levels. Large number of images have been collected and labeled in this research to evaluate this model. The studies described above, constructed using probability, statistics, and stochastic concepts, bring differing perspectives to the attention/saliency problem, and served as the basis for many useful models.
Unlike most models, which compute saliency in the spatial domain, a few others attempt to compute saliency in the frequency domain. Hou and Zhang (2007) proposed a spectral residual (SR) approach for detecting visual saliency. They used a Fourier transform and log-spectrum analysis to extract spectral residual, thereby implementing a system that needs no prior knowledge and parameter determination. Guo et al. (Guo, Ma, & Zhang, 2008; Guo & Zhang, 2010) proposed a method of calculating spatiotemporal saliency maps using phase spectrum of quaternion Fourier transform (PQFT) instead of the amplitude spectrum. Using PQFT and a hierarchical selectivity framework, they also proposed an application model called Multiresolution Wavelet Domain Foveation, which can improve coding efficiency in image and video compression. Li, Tian, Huang, and Gao (2010) proposed a multitask approach for visual saliency using multiscale wavelet decomposition in video. This model can learn some top-down tasks and integrate top-down and bottom-up saliency by fusion strategies. Li et al. (2013) proposed a method using an amplitude spectrum convoluted with an appropriately scaled kernel as a saliency detector. Saliency maps can be obtained by filtering both the phase and amplitude spectrum with a scale selected by minimizing entropy. To summarize these frequency-based models, their main advantages are lower computational effort and faster computing speed.

Several researchers have proposed top-down or goal-driven saliency in their models. They use some high-level features based on those from earlier databases, and conduct learning mechanisms to determine model parameters. Tatler and Vincent (2009) proposed a model incorporating the oculomotor behavioral biases and statistical model to improve fixation prediction. They believe that a good understanding of how humans move their eyes is more beneficial than salience-based approaches. Torralba, Oliva, Castelhano, and Henderson (2006) proposed an attentional guidance approach that combines bottom-up saliency, scene context, and top-down mechanisms to predict image regions likely to be fixated by humans in real-world scenes. On the basis of a Bayesian framework, the model computes global features by learning the context and structure of images, and the top-down tasks can be implemented in the scene priors. Cerf, Frady, and Koch (2009) proposed a model that adds several high-level semantic features such as faces, text, and objects to predict human eye fixations. Judd, Ehinger, Durand, and Torralba (2009) proposed a learning-based method to predict saliency. They used 33 features including low-level features such as intensity, color, and orientation; mid-level features such as a horizon line detector; and high-level features such as a face detector and a person detector. The model used a support vector machine (SVM) to train a binary classifier. Lee, Huang, Yeh, and Chen (2011) proposed a model that adds faces as high-level features to predict salient positions in video. They used a support vector regression (SVR) technique to train the relationship between features and visual attention. Zhao and Koch (2011) proposed a model similar to that of Itti, Koch, and Niebur (1998), but with faces as an extra feature. Their model combines feature maps with learned weightings and solves the minimization problem using an active set method. The model also implements center bias in two ways. Among the models described above, some focus on adding high-level features to improve predictive performance, while others use machine learning techniques to clarify the relationship between features and their saliency. However, the so-called high-level features are blur concepts and do not encompass all types of environments. Moreover, most of the learning processes failed to consider that the same feature values likely have different saliency values in different contexts.

**Assumptions and Bayesian formulation**

In this section, we will discuss the assumptions of the visual saliency concepts we have considered and the relationship among them. We assume that saliency values in an image are relative to at least four properties, as described below.

**Assumption 1: Saliency is relative to the strength of features in the pixel**

We assume that a pixel with strong features tends to be more salient than one with weak features. Features are traditionally separated into two types, high- and low-level features. High-level features include face, text, and events. Low-level features include intensity, color, regional contrast, and orientations. Since high-level features are more complex to define and extract, we only consider low-level features in this paper.

**Assumption 2: Saliency is relative to the distinctiveness of features**

As long as the feature in a pixel is strong, it is not salient if there are many pixels with similar features in the image. In other words, the feature is conspicuous only if it is distinctive. For example, a red dot is salient...
Assumption 3: Saliency is relative to the location of the pixel in the image

The probability of saliency for every pixel in an image is the same if the distribution is uniform. However, previous research and experience have shown that humans have a strong center bias in their visual behavior (Borji, Sihite, & Itti, 2013; Erdem & Erdem, 2013; Tatler, 2007; Tseng, Carmi, Cameron, Munoz, & Itti, 2009; L. Zhang et al., 2008). There are several hypotheses for the root cause of center bias. For example, when humans look at a picture or a video, they naturally assume the important object will appear in the center of picture, and search the center part of image first (this is called the subject-viewing strategy bias). One of the other reasons is that people tend to place objects or interesting things near the center of an image when taking a picture (the so-called photographer bias). Several hypotheses for center bias have been proposed, including orbital movement, motor bias, and center of screen bias. Furthermore, some research has found that humans tend to conduct visual searches horizontally rather than vertically (the horizontal bias; Hansen & Essock, 2004). In any case, locational preferences in human visual behavior will be considered in our approach.

