AN IMPROVED SALIENCY MECHANISM FOR COMPUTER VISION

By

MOHSEN EMAMI

Bachelor of Science in Mechanical Engineering
Sharif University of Technology
Tehran, Iran
2005

Master of Science in Mechanical Engineering
Sharif University of Technology
Tehran, Iran
2008

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2013

AN IMPROVED SALIENCY MECHANISM FOR

COMPUTER VISION

Dissertation Approved:

Dr. Larry L. Hoberock

Dissertation Adviser

Dr. Jay Hanan

Dr. Gary Young

Dr. Damon Chandler

ACKNOWLEDGEMENTS

For his ingenious suggestions, ideas and supports, I wish to express my sincere gratitude to my adviser Professor Larry Hoberock. He always sees the best in people and is generous, caring and wise like nobody I have ever met. Without his constant guidance and persistent help this dissertation would not have been possible.

I would like to thank my thesis committee members, Dr. Jay Hanan, Professor Gary Young and Dr. Damon Chandler whose insightful comments and creative suggestions added much value to my work.

I am also grateful to my fellow labmates Dr. Trung Duong and Dr. Venu Lolla who helped me understand many new theories and implement them in my research.

I would like to thank my wife, Masoumeh Sheikhloo, whose constant support has been a source of great encouragement. She makes my heart warm and cheers me up in good times and bad. Last, but by no means least, I thank my parent, who always stood by me and supported me in any possible way.

Name: MOHSEN EMAMI

Date of Degree: JULY, 2013

Title of Study: AN IMPROVED SALIENCY MECHANISM FOR COMPUTER VISION

Major Field: Mechanical Engineering

Abstract: The objective of this project is to find an efficient biologically plausible model for the bottom-up saliency mechanism of the human vision system (HVS) and employ it in computer vision applications. In practice, analyzing or storing all information entering the human eye at every moment is beyond the capabilities of the human neural system [1]. The saliency mechanism controls the process of selecting and allocating attention to the most "prominent" locations in the scene [2], which are mostly referred to as "salient points" or "interesting points" in the literature. The same problem of information overload exists in most of the computer vision applications and an efficient visual saliency model can help reducing time consumption of the algorithm. These applications comprise, but are not limited to, automatic target detection, robotics and image and video compression.

In the report herein, the general architecture of models of the HVS saliency mechanism is presented and some of the well-known models are illustrated. There are several metrics to compare saliency models; however, results from different metrics vary widely in evaluating models. Since it is important to know which models perform the best in mimicking the saliency mechanism of the human visual system, first a procedure is proposed for evaluating metrics for comparing saliency maps using a database of human fixations on approximately 1000 images. This procedure is then employed to identify the best metric. This best metric is then used to evaluate nine published bottom-up saliency models on two databases, one containing natural images and the other synthetic ones.

Furthermore, a new method for normalizing feature saliency maps in the saliency detection mechanisms is introduced. Also, the best visual saliency model in the literature is modified to overcome some deficiencies by automatically selecting different parameters for different regions of the image. As an application of the models of the saliency mechanisms, a saliency mechanism with the new normalization method is then applied to dishware inspection that shows interesting results.

Keywords: Bottom-up saliency mechanism, Top-down saliency, Computer vision applications, Human visual system, Automatic dish inspection.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

1 INTRODUCTION

Many researchers in the past decades have suggested employing physiological and psychological aspects of the human vision system (HVS) in computer vision algorithms [2-6]. An important aspect of the HVS is the visual saliency mechanism [2, 7]. This mechanism controls the process of selecting and allocating attention to the most "prominent" locations in the scene [2], which are mostly referred to as "salient points" or "interesting points" in the literature.

In practice, analyzing or storing all information entering the human eye at every moment is beyond the capabilities of the human neural system [1]. Controlling fixations and saccades[1] of the eye, the visual saliency mechanism enables the HVS to focus its limited perceptual and cognitive resources on the most important locations of the scene [8]. It has been shown that the HVS gathers information mostly at the fixation points and little information is collected in saccades [5]. As a result, the saliency mechanism allows the HVS to interact with the visual environment efficiently and extract only useful information from the scene [9]. Moreover, it helps the visual perceptual system to organize visual information faster [2, 9, 10]. Additional properties of different phases of human eye movement and information gathered during each phase are given in [5, 8].

The information overload problem exists in computer vision applications too. Models of the

[1] Fast eye movements.

saliency mechanism can be employed in the beginning step of many computer vision algorithms to find the prominent regions of the image. This way, only those regions can be examined thoroughly and the computation time of the vision algorithm can be reduced.

The saliency mechanism in the human vision system is an interaction between two mechanisms, bottom-up and top-down saliency mechanisms [1, 2, 9]. Bottom-up saliency is a fast and purely stimulus driven mechanism (independent of any high-level visual task) which biases the observer towards selecting locations in the scene based on the saliency of the locations only [2]. In this case, the saliency of a stimulus can be defined as the state or quality of standing out relative to other stimuli in the scene. Top-down saliency is a slower mechanism and a memory dependent process. It directs the visual attention based on activities in which the human neural system is engaged [1]. Given the same scene, salient points and patterns of saccades change for different questions that were asked of observers prior to viewing the scene, which is believed to be a property of top-down saliency [2, 5].

1.1 Computational Models for Saliency Mechanism of the HVS

Herein, we intend to investigate models of bottom-up saliency mechanisms of the HVS. Several bottom-up models are explained in Section 1.1.1. Nine well-known models are selected to be applied on an image database for a thorough performance evaluation in chapter III. Also, some top-down saliency models are explained in 1.1.2.

1.1.1 Bottom-Up Saliency Mechanism Models

There has been increasing effort to present computational principles of the HVS saliency mechanism in the last decades. According to Harel, et al. [11], models of the bottom-up visual saliency can be organized into the following three stages:

1. ***Extraction***: Given an image of the scene, several feature spaces such as image intensity, orientation and color are extracted by linearly filtering the input image.

2. ***Activation***: Computing feature saliency maps (activation maps) from feature spaces.
3. ***Normalization***/***Combination***: Normalizing feature saliency maps and combining them together to form the saliency map.

While different models usually share the same method for extraction, they use different approaches for activation and normalization.

1.1.1.1 Koch and Ullman Visual Saliency Model

One of the first models was proposed by Koch and Ullman in 1985 [12]. They modeled the bottom-up saliency mechanism of the HVS in three main steps. In step 1, a set of elementary feature spaces, such as image intensity, color and orientation is computed in parallel across the visual field. Each feature space is analyzed using the center-surround approach to produce feature saliency maps. Applying this approach, locations in visual space that differ from their immediate surroundings, with respect to the elementary feature, gain larger values in the corresponding map. A linear combination of the feature saliency maps results in a saliency map, in which points with larger magnitudes are considered to be more salient. In the second step, a winner-takes-all (WTA) mechanism chooses the most salient location. Finally, in the third step, using an inhibitory mechanism, the WTA mechanism shifts automatically to the next most salient location.

The Koch and Ullman algorithm is based on the "Feature Integration Theory" of Treisman and Gelade [13]. According to this theory, features are extracted early, automatically, and in parallel across the visual field, while objects are identified separately and only at a later stage, which requires focused attention. Treisman and Gelade assumed that at the first step of visual processing in the HVS, several feature spaces such as color, orientation, spatial frequency, brightness, and direction of movement are initially extracted from the scene. Then, feature spaces are processed in parallel to generate feature saliency maps that are later integrated in a saliency map. The

saliency map then will be used to direct attention to the most important parts of the scene. They claimed that without focused attention, features cannot be related to each other.

1.1.1.2 Itti and Koch Visual Saliency Model (IK)

Following the general framework of Koch and Ullman, other researchers have presented different models to generate saliency maps [2, 6, 14, 15]. Itti et al. [6] further developed the Koch and Ullman model. As illustrated in Figure 1-1, they considered one intensity, two color, and four orientation feature spaces, which are the most important feature spaces based on work by Wolfe et al. [16]. Then, feature spaces are analyzed with six sets of radii for center and surround circles, which results in forty two feature saliency maps: six for intensity, twelve for color, and twenty four for the orientation feature spaces. Itti et al. [6] proposed to normalize feature saliency maps before combining them together and presented a new normalization method, which is the main difference from the Koch and Ullman model. Normalizing and summing feature saliency maps corresponding to each elementary feature space results in three "conspicuity maps". Finally, the saliency map is calculated as an average of the three conspicuity maps. Similar to the Koch and Ullman model, the winner-takes-all (WTA) neural network, with a local inhibition of return, is then employed to select the peaks in the saliency map.

Itti and Khoch [1] further developed [6] and introduced a more efficient feature saliency map normalization method. Since the Itti and Khoch [1] model of visual saliency mechanism is being used by many researchers to compare with their saliency model, we select this model to be applied on the image dataset for our performance evaluation.

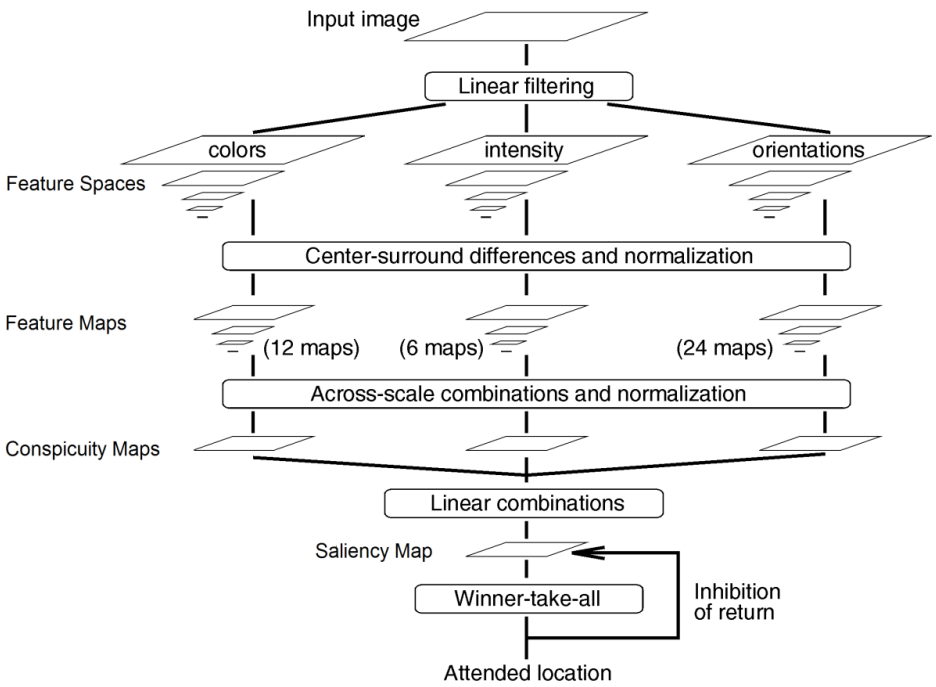


Figure 1-1. Itti et.al model general architecture [6].

1.1.1.3 Coherent Computational Approach (CC)

Meur et al. [17] proposed a model based on the architecture of the Koch and Ullman model [12] to overcome the following drawbacks of the classical models of visual saliency such as the Itti et al. [6] model:

1. Applying several normalization steps during the process
2. Ineffective normalization methods
3. Ignoring or overlooking some aspects of HVS saliency mechanism

Meur et al. [17] introduced the flow chart shown in Figure 1-2 which presents three aspects of the HVS saliency mechanism: visibility, perception, and perceptual grouping. In their model, the visibility part, which is also called the psychovisual space, simulates the limited sensitivity of the human eyes and takes into account the major properties of the retinal cells. Also, they included a perception unit to suppress the redundant visual information by simulating the behavior of cortical cells. In the last unit, the saliency map building is achieved by the perceptual grouping.

This method will be selected later for our performance evaluation.

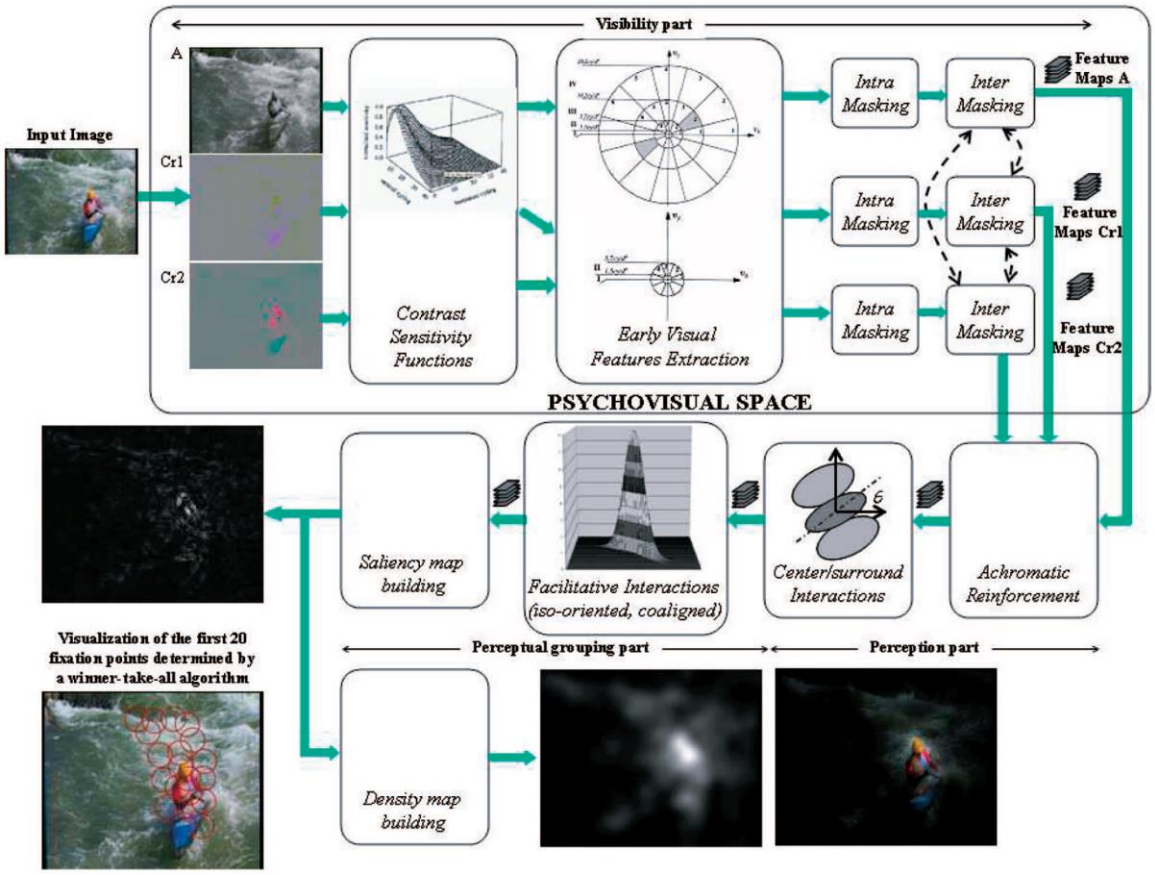


Figure 1-2. Meur et al. flow chart of the computational model of bottom-up visual saliency mechanism [17].

1.1.1.4 Visual Object detection with a Computational attention System

Frintrop et.al [18, 19] introduced a model of the saliency mechanism based on the Koch and Ullman [12] and Itti et al. [6] saliency mechanisms and named it VOCUS (Visual Object detection with a CompUtational attention System), shown in Figure 1-3. They similarly used three feature spaces namely intensity, color and orientation. Instead of rescaling the filters, they rescaled the image, resulting in reduced computational time. Unlike the Koch and Ullman and Itti et al. models, Frintrop et al. computed on-center and off-center differences separately in the center-surround mechanism. In the Koch and Ullman and Itti et al. methods, the faster approach of taking the absolute value of the difference ($center - surround$) is used, which causes some problems. For example in an image with a gray background, with one white and several black

dots, the white dot pops out in human perception. But the Koch and Ullman, and Itti et al. models calculate the same value of saliency for all black and white dotes. To integrate different feature saliency maps into a saliency map, Frintrop et al. first weighted each map by a uniqueness weight function, and then summed up the weighted feature saliency maps, similar to the Itti and Koch model [14]. In Figure 1-3, the most salient part of the input image is shown in the red circle in the output image.

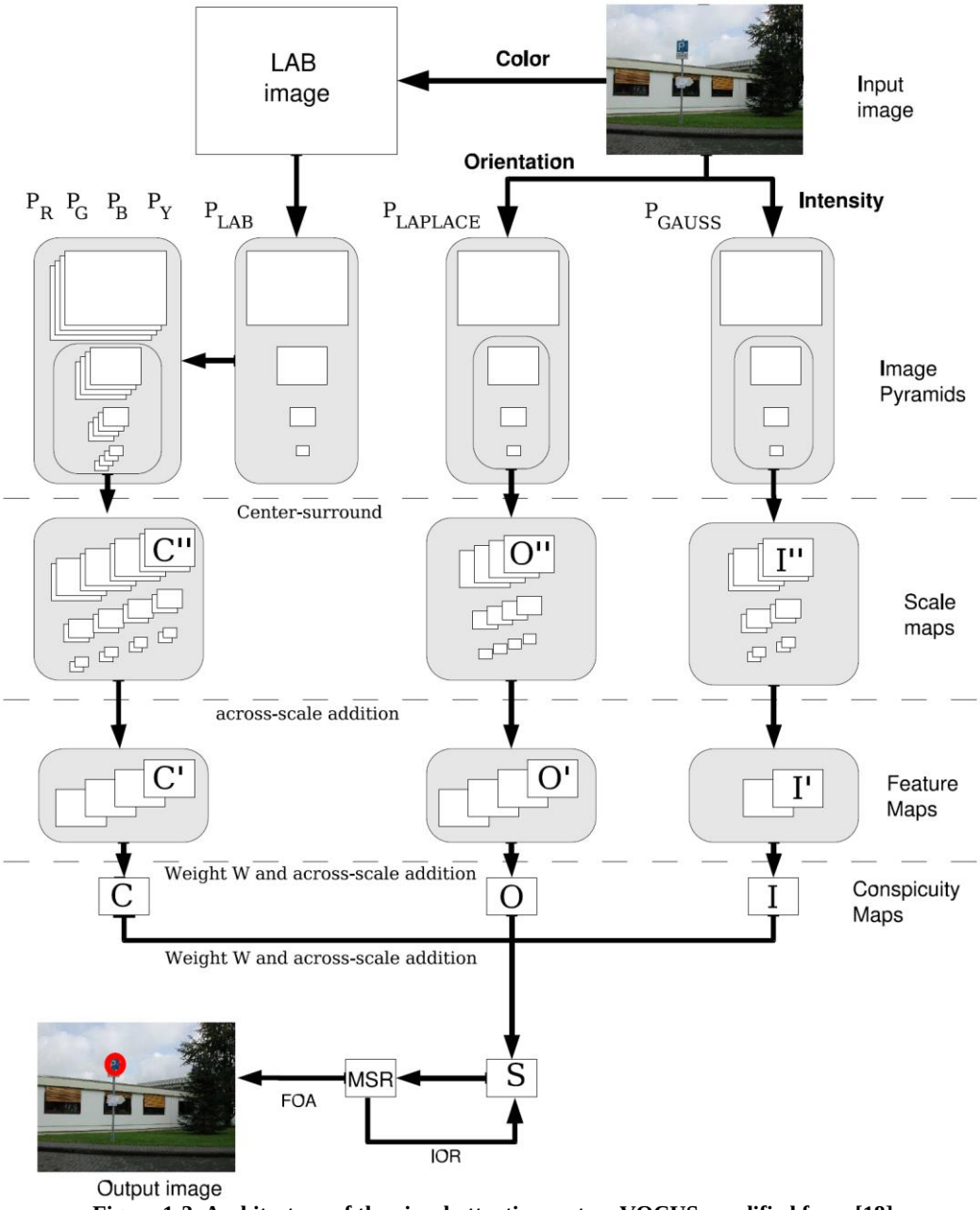


Figure 1-3. Architecture of the visual attention system VOCUS, modified from [18].

1.1.1.5 Graph Based Visual Saliency (GBVS)

Harel, et al. [11] introduced a new approach for modeling bottom-up visual saliency. To calculate the feature saliency map ($\boldsymbol{M}$) corresponding to a feature space ($\boldsymbol{F}$), the authors first generated a fully connected graph $\boldsymbol{G}$, obtained by connecting every two pixels in $\boldsymbol{F}$. Then, a weight $\boldsymbol{w}$ was assigned to the edge from pixel (i,j) to (p,q), defined by:

$$\boldsymbol{w}\big((i,j),(p,q)\big) = \left| log\,\frac{\boldsymbol{F}(i,j)}{\boldsymbol{F}(p,q)} \right| . exp\left(-\frac{(i-p)^2+(j-q)^2}{2\sigma^2}\right) \qquad (1\text{-}1)$$

where the first term on the right is called the dissimilarity between $\boldsymbol{F}(i,j)$ and $\boldsymbol{F}(p,q)$ and the second term is a Gaussian function to increase the weight of two close pixels and decrease the weight of pixels which are far from each other. Subsequently, a Markov chain is defined on $\boldsymbol{G}$ by normalizing the weights of the outbound edges of each node/pixel to 1. The equilibrium distribution of the Markov chain reflects the fraction of time a random walker would spend at each node if he were to walk forever [11]. It would naturally accumulate mass at nodes that have high dissimilarity with their surrounding nodes. This equilibrium distribution is considered the feature saliency map in the Harel, et al. approach. This model also will be selected later for our performance evaluation.

1.1.1.6 Saliency Using Natural Statistics (SUN)

Zhang et al. [20] proposed a visual saliency model using a Bayesian framework from which bottom-up saliency is defined as the self-information of visual features, and prior information emerges as the pointwise mutual information between the features and the target when searching for a target. Differing from most of the visual saliency models, Zhang et al. defined saliency based on the natural image statistics instead of considering the image of interest only. Let the binary random variable C represent whether or not a pixel location L belongs to the target. To calculate the feature saliency map value at pixel location (i,j) ($\boldsymbol{M}(i,j)$), using the corresponding feature space $\boldsymbol{F}(i,j)$, they offered the following equation:

$$log[\boldsymbol{M}(i,j)] = -log\left[p\left(F = \boldsymbol{F}(i,j)\right)\right] + log[p(F = \boldsymbol{F}(i,j)|C = 1)] + log\left[p\left(C = 1|L = (i,j)\right)\right] \quad (1\text{-}2)$$

The first part on the right hand side of (1-2) ($-log\left[p\left(F = \boldsymbol{F}(i,j)\right)\right]$) is known as self-information of the random variable F when it takes the value $\boldsymbol{F}(i,j)$. It increases when the probability of a feature decreases (the rarer a feature, the more informative it is). The second part is a log-likelihood term which support feature values that are consistent with the prior knowledge about the target. The third part on the right hand side of (1-2) is independent of visual features and reflects the prior knowledge about where the target is more likely to appear. The performance of this model is evaluated in Chapter III against eight other bottom-up saliency models.

1.1.1.7 Frequency-tuned Salient Region Detection (FTSRD)

Achanta et al. [21] introduced a simple definition for computing the feature saliency map $\boldsymbol{M}$ using the feature space $\boldsymbol{F}$ as follows

$$\boldsymbol{M} = |\mu_{\boldsymbol{F}} - \boldsymbol{F}_{whc}| \quad (1\text{-}3)$$

where $\mu_{\boldsymbol{F}}$ is the mean value of $\boldsymbol{F}$ and $\boldsymbol{F}_{whc}$ is a smoothed version of $\boldsymbol{F}$ with a 2-D Gaussian kernel. They used the Euclidian length of the vector $\{\boldsymbol{M}_1(i,j), \boldsymbol{M}_2(i,j), \dots, \boldsymbol{M}_d(i,j)\}$ to combine d feature saliency maps together and compute the saliency map value at pixel location (i,j). Since this is a new approach, this model will be selected for our performance evaluation.

1.1.1.8 Spectral Residual Approach (SR)

Another recently introduced approach for modeling the saliency mechanism of the HVS employs the Fourier transform, which is not biologically motivated, but is computationally fast and has good consistency with psychophysics [22, 23]. Hou and Zhang [24] introduced the spectral residual approach. Their method is mainly based on the general property of natural images, described by the $1/f$ law. This law states that the amplitude of the averaged Fourier spectrum, $\boldsymbol{A}(f)$, of the ensemble of natural images is proportional to $1/f$, in which f is the frequency. They

use this law to find the statistical similarities between input images and calculate the residual spectrum, which they called the bottom-up saliency map of the input image. This method can be summarized in the following steps, given the input image $\boldsymbol{I}$:

$$\boldsymbol{A}(f) = Amplitude\big(\mathcal{F}(\boldsymbol{I})\big), \tag{1-4}$$

$$\boldsymbol{P}(f) = Phase\big(\mathcal{F}(\boldsymbol{I})\big), \tag{1-5}$$

$$\boldsymbol{\mathcal{L}}(f) = log\big(\boldsymbol{A}(f)\big), \tag{1-6}$$

$$\boldsymbol{\mathcal{R}}(f) = \boldsymbol{\mathcal{L}}(f) - \boldsymbol{H}_k * \boldsymbol{\mathcal{L}}(f), \tag{1-7}$$

$$\boldsymbol{M}(x,y) = \boldsymbol{G} * \mathcal{F}^{-1}\left[e^{\boldsymbol{\mathcal{R}}(f)+\boldsymbol{P}(f)}\right]^2 \tag{1-8}$$

where $\boldsymbol{H}_k$ is a $k \times k$ matrix defined by:

$$\boldsymbol{H}_k = \frac{1}{k^2}\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & & \ddots & 1 \\ \vdots & & & \\ 1 & & 1 & 1 \end{bmatrix} \tag{1-9}$$

$\boldsymbol{G}$ is a 2D Gaussian filter to smooth the saliency map, and $*$ indicates the convolution of the term before the $*$ with the term following it, given by Gonzalez, Woods et al. [25] as:

$$\boldsymbol{w}(x,y) * \boldsymbol{f}(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} \boldsymbol{w}(s,t)\boldsymbol{f}(x-s,y-t) \tag{1-10}$$

where for matrix $\boldsymbol{w}$ of size $m \times n$, constants a and b are given by $a = (m-1)/2$ and $b = (n-1)/2$ (assuming m and n are odd integers), and matrix $\boldsymbol{f}$ can be of any dimension.

Guo et al. [26] stated that the phase spectrum, not the amplitude spectrum, of an image's Fourier transform is the key to calculating the bottom-up saliency map. They showed that locations with less periodicity, or less homogeneity, in an image "pop out" in the reconstructed image's phase spectrum. Accordingly, they proposed the phase spectrum of the Fourier transform (PFT) as the saliency detection model for grayscale images. Guo et al. proved that the PFT model is less computationally expensive in comparison with residual spectrum models, and PFT provides saliency maps very similar to residual spectrum saliency maps.

Implementing the concept used in designing the PFT model, Guo and Zhang [26, 27] introduced the spectrum of quaternion Fourier transform (PQFT) model for the bottom-up saliency mechanism. The PQFT model analyzes color, orientation and motion, in addition to intensity (in PFT), to calculate the saliency map. The PQFT model is independent of prior knowledge and parameters, and Guo and Zhang experimentally showed that it is fast enough to meet real-time requirements. They also claimed their method outperforms biologically-based saliency detection mechanism, however, they did not justify this conclusion.

Since in practice SR, PFT and PQFT saliency maps are very similar to each other [26], and due to the simplicity of the SR model, we will select it for performance evaluation.

1.1.1.9 Attention based on Information Maximization (AIM)

More recent research modeled the saliency mechanism in an information-theoretic way, and proposed an attention mechanism based on information theory. Bruce and Tsotsos [28, 29] proposed a saliency mechanism model based on the principle of maximizing information sampled from a scene. Information in their model is computed using Shannon self-information [30]. Similarly, Gao and Vasconcelos [31] used the concepts of entropy and mutual information to combine feature saliency maps together in the Koch and Ullman model. They maximize the mutual information between the feature distributions of center and surround regions in an image to build the bottom-up saliency map. These models offer good consistency with psychophysical and physiological data, but they are more complicated than previous models (e.g. the Itti et al. model [6]). Also, they are computationally expensive and difficult to implement in real-time systems [22]. The Gao and Vasconcelos model is explained in more detail in Chapter II. Among information theoretic approaches, we select the most popular one, which is the Bruce and Tsotsos method [28], for performance evaluation in Chapter III.

1.1.1.10 Context-Aware Saliency Detection (CASD)

Goferman et al. [32] proposed to employ the following four basic principles of the HVS saliency mechanism in the visual saliency model:

1. Local low-level considerations, to promote regions which differ from their immediate surroundings.
2. Global considerations, to suppress frequently occurring features.
3. Visual organization rules, which state that visual forms may possess one or several centers about which the form is organized.
4. High-level factors, for example including human face match detection in the visual saliency model.

They used the Euclidian distance between feature values and positions to define dissimilarities between two pixels. CASD's performance will be evaluated against eight other visual saliency models in Chapter III.

1.1.1.11 Image Signature Method (IS)

Hou et al. [33] introduced a new image descriptor named image signature to create saliency maps. They used Discrete Cosine Transform (DCT) to define the image signature ($\boldsymbol{IS}$) of gray scale image $\boldsymbol{I}$ as

$$\boldsymbol{IS}(\boldsymbol{I}) = sign\big(DCT(\boldsymbol{I})\big) \tag{1-11}$$

and defined the saliency ($\boldsymbol{M}$)

$$\boldsymbol{M} = \boldsymbol{G} * (\bar{\boldsymbol{X}} \circ \bar{\boldsymbol{X}}) \tag{1-12}$$

where $\bar{\boldsymbol{X}}$ is the inverse discrete cosine transform of $\boldsymbol{IS}$ and the operator $\circ$ is the Hadamard (entrywise) product operator and $\boldsymbol{G}$ is a 2-D Gaussian kernel to smooth the saliency map. The performance of IS will also be examined in Chapter III.

1.1.2 Top-down Saliency Mechanism Models

Studies in visual cognition have shown that while looking for a specific object in a scene, human observers use context information to facilitate finding objects of interest in a scene [34, 35]. However, most saliency mechanism models focus only on bottom-up information and ignore scene context. For example human observers usually know where an object is more likely to occur in a particular class of scenes (we have learned that in outdoor urban scenes, cars tend to be on the ground plane). As a result, human observers apply this prior knowledge to search in new scenes of the same class [35] (For example, when we are faced with an outdoor urban scene and asked to look for cars, we look preferentially at the ground plane).

Oliva et al. [34] employed the Bayesian rule to include in their saliency algorithm contextual priors which learn the relationship between context features and the location of a target from past experience. The role of the visual context factor in modulating attention is to provide information about past search experience in similar environments and the strategies that were successful in finding the target. Oliva et al. implemented their learning process on a database of images for which the location of the target was known. They showed their new model outperforms the Itti et al. model of the bottom-up saliency mechanism.

Ehinger et al. [36] also presented a model that include contextual information to narrow down the search to locations in the scene which are learned to more likely contain the object of interest. Their model incorporates a bottom-up saliency mechanism, the contextual information of where observers expect to find people, and a person-detector algorithm. They generated a large dataset containing where observers look in a scene when they were instructed to decide as quickly as possible whether a person was present in the scene. Their results showed good consistency with observers' fixation points in images of natural scenes.

Hamker [37] introduced a more comprehensive model of the saliency mechanism based on the Itti et.al method. As illustrated in Figure 1-4, Hamker modeled the interactions of several brain areas involved in visual attention processing. His model is able to learn a target by memorizing the feature values of a presented sample of the target. The feature information of the target template ($\hat{r}^F_{d,i}$) is employed in a match detection unit to compare every location in feature conspicuity maps with the target template. Search for the target is done iteratively and in different levels. If patterns are similar, an eye movement is initiated towards the region. Hamker's model shows 81% success in finding the object of interest within four shifts. The author claims that this model can be used for object detection and tracking purposes.

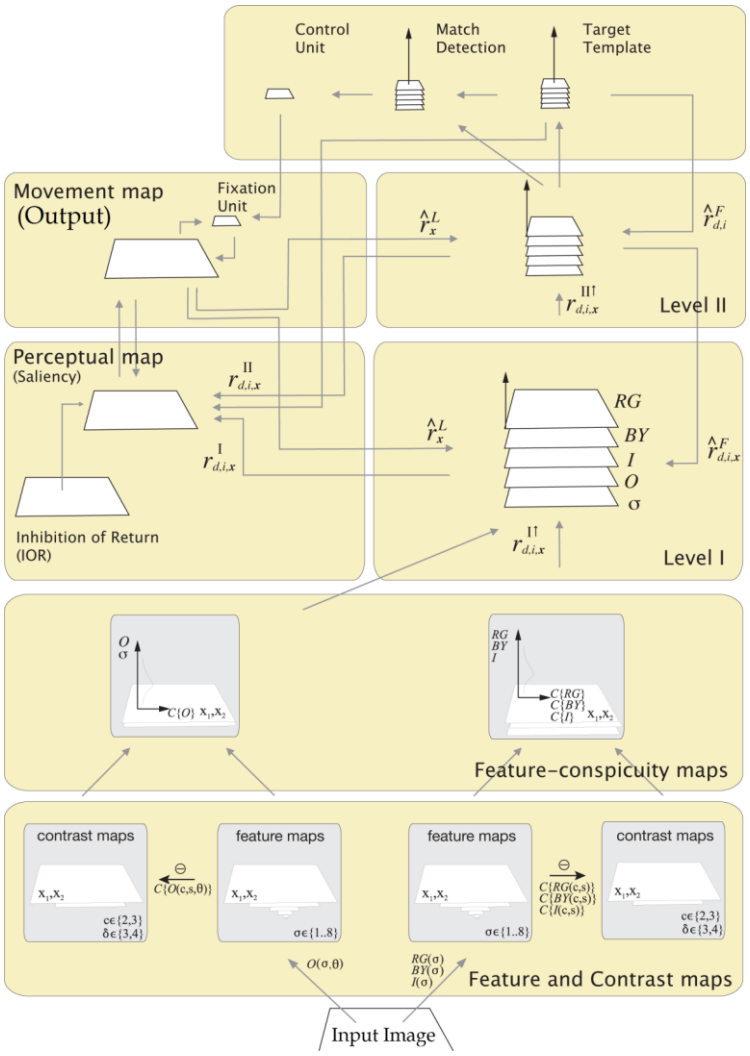


Figure 1-4. Hamker model of HVS saliency mechanism, modified from [37].

1.1.3 Other Studies

One of the fundamental pieces of information needed to model the saliency mechanism of the HVS is the importance of each feature space in the process of constructing the saliency map in the HVS. This information will help us in including important feature spaces in the model (and giving a proper weight to the corresponding feature saliency map before combining it into a saliency map), and ignoring feature spaces which are not sources of guidance of attention. Several studies have attempted to find the effect of each feature space on generating the saliency map in the bottom-up saliency mechanism of the HVS. Treisman and Gormican [38] stated the following features as the most important: colors; different levels of contrast; curvature; line tilt or misalignment; quantitative values like length and number; proximity; and closure and direction of movement. Wolfe and Horowitz [16] studied how likely feature spaces might guide the deployment of attention and classified them into five groups: undoubted attributes; probable attributes; possible attributes; doubtful attributes; and probable non-attributes, illustrated in Table 1-1.

Table 1-1. Feature Space which might guide the deployment of attention [39].

Undoubted attributes	**Probable attributes**	**Possible attributes**	**Doubtful attributes**	**Probable non-attributes**
- Color - Motion - Orientation - Size (including length and spatial frequency)	- Luminance onset (flicker) - Luminance polarity - Vernier offset - Stereoscopic depth and tilt - Pictorial depth cues - Shape	- Lighting direction (shading) - Glossiness (luster) - Expansion - Number - Aspect Ratio	- Novelty - Letter identity - Alphanumeric category	- Intersection - Optic flow - Color change - 3D volumes - Faces

Rajashekar [5] investigated human eye movement to provide an understanding of strategies used by observers in visual tasks. He showed that even in very noisy displays, observers do not search randomly, but instead the subject's attention is drawn to regions in the image whose luminance, contrast, and output of center-surround filters for luminance and contrast are significantly higher than other points in the image. He reported some differences across observers, even when the

displays contained simple geometric shapes as targets, which indicated that the observers adopt idiosyncratic behavior.

Instead of using a set of biologically plausible filters, for example Gabor or Difference-of-Gaussians filters (as in [2, 6, 14, 15]), Kienzle et.al [39] proposed to learn a visual saliency model directly from human eye tracking data. Judd et al. [40] collected eye movement data and used them as training examples to learn a model of saliency mechanism. They created continuous reference saliency maps by convolving a Gaussian function over the fixation locations of all users. Then, employing a machine learning algorithm, they found a model for both bottom-up and top-down saliency mechanisms. In their model, they used four different types of features:

1. **Low-level features:** Such as intensity, orientation and color.
2. **Mid-level features:** Since most of the objects rest on the surface of the earth, the horizon is a place that humans usually look for salient objects. They trained a horizon line detector and used the horizon line in the image as a mid-level feature.
3. **High-level features:** Such as human face, human body and cars.
4. **Center prior:** Since photographers usually frame objects of interest in the center of the image, they included a feature that indicates the distance to the center for each pixel.

Judd et al. method outperformed Olivia and Torralba [41], Rosenholtz [42], Itti and Koch [1] and Cerf et al. [43] visual saliency models.

1.1.3.1 Bottom-Up Saliency Mechanism Employed in Object Detection

Walther et al. [44] combined the Itti et al. [6] saliency mechanism with the Lowe [45] algorithm for object recognition and showed that object detection results are improved by concentrating on regions of interest. In another paper, using a saliency map for object recognition, Walther and

Koch [46] proposed a model of attending to proto-objects[2] in the image. After calculating the saliency map, looking back at the conspicuity maps, the feature saliency map with the highest contribution at the attended location is found. Then, the approximate extent of the proto-object at that location is determined in that feature saliency map.

1.1.3.2 Fixation Data Study by the MIT Computer Science and Artificial Intelligence Lab

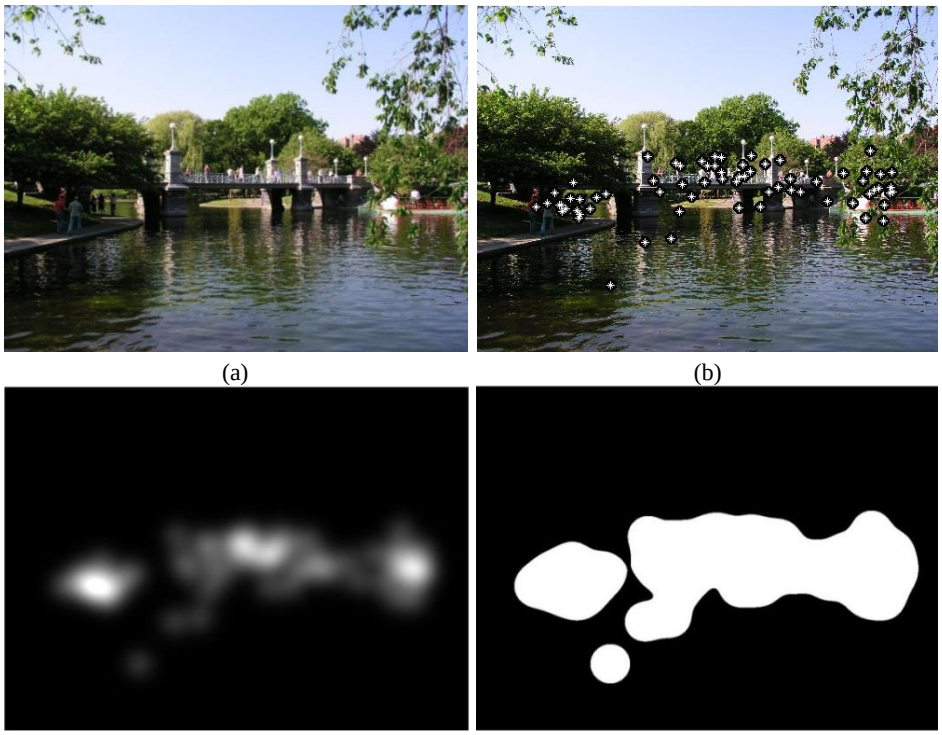
Visual saliency models have often been validated against human eye movement data. In a study at the MIT Computer Science Artificial Intelligence Laboratory, "Learning to Predict Where Humans Look" (LPWHL) [40], eye tracking data of 15 human observers on 1003 images of different scenes was collected. Gaze tracking paths and the first 5 fixation locations were recorded for each viewer. For every image in the database, a continuous saliency map was found by convolving a 2 dimensional Gaussian over the fixation locations of all observers. As an example, an image, human observers' fixations, and its saliency map are shown in Figure 1-5 (a), (b) and (c). A binary map showing the 20% most salient regions in the image is demonstrated in Figure 1-5 (d). The database created in the LPWHL study will be used in this report to find the best comparison metric and also to examine visual saliency models.

