Are All Pixels Equally Important? Towards Multi-Level Salient Object Detection

THÈSE N^{O} 6700 (2015)

PRÉSENTÉE LE 10 JUILLET 2015

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE D'IMAGES ET REPRÉSENTATION VISUELLE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Gökhan YILDIRIM

acceptée sur proposition du jury:

Dr R. Boulic, président du jury
Prof. S. Süsstrunk, directrice de thèse
Prof. M. Kankanhalli, rapporteur
Prof. A. Baskurt, rapporteur
Prof. R. Hersch, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Serenity to accept the things I cannot change,
Courage to change the things I can,
And wisdom to know the difference.

To my family…

Acknowledgements

My journey began on August 22, 2010, when I arrived in Lausanne on a sunny, Sunday afternoon. Since then, EPFL changed my life in so many positive ways that I did not think was possible. I cannot imagine a better place to work and better people to befriend and to collaborate with. I would like to thank everyone for their support throughout my doctoral studies.

First of all, I would like to thank my supervisor, Prof. Sabine Süsstrunk, for her tremendous guidance and support, with which I was able to immensely improve my academic skills. We started working together on semester projects when I was not affiliated with her laboratory. From then until the end of my Ph.D., she always encouraged me to pursue the topics I enjoy working on. The counseling and the freedom she provided me are the dream of every doctoral student. I am very grateful to her for the wonderful opportunity of working with her.

I would like to thank to Prof. Mohan Kankanhalli and Prof. Debashis Sen for their guidance during and after my summer internship at the National University of Singapore. They encouraged and helped me to pursue the academic directions that turned out to be the skeleton of my thesis.

It was an honor to have a jury of distinguished committee members: Prof. Mohan Kankanhalli, Prof. Atilla Baskurt, Prof. Roger Hersch, and Dr. Ronan Boulic. I am very grateful for them for thoroughly investigating my thesis and for giving me valuable feedback.

Many thanks go to the unsung heroines of our laboratory, our secretaries, Jacqueline Aeberhard, Virginie Rebetez, and Françoise Behn, who were very helpful during my time at EPFL. I also am very grateful to Holly Cogliati-Bauereis, whose cheerful attitude made me forget the stress of pressing deadline; while correcting my English mistakes.

I am very happy to have known my colleagues, who have contributed to both my professional and personal lives in many ways. I am very grateful to have worked together with Dr. Appu Shaji and Dr. Radhakrishna Achanta. I am also very lucky to have very good friends who were or still are in our laboratory: Neda, Albrecht, Cheryl, Dominic, Damien, Nikolaos, Bin, Marjan, Sami, and Gökçen. I would especially like to thank my office-mate Zahra, who was always there for me during the ups and downs of the life of a doctoral student.

My friends have always been a very important part of my life and this did not change in Switzerland. I would like to thank my friends Anil, Emre, Emrah Tas, Ali Galip, Gözen, Cihangir, Damla, Ezgi, Tugba, Ebru, Mustafa, Cansaran, Emrah Bostan, Kerem Kapucu, Zafer, Arda, Eray, Kerem Seyid, Tuna, Yasemin, Gürkan, Seniz, Deniz, Engin, Basak, Ismail, Cem, Meric, Nariye, Murat, Onur, Gözde, Can, Bugra, Burak, Eymen, Tugce, Pinar, Cihangir, and many others for every happy moment we shared together.

Finally and most importantly, I would like to thank my parents and my brother for their unconditional support and love, without which none of this would have been possible. I thus dedicate this thesis to my amazing family.

Lausanne, June 24, 2015

Abstract

When we look at our environment, we primarily pay attention to visually distinctive objects. We refer to these objects as visually important or *salient*. Our visual system dedicates most of its processing resources to analyzing these salient objects. An analogous resource allocation can be performed in computer vision, where a salient object detector identifies objects of interest as a pre-processing step.

In the literature, salient object detection is considered as a foreground-background segmentation problem. This approach assumes that there is no variation in object importance. Only the most salient object(s) are detected as foreground. In this thesis, we challenge this conventional methodology of salient-object detection and introduce multi-level object saliency. In other words, all pixels are **not** equally important.

The well-known salient-object ground-truth datasets contain images with single objects and thus are not suited to evaluate the varying importance of objects. In contrast, many natural images have multiple objects. The saliency levels of these objects depend on two key factors. First, the duration of eye fixation is longer for visually and semantically informative image regions. Therefore, a difference in fixation duration should reflect a variation in object importance. Second, visual perception is subjective, hence the saliency of an object should be measured by averaging the perception of a group of people. In other words, objective saliency can be considered as the collective human attention. In order to better represent natural images and to measure the saliency levels of objects, we thus collect new images containing multiple objects and create a Comprehensive Object Saliency (COS) dataset. We provide ground truth multi-level salient object maps via eye-tracking and crowd-sourcing experiments.

We then propose three salient-object detectors. Our first technique is based on multi-scale linear filtering and can detect salient objects of various sizes. The second method uses a bilateral-filtering approach and is capable of producing uniform object saliency values. Our third method employs image segmentation and machine learning and is robust against image noise and texture. This segmentation-based method performs the best on the existing datasets compared to our other methods and the state-of-the-art methods.

The state-of-the-art salient-object detectors are not designed to assess the relative importance of objects and to provide multi-level saliency values. We thus introduce an Object-Awareness Model (OAM) that estimates the saliency levels of objects by using their position and size information. We then modify and extend our segmentation-based salient-object detector with the Object-Awareness Model (OAM) and propose a Comprehensive Salient Object Detection (CSD) method that is capable of performing multi-level salient-object detection. We show that the Comprehensive Salient Object Detection (CSD) method significantly outperforms the state-of-the-art methods on the Comprehensive Object Saliency (COS) dataset.

We use our salient-object detectors as a pre-processing step in three applications. First, we show that multi-level salient-object detection provides more relevant semantic image tags compared to conventional salient-object detection. Second, we employ our salient-object detector to detect salient objects in videos in real time. Third, we use multi-level object-saliency values in context-aware image compression and obtain perceptually better compression compared to standard JPEG with the same file size.

Keywords: saliency, multi-level, salient-object detection, segmentation, image tagging

Résumé

Lorsque nous regardons notre environnement, nous faisons particulièrement attention aux objets qui sont visuellement distincts. Nous considérons ces objets comme visuellement importants ou saillants. Notre système visuel dédie une grande partie de nos ressources à l'analyse de ces objets. De manière similaire, un ordinateur pourrait effectuer une allocation similaire de ressources grâce à un détecteur d'objets saillants.

Dans la littérature, la détection d'objets saillant est traitée comme un problème de séparation de l'arrière plan de l'image par rapport au premier plan. Cette approche part du principe qu'il n'y a qu'un seul niveau d'importance parmi les objets. Seulement les objets les plus saillants sont considérés comme premier plan. Dans cette thèse, nous discutons cette approche conventionnelle en introduisant une détection d'objets saillants à plusieurs niveaux d'importance. Autrement dit, tous les pixels n'ont pas la même importance.

Les collections d'images couramment utilisées comme modèles pour la détection d'objets saillants contiennent en général qu'un seul objet important par image, ce qui n'est pas adapté à notre cas. Dans le cas d'images naturelles, elles contiennent généralement plusieurs objets. Le niveau d'importance de ces objets dépend de deux facteurs. Premièrement, l'œil fixe plus longuement sur les régions visuellement et sémantiquement intéressantes d'une image. Par conséquent, une différence dans la durée de fixation doit refléter une variation de l'importance de l'objet. Deuxièmement, la manière dont chaque individu perçoit un objet est subjective, il est donc nécessaire que l'importance d'un objet soit mesurée en considérant un groupe de personnes. Autrement dit, un objet est considéré saillant s'il est perçu comme tel par l'attention collective de ces personnes. Pour mieux refléter les différents niveaux d'importance des objets dans un contexte naturel, nous avons collecté de nouvelles images contenant plusieurs objets pour créer une collection de donnée exhaustive d'importance d'objets (Comprehensive Object Saliency). Nous proposons des images contenant des modèles d'importance d'objets à travers des tests qui consistent à suivre le regard des gens ainsi qu'à leur demander de marquer les zones intéressantes.

Par la suite, nous proposer trois détecteurs d'objets saillants. Notre première méthode consiste à utiliser un filtre linéaire de plusieurs tailles, et permet ainsi de détecter des objets saillants de différentes dimensions. La deuxième méthode consiste à utiliser un filtre bilatéral, ce qui permet d'améliorer la première méthode en détectant des régions plus uniformes. Notre troisième méthode segmente l'image par région et utilise des méthodes d'apprentissage automatique par ordinateur qui permet d'être plus robuste au bruit et aux textures. Cette dernière méthode est la plus performante sur les bases de données existantes comparées à d'autres méthodes et aux méthodes de pointe.