Assumption 4: Absolute saliency is relative to feature distribution in an image

Some images carry more information and some carry less, and the salient degree is not the same for all images. For example, a blank paper or a scene with only salt and pepper noise may have no salient region, since such images carry very little information. However, subjects in eye fixation experiments must be seeing something when they are looking at a scene, whether it contains a significant salient region or not. This fact leads to some mismatches in saliency analysis, when the fixation density maps are usually used as ground truth to evaluate the performance of saliency detection after being normalized to a range between one to zero. As a result, the absolute saliency level of an image should be considered. When an image is decomposed into several feature maps, the saliency degree of each feature map should also be determined. Entropy is one of the indices often used to determine the amount of information in a system in information theory. Note that entropy only reflects the energy in an image/feature map, but cannot describe the distribution of them.

Considering the assumptions described above, the saliency of a pixel can be defined as the probability of saliency given the features and positions. Denoted $F_\rho = [f_1^\rho, f_2^\rho, \ldots, f_K^\rho]$, $\rho = [x, y] \in I$ as a feature set include $K$ features located in a pixel position $\rho$ of image $I$. The saliency value can be denoted as $p(s|\rho, F_\rho)$; for convenience we assume $\rho$ and $F_\rho$ are independent of each other, as L. Zhang et al. (2008) did. The derivation of $p(s|\rho, F_\rho)$ using Bayesian theory is shown in Equation 1.

$$p(s|\rho, F_\rho) = p(\rho, F_\rho|s)p(s) = \frac{p(\rho|s) \cdot p(F_\rho|s) \cdot p(s)}{p(\rho) \cdot p(F_\rho)} = \frac{p(\rho|s)}{p(s)} \cdot \frac{p(s|F_\rho)}{p(s)} = \frac{p(s|\rho) \cdot p(s|F_\rho)}{p(s)}$$

(1)

In Equation 1, the term $p(s|\rho)$ is the probability of saliency given a position $\rho$. We called it Position-Prior and it corresponds to assumption 3. $p(s|F_\rho)$ is the probability of saliency of the features appearing in location $\rho$. Two aspects of this term can be examined. One of these is the probability of saliency of the feature over all image sets, which we call Feature-Prior denoted as $p(s|F_\rho, I)$. It assumes that some features are more salient than others due to the nature of the features themselves, and relates to assumption 1. The second aspect is the probability of a features saliency in a certain image, which we called Feature-Distribution and denoted as $p(s|F_\rho, I)$. This assumes that some features are more salient than others in a given image due to the image’s construction. We can observe that features are more salient if they are infrequent in an image, and different images/feature maps are salient at different levels, as assumptions 2 and 4 presume. Since $p(s|F_\rho)$ should address both of these two aspects, we defined $p(s|F_\rho)$ as equal to the product of these two terms, as Equation 2 shows.

$$p(s|F_\rho) = p(s|F_\rho, I) \cdot p(s|F_\rho, U)$$

(2)

Combining Equations 1 and 2, $p(s|\rho, F_\rho)$ can be rewritten as Equation 3.

$$p(s|\rho, F_\rho) = \frac{p(s|\rho) \cdot p(s|F_\rho)}{p(s)} = \frac{p(s|\rho) \cdot p(s|F_\rho)}{p(s)} \cdot \frac{p(s|F_\rho, U) \cdot p(s|F_\rho, I)}{p(s)} \propto p(s|\rho) \cdot p(s|F_\rho, U) \cdot p(s|F_\rho, I)$$

(3)

In Equation 3, $p(s)$ is a constant and not relative to features or positions. As a result, the probability of saliency is clearly relative to three terms: Position-Prior, Feature-Prior, and Feature-Distribution. We will describe the details of how to implement these terms in the next sections.
Based on Equation 3, there are three terms that affect saliency value in a pixel of an image: Feature-Prior, Position-Prior, and Feature-Distribution. Among these, Feature-Prior can be learned from training samples using SVR; Position-Prior can be learned from ground truths of training images; and Feature-Distribution can be computed directly from features in the image using information theory. As shown in Figure 1, several images and their density maps are selected in the learning process from the test database as training images and ground truths. After the feature extraction process, the features of several points are selected as training samples in each training image. All of the training samples are sent to SVR to train an SVR model. At the same time, Position-Prior also can be obtained from training images and their ground truths. In the saliency computing process, the images remaining from the test database are treated as test images. A test image can be decomposed into several feature maps after the feature extraction process. Position-Prior can be obtained from the learning process and Feature-Distribution can be obtained from the similarity and information computation process. Finally, the three parts are combined, and a saliency map can be obtained after being convoluted with a Gaussian filter.

Feature extraction

Both color features and region contrast features have been used as features in this study. We choose CIELab color space instead of other common color spaces such as RGB color space because it is more representative in its color contrast. Let $L$, $a$, $b$ as denoted the lightness, red/green, and blue/yellow channels of the input image in CIELab color space, hierarchically downscale the three channels respectively to obtain $L$, $a$, and $b$ maps in different scales, each scale being half the size of the previous scale.

$$L^{(i+1)} = DS(L^{(i)}) ,$$
$$a^{(i+1)} = DS(a^{(i)}) ,$$
$$b^{(i+1)} = DS(b^{(i)})$$

Here $DS(\bullet)$ denotes the downscaling operation, $L^{(i)}$ is the $L$ map in $i-th$ layer, and so on. Next, the different
scales \( L, a, \) and \( b \) are upscaled to their original size to obtain the color feature maps. Note that these maps have the same size but different resolutions.

\[
F^{(i)}_{(L)} = \text{US}\left( L^{(i)} \right), \\
F^{(i)}_{(a)} = \text{US}\left( a^{(i)} \right), \\
F^{(i)}_{(b)} = \text{US}\left( b^{(i)} \right)
\]