1.2 Organization of the Report

Although much work has been done on modeling the saliency mechanism, saliency maps constructed with the previous methods sometimes extract unimportant locations in the display as salient points. We believe the following factors give rise to this problem:

1. Predetermined parameters; for example the center-surround mechanism is applied with fixed sets of radii, while the radii should be determined based on the properties of the

[2] Prior to focused attention, there is a stage of early low-level (involving only the geometric and photometric properties) and rapid (occurring within a few hundred ms) processing which is carried out in parallel across the visual field. It results in structures that are volatile units of visual information which can be switched into a coherent and stable object when accessed by focused attention. These structures are called proto-objects [18, 19].



(a) (b)

(c) (d)

Figure 1-5. Original image (a); first 5 fixation points of 15 viewers from LPWHL study (b); the saliency map created by convolving a 2D Gaussian function on the fixation locations (c); and the top 20% salient regions (d) [40].

display, such as size of the objects in the display, distances between objects in the display, and textures of the objects.

2. Only one set of parameters is employed to analyze the entire image. We believe that to effectively imitate the HVS saliency mechanism, different parts of the visual scene should be analyzed with different sets of parameters (analyzing the image locally).
3. Methods employed for combining feature saliency maps, or conspicuity maps, to construct saliency maps are not well-designed. In some cases, each feature saliency map identifies some of the salient locations of the image properly, but these points are often lost during the process of calculating the saliency map. We believe more thoughtful approaches should be applied to combine feature saliency maps into a saliency map.

The study herein proposes to address the first two problems and introduce a more precise model of the saliency mechanism of the HVS. In what follows, Chapter II explains the steps of the

bottom-up saliency mechanism and the center-surround mechanism, and introduces a new method for normalizing feature saliency maps in saliency detection mechanisms. In Chapter III, three new saliency map comparison metrics are introduced and six published saliency map comparison metrics are explained. An evaluation procedure to evaluate comparison metrics is introduced and all metrics are ranked accordingly. Afterwards, in Chapter IV, employing the best comparison metric, nine selected visual saliency models and a model introduced in this study are examined for their performances on a database of human observers' fixation data on a set of natural images. They also are test on a database of synthetic images and the best models are identified. The best saliency model found in Chapter IV is modified in Chapter V to overcome the first two problems mentioned above. As an application of modeling the saliency mechanism of the HVS, a saliency mechanism with the new normalization method is applied to dishware inspection in Chapter VI. Finally, Chapter VII provides conclusions and proposes future works.

CHAPTER II

2 METHOD

The general architecture of bottom-up saliency mechanism models and some well-known bottom-up and top-down saliency mechanism models were explained in Chapter I. In this chapter, common mathematics behind models of bottom-up saliency mechanisms are explained, as well as some normalization methods. As shown in Figure 2-1, the first step in calculating bottom-up saliency maps is to extract a set of feature spaces such as image intensity, color, orientation and texture from the image. Wolfe and Horowitz [16] studied the importance of different features in deployment of visual attention and grouped them by the probability that they are sources of guidance of attention. Note that usually salient qualities of the image are spread in different feature spaces. As a result, each feature space reflects some of the salient locations, and it is also possible that a feature space contains no salient quality.

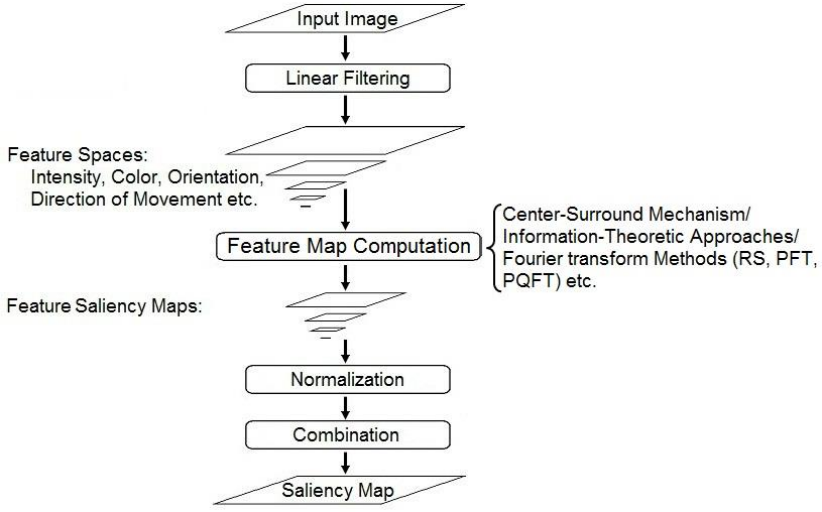
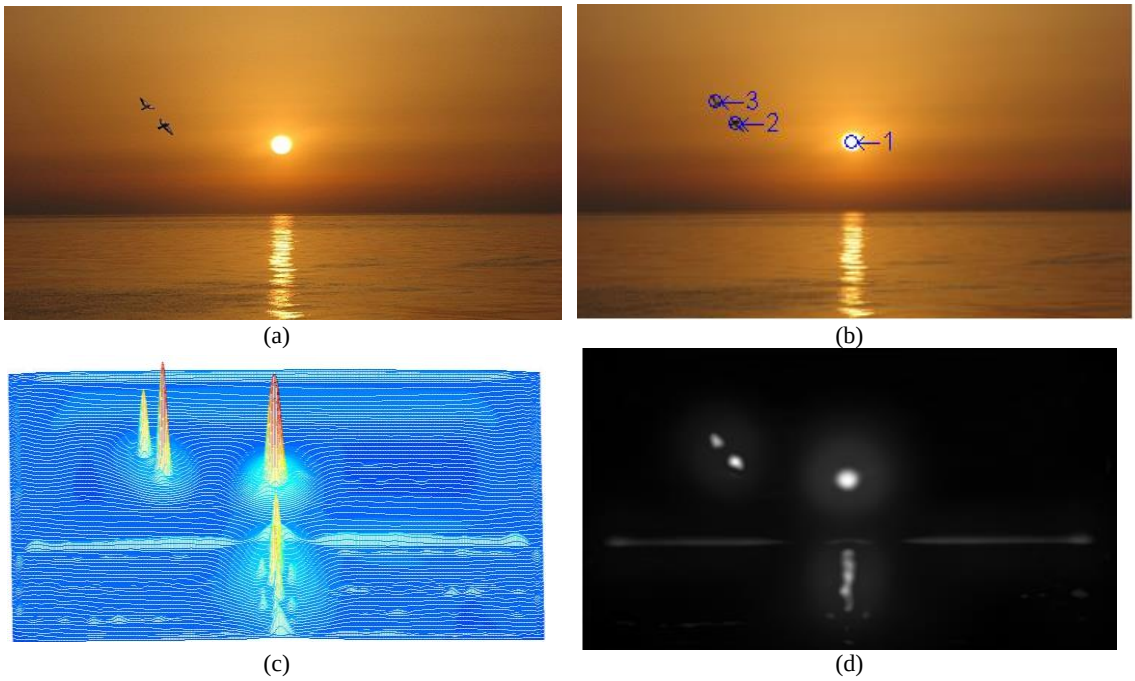


Figure 2-1. General architecture of the saliency map calculation, modified from [6].

As illustrated in Figure 2-1, the next step is to analyze feature spaces to compute feature saliency maps. Feature saliency maps are two dimensional matrices of scalar numbers to indicate the saliency of each location in view of the elementary features to which they correspond. A popular approach in bottom-up models of the saliency mechanism for computing feature saliency maps is to employ the center-surround mechanism. Applying this method, locations in the visual space that differ from their surroundings with respect to an elementary feature gain larger values in the corresponding feature saliency map.

Itti et al. [6] proposed to normalize feature saliency maps before combining them together and constructing the saliency map. This is to promote regions of feature saliency maps that highlight salient locations and suppress regions which contain little useful information. Combining normalized feature saliency maps of the same elementary feature allows corresponding conspicuity maps to be calculated. Conspicuity maps show saliency of each pixel location based on only one type of feature spaces for example color. The final step is to combine conspicuity maps into a saliency map. However, extracting conspicuity maps is not essential, and saliency maps can be computed directly from feature saliency maps [2]. Similar to feature saliency maps, conspicuity maps and saliency maps are two dimensional matrices of scalar numbers. Saliency maps present the saliency of each location based on all features and demonstrate the general saliency of each point. Figure 2-2 presents an image, its saliency map and the first three salient locations in the image (based on maximum saliency heights) found using our visual saliency model explained in Section 2.5.

As illustrated in Figure 2-2 (c) and (d), saliency maps, conspicuity maps and feature saliency maps can be presented in three and two-dimensional images. In three-dimensional images, the height of the map represents the saliency of each point. In two dimensional black and white images, brighter points indicate higher saliency. Clearly in Figure 2-2, the most salient part of the



(a) (b)

(c) (d)

Figure 2-2. An image (a) and its 3 most salient points (b), with 3-D (c) and 2-D (d) saliency maps.

image is the sun. In order of decreasing saliency are two geese, the reflection of sunlight in water and the horizon line. We employ 2-D saliency maps in this report and switch to 3-D maps when they are required.

The process of extracting feature spaces, the center surround mechanism, two normalization methods, and a new normalization method are explained in what follows.

2.1 Extracting Feature Spaces

In most of the models of the saliency mechanism [1, 2, 6] a same approach is used in extracting early visual features. The intensity feature space $\boldsymbol{I}$ is calculated using:

$$\boldsymbol{I} = (\boldsymbol{r} + \boldsymbol{b} + \boldsymbol{g})/3, \tag{2-1}$$

where $\boldsymbol{r}$, $\boldsymbol{b}$ and $\boldsymbol{g}$ are matrices of pixel intensities in the red, blue, and green color channels, respectively, of the input image [6]. Then, the red, blue and green color intensities are normalized using the intensity feature space

$$\tilde{r}_{ij} = r_{ij}/I_{ij}, \tag{2-2}$$

$$\tilde{b}_{ij} = b_{ij}/I_{ij}, \tag{2-3}$$

$$\tilde{g}_{ij} = g_{ij}/I_{ij}, \tag{2-4}$$

where quantities on the left are the normalized matrix elements. Afterwards, four new color intensity matrices for red, green, blue and yellow ($\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}$ and $\boldsymbol{Y}$ respectively) are calculated following [2] by:

$$\boldsymbol{R} = \left\lfloor \tilde{\boldsymbol{r}} - \left(\tilde{\boldsymbol{b}} + \tilde{\boldsymbol{g}}\right)/2 \right\rfloor_+, \tag{2-5}$$

$$\boldsymbol{G} = \left\lfloor \tilde{\boldsymbol{g}} - \left(\tilde{\boldsymbol{b}} + \tilde{\boldsymbol{r}}\right)/2 \right\rfloor_+, \tag{2-6}$$

$$\boldsymbol{B} = \left\lfloor \tilde{\boldsymbol{b}} - (\tilde{\boldsymbol{r}} + \tilde{\boldsymbol{g}})/2 \right\rfloor_+, \tag{2-7}$$

$$\boldsymbol{Y} = \lfloor (\tilde{\boldsymbol{r}} + \tilde{\boldsymbol{g}})/2 - |\tilde{\boldsymbol{r}} - \tilde{\boldsymbol{g}}|/2 - \boldsymbol{b} \rfloor_+, \tag{2-8}$$

where $\lfloor x \rfloor_+ = max(x, 0)$ which means discarding all negative values in the result.

Orientation feature spaces are decomposed from intensity feature space using 2-D Gabor filters. The 2-D Gabor function GW is defined by:

$$GW(x, y) = s(x, y) w_r(x, y) \tag{2-9}$$

where $s(x, y)$ is a complex sinusoid (defined below) known as the carrier; $w_r(x, y)$ is a 2-D Gaussian shaped function called the envelope [47], and is defined by:

$$w_r(x, y) = e^{-\left[x_g^2/2\sigma_x^2 + y_g^2/2\sigma_y^2\right]}, \tag{2-10}$$

where

$$x_g = (x - \mu_x)\cos\theta - \left(y - \mu_y\right)\sin\theta,$$

$$y_g = (x - \mu_x)\sin\theta + \left(y - \mu_y\right)\cos\theta.$$

$w_r(x, y)$ has five parameters: σ_x^2 and σ_y^2 are the variance of x_g and y_g; μ_x and μ_y are the mean of the distribution of $w_r(x, y)$ along the x and y axes, respectively; and θ is the rotation angle measured counterclockwise from the x_g axis [47]. The 2-D sinusoidal function, $s(x, y)$, is defined by [47] as:

$$s(x, y) = e^{-2\pi j[U_0(x - x_m) + V_0(y - y_m)]}. \tag{2-11}$$

U_0 and V_0 are spatial frequencies in cpd (cycle/degree). Usually only the real part of $s(x, y)$ ($s_r(x, y)$) is used in Gabor filters, given by:

$$s_r(x, y) = cos[-2\pi(U_0 x + V_0 y) + P], \tag{2-12}$$

where

$$P = 2\pi(U_0 x_m + V_0 y_m).$$

In this research, Gabor filters are generated for a set of four, six or eight angles evenly spread from 0 to π. Also, spatial frequencies of 2.5, 5 and 10 cpd are employed for both directions, as used in [2].

2.2 Modeling the Center-Surround Mechanism

The center-surround mechanism of biological vision systems and its application in the visual attention mechanism have been extensively analyzed in the literature [1, 6, 7, 48]. This mechanism simply implies the more different a stimulus is from its surrounding, the more salient is the stimulus. Among functions offered for modeling this mechanism, difference-of-Gaussian (DOG) functions have been successfully applied [1, 6, 49]. The general form of DOG function W_{DOG} is given by:

$$W_{DOG}(x, y) = \frac{c_{ex}^2}{2\pi\sigma_{ex}^2} e^{-(x^2+y^2)/2\sigma_{ex}^2} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} e^{-(x^2+y^2)/2\sigma_{inh}^2}. \tag{2-13}$$

The first and the second terms in (2-13) represent excitatory and inhibitory characteristics of the center-surround mechanism, respectively. The suggested parameters in [1] to generate the DOG filter W_{DOG} are: $\sigma_{ex} = 0.02$ and $\sigma_{inh} = 0.25$ times the input image width, $c_{ex} = 0.5$, and $c_{inh} = 1.5$. We employ the same parameters in this report. Figure 2-3 illustrates $W_{DOG}(x, y)$ for $\sigma_{ex} = 4$ and $\sigma_{inh} = 25$.

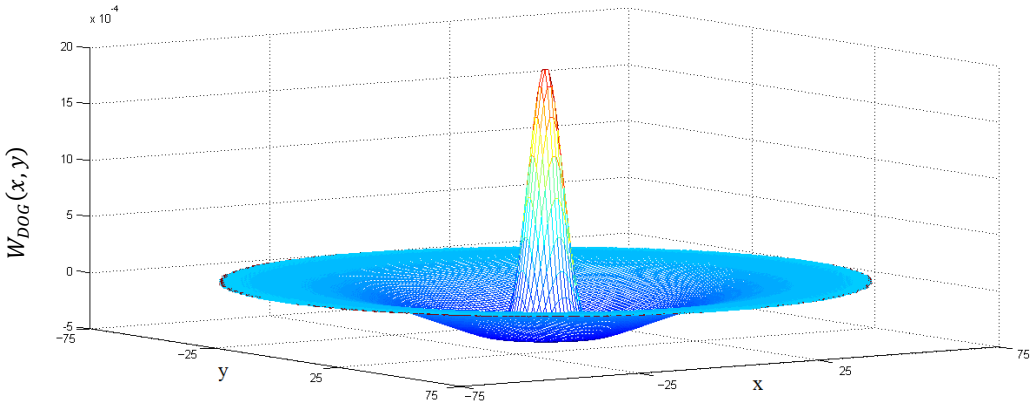


Figure 2-3. Difference-of-Gaussian filter for $\sigma_{ex} = 4$ and $\sigma_{inh} = 25$.

2.3 Normalizing Feature Saliency Maps

In the absence of any top-down supervision, Itti et al. [6] proposed normalizing feature saliency maps before combining them together and constructing the saliency map. The Itti et al. normalization method [6] consists of three main steps: First, normalizing the values in the map to a range of $[0 \quad M]$; second, computing the average $\bar{m}$ of all local maxima in the feature saliency map; third, globally multiplying the feature saliency map by $(M - \bar{m})^2$, where M is the global maximum of the map. In practice, this method calculates weights that indicate the importance of each feature saliency map in comparison with other feature saliency maps, while the feature saliency map is not changed locally. We suggest application of normalization methods that are capable of promoting regions of feature saliency maps which highlight salient locations and suppressing regions which contain little useful information. To do so, every location in the feature saliency map should be compared to its surroundings and the entire feature saliency map, and then be promoted or suppressed correspondingly.

Itti and Koch [1] suggest normalizing each feature saliency map to the interval $[0 \quad 1]$ and then subjecting each feature saliency map to 10 iterations of the following process:

$$\boldsymbol{\mathcal{M}}_{New} = \lfloor \boldsymbol{W}_{DOG} * \boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{M}} - C_{inh} \rfloor_{+} \tag{2-14}$$

where $C_{inh} = 0.02$ and $\lfloor x \rfloor_{+} = max(x, 0)$, which means discarding all negative values in the result, and $*$ indicates the convolution of the term before the $*$ with the term following it.

Employing the center-surround mechanism, Gao [2] defined two circular windows W_l^0 and W_l^1 at each location l in the visual field. W_l^1 is the inner window that represents the center neighborhood, and W_l^0 is the outer annulus that defines the surroundings. The saliency $\boldsymbol{S}(l)$ of each location l in the visual field is calculated by:

$$\boldsymbol{S}(l) = I(\boldsymbol{X}; Y) = \sum_{k=1}^{d} I_l(\boldsymbol{X}_k; Y) \tag{2-15}$$

where $I_l(\boldsymbol{X}_k; Y)$ is the mutual information between class Y and feature space $\boldsymbol{X}_k$ at location l given in (2-16), and d is the number of feature spaces used for calculating the saliency map. $\boldsymbol{X}(l) = [\boldsymbol{X}_1(l), \dots, \boldsymbol{X}_d(l)]$ where $\boldsymbol{X}_k(l)$ includes values of the k^{th} feature saliency map at all image locations j within the two windows, and $Y(l) = 1$ if $j \in W_l^1$, and $Y(l) = 0$ if $j \in W_l^0$.

For our case, the mutual information between class Y and feature vector $\boldsymbol{X}$ is defined by [30]:

$$I(\boldsymbol{X}; Y) = \sum_{i=1}^{2} \int p_{\boldsymbol{X},Y}(\mu, i) log \frac{p_{\boldsymbol{X},Y}(\mu, i)}{p_{\boldsymbol{X}}(\mu) p_Y(i)} d\mu, \tag{2-16}$$

Where $p_{\boldsymbol{X},Y}(\mu, i)$ is the joint density probability of the feature vector $\boldsymbol{X}$ and class $Y = i$, $p_{\boldsymbol{X}}(\mu)$ is the marginal density of $\boldsymbol{X}$ and $p_Y(i)$ is the probability of class $Y = i$. The Gao's saliency mechanism has two free parameters: the size of the center and the surround windows, r_i and r_o. These parameters are selected as follows: $r_i = 0.04 \times \max(K, L)$ and $r_o = 6 \times r_i$, where $K \times L$ is the image size.

2.4 A New Normalization Method

We propose a new normalization method to overcome difficulties of other methods in the literature. As shown in Figure 2-2 (c) feature saliency maps and saliency maps can be treated as 3-D surfaces. We employ the weighted volume trapped between the feature saliency map and a surface parallel to the xy plane passing through the point $[x_l, y_l, M(x_l, y_l)]$ as the normalized value $M_{Normalized}(x_l, y_l)$ of the feature saliency map at (x_l, y_l), given by:

$$M_{Normalized}(x_l, y_l) = \iint f(x,y) \times [M(x_l, y_l) - M(x,y)]dxdy, \quad (2\text{-}17)$$

where (x_l, y_l) is any point in the feature saliency map, and

$$f(x,y) = e^{-((x-x_l)^2+(y-y_l)^2)/2\sigma^2}, \quad (2\text{-}18)$$

is the weight function. Equation (2-17) describes the process in the continuous form. Since feature saliency maps are discrete maps herein, we rewrite (2-17) as (2-19):

$$\boldsymbol{M}_{Normalized}[i,j] = \sum_{k=1}^{K}\sum_{l=1}^{L} \boldsymbol{f}_{ij}[k,l] \times (\boldsymbol{M}[i,j] - \boldsymbol{M}[k,l]), \quad (2\text{-}19)$$

where $\boldsymbol{f}_{ij}$ is the weight function centered at pixel $[i,j]$, and both $\boldsymbol{M}$ and $\boldsymbol{f}$ are matrices of size $K \times L$. The process introduced in (2-19) is applied to all points in the feature saliency map to calculate the normalized feature saliency map. (2-17) and (2-19) connect the saliency of each point to the difference between its value and the value of other points in the feature saliency map. The parameter σ in this method controls the spread of the weight function. In our implementation σ is selected such that

$$e^{-r^2/2\sigma^2} = 0.01, \quad (2\text{-}20)$$

where r is selected such that $0.05 \times min(K,L) \leq r \leq 0.25 \times min(K,L)$. The effect of r on the normalization process will be discussed in following sections.

At a local maximum (x_M, y_M), the volume trapped between the feature saliency map and a surface parallel to the xy plane passing through (x_M, y_M) decreases as the number of local maxima around (x_M, y_M) increases. Accordingly, we expect our normalization method to suppress a local maximum surrounded by other local maxima and to promote a local maximum which is far from other local maxima in the feature saliency map. For the same reason, it is expected to suppress noisy parts of the feature saliency map.

Figure 2-4 illustrates the new normalization method applied to a continuous one-dimensional example. The feature saliency map $M(x)$ and the horizontal line passing through $\big(x_l, M(x_l)\big)$ is

shown in Figure 2-4 (a) and (e) for $x_l = 3$ and $x_l = 4$, respectively; Figure 2-4 (b) and (f) show $[M(x_l) - M(x)]$; Figure 2-4 (c) and (g) present the weight functions, $f(x) = e^{-(x-x_l)^2/2\sigma^2}$, for $\sigma = 0.8$ and $x_l = 3$ and $x_l = 4$, respectively; Figure 2-4 (d) and (h) show $f(x) \times [M(x_l) - M(x)]$. The magnitudes of the normalized feature saliency map at $x_l = 3$ and $x_l = 4$ are equal to $\int_0^{10} f(x) \times [M(x_l) - M(x)]dx$, which are -8.22 and 73.44, respectively.

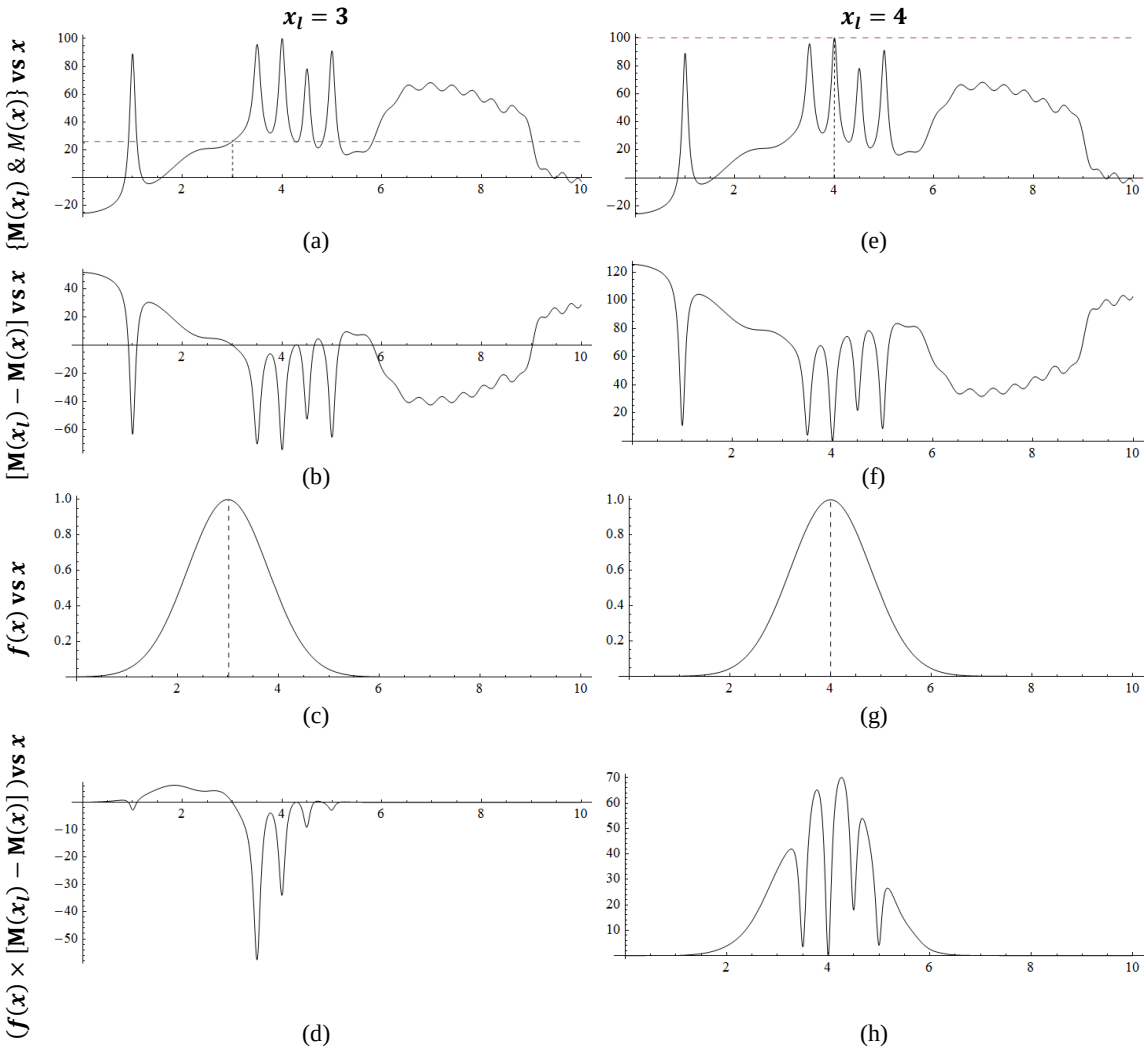


Figure 2-4. The new normalization method for one-dimensional examples.

Applying this process for all points in the feature saliency map results in the normalized feature saliency map, Figure 2-5. The proposed normalization method has the following properties:

1. Local maxima of the normalized feature saliency map often happen at points quite close to the local maxima of the feature saliency map.
2. A local maximum surrounded by other local maxima in the feature saliency map will be suppressed in the normalized feature saliency map (for example at $x_l = 4$ in Figure 2-5).
3. A local maximum which is far from other local maxima in the feature saliency map will be promoted in the normalized feature saliency map (for example at $x_l = 1$ in Figure 2-5).
4. Increasing σ in the weight function enlarges the area over which the normalization process is applied, (2-18).
5. Smooth areas of the feature saliency map (areas with magnitudes very close together) will be suppressed to zero (or small magnitudes) in the normalized feature saliency map (for example at $6 < x_l < 9$ in Figure 2-5).

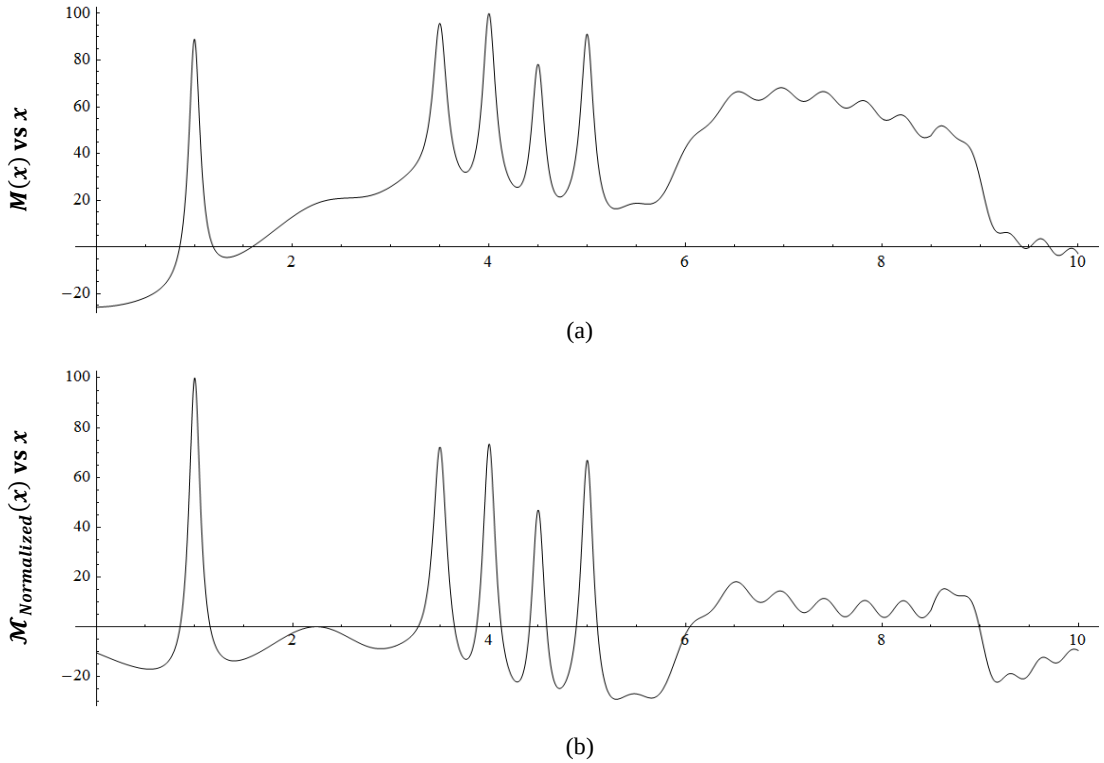


Figure 2-5. A one-dimensional feature saliency map (a) and its normalized map (b).

2.5 A New Visual Saliency Model (EH)

In our new method proposed, first feature spaces are extracted as explained in Section 2.1. Then feature saliency maps are extracted from feature spaces using the center-surround mechanism. To generate the DOG filter, the suggested parameters by Itti and Koch [1] are used, which are: $\sigma_{ex} = 0.02$ and $\sigma_{inh} = 0.25$ times the input image width, $c_{ex} = 0.5$, and $c_{inh} = 1.5$. Afterwards, feature saliency maps are normalized using the new normalization method in Section 2.3, and finally the saliency map is created as the average of the feature saliency maps.

In Chapter IV, our saliency mechanism is compared to 9 other visual saliency models. It also has been incorporated in a saliency detection mechanism and is used for dish inspection in Chapter VI.

2.6 A Comparison between Normalization Methods

The definition of the saliency of objects or areas in a scene is not well-defined. In many cases, it is difficult for even a human observer to distinguish between the saliency of two objects in a display and select the more salient object. Consequently, it is difficult to compare the performance of different normalization methods in saliency detection mechanisms. Based on the purpose of the feature saliency map normalization, we define some criteria for evaluation of normalization methods.

A normalization method is expected to promote the regions of the feature saliency map which highlight salient locations in the feature space. It should be mentioned that not all of the salient objects in an image are reflected in a single feature space. In practice, each feature space highlights some of the salient objects. Therefore, normalization methods may highlight salient objects reflected in a given feature space, but not in others.

Suppressing regions that contain little or no useful information is another purpose of a normalization method. This is as important as highlighting salient locations, because the values of

the feature saliency map in areas other than salient locations act similar to noise and misleads the process of constructing the saliency map.

Finally, simplicity of the normalization method is a crucial parameter. Usually the most time consuming part of the saliency detection mechanisms is normalization. Therefore, the simpler the normalization method is, the faster is the saliency detection mechanism. Moreover, a small number of free parameters in the normalization method is important for simplicity of utilization of the saliency mechanism. That is because the best set of free parameters for the normalization method applied in a specific application must often be found by trial and error. As a result, application of the saliency mechanism in a new field is easier when the number of free parameters is small.

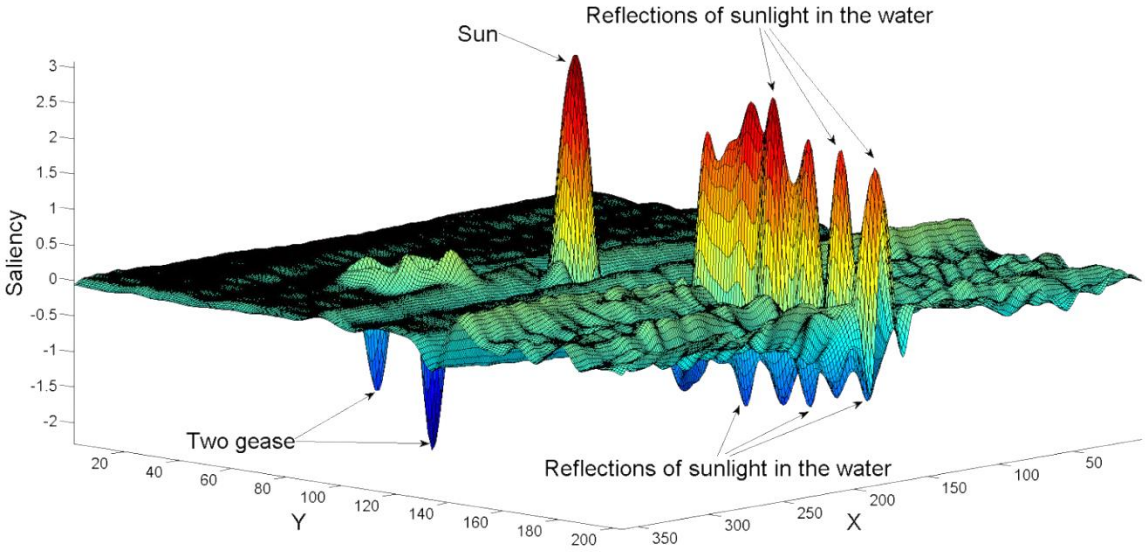
In the next Section, a qualitative comparison of normalization methods is given. A quantitative analysis of the visual saliency models is given in Chapter 3.

2.6.1 A Qualitative Comparison of Normalization Methods

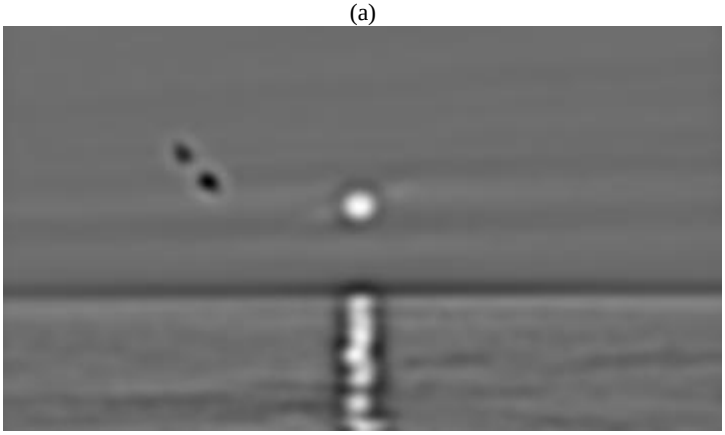
To compare the efficiency of our proposed new normalization method with other methods explained in this chapter, we apply them to the image shown in Figure 2-2 (a). The three most salient objects in this image are the sun and the two geese. These are followed in saliency by the reflection of sunlight in the water and the horizon line. This image is selected because all the salient parts of the image are suitably portrayed in the gray level image in Figure 2-6, which facilitates discussion.

First, we extract the feature saliency map from the feature space (the gray level image) using the center-surround mechanism. To generate the DOG filter, the suggested parameters by Itti and Koch [1] are used, which are: $\sigma_{ex} = 0.02$ and $\sigma_{inh} = 0.25$ times the input image width, $c_{ex} = 0.5$, and $c_{inh} = 1.5$. Convolving the gray level image with the DOG filter, the corresponding feature saliency map is calculated, as shown in Figure 2-7.

Figure 2-6. The gray-level image used for evaluating the performance of the normalization methods.



(a)



(b)

Figure 2-7. The feature saliency map extracted from gray level image using DOG filter, $\omega = 0.06$ cycles/pixel, in 3-D (a) and 2-D (b).

As illustrated in Figure 2-7, the center-surround mechanism highlights the salient points in the image as well as some areas with little importance. It is important to suppress areas with little

information before calculating the saliency map. Now, we apply the normalization methods explained in the previous chapter to this feature saliency map. Since the Itti et al. [6] normalization method finds only a normalization weight to be multiplied by the feature saliency map globally, the location of salient points in the image will be the same as in Figure 2-7, and the areas with little information will not be suppressed. Figure 2-8 shows, respectively, the results of the Itti and Khoch [1], Gao's method, and the new method proposed in this report.

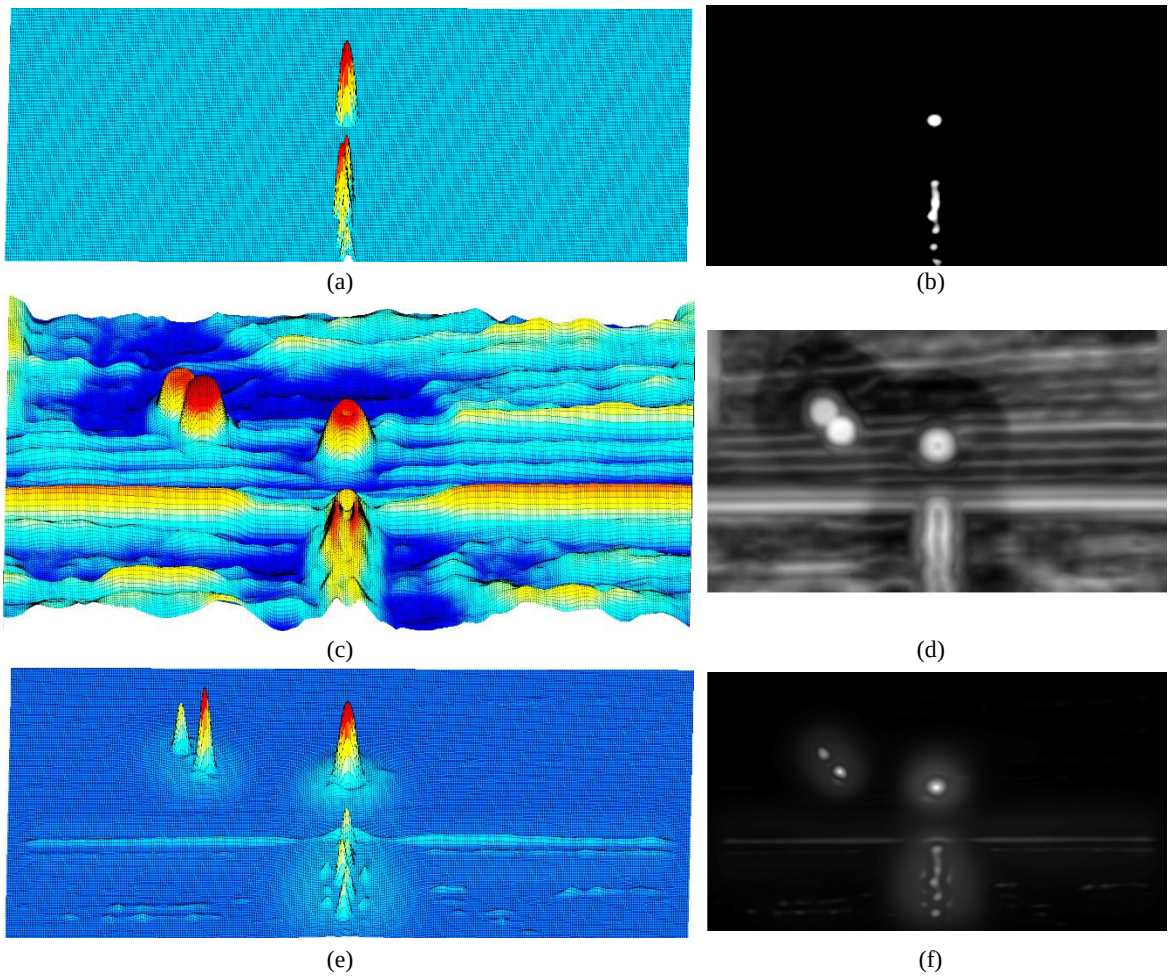


Figure 2-8. Normalized 3-D and 2-D feature saliency maps using Itti and Koch method (a) and (b); Gao's method (c) and (d); and the method presented in this report (e) and (f).