Les méthodes de pointe en matière de détection d'objets saillants ne font pas la différence entre l'importance relative de différents objets afin de calculer leur valeur à plusieurs niveaux. Dans ce but, nous proposons un modèle qui prend en compte l'importance relative d'objets (Object-Awareness Model, OAM) en considérant leur position et leur taille. Nous introduisons ensuite une extension de notre détecteur basé sur la segmentation (Comprehensive Salient Object Detection, CSD) capable de détecter la saillance d'un objet sur plusieurs niveaux. Nos résultats montre une nette amélioration de cette méthode en comparaison aux méthodes de pointes.

Nous démontrons ensuite l'application de notre détecteur agissant comme un préprocesseur pour trois différentes applications. En premier lieu, nous montrons que la détection de saillance à plusieurs niveaux se montre plus appropriée que les méthodes traditionnelles pour l'annotation d'une image à l'aide de mots-clés. Deuxièmement, nous utilisons notre détecteur pour faire de la détection en temps réel dans une vidéo. Troisièmement, nous montrons que notre détecteur peut améliorer la compression d'image en considérant les régions d'intérêts en termes de qualité pour une taille d'image similaire à un fichier JPEG.

Mots clefs : saillance, multi niveaux, détection d'objets saillants, segmentation, annotation d'image

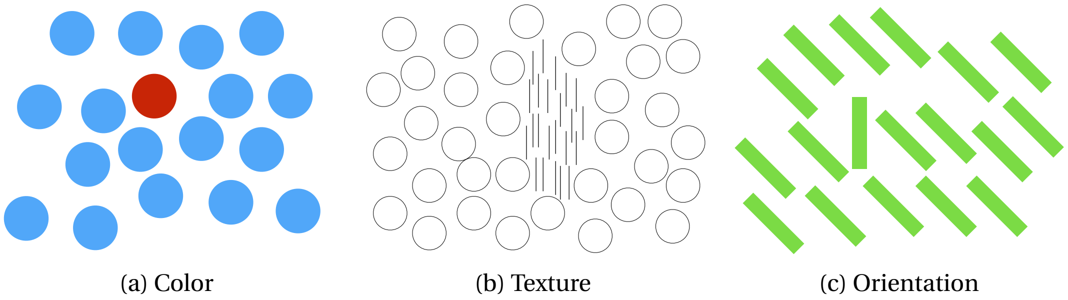
Contents

List of Figures

List of Tables

1 Introduction

While sensing our surroundings, we primarily focus our attention on distinctive stimuli that are often referred to as being "salient". On a very broad perspective, a salient stimulus is an entity that stands out relative to its neighbors. The aspect of saliency can have different forms, such as haptic, aural, and visual. A rough patch on a smooth surface creates a pop-out feeling, making it salient to touch [1]. A distinctive and representative part of a song, such as the chorus, can be salient in an aural sense [2]. A visual stimulus is salient if it has a striking visual quality compared to its surrounding area. For example, as illustrated in Figure 1.1, color, texture, and orientation can be a distinctive factor and grab our attention [3].



(a) Color

(b) Texture

(c) Orientation

Figure 1.1: Visual saliency can originate from different sources including but not limited to (a) color, (b) texture, and (c) orientation. Something that has a distinctive quality compared to its surroundings is called "salient".

When we look at an image, our eyes usually fixate on salient regions, one at a time. Our visual system senses the incoming light at the fovea: the central two degrees of the visual field that is responsible for sharp vision and that consists of color-sensitive cone cells [4]. Although the foveal vision takes up 1% of the retinal space, it occupies 50% of the processing resources of the visual cortex in the human brain [5]. This shows that the visual cortex performs an asymmetric resource allocation to the salient regions of an image and subjects only these regions to more comprehensive analysis [6–8], because exhaustive identification of visual content is prohibitive, even for the human brain [9].

The main endeavor of computer vision algorithms is to duplicate the abilities of the Human Visual System (HVS) by digitally processing visual data. As far as the computational capacity of modern computers is concerned, these algorithms are subject to a processing limitation that is tighter than that of the human brain. Inspired by the saliency-detection and resource-allocation mechanisms in humans, researchers have developed **salient-object detection** methods that can automatically find the objects of interest in an image. These methods are crucial to overcoming the processing limitations in computer vision tasks and have been employed as a pre-processing step.

In recent studies, salient-object detection has been considered as a binary, foreground versus background segmentation problem. The well-known salient-object detection datasets, such as MSRA-1000 [10], SED-100 [11], and SOD [12], have mostly a single object per image, and objects are assumed to be equally salient as shown in Figure 1.2. Moreover, the state-of-the-art salient-object detectors [10, 13–20] are designed to highlight a single salient object per image and do not quantify the importance of these objects. In contrast, a large number of natural images contain multiple salient objects that have different levels of importance.

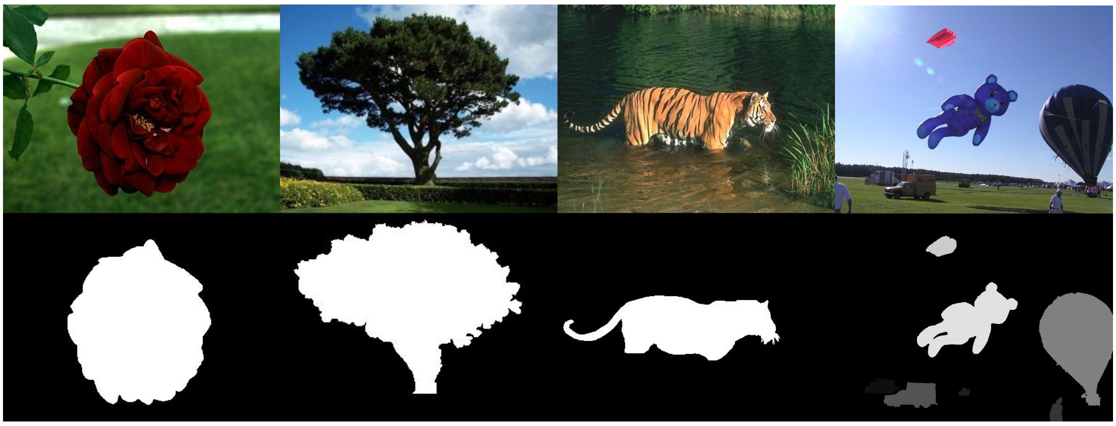

(a) MSRA-1000 (b) SED-100 (c) SOD (d) Our Approach

Figure 1.2: (a)-(c) Sample images from the well-known salient-object detection datasets and their corresponding salient objects. (d) Our approach to salient-object detection, where objects are not equally important, i.e. saliency is multi-level.

In this thesis, we show that all objects are **not** equally salient and introduce **multi-level** salient-object detection. Here, the term multi-level refers to the multiple levels of object saliency in a single image as shown in Figure 1.2(d), as opposed to the binary approach in Figure 1.2(a),(b), and (c). The well-known image datasets [10–12] and the state-of-the-art salient-object detectors [10, 13–20] have investigated object saliency by focusing mostly on simple images with a single salient object and thus, they have overlooked the varying importance of objects. The main objective of this thesis is to overhaul this conventional methodology for salient-object detection by fulfilling the following four goals:

Goal # 1 The well-known datasets [10–12] do not sufficiently represent the variety in natural images. The majority of the images in these datasets contain a single object. In addition, object saliency is assumed to be binary, i.e., objects are either salient or non-salient. On the contrary, natural images include multiple objects with multiple levels of saliency. The variation in object saliency depends on two major factors that are related to the Human Visual System (HVS). First, humans tend to fixate longer on visually and semantically informative objects. In other words, fixation duration is related to the importance or saliency level of an object. Second, human perception is subjective, thus unbiased saliency level of an object should be measured by averaging the perception of a group of people that can be referred to as collective human attention. In order to investigate these factors, our first goal is to accurately represent natural images and to show that object saliency is multi-level. We collect new images containing multiple objects and form our Comprehensive Object Saliency (COS) dataset. We then perform subjective eye-tracking and crowd-sourcing experiments. The experimental data is used to measure the effect of fixation duration and collective human attention on multi-level object saliency.

Goal # 2 The state-of-the-art techniques [10, 13–20] employ heuristic visual features for salient-object detection. In addition, they are not designed for detecting multiple salient-objects in an image. Our second goal is to design a non-heuristic salient-object detector that can find multiple salient-objects in an image. We propose three methods that successively solve more challenging problems of salient-object detection. Our last method is machine-learning and segmentation based. It avoids hand-crafted salient-object detection rules by learning the relationship between visual features and object saliency. Moreover, image segmentation simplifies the detection of multiple objects. We show that our method outperforms the state-of-the-art methods on the well-known datasets.