Where \( \text{US} (\bullet) \) denoted the upscale operation. For the present center-surround mechanism and orientation features, we used one DoG filter and four Gabor filters, which are shown in Equations 6 and Equation 7.

\[
\text{DoG}(x,y) = \frac{1}{\sqrt{2\pi \sigma_1^2}} \exp \left( -\frac{x^2 + y^2}{2\sigma_1^2} \right) \\
- \frac{1}{\sqrt{2\pi \sigma_2^2}} \exp \left( -\frac{x^2 + y^2}{2\sigma_2^2} \right)
\]

\[GF(x,y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left( -\frac{x^2 + y^2^2}{2\sigma^2} \right) \times \exp \left( i \left( 2\pi \frac{x}{\lambda} + \psi \right) \right)
\]

Where \( x = x \cos \theta + y \sin \theta, \) \( y = -x \sin \theta + y \cos \theta, \) and \( x \) and \( y \) are the horizontal and vertical direction coordinates. In our implementation, we set \( \sigma_1 = 0.8 \) and \( \sigma_2 = 1 \) in Equation 6, \( \lambda = 7, \psi = 0, \sigma = 2, \gamma = 1, \theta = 0^\circ, 45^\circ, 90^\circ, 270^\circ \) respectively in Equation 7. The \( L, a, b \) maps in different scales are convoluted with the DoG filter and Gabor filters and upscaled to the original scale, as shown in Equation 8.

\[
F^{(i)}_{(L,cs)} = \text{US}\left( L^{(i)} \ast \text{DoG} \right) \\
F^{(i)}_{(a,cs)} = \text{US}\left( a^{(i)} \ast \text{DoG} \right) \\
F^{(i)}_{(b,cs)} = \text{US}\left( b^{(i)} \ast \text{DoG} \right) \\
F^{(i)}_{(L,ori)} = \text{US}\left( L^{(i)} \ast \text{GF} \right) \\
F^{(i)}_{(a,ori)} = \text{US}\left( a^{(i)} \ast \text{GF} \right) \\
F^{(i)}_{(b,ori)} = \text{US}\left( b^{(i)} \ast \text{GF} \right)
\]

In our experiments, we used five scales and four orientations. Therefore, out of 15 color maps and 75 region contrast maps, a total of 90 feature maps were obtained, as shown in Figure 2.

**Feature-Prior**

In Equation 3, Feature-Prior is the relationship between a given feature set \( F^k \) appearing in position \( \rho \) with saliency value \( s \). One of the simplest methods to determine saliency is to average all the feature values. However, some features may be more important than others, so giving the same weight to all features is not appropriate and will give poor results. Instead, we use SVR to implement Feature-Prior. In this paper, we used \( \delta = SV/\bar{R} \) derived with the tool LIBSVM developed by Chang and Lin (2011). There are four steps to training an SVR model.

**Step 1.** Transfer the feature maps of training images to local feature maps and global feature maps.

\[
F^{\text{local}}_i = \frac{iF^k - \text{average}(F^k)}{\text{std}(F^k)/2}
\]

\[
F^{\text{global}}_i = \frac{iF^k - \text{average}(\text{set}(F^k))}{\text{std}(\text{set}(F^k))/2}
\]

Here, \( k = N, k \leq K, k \neq 0 \), \( iF^k \) denoted the \( k \)-th feature map of training image \( i \), \( \text{average}(\bullet) \) is the average operation of all elements, \( \text{std}(\bullet) \) is the operation to compute the standard deviation of all elements, \( \text{set}(F^k) = \{ F^k_1, F^k_2, \ldots, F^k_q \} \) is the set of the \( k \)-th feature map of all \( q \) training images. For each training images, training features are constructed by local features and global features, \( \text{training} F = [\text{local} F, \text{global} F] \). There are in total 180 training features in our implementation.

**Step 2.** As recommended by the SVR developer, all attributes should be in the range from 0 to 1 or from \(-1 \) to \( 1 \) for best training performance. Thus, all the feature maps were normalized from \(-1 \) to \( 1 \) by Equation 11.

\[
F^k = \frac{\text{training} F^k}{Z}, \quad Z = \max \left( \max \text{(training} F^k), \min \text{(training} F^k) \right)
\]

Here \( k = N, k \leq 2K, k \neq 0 \). It must be noted that the normalization operation will preserve the positive or negative symbols of the original values.

**Step 3.** Training samples \( \tilde{F}_\rho \) were selected from training images in position \( \rho \). \( \tilde{F}_\rho = [F^1_\rho, F^2_\rho, \ldots, F^{2K}_\rho] \) is the set of all the features in position \( \rho \). In order to present both the salient and non-salient regions, we chose the same number of training samples randomly from the top \( 20\% \) and bottom \( 70\% \) of the salient locations in each training image. For the Toronto database (N. Bruce & Tsotsos, 2005), 100 training samples were chosen from each of
the 96 training images, for a total of 9,600 training samples. For the Li 2013 database (Li et al., 2013), 50 training samples were chosen from each of the 188 training images, for a total of 9,400 training samples.

Step 4. All of the training samples were sent to LIBSVM and a SVR model was obtained. The training samples \( F \) were treated as attributes, and ground truth values at the same location \( q \), provided by the test databases providers, are treated as labels.