As shown in Figure 2-8 (a) and (b), the Itti and Koch method suppresses non-salient areas well and highlights some of the salient locations, but it misses the two geese. As illustrated in (2-14), Itti and Koch discard the negative part of the new feature saliency map in each iteration, which

results in suppressing the non-salient parts of the map properly; however, the negative parts, which may contain useful information, are lost. In practice, the Itti and Koch method analyzes the presence of qualities to calculate the saliency of an image, but in many cases the absence of a quality is the reason for the saliency at a location. For example, two geese in Figure 2-6 are salient for a human observer because they are dark and are placed in a bright background. They are lost in the saliency map because only the intensity of the image (presence of the quality) is analyzed in this method.

Gao's method analyzes both presence and absence of qualities and highlights all salient points in the image, Figure 2-8 (c) and (d). But it was not successful in suppressing the areas with little importance, and the normalized feature saliency map is very wavy or blurred. As a result, using the normalized map calculated using Gao's method can mislead the process of calculating the saliency map. Another disadvantage of Gao's method is that it is a complicated method with large amounts of time required to calculate the normalized map, compared to Itti and Koch, and the new method herein. The computation times for normalizing the feature saliency map, Figure 2-7, using the Itti and Koch method, Gao's method, and the new method introduced in this report are 148.3, 1350.8 and 77.9 seconds, respectively (using MATLAB® R2010a 32-bit, Image Processing Toolbox V7.0, Window XP Professional, Pentium®4 3.2GHz, 2GB RAM).

As shown in Figure 2-8, the normalization method introduced in this report outperformed the other normalization methods in both highlighting the salient points and suppressing other points. Here the absolute value of the result is used as the normalized feature saliency map. Similar to Gao's method, the introduced method analyzes both presence and absence of a quality to normalize a map. Consequently, all salient locations are highlighted in the normalized feature saliency map. Also this method is successful in suppressing parts with little or no information.

CHAPTER III

3 SELECTION OF A BEST SALIENCY MAP COMPARISON METRIC

Some researchers define the goal of attention modeling to be finding a model that minimizes the error in locating human observer's fixations [50]. However, the most common approach is finding a model which predicts a human eye saliency map [11, 17, 40, 51, 52]. Accordingly, all comparison metrics analyzed herein are designed to find similarities between a saliency map calculated by a visual saliency model and a same-size reference saliency map. In this report, the saliency map computed with a visual saliency model is referred to as the Predicted Saliency Map (PSM), denoted by saliency values $\boldsymbol{M}(k,l)$. The reference maps are extracted from the LPWHL database [40], which are called Reference Human Saliency Maps (RHSMs), and are denoted by saliency values $\boldsymbol{\mathcal{M}}(k,l)$. $1 \leq k \leq K$ and $1 \leq l \leq L$, where K and L are height and width of the maps in pixels, respectively. The method to compute saliency maps is explained in Section 3.2.1

3.1 Saliency Map Comparison Metrics

In this section, three new metrics along with six already published metrics to compare saliency maps are explained.

3.1.1 Cosine of the Angle between Two Maps ($Cos\theta$)

Rearranging a PSM and an RHSM into KL length vectors $\vec{m}_1$ and $\vec{m}_2$, we define cosine of the angle between two maps as:

$$Cos\theta = \frac{\langle \vec{m}_1, \vec{m}_2 \rangle}{\|\vec{m}_1\|.\|\vec{m}_2\|} \tag{3-1}$$

where $\langle .,. \rangle$ is the inner product of the vectors and $\|.\|$ is the vector 2-norm. $Cos\theta = 1$ indicates two maps are identical. We introduce $Cos\theta$ as a measure of similarities between two maps.

3.1.2 Score2

We define $Score2$ as the average of an RHSM values at the first (highest) N peaks of a PSM, given by (3-2):

$$Score2 = \sum_{i=1}^{N} \boldsymbol{M}(k,l)_i \Big/ N, \tag{3-2}$$

where $(k,l)_i$ is the pixel location of the i^{th} peak of the PSM (i^{th} fixation of the visual saliency model). A large $Score2$ value shows that the N highest salient points found by the algorithm are the prominent locations of the image found by human observers. Therefore, the larger is $Score2$, the better is the performance of the visual saliency mechanism. Herein, *Score2* is introduced to find resemblances between a PSM and the RHSM, when $N = 5$.

3.1.3 1-Norm of the Difference Map (*NDM*)

The difference map is defined as the difference between the PSM computed for an image and the image RHSM. We suggest treating the difference map as a vector with length $K \times L$ and using its vector 1-norm as a comparison metric to find dissimilarities between two maps, as follows:

$$DM = \sum_{k=1}^{K} \sum_{l=1}^{L} |\boldsymbol{M}(k,l) - \boldsymbol{\mathcal{M}}(k,l)| \Big/ K \times L\,. \tag{3-3}$$

NDM defined as:

$$NDM = 1 - DM \tag{3-4}$$

is introduced in this report as a measure of similarities between two maps.

3.1.4 Hit Rate

In [26] the authors used the number of correct objects detected in the first 5 fixations of a saliency model as a measure of its performance, where, local maxima of the PSM are considered as fixation points. *Hit rate* is defined as follows:

$$Hit\ Rate = \frac{Number\ of\ correct\ Hits}{N} \tag{3-5}$$

In [40] images are not divided into objects of interest and background. However, the most salient 20% of the image based on its RHSM is selected as the foreground, and the rest is called the background of the image. Accordingly, if a fixation point happens to be in the foreground, it is considered as correct object detection ($0 \leq Hit\ Rate \leq 1$). Herein we use:

$$N = min(5, Number\ of\ peaks\ in\ the\ predicted\ saliency\ map).$$

3.1.5 Finding the Most Salient Location in the Image (DS)

In addition to "Hit Rate", the ability to find the most salient object in the image in the first four fixations is employed in [26] to evaluate visual saliency models. In this project, the global maximum of the RHSM is considered as the most salient location of the image, and the minimum distance (d) of the first four fixations of a visual saliency model is defined as a metric to evaluate its performance. Since in this report comparison metrics are defined to measure similarities between two maps, DS is defined as a comparison metric ($0 \leq DS \leq 1$) by:

$$DS = 1 - \frac{d}{\sqrt{L^2 + K^2}} \tag{3-6}$$

The larger is DS, the closer are the two maps.

3.1.6 Normalized Correlation Coefficient (NCC)

The correlation coefficient of two saliency maps defined in (3-7) is used in [17, 53, 54] to find the linear relationship between two maps.

$$CC = \frac{\sum_{k,l}[(\boldsymbol{M}_1(k,l) - \mu_1) \times (\boldsymbol{M}_2(k,l) - \mu_2)]}{\sqrt{\sum_{k,l}(\boldsymbol{M}_1(k,l) - \mu_1)^2 \times \sum_{k,l}(\boldsymbol{M}_2(k,l) - \mu_2)^2}} \quad (3\text{-}7)$$

where μ_i is the mean value of the map M_i. For images in the database, CC is computed for the PSM and the image RHSM. Normalizing CC in the [0 1] interval produces the Normalized Correlation Coefficient, NCC, defined by:

$$NCC = \frac{1 + CC}{2} \quad (3\text{-}8)$$

$NCC = 1$ implies that two maps are either exactly equal or are different by a constant value.

3.1.7 Receiver Operator Curve Area (ROC)

One of the commonly used saliency map comparison metrics in the literature is the ROC area [11, 24, 28, 33, 51, 52, 55, 56]. This metric determines how well salient and non-salient regions of the image can be discriminated by their saliency value in a PSM using a simple threshold [52]. Similar to "*Hit Rate*", the most salient 20% of the image based on its RHSM are selected as salient regions of the image, and the rest are called non-salient. Also, a binary map is created by thresholding the PSM. The threshold is increased gradually from the minimum of the map to its maximum, which changes both the hit rate (labeling salient locations as salient by the PSM) and false alarm rate (labeling a non-salient location as salient). The ROC is a curve that plots the false alarm rate as a function of the hit rate ($0 \leq Hit\ Rate\ \&\ False\ alarm\ rate \leq 1$). The area under the ROC is a well-known measure of similarity between two saliency maps.

3.1.8 Kullback-Leibler Divergence (NKL)

The Kullback-Leibler divergence value introduced in (3-9) is used as a measure of dissimilarities between two maps in [17, 57].

$$KL(p||h) = \sum_x p(x)\, log\left(\frac{p(x)}{h(x)}\right) \quad (3\text{-}9)$$

where $p(x)$ is the predicted probability density function calculated from the PSM. $h(x)$ is the probability density drawn from the RHSM of the image. KL is generally used to measure distance between two probability distributions. NKL, defined as

$$NKL = 1 - KL \tag{3-10}$$

is used in this report to find similarities between two maps.

3.1.9 Score

Define average fixation saliency ($\bar{\boldsymbol{M}}_{fix}$) obtained when sampling the PSM at the fixations of human observers, and average saliency ($\mu_{\boldsymbol{M}}$) as the mean of the PSM. Then Score [58] is defined as:

$$Score = \left(\bar{\boldsymbol{M}}_{fix} - \mu_{\boldsymbol{M}}\right)/\mu_{\boldsymbol{M}}, \tag{3-11}$$

$Score$ is also used in [15, 59] to capture similarities between PSMs and RHSMs. In this report, Score values are normalized to lie in the interval [0 1]. A similar metric is used in [60].

3.2 Evaluating Saliency Map Comparison Metrics

There are several saliency map comparison metrics in the literature, but it can be easily shown that results from these in ranking different saliency models do not agree. The best visual saliency model identified by one comparison metric might show poor results when evaluated by another metric. Therefore, before ranking visual saliency models, it is important to identify which metric could be considered the best. The database created in [40] is used herein to design an evaluation procedure for comparison metrics. The longest dimension of each image in the data base is 1024 and the other dimension varies from 405 to 1024, with the majority having 768 pixels.

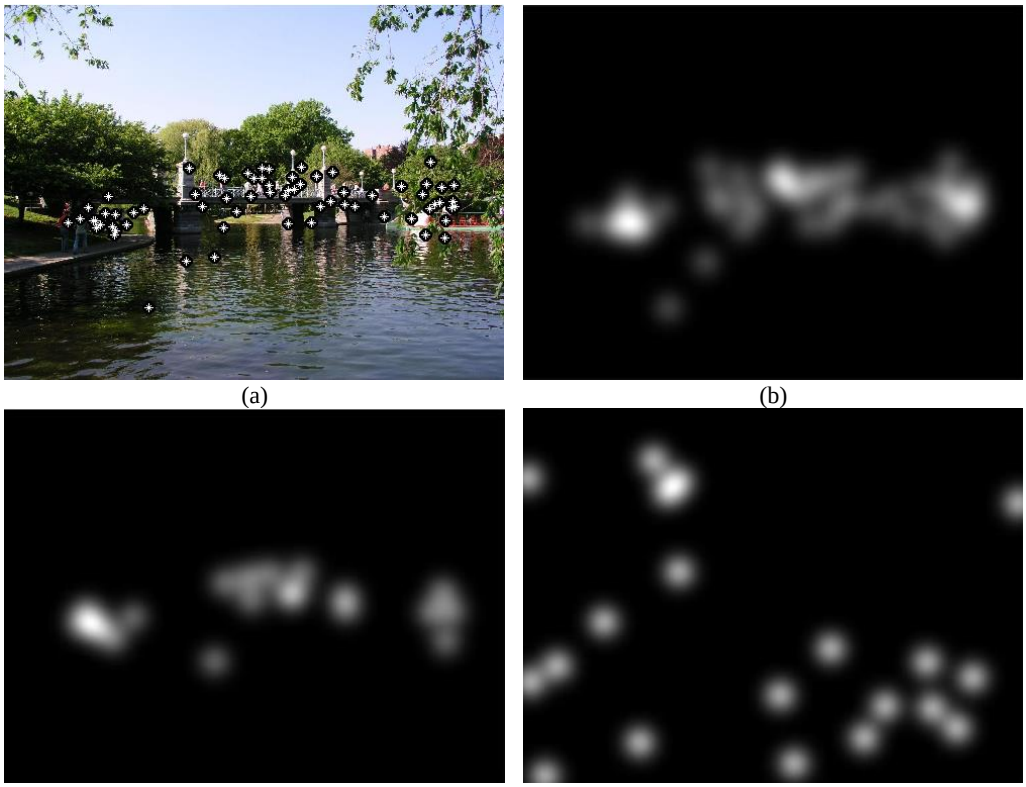
3.2.1 A Method to Evaluate Comparison Metrics

It is commonly accepted that the best saliency map for an image is the one created using human observers' fixation data. In the LPWHL study [40], the first 5 fixations of 15 observers has been recorded for each image, and a fixation map database is created for 1003 images. Using this

database, two fixation maps are defined for each image herein. Randomly selecting a simple majority of fixations for a given image, a *reference fixation map* is created. The remaining fixations for that image are used to generate another map called the *human fixation map*. Saliency maps are generated from these fixation maps by convolving them with a 2-D Gaussian function, namely the Gaussian function used in [40]. The saliency map computed from the human fixation map is called the Human Saliency Map (HSM), and the saliency map computed from the reference fixation map is designated the RHSM. Accordingly, we propose for our first criterion a good comparison metric would be expected to find HSMs analogous to the corresponding RHSMs.

Observers in the LPWHL study in [40] might have had different priorities during the experiment. It has been proved [5] that given the same image, fixations and patterns of saccade do change for different questions that were asked of an observer prior to viewing the image, which is believed to be a property of top-down saliency [2], [5]. Accordingly, we believe it is important to select fixations randomly for the reference fixation map, instead of choosing all fixations of a fixed sample of observers.

In this project, it has been assumed that the worst fixation map for a specific image is a random selection of image points that are designated as "random fixation points", when in fact they typically would not be true fixation points. Such a map is generated for each image by randomly selecting some locations across the image as "fixations". For a given image, we select the same number of random "fixations" as true fixations in the corresponding HSM. Similar to HSMs and RHSMs, a map is created convolving this random fixation map with the 2-D Gaussian function used in [40], and it is designated a Random Saliency Map (RSM). Our second criterion for a good comparison metric is that it should clearly distinguish an RSM for a given image as dissimilar from an RHSM for that image. Figure 3-1 shows an image along with an RHSM, HSM and an RSM created for this image.



(a) (b)

(c) (d)

Figure 3-1. Original image with fixation points [40] (a); an RHSM with 55 random fixations (b); the remaining HSM with 20 fixations (c); and an RSM with 20 fixations (d).

3.2.2 Comparison Metric Evaluation Results

In this Section, for each image with human fixation points from [40] (with 75 fixation points for each image), reference fixation maps are created using $N_{Ref.fix.} = 40, 45, \dots, 70$ fixations chosen randomly from the fixation database. The remaining fixations in the database (35, 30,…, 5 fixations) are used to create human fixation maps. Then for each image, a number of random points are selected equal to the number of fixations chosen for the human fixation map. Three sets of examples of RHSM, HSM and RSM for $N_{Ref.fix.} = 40, 55$ and 70, as well as the average over 100 maps are shown in Figures 3-2, 3-3 and 3-4. All maps are normalized to the [0 1] interval before comparison.

As shown in Figures 3-2, 3-3 and 3-4, the higher is the number of fixations used in creating RHSM and HSM, the closer are these maps to the original saliency map. The original saliency map is the map created using all fixations in the database, demonstrated in the Figures 3-2, 3-3

and 3-4 (b). Although HSMs created using 5 fixations do not seem similar to the original saliency map, the average maps RHSMs and HSMs (over 100 samples), shown in the right column of Figures 3-2, 3-3 and 3-4, are very similar to the original saliency map. As expected, the average of the RSMs is still a random map and different regions are highlighted randomly.

Since the number of salient pixels usually varies in different saliency maps, Judd et al. [61] suggest matching the histogram of the saliency maps created for an image with the histogram of the reference saliency map of the image before comparing them together. Histogram matching, or histogram specification, is a technique to modify the histogram of a grayscale image in a way that its histogram matches a specified histogram [62]. Assuming that intensity levels of the image are continuous in the range [0 1], $p_r(r)$ denotes the probability density function of the intensity levels of a given image. The following transformation modifies the image in a way that its intensity levels are equally likely:

$$T(r) = \int_0^r p_r(w)dw \tag{3-12}$$

Now, if we desire to have the histogram of the output image matching a specific histogram $p_z(z)$, defining $H(z)$ as

$$H(z) = \int_0^z p_z(w)dw \tag{3-13}$$

$z = H^{-1}[T(r)]$ gives the new intensity value in the output image of those pixels that have intensity value of r in the input image.

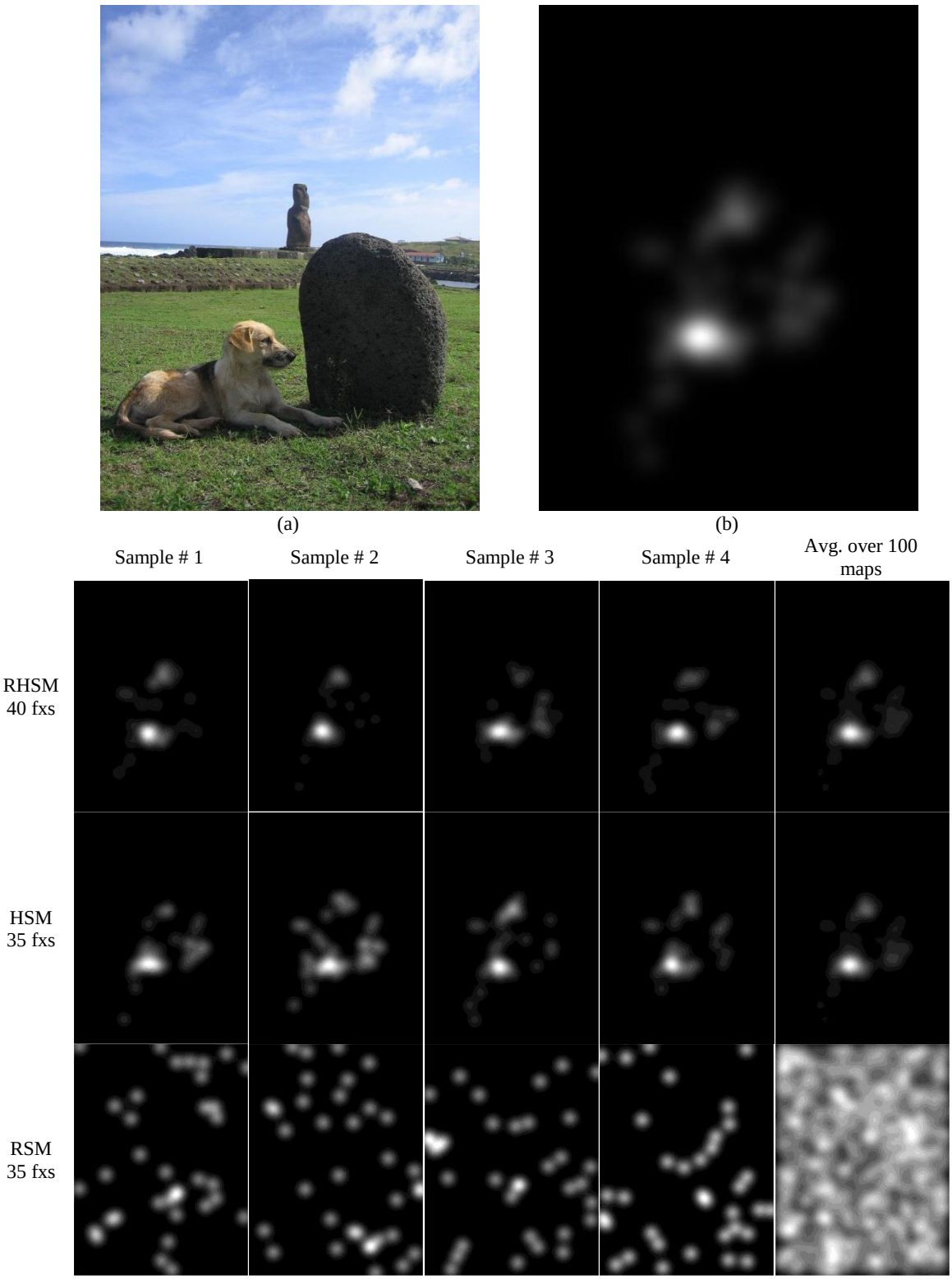


Figure 3-2. Original image (a); Original saliency Map with 75 fixation points (b); four samples of RHSM, HSM and RSM for $N_{Ref.fix.} = 40$; and the average of RHSMs, HSMs and RSMs over 100 samples.

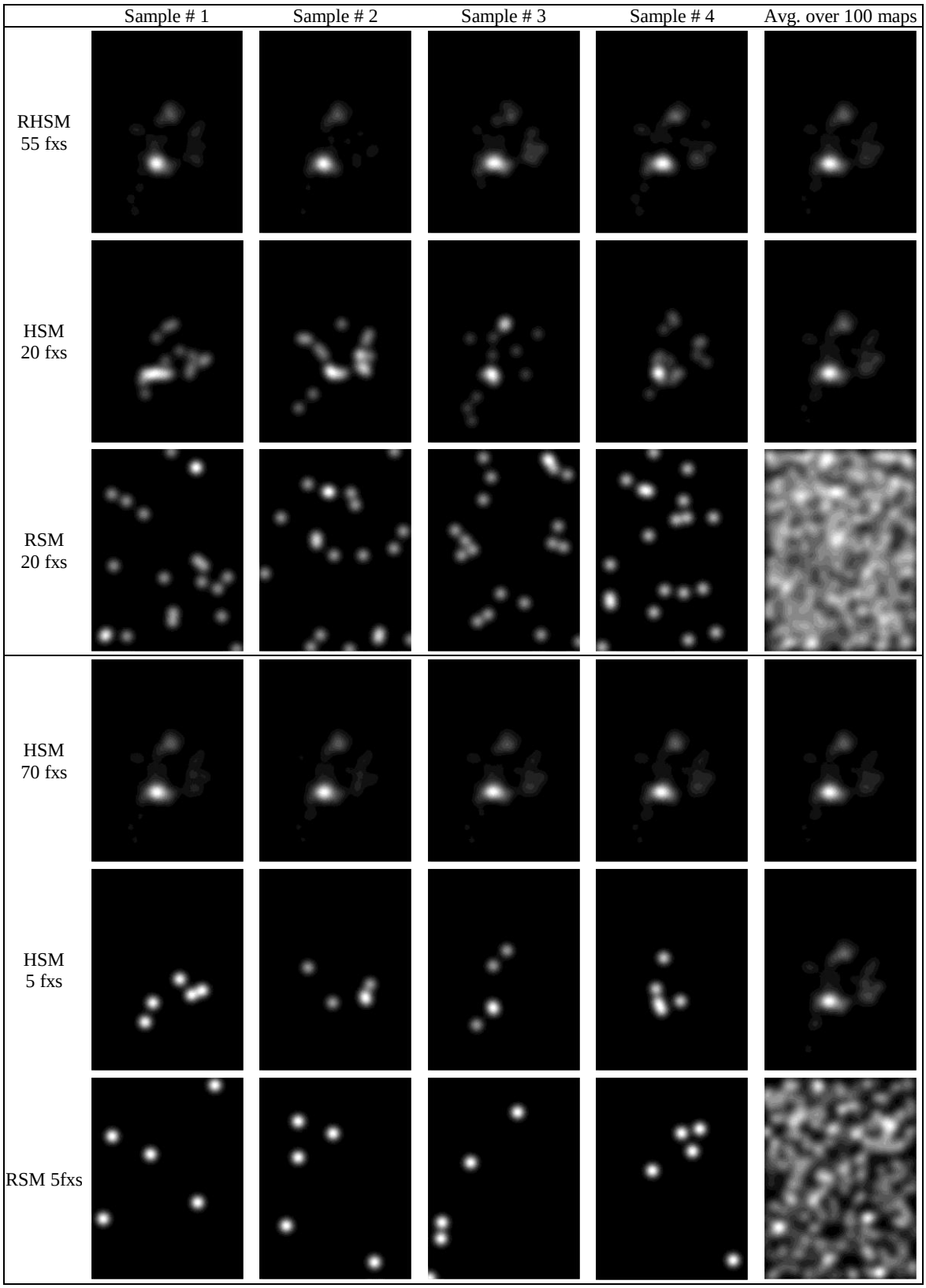


Figure 3-3.Four samples of RHSM, HSM and RSM for the image shown in Figure 3-2 (a) with $N_{Ref.fix.} =$ 55 and 70; and the average of RHSMs, HSMs and RSMs over 100 samples.

After matching histograms of some saliency maps with a specific reference map, the number of salient pixels and saliency distributions in different saliency maps would be very close together. It creates a fairer basis for comparing saliency maps. Figure 3-4 shows an RHSM with 50 fixations (c), HSM and RSM (d) and (f) with 25 fixations, and the HSM and RSM after histogram matching (e) and (g). Cumulative frequency distributions with 256 bins for these maps are plotted in Figure 3-4 (h). For simplicity in analyzing graphs, in the cumulative frequency distribution curves, the numbers of zeros in the maps are not counted.

As demonstrated in Figure 3-4 (h), although the number of fixations in the sample HSM and RSM are equal, and fixation maps are convolved with the same Gaussian filter, the number of salient points in the RSM is almost twice the HSM. This is caused by the fact that in the random fixation maps, fixations are distributed widely and usually far from each other. On the other hand, in the human fixation maps, fixations are distributed mostly around salient locations in the central parts of the map. The number of salient pixels in HSM and RSM will be close to the number of salient pixels in the RHSM after histogram matching, as shown in Figure 3-4 (h). RHSMs, HSMs and RSMs are normalized to the [0 1] interval before histogram matching.

Two approaches were taken in our investigations. All comparison metrics were employed to compare RSMs and HSMs with the corresponding RHSMs before and after histogram matching. This process was repeated 100 times, with seven $N_{Ref.fix.}$ values for all images in the database (702,100 repetitions) and the results are illustrated in Figure 3-5 and Figure 3-6. 100 repetitions are chosen so that the results do not change as the number of repetitions increases from 70 to 100. RHSMs, HSMs and RSMs are normalized to the [0 1] interval before comparison.

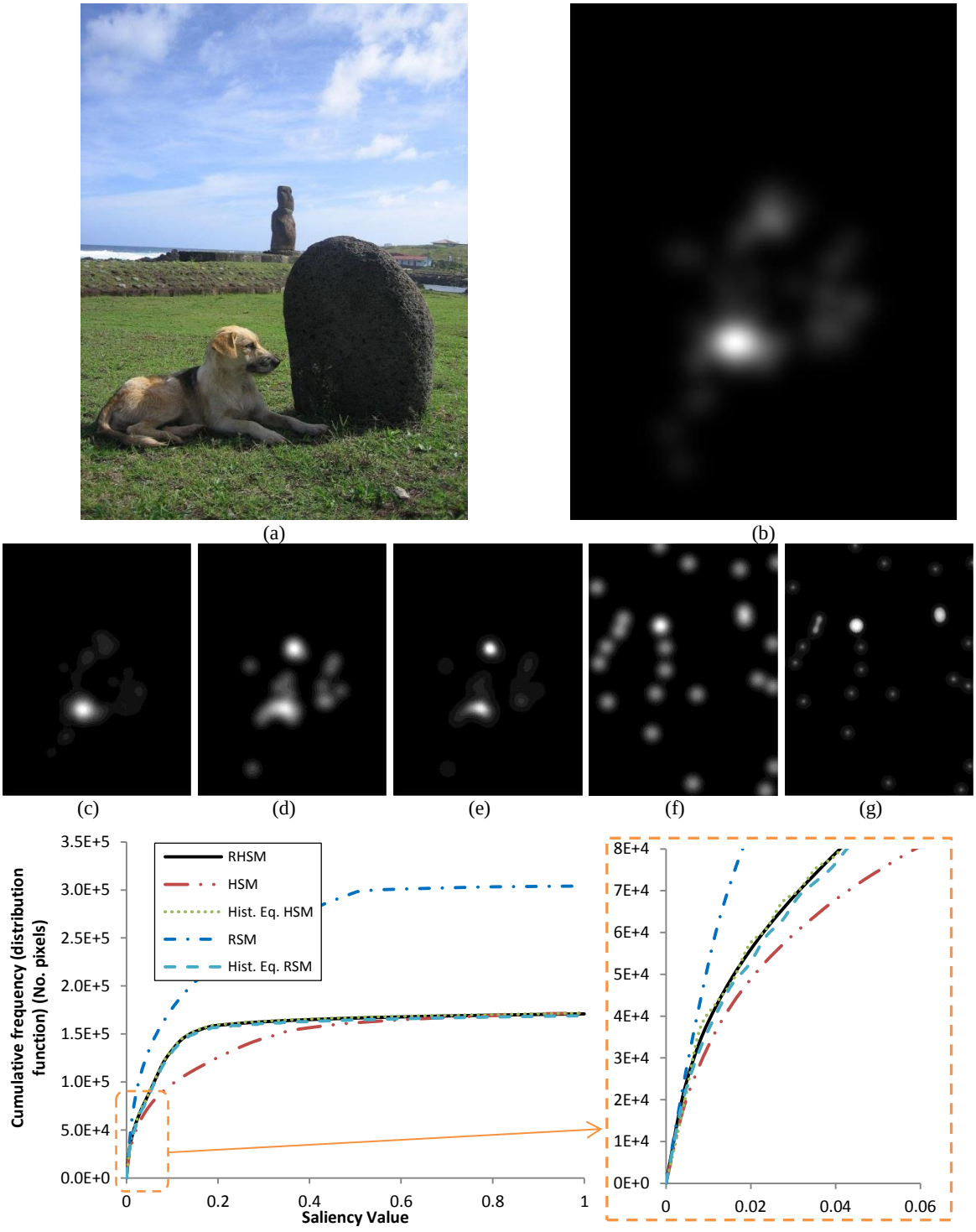


Figure 3-4. Original image (a); Original saliency Map with 75 fixation points (b); RHSM with $N_{Ref.fix.} = 50$ (c); HSM and RSM with 25 fixations (d) and (f); HSM and RSM after histogram matching (e) and (g); cumulative frequency of (c) to (g) in bottom plots (h).

In Figures 3-5 and 3-6, solid red curves give the histograms of all comparison metrics applied to compare HSMs with RHSMs, and dashed blue curves give the histograms of the comparison

results between RSMs and RHSMs. Histograms were generated using 1000 bins. All comparison metrics compute scalar values in the [0 1] interval. Therefore, good metrics are expected to produce high values when comparing HSMs with RHSMs, and low values when comparing RSMs with RHSMs. Accordingly, we designate the best comparison metric as the one which discriminates the best between HSMs and RSMs. We use thresholding to determine how HSMs are discriminated from RSMs by comparison metrics.

The vertical dashed black lines in Figures 3-6 and 3-7 show the threshold values for which RSMs and HSMs can be classified with minimum error. The threshold for each metric is given by:

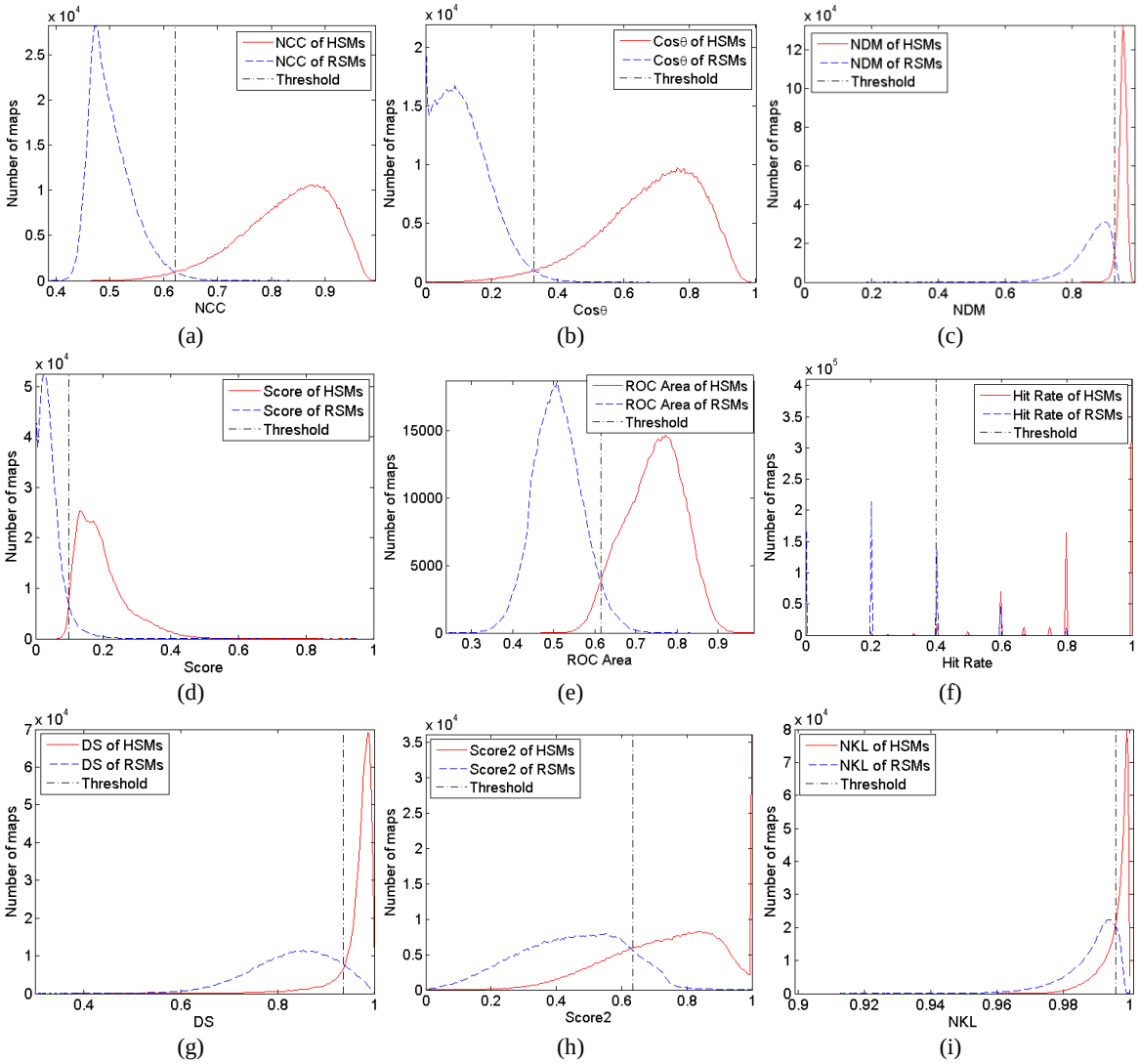


Figure 3-5. Histograms of all comparison metrics comparing RSMs and HSMs with RHSMs without histogram matching.

$$Threshold = \underset{Thr}{\text{Argmin}}\left[\int_0^{Thr} y_1 dx + \int_{Thr}^{1} y_2 dx\right] \quad (3\text{-}14)$$

where y_1 is the HSM histogram and y_2 is the RSM Histogram, namely the red and blue curves in Figures 3-5 and 3-6, respectively; x is the metric value (horizontal axes in Figures 3-6 and 3-7); and Thr stands for threshold. For example in Figure 3-5 (a), the *NCC* threshold value is 0.621 for which *NCC* produces results larger than this threshold for 688,915 of HSMs (out of 702,100) and produces results smaller than this threshold for 693,443 of RSMs. These numbers represent, respectively, the area under the red curve from the threshold to 1 and the area under the blue

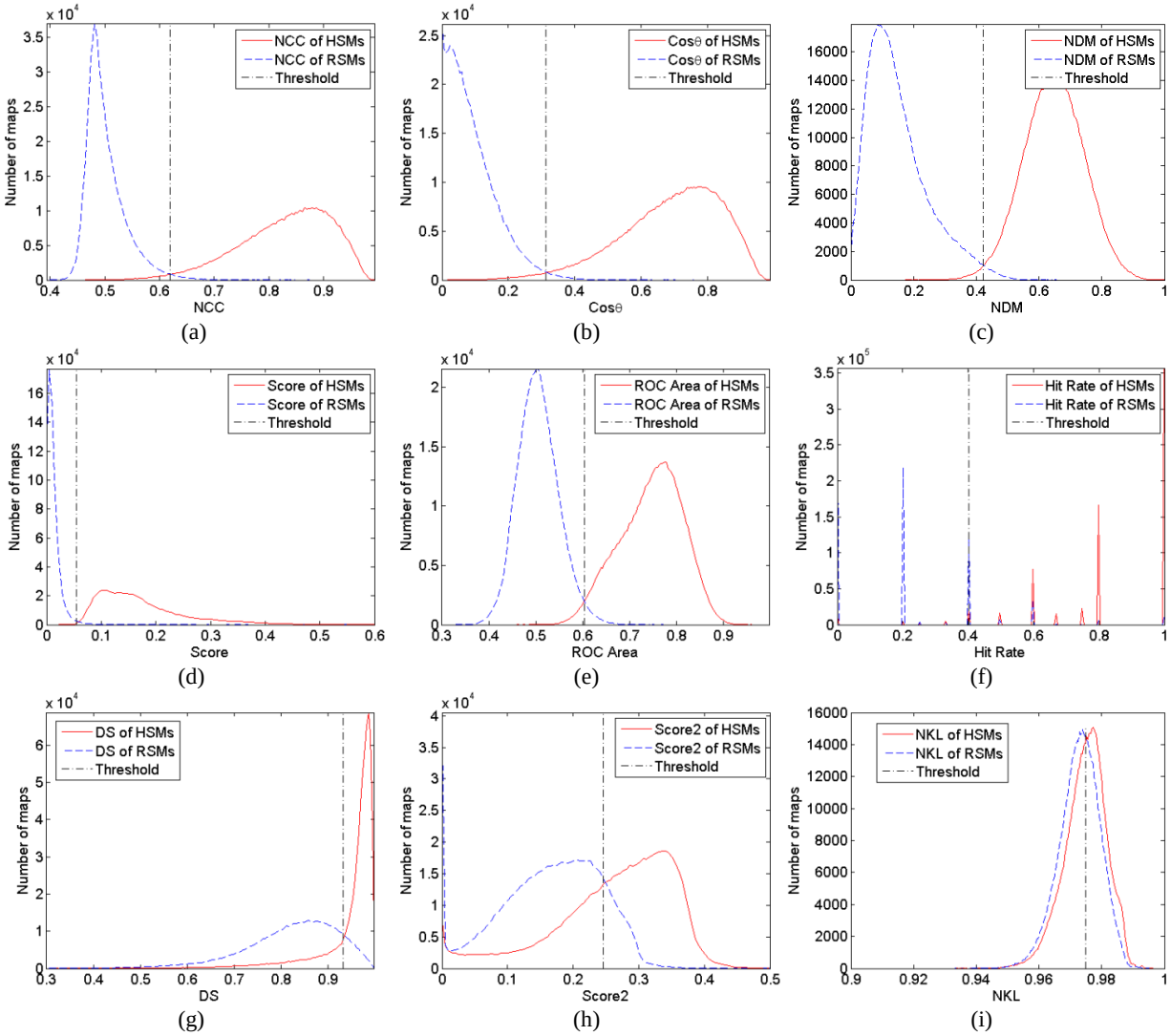


Figure 3-6. Histograms of all comparison metrics comparing RSMs and HSMs with RHSMs after histogram matching.

curve from zero to the threshold. Accordingly, we consider *NCC* as correctly classifying 688,915 HSMs and 693,443 RSMs, and the rest (13,185 HSMs and 8,657 RSMs) are misclassified.