Goal # 3 The state-of-the-art salient-object detectors are not designed to provide multi-level object-saliency values. Our third goal is to create an accurate multi-level salient-object detector. We thus introduce an Object-Awareness Model (OAM) that estimates the saliency levels of objects. We incorporate the OAM into our segmentation-based salient-object detector and propose a Comprehensive Salient Object Detection (CSD) method. This method is capable of determining the saliency levels of multiple objects. We show that our CSD method significantly surpasses the state-of-the-art methods on our COS dataset.

Goal # 4 Salient-object detection is a pre-processing step to an image processing or a computer vision algorithm. Our fourth goal is to show the benefits of salient-object detectors in various tasks. In our thesis, we exhibit three applications. First, we determine the semantic labels of images by using multi-level salient-object detection prior to object recognition. We show that estimating the saliency value of an object provides more relevant semantic image tags compared to the conventional salient-object detection. Second, we demonstrate our salient-object detector on videos by finding salient objects in real time. Third, we use saliency levels of objects and compress images in a content-aware fashion. We show that our approach yields visually better compression compared to a standard JPEG file with the same file size.

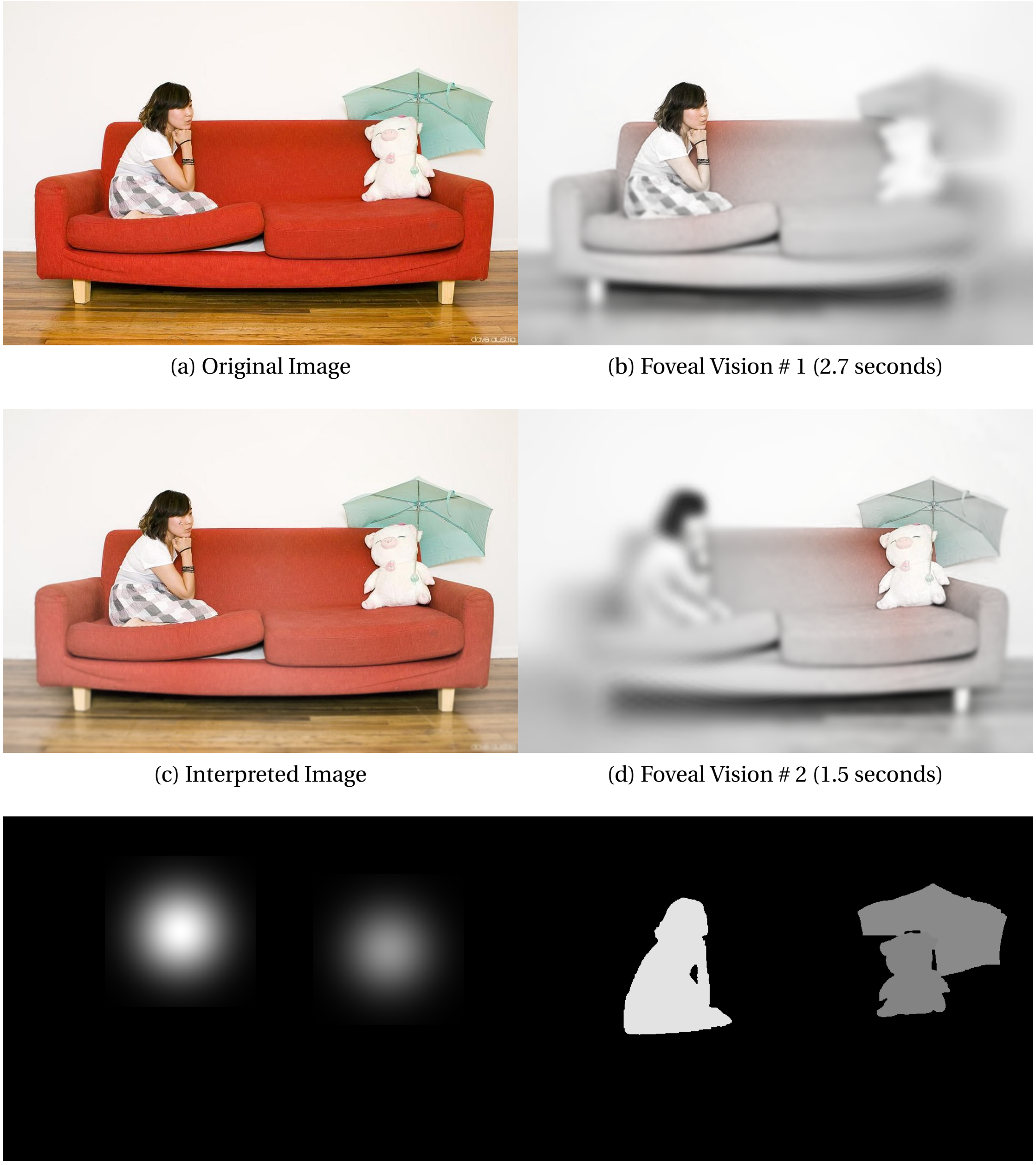
We explain the remaining sections of this chapter in three parts. First, we justify the necessity of multi-level object saliency by discussing two key factors on the HVS: duration of eye fixation and collective human attention. We then discuss the significance of multi-level salient-object detection in computer vision applications. Finally, we outline each chapter in detail.

1.1 Duration of Eye Fixation

The size of the foveal region, which is responsible for acute vision, is limited to two degrees [4]. Therefore, when we look at a still image, such as Figure 1.3(a), we need to scan it -one object of interest at a time- by moving our eyes and head. Every time we fixate our eyes on an object, our fovea only senses a small, sharp, and colorful region at the center of the gaze and a large, blurry, and pale surrounding, namely peripheral, region as simulated in Figure 1.3(b) and (d). Our brain then integrates multiple foveal and peripheral visions, and saccades[1] over time and interprets them as a sharp, colorful still image as illustrated in Figure 1.3(c). Foveal and peripheral visions of human attention are discussed in various studies under different names, such as center-surround vision in [3, 6, 21] and focus-fringe vision in the Zoom-Lens Model [22].

According to Henderson et al. [23] and Yarbus [24], the fixation duration is longer for visually and semantically informative image regions, i.e. important regions. These durations have already been used as a measure of quantifying saliency through eye-fixation datasets [25–28]. When vision is integrated over time, the difference in fixation duration of two objects indicates that one object is more important or salient than the other. In Chapter 3, we conduct a subjective experiment, where we show various natural images to human subjects and track their eye fixations. When the subjects looked at the image in Figure 1.3(a), they fixated on the woman and the toy for 2.7 and 1.5 seconds on average, respectively. These fixations are integrated over time and are represented as an "eye-fixation map" in Figure 1.3(e). Here, the intensity is linearly correlated to the average time that was spent on that pixel position. The differences in fixation durations should be reflected to salient-object detection as shown in Figure 1.3(f). Here, similar to eye-fixation maps, intensity encodes saliency level. In Chapter 3, we introduce a new image dataset that takes duration of eye fixation into account.

[1] Rapid eye movements between two fixations



(a) Original Image

(b) Foveal Vision # 1 (2.7 seconds)

(c) Interpreted Image

(d) Foveal Vision # 2 (1.5 seconds)

(e) Eye-Fixation Map

(f) Multi-Level Object Saliency

Figure 1.3: When we look at (a) an image, due to the limited size of the fovea, we focus on the objects (b) and (d) one at a time. (c) If we integrate the foveal and peripheral visions over time, we perceive the important or salient regions of the image. The time-integrated vision can be represented by (e) an eye-fixation map. The differences in this map should be reflected in (f) salient-object detection.

1.2 Collective Human Attention

Visual perception is subjective. The contribution of each low-level factor, such as color and texture, to whether an object is salient or not depends on the observation goal and

and the observer [29]. In this regard, the perception of a single person is not sufficient to accurately quantify the multi-level saliency value of an object, both statistically and psychophysically. Instead, we intend to investigate "collective human attention", i.e. the average attention of a group of people, as a method for quantifying object importance.

When an image contains a single, prominent object, the ambiguity on whether that object is salient or not diminishes and the problem of subjective human perception is avoided. However, for more complex images with multiple objects, we cannot neglect the variance of individual opinions on object saliency. In order to quantify object saliency in an unbiased manner, we should consider the collective human attention. In Chapter 3, we perform a subjective experiment, where we display natural images, such as in Figure 1.4(a), to a group of people and ask them to click on the objects they notice at first glance. The resulting clicks are shown in Figure 1.4(b). The closer and further away satellite dishes were clicked by 30 and 15 subjects, respectively. If we considered the perception of only a single person, we would deduce that the satellite dishes are equally salient as shown in Figure 1.4(c). Whereas, collective human attention in Figure 1.4(d) suggests that the closer (larger) satellite dish is more important than the other one by a factor of 2 (= 30 subjects / 15 subjects). In Section 1.1, we discussed that, when we integrate human vision over time, the differences among fixation durations imply multi-level object saliency. When we calculate the average or collective human attention, we can again see that objects have multi-level saliency. In Chapter 3, we introduce a new image dataset that takes the collective human attention into account.