In the saliency computing process, the feature maps of a test image were extracted and normalized in the same manner as the learning process, and the features in each location were sent to the SVR model sequentially to obtain the predicted regression values. Feature-Prior matrix \( M_{FP} \) was defined as the set of the predicted regression values \( PR_p \) in every pixel of a test image.

\[
PR_p = Z \times SVR(F_p) \tag{12}
\]

\[
M_{FP} = \{ PR_p | \rho \in I \} \tag{13}
\]

where \( SVR(\bullet) \) is the SVR operation by trained model that returns the predict regression value. \( Z \) is the normalization parameter computed in Equation 11.

**Position-Prior**

As shown in Equation 3, Position-Prior presents human visual preference for locations in an image. We implemented Position-Prior using a simple statistical method: Sum up the values in the same position of density maps of training images, and normalize the result from zero to one. In the experiments, strong center bias can be observed in the Position-Prior matrix \( M_{PP} \).

\[
DM_{total} = \bigoplus_{i \in \text{training set}} DM_i \tag{14}
\]

\[
M_{PP} = \frac{DM_{total} - \min(DM_{total})}{\max(DM_{total}) - \min(DM_{total})} \tag{15}
\]
Here $DM_i$ represents the density maps of the $i$-th training image, and $\oplus$ denoted matrix addition. Equation 15 is the normalization operation. However, Position-Prior is not important and is expected to be removed in some situations and applications. In these cases, $M_{PP}$ is set to a matrix of all elements equals to 1. We implement this mode of $M_{PP}$ in our result SSM_1.

**Feature-Distribution**

Feature-Distribution represents the probability of saliency given a feature $F_{\rho}$ in a given image $I$. We can hypothesize that more the feature appears in an image, the less salient it will be, and vice versa. Thus, we implement Feature-Distribution using the pixel similarity computation and information theory. First, to enhance the probability of a feature appearing in an image, a similarity function of the $k$-th feature in position $\rho$ is computed as $H^k_{\rho}$, as seen in Equation 16.

$$H^k_{\rho} = \frac{1}{M \times N} \sum_{i \in I, \rho \neq \rho} Q(a|f^k_{\rho} - f^k_i| + 1),$$

$$Q(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

(16)

$M$ and $N$ are the horizontal and vertical size of the image. $a$ is a constant, and we used $a = -0.02$ in our implementation. Figure 3 shows the relationship between $(f^k_{\rho} - f^k_i)$ and $Q(a|f^k_{\rho} - f^k_i| + 1)$; it can be seen that the smaller the contrast between $f^k_{\rho}$ and $f^k_i$, the higher $Q$ value. This means that similar features will contribute energy to $f^k_{\rho}$. In other words, $H^k_{\rho}$ can be viewed as the appearance probability of a feature in position $\rho$ enhanced by all of the similar features in other positions of image $I$. Second, we assume from the information theory that the information carried by a point in a distribution is relative to its logarithm, so we define the energy of the pixel $\rho$ in $k$-th feature as equal to $-\eta \log(H^k_{\rho})$, where $\eta$ is a constant, which we set to 1 in this paper.

![Figure 3](https://jov.arvojournals.org/pdfaccess.ashx?url=/data/journals/jov/933550/)  
**Figure 3.** The relationship between $x$ and $Q(a|x| + 1)$ where $a = -0.02$.

![Figure 4](https://jov.arvojournals.org/pdfaccess.ashx?url=/data/journals/jov/933550/)  
**Figure 4.** An example of maps using to combine a saliency map. (a) Input image; (b) Feature-Prior matrix; (c) Feature-Distribution matrix; (d) Position-Prior matrix; (e) production of (b), (c), and (d); (f) saliency map.
\[ E^k_\rho = -\eta \log(H^k_\rho) \]  

(17)

Feature-Distribution in a position can be defined as the weighted sum of in all feature layers, as shown in Equation 18.

\[ R_\rho = \sum_{k=1}^{K} w_k \cdot E^k_\rho \]  

(18)

The weighting \( w_k \) reflects the importance of the \( k \)-th feature map and is directly proportional to the total energy contained in the \( k \)-th feature map.

\[ w_k = \sum_{\rho \in I} E^k_\rho \]  

(19)

Indeed, the weighting \( w_k \) is similar to the entropy of \( H^k_\rho \) for all positions. Finally, the Feature-Distribution matrix can be defined as the set of \( R_\rho \), as Equation 20 shows.

\[ M_{FD} = \{ R_\rho | \rho \in I \} \]  

(20)

### Property combination

The three matrices \( M_{FD}, M_{PP}, \) and \( M_{FP} \) are combined by an intersection operation and a convolution operation, as shown in Equation 21.

\[ SM = (M_{FD} \odot M_{PP} \odot M_{FP})*GF \]  

(21)

Here \( \odot \) denotes a Hadamard product and \( * \) denotes convolution operation. The parameter setting of the Gaussian filter is exactly the same as the ground truths production process if it is notified by database provider, or estimated by us if it is not notified by database provider. We used \( \sigma = 21 \) in the Toronto database and \( \sigma = 10 \) in the Li 2013 database. An example of maps using in the combination process is shown in Figure 4.

\[ M_{FD} = \{ R_\rho | \rho \in I \} \]