As shown in Figure 3-6, histogram matching highly affects the results from *NDM* (c), *Score* (d), *ROC Area* (e) and *NKL* (i). It decreases the overlap of the RSM curves with HSM curves for *Score*, *NDM* and *ROC Area*. However, it highly increases the overlap of the curves of the *NKL* graph. The percentage of misclassifications and related threshold values are shown in Table 3-1 for each of the metrics. We designate the best metric as that which produces the lowest misclassification percentage.

Table 3-1 ranks the metrics for both cases, namely with and without histogram matching, according to their misclassification percentage, with evaluation rank 1 being the best.

Table 3-1. Percentage of misclassification error and the threshold value computed for each comparison metric

Comparison metric	Percentage of misclassification error without histogram matching	Threshold value without histogram matching	Evaluation rank without histogram matching	Percentage of misclassification error with histogram matching	Threshold value with histogram matching	Evaluation rank with histogram matching
Score	3.90	0.098	4	0.634	0.0553	1
NDM	3.55	0.928	3	1.2	0.4221	2
NCC	1.56	0.621	1	1.39	0.6196	3
$\boldsymbol{Cos\theta}$	1.89	0.329	2	1.5	0.3126	4
ROC Area	4.16	0.614	5	1.96	0.6035	5
Hit Rate	6.09	0.400	6	7.77	0.402	6
DS	11.22	0.936	7	15.6	0.9332	7
Score2	20.62	0.635	8	21.27	0.2462	8
NKL	21.45	0.996	9	43.56	0.9749	9

Table 3-1 shows that without histogram matching, the best metrics are *NCC* and $Cos\theta$, which result in misclassification percentages of 1.56% and 1.89%, respectively. However, we believe matching the histograms of the saliency maps with RHSMs is essential in creating an impartial evaluation. Accordingly, since *Score* creates minimum misclassification error after histogram matching (0.634%), we designate *Score* as the best metric for comparing saliency maps. On the other hand, *NKL* (Kullback-Leibler divergence) with 21.45% and 43.56% misclassification,

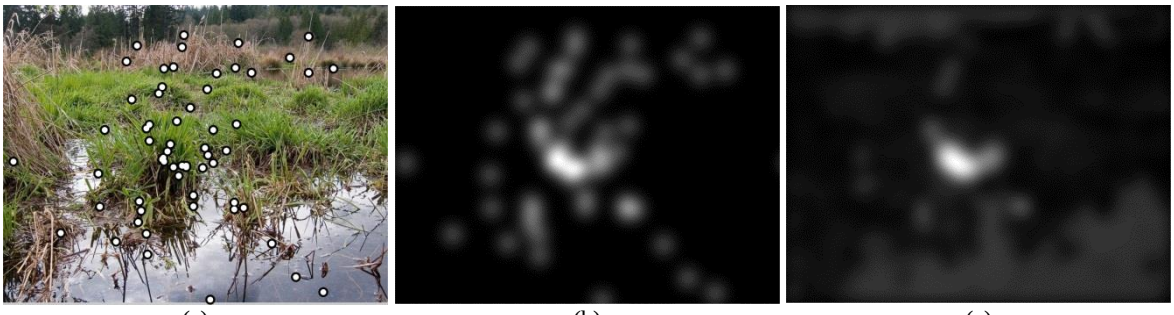
without and with histogram matching respectively, is the worst metric. *NKL* compares the probability density functions of the saliency maps. After histogram matching probability density functions of RSMs and HSMs would be very similar to each other, such that *NKL* cannot distinguish between them.

As shown in (3-1) and (3-7), if the average of the saliency maps are 0, $Cos\theta$ and *NCC* would be identical. Therefore, we expected them to produce similar results. Table 3-1 demonstrates that their results are very similar for both cases.

3.3 Discussion

By employing a novel method to evaluate comparison metrics, the best metrics were found to be *NCC* (the normalized value of the correlation coefficient between two maps) and *Score* for comparison without and with histogram matching, respectively. *NCC* and *Score* produce minimum misclassification errors on discriminating human saliency maps from random saliency maps. Interestingly, two commonly used comparison metrics, *ROC Area* and Kullback-Leibler divergence (*NKL*), were ranked 5th and 9th among 9 metrics.

NCC is based on the correlation coefficient between two maps and is a general metric to compare any two 2-D matrices. It performed well on comparing saliency maps, producing 1.39% and 1.56% misclassification error with and without histogram matching, respectively. However, *Score* is a metric designed to compare PSMs with HRSMs. It examines at every human fixation in a PSM, averages PSM values, and produces a scalar that shows how well PSMs mimic RHSMs.



(a) (b) (c)

Figure 3-7 Original image (a); its RHSM (b); and a PSM for the image (c).

Accordingly, *Score* can be considered a more demanding metric and we select it as the best saliency map comparison metric. Figure 3-7 shows an image from the LPWHL database (a), its RHSM created with all 75 fixations in the dataset (b), and a PSM created for this image (c).

Saliency maps shown in Figure 3-7 (b) and (c) appear similar, and both highlight the central regions of the image. When comparing PSM (c) with RHSM (b) the *NCC* value is 0.793, which is fairly high compared with the *NCC* threshold of 0.621. However, since many scattered fixation locations in the RHSM are not highlighted in the PSM, the *Score* value is 0.0470 which is less than the threshold of 0.0553. This example shows that while *NCC* classifies (c) as a very good saliency map for the image in (a), *Score* classifies (c) as a random saliency map.

CHAPTER IV

4 FINDING THE BEST VISUAL SALIENCY MODEL

4.1 Visual Saliency Mechanisms on Natural Images

All visual saliency models summarized in Chapter I and our EH model explained in Chapter II have been applied to all the 1003 images in the LPWHL database [40]. Their codes were downloaded from their websites, except for the CC model [17] for which the authors provided us with their saliency maps on the LPWHL database. Figure 4-1 gives pictorial saliency results for one image without histogram matching, and Figure 4-2 gives results for the same image after

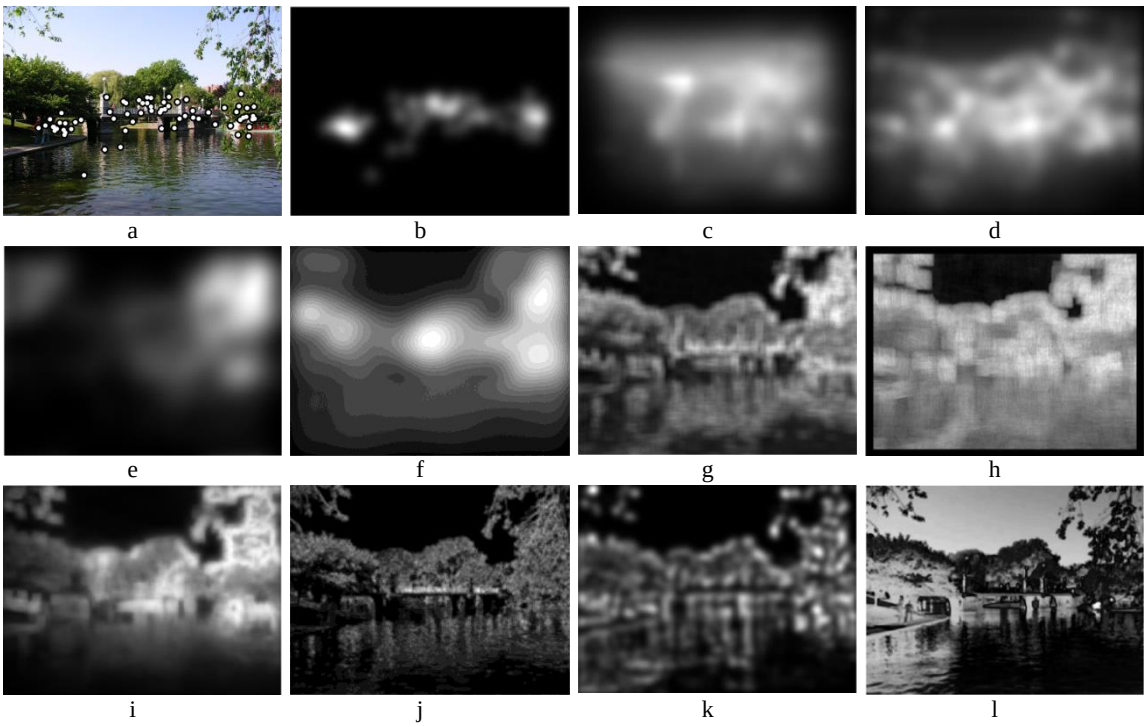


Figure 4-1. Original image with fixation points [40] (a); RHSM (b); and original PSMs using GBVS (c), EH (d), CC (e), IS (f), CASD (g), AIM (h), SUN (i), SR (j), IK (k) and FTSRD (l).

histogram matching. Abbreviations for each of these methods are given in Chapters I and II.

As discussed before and demonstrated in Figure 4-1, the number of highlighted (salient) pixels returned by each visual saliency model for an image varies greatly. Herein, two approaches will be used to test performance of the visual saliency mechanisms. In the first approach, PSMs will be compared with RHSMs using *NCC* without histogram matching. In the second approach, histogram matching is used to match histograms of the PSMs with the RHSM histograms before comparison and are compared using *Score*. This helps us find locations that each model finds most salient and also establish a fair starting point to analyze the performance of visual saliency models. In this section, the RHSM for each image in the database is computed using *all* of the fixation points for that image.

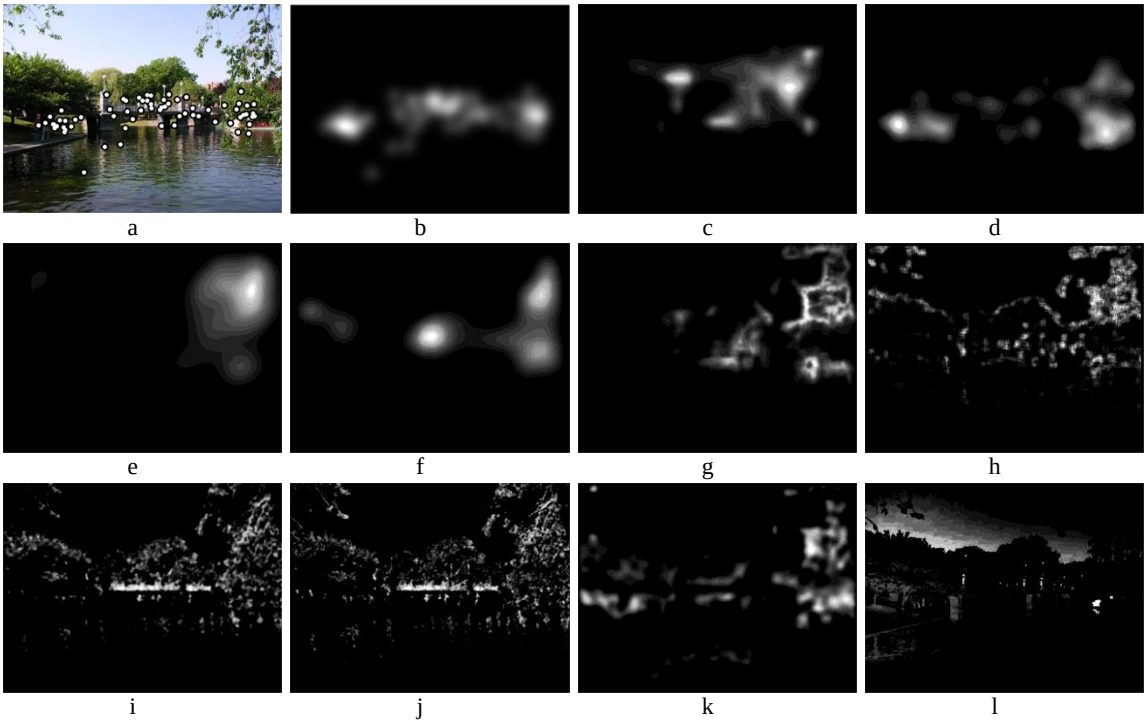


Figure 4-2. Original image with fixation points [40] (a); RHSM (b); and PSMs using GBVS (c), EH (d), CC (e), IS (f), CASD (g), AIM (h), SUN (i), SR (j), IK (k) and FTSRD (l) after histogram matching.

4.1.1 Comparisons without Histogram Matching

In this section, saliency maps are compared to RHSMs without histogram matching. The PSM for each image produced by each method is compared with the RHSM for that image using the *NCC*

comparison metric (the metric with minimum misclassification percentage without histogram matching). Figure 4-3 present box-plots of these *NCC* results.

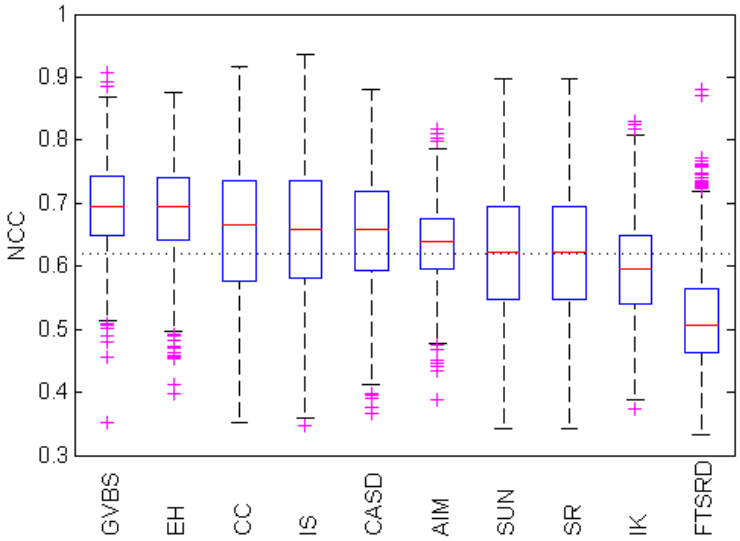


Figure 4-3. *NCC* box-plots for all visual saliency models.

In each box in Figure 4-3, the central horizontal red line is the median, the higher the median, the better the performance of the visual saliency model. The lower and upper edges of the box mark the 25th (q_1) and 75th (q_3) percentiles, respectively, the horizontal whiskers (if plotted) show ±2.7 σ (or 99.3% coverage if the data are normally distributed) and magenta pluses show outliers. In boxplots, data-points larger than $q_3 + 1.5 \times (q_3 - q_1)$ or smaller than $q_1 - 1.5 \times (q_3 - q_1)$ are drawn as outliers. The horizontal dotted line (*NCC* =0.621) shows the threshold for *NCC* in Table 3-1 that discriminates random maps from human saliency maps.

Accordingly, saliency maps for which *NCC* is larger than the threshold are considered more similar to the corresponding RHSM, the larger indicating the more similar. Maps for which *NCC* is less than the threshold line are closer to an RSM than the corresponding RHSM. For example, using the threshold value of 0.621, 164 saliency maps (out of 1003) computed by the GBVS algorithm were found more similar to an RSM than the corresponding RHSM. For each visual saliency model, the mean of *NCC* over the entire database and the number and percent of maps classified as RSM are shown in Table 4-1.

Table 4-1. The mean of the NCC values computed for all visual saliency models and the number of maps classified as RSM (threshold used: 0.621, see Table 3-1) on 1003 images.

Visual saliency model	Mean of *NCC*	Standard Deviation of *NCC*	Number (%) of maps classified as RSM	Ranking based on the number of maps classified as RSM
GBVS	0.692	0.076	164 (16.35%)	1
EH	0.688	0.068	174 (17.35%)	2
CC	0.657	0.104	352 (35.09%)	3
IS	0.661	0.108	357 (35.59%)	4
CASD	0.655	0.093	357 (35.59%)	4
AIM	0.636	0.063	390 (38.88%)	6
SUN	0.623	0.079	499 (49.75%)	7
SR	0.624	0.065	500 (49.85%)	8
IK	0.595	0.089	617 (61.52%)	9
FTSRD	0.520	0.079	879 (87.64%)	10

As shown in Table 4-1, the GBVS model produces the maximum mean of the *NCC* metric and the minimum number of maps classified as RSM. Our visual saliency model (EH) is ranked 2nd in this table. In fact, GBVS and EH reduce by more than one-half the misclassification percentage of the number 3 ranked saliency model, namely CC. Moreover, the models ranked 3 through 6 are very close in misclassification percentages, such that the performances of GBVS and EH are even more impressive. Accordingly, we select GBVS (Graph Based Visual Saliency) as the best bottom-up visual saliency model for evaluation without histogram matching. The saliency model FTSRD, with an *NCC* mean of 0.520 and 879 maps classified as RSM, is the lowest ranked visual saliency model.

To check if there are statistically significant differences between models, first the normality of their *NCC* values are checked using the Shapiro-Wilk method [63]. Only, GBVS and IK results are normally distributed with a significance level of 0.05. Accordingly, one of the best non-parametric methods, the Wilcoxon test [64], was used and results are demonstrated in Table 4-2 with 0 for no statistically significant difference and 1 for significant difference.

Table 4-2 shows that there is no significant difference between GBVS and EH and they are statistically different with the rest. No significant difference was found between CC, IS and

Table 4-2. Wilcoxon test results on *NCCs* of all visual saliency models.

Saliency Model	GBVS	EH	CC	IS	CASD	AIM	SUN	SR	IK	FTSRD
GBVS	---	0	1	1	1	1	1	1	1	1
EH	0	---	1	1	1	1	1	1	1	1
CC	1	1	---	0	0	1	1	1	1	1
IS	1	1	0	---	0	1	1	1	1	1
CASD	1	1	0	0	---	1	1	1	1	1
AIM	1	1	1	1	1	---	1	1	1	1
SUN	1	1	1	1	1	1	---	0	1	1
SR	1	1	1	1	1	1	0	---	1	1
IK	1	1	1	1	1	1	1	1	---	1
FTSRD	1	1	1	1	1	1	1	1	1	---

CASD, and between SUN and SR. AIM, IK and FTSRD are significantly different with all models.

As a further note, we conducted a similar study using other comparison metrics. Each metric average and the ranking based on the metric average in parentheses (the lower the rank number, the better) for all visual saliency models are shown in Table 4-3. The number of maps classified as RSM based on each metric and the ranking based on the number of RSM maps in parentheses are shown in Table 4-4 as well as the average number of maps classified as RSM based on all saliency comparison metrics.

As demonstrated in Tables 4-3 and 4-4, visual saliency models are ranked differently based on different comparison metrics. This shows the importance of finding the best comparison metrics before ranking visual saliency models. However, the 2nd ranked metric, namely $Cos\theta$, found the ranking of the visual saliency models similar to the 1st ranked metric, *NCC*. A method to compare visual saliency models based on all comparison metrics at the same time is comparing their average number of maps classified as RSM. In Table 4-4, right column, the average number RSMs for each visual saliency model is shown as well as their rankings in parenthesis. The

ranking is similar to the ranking based on *NCC*, and GBVS produces the minimum number of RSMs. The only difference is that SR is ranked 4th above IS instead of 8th.

Table 4-3. Average of each metric for all visual saliency models and their rankings based on the average of each metric.

		Comparison metrics in increasing rank number (from left to right)								
		NCC	***Cosθ***	***NDM***	***Score***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***
Visual saliency mechanisms in increasing rank number (top to bottom) based on *NCC*	**GBVS**	0.692 (1)	0.476 (1)	0.67 (3)	0.0595 (3)	0.821 (1)	0.548 (2)	0.918 (1)	0.176 (2)	0.993 (7)
	EH	0.688 (2)	0.456 (2)	0.584 (8)	0.0705 (1)	0.757 (3)	0.527 (3)	0.908 (2)	0.173 (3)	0.995 (2)
	IS	0.661 (3)	0.438 (3)	0.503 (10)	0.0412 (7)	0.796 (2)	0.501 (4)	0.864 (3)	0.18 (1)	0.994 (4)
	CC	0.657 (4)	0.436 (4)	0.623 (4)	0.0609 (2)	0.753 (5)	0.552 (1)	0.861 (4)	0.17 (7)	0.997 (1)
	CASD	0.655 (5)	0.428 (5)	0.62 (5)	0.0574 (4)	0.756 (4)	0.498 (5)	0.851 (6)	0.172 (5)	0.994 (3)
	AIM	0.637 (6)	0.388 (6)	0.514 (9)	0.0379 (8)	0.682 (6)	0.481 (6)	0.84 (8)	0.163 (9)	0.993 (8)
	SUN	0.623 (7)	0.379 (7)	0.61 (6)	0.0444 (6)	0.645 (8)	0.431 (9)	0.854 (5)	0.172 (6)	0.994 (4)
	SR	0.624 (8)	0.363 (8)	0.921 (1)	0.0531 (5)	0.665 (7)	0.444 (8)	0.849 (7)	0.173 (3)	0.994 (4)
	IK	0.595 (9)	0.25 (9)	0.584 (7)	0.034 (10)	0.589 (9)	0.473 (7)	0.822 (9)	0.166 (8)	0.993 (7)
	FTSRD	0.52 (10)	0.25 (10)	0.672 (2)	0.0361 (9)	0.521 (10)	0.344 (10)	0.796 (10)	0.153 (10)	0.993 (7)

Table 4-4. Number of maps classified as RSM for all visual saliency models and their rankings based on their number of RSMs (for threshold values see Table 3-1).

		Comparison metrics in increasing rank number (from left to right)									
		NCC	***Cosθ***	***NDM***	***Score***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***	**Average**
Visual saliency mechanisms in increasing rank number (top to bottom) based on *NCC*	**GBVS**	164 (1)	90 (1)	996 (7)	583 (2)	17 (1)	406 (2)	506 (1)	464 (4)	991 (7)	468.6 (1)
	EH	174 (2)	94 (2)	975 (4)	606 (3)	131 (3)	315 (1)	615 (2)	433 (2)	970 (4)	479.2 (2)
	CC	352 (3)	189 (3)	994 (6)	572 (1)	144 (4)	433 (3)	701 (6)	420 (1)	542 (1)	483 (3)
	IS	357 (4)	225 (4)	998 (9)	772 (7)	23 (2)	503 (5)	669 (3)	453 (3)	997 (9)	555.2 (5)
	CASD	357 (4)	228 (5)	996 (7)	611 (4)	150 (5)	497 (4)	711 (7)	480 (5)	979 (6)	556.6 (6)
	AIM	390 (6)	230 (6)	999 (10)	875 (8)	239 (6)	515 (6)	757 (9)	506 (6)	999 (10)	612.2 (7)
	SUN	499 (7)	333 (7)	972 (2)	767 (6)	355 (8)	572 (8)	693 (5)	506 (6)	974 (5)	630.1 (8)
	SR	500 (8)	371 (8)	421 (1)	664 (5)	303 (7)	577 (9)	690 (4)	518 (8)	951 (3)	555 (4)
	IK	617 (9)	770 (9)	974 (3)	878 (9)	625 (9)	528 (7)	756 (8)	521 (9)	759 (2)	714.2 (9)
	FTSRD	879 (10)	786 (10)	982 (5)	888 (10)	797 (10)	671 (10)	830 (10)	602 (10)	994 (8)	825.4 (10)

4.1.1.1 Optimizing Blurriness and Center-Bias

Judd et al. [61] and Borji et al. [65] stated that visual saliency models that create blurrier saliency maps usually are ranked higher than models that create saliency maps with sharp edges. Also, maps that are biased towards the image center tend to gain better results than others at predicting HRSMs. In this section, we optimize the level of blurriness and center-bias of each visual saliency model by varying appropriate parameters characterizing these effects and choosing the parameter values that maximize the performance of the visual saliency model. This creates the opportunity to compare visual saliency models at the best levels of blurriness and degree of center-bias for each model. For blurriness, PSMs are convolved with 2D Gaussian filters with $\sigma = 10, 20, 30, \dots, 150$ pixels to produce a map designated $\boldsymbol{M}_{Blurred}$. Then a weighted Center-Map ($\boldsymbol{CM}$) is added to a weighted $\boldsymbol{M}_{Blurred}$ to produce a new saliency map $\boldsymbol{M}_{New}$, using the following equation as suggested in [61]:

$$\boldsymbol{M}_{New} = w \times \boldsymbol{CM} + (1 - w) \times \boldsymbol{M}_{Blurred} \tag{4-1}$$

where $w = 0, 0.1, 0.2, \dots, 1$, and the closest 2D Gaussian blob to the average of all RHSMs, shown in Figure 4-14, is selected as the $\boldsymbol{CM}$. $w = 0$ indicates that the new map is identical to the blurred map, and $w = 1$ indicates the new map is equal to the center map. NCC values are calculated for new saliency maps by comparing them with the corresponding RHSM. As an example of this process, Figure 4-4 shows a sample image, its RHSM, PSMs produced by the SR visual saliency model, blurred and center-biased saliency maps, and their NCC results.

As shown in Figure 4-4, the SR saliency model benefits from a certain degree of blurriness and center-bias in this example with highest value of $NCC = 0.795$ for $\sigma = 20$ and $w = 0.2$. PSMs created by all visual saliency models are modified using (4-1) and the *NCC* metric is used to compare saliency maps to RHSMs. Figure 4-5 shows how the average *NCC* and number of maps classified as RSM for each visual saliency mechanism changes by changing the blurring value σ and weight w of the center-map.

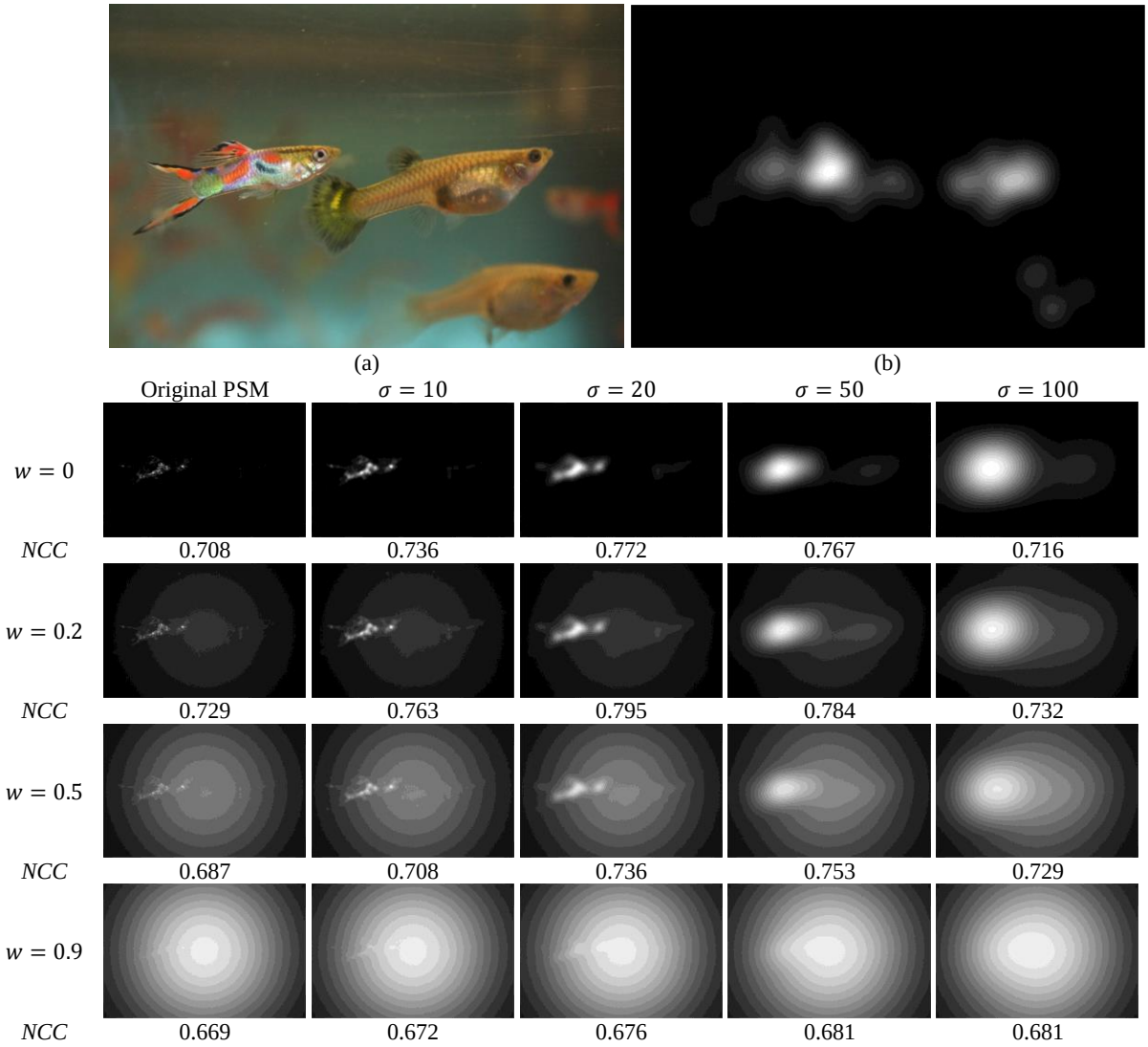


Figure 4-4. Original image (a); its RHSM (b); blurred and center-biased maps and their NCC values using SR.

In Figure 4-5, each data point in the graphs shows results of one visual saliency model averaged over the entire LPWHL database [40]. As shown in Figure 4-5, *w* has more effect than σ on both the average *NCC* and number of maps classified as RSM. Table 4-5 gives the optimum Gaussian blurring value for σ of each visual saliency map, the best weight value *w* for adding the center-map, new average *NCC* for each visual saliency model together with standard deviations and their percentages of improvement.

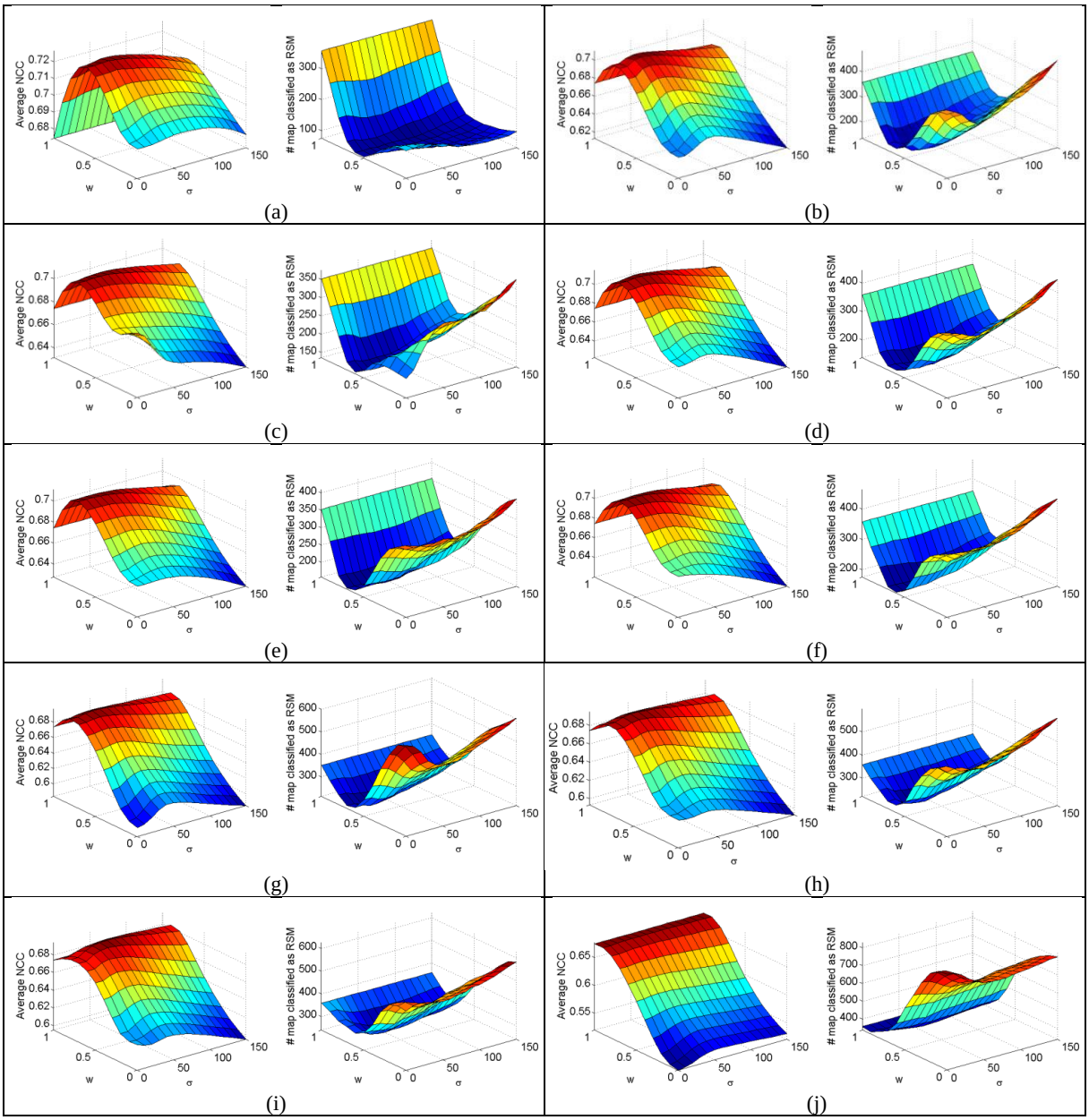


Figure 4-5. Effect of blurring and adding the center-map with different weights on the performance of the visual saliency models, GBVS (a); AIM (b); EH (c); CASD (d); CC (e); IS (f); IK (g); SUN (h); SR (i); and FTSRD (j).

In Table 4-5, degrees of improvement are computed with reference to Tables 4-3 and 4-4. As shown in Table 4-5, GBVS again outperforms the other visual saliency models after the blurring and center-biasing process, with only 75 maps classified as RSM, which shows GBVS is the best bottom-up visual saliency model independent of its level of blurriness and center bias. However, AIM, IK CASD and FTSRD are improved by this process the most. AIM is ranked second with 135 maps classified as RSM, with 65.4% improvement.

Table 4-5. Best blurring σ and center map weight *w* and its effect on the performance of the visual saliency models using *NCC*.

Visual saliency model	Best σ	Best *w*	Average *NCC*	Improvement Percentage	Min # maps classified as RSM	Improvement Percentage	New Ranking
GBVS	30	0.5	0.724	4.6%	75	54.3%	1
AIM	20	0.6	0.703	10.5%	135	65.4%	2
EH	20	0.6	0.704	2.3%	136	21.8%	3
CASD	40	0.7	0.714	9.0%	138	61.3%	4
CC	30	0.7	0.711	8.2%	158	55.1%	5
IS	10	0.6	0.707	11.1%	176	50.7%	6
IK	30	0.7	0.695	16.7%	214	65.3%	7
SUN	40	0.7	0.693	11.1%	222	55.5%	8
SR	50	0.7	0.692	10.9%	240	52.0%	9
FTSRD	60	0.8	0.669	28.6%	339	61.4%	10

Table 4-5 indicates that blurring and center-biasing improves visual saliency models. It shows that the CASD, SUN, SR and FTSRD visual saliency models can benefit greatly from blurring. The best *w* value is an indication of how much of the original saliency map is used to create new saliency maps. GBVS uses the smallest value of *w* (0.5), which is another indication that GBVS is a well-designed visual saliency model. On the other side, FTSRD uses the largest value of *w* (0.8). This indicates FTSRD has used 80% of the center map and only 20% of the blurred original saliency map.

4.1.2 Comparisons after Histogram Matching

In this section, all comparison metrics are used to evaluate visual saliency models. PSMs are compared with RHSMs after matching the histogram of the PSM produced for one image with the RHSM of that image. Saliency maps for which their metric values are smaller than the thresholds shown in Table 3-1 are considered to be more similar to an RSM than the corresponding RHSM. Similarly, the visual saliency models that create higher average metric values and minimum maps classified as RSM are ranked higher. Box-plots of the *Score* (the highest ranked saliency map comparison metric for the case with histogram matching) results are shown in Figure 4-6. Similar to Section 4.1.1, the Shapiro-Wilk method is used to check the

normality of the data. *Score* results of none of the models are normally distributed with a significance level of 0.05. The Wilcoxon test is used again and results are given in Table 4-6.

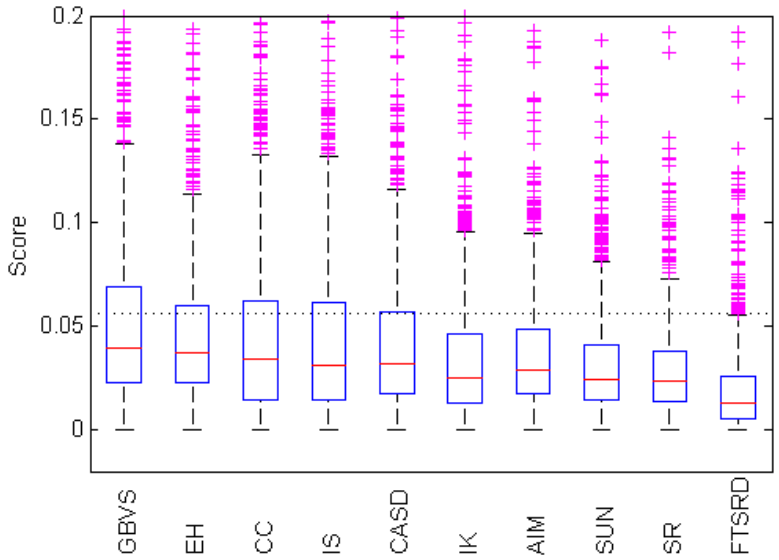


Figure 4-6. *Score* Box-plots for all visual saliency models.

Table 4-6. Wilcoxon test results on *Score*s of all visual saliency models.

Saliency Model	GBVS	EH	CC	IS	CASD	IK	AIM	SUN	SR	FTSRD
GBVS	---	0	1	1	1	1	1	1	1	1
EH	0	---	1	1	1	1	1	1	1	1
CC	1	1	---	0	0	1	1	1	1	1
IS	1	1	0	--	0	1	0	1	1	1
CASD	1	1	0	0	---	1	1	1	1	1
IK	1	1	1	1	1	--	1	0	0	1
AIM	1	1	1	0	1	1	---	1	1	1
SUN	1	1	1	1	1	0	1	---	0	1
SR	1	1	1	1	1	0	1	0	---	1
FTSRD	1	1	1	1	1	1	1	1	1	---

Figure 4-6 shows GBVS outperforms other models and EH is ranked 2nd similar to the results in Section 4.1.1; however, IK is ranked 6th here. All medians are under the threshold line, which indicates the majority of the PSMs created by each model are classified as RSMs. The Wilcoxon test found no significant difference between GBVS and EH, which are statistically different from the rest of the models as demonstrated in Table 4-6. CC, IS and CASD differences are not significant, as well as the difference between IK, SUN and SR. Saliency comparison metric

averages and the ranking base on them in parentheses (the lower the rank number, the better) for all visual saliency models are shown in Table 4-7. The number of maps classified as RSM based on each metric and the ranking based on the number of RSM maps in parentheses are also shown in Table 4-8. The right column of Table 4-8 shows the average number of maps classified as RSM based on all saliency comparison metrics.

Table 4-7. Average of each metric for all visual saliency models and their rankings based on the average of each metric.