1.3 Multi-Level Salient Object Detection in Computer Vision

The recent improvements in distributed computing and machine-learning techniques, such as convolutional neural-networks, were shown to have a potential [30, 31] for advancing the accuracy of computer-vision methods in high-level tasks, such as classifying objects. In order to reach that potential, these methods have two major requirements: a big-data source and resource allocation.

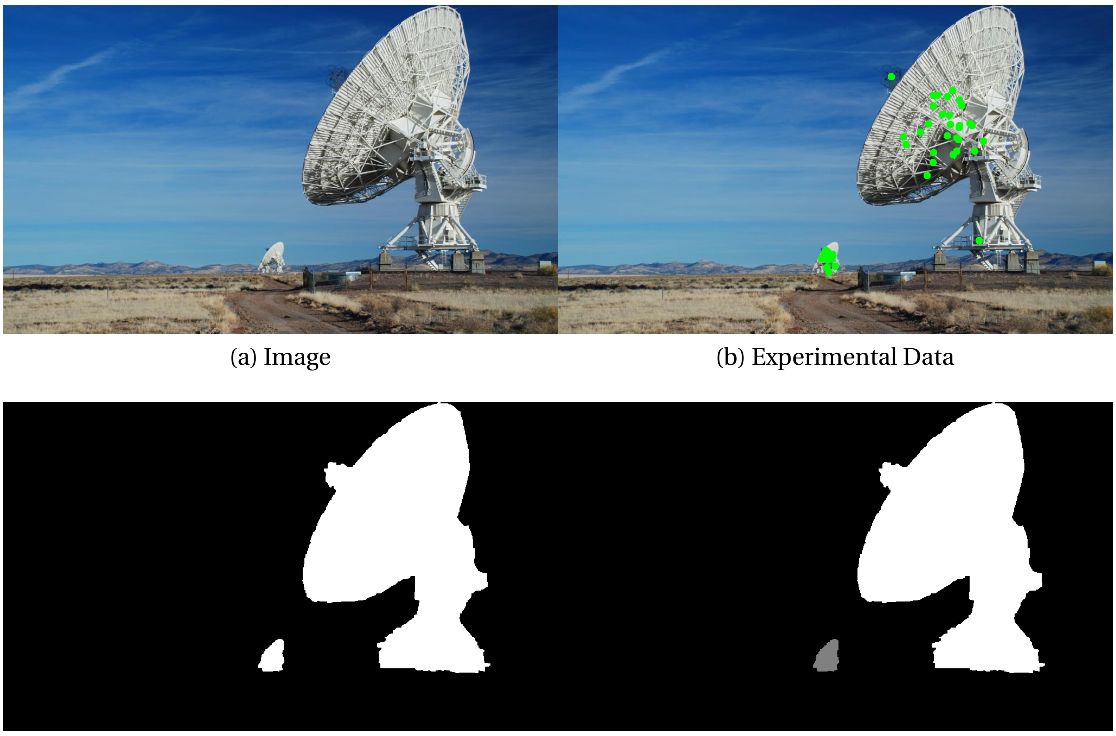
Big-Data Source The improvements in sensor technology [32–34] were the basis for the design of affordable camera phones and smart phones that are equipped with competent digital cameras, hence photos have become a part of our everyday lives. In addition, we can share our photographs with millions of people over the Internet in one touch. The Internet is the largest information source with the easiest access, making it the perfect candidate as a big-data source. According to Merker[2], we upload 1.8 billion images to social networks, such as Facebook[3] Instagram[4], and WhatsApp[5] everyday. Very large and popular image datasets, such as the ImageNet [35] dataset,

[2] http://www.kpcb.com/internet-trends

[3] http://www.facebook.com/

[4] https://www.instagram.com/

[5] https://www.whatsapp.com/



(a) Image

(b) Experimental Data

(c) Single Person

(d) Collective Human Attention

Figure 1.4: When we show (a) an image to a group of people and ask them to (b) click on the objects that they notice (green circles), the attention of (c) a single person cannot explain the subjectivity of visual perception. Whereas, (d) collective human attention reveals the difference in object importances.

have become possible with a combination of image mining and image tagging via crowd-sourcing experiments in the Internet.

Resource Allocation Even though Moore's Law [36] predicts an exponential increase in computational power each year, an exhaustive processing of billions of images with modern computers is still impractical in terms of time and funds. One of the very few structures that is capable of handling a substantial amount of visual data with limited resources is the HVS. In a simple analogy, the visual cortex of the human brain is able to continuously process and selectively record the input from two high-resolution cameras for nearly 16 hours a day. Unlike computers, the HVS does not process a visual input using overlapping windows at multiple scales, which might take a long time and consume too much energy. Instead, it overcomes its processing limitations by quickly and subconsciously detecting the important or salient objects of a visual input and by allocating its computational resources to high-level processing (e.g., object recognition) of important regions as illustrated in Figure 1.5.

As we mentioned in Section 1.1 and 1.2, eye fixation and subjective human atten-

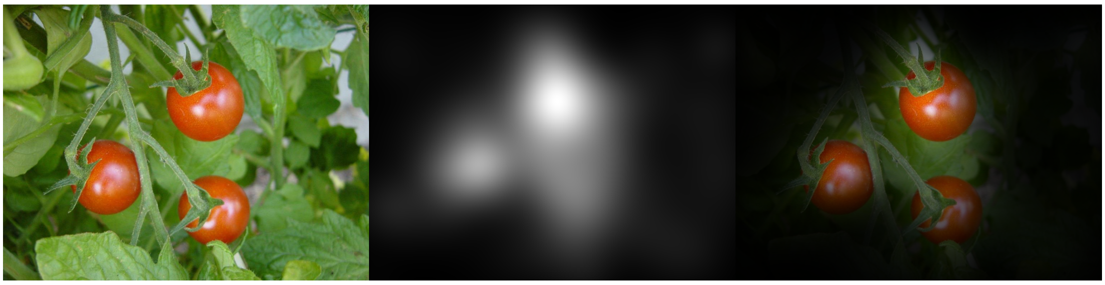

(a) Image (b) Importance Map (c) Resource Allocated Regions

Figure 1.5: The HVS does not exhaustively process an (a) input image. Instead, it effectively generates (b) an importance map and allocate its processing resources to (c) important or salient objects in the image.

tion imply multi-level object saliency. Therefore, the HVS addresses each region of a visual input with a different level of importance and adjusts its resources in order to quickly understand and respond to a visual stimulus. We can perform an analogous resource allocation in computers via multi-level salient-object detection. Modern computers cannot exhaustively process billions of images in a reasonable amount of time. Therefore, we need a pre-processing step, i.e. salient-object detection, similar to the one in the HVS that identifies objects of interest and conveys this information to a computer vision algorithm for resource allocation.

1.4 Outline of the Thesis

The remaining six chapters manifest our work on multi-level object saliency as follows: We begin our thesis by investigating the related work on salient-object detection in Chapter 2. We then present four chapters, each of which is dedicated to a thesis goal. In the final chapter, we summarize the thesis and reveal promising research directions for improving and applying multi-level salient-object detection.

1.4.1 Chapter 2

In this chapter, we review the well-known salient-object datasets and the state-of-the-art salient-object detectors. We then evaluate the performance of these detectors on the well-known datasets.

We explain the data-collection procedures of the well-known datasets. These procedures include the criteria that was used in image selection and subjective experiments. We show that the images and experimental data in these datasets are not sufficient to represent multi-level object saliency.

We group the state-of-the-art salient-object detectors under three saliency mechanisms -uniqueness, spatial variance, and spatial connectivity- based on how they use the low-level visual cues. Uniqueness-based methods measure how rare a visual cue, such as color, is by computing the contrast between a center and a surrounding region. In spatial variance, the methods calculate the spatial distribution of visual cues and regard spatially compact objects as being salient. Spatial connectivity benefits from over-segmentation and graph representation of an image, both of which make a detector robust against object texture.

We evaluate the performance of the state-of-the-art methods on the well-known evaluation datasets. We compare the significance of three saliency mechanisms (uniqueness, spatial variance, spatial connectivity) in salient-object detection using three measurement metrics: precision-recall curves, F-measure, and mean absolute error. We discuss the potential performance of these methods in estimating multi-level object saliency.

1.4.2 Chapter 3

Multi-level object saliency requires experimental data on eye fixation and collective attention. In this chapter, we introduce a new image dataset called Comprehensive Object Saliency (COS), which includes 588 natural images with multiple objects and data from three subjective experiments. We use this dataset to measure saliency levels of the objects in our dataset. A visualization of our dataset is given in Figure 1.6.

For each image in the COS dataset, we obtain multiple modalities of experimental data. In order to investigate the effect of time-integrated foveal vision in Section 1.1, we perform eye-tracking experiments. We show each image in the dataset to a group of people and collect their eye-fixation maps for five seconds.