In this section, we evaluate the performance of the results computing by our approach using four metrics in two databases. Twelve state-of-art saliency models were chosen for comparison with our approach. These are denoted as: AIM (N. Bruce & Tsotsos, 2005; N. D. B. Bruce & Tsotsos, 2009), AWS (Garcia-Diaz, Fdez-Vidal, Pardo, & Dosil, 2012), Judd (Judd et al., 2009), Hou08 (Hou & Zhang, 2008), HFT (Li et al., 2013), itti-koch (Itti et al., 1998; the code we used is implemented by Harel et al., 2006), GBVS (Harel et al., 2006), SDSR (Seo & Milanfar, 2009a, 2009b), SUN (L. Zhang et al., 2008), STB (Walter & Koch, 2006), SigSal (Hou, Harel, & Koch, 2012), and CovSal (Erdem & Erdem, 2013; there were two implementation methods in CovSal, using only covariance feature and using both covariance and mean features, respectively and denoted as CovSal_1 and CovSal_2). Besides these 12 models, we also used the density map (ground truths) denoted as GT as the upper bound and the Gaussian model denoted as Gauss as the lower bound. The Gaussian model is set to \( 51 \times 51 \) size and \( \sigma = 10 \), then resized to equal to the original images, as Figure 5 shows. A saliency map of our approaches using uniform Position-Prior and learned Position-Prior are denoted as SSM_1 and SSM_2, respectively. Since our approach is a learning-based method and needs training images to train the SVR model to obtain saliency maps of the whole database, the 5-fold cross validation method is used. The method partitions the database into five subsets randomly that have the same number of images. Every subset is selected sequentially as a test set and the remainders serve as the training set. There are 96 images and 188 images in a training set for the Toronto database and Li 2013 database respectively; thus, 9,600 training samples and 9,400 training samples are selected to train a SVR model for these two databases respectively. Because of the randomness, the validation process is performed 10 times and using the average value as our performance.

In our implementation, all images are resized to 320 \( \times \) 240 in the computation process and saliency maps are resized back to original size of the input image to save computational effort.

### Evaluation metric

Five metrics are used to evaluate the performance of models, including Receiver Operating Characteristic (ROC), shuffled Area Under Curve (sAUC), Normalized Scanpath Saliency (NSS), Earth Movers Distance (EMD), and Similarity Score (SS).

### Receiver Operating Characteristic

ROC is the most widely used metric for evaluating visual saliency. The density maps constructed from subjects’ fixation data are treated as ground truths, and saliency maps computed from algorithms are treated as binary classifiers under various thresholds. In Equation 22, the computation of the true-positive rate (TPR) and the false-positive rate (FPR) in each threshold are shown, and a curve can be plotted by the set of these two values. Computing the area under curve (AUC) can serve to represent how close the two distributions are.
Here, TP is true positive, FN is false negative, FP is false positive, and TN is true negative. The two distributions are exactly equal when AUC is equal to 1, not relative when AUC equal to 0.5, and negative relative when AUC equal to 0. However, there are three problems with the ROC curve. First, as long as TPR is high, AUC will be high regardless of FPR. This means saliency maps, which have high salient values, perhaps have a better AUC. The second problem is that AUC cannot represent the distance error of prediction positions. When the predicted salient position misses the actual salient position, AUC will be the same whether they are close in distance or not. The third problem is that AUC is suitable for evaluating two distributions, but the fixation maps recorded by eye tracker equipment are generally dispersed. In practice, the fixation maps are usually transferred to density maps by convoluted the fixation maps with a Gaussian filter. However, different parameters of the Gaussian filter sometimes lead to different results. For these three reasons, we also used other metrics to evaluate performance.

**Normalized Scanpath Saliency**

NSS is a metric to evaluate how accurately a saliency map predicts the fixation points. Saliency maps are normalized to have zero mean and unit standard deviation, so the definition of NSS is the average value of the fixation points response on saliency maps. Higher NSS means a saliency map predicts the position of the fixation points more accurately. NSS equal to zero means a saliency map is predicting the fixation points by chance. This metric uses the fixation maps (which contain all the fixation points) instead of density maps, so the influence of convolution with Gaussian filter can be avoided.

**Earth Movers Distance**

EMD represents the minimum cost of change a distribution to another distribution. In this study, we use the fast implementation of EMD provided by Pele and Werman (2008, 2009).

\[
EMD(P, Q) = \left( \min_{\{f_{ij}\}} \sum_{i,j} f_{ij} d_{ij} \right) \left[ \sum_i P_i - \sum_j Q_j \right] + \max_{i,j} d_{ij}
\]

\[
\text{s.t. } f_{ij} \geq 0, \sum_j f_{ij} \leq P_i, \sum_i f_{ij} \leq Q_j, \sum_{i,j} f_{ij} \\
= \min \left( \sum_i P_i, \sum_j Q_j \right)
\]

where \( f_{ij} \) presents the amount transported from \( i \)-th supply to the \( j \)-th demand. \( d_{ij} \) is the ground distance between bin \( i \) and bin \( j \) in the histogram. EMD equal to zero means the two distributions are identical; a larger EMD means the two distributions are more different.
Similarity Score

SS is another metric for measuring the similarities of two distributions. It first normalizes two distributions to let the sum equal to one, then to sum the minimum value in each position.

\[
SS = \sum_{i,j} \min(P_{ij}, Q_{ij}),
\]

Where

\[
\sum_{i,j} P_{ij} = \sum_{i,j} Q_{ij} = 1
\]

SS is always between zero and one. SS equal to one means two distributions are identical and SS equal to zero means two distributions are totally different.