		Comparison metrics in increasing rank number (from left to right)								
		Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***
Visual saliency mechanisms in increasing rank number (top to bottom) based on *Score*	**GBVS**	0.0538 (1)	0.441 (1)	0.648 (1)	0.369 (1)	0.709 (1)	0.543 (2)	0.911 (1)	0.273 (4)	0.995 (3)
	EH	0.0482 (2)	0.414 (2)	0.629 (3)	0.336 (3)	0.685 (2)	0.512 (3)	0.898 (2)	0.267 (6)	0.995 (4)
	CC	0.0480 (3)	0.408 (3)	0.631 (2)	0.339 (2)	0.668 (3)	0.552 (1)	0.851 (5)	0.297 (1)	0.996 (1)
	IS	0.0474 (4)	0.401 (4)	0.624 (4)	0.327 (4)	0.66 (4)	0.451 (6)	0.872 (4)	0.227 (8)	0.995 (5)
	CASD	0.0448 (5)	0.391 (5)	0.615 (5)	0.311 (5)	0.66 (4)	0.427 (7)	0.839 (8)	0.19 (9)	0.996 (2)
	SUN	0.0386 (6)	0.344 (6)	0.587 (6)	0.26 (6)	0.612 (6)	0.47 (4)	0.843 (7)	0.288 (2)	0.993 (8)
	AIM	0.0359 (7)	0.333 (7)	0.58 (7)	0.247 (7)	0.601 (7)	0.353 (9)	0.8 (10)	0.163 (10)	0.995 (6)
	SR	0.0334 (8)	0.325 (8)	0.574 (8)	0.236 (8)	0.595 (8)	0.465 (5)	0.846 (6)	0.28 (3)	0.993 (7)
	IK	0.0307 (9)	0.301 (9)	0.566 (9)	0.224 (9)	0.584 (9)	0.425 (8)	0.874 (3)	0.271 (5)	0.97 (10)
	FTSRD	0.020 (10)	0.225 (10)	0.528 (10)	0.154 (10)	0.525 (10)	0.339 (10)	0.801 (9)	0.251 (7)	0.987 (9)

Table 4-8. Number of maps classified as RSM for all visual saliency models and their rankings based on their number of RSMs (for threshold values see Table 3-1).

		Comparison metrics in increasing rank number (from left to right)									
		Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***	**Average**
Visual saliency mechanisms in increasing rank number (top to bottom) based on *Score*	**GBVS**	661 (1)	612 (1)	450 (1)	427 (1)	145 (1)	414 (2)	506 (1)	445 (2)	15 (4)	408.3 (1)
	EH	700 (2)	693 (4)	484 (2)	446 (2)	153 (2)	348 (1)	656 (3)	468 (3)	10 (3)	439.8 (2)
	CC	701 (3)	649 (2)	510 (3)	488 (3)	293 (4)	432 (3)	703 (5)	438 (1)	2 (1)	468.4 (3)
	IS	718 (4)	685 (3)	548 (4)	538 (4)	310 (5)	531 (5)	628 (2)	578 (7)	20 (6)	506.2 (4)
	CASD	742 (5)	731 (5)	566 (5)	545 (5)	292 (3)	522 (4)	746 (7)	712 (9)	7 (2)	540.3 (5)
	IK	802 (6)	856 (6)	690 (6)	666 (6)	501 (7)	601 (8)	805 (9)	729 (10)	15 (4)	629.4 (8)
	AIM	814 (7)	922 (7)	690 (6)	667 (7)	464 (6)	535 (6)	718 (6)	485 (4)	48 (7)	593.7 (6)
	SUN	862 (8)	927 (8)	744 (8)	717 (8)	537 (8)	546 (7)	746 (8)	508 (5)	69 (8)	628.4 (7)
	SR	891 (9)	957 (9)	795 (9)	751 (9)	616 (9)	604 (9)	674 (4)	510 (6)	591 (10)	709.9 (9)
	FTSRD	927 (10)	981 (10)	888 (10)	877 (10)	818 (10)	672 (10)	814 (10)	584 (8)	266 (9)	758.6 (10)

Tables 4-7 and 4-8 show that the GBVS model outperforms other visual saliency models based on *Score* (the highest ranked saliency map comparison metric for the case with histogram matching) and also based on the average maps classified as RSM over all metrics. Similar to the results of the study without histogram matching, our visual saliency model (EH) is ranked 2nd and FTSRD is ranked lowest in these tables. As a further note, the rankings based on *Score* are very similar to the overall ranking based on the average RSMs for all metrics. Also, after histogram matching saliency map comparison metrics tend to agree more on ranking visual saliency models comparing to the case without histogram matching shown in Tables 4-3 and 4-4.

4.1.2.1 Optimizing Blurriness and Center-Bias

Exploring the histogram matching, the optimized level of blurriness and center-bias of each visual saliency model by varying the variance σ of the Gaussian filter and the weight w in (4-1) is studied in this Section. As an example of this study, Figure 4-7 shows a sample image (the same as in Figure 4-4), its RHSM, PSMs produced by the SR visual saliency model, blurred and center-biased saliency maps, and their *Score* results. The same Center-Map (***CM***) as in section 4.1.1.1 is used; however, in this section, the histograms of the center-map and the PSMs are matched to the histograms of the RHSMs. The histograms of the resulting saliency maps from (4-1) are also matched to the histograms of the corresponding RHSM before comparison.

Note in Figure 4-7 that as was the case for no histogram matching, SR benefits from a certain degree of blurriness and center bias, with the highest value of $Score = 0.0991$ for $\sigma = 20$ and $w = 0.2$. After modifying PSMs using (4-1), they are compared to the corresponding RHSM using *Score*. Figure 4-8 demonstrates how the average *Score* and number of maps classified as RSM for each visual saliency mechanism is affected by changing the blurring σ value and weight w of the center-map.

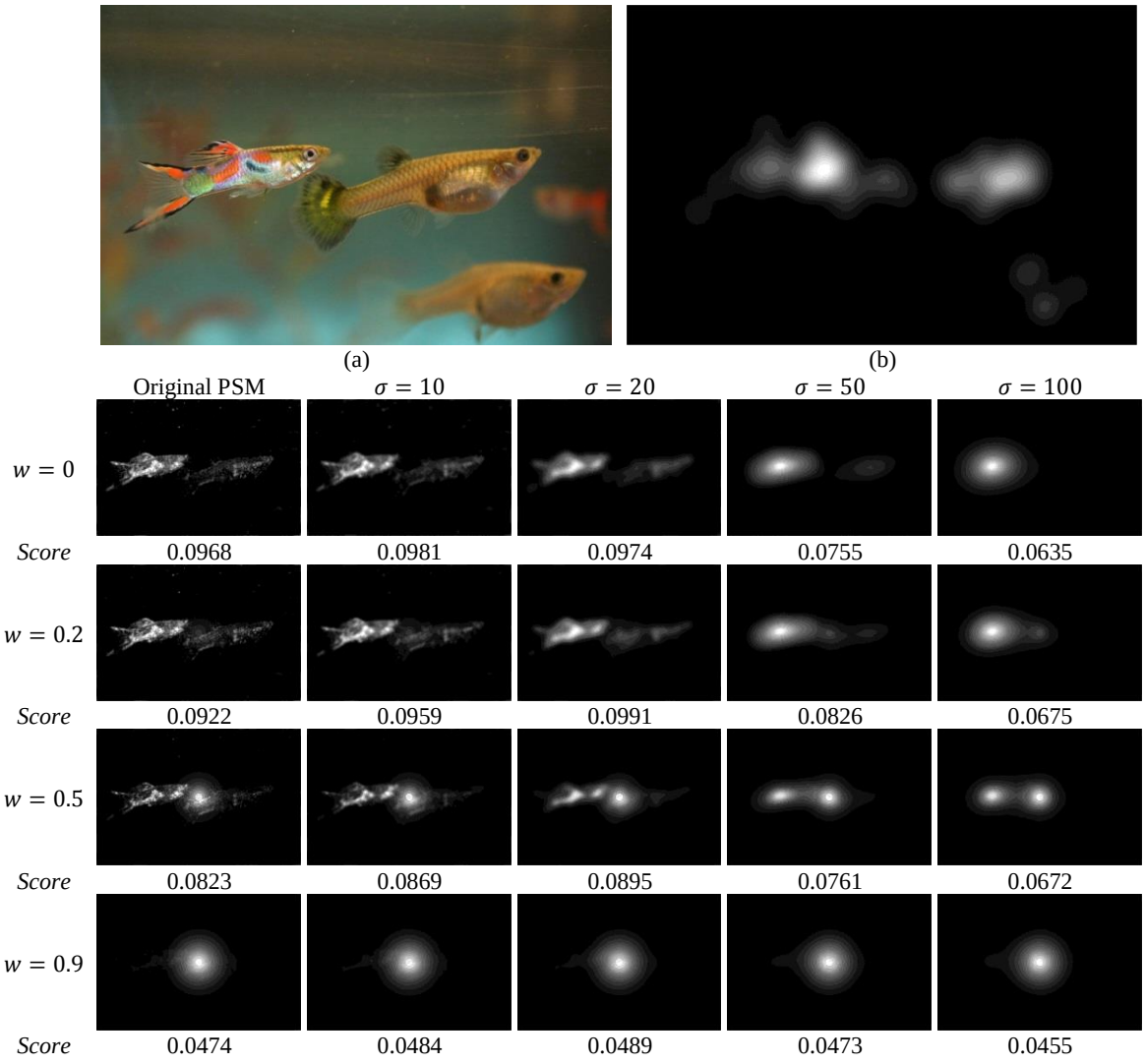


Figure 4-7. Original image (a); its RHSM (b); blurred and center-biased maps (all after histogram matching) and their *Score* values using SR.

The optimum Gaussian blurring level of each visual saliency map, the best weight values for adding the center-map and visual saliency models new average *Score*, standard deviation and their percentage of improvement are shown in Table 4-9.

As demonstrated in Table-4-9 and Figure 4-8, all visual saliency models can improve their performance with blurring and adding a center biased map. Table 4-5 shows that GBVS

performed better than other models. However, all visual saliency models produce 614 to 667 minimum number of maps classified as RSM which are 61.2% to 66.5% of the images in the database. This indicates that visual saliency models under histogram matching produce relatively

poor results when are evaluated by *Score*. This also indicates that there is clearly an opportunity for an improved visual saliency model. GBVS used the minimum of the center map (50%), and FTSRD and SR used the maximum (100%). This means that the center map outperforms FTSRD and SR in mimicking human observers.

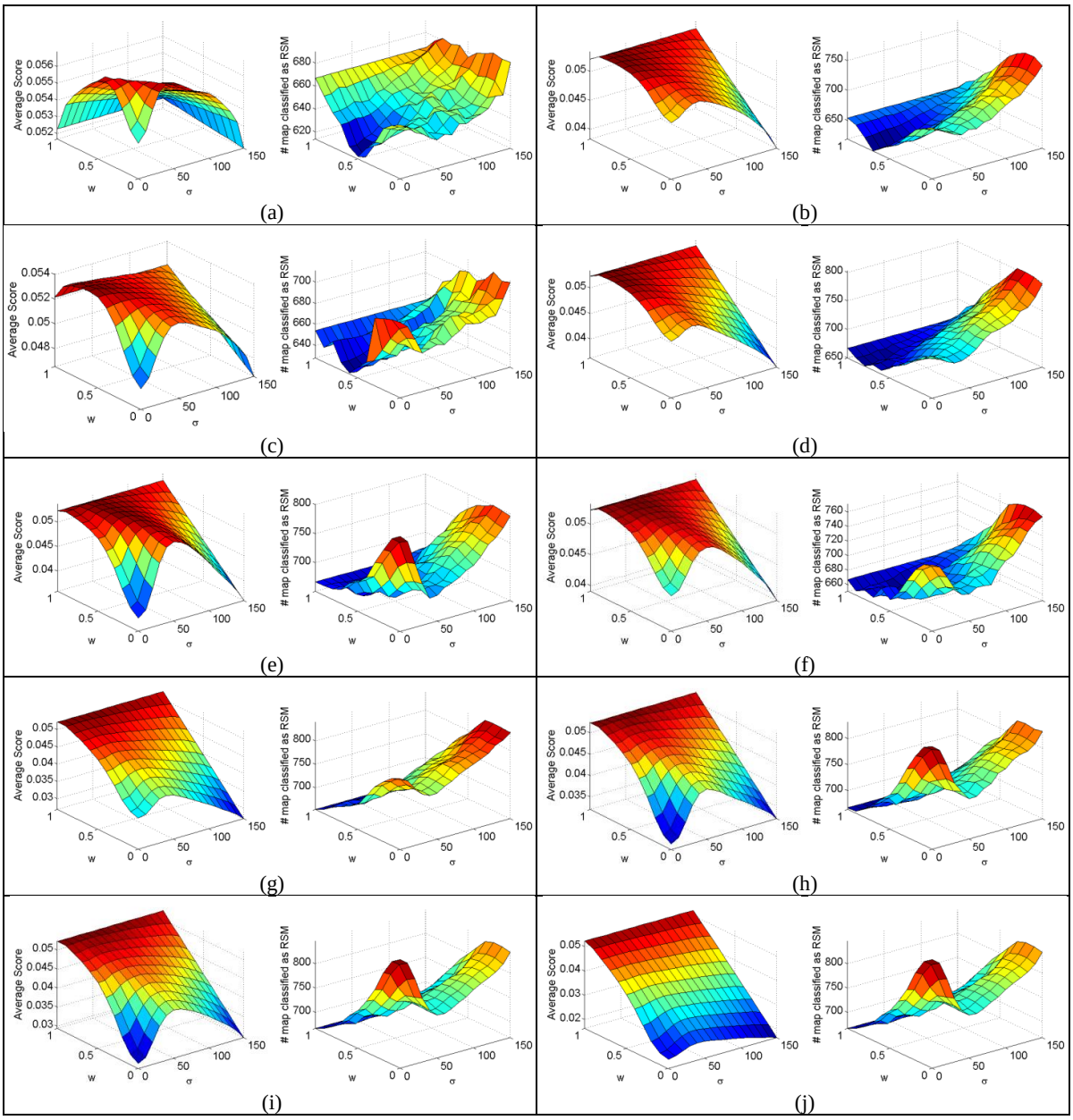


Figure 4-8. Effect of blurring and adding the center-map with different weights on the performance of the visual saliency models, GBVS (a); CC (b); EH (c); IS (d); AIM (e); CASD (f); IK (g); SUN (h); SR (i); and FTSRD (j).

Table 4-9. Best blurring σ and center map weight *w* and its effect on the performance of the visual saliency models using *Score*.

Visual saliency model	Best σ	Best *w*	Average *Score*	Improvement Percentage	Min # maps classified as RSM	Improvement Percentage	New Ranking
GBVS	10	0.5	0.0564	4.9%	614	7.1%	1
CC	10	0.7	0.0534	11.3%	619	11.7%	2
EH	10	0.6	0.0541	12.2%	628	10.3%	3
IS	10	0.8	0.0534	12.6%	650	9.5%	4
AIM	40	0.7	0.0537	39.1%	651	20.0%	5
CASD	40	0.7	0.0532	18.6%	651	12.3%	5
IK	10	0.9	0.0521	45.3%	654	18.5%	7
SUN	50	0.9	0.0524	57.1%	664	23.0%	8
FTSRD	0	1	0.0523	159.2%	667	28.0%	9
SR	0	1	0.0523	70.6%	667	25.1%	9

4.2 Visual Saliency Mechanisms on Synthetic Images

In this section we test visual saliency mechanisms on a benchmark of 54 synthetic images used in [65], shown in Figure 4-9. CC [17] is not included here because we did not possess the computer codes for CC. Since there are no reference saliency maps available for this dataset, reference fixation maps were generated by manually selecting 1 to 5 pixels at the central parts of the salient regions, depending on the number of salient regions in the image. Then, reference saliency maps were created by convolving the fixation maps by a 2D Gaussian filter with proper σ_s that highlight the salient regions, with results depicted in Figure 4-10. In Figure 4-9, images are sorted based on their average *Score* values from all saliency models from high at the top-left to low at the bottom right. All visual saliency models could find the salient regions in Figure 4-9 (aa) and (ab) and none of the models could detect the salient parts of (ca) and (cb).

To evaluate and rank saliency model performances on the synthetic images, first the histograms of the PSMs for each image were matched to the to the corresponding reference saliency map. Then, PSMs are compared to reference maps using *Score*, and results are depicted in Figure 4-11 and Table 4-11.

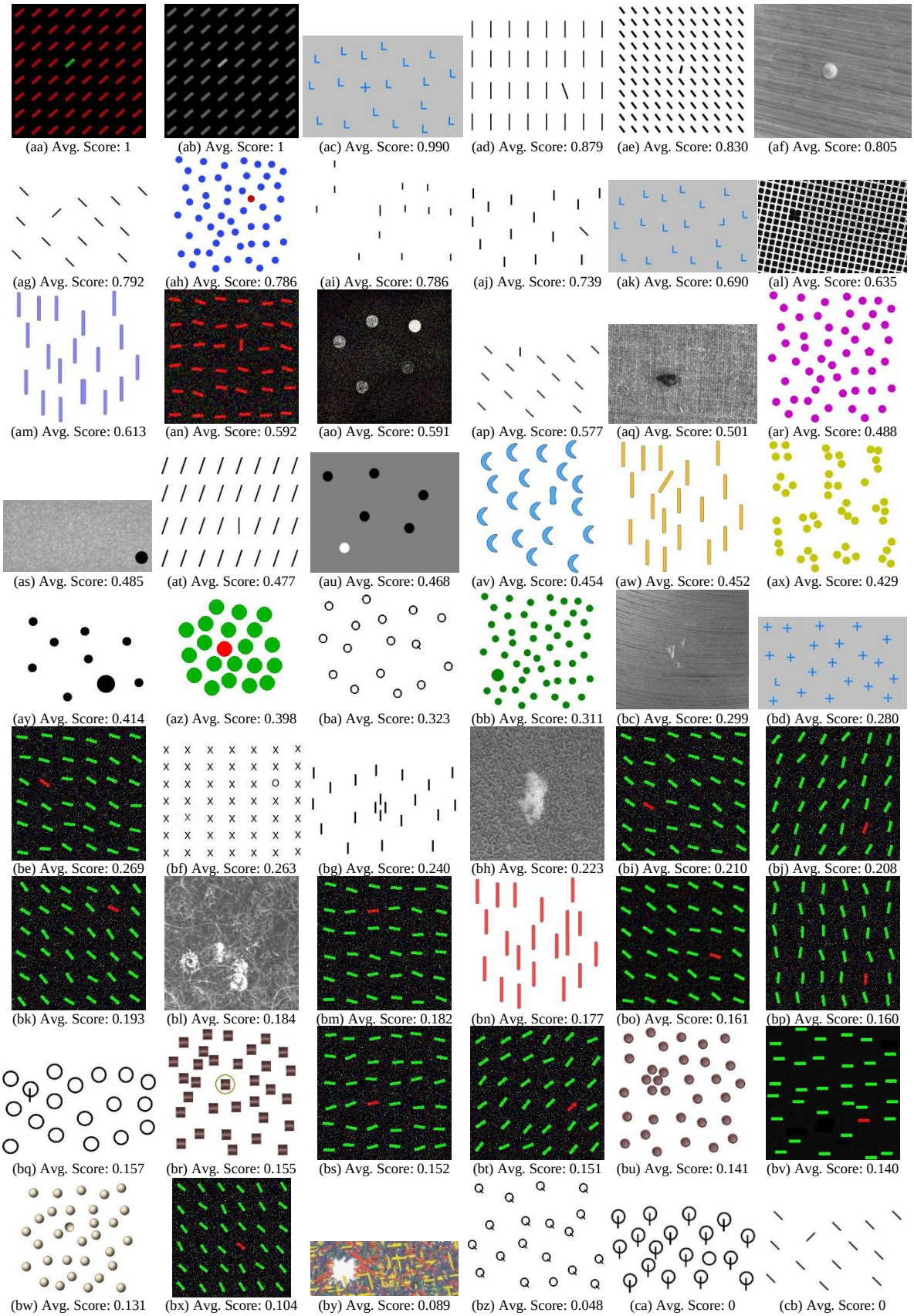


Figure 4-9. Synthetic images database and their average *Score* values.

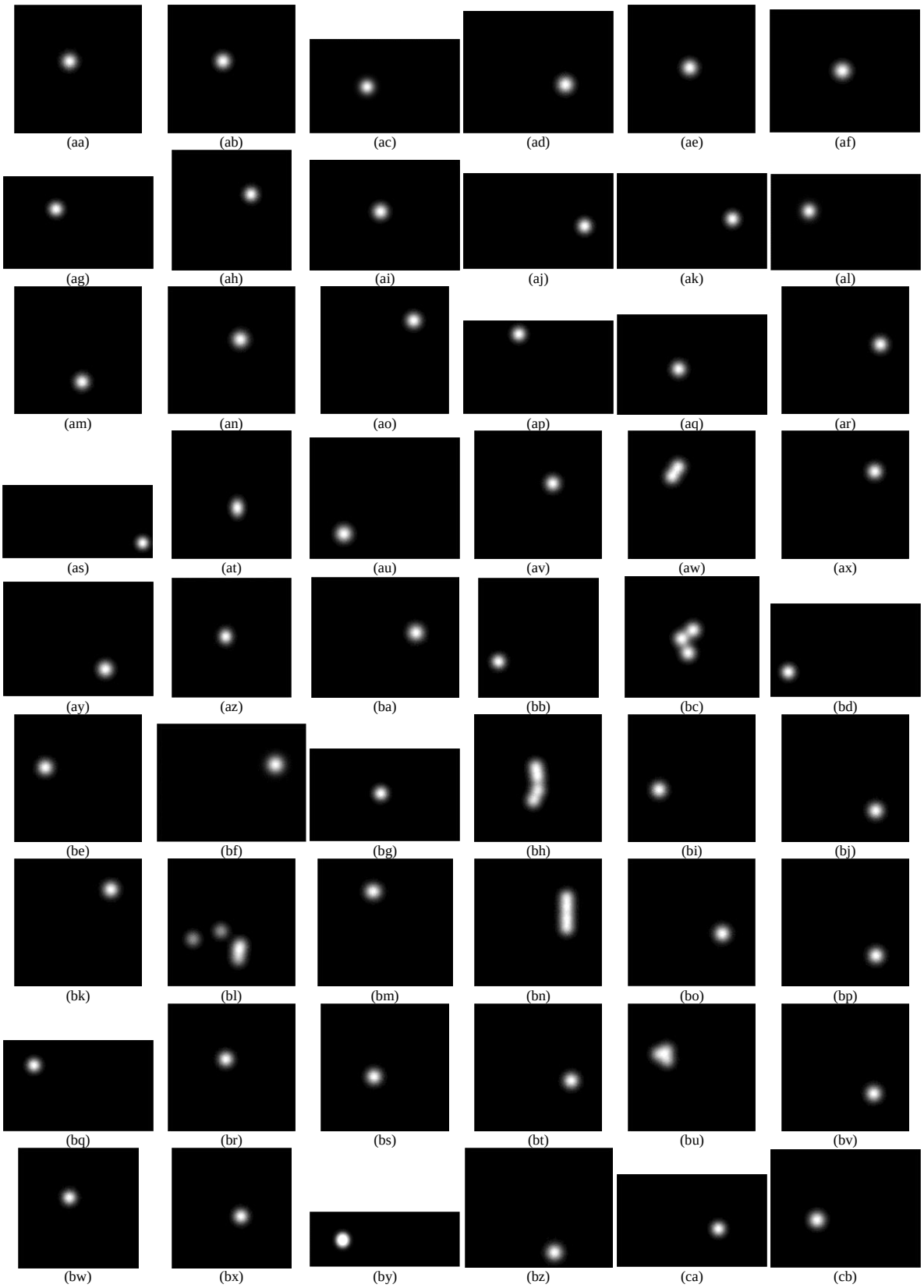


Figure 4-10. Reference saliency maps created manually for synthetic images database.

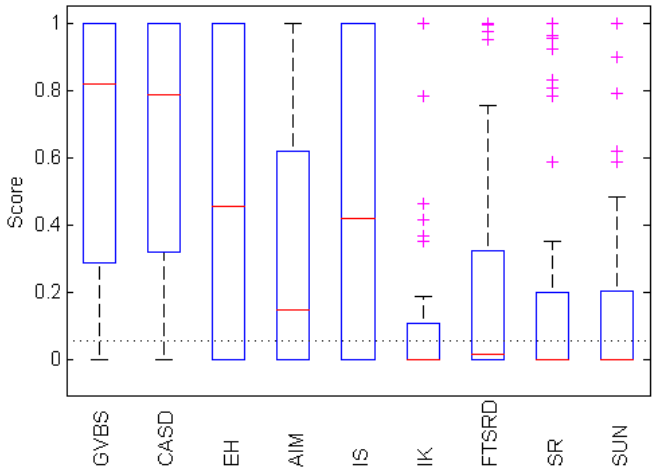


Figure 4-11. *Score* box-plots for all visual saliency models on the synthetic image dataset.

Table 4-10. Wilcoxon test results on the *Scores* of all visual saliency models on the synthetic image dataset.

Saliency Model	GBVS	CASD	EH	AIM	IS	IK	FTSRD	SR	SUN
GBVS	---	0	1	1	0	1	1	1	1
CASD	0	---	1	1	0	1	1	1	1
EH	1	1	---	0	0	1	1	1	1
AIM	1	1	0	---	1	1	0	1	1
IS	0	0	0	1	---	1	1	1	1
IK	1	1	1	1	1	--	1	0	0
FTSRD	1	1	1	0	1	1	---	1	0
SR	1	1	1	1	1	0	1	---	0
SUN	1	1	1	1	1	0	0	0	---

As shown in Table 4-11, EH and IS standard deviations of the data is so large that 25th and 75th percentiles in Figure 4-11meet the lower and the upper limits of the data (0 and 1, respectively). Table 4-10 demonstrates that GBVS, CASD and IS are not found statistically different by the Wilcoxon test; however, we believe that the differences between IS and the first two ranked models are significant. This is a drawback of non-parametric statistical models like Wilcoxon, that when the number of data points is small, their results are not as reliable as when there are many data points. Also, no significant difference between EH results and AIM and IS were found, as well as between SUN, SR, FTSRD and IK.

Table 4-11. The mean of the *Score* values computed for all visual saliency models and the number of maps classified as RSM (threshold used: 0.0553, see Table 3-1) on 54 synthetic images.

Visual saliency model	Mean of *Score*	Standard Deviation of *Score*	Number (%) of maps classified as RSM	Ranking based on the number of maps classified as RSM
GBVS	0.712	0.364	6 (11.1%)	1
CASD	0.739	0.398	10 (18.5%)	2
EH	0.611	0.454	14 (25.9%)	3
AIM	0.408	0.429	22 (40.7%)	4
IS	0.517	0.481	23 (42.6%)	5
IK	0.176	0.333	24 (44.4%)	6
FTSRD	0.268	0.375	33 (61.1%)	7
SR	0.217	0.363	34 (62.9%)	8
SUN	0.232	0.384	36 (66.7%)	9

As shown in Figure 4-11 and Table 4-11, GBVS with 6 maps classified as RSM (11.1%) produces the best PSMs for the synthetic image database. CASD and EH are 2nd and 3rd, respectively, with the RSM percentage of 18.5 and 25.9. Models ranked 4th to 6th are very close in RSM percentages (around 42%). The saliency models FTSRD, SR and SUN, with *Score* means of 0.268, 0.217 and 0.232, have 61.1%, 62.9%, and 66.7% of maps classified as RSM, which rank them lowest. Accordingly, we select GBVS as the best bottom-up visual saliency model on synthetic images. Comparison of Table 4-11 to Tables 4-5 and 4-6 shows bottom-up visual saliency models performed much better on synthetic images. This is because, synthetic images contain solely bottom-up features and information, e.g. change in color or orientation. Also, reference saliency maps created for synthetic images highlight only salient regions of the image and include all such regions. In the RHSMs created in the LPWHL database [40] sometimes salient regions close to the boundary of the image are missed, and non-important parts of the images close to the center are highlighted in the saliency maps.

4.2.1 Optimizing Blurriness

As suggested in [61] and [65] for fair model comparison, PSMs were convolved with 2D Gaussian filters with $\sigma = 5, 10, 15, 20 \text{ and } 25$ pixels and then were compared with the corresponding reference saliency maps using *Score* after histogram matching. Since, most of the

saliency maps highlight parts other than the center, analyzing center bias would not create useful results. Accordingly, only the level of blurriness is studied here. Figure 4-13 (a) and (b) depict a synthetic image and its reference saliency maps, as well as original PSMs created by all visual saliency models, matched histogram saliency maps and blurred matched histogram saliency maps with σ=15.

The effect of blurred saliency maps on the number of maps classified as RSM for each visual saliency model is demonstrated in Figure 4-12. The optimum Gaussian blurring level for each visual saliency model, their new average *Score*, number of maps classified as RSM and their percentages of improvement are shown in Table 4-12.

Table 4-12 and Figure 4-12 show that blurring improves the performance of most of the visual saliency models. After blurring, with 9.25% improvement compared to Table 4-11, CASD reaches a mean of *Score* of 0.846 with only 9.25% of maps classified as RSM. However, we see that the performances of GBVS and IS visual saliency mechanisms deteriorates with blurring.

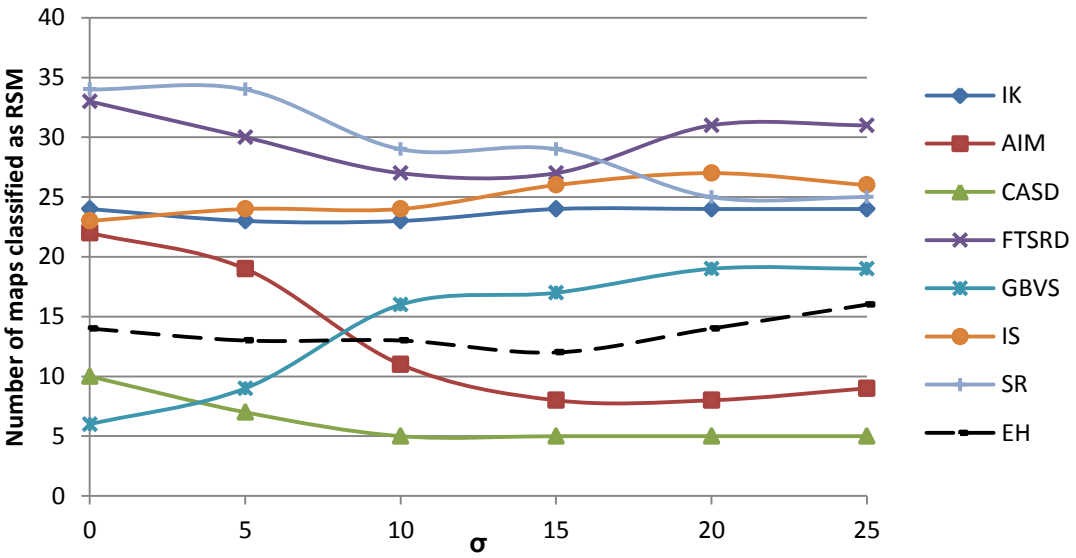


Figure 4-12. Effect of blurring on number of maps classified as RSM for each visual saliency model.

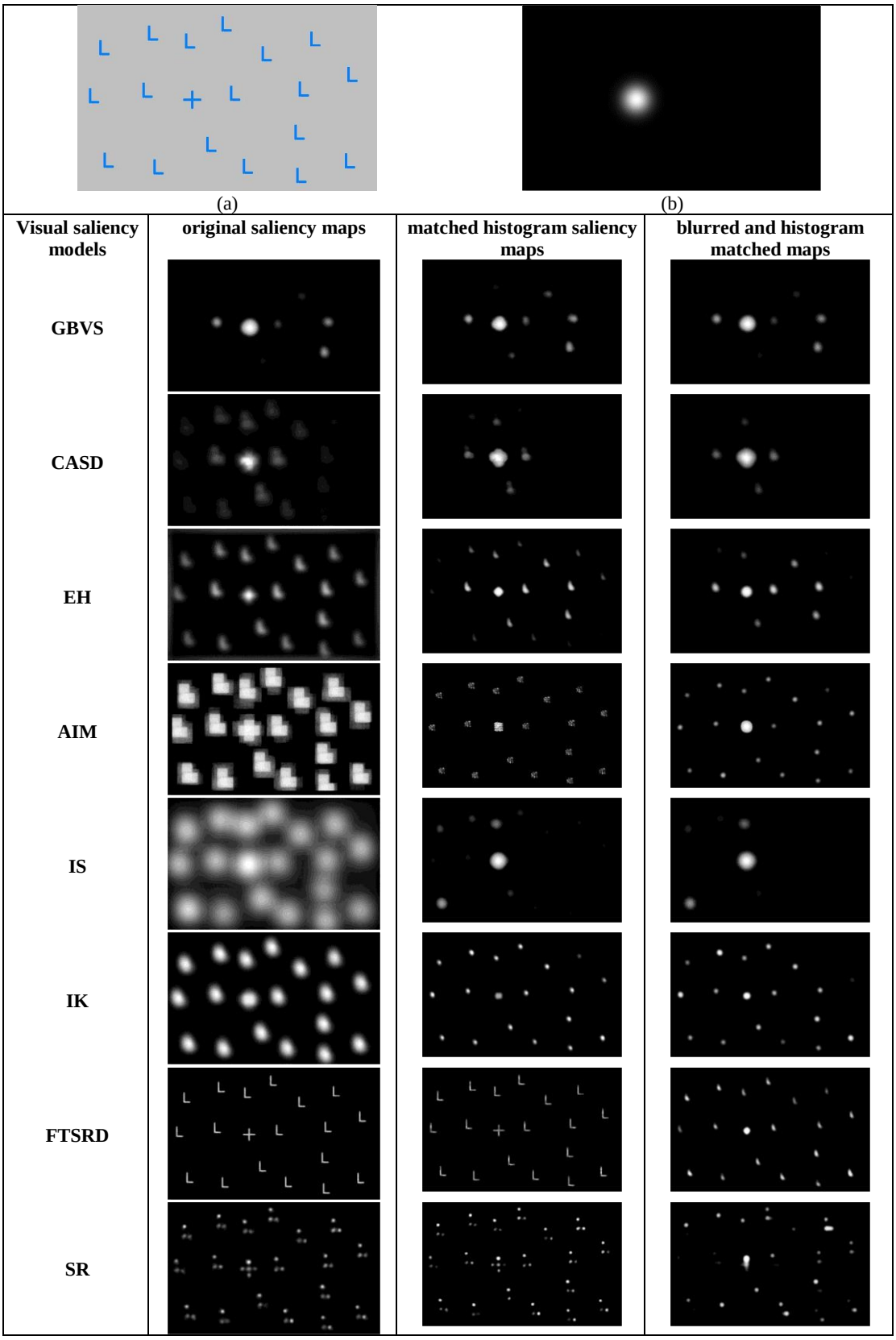


Figure 4-13. Original synthetic image (a); its reference saliency map (b); original PSMs, matched histogram maps and blurred histogram matched maps with blurring σ=15 pixels.

Table 4-12. Best blurring σ and its effect on the performance of the visual saliency models on the synthetic image dataset.

Visual saliency model	Best Blurring σ (pixels)	Mean of Score	Number (%) of maps classified as RSM	Improvement Percentage	Ranking based on the number of maps classified as RSM
CASD	25	0.846	5 (9.25%)	9.25%	1
GBVS	0	0.712	6 (11.1%)	0	2
AIM	25	0.759	11 (20.1%)	20.14%	3
EH	15	0.619	12 (22.2%)	3.7%	4
IS	0	0.517	23 (42.6%)	0	5
IK	10	0.167	23 (42.6%)	20.14%	6
FTSRD	25	0.328	27 (50%)	14.8%	7
SR	25	0.444	29 (53.7%)	9.25%	8
SUN	25	0.311	29 (53.7%)	13%	9

4.3 Discussion

We believe that GBVS, the highest ranked visual saliency model for all but the synthetic images, mimics human observer behavior better than other bottom-up saliency mechanisms discussed herein for two main reasons. First, in an RHSM, the central parts of salient objects are highlighted, not their boundaries. Visual saliency algorithms based on the center-surround mechanism and self-information have difficulty activating salient regions far from the salient object boundaries in the feature saliency map. On the other hand, the GBVS algorithm highlights salient regions that are distant from the object boundaries. Accordingly, saliency maps computed by GBVS algorithm are more similar to human saliency maps than saliency maps from other algorithms. Second, saliency maps generated in the LPWHL study [40] highlight the central areas of the images as salient regions more often than non-central areas. Figure 4-14 shows that on average, saliency maps in this dataset are center-biased. It is worth mentioning that the saliency map datasets from Bruce and Tsotsos [28] and ORIG [66] also are highly center-biased [50]. Due to the Gaussian weight function in (1-1), the GBVS model tends to highlight salient objects in the central regions of an image more than objects close to the image boundary. This means that GBVS saliency maps tend to be center-biased as well. However, results in sections 4.1.1.1 and

4.1.2.1 showed that GBVS outperforms other bottom-up visual saliency models in spite of this feature.

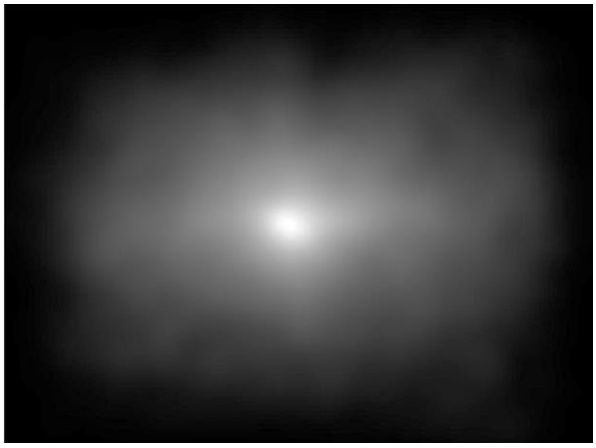


Figure 4-14. Average of all RHSMs generated in the LPWHL study [40].

Since the IK [1] and EH saliency models differ only in the normalization methods employed to normalize feature saliency maps, Table 4-3 and Table 4-7 show that our normalization method in the EH performs much better than the Itti and Koch [1] normalization method. This also shows the importance of normalizing feature saliency maps before combining them to produce a calculated saliency map.

Tables 4-1 and 4-5 show that for saliency maps generated by GBVS (the best model), 164 (16.35%) and 661 (65.9%), of the PSMs (calculated using *NCC* and *Score*, respectively) are more similar to RSMs than RHSMs. This fact demonstrates that there is significant room for improvement in bottom-up visual saliency models. Figure 4-15 (a) presents an image from [40] for which all PSMs (c) to (k) have *NCC* and *Score* values smaller than the thresholds. Therefore, they are closer to an RSM than the RHSM of the image shown in Figure 4-15 (b). This occurs because the bottom of the image shows a written phrase that attracts a human observer's attention. Bottom-up visual saliency models find no difference between texts and textures. The reason that the text is highlighted in some saliency maps is that it has high contrast to its surroundings. This shows the importance of incorporating high level top-down visual saliency attributes in an over-all visual saliency model. However, results on the synthetic images showed

that the high-rated bottom-up visual saliency models perform satisfactorily on synthetic images that contain bottom-up salient features.

One of the main drawbacks of all visual saliency models is the use of predetermined parameters in the algorithms. For example center-surround mechanisms are applied with predetermined sets of radii, when in fact the radii should be determined based on the properties of the display, such as: size of the objects in the display, distances between objects in the display, and textures of the objects. Another problem is that only one set of parameters is employed to analyze an entire image. We propose that to effectively imitate the HVS saliency mechanism, different parts of the visual scene should be analyzed with different sets of parameters (analyzing the image locally). In the next chapter, the GBVS algorithm will be modified to include a parameter selection step based on local image areas and this modification will be compared with the original GBVS.