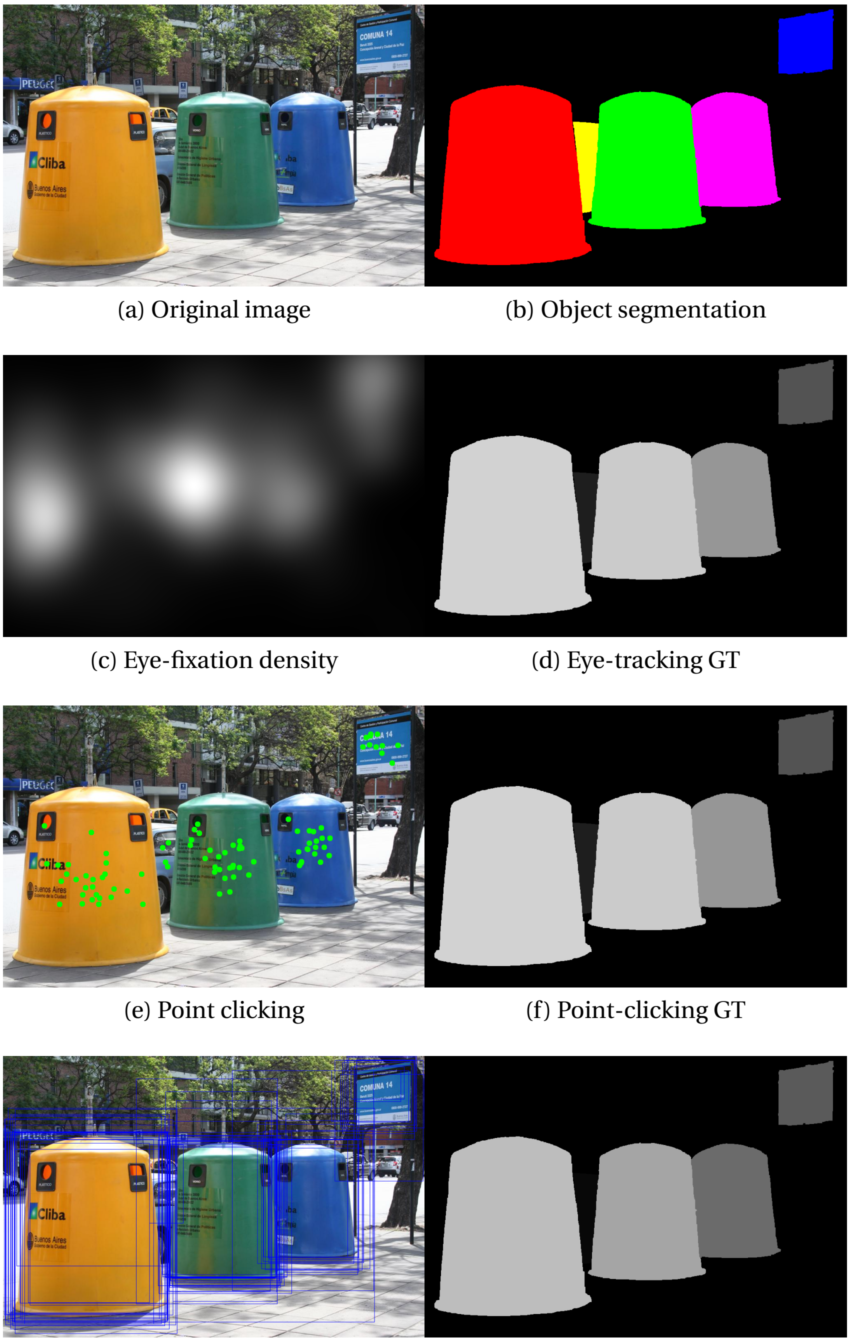
The experimental data we require for collective human attention is acquired via crowd-sourcing experiments. We display images on a web page and ask subjects to perform two tasks. In the first one, we ask them to click on the objects that they notice at first glance. In the second one, we ask them to draw rectangles around the objects.

We analyze the experimental data from the subjects and manually segment out the salient objects that were attended by at least three persons. These segmentations are used to measure saliency levels of different objects. Eye-tracking experiment data is used to measure the time-integrated human vision on still images, whereas the number of clicks and rectangles are used to measure collective human attention. Our main **contributions** in this chapter are as follows:

- We introduce a new image dataset called COS. The COS dataset includes 588 natural images with a total of 2434 objects. For each image, there are manual object segmentations, saliency levels of objects, and data from three subjective experiments
- By using the experimental data of our dataset, we measure the importance values of objects and show that object saliency is multi-level, i.e., all pixels are not equally salient.

1.4.3 Chapter 4

In this chapter, we investigate the basics of identifying distinctive objects in images and by proposing three salient-object detectors. Each method successively solves more



(a) Original image (b) Object segmentation

(c) Eye-fixation density (d) Eye-tracking GT

(e) Point clicking (f) Point-clicking GT

(g) Rectangle drawing (h) Rectangle-drawing GT

Figure 1.6: In order to measure the saliency of an object, for (a) each image in our COS dataset, we collected (c,e,g) three different type of subjective data and (b) segmented the objects. The subjective data and the segmentations are used to generate (d,f,h) multi-level ground-truth maps for each subjective data type (GT: Ground Truth).

challenging problems in salient-object detection. A visual comparison of our methods is given in Figure 1.7.

According to the Zoom-Lens Model [22], human visual attention is divided into a sharp, central foveal region (focus) and a peripheral region (fringe). The size of the focus region represents the trade-off between visual processing area and processing efficiency. There is an analogous trade-off in modern computers, because exhaustive processing of an image is still prohibitive. We use the Zoom-Lens Model to link our three methods with each other and with the existing techniques. We investigate the effect of the size and shape of focus-fringe pairs on object saliency.

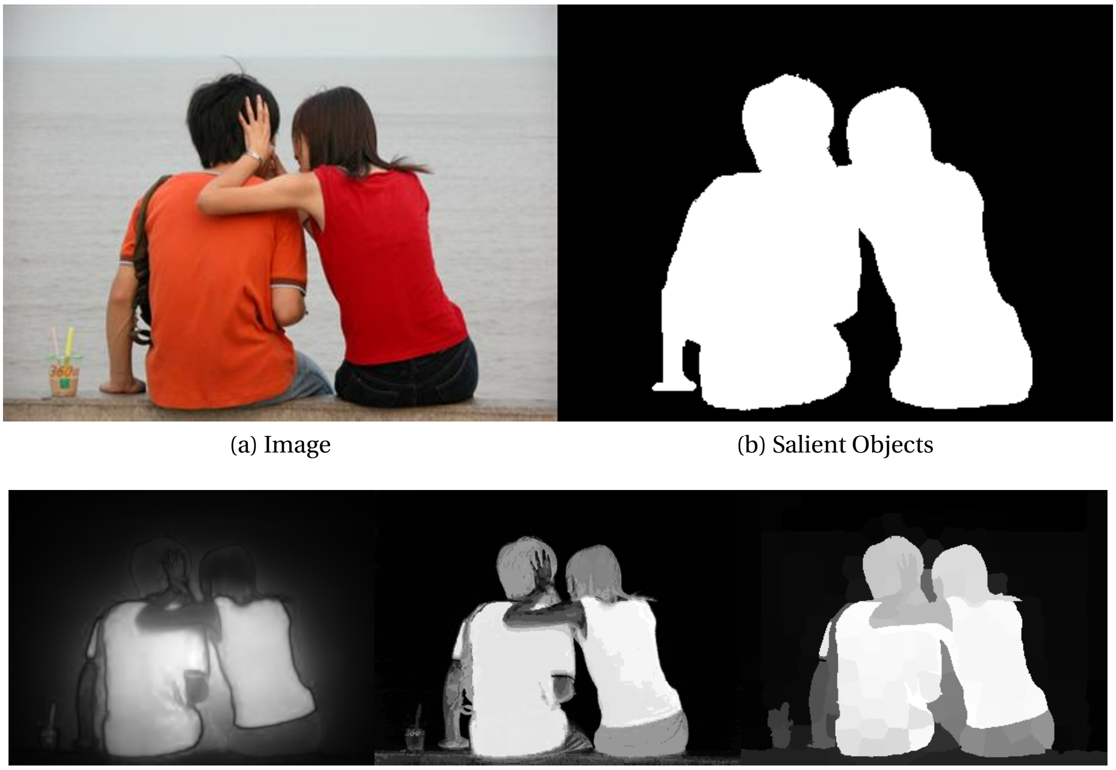
The state-of-the-art salient-object detectors identify an object by looking at the focus-fringe (center-surround) contrast at each position in the image in various scales and shapes. We begin our investigation with a simple, color-contrast based salient-object detector as a baseline. This method finds salient objects via single-scale linear filtering on image colors.