Database description

We used two eye fixation databases to evaluate our results. The first eye fixation database is the Toronto database (N. Bruce & Tsotsos, 2005). It contains the results of 120 color images with resolutions of 681 x 511, presented 4 s each in random order on a 21-in. monitor with 1024 x 768 pixels posted of 0.75 m before the 20 subjects. The subjects were asked to perform a free-viewing task, and an eye tracker recorded the positions that subjects focused on. The database provided density maps as ground truth recorded the positions that subjects focused on. The database not only provides human fixation records but also human labeled results. The database provided density maps constructed from the fixation positions convoluted with a Gaussian filter. The second test database is the Li 2013 database (Li et al., 2013). It contains 235 color images divided into six categories: images with large salient region (50 images), intermediate salient region (80 images), small salient region (60 images), cluttered backgrounds (15 images), repeating distractors (15 images), and both large and small salient region (15 images). The image resolution is 480 x 640. The images were shown in random order on a 17-in. monitor, with each of the 21 subjects positioned 0.7 m from the monitor when asked to perform the free-viewing task. The database not only provides human fixation records but also human labeled results. The fixation results contain both the fixation positions and density maps constructed from the fixation positions and the Gaussian filter.

Evaluation performance

Because the training samples were selected randomly, we performed the 5-fold cross validation 10 times to analyze stability. The statistical results are shown in Tables 1 and 2. The difference of the best case and the worst case is small, and the standard deviation for the 10 test times is similarly small. These results show that the randomness in sample selection had little influence on the results, and our method is robust. Table 3 shows the comparison of evaluation performance of the 12 models in the Toronto database. In our results (SSM_1 and SSM_2), the average value of 10 times 5-fold cross validation in Table 1 are used for comparison. Table 3 shows that SSM_2 has the best performance in AUC, NSS, and SS, with values of 0.934, 1.853, and 0.577, respectively. AWS has the best performance in sAUC with a value of 0.705. However, sAUC of SSM_1 is 0.687, close to the best result. STB has the best performance in EMD with a value of 1.628, while SSM_2 has the third best performance at 2.186. Table 4 shows the comparison of evaluation performance of the 12 models in the Li 2013 database. Like the experiment in the Toronto database, the average value of the 10 times five-fold cross validation in Table 2 is used to represent our model’s results. In this experiment, SSM_2 has the best value in AUC, NSS, and SS with values of 0.941, 1.891, and 0.559, respectively. AWS still has the best performance in sAUC at 0.685. SSM_1 has the second best value as 0.668. STB has the best performance in EMD as 0.946, when SSM_1 has the second best performance as 1.132. Generally speaking, SSM_1 and SSM_2 have good performance in these five metrics. The advanced discussion of the results is described in the next section. Figure 6 presents some examples of the saliency maps produced by our approach and the other 12 saliency models in the Toronto database and Li 2013 database. In these examples, we can find that our saliency maps are more similar with the ground truths than other models’ saliency maps, regardless of if the saliency regions are large or small in ground truths.

Discussion

Influence of center bias in AUC and sAUC

Tables 3 and 4 show that both AUC and NSS are heavily influenced by center bias since the Gaussian model gets above-average scores in AUC and NSS. sAUC is not influenced by center bias, since it stays within 0.5 according to the Gaussian model. However, the use of sAUC does not always match the intuitive sense. There are two reasons for this. First, if the fixation points of an image are clustered near the center, the minority of off-center fixation points will dominate the results. This is because the importance of the fixation points located in the center is strongly reduced by many fixation points of other images that are also located in the same place. The fixation points...
### Table 1. Statistics of running our method 10 times in the Toronto database.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>GT</th>
<th>Gauss</th>
<th>AIM</th>
<th>HFT</th>
<th>ittikoch</th>
<th>GBVS</th>
<th>SUN</th>
<th>SDSR</th>
<th>Judd</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>1.000</td>
<td>0.884</td>
<td>0.784</td>
<td>0.910</td>
<td>0.871</td>
<td>0.915</td>
<td>0.715</td>
<td>0.849</td>
<td>0.922*</td>
</tr>
<tr>
<td>sAUC</td>
<td>0.822</td>
<td>0.500</td>
<td>0.659</td>
<td>0.664</td>
<td>0.652</td>
<td>0.636</td>
<td>0.611</td>
<td>0.613</td>
<td>0.615</td>
</tr>
<tr>
<td>NSS</td>
<td>3.210</td>
<td>1.250</td>
<td>0.882</td>
<td>1.637*</td>
<td>1.290</td>
<td>1.514</td>
<td>0.578</td>
<td>1.213</td>
<td>1.381</td>
</tr>
<tr>
<td>EMD</td>
<td>0.000</td>
<td>5.708</td>
<td>7.754</td>
<td>2.985</td>
<td>5.968</td>
<td>4.911</td>
<td>5.425</td>
<td>5.417</td>
<td>11.238</td>
</tr>
<tr>
<td>SS</td>
<td>1.000</td>
<td>0.473</td>
<td>0.383</td>
<td>0.506*</td>
<td>0.448</td>
<td>0.488</td>
<td>0.343</td>
<td>0.442</td>
<td>0.407</td>
</tr>
</tbody>
</table>

### Table 2. Statistics of running our method 10 times in the Li 2013 database.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>SSM_1</th>
<th>SSM_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.861</td>
<td>0.861</td>
</tr>
<tr>
<td>sAUC</td>
<td>0.689</td>
<td>0.689</td>
</tr>
<tr>
<td>NSS</td>
<td>1.297</td>
<td>1.278</td>
</tr>
<tr>
<td>EMD</td>
<td>5.519</td>
<td>5.319</td>
</tr>
<tr>
<td>SS</td>
<td>0.439</td>
<td>0.435</td>
</tr>
</tbody>
</table>