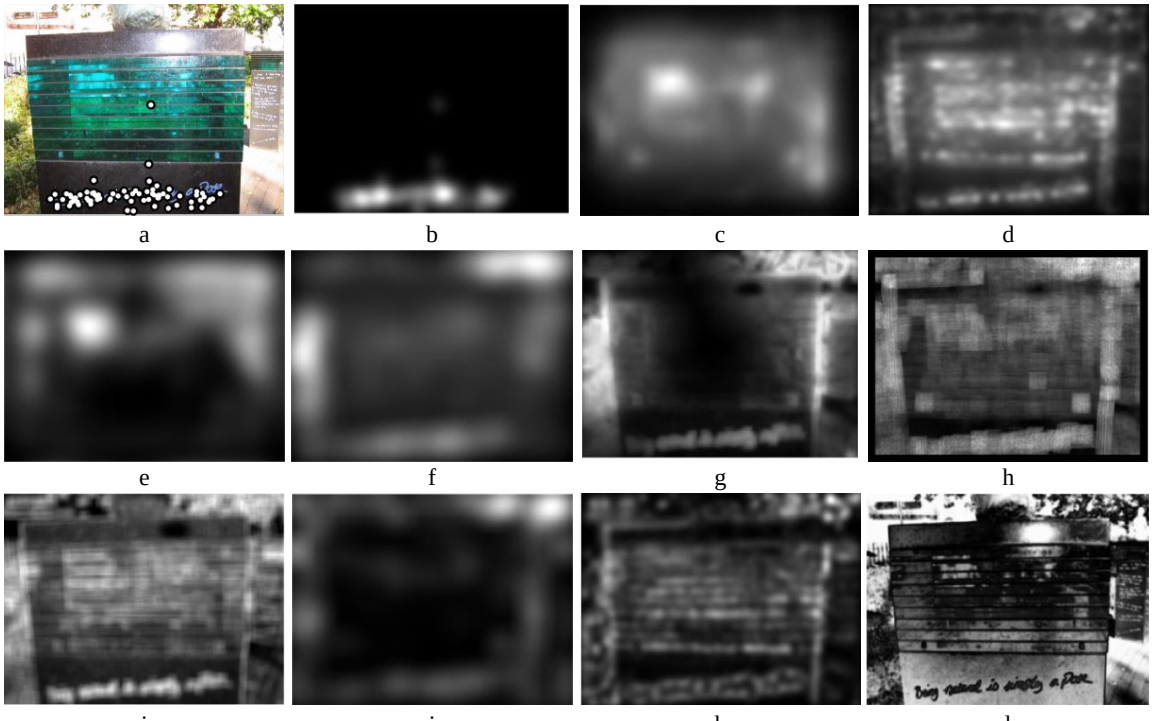


Figure 4-15 Original image with fixation points [40] (a); RHSM (b); and PSMs using GBVS (c), EH (d), CC (e), IS (f), CASD (g), AIM (h), SUN (i), SR (j), IK (k) and FTSRD (l).

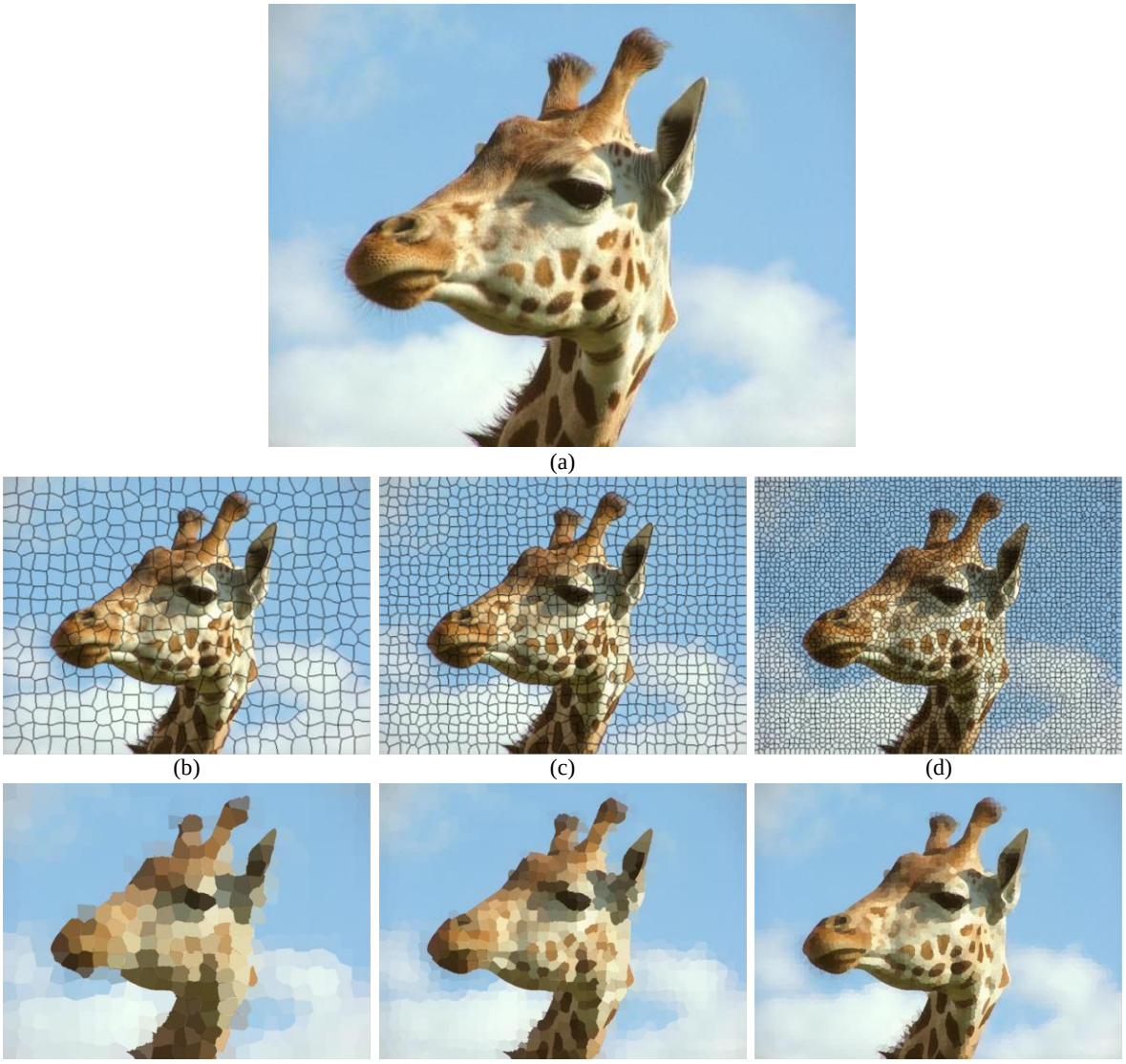
CHAPTER V

5 A New Visual Saliency Model (EMHO)

As we discussed in Chapter IV, using predetermined parameters is one of the most important drawbacks of the visual saliency models. Also, in all models tested in the previous chapter, only one set of parameters is used to analyze the entire image. We believe that to effectively imitate the HVS saliency mechanism, different parts of the visual scene should be analyzed with different sets of parameters (analyzing the image locally). In this chapter, we develop a new algorithm based on the GBVS saliency model that selects different parameters for different parts of the image.

5.1 Using Superpixel Images Instead of Original Images

Since selecting different parameters for every pixel in the image is cumbersome, we propose working on a reduced version of the image, namely the superpixel image. By grouping similar pixels in a neighborhood and capturing redundancies in images [67], superpixel images greatly reduce the number of nodes for which parameter selection should be performed. Figure 5-1 shows an image from the LPWHL database containing $1024 \times 771 = 789504$ pixels (a), superpixel boundaries with 400 (b), 1000 (c) and 4000 (d) superpixels and the corresponding superpixel images in (e), (f) and (g) created using the TurboPixel algorithm [68]. This algorithm has been selected herein because it is one of the fastest and most accurate superpixel algorithms in the literature [67].



(a)

(b) (c) (d)

(e) (f) (g)

Figure 5-1 Original image with 789504 pixels (a); superpixel boundaries with 400 (b), 1000 (c) and 4000 (d) superpixels; and the corresponding superpixel images in (e) (f) and (g).

The superpixel images in Figure 5-1 (e) to (g) are created by averaging pixel values of all pixels that belong to a superpixel. The average of color values of neighbor superpixels are assigned to the pixels on the boundaries (black lines in (b), (c) and (d)). As shown in Figure 5-1, when the number of superpixels is small many details are lost (e); however, when the number of superpixels is high, it is difficult to distinguish between the original image and the superpixel image (g).

5.2 The EMHO Algorithm

We propose using a modified version of the GBVS algorithm on the superpixel images instead of the original images, which we call the EMHO algorithm. Using superpixel images will decrease the saliency computational time of EMHO and will also render the parameter selection step achievable. Similar to work by Itti et al. [6], we use one intensity, two color, and four orientation feature spaces here. In the EMHO algorithm, the first step is to create a superpixel version of a feature space ($\boldsymbol{F}$), a vector $\boldsymbol{f}_{SP}$ with length N_{SP} (number of superpixels). $\boldsymbol{f}_{SP}$ contains the superpixel by superpixel average of $\boldsymbol{F}$, whose l^{th} value is given by:

$$\boldsymbol{f}_{SP}(l) = \sum_{k=1}^{N_l} \boldsymbol{F}(i,j)_{l_k} \Big/ N_l \tag{5-1}$$

where N_l is the number of pixels in the l^{th} superpixel and $(i,j)_{l_k}$ is the pixel location of the k^{th} pixel in the l^{th} superpixel. Similar to the GBVS algorithm, to calculate the feature saliency map ($\boldsymbol{M}$) corresponding to a feature space ($\boldsymbol{F}$), first, a fully connected graph $\boldsymbol{G}$ is generated by connecting every two superpixels in $\boldsymbol{f}_{SP}$. Then, modifying the original GBVS formulation [11], a weight $\boldsymbol{W}$ is assigned to every edge. The weight value from superpixel (l) to (m) is defined by:

$$\boldsymbol{W}(l,m) = \left| log \frac{\boldsymbol{f}_{SP}(l)}{\boldsymbol{f}_{SP}(m)} \right| \times \left[\frac{1}{\boldsymbol{\sigma}^2(l)} exp\left(- \frac{(i_l - i_m)^2 + (j_l - j_m)^2}{2\boldsymbol{\sigma}^2(l)} \right) \right] \tag{5-2}$$

where (i_l, j_l) is the pixel location of the center of the l^{th} superpixel and $\boldsymbol{\sigma}(l)$ is the standard deviation of the Gaussian weight function corresponding to the l^{th} superpixel. The first term on the right computes the dissimilarity between two superpixels. The second term is a Gaussian function to increase the weight of two close superpixels and decrease the weight of superpixels which are far from each other. Weight matrix ($\boldsymbol{W}$) created using (5-2) is not symmetric. Instead of the Markov chain solution of the fully connected graph $\boldsymbol{G}$ (in the GBVS, see Section 1.1.1.5), a saliency map of the size of the image is created in which all pixels grouped in the l^{th} superpixel gain the value given by:

$$\boldsymbol{s}(l) = \sum_{m=1}^{N_{SP}} \boldsymbol{W}(l,m) \tag{5-3}$$

where $\boldsymbol{s}$ is called the saliency vector, which carries the saliency information of all superpixels. The Markov chain solution in GBVS (the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{W}$) is replaced with a simple summation. That is because the Markov chain solution does not provide any useful extra information, and all the saliency information is already extracted using (5-2). Saliency maps created using (5-3) are very similar to the maps created using the Markov chain solution, which on average provides no superiority to the summation solution. However, (5-3) is much faster and less computationally expensive than eigenvector computation.

In superpixel images, pixels on the boundary of superpixels are not assigned to any superpixels. Accordingly, they do not gain any saliency value in the introduced saliency algorithm. To estimate a saliency value for the boundaries, we use the average saliency of the neighbor superpixels. To smooth saliency maps, they are convolved with a Gaussian filter with $\sigma_{Blurring} = 20$, which was found to create best results, on average. Figure 5-2 illustrates this process for superpixel images created for an image in the database with 400 and 4000 superpixels. In this Section, the standard deviation of the Gaussian weight function for all superpixels was selected to be 375 pixels, which is the best single value that produces the maximum average of *Score* when PSMs are compared to the RHSMs from the LPWHL database.

5.2.1 Selecting the Number of Superpixels in EMHO

Since pixels are grouped together and the average pixel information is used, part of the image information is lost, so the performance of the EMHO algorithm is affected by this fact. Increasing the number of superpixels reduces the information lost; however, it increases the number of nodes in the graph and the algorithm time consumption.

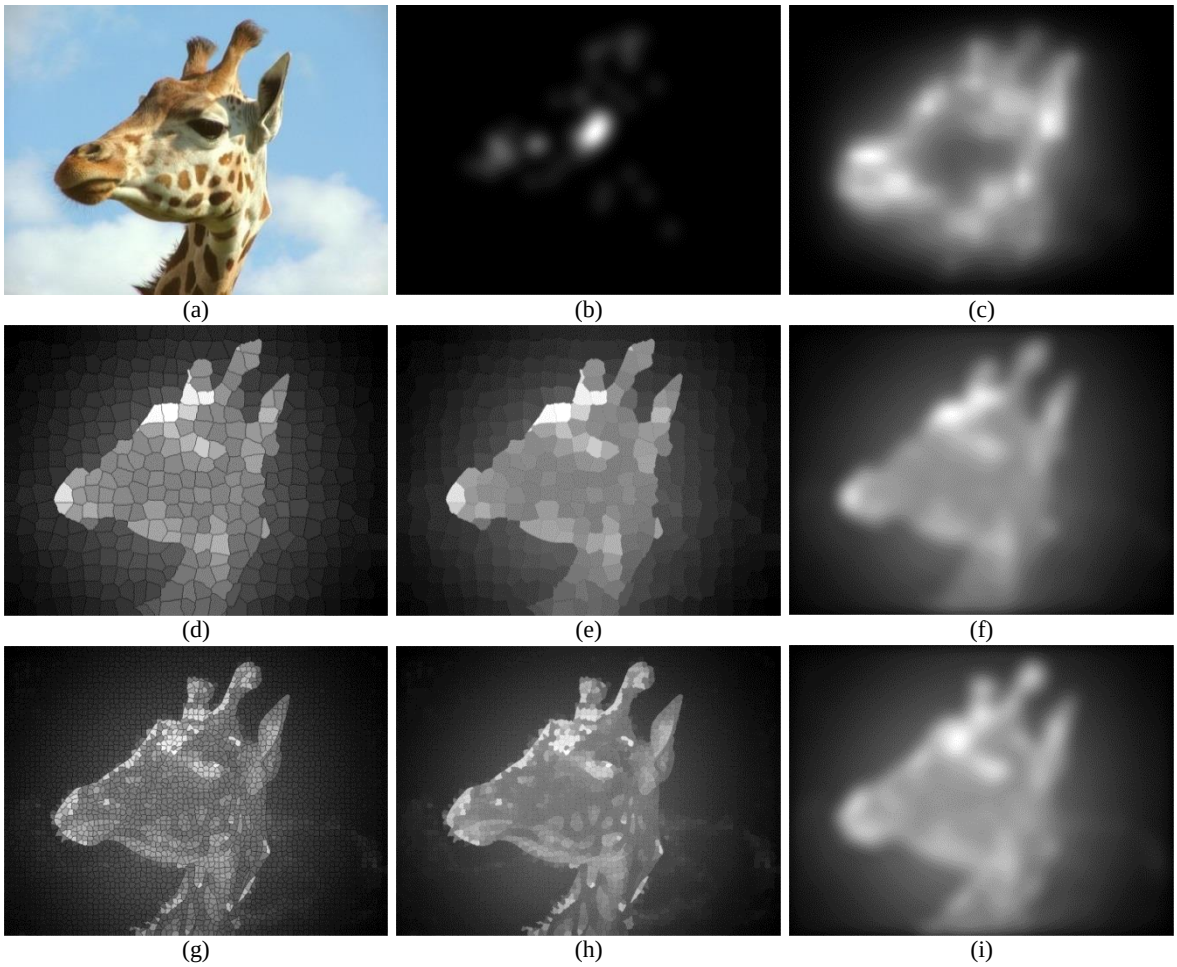


Figure 5-2. Original image (a); its RHSM (b); its GBVS saliency map (c); saliency maps with zeros on the boundaries (d) and (g); saliency maps with average saliency of neighbor superpixels on the boundaries (e) and (h); smoothed saliency maps with $\sigma_{Blurring} = 20$ (f) and (i), N_{SP} =400 in the second row and 4000 in the third row).

To find the effective number of superpixels in EMHO, different numbers of superpixels are used to find PSMs for 100 images randomly selected from LPWHL, and results are compared with GBVS in Table 5-1. The average number of pixels was 779,469. *Score* is used to evaluate PSMs after equalizing the histogram of the PSM of an image with its RHSM from the LPWHL study. Table 5-1 shows the average *Score*, number of maps classified as RSM, and computational times in this test.

Table 5-1. For 100 random LPWHL images, *Score* average, and number of maps classified as RSM for EMHO algorithm compared with GBVS (time consumption computed using: MATLAB® R2012a 64-bit, Image Processing Toolbox V8.0, Window 7 Enterprise, Intel® Core™ i7-2600 @ 3.40GHz, 16GB RAM).

	GBVS	EMHO				
		$N_{SP} = 400$	$N_{SP} = 1000$	$N_{SP} = 2000$	$N_{SP} = 4000$	$N_{SP} = 10000$
Average *Score*	0.0633	0.0577	0.0603	0.0626	0.0644	0.0654
Number of maps classified as RSM	58	70	66	61	59	58
Standard deviation of *Score*	0.0566	0.0537	0.0533	0.0559	0.0586	0.0566
Superpixel algorithm time consumption (on average) (s)	0	4.09	4.32	4.51	4.58	10.98
Saliency algorithm time consumption (on average) (s)	6.8	0.44	0.75	1.53	3.63	10.95
Total time consumption (s)	6.8	4.53	5.07	6.04	8.21	21.93

As shown in Table 5-1, although part of the image information is not reflected in the superpixel image, the EMHO algorithm with 10,000 superpixels performs as well the GBVS algorithm. Numbers of maps classified as RSM for EMHO with 10,000 and 4,000 superpixels are 58 and 59, respectively; and GBVS produced 58 RSMs. However, since EMHO time consumption with 10,000 superpixels is about 2.67 times higher than 4,000, we selected the EMHO algorithm with 4000 superpixels for the rest of this study. For images of size $K \times L$, GBVS has the computational complexity of $O(K^2L^2)$, here $O(779{,}469^2)$. However, the computational complexity of EMHO is $O(N_{SP}^2)$, here $O(4000^2)$. We note that in Table 5-1, the saliency algorithm computational times between GBVS and EMHO are not comparable, since GBVS is written mostly in the C language, whereas EMHO is written in MATLAB. It is well known that in general, MATLAB code execution is slower than C code execution. We anticipate that the EMHO algorithm time consumption would drop significantly if written in C.

5.2.2 Finding the Best Parameter

We define the Reference Human Saliency Vector $\boldsymbol{rhsv}$ of an image with length N_{SP} as the superpixel by superpixel average of the image RHSM, whose l^{th} value is computed using:

$$\boldsymbol{rhsv}(l) = \sum_{k=1}^{N_l} \boldsymbol{RHSM}(i,j)_{l_k} \Big/ N_l \tag{5-4}$$

where $(i,j)_{l_k}$ is the k^{th} pixel location in the l^{th} superpixel.

Since EMHO is designed to use bottom-up visual information, we need to perform the optimization process on a group of images that carry only low-level visual information. However, most of the images in the LPWHL study contain high-level information. Grouping images to low-level and high-level information may be subjected to personal opinion. We counted LPWHL images and found that there are only 105 images that carry no high-level information, and we designate this group of images the LL database in this chapter. 231 images contain human/animal faces, 387 contain groups of people/animals, 257 contain texts, 184 contain vehicles/bikes/boats, 52 contain traffic/other signs, 292 contain buildings, and 119 contain tools/furniture. Some images contain several different types of high-level information.

5.2.2.1 Finding the Best Single Value of σ manually

In this section, we first find the best single values of σ for 40 images randomly selected from the LL database for the EMHO algorithm. This means that a single value is used for the standard deviation of the Gaussian weight function for all superpixels in a given image. The particle swarm optimization (PSO) algorithm, explained in Figure 5-3, is employed to find the best single value of σ for each image. PSO is an evolutionary computational optimization technique developed by Eberhart and Kennedy in 1995 [69, 70]. The objective function of the optimization is to find the value of σ that produces the maximum *Score* value for the PSM compared with the RHSM for that image after histogram matching. In this algorithm, particles' positions are

solutions (σ values here) and particles' velocities are the rates at which solutions are updated in each iteration. PSO starts with randomly initializing N particles and searches for optimum results by updating the position of each particle iteratively [70]. Particles move through the solution space and their velocity vectors are modified using (5-5) towards the current global best solution (G_{Best}) and their own best achieved solution ($\boldsymbol{P}_{best}$) [70]. G_{Best} is the best solution among all particles up to the current iteration. It is the best σ value found by all particles that produces maximum *Score*. $\boldsymbol{P}_{best}$ is an $N \times 1$ vector, with its i^{th} value containing the best solution (σ) found by the i^{th} particle.

$$\boldsymbol{v}_{\{n+1\}} = w \times \boldsymbol{v}_{\{n\}} + c_1 \times r_1 \times \left(\boldsymbol{P}_{best} - \boldsymbol{x}_{\{n\}}\right) + c_2 \times r_2 \times \left(G_{best} - \boldsymbol{x}_{\{n\}}\right) \tag{5-5}$$

where c_1, c_2 and w are weights and $r_1$and r_2 are random numbers all between 0 and 1. $\boldsymbol{v}_{\{n\}}$ and $\boldsymbol{x}_{\{n\}}$ are $N \times 1$ vectors containing velocity and position, respectively, of the particles in iteration n. Positions of the particles are updated using:

$$\boldsymbol{x}_{\{n+1\}} = \boldsymbol{x}_{\{n\}} + \boldsymbol{v}_{\{n+1\}} \tag{5-6}$$

The process is depicted in Figure 5-3.

Initialization
- Select the number of particles.
- Initialize particles' initial position and velocity, $x_{\{0\}}$ and $v_{\{0\}}$ randomly.
- Set w, c_1, c_2 and N_{max},
- Store particles' positions in $\boldsymbol{P}_{best}$,
- Store the first particle's positions as the G_{best},

Particle swarm algorithm:
- While $n < N_{max}$
- for each particle i
  1. Calculate *Score*,
  2. Update $\boldsymbol{P}_{best}(i)$ if *Score* of current position is better than the *Score* of position contained in the memory,
  3. Update the G_{best} if *Score* of current position is better than the *Score* of position contained in the memory,
  4. If the G_{best} solution is acceptable terminate the algorithm,
  5. Otherwise, update the velocity and position of the particle using (5-5) and (5-6).
- End

Figure 5-3. Particle swarm optimization algorithm [69, 70]

PSO has been used effectively in many different research and application areas; however, the trial-and-error in tuning the PSO parameters is one of its major drawbacks [71]. Here we used $c_1 = 0.12, c_2 = 0.08$ and $w = 0.12^{1/N_{max}}$ (N_{max}: maximum number of iterations), which were selected by trial-and-error.

Since PSO does not guarantee getting to the best result, we ran the PSO algorithm 5 times and used the average of the results. The average *Score* and the number of maps classified as RSM for the case with the best single σ values for the 40 images from LL are shown in Table 5-2.

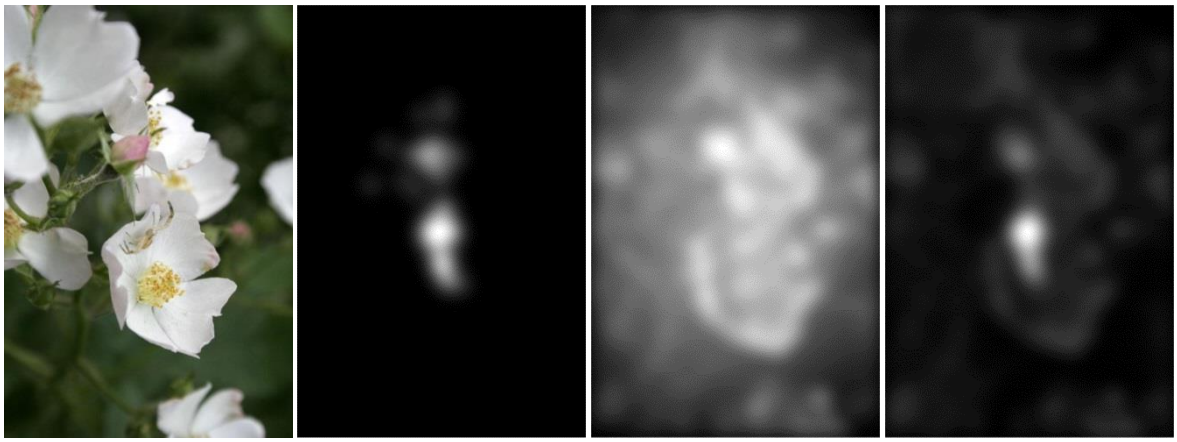
5.2.2.2 Finding the Best Distributed σ manually

We now consider finding the best distributed $\boldsymbol{\sigma}$, ($\boldsymbol{\sigma}_{Best}$: a vector that contains best standard deviation values for all superpixels in an image) for these 40 images in the LL database. Results found manually by trial and error are shown in Table 5-2. As shown in (5-2) and (5-3), the saliency value of each superpixel is independent of the saliency and σ value of other superpixels. Accordingly, we can find the best σ values superpixel by superpixel. We simply calculated the saliency value of a given superpixel in an image with σ incrementally changed between 20 to 600 pixels and selected the value that results in saliency value closest to the $\boldsymbol{rhsv}$ of that superpixel. The average size of superpixels is approximately 14 pixels. Accordingly, any value of σ less than 20 pixels would prevent superpixels from being effectively compared with their immediate neighbors. $\sigma = 600$ is large enough to compare a superpixel with all superpixels in the image. shows an image in the database (a), its RHSM (b), PSMs found using the best single **σ** value and best distributed **σ** (c) and (d), respectively.

As demonstrated in Table 5-2, EMHO with all three conditions outperformed GBVS on this database of low-level information images. EMHO with distributed **σ** generates the best results (highest *Score* value), and only 1 map is classified as RSM. On the other hand, finding the best single **σ** for each image does not significantly improve performance of the EMHO algorithm. The

Table 5-2. *Score* average, and number of maps classified as RSM for EMHO algorithm for 3 different conditions compared with the GBVS for 40 images in LL.

	GBVS	EMHO $N_{SP} = 4000$, with		
		$\sigma = 375$ pixels (Best predetermined Value)	Best single σ found for each image	Best distributed σ Found for each Image
Average *Score*	0.0404	0.0607	0.0827	0.1314
Number of maps classified as RSM	25 (%62.5)	21 (%52.5)	17 (%42.5)	1 (%2.5)
Standard deviation of *score*	0.0324	0.0541	0.0665	0.0566



(a) (b) (c) (d)

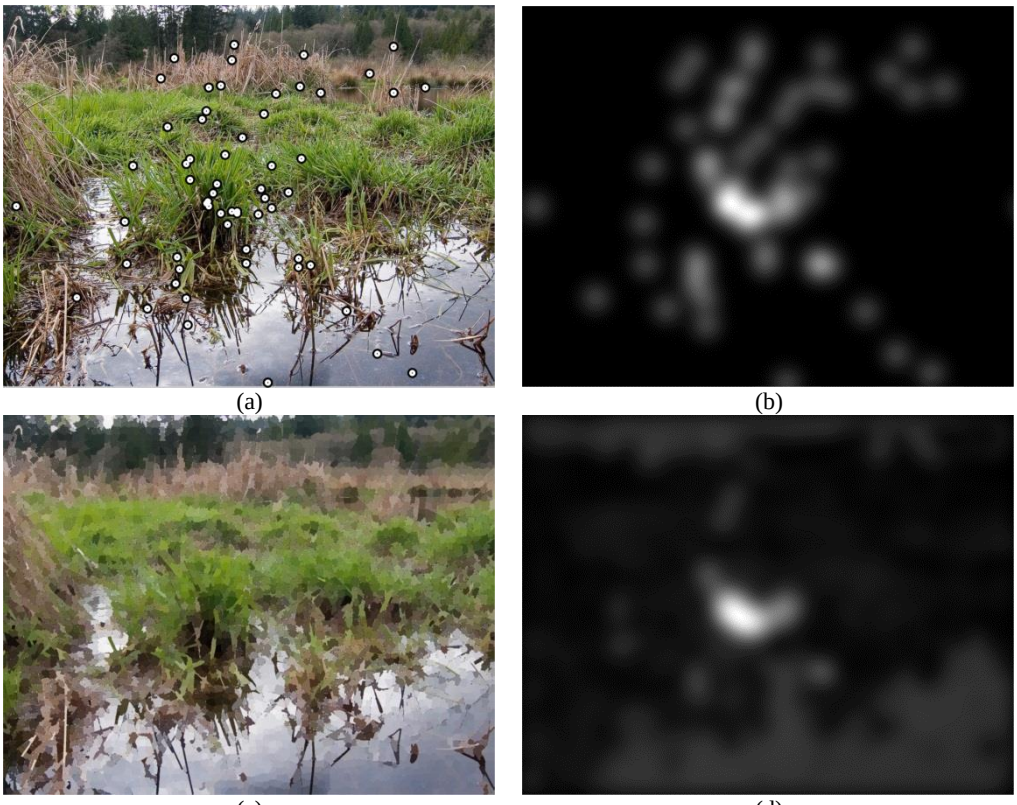
Figure 5-4 Original image (a); its RHSM (b); best saliency map using a single σ (c); and best saliency map found by distributed σ (d).

average of all single **σ** values found for the 40 images is 472.49 pixels. Its standard deviation is 208.33, which shows the range of the best single **σ** is small. We conclude that distributed **σ** helps EMHO algorithm greatly, and we will explore this idea more in what follows. The key remaining problem is to find the distribution of $\boldsymbol{\sigma}_{Best}$ automatically for every image without using RHSMs, because for general wide application, we would not have RHSMs available.

As shown in Table 5-2, there is one image, shown in Figure 5-5 (a), for which the EMHO algorithm with the best distributed **σ** cannot create an acceptable saliency map (based on *Score*), shown in Figure 5-5 (d).

As discussed in Section 3.3, *Score* is a very demanding comparison metric and produces the value of 0.0470 when comparing (d) to (b), which is lower than its threshold value of 0.0553. On the

other hand, *NCC* results in a fairly high value of 0.793 (the *NCC* threshold to distinguish between RSMs and HSMs is 0.621). As shown in Figure 5-5 (a), there are many fine details in this image that are not reflected in the superpixel image (c). This can be a reason that EMHO performed poorly on this image based on *Score*.



(a) (b) (c) (d)

Figure 5-5 Original image (a); its RHSM (b); its superpixel image (c); and the best saliency map found by distributed σ (d).

5.2.3 Finding the Distributed σ Automatically

Figure 5-6 shows 3 images (first row) and their RHSMs (second row), EMHO algorithm PSMs (third row), the inverse distributed **σ** maps (fourth row) (inverse means the brighter each superpixel is, the lower is its σ value), and $\boldsymbol{\sigma}_{Best}$ plotted versus $\boldsymbol{rhsvs}$ (bottom row). To create distributed **σ** maps, the $\boldsymbol{\sigma}_{Best}$ value for each superpixel is assigned to all pixels that belong to that superpixel.

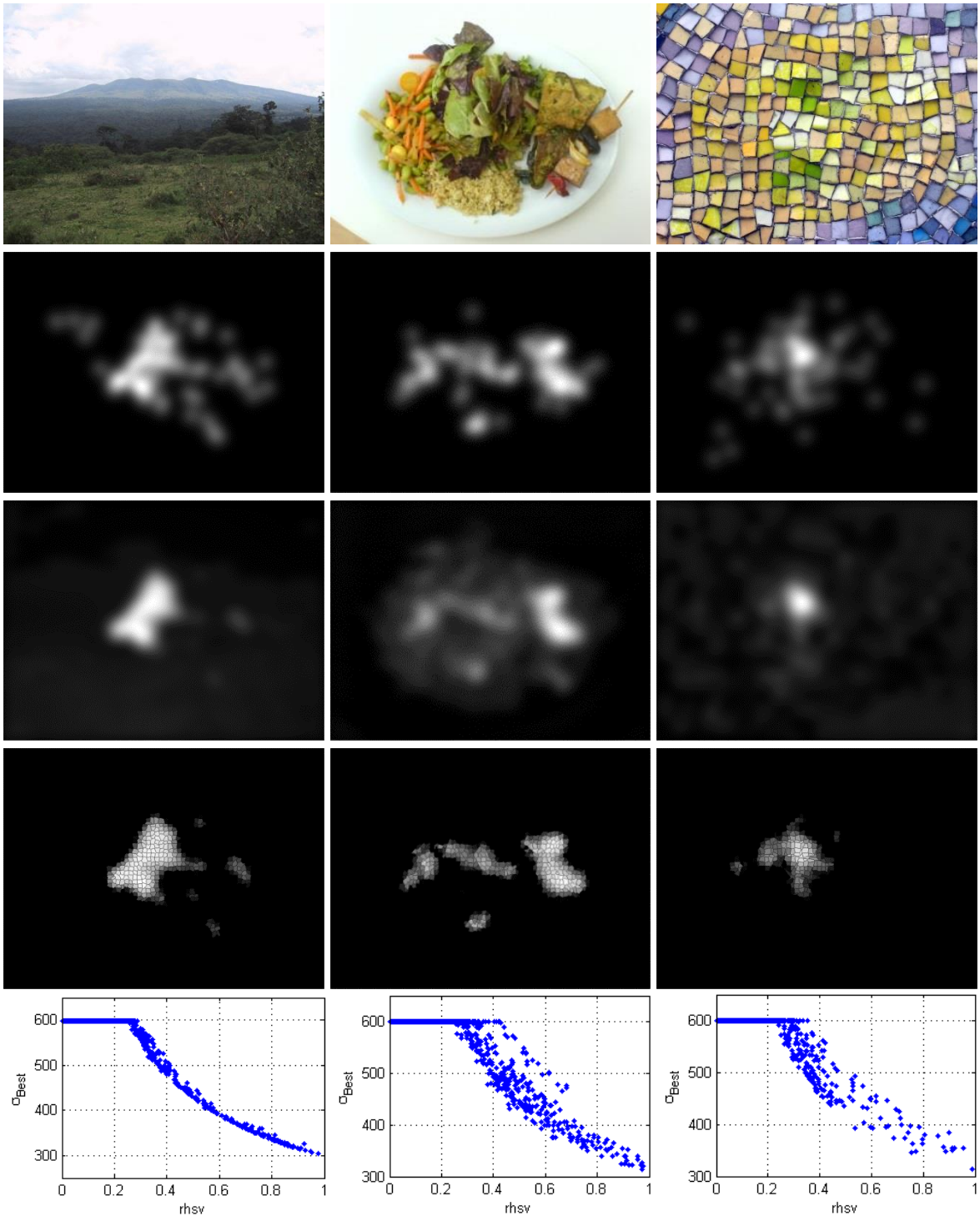


Figure 5-6. Original images (first row); their RHSMs (second row); EMHO algorithm PSMs (third row); the inverse distributed σ maps (fourth row) (the brighter each superpixel, the lower its σ value); and best distributed σ plotted versus $rhsv$s (bottom row).

As shown in Figure 5-6, the inverse **σ** map for each image looks very similar to its RHSM. Also plots of $\boldsymbol{\sigma}_{Best}$ versus $\boldsymbol{rhsvs}$ indicates there may be a nonlinear relation between these variables.

The correlation between $\boldsymbol{rhsv}$ of the 40 images and their $\boldsymbol{\sigma_{Best}}$ were calculated, and the average correlation and standard deviation are -0.887 and 0.0448, respectively. The high average correlation and relatively low standard deviation indicates $\boldsymbol{rhsvs}$ and distributed $\boldsymbol{\sigma}$ vectors are highly (inversely) correlated. From this we conclude that the problem of finding $\boldsymbol{\sigma_{Best}}$ and a best saliency map for an image are very similar.

For the 40 selected images color, orientation, brightness and edge feature spaces were also converted into feature vectors using (5-1), and their correlations with the image $\boldsymbol{\sigma_{Best}}$ were calculated. The maximum correlation found was 0.242 for the red color channel. Accordingly, we concluded there is no significant relation between color, orientation, brightness and edge feature spaces of an image and its $\boldsymbol{\sigma_{Best}}$. Therefore, these feature spaces are not used in the next sections to find the best distributed **σ.**

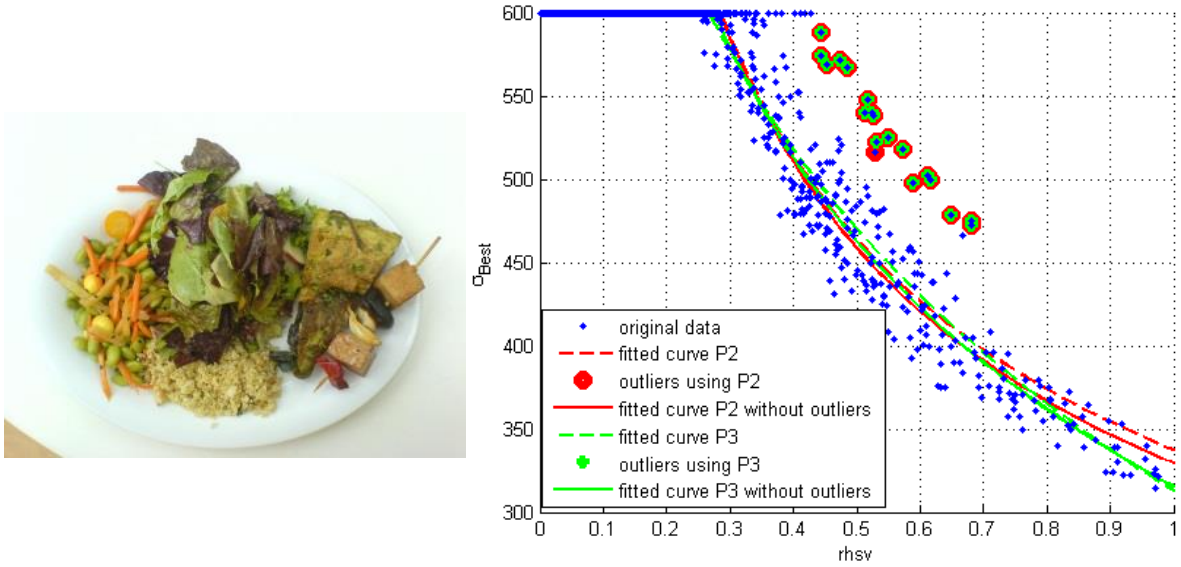
5.2.4 Finding a Relation between $\boldsymbol{\sigma_{Best}}$ and $\boldsymbol{rhsv}$ of images

As shown in the previous section, the inverse distributed σ map of an image appears similar to its RHSM, and they are highly correlated. Two power functions shown in (5-7) and (5-8) were tested to fit the data for the $\boldsymbol{\sigma_{Best}}$ and $\boldsymbol{rhsvs}$ relationship, namely:

$$\boldsymbol{\sigma} = a \times \boldsymbol{s}^b \tag{5-7}$$

$$\boldsymbol{\sigma} = a \times \boldsymbol{s}^b + c \tag{5-8}$$

Both power functions have been used to fit curves to $\boldsymbol{\sigma_{Best}}$ of images versus their $\boldsymbol{rhsvs,}$ by minimizing root mean square error (RMSE), and results are shown in Table 5-3. First, each power function is used to fit the original data. Afterwards, the fitted curve is used to remove outliers, and then another curve is fitted to the data without outliers. As an example, Figure 5-7 (b) shows $\boldsymbol{\sigma_{Best}}$ of the image shown in (a) plotted vs. its *rhsv* and the corresponding fitted curves.



(a) (b)

Figure 5-7. An image (a); and its σ_{Best} plotted vs. *rhsv* and fitted curves (b).

In Figure 5-7, P2 and P3 stand for power functions with 2 and 3 parameters shown in (5-7) and (5-8), respectively. To find outliers in Figure 5-7, the difference between the $\boldsymbol{\sigma_{Best}}$ values and the predicted $\boldsymbol{\sigma_{Best}}$ values are calculated. The 25th (q_1) and 75th (q_3) percentiles of the differences are calculated, and data-points with differences larger than $q_3 + 1.5 \times (q_3 - q_1)$ or smaller than $q_1 - 1.5 \times (q_3 - q_1)$ are drawn as outliers.

Table 5-3. Results of fitting power functions to σ_{Best} vs. *rhsv* of 40 images randomly selected from LL.