In the first proposed method, we extend the baseline technique to multi-scale filtering. This extension enables us to compute focus-fringe contrast at different object scales and eliminates the need to select an optimal scale-parameter that is discussed in Chapter 2. The main drawback of this approach is that it assigns non-uniform saliency values at pixels within the same object.

In our second method, we propose a solution to non-uniformity by following a bilateral-filtering approach. Bilateral filtering enables us to adapt the shape of the focus region to object edges, which provides homogeneously detected salient objects. Due to the computational complexity of bilateral filtering, we perform several approximations. As a result, we introduce a very fast method that can have real-time applications and it adequately performs on natural images. When computing focus-fringe contrast, this algorithm uses only color, which limits its performance in highly-textured images.

In the third method, we overcome the texture problem by introducing a segmentation-based detector. This method first oversegments an image into edge-aware groups of pixels. It then uses machine learning to model the relationship between visual cues and saliency. Finally, it detects salient objects by employing the saliency model on image segments. Our segmentation-based technique is robust against object texture and effectively adapts the size and the shape of both focus and fringe regions via pixel groups.

We compare our methods with the state-of-the-art techniques and show that our segmentation-based performs the best on the well-known evaluation datasets. Our main **contribution** in this chapter is the segmentation-based salient-object detector, which does not use heuristics and is able to find multiple salient objects. In Chapter 5, we modify this method by incorporating an Object-Awareness Model (OAM). The resulting Comprehensive Salient Object Detection (CSD) method is capable of performing multi-level salient-object detection.



(a) Image

(b) Salient Objects

(c) Multi-Scale Filtering

(d) Bilateral Filtering

(e) Segmentation

Figure 1.7: We propose three salient object detectors that takes (a) an image and finds (b) the salient objects in it. Our methods are based on (c) multi-scale filtering, (d) bilateral filtering, and (e) image segmentation.

1.4.4 Chapter 5

Our focus in this chapter is the generalization of our segmentation-based salient-object detector in Chapter 4 for multi-level object saliency. We add a simple step that can estimate the saliency levels of objects to our segmentation-based technique and obtain the Comprehensive Salient Object Detection (CSD) method. An example output from our method is shown in Figure 1.8.

We show that the position and the size of an object have a significant impact on its saliency level. Therefore, we introduce the Object-Awareness Model (OAM) that has the capability of estimating saliency levels of objects. We use this model in CSD and achieve a multi-level salient-object detector.

We evaluate the performance of CSD and all other methods on the COS dataset by using two performance metrics. The first one measures an absolute error of an algorithm in estimating multi-level object saliency, whereas the second one calculates how well a technique can rank objects with respect to their saliency level, i.e. relative performance. We show that our CSD method significantly outperforms the state-of-the-art methods in both metrics. Our **contributions** in this chapter are as follows:

- We introduce the OAM, which is the step between binary and multi-level salient-object detection.
- We introduce a generalized method, called CSD, that significantly outperforms all state-of-the-art methods in estimating multi-level object saliency.

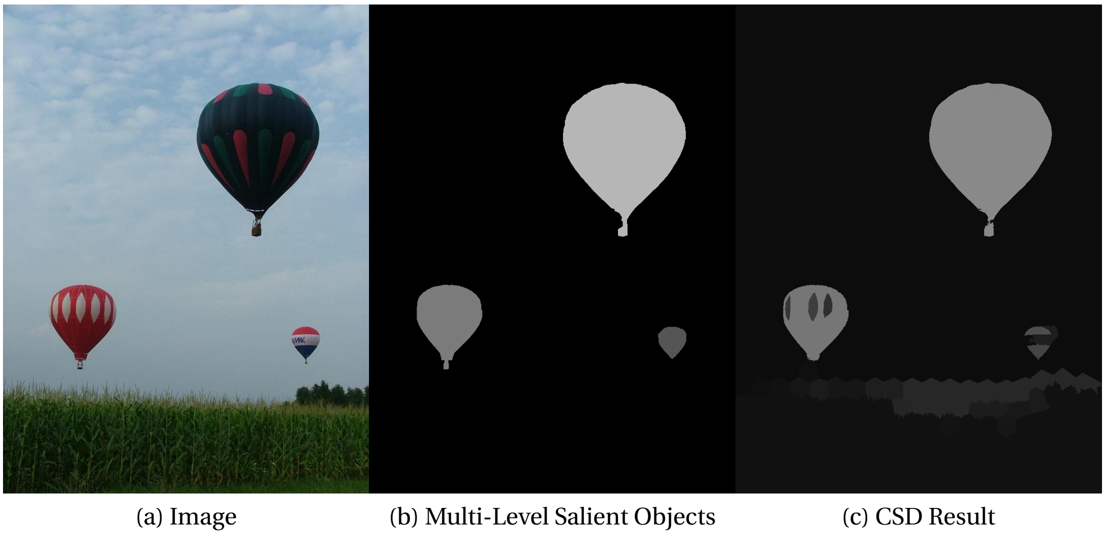

(a) Image (b) Multi-Level Salient Objects (c) CSD Result

Figure 1.8: (a) An image and (b) the corresponding salient objects with their saliency levels. CSD (c) finds the multiple salient objects and estimates their multi-level object saliency.

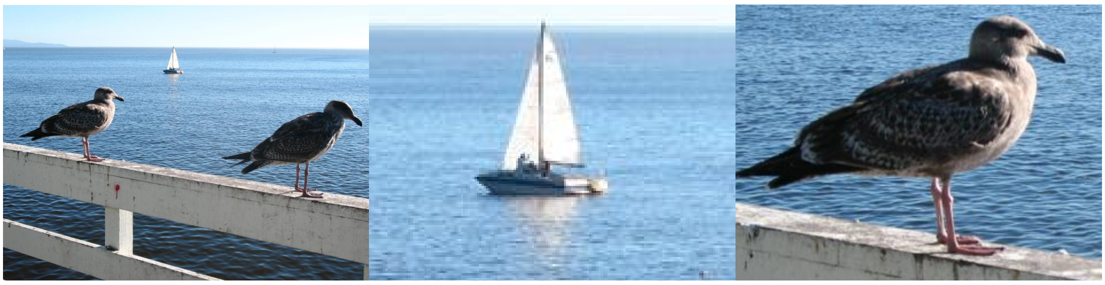
1.4.5 Chapter 6

In this chapter, we use our salient object detectors as a pre-processing step to computer vision and image processing tasks. We present three applications: image tagging, object detection, and image compression.

We use CSD in an image-tagging application. CSD is capable of finding the salient objects and estimating their importance level. Here, we identify the most important object and pass it to an object recognizer in order to retrieve an image label. We show that using the most important object provides more relevant tags than using a random object. This confirms that, compared to conventional binary saliency, our multi-level object-saliency approach is more beneficial to a computer vision task. An example image tagging is illustrated in Figure 1.9.

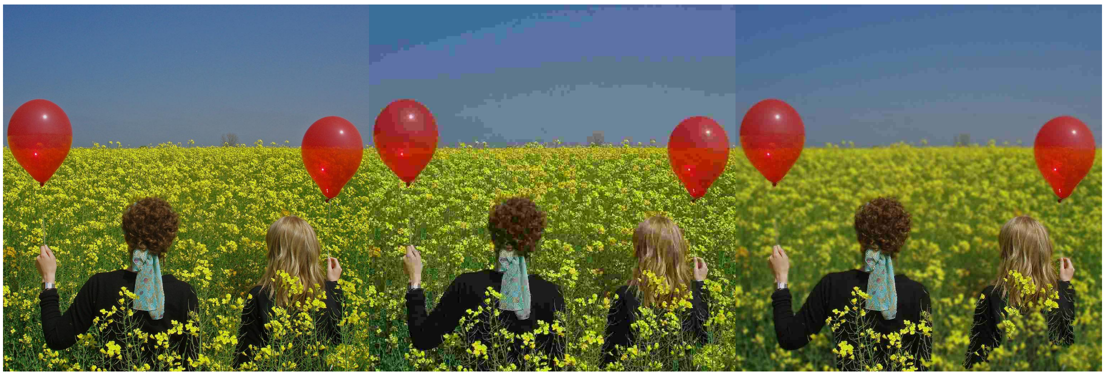
Our bilateral-filtering approach in Chapter 4 is optimized in terms of execution time. In addition, it can detect the position and the size of a salient object. We use these two properties and employ this method in real-time processing of videos. Detected salient objects along with their position and size is illustrated in Figure 1.11.

In the image compression application, we first estimate the saliency levels of the objects by using CSD. We then remove the high-frequency components of the image



(a) Real Tag = "Seagull" (b) Estimated Tag = "Boat" (c) Estimated Tag = "Pigeon"

Figure 1.9: We use our salient object detector to extract salient objects from (a) an image with a label "seagull". (c) The most important salient object leads to a more relevant tag compared to (b) random salient object.