### Table 3. Ten models’ performance comparison of the Toronto dataset. Notes: ** denotes the best result and * denotes the second best result of all besides GT and Gauss.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>GT</th>
<th>Gauss</th>
<th>AIM</th>
<th>HFT</th>
<th>ittikoch</th>
<th>GBVS</th>
<th>SUN</th>
<th>SDSR</th>
<th>Judd</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td>1.000</td>
<td>0.866</td>
<td>0.817</td>
<td>0.928</td>
<td>0.900</td>
<td>0.930</td>
<td>0.745</td>
<td>0.866</td>
<td>0.937*</td>
</tr>
<tr>
<td>sAUC</td>
<td>0.746</td>
<td>0.500</td>
<td>0.634</td>
<td>0.645</td>
<td>0.642</td>
<td>0.636</td>
<td>0.602</td>
<td>0.658</td>
<td>0.615</td>
</tr>
<tr>
<td>NSS</td>
<td>3.401</td>
<td>1.252</td>
<td>0.940</td>
<td>1.774*</td>
<td>1.443</td>
<td>1.641</td>
<td>0.735</td>
<td>1.269</td>
<td>1.472</td>
</tr>
<tr>
<td>EMD</td>
<td>0.000</td>
<td>5.468</td>
<td>6.658</td>
<td>2.281</td>
<td>4.739</td>
<td>4.199</td>
<td>7.849</td>
<td>4.923</td>
<td>9.748</td>
</tr>
<tr>
<td>SS</td>
<td>1.000</td>
<td>0.466</td>
<td>0.377</td>
<td>0.514*</td>
<td>0.460</td>
<td>0.494</td>
<td>0.351</td>
<td>0.434</td>
<td>0.408</td>
</tr>
</tbody>
</table>

### Table 4. Ten models’ performance comparison of the Li 2013 dataset. Notes: ** denotes the best result and * denotes the second best result of all besides GT and Gauss.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>GT</th>
<th>Gauss</th>
<th>AIM</th>
<th>HFT</th>
<th>ittikoch</th>
<th>GBVS</th>
<th>SUN</th>
<th>SDSR</th>
<th>Judd</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC</td>
<td>1.000</td>
<td>0.866</td>
<td>0.817</td>
<td>0.928</td>
<td>0.900</td>
<td>0.930</td>
<td>0.745</td>
<td>0.866</td>
<td>0.937*</td>
</tr>
<tr>
<td>sAUC</td>
<td>0.896</td>
<td>0.867</td>
<td>0.692</td>
<td>0.881</td>
<td>0.905</td>
<td>0.883</td>
<td>0.916</td>
<td>0.941*</td>
<td>0.941**</td>
</tr>
<tr>
<td>NSS</td>
<td>1.493</td>
<td>1.405</td>
<td>0.978</td>
<td>1.433</td>
<td>1.539</td>
<td>1.274</td>
<td>1.601</td>
<td>1.891**</td>
<td>1.891**</td>
</tr>
<tr>
<td>EMD</td>
<td>4.386</td>
<td>1.684</td>
<td>0.946**</td>
<td>6.157</td>
<td>3.702</td>
<td>12.390</td>
<td>2.717</td>
<td>1.132*</td>
<td>1.132*</td>
</tr>
<tr>
<td>SS</td>
<td>0.436</td>
<td>0.452</td>
<td>0.202</td>
<td>0.432</td>
<td>0.457</td>
<td>0.349</td>
<td>0.463</td>
<td>0.559**</td>
<td>0.559**</td>
</tr>
</tbody>
</table>
lose their credit when they belong to both positive points and negative points. Second, sAUC cannot respond to the fixation density. Fixations are considered with the same importance whether they appear in a large group or as solitary points. Density maps show the aggregation of fixations since they are produced by convolution of fixations and a Gaussian filter. For example, as shown in Figure 7b and c, some solitary fixations appeared in fixation map but not in density map. Figure 6d through f shows the saliency map produced by AWS, SSM_1, and SSM_2. The saliency map of SSM_2 correctly marked the center region and with highest AUC and lowest sAUC. The saliency map of AWS marked many regions and with the lowest AUC and highest sAUC. The average saliency values of Figure 6d through f are 67.32, 24.68, and
8.44, respectively. In this case, it seems that choosing more off-center regions as salient might result in a higher possibility of obtaining a higher sAUC. On the other hand, if our aim is to approach human visual behavior, considering naturally occurring center bias, is perhaps more reasonable. Although sAUC is powerful for evaluating saliency maps while ignoring center bias, for a comprehensive evaluation, it might be better to consider all the other representative metrics, regardless of whether they are influenced by center bias.

Modification of learning-based saliency models

As described in the Introduction section, some computational saliency models have the ability to learn adaptive parameters from training samples. However, we used default settings for every comparison model in previous section evaluations. For fair comparison, some of the models should be modified by a training process. Three models are modified: AIM, Hou08, and Judd. Among these, AIM used principal component analysis (PCA) and ICA techniques, Hou08 used the ICA technique, and Judd used the SVM technique. The details of training processes of these three models are illustrated in the Appendix. The result is shown in Table 5. From these, we can observe that some results are better and some are worse than using default parameter settings. Nevertheless, the difference is small and does not change the performance rank of the models, which may be caused by the default parameters from learning of a large set of training images, but the training images in our database are limited. The Judd model used the SVM technique as did our method, but two main differences distinguish these two models. First, the Judd model used 33 different features, including some high-level features such as face, human, and car detectors. Although these detectors are useful, they came with a heavy computational effort and cost. Our model used only low-level features and imposed less computational effort and cost. Second, our model considered the frequency of features appearing in an image, but the Judd model did not. Furthermore, from Tables 3, 4, and 5, our model’s performance was better than Judd’s in almost all metrics.