	Average of *a*s	***a*s standard deviation**	**Average of *b*s**	***b*s standard deviation**	**Average of *c*s**	***c*s standard deviation**
P2, original data	315.6	75.0	-0.434	0.061	---	---
P2, outliers removed	311.1	78.4	-0.451	0.066	---	---
P3, original data	3061.6	13475.8	-0.027	0.424	-3036.4	13475.1
P3, outliers removed	3131.7	13455.3	-0.127	0.410	-3106.9	13454.2

Table 5-3 shows the parameters founds for the power function to fit $\boldsymbol{\sigma_{Best}}$ versus *rhsv* of 40 images selected from the LL database. Both cases of the P2 function parameters have low standard deviations in comparison with both cases of the P3 function. From this, we expect the average P2 curves to produce lower error on predicting the relation between $\boldsymbol{\sigma_{Best}}$ and $\boldsymbol{rhsv}$ of all images.

Table 5-4. Average RSME and RSQ of power functions used to fit a curve to σ_{Best} vs. *rhsv* for 40 images from the LL database.

	P2, original data	P2, outliers removed	P3, original data	P3, outliers removed
RMSE	111.6	114.9	773.5	4975.7
RSQ	0.909	0.909	0.912	0.913

Here R-Square (RSQ) is the square of the correlation between the $\boldsymbol{\sigma_{Best}}$ values and the curve-predicted $\boldsymbol{\sigma_{Best}}$ values. It measures the effectiveness of the fit and accounts for the variation of the data. As shown in Table 5-4, P2 on original data fits the data very well, with average RSME and R-squares of 111.6 and 0.909, respectively. Accordingly, the power function with 2 parameters (a and b shown in Table 5-3) is selected here to fit the data. Figure 5-8 shows the three images depicted in Figure 5-6 and their $\boldsymbol{\sigma_{Best}}$ plotted versus their $\boldsymbol{rhsvs}$, together with the P2 power functions that best fits the data.

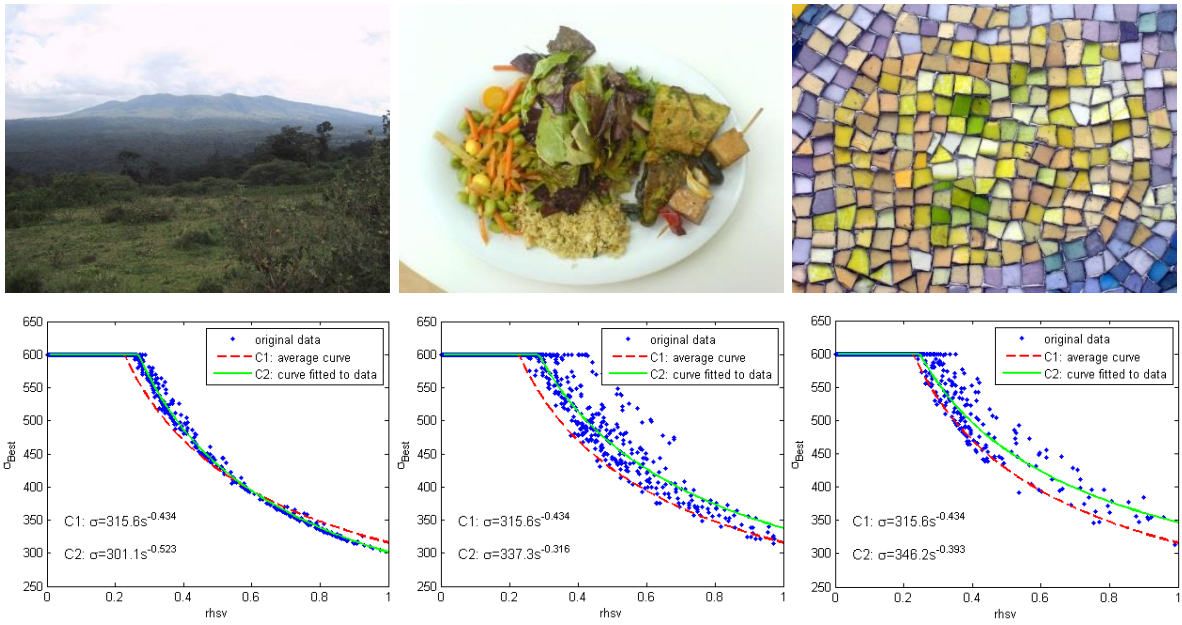


Figure 5-8. 3 images from the LPWHL database, their σ_{Best} plotted vs. their *rhsvs*, and the fitted curves.

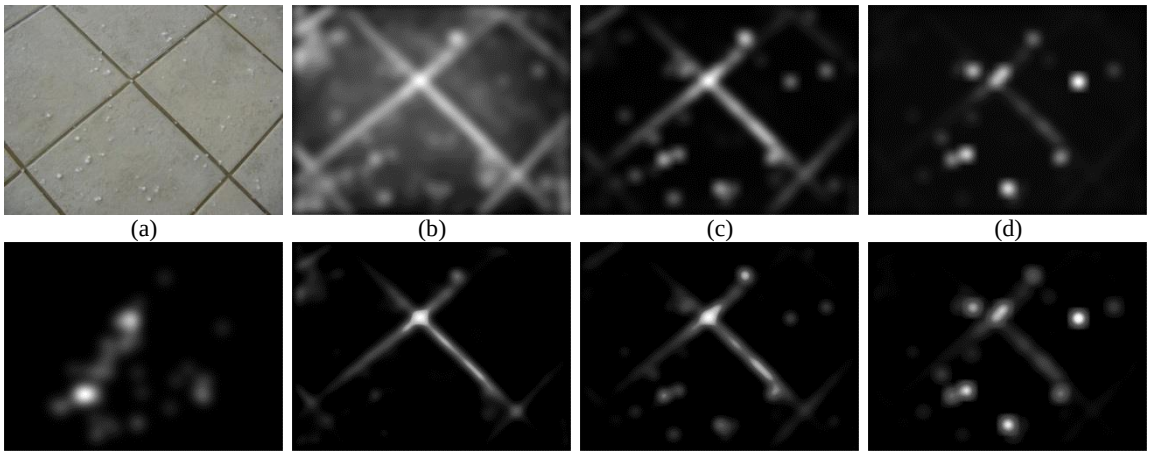
In Figure 5-8, C1 stands for the P2 curve with parameters shown in the first row of Table 5-3 (the average of all P2 curves) and C2 stands for a P2 curve fitted to the original data for the given image. As demonstrated in Figure 5-8, C1, the average of all P2 curves, fits the data of these images very well.

5.2.5 EMHO with an Iterative Approach to Estimate $\boldsymbol{\sigma_{Best}}$

We propose an iterative method to estimate $\boldsymbol{\sigma_{Best}}$ of an image. First, the EMHO algorithm is applied using a single value $\sigma = 375$ and a saliency vector ($\boldsymbol{s}$ in (5-3)) is computed. Then, the distributed $\boldsymbol{\sigma}$ of the next steps, $\boldsymbol{\sigma}_{\{n+1\}}$, is computed using the saliency vector of the current step, as follows:

$$\boldsymbol{\sigma}_{\{n+1\}} = a \times \boldsymbol{s}_{\{n\}}^{b} \tag{5-9}$$

where n stands for the current step. We found that 4 iterations produce best results. Figure 5-9 shows an original image (a), its RHSM (b), EMHO saliency map using $\sigma = 375$ (c), PSMs after one, two and four iterations (d), (e) and (f). Here $a = 315.6$ and $b = -0.434$ (averages of all as and bs found earlier).



(a) (b) (c) (d)

(e) (f) *Score*: 0.0594 (g) *Score*: 0.0713 (h) *Score*: 0.0814

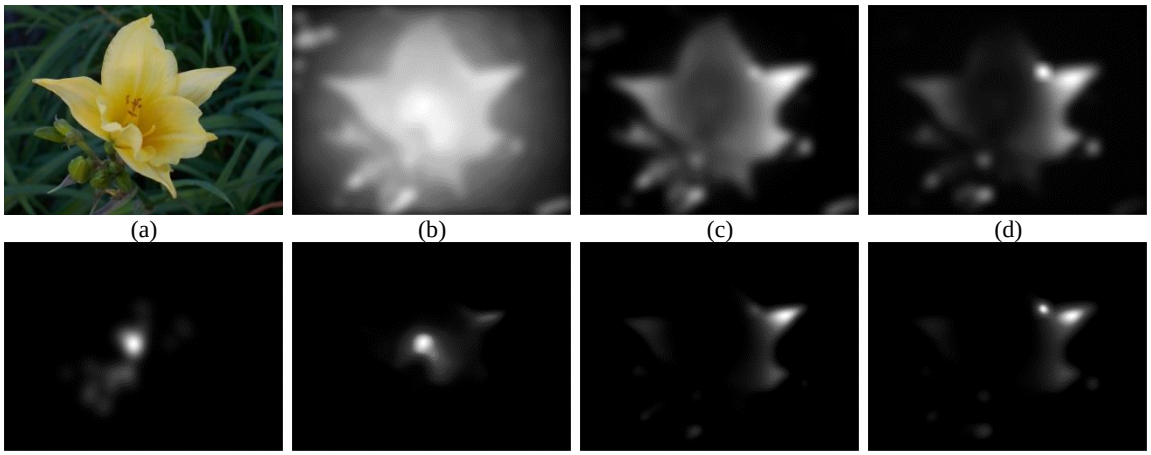
Figure 5-9. Original images (a); its RHSM (e); PSM using a single σ=375 (b); PSMs found by distributed σ (Case II) iteratively in 1st (c) and 4th (d) iterations; and PSMs after histogram matching (f), (g) and (h).

As shown in Figure 5-9, the iterative approach helps EMHO to create better saliency maps for the image shown in (a). The EMHO algorithm with this iterative approach is then applied to 40 images from the previous section for two cases (Case I and Case II), and the results are shown in Table 5-5. In Case I, a and b are found for each image. In Case II, the average of all as and bs over all 40 images are used. We note that Case I is feasible here because the $\boldsymbol{\sigma_{Best}}$ values for these images have already been found, and this approach is not applicable for an image outside the LPWHL database since its RHSM would be required to find its $\boldsymbol{\sigma_{Best}}$.

Table 5-5. *Score* average, and number of maps classified as RSM for EMHO algorithm for different conditions, compared with the GBVS.

	GBVS	EMHO, $N_{SP} = 4000$, with		
		$\sigma = 375$ pixels (Best predetermined Value)	σ found iteratively using (5-9), Case I	σ found iteratively using (5-9), Case II
Average *Score*	0.0404	0.0607	0.0771	0.0766
Number of maps classified as RSM	25 (%62.5)	21 (%52.5)	14 (%35)	16 (%40)
Standard deviation of *Score*	0.0324	0.0541	0.0394	0.0485

As shown in Table 5-5, the EMHO algorithm Case I outperforms other approaches; however, Case II is very close. Since a and b in Case II are fixed and image invariant, this approach is proposed to be applied to any image in the database, or for that matter, to any image outside the database. Figure 5-10 (a), shows an example for which the iterative approach does not help EMHO to create better results.



(a) (b) (c) (d)

(e) (f) *Score*: 0.2105 (g) *Score*: 0.0407 (h) *Score*: 0.0396

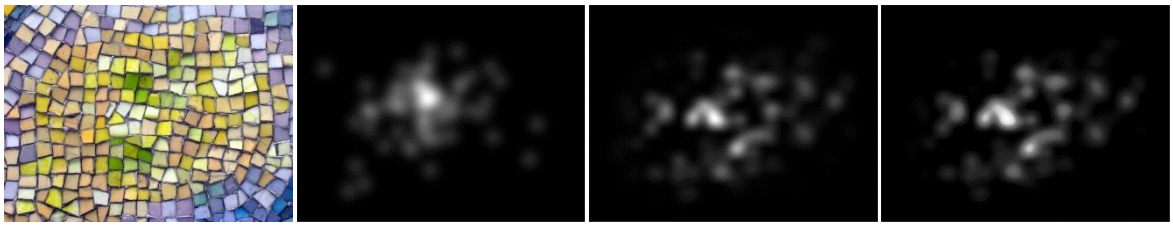
Figure 5-10. Original images (a); its RHSM (e); PSM using a single σ=375 (b); PSMs found by distributed σ (Case II) iteratively in 1st (c) and 4th (d) iterations; and PSMs after histogram matching (f), (g) and (h).

As shown in Figure 5-10 (b), this RHSM is highly center biased, which can be a reason for poor EMHO results on (a).

From this point on in our work, the EMHO algorithm refers to the algorithm with the iterative approach using average a and b shown in Table 5-3.

5.3 EMHO Applied to Images in LL

In the previous section EMHO parameters were selected using 40 images randomly selected from the LL database. Herein, the EMHO algorithm has been applied to the rest of the images in the LL database (65 images). Histograms of PSMs are matched to the corresponding RHSMs before using *Score*. The results are presented in Tables 5-6 and 5-7. Figure 5-11 shows an image, its RHSMs and EMHO PSMs before and after histogram matching. Herein, we used all saliency comparison metrics to evaluate the performance of EMHO and compared the results to those of all algorithms analyzed in Chapter IV.



(a) (b) (c) (d)

Figure 5-11. Original images (a); its RHSMs (b); PSM using EMHO (c); and histogram equalized PSM (d).

Table 5-6. Averages for all visual saliency models on 65 images from the LL database and their rankings based on their metric average.

		Comparison metrics in increasing rank number (from left to right)								
		Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***
Visual saliency mechanisms in increasing rank number (top to bottom) based on *Score*	**EMHO**	0.0818 (1)	0.597 (1)	0.778 (1)	0.604 (1)	0.786 (1)	0.66 (1)	0.939 (1)	0.288 (1)	0.997 (1)
	GBVS	0.0594 (2)	0.468 (2)	0.668 (2)	0.409 (2)	0.72 (2)	0.573 (2)	0.921 (2)	0.263 (5)	0.995 (2)
	IS	0.0567 (3)	0.426 (4)	0.637 (5)	0.356 (5)	0.665 (5)	0.399 (10)	0.907 (3)	0.231 (9)	0.995 (2)
	EH	0.0536 (4)	0.441 (3)	0.648 (3)	0.374 (3)	0.697 (3)	0.523 (4)	0.906 (4)	0.256 (6)	0.995 (2)
	CC	0.0461 (5)	0.414 (5)	0.64 (4)	0.361 (4)	0.667 (4)	0.568 (3)	0.861 (8)	0.286 (2)	0.992 (8)
	CASD	0.0453 (6)	0.39 (6)	0.613 (6)	0.313 (6)	0.653 (6)	0.47 (6)	0.859 (9)	0.192 (10)	0.995 (2)
	SR	0.0417 (7)	0.35 (7)	0.586 (8)	0.264 (8)	0.612 (8)	0.477 (5)	0.866 (7)	0.264 (4)	0.993 (7)
	SUN	0.0394 (8)	0.35 (7)	0.588 (7)	0.266 (7)	0.613 (7)	0.455 (7)	0.869 (6)	0.27 (3)	0.992 (8)
	IK	0.035 (9)	0.289 (10)	0.574 (9)	0.247 (9)	0.599 (9)	0.434 (8)	0.887 (5)	0.251 (8)	0.966 (11)
	AIM	0.0311 (10)	0.324 (9)	0.57 (10)	0.233 (10)	0.59 (10)	0.271 (11)	0.78 (11)	0.131 (11)	0.995 (2)
	FTSRD	0.0275 (11)	0.239 (11)	0.549 (11)	0.197 (11)	0.528 (11)	0.425 (9)	0.82 (10)	0.254 (7)	0.986 (10)

As shown in Tables 5-6 and 5-7, EMHO significantly outperforms the other visual saliency models based on both metrics average and the number of maps classified as RSMs. The number

of maps classified as RSM for EMHO is less than half the second best visual saliency model based on *Score* and also using the averages over all comparison metrics.

Table 5-7. Number of maps classified as RSM for all visual saliency models on 65 images from the LL database and their rankings based on their number of RSMs (for threshold values see Table 3-1).

		Comparison metrics in increasing rank number (from left to right)									
		Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***	**Average**
Visual saliency mechanisms in increasing rank number (top to bottom) based on *Score*	**EMHO**	16 (1)	8 (1)	2 (1)	2 (1)	0 (1)	14 (1)	19 (1)	25 (1)	0 (1)	9.6 (1)
	GBVS	40 (2)	32 (2)	25 (3)	24 (4)	10 (2)	24 (3)	26 (2)	27 (3)	8 (3)	23.2 (2)
	EH	41 (3)	38 (3)	24 (2)	23 (2)	10 (2)	19 (2)	34 (4)	30 (4)	16 (6)	24.3 (3)
	CC	45 (4)	40 (4)	25 (3)	23 (2)	16 (4)	27 (4)	41 (6)	25 (1)	5 (2)	27.1 (4)
	IS	46 (5)	43 (5)	35 (5)	33 (5)	20 (5)	39 (10)	34 (3)	40 (9)	12 (4)	32.2 (5)
	CASD	46 (5)	49 (6)	37 (6)	38 (6)	22 (6)	31 (5)	45 (9)	41 (10)	13 (5)	34.3 (6)
	AIM	53 (7)	58 (8)	43 (8)	41 (8)	35 (8)	36 (8)	38 (5)	35 (7)	20 (9)	37.7 (7)
	IK	54 (8)	55 (7)	42 (7)	40 (7)	36 (9)	44 (11)	55 (11)	49 (11)	17 (7)	41.7 (9)
	SUN	54 (8)	58 (8)	45 (9)	44 (9)	33 (7)	35 (6)	43 (7)	31 (5)	18 (8)	38.3 (8)
	SR	54 (8)	61 (10)	52 (10)	50 (10)	38 (10)	38 (9)	44 (8)	32 (6)	44 (11)	43.9 (10)
	FTSRD	55 (11)	62 (11)	52 (10)	51 (11)	51 (11)	35 (6)	48 (10)	38 (8)	43 (10)	44.7 (11)

5.3.1 Optimizing Blurriness and Center-Bias for Images in the LL Database

Similar to the previous chapter, we find the optimized level of blurriness and center-bias of all visual saliency models by varying the variance σ of the Gaussian filter and the weight w in (4-1) for those 65 images from the LL database. The optimization is performed base on the number of maps classified as RSMs. The same Center-Map (***CM***) as in Section 4.1.2.1 is used. Similarly, the histograms of the center-map and the PSMs are matched to the histograms of the RHSMs. Figure 5-12 demonstrates how *Score* averages and number of maps classified as RSM is affected by changing the blurring σ value and weight w of the center-map.

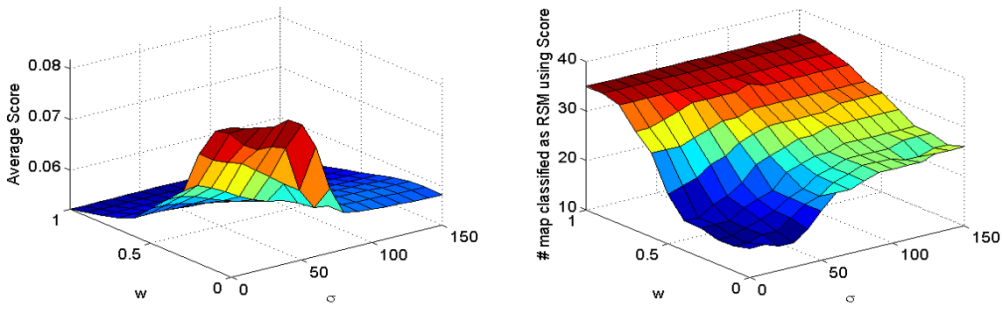


Figure 5-12. Effect of blurring and adding the center-map on the performance of the EMHO.

Table 5-8. Best blurring σ and center map weight *w* and its effect on the performance of the visual saliency models on 65 images from the LL database using *Score*.

Visual saliency model	Best σ	Best *w*	Average *Score*	Min # maps classified as RSM	Improvement Percentage	New Ranking
EMHO	10	0	0.0808	15 (23.1%)	6.25%	1
GBVS	0	0.3	0.0815	23 (35.4%)	42.50%	2
CC	10	0.5	0.0712	26 (40%)	36.59%	3
EH	10	0.3	0.0718	27 (41.5%)	40.00%	4
IS	0	0.2	0.0805	31 (47.7%)	32.61%	5
AIM	10	0.5	0.0714	32 (49.2%)	30.43%	6
CASD	10	0.6	0.0721	33 (50.8%)	38.89%	7
SUN	20	0.5	0.0682	34 (52.3%)	35.85%	8
IK	10	0.7	0.0582	35 (53.8%)	35.19%	9
SR	0	1	0.0523	35 (53.8%)	35.19%	10
FTSRD	0	1	0.0523	35 (53.8%)	36.36%	10

As shown in Table 5-8, all visual saliency models' performances improved with blurring and adding a center biased map. EMHO used no center map and outperformed other visual saliency models in terms of number of maps classified as RSM. Other models have the same order as in Table 4-9 except for IK and SUN that their orders are switched. FTSRD and SR used 100% of the center map, which means they are outperformed by the center map here.

5.4 EMHO Applied to all Images in the LPWHL Database

In this section, the EMHO algorithm has been applied to all the 1003 images in the LPWHL database [40], and the results are presented in Tables 5-9 and 5-10. Here, we used all saliency comparison metrics to evaluate the performance of EMHO and compared results to the GBVS algorithm after histogram matching (the best approach from Chapter IV).

Table 5-9. Average of all metrics for the GBVS and EMHO algorithms on the natural images LPWHL database (higher is better).

	Comparison metrics in increasing rank number (from left to right)								
	Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***
EMHO	0.066	0.840	0.6786	0.422	0.745	0.644	0.914	0.165	0.997
GBVS	0.0538	0.441	0.648	0.369	0.709	0.543	0.911	0.273	0.995

Table 5-10. Number of maps classified as RSM for GBVS and EMHO in the LPWHL database (Thr=0.0553, see Table 3-1) (lower is better).

	Comparison metrics in increasing rank number (from left to right)									
	Score	***NDM***	***NCC***	***Cosθ***	***ROC Area***	***Hit Rate***	***DS***	***Score2***	***NKL***	**Average**
EMHO	542	381	331	347	68	273	503	479	106	336.7
GBVS	661	612	450	427	145	414	506	445	15	408.3

As shown in Tables 5-9 and 5-10, EMHO with iterative distributed σ outperforms the GBVS algorithm based on all saliency comparison metrics except for *Score2* and *NKL*, the two lowest ranked metrics after histogram matching. However, comparing results in Table 5-10 with those in Table 5-7 shows that EMHO results, while the best, are not significantly better than those from other models when it is applied to all images in LPWHL. This is because EMHO is designed for images with low-level information, and 89.5% of the images in this database carry high-level visual information.

5.4.1 Optimizing Blurriness and Center-Bias for all LPWHL Images

The optimum level of blurriness and center-bias for EMHO are found by varying the variance σ of the Gaussian filter and the weight w in (4-1) for all images in the LPWHL database, and results are shown in Figure 5-13.

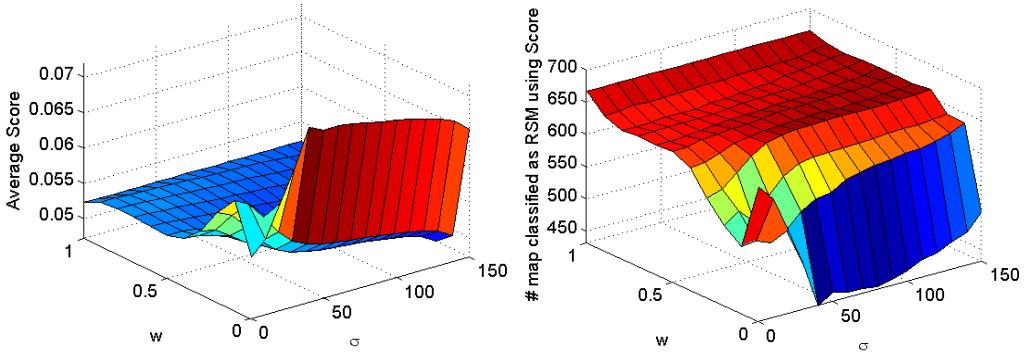


Figure 5-13. Effect of blurring and adding the center-map with different weights on the performance of the EMHO on the LPWHL database.

EMHO results are compared with the results from the GBVS algorithm from the previous chapter in Table 5-11.

Table 5-11. Best blurring σ and center map weight *w* and its effect on the performance of the EMHO and GBVS using *Score* on all LPWHL images.

Visual saliency model	Best σ	Best *w*	Average *Score*	Improvement Percentage	Min # maps classified as RSM	Improvement Percentage
EMHO	40	0	0.0720	9.4%	432 (43.1%)	20.3%
GBVS	10	0.5	0.0564	4.9%	614 (61.2)	7.1%

By comparing results in Table 5-11 with those in Tables 5-9 and 5-10, EMHO again benefits from blurring its PSMs. EMHO outperformed the GBVS in terms of both average of *Score* and number of maps classified as RSM.

Figure 5-14, top row shows three images from the synthetic image database, and their superpixel images created using The TurboPixel algorithm [68] in the bottom row. These are examples to illustrate that the superpixel algorithm is not able to create acceptable results on some of the images in the synthetic image database used in Chapter IV, such that EMHO was not applied to the synthetic image database here.

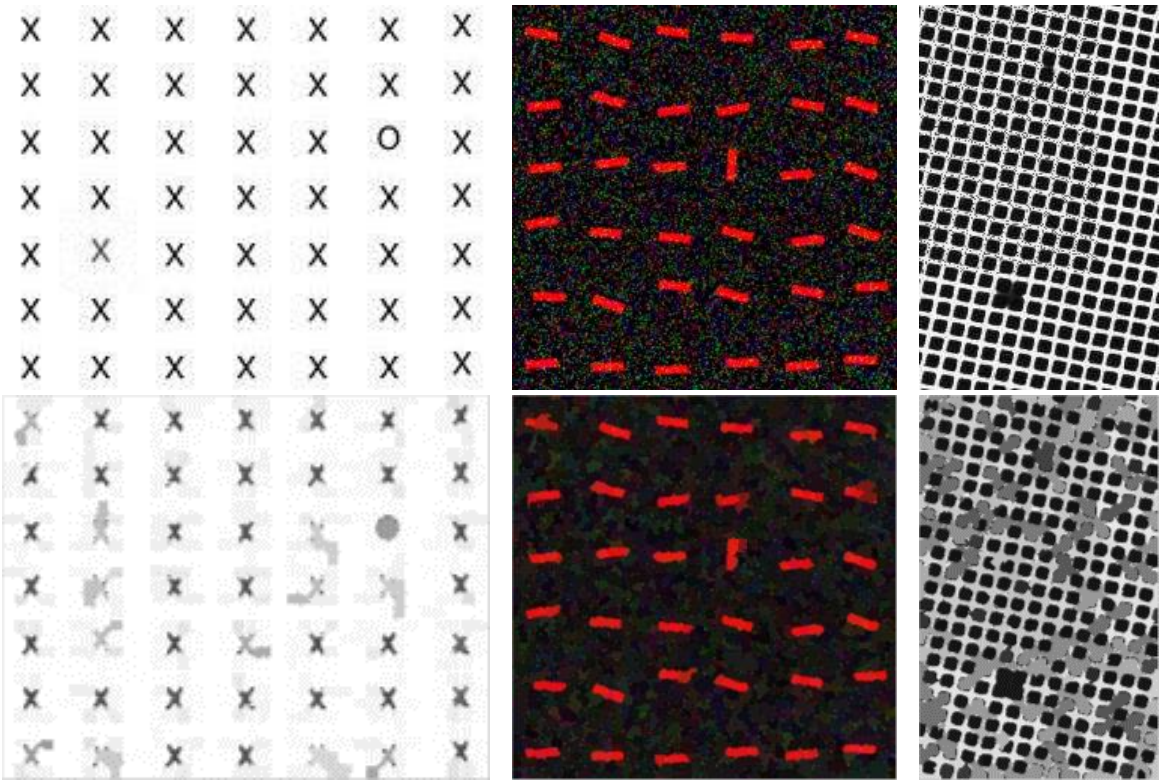


Figure 5-14. Three images from the synthetic image database (top row) with troublesome superpixel images (bottom row).

5.5 Discussion

A new visual saliency model, EMHO, was introduced in this chapter derived from GBVS, the best saliency model in the literature. EMHO works with superpixel images and provides a method to find parameters for each region of the image separately. Images in the LPWHL database were classified into two groups: (1) those that carry only low-level information; and (2) those that contain some type of high-level information. EMHO outperformed all other models from Chapter IV on both image classes. However, the differences between EMHO and other models on images with low-level information were significant.

Out of 105 images in the LPWHL database with only low-level information (the LL database), EMHO created 28 maps classified as RSM based on *Score* (*Score* was selected as the best saliency map comparison metric in Chapter III). Although EMHO produces significantly better results than other visual saliency models, we conclude that there continues to be opportunity to improve bottom-up visual saliency models.

Figure 5-15 shows six images from the LPWHL database and their fixations in the database. Most of the images in the database contain some high-level visual information, such as human or animal faces, text, and vehicles. As shown in Figure 5-15, human observers tend to look at human or animal eyes, (a) and (c), or read texts, (b) and (d). On the other hand, this database contains only a few images (approximately 10%) that reflect solely low-level visual information, e.g. (e) and (f). To study purely bottom-up saliency mechanism of the human visual system, many more images are required that contain only low-level information. Images with high-level information may misdirect the training and parameter selection investigations.

Although Figure 5-15 (e) shows only low-level visual information (different colors), most of the fixations occur in the central parts of the image. This shows that human observers tend to focus on the central regions of the displays. This is because humans can see and analyze visual information around fixation location of the eye up to a certain angle. Thus, when observers fixate

on a point close enough to the image boundary, their visual system analyzes visual information of the boundary pixels as well as the fixation point. Therefore, in practice fixations rarely happen to be on the boundary. To overcome this problem in eye tracking studies, we suggest showing larger images (with many more details) in larger displays and record more (for example 20) than the first 5 fixations of human observers. With larger images, human observers would move eyes and head more frequently. This would help in tracking human eye movement as the information flows in the visual system and as the human saliency mechanism moves to the next salient location.

(a) (b) (c)

(d) (e) (f)

Figure 5-15. Some images from the LPWHL database and their fixations in the database.

CHAPTER VI

6 APPLICATIONS OF SALIENCY MAPS TO DISHWARE INSPECTION

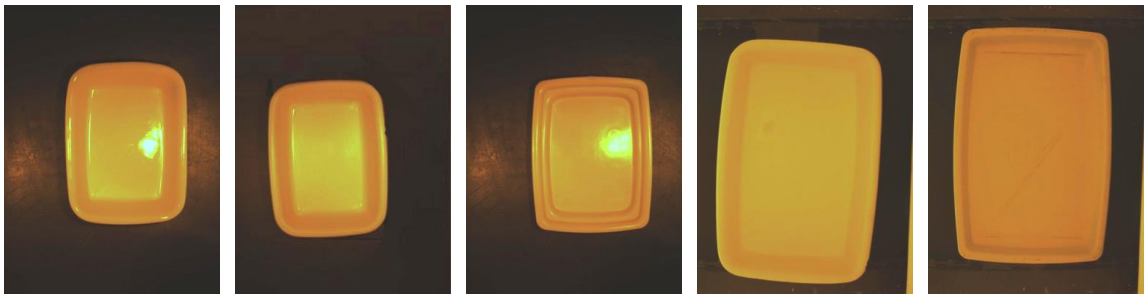
When a human observer views an image his visual saliency mechanism automatically selects salient locations in the display and allocates attention to them, regardless of specific qualities of the saliency, such as shape, color, brightness, or orientation. Analyzing if any of the selected locations correspond to a desired target is the next step. In this Chapter, the application of this approach is tested in dirt detection in commercially washed dishware. First, a simple edge-detection method is employed to find the dish in the image, and the area of the dish is used to set the parameters of the algorithm. Afterwards, a saliency detection mechanism is applied to calculate the saliency map of the dish image. Finally, the height of the global maximum of the saliency map is analyzed for dish inspection, and the local maxima of the saliency map are tested to find dirty spots.

Although the process of dirt detection in dishware seems straightforward, there are many difficulties in designing a computer vision system for this purpose. Duong and Hoberock [72] count the following items as challenges in applying image processing to dish inspection: (I) The definitions of a clean dish and a dirty dish are not well-defined; (II) Color and intensity of dish images vary because of the non-flat geometry of the dish; (III) Dish images usually include glares and shadows; (IV) Color and intensity of dirty spots made with different foods are dissimilar; (V) Dish images are sensitive to changes in lighting and power fluctuation.

The dish set used in this research is the same as the set used by Duong and Hoberock [72]. They implied that image intensity contains enough information to inspect this dish set. To do so, they investigated the darkness of each point/region in the gray level image and compared it with its surroundings. Accordingly, we analyze the saliency map extracted from the gray level image, instead of the color image. This is advantageous because decreasing the number of feature spaces used for constructing the saliency map increases the speed of the method.

6.1 Dishware Inspection Using Saliency Maps

We analyze the application of two visual saliency models, GBVS and EH, to inspect a set of 112 dish images of 5 different dish types and sizes, as shown in Figure 6-1. Dishes (a), (c) and (e) are of plastic material; dishes (b) and (d) are of ceramic material.



(a) (b) (c) (d) (e)

Figure 6-1. The dish set used in this research.

Visual saliency models are applied to 35 clean and 77 dirty dish images, which is part of the dataset used by Duong and Hoberock [72]. Dirt points were created manually using different food particles in different locations, with different shapes and sizes. As illustrated in Figure 6-2, the image dataset consists of images of different dishes in various locations and orientations.

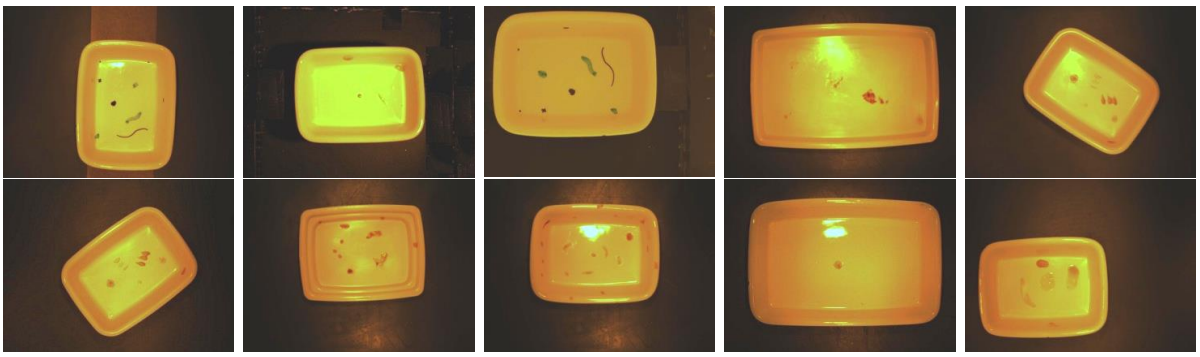


Figure 6-2. Ten dirty dish images in the dataset.

6.1.1 Dishware Inspection Using GBVS Saliency Maps

First, the GBVS visual saliency model is incorporated in the dishware inspection algorithm. Figure 6-3 shows four dishes (a) to (d) and their saliency maps (e) to (h) computed using the GBVS visual saliency model. For convenience, we present saliency maps in 2-D heat maps. The color scale on the right explains that dark red show regions with high saliency values and blue regions have lowest saliency value. As mentioned above, background of images are eliminated in the algorithm. Therefore, no saliency data is gathered for background and they are shown in white in the heat maps.

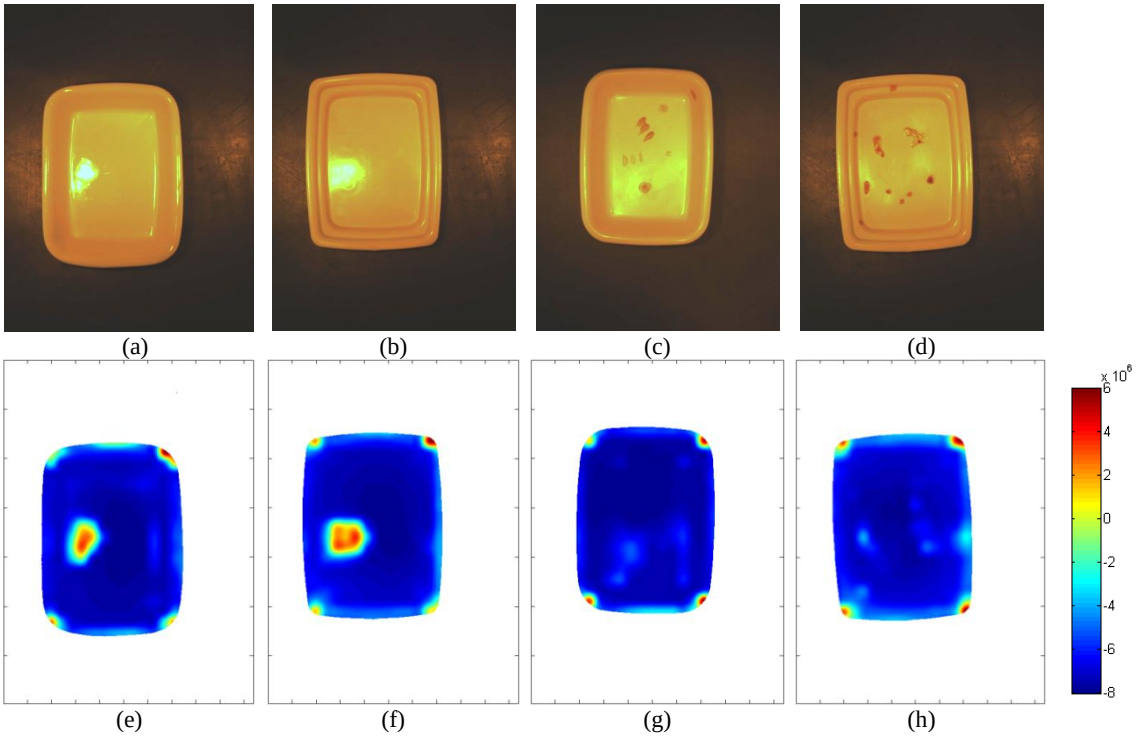


Figure 6-3. Four dishes (a) through (d); and their saliency maps (e) through (h).

As demonstrated in Figure 6-3, the GBVS algorithm highlighted the corner of the dishes as the most salient regions in the images, in both dirty and clean dishes. This means that corner of the dishes are found to be more salient than the dirty regions in the dishes. The sharp change in the intensity at the boundary of the dish causes this problem. GBVS highlighted glare (or specular reflection) in the dishes as well as some of the dirty points. Some of the dirty regions in dishes (c) and (d) are not highlighted in their saliency maps (g) and (h). GBVS was found to be the best

bottom-up visual saliency model in Chapter IV. We believe the reason that GBVS performed poor in dish clinginess application might be that its parameters are set for natural images. Accordingly, we conclude that GBVS cannot be used in our dishware inspection algorithm.

6.1.2 Dishware Inspection Using EH Saliency Maps

Herein, we employ our EH visual saliency model for dishware inspection and results are shown in Figure 6-4. As mentioned above, feature saliency maps extracted from the gray level images are analyzed here for dishware inspection. Also, only one DOG filter in the center-surround mechanism, which had the best performance during our trial-and-error process, is used to calculate the feature saliency map. Consequently, there is only one feature saliency map in this method, such that the problem of finding an applicable method to combine feature saliency maps does not occur here. After several experiments, the following parameters were used in (2-13) to estimate the DOG function in the center-surround mechanism:

$$\sigma_{inh} = A^{0.5}/5, \tag{6-1}$$

$$\sigma_{ex} = \sigma_{inh}/6, \tag{6-2}$$

where A is the area of the dish in pixels. The spread of the weight function (2-21) in the normalization process was determined such that:

$$A^{0.5}/6 \leq r \leq A^{0.5}/3 \tag{6-3}$$

In what follows, we employ this saliency mechanism to calculate saliency maps of dish images for dirt detection in dishware.

Figure 6-4 shows four dishes (same as in Figure 6-3) and their 2-D color saliency maps produced by the EH saliency model. As we expected from a good visual saliency model, dirty spots gained high saliency values in the saliency map and clean areas of the dishes gained relatively low saliency values (mostly are plotted in green and yellow). Also, since the darkness of the gray level image is analyzed, glare gained lowest saliency value and are plotted in blue.