(a) Image (b) Standard JPG (47 KB) (c) Saliency + JPG (47 KB)

Figure 1.10: To save disk space, (a) an image can be compressed by following (b) the JPEG standard. (c) In our framework, in order to preserve the visual quality of salient objects, we sacrifice the background quality and then use the JPEG compression.

blocks based on their multi-level saliency values. This operation helps us trade-off the background quality with the quality of salient-object pixels. Finally, we compress our image with the standard JPEG. Compared to using only JPEG standard, our saliency-based image processing prior to the JPEG compression achieves visually better results (compared in equal file sizes) as shown in Figure 1.10.

Our main **contributions** in this chapter are as follows:

- We use our CSD method in an image-tagging application and show the benefits of multi-level object saliency over binary object -saliency.
- We employ the bilateral-filtering-based method of Chapter 4 for object detection.
- We demonstrate our CSD method on image compression.

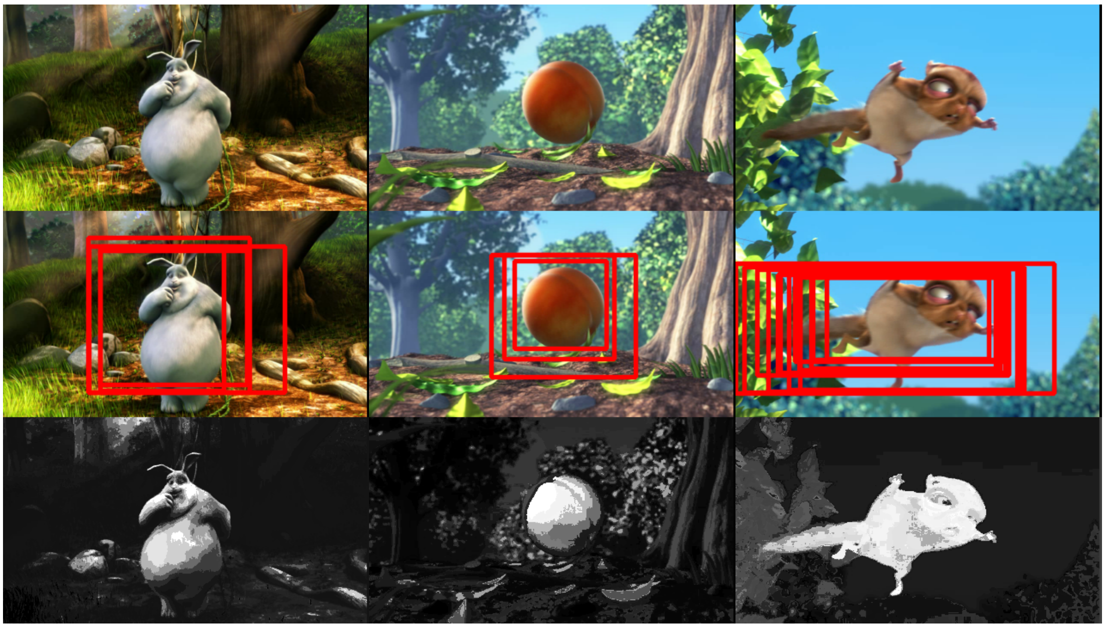


Figure 1.11: Our bilateral-filtering-based approach can find the salient objects and estimate their position and size in under 30 milliseconds, which allows for real-time operation. (First row: video frame, second row: object position and size, third row: detected salient objects)

1.4.6 Chapter 7

In this chapter, we summarize our thesis in detail and discuss its contributions to computer vision. In addition, we propose future research topics and applications, where our contributions can be used as a guideline.

2 Related Work

A salient-object detector identifies all pixels that belong to salient objects. This is usually achieved by generating a full-resolution "salient-object map", where each pixel of the original image is mapped to a saliency value in the interval [0,1]. Conventionally, 0 corresponds to a non-salient pixel, whereas 1 indicates a fully salient pixel. An example of this mapping is illustrated in Figure 2.1(d). Throughout the thesis, we use the same naming convention for various maps. The ideal output of a salient-object detector is called "ground-truth map" as shown in Figure 2.1(b). Each image in the well-known datasets have a corresponding map, where pixels are classified as either salient (saliency value = 1) or non-salient (saliency value = 0). In this thesis, we allow ground -truth maps to have any saliency value in the interval [0,1]. We refer to these maps as "multi-level ground-truth maps", an example of which is shown in Figure 2.1(c).

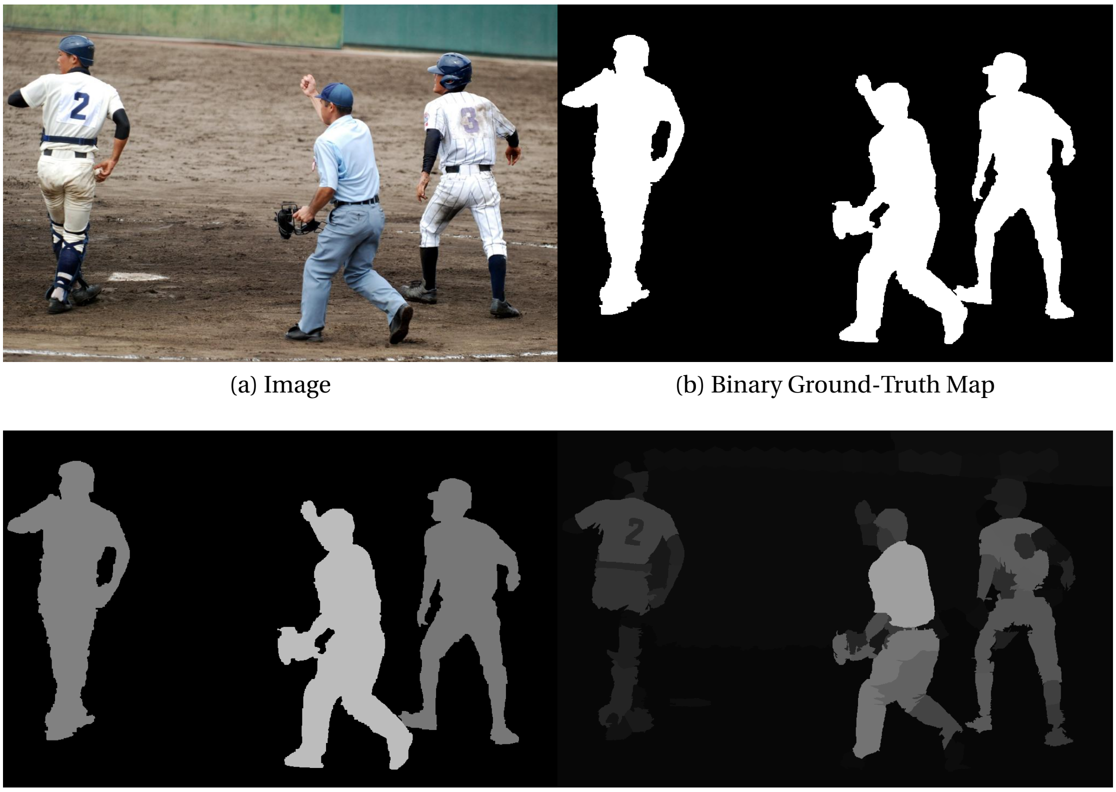
The main objective of this chapter is to present the related work on salient-object detection. This chapter consists of three main parts. In the first part, we review the well-known salient-object datasets and discuss their limitations for computer vision applications. In the second part, we summarize the state-of-the-art methods that are designed for salient-object detection and group them with respect to the way they find salient objects. In the final part, we evaluate the performance of these methods on the well-known evaluation datasets and discuss their potential use in estimating multi-level object saliency.

2.1 Salient-Object Detection Datasets

For development and evaluation purposes, the image datasets, such as MSRA-1000 [10], SED-100 [11], and SOD [12] have been widely used by the state-of-the-art salient object detectors in Section 2.2. Here, we explain the data collection procedures of these datasets and then discuss their limitations in representing multi-level object saliency in complex natural images.

2.1.1 Collection of the Datasets

In this chapter, we explain the image selection and subjective experiment procedures of the well-known datasets. Their limitations in measuring multi-level object saliency are explained and discussed in Section 2.1.2.



(a) Image

(b) Binary Ground-Truth Map

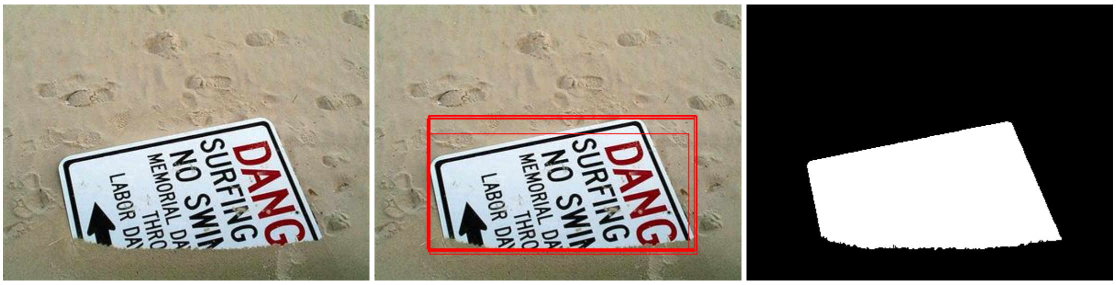
(c) Multi-Level Ground-Truth Map

(d) Salient-Object Map

Figure 2.1: For (a) an input image, (b) a binary ground -truth map is an "ideal" output of a salient-object detector, when we assume all objects to be equally salient. In our thesis, we invalidate this assumption by showing the varying importance of objects on (c) multi-level ground-truth maps. A method takes (a) a natural image as input and outputs (d) a pixel-precise salient-object map. Similarity between this map and a ground-truth map implies high accuracy in salient-object detection.

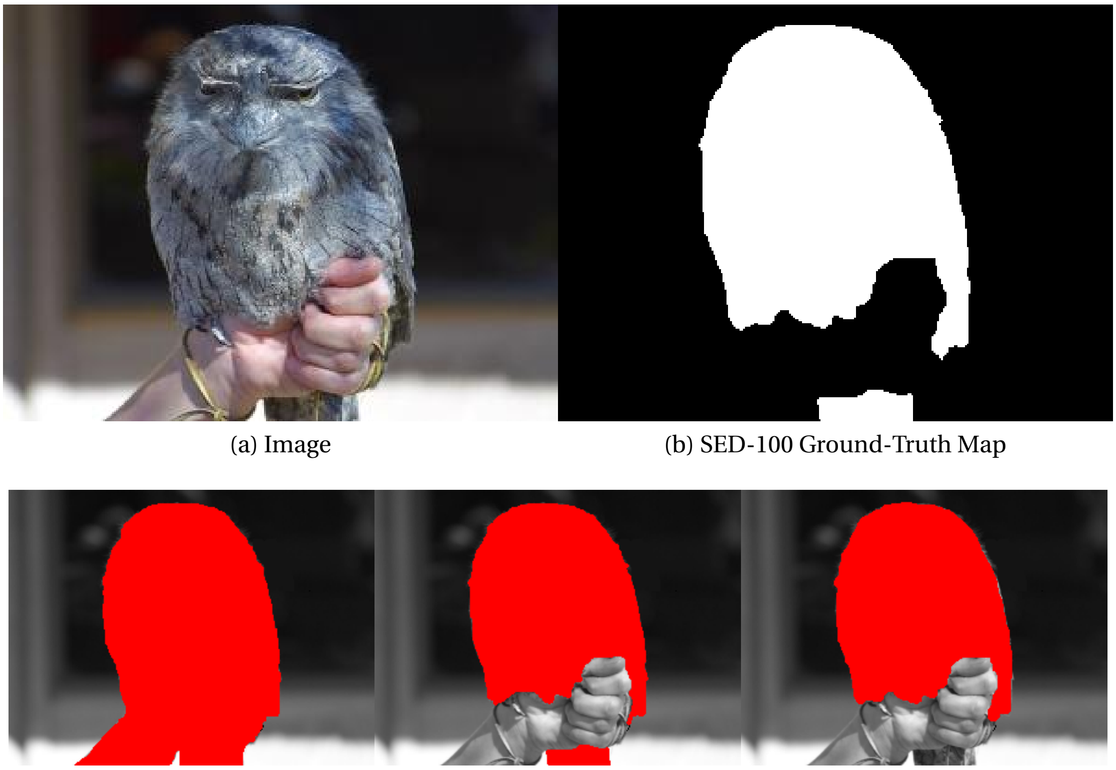
The MSRA-1000 Dataset [10] consists of 1000 natural images that are taken from the larger Microsoft Research Asia Dataset [37] (MSRA). The images in the original MSRA dataset [37] are specifically selected for having a distinctive foreground object. Then, nine subjects were asked to draw a rectangle around the most salient object in each image. In the derived MSRA-1000 dataset [10], these rectangles are used only to identify a single object, later segmented by one person, and to produce a binary ground-truth map as shown in Figure 2.2.

In order to avoid the ambiguity on object saliency, all 100 images in the Segmentation Evaluation Dataset [11] (SED-100) were selected for having a clear foreground object. Unlike MSRA-1000, the objects in each image were segmented by three subjects. The segmentations of the subjects were combined into a binary ground-truth map by considering a pixel to be salient when it is marked by more than one person. Examples of subjective segmentations are given in Figure 2.3.



(a) Image (b) MSRA - Experimental Data (c) MSRA-1000 Ground-Truth Map

Figure 2.2: (a) Each image in the MSRA dataset are shown to nine subjects. They were asked to (b) draw rectangles on the most salient object. (c) The ground-truth map of MSRA-1000 is manually obtained by one person, based on the drawn rectangles.

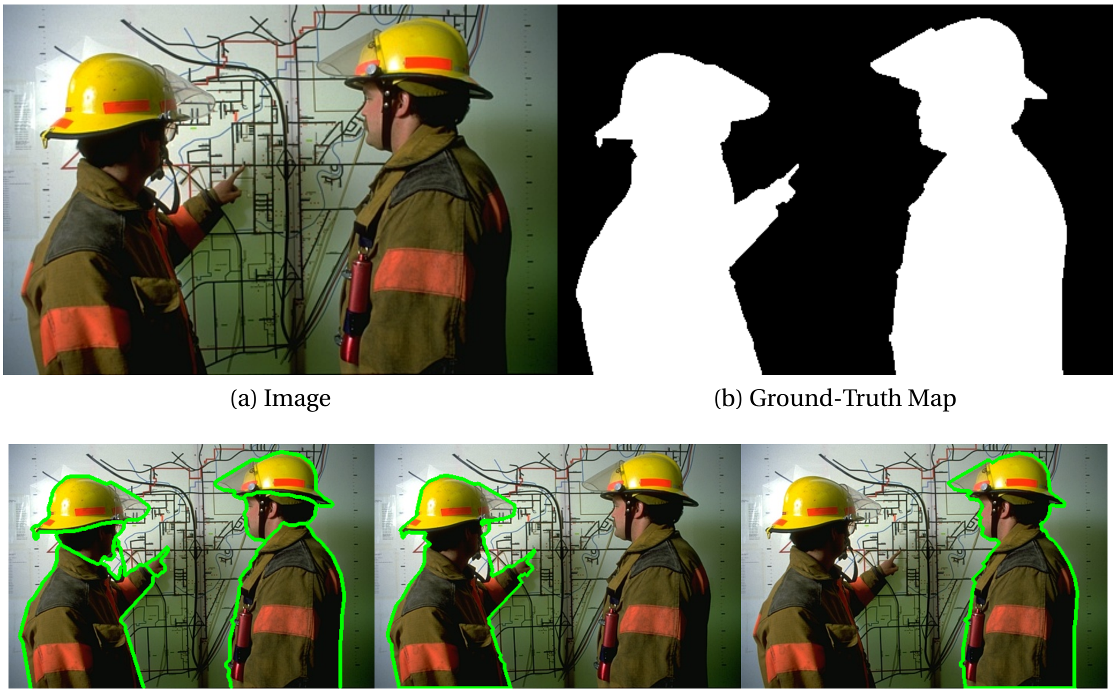


(a) Image (b) SED-100 Ground-Truth Map

(c) SED-100 - Subject #1 (d) SED-100 - Subject #2 (e) SED-100 - Subject #3

Figure 2.3: (a) Each image in the SED-100 dataset are shown to three subjects. They were asked to (c)-(e) segment the salient object. (cb) The pixels, which are marked by more than one person, are considered as salient in the ground truth map of SED-100 images.

The Salient Object Dataset [12] (SOD) was formed using the Berkeley Segmentation Dataset [38] (BSD). In BSD, images are segmented by three subjects. The sub-segments are not at object-level, i.e., the objects were divided into multiple sub-segments. Then in SOD, seven subjects are asked to identify salient objects by combining BSD subsegments. Different from the other datasets, during the collection of SOD, subjects were required to rank the objects with respect to their saliency, if they detect more than one object. Although object ranking emulates multi-level object saliency, it has several drawbacks, that are discussed in Section 2.1.2. The state-of-the-art salient-object detectors usually ignore this ranking and use a binary version of the ground-truth map as shown in Figure 2.4.



(a) Image

(b) Ground-Truth Map

(c) BSD - Human Segmentations

(d) SOD - Saliency Rank # 1

(e) SOD - Saliency Rank # 2

Figure 2.4: (a) Each image and (c) their BSD segmentations are shown to