Size of the salient region and the ROC curve

The size of the salient region plays an important role in the ROC method. Since it cannot separate salient regions from backgrounds using a certain threshold, the ROC method treats a saliency map as a binary classifier for ground truth under various thresholds. AUC is the average value of many ROC curves built by these thresholds. It cannot indicate the performance of a saliency map in certain sizes of salient regions, and
may not be representative of real performance in all cases. As a result, we assume the salient region is in certain percentages of the image, binarizing the ground truths and saliency maps under the percentages, then computing the TPR. This method is also used by Judd et al. (2009), and noted as AUC Type 2 by Borji et al. (2013). Figures 8 and 9 present the TPR of 13 models under different percentages of salient regions in two databases we examined. TPR was shown to increase when the salient region expands. In the Toronto database, TPR of SSM_2 was higher than all other models from the 5% to 50% salient region, as shown in Figure 8. In the Li 2013 database, TPR of SSM_2 was higher than other models from the 5% to 30% salient region, GBVS got higher TPR when the salient region was larger than 30%, as shown in Figure 9. This means when the definition of the salient region changes, the rank of performance using the ROC method may change. However, the salient region in an image is generally under 50%, even smaller in many cases. As a result, the performance of our method is better than other models, especially when the salient region is small.

### Conclusion

In this paper, we proposed a new visual saliency model using the Bayesian probability theory and machine learning techniques. Based on four visual attention assumptions, three properties are extracted and combined by a simple intersection operation to obtain saliency maps. Our model is unlike traditional contrast-based bottom-up methods in that its learning mechanism has the ability to automatically learn the relationship between saliency and features. Moreover, unlike existing learning-based models that only consider the components of features themselves, our model simultaneously considers appearing frequency of features, information contained in the image, and the pixel location of features, which intuitively have a strong

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Toronto database</th>
<th>Li 2013 database</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIM</td>
<td>Hou08</td>
</tr>
<tr>
<td>AUC</td>
<td>0.757</td>
<td>0.853</td>
</tr>
<tr>
<td>sAUC</td>
<td>0.637</td>
<td>0.625</td>
</tr>
<tr>
<td>NSS</td>
<td>0.775</td>
<td>1.302*</td>
</tr>
<tr>
<td>EMD</td>
<td>7.526*</td>
<td>1.776*</td>
</tr>
<tr>
<td>SS</td>
<td>0.376</td>
<td>0.418</td>
</tr>
</tbody>
</table>

Table 5. Average performance of three learning models after training process and 10-fold cross validation. Note: *Denotes the better result than original parameters.
influence on saliency. As a result, our model can determine saliency regions more precisely. Experimental results indicate that the proposed model has significantly better performance than 12 other state-of-art models. However, the proposed model is not perfect and still has some shortcomings. One of these is it needs samples to train the SVR model, so it may be weak for novel or unusual scenes that it has not seen before. It is also weak with certain human preference objects and scenes, such as faces, humans, and vehicles, which are generally believed to come from a high-level, top-down attention mechanism. The other shortcoming is the heavy computational cost of SVR in computing pixel-by-pixel. Future improvements to three areas are worth pursuing. The first is to try to reduce computational effort by simplifying some parts of the algorithms and their implementation. The second is to expand the approach to the spatial-temporal form and implement it with video. The last one is to consider relevant high-level features and tasks for achieving top-down saliency.

Keywords: selective attention, visual saliency, support vector regression, center bias, information theory, Bayesian theory

Acknowledgments

Commercial relationships: none.
Corresponding author: Ru-Je Lin.
Email: d95921005@ntu.edu.tw.

Address: Department of Electrical Engineering, National Taiwan University, Taiwan.

References


**Appendix**

For the three learning-based saliency models including AIM, Hou08, and Judd, we used 5-fold cross validation 10 times to evaluate their performance, the same method used to evaluate our method, indicating 96 images in the training set and 24 images in the testing set for each fold in the Toronto database, and 188 images in the training set and 47 images in the testing set for each fold in the Li 2013 database. The average value of 10 trials was used to determine the final performance. The training process for each of the three models is described as following.

**AIM:** Training involved random selection of 100 21 × 21 size patches from each of the images of the training sets. FastICA (download from this site: http://research.ics.aalto.fi/ica/fastica/) was applied with PCA for preprocessing, thereby retaining 95% variance. This training process retains 191–211 filters for the Toronto database, and 31–38 filters for the Li 2013 database. These filters were used to produce saliency maps.

**Hou08:** Training involved random selection of 100 8 × 8 size patches from each of the images of the training sets. FastICA was applied to obtain a set of 8 × 8 × 3 = 192 basis functions. The estimated sparse basis A and the bank of filter functions W were recorded and used to produce saliency maps.

**Judd:** The training program provided by Judd was used. It involved random selection of 50 positively labeled pixels from the top 20% of the salient locations and 50 negatively labeled pixels from the bottom 70% of the salient locations in each training image, then used all the selected pixels and their salient values to train a SVM model. The SVM model will decide the combined weightings of 33 features in producing saliency maps.