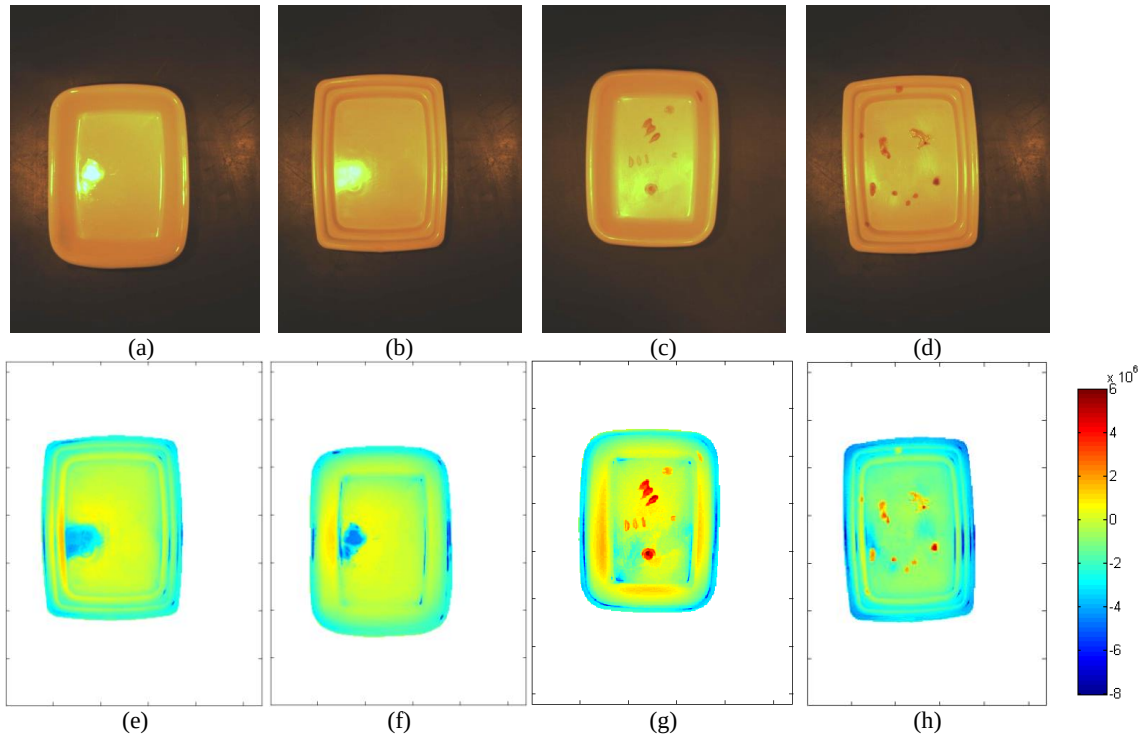


Figure 6-4. Four dishes (a) to (d); and their saliency maps (e) to (h).

In this study, we checked the maximum saliency (MS) of the dishes for dish inspection. Figure 6-5 illustrates the MS distribution of all dishes in the dataset. Blue bars show the MS of dirty dishes and green bars present the MS of clean dishes.

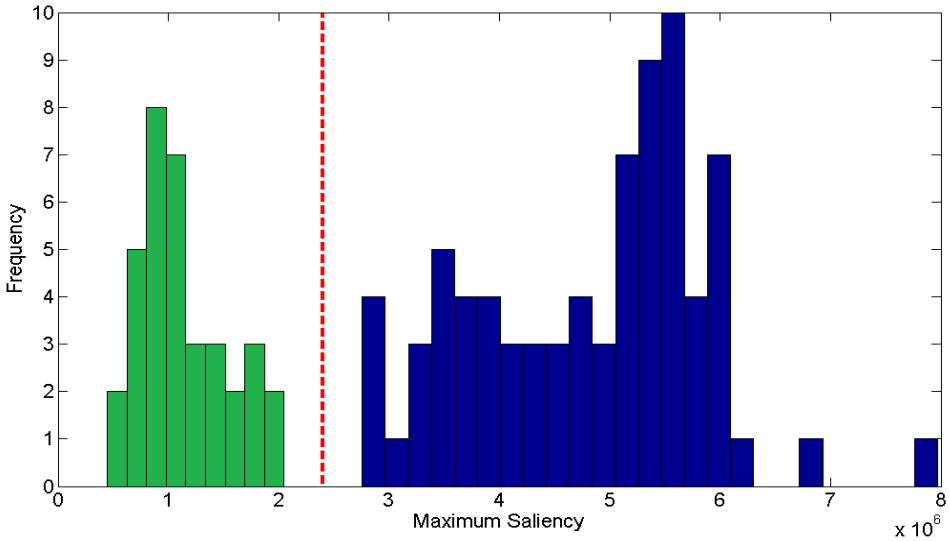


Figure 6-5. Histogram of the maximum saliency of 35 clean (green bars) and 77 dirty (blue bars) dishes.

The maximum of the MS of clean dishes is 2.05×10^6 and the minimum of the MS of dirty dishes is 2.76×10^6. As a result, the MS distribution of clean dishes is completely separable

from dirty dishes using the threshold value of 2.405×10^6 (dashed red line in Figure 6-5), which indicates that employing our saliency detection mechanism for dishware inspection yields 100% accuracy. The accuracy of Duong and Hoberock [72] method on the same image dataset was 94%. Clean and dirty dishes with maximum and minimum saliency are shown in Figure 6-6.

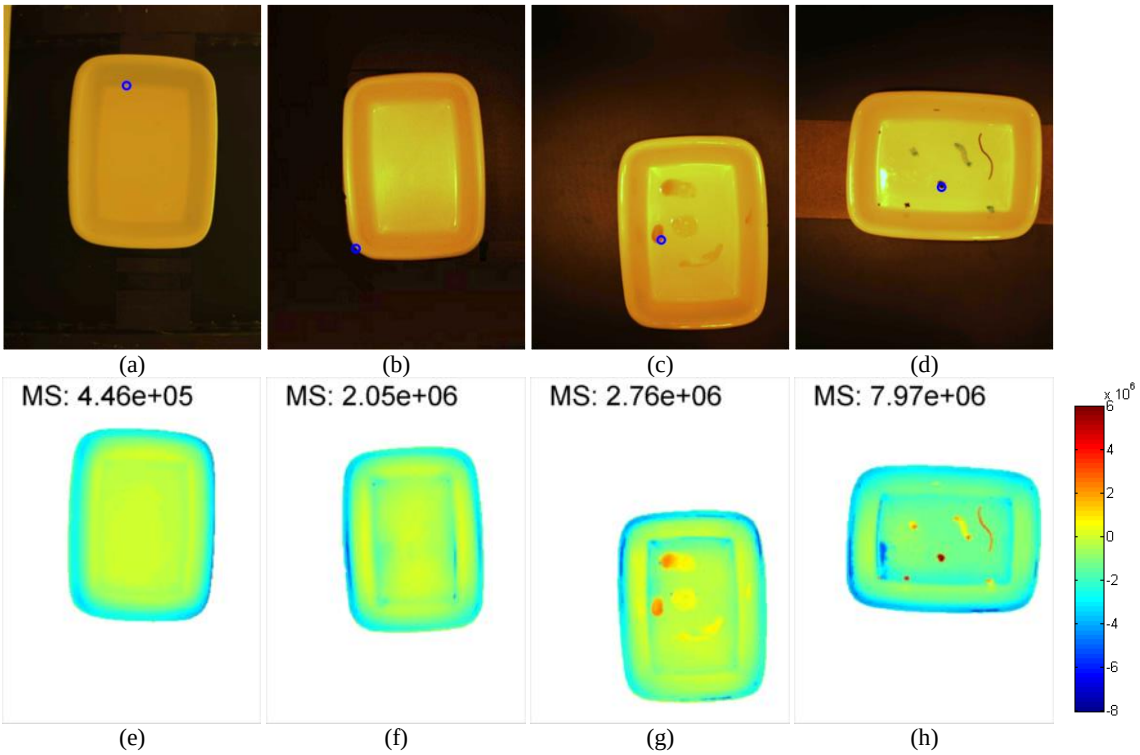


Figure 6-6. Clean dishes with minimum (a) and maximum (b) saliency values; dirty dishes with minimum (c) and maximum (d) saliency values; and their saliency maps computed with EH (e) to (h); the most salient locations are shown with blue circles.

Figure 6-5 shows that neither of the problems mentioned by Duong and Hoberock [72] is a source of difficulties in the application of our method to dish inspection. Both the center-surround mechanism and the normalization method proposed in this report analyze images locally. Accordingly, changes in color and intensity of the image because of the non-flat geometry of the dishes do not affect the dish inspection process. Since the darkness of the gray level image is analyzed, glare does not cause any problem. Also, given that the image is analyzed locally, every dirty spot is compared to its surrounding, and differences in dirty spots made by different food particles do not affect the result. Sensitivity of the dish image to changes in lighting and camera

sensitivity drift also do not cause problems. Moreover, changes in the position of the dish and its orientation do not affect the results.

The average computation time for dish inspection using the proposed method was 4.16 second per dish (using MATLAB® R2010a 32-bit, Image Processing Toolbox V7.0, Window XP Professional, Pentium®4 3.2GHz, 2GB RAM). The average computation time in the Duong and Hoberock [72] method was 1.28 second per dish (using MATLAB® R14, Image Processing Toolbox V5.0, Window Vista, dual core 1.6GHz, 2GB RAM).

6.2 Dirt Detection in Dishware Using Saliency Maps

In this section, we examined only those dishes previously classified as "dirty", as shown in Figure 6-7. Dirty spots appear as local maxima in the saliency map of the dish. Therefore, our saliency detection mechanism can also be applied to dirty point detection. Accordingly, we analyze heights of local maxima in the saliency map to detect dirty spots in the dish.

Other locations in the dish image also appear as local maxima. Since the intensity of the image varies from dish walls to dish floor, in some cases the saliency detection mechanism creates local maxima on dish walls. For the same reason, locations close to specular reflections may also appear as local maxima. Therefore, another criterion is needed to remove false local maxima in the saliency map. Figure 6-7 (a) shows a dish which its saliency map, shown in Figure 6-7 (c), contains both types of false local maxima. A local maximum generated on the dish wall and some local maxima created close to a glare are shown in this figure. Figure 6-7 (b) illustrates how false local maxima can mislead the process of dirt detection; the detected dirty spots are shown by blue circles.

As shown in Figure 6-7 (c), saliency maps can be treated as 3-D surfaces. We found that the local maxima corresponding to dish walls usually have large radii of curvature in the wall direction, while local maxima created by dirty spots usually have relatively small radii of curvature in every direction. Accordingly, to eliminate local maxima generated by dish geometry, the maximum

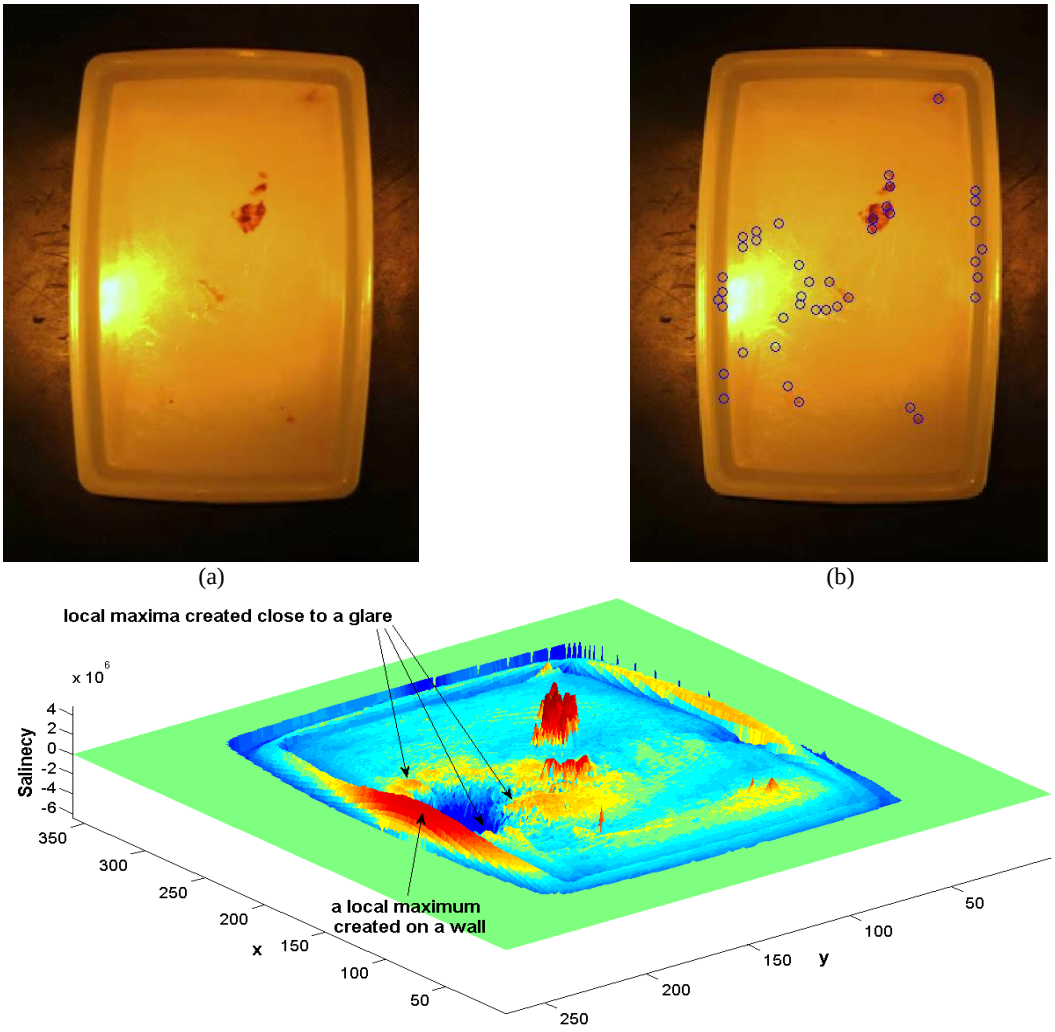


Figure 6-7. A dirty dish, detected dirty points and the dish saliency map.

radius of curvature of the saliency map at each local maximum is checked. Local maxima with maximum radius of curvature greater than a threshold are considered as false detections. On the other hand, local maxima created close to glares usually have short heights and can often be eliminated by thresholding.

Figure 6-8 (a), (b) and (c) show 3 different dishes with different dirty spots; (d), (e) and (f) show the corresponding 2-D saliency maps; and (g), (h) and (i) show the corresponding dirty spots circumscribed by circles found by our method.

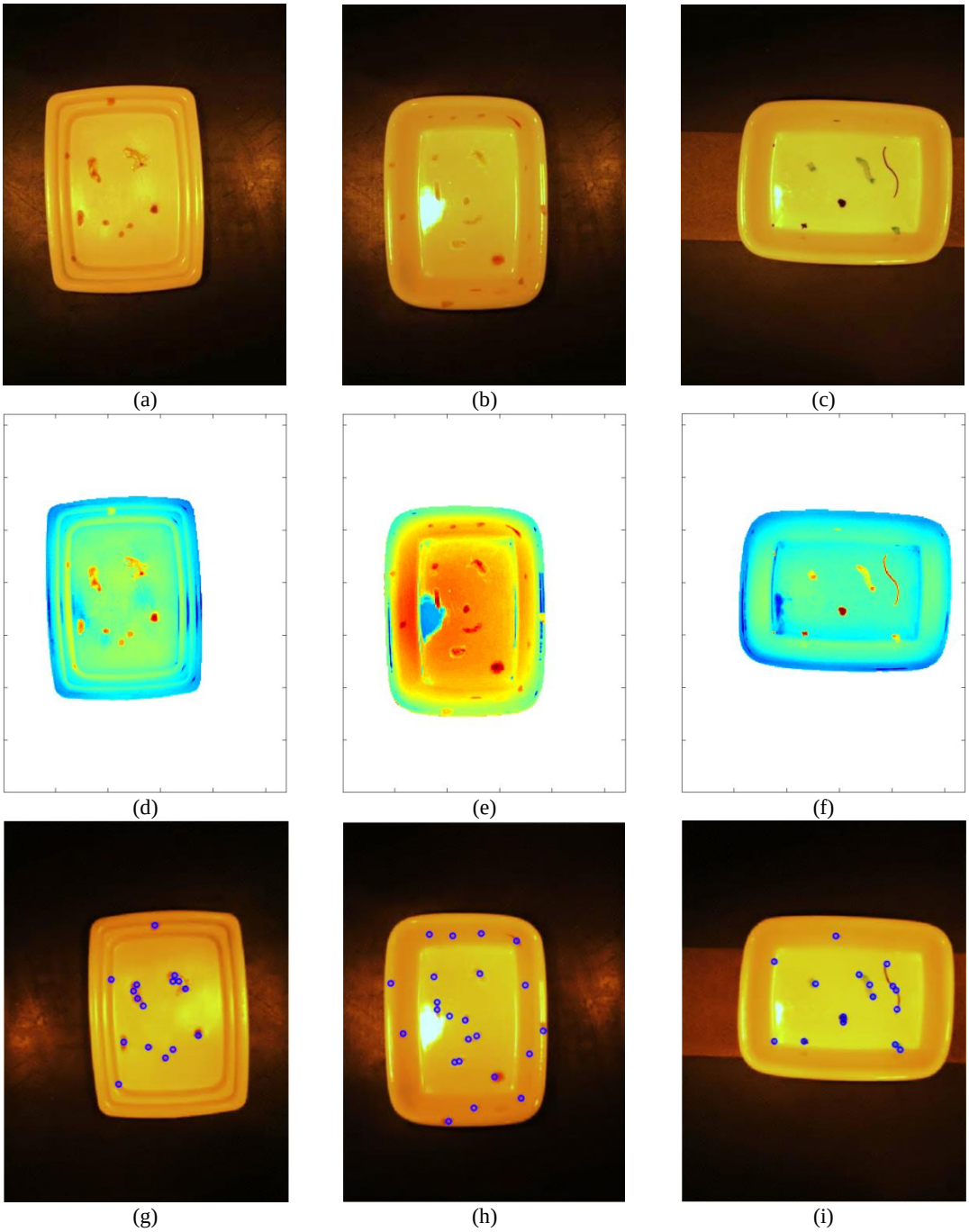
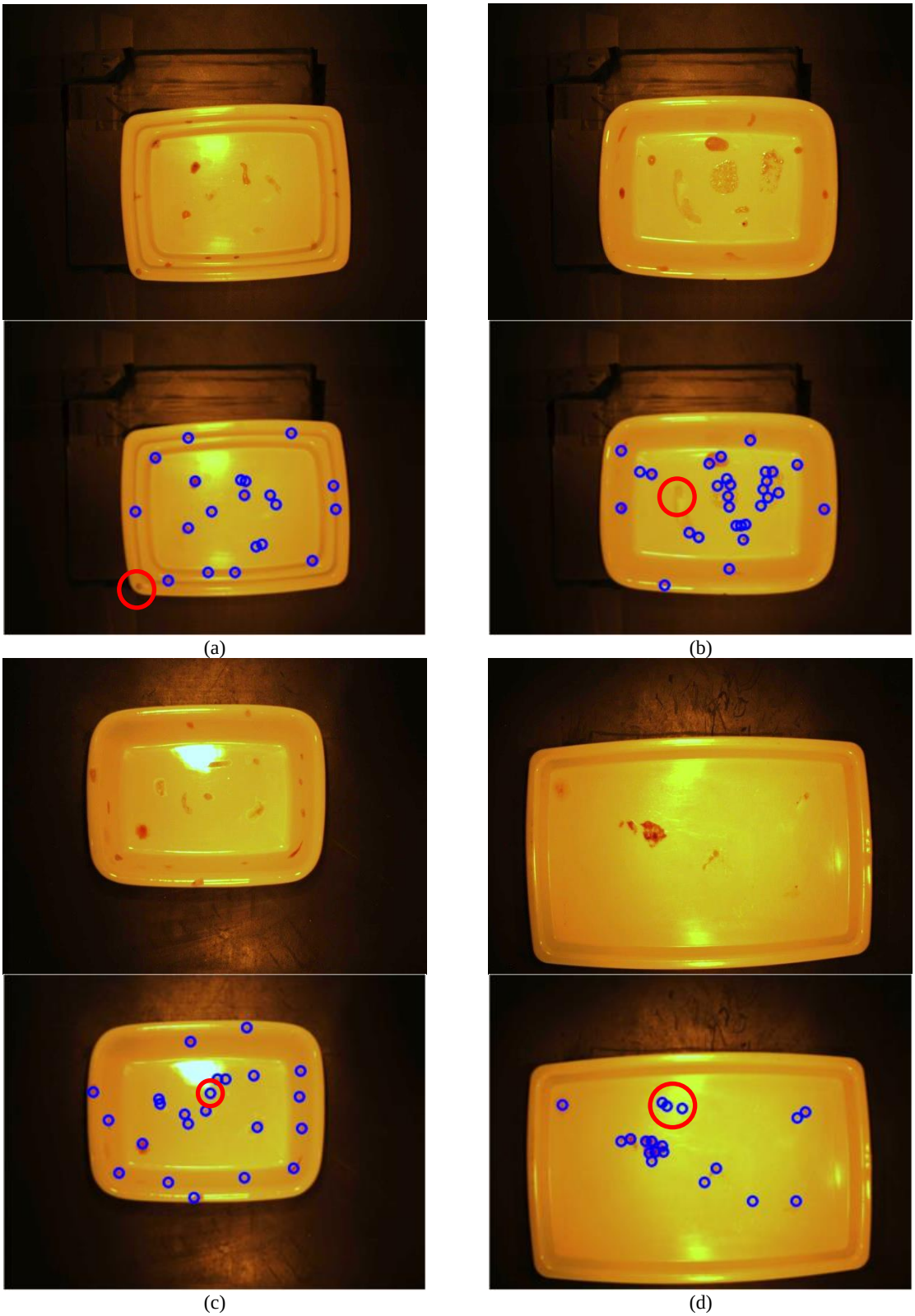


Figure 6-8. Dirt detection using saliency maps.

We note by comparing Figure 6-8 (g), (h), and (i) with the corresponding Figure 6-8 (a), (b), and (c) that the dirt detection algorithm is quite successful in finding dirty points in dishware. This method was applied to 77 dirty dishes in the dataset. There are 799 dirty points in these dishes, and our new method detects 764 (95.6%) of them. Most of the missed dirty locations correspond to bright dirty spots that have intensities close to the intensity of the dish floor or dirty spots on

the dish walls and edges, see Figure 6-9 (a) and (b). We expect that performance of the dirt detection algorithm could be increased using color feature spaces. The method did produce 98 false dirt detections, which mainly corresponded to locations close to glare, as shown in Figure 6-9 (c) and (d). The number of false dirt detections in Duong and Hoberock [72] is not reported directly, but their method produced 12% false alarms, which means a clean dish was classified as dirty. To eliminate specular reflections created on the dish surface in images, we suggest taking dish images in an enclosure with uniform illumination. This may reduce the number of false dirt detections and improve the performance of the algorithm.



(a) (b)

(c) (d)

Figure 6-9. Missed dirty locations (a) and (b); false alarms (c) and (d) shown with red circles.

CHAPTER VII

7 CONCLUSION AND FUTURE WORK

In this report, characteristics of the saliency mechanism of the human vision system (HVS) were presented, together with studies carried out to introduce computational principles of this mechanism. Models of the saliency mechanism of the HVS can be employed in different applications of computer vision. Most of the computer vision algorithms depend upon scanning a scene from left to right or top to bottom to locate objects of interest. Saliency mechanisms offer a reasonably fast method to find locations in the scene that contain key information and may include the object of interest. Applications of the saliency mechanisms comprise, but are not limited to, automatic target detection, robotics, image and video compression, and advertising.

In this study, the general architecture of models of the HVS saliency mechanism was presented and some of the well-known models are illustrated. A new method, called the EH method, for normalizing feature saliency maps in saliency detection mechanisms was introduced, which can be applied to outdoor images as well as indoor images. The new method outperformed some normalization methods in the literature in both promoting regions of the feature saliency map that contain salient locations and suppressing regions which contain little or no useful information. The normalization method is relatively fast, computationally, and has only one free parameter to be determined.

Several saliency map comparison metrics were explained in Chapter III. The fact that results from different metrics vary widely in evaluating models is also shown. It is important to know which models perform the best in mimicking the saliency mechanism of the human visual system. It is important to identify best saliency comparison metrics in order to accurately compare different saliency models. A novel procedure was proposed for evaluating metrics for comparing saliency maps using a database of human fixations on approximately 1000 images. This procedure was then employed to identify the best metrics. Two metrics were found to be the best, namely *NCC* (the normalized value of the correlation coefficient between two maps) for situations in which histogram matching of the predicted saliency map to the reference maps is not used, and *Score* for situations in which such histogram matching is used. These metrics produce minimum misclassification error on discriminating human saliency maps from random saliency maps. Interestingly, two commonly used comparison metrics, ROC area and Kullback-Leibler divergence, were ranked 5^{th} and 9^{th} among 9 metrics.

In Chapter IV, all visual saliency models were applied to a database of 1003 natural images and a database of 54 synthetic images. The resulting saliency maps were compared with reference human saliency maps using all metrics. The Graph Based Visual Saliency (GBVS) model significantly outperformed other visual saliency models on mimicking human observers' behavior on both databases. Afterwards, optimum levels of blurring and center-bias were studied for all visual saliency models. GBVS remained the best model for the database of the natural images, and Context-Aware Saliency Detection (CASD) was ranked first on synthetic images after applying the optimum level of blurriness.

As shown in Chapter IV, although much work has been done on modeling the bottom-up saliency mechanism, saliency maps constructed with existing methods often extract unimportant locations in the display, along with actual salient locations. Two of the factors pointed out in the introduction as the likely causes of this problem are:

(1) Predetermined parameters. For example the center-surround mechanism is applied with predetermined sets of radii, while the radii should be determined based on the properties of the display.

(2) Only one set of parameters is employed to analyze the entire image. We believe that to effectively imitate the HVS saliency mechanism, different parts of the visual scene should be analyzed with different sets of parameters (analyzing the image locally).

A new visual saliency model called EMHO developed from GBVS was introduced in Chapter V to address these problems. An iterative approach was used to find the best distributed values of **σ** (the only parameter in the algorithm) for each image. Since selecting different parameters for each pixel of the image is cumbersome, to reduce the number of parameters, we employed superpixels in this new approach. The algorithm automatically selects the best values for **σ** for each superpixel in the image. The average *Score* for EMHO is higher than that for GBVS, and the number of maps classified as random maps is significantly less.

In Chapter VI, the EH and GBVS saliency mechanisms were employed in two dish inspection algorithms. The method based on EH performed better than the dish inspection algorithms in the literature. In this algorithm, the height of the global maximum of the saliency map was analyzed for dish inspection, and the local maxima of the saliency map were tested to find dirty spots. This method was applied to 112 dish images, with experimental results showing 100% and 95.6% accuracies in discriminating clean from dirty dishes and dirty spot detection, respectively.

7.1 Recommended Future Work

As shown in Chapter IV, the EMHO algorithm with the best distributed **σ** found manually for each image performs very well, with only 1% of its maps classified as random saliency maps based on *Score*. However, in general, with the iterative approach used, 26.7% of the maps are classified as RSM. For future work, one might find a better method to estimate the parameters a

and b in the formula for $\boldsymbol{\sigma}$ (instead of using the average of as and bs), or find a better approach to find the distributed $\boldsymbol{\sigma}$.

In the EMHO method, all feature saliency maps are added together to construct saliency maps. In some cases, each feature saliency map identifies some of the salient locations of the image properly, but these points are lost during the process of calculating the saliency map. One might find more thoughtful approaches to combine feature saliency maps into a saliency map. This can be done by finding criteria to distinguish between good and bad feature saliency maps and weighting feature saliency maps globally or locally before adding them together.

There is a need for a database of natural images which contains only low-level feature attributes. Most of the images in the existing eye tracking databases contain high-level information, e.g. human faces, texts, and animals, as well as low level information. We believe that high-level information in an image misleads the process of designing, training or optimizing bottom-up visual saliency models. It has been shown in Chapter V that even in an image containing only low level information, human observers tend to look at the central regions of the display. Accordingly, all databases in the literature are highly center biased. We suggest using larger images with many details shown in larger displays. This forces observers to move eyes and head more frequently. We also suggest recording many more fixations per observer (e.g. 20), rather than only 5.

8 References

[1] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research*, vol. 40: p. 1489:1506, 2000.

[2] D. Gao, *A discriminant hypothesis for visual saliency, computational principles, biological plausibility and applications in computer vision*, Electrical Engineering, University of California: San Diego, 2008.

[3] M.D. Levine, *Vision in man and machine*, McGraw-Hill Book Company, 1985.

[4] T. Pavlidis, "Can machine vision be helped from insights into human vision?" SPIE. p. 402:407, 1992.

[5] U. Rajashekar, *Statistical analysis and selection of visual fixations*, University of Texas at Austin, 2005.

[6] 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*, vol. 20(11): p. 1254:1259, 1998.

[7] Z. Li, "A saliency map in primary visual cortex," *Trends in Cognitive Sciences*, vol. 6(1): p. 9:16, 2002.

[8] A.L. Yarbus, *Eye movement and vision*, New York: Plenum Press, 1967.

[9] L. Itti and C. Koch, "Computational modeling of visual attention," *Neuroscience*, vol. 2: p. 194:203, 2001.

[10] S. Grossberg, "How does the cerebral cortex work? Learning, attention, and grouping by the laminar circuits of visual cortex," *Spatial Vision*, vol. 12(2): p. 163:185, 1999.

[11] J. Harel, C. Koch, and P. Perona, "Graph-Based Visual Saliency, in Advances in Neural Information Processing Systems," *MIT Press*, p. 545-552, 2006.

[12] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4: p. 219:227, 1985.

[13] A.M. Treisman and G. Gelade, "A Feature Integration Theory of Attention*," Cognitive Psychology*, vol. 12: p. 97:136, 1980.

[14] L. Itti and C. Koch, "Feature combination strategies for saliency-based visual attention systems," *Journal of Electronic Imaging*, vol. 10(1): p. 161:169, 2001.

[15] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vision Research*, vol. 42 p. 107:123, 2002.

[16] J.M. Wolfe and T.S. Horowitz, "What attributes guide the deployment of visual attention and how do they do it? Nature Reviews," *Neuroscience*, vol. 5: p. 495:501, 2004.

[17] O. Le Meur, P. Le Callet, D. Barba, and D. Thoreau, "A coherent computational approach to model bottom-up visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(5): p. 802-817, 2006.

[18] S. Frintrop, M. Klodt, and E. Rome, "A Real-time Visual Attention System Using Integral Images," in *5th International Conference on Computer Vision Systems*, Bielefeld, Germany, March 2007.

[19] Frintrop, S., E. Rome, and H.I. Christensen, "Computational Visual Attention Systems and their Cognitive Foundation: A Survey," *ACM Transactions on Applied Perception (TAP)*, vol. 7(1): p. 1:39, 2010.

[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*, vol. 8(7), 2008.

[21] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[22] Yu, Y., B. Wang, and L. Zhang, "Bottom–up attention: pulsed PCA transform and pulsed cosine transform", *Coognitive Neurodynamics*, 2011.

[23] P. Bian and L. Zhang, "Visual saliency: a biologically plausible contourlet-like frequency domain approach," *Cognive Neurodynamics*, vol 4, 2010.

[24] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[25] R.C. Gonzalez, R.E. Woods and S.L. Eddins, *Digital image processing using MATLAB*, Gatesmark Publishing, 2009.

[26] G. Chenlei, M. Qi, and Z. Liming, "Spatio-temporal Saliency detection using phase spectrum of quaternion Fourier transform," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[27] C. Guo, C. and L. Zhang, "A Novel Multiresolution Spatiotemporal Saliency Detection Model and Its Applications in Image and Video Compression," *IEEE Transactions on Image Processing*, vol. 19: p. 185:198, 2010.

[28] N.D.B. Bruce and J.K. Tsotsos, "Saliency Based on Information Maximization," *Advances in Neural Information Processing Systems*, vol. 18: p. 155:162, 2006.

[29] N.D.B. Bruce and J.K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Vision*, vol. 9(3): p. 1:24, 2009.

[30] D.J.C.MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2005.

[31] D. Gao and N. Vasconcelos, "Bottom-up saliency is a discriminant process," *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[32] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[33] H. Xiaodi, J. Harel, and C. Koch, "Image Signature: Highlighting Sparse Salient Regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(1): p. 194-201, 2012.

[34] A. Oliva, A. Torralba, M.S. Castelhano and L.M. Henderson, "Top-Down Control of Visual Attention in Object Detection," *International Conference on Image Processing* 2003.

[35] B.W. Tatler, R.J. Baddeley, and I.D. Gilchrist, "Visual correlates of fixation selection: effects of scale and time," *Vision Research*, 45(5): p. 643-659, 2005.

[36] K.A. Ehinger, B.H. Sotelo, A. Torralba, A. Oliva, "Modelling search for people in 900 scenes: A combined source model of eye guidance," *Visual Cognition*, vol. 17: p. 945:980, 2009.

[37] F.H. Hamker, "The emergence of attention by population-based inference and its role in distributed processing and cognitive control of vision," *Computer Vision and Image Understanding*, vol.100 p. 64:106, 2005.

[38] A. Treisman and S. Gormican, "*Feature Analysis in Early Vision: Evidence From Search Asymmetries," Psychological Review*, vol. 95(1): p. 15:48, 1988.

[39] W. Kienzle, F.A. Wichmann, B. Sch¨olkopf, M.O. Franz, *A nonparametric approach to bottom-up visual saliency*. NIPS MIT Press, 2006: p. 689:696.

[40] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," *International Conference on Computer Vision*, 2009.

[41] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42: p. 145:175, 2001.

[42] R. Rosenholtz, "A simple saliency model predicts a number of motion popout phenomena," *Vision Research*, vol. 39(19): p. 3157:3163, 1999.

[43] M. Cerf, J. Harel, W. Einhauser, C. Koch. *Predicting human gaze using low-level saliency combined with face detection*, MIT Press, 2007.

[44] D. Walther, U. Rutishauser, C. Koch and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," *Computer Vision and Image Understanding*, vol. 100: p. 41:63, 2005.

[45] D.G. Lowe, "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, p. 1150:1157, 1999.

[46] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19: p. 1395:1407, 2006.

[47] J.P. Jones and L.A. Palmer, "An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex," *Journal of Neurophisiology*, vol. 58(6): p. 1233:1257, 1987.

[48] J.R. Cavanaugh, W. Bair, and J.A. Movshon, "Nature and interaction of signals from the receptive field center and surround in macaque V1 neurons," *Journal of Neurophysiology*, vol. 88: p. 2530:2546, 2002.

[49] B.A. Steinman, S.B. Steinman, and S. Lehmkuhle, "Visual attention mechanisms show a center-surround organization," *Vision Research*, vol. 35(13): p. 1859:1869, 1995.

[50] ABorji, and L. Itti, "State-of-the-art in Visual Attention Modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[51] D. Gao, V. Mahadevan, and N. Vasconcelos, “On the plausibility of the discriminant center-surround hypothesis for visual saliency,” *Journal of Vision*, 8(7), 2008.

[52] B.W. Tatler, R.J. Baddeley, and I.D. Gilchrist, *Visual correlates of fixation selection: effects of scale and time.* Vision Research, vol. 45(5): p. 643:659, 2005.

[53] N. Ouerhani, R. Wartburg, H. Hugli, and R. Muri, “Empirical Validation of the Saliency-based Model of Visual Attention,” *Electronic Letters on Computer Vision and Image Analysis*, vol. 3(1): p. 13-24, 2004.

[54] Y. Lin, B. Fang, and Y. Tang, “A Computational Model for Saliency Maps by Using Local Entropy,” *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[55] A. Garcia-Diaz, X. Fdez-Vidal, X. Pardo, and R. Dosil, “Saliency Based on Decorrelation and Distinctiveness of Local Responses Computer Analysis of Images and Patterns,” *Springer Berlin / Heidelberg*, p. 261-268, 2009.

[56] V. Mahadevan and N. Vasconcelos, “Spatiotemporal Saliency in Dynamic Scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32(1): p. 171-177, 2010.

[57] L. Itti and P. Baldi, “Bayesian surprise attracts human attention," *Vision Research*, vol. 49(10): p. 1295-1306, 2009.

[58] N. Ouerhani, A. Bur, and H. Hügli, “Linear vs. Nonlinear Feature Combination for Saliency Computation: A Comparison with Human Vision Pattern Recognition,” *Springer Berlin / Heidelberg*, p. 314-323, 2006.

[59] A. Bur and H. Hügli, “Optimal Cue Combination for Saliency Computation: A Comparison with Human Vision Nature Inspired Problem-Solving Methods in Knowledge Engineering,” *Springer Berlin / Heidelberg*, p. 109-118, 2007.

[60] R.J. Peters, A. Iyer, L. Itti, and C. Koch, “Components of bottom-up gaze allocation in natural images,” *Vision Research*, 45(18): p. 2397-2416, 2005.

[61] T. Judd, F. Durand, and A. Torralba, “A Benchmark of Computational Models of Saliency to Predict Human Fixations,” *CSAIL Technical Reports*, 2012.

[62] S.E. Umbaugh, *Computer Imaging, Digital Image Analysis and Processing*, Boca Raton, Florida, CRS Press, 2005.

[63] Sá, J.P.M.d., *Applied* Statistics *Using SPSS, STATISTICA, MATLAB and R*, Springer, 2007.

[64] Johnson, R.A. and G.K. *Bhattacharyya, Statistics: principles and methods*, 4th ed., New York: John Wiley, 2001.

[65] Borji, D.N. Sihite, and L. Itti, “Quantitative Analysis of Human-Model Agreement in Visual Saliency Modeling: A Comparative Study,” *IEEE Transactions on Image Processing*, 22(1): p. 55-69, 2013.

[66] R. Rosenholtz, Y. Li, and L. Nakano, “Measuring visual clutter,” *Journal of Vision*, 2007.

[67] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, X. Su and S. Strunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34(11): p. 2274-2282, 2012.

[68] Levinshtein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson and K. Siddiqi, "TurboPixels: Fast Superpixels Using Geometric Flows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31(12): p. 2290-2297, 2009.

[69] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference on Neural Networks*, Perth, WA, 1995.

[70] *Handbook of Swarm Intelligence, Concepts, Principles and Applications*, ed. B.K. Panigrahi, Y. Shi, and M.-H. Lim. Springer, 2010.

[71] W.F. Abd-El-Wahed, A.A. Mousa, and M.A. El-Shorbagy, "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems," *Journal of Computational and Applied Mathematics*, vol. 235(5): p. 1446-1453, 2011.

[72] T.H. Duong and L.L. Hoberock, "New methods for dishware identification and inspection," *Proceedings of the 6th International Conference on Informatics in Control, Automation, and Robotics*, Milan, Italy, 2009.

VITA

Mohsen Emami

Candidate for the Degree of

Doctor of Philosophy

Thesis: AN IMPROVED SALIENCY MECHANISM FOR COMPUTER VISION

Major Field: Mechanical Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in July, 2013.

Completed the requirements for the Master of Science in Mechanical Engineering at Sharif University of Technology, Tehran, Tehran/Iran in 2008.

Completed the requirements for the Bachelor of Science in Mechanical Engineering at Sharif University of Technology, Tehran, Tehran/Iran in 2005.

Experience:

Research Associate, New Product Development Center, Oklahoma State University; May 2012-Present

Design Engineer/Field Engineer, Deep Offshore Technology (DOT), Tehran, Iran; Jan 2009-Dec 2009.

Design Engineer, Sadrafan Gostar, Tehran, Iran; Mar 2006-Sep 2007.

- Employing Pro/ENGINEER, ANSYS and AutoCAD, designed and fabricated a zero–backlash rotary-axis for a CNC milling machine.

Design Engineer, Paydar Sanat Tiva, Tehran, Iran; Nov 2004-Feb 2006.

- Using Pro/ENGINEER, ANSYS and AutoCAD, designed and built a hydraulic mechanism and assembled it on a truck chassis to make it capable of lifting its cabin.

Member of RoboCup Humanoid Research Team, Center of Excellence in Design, Robotics and Automation/Sharif University of Technology; Apr 2003-Jul 2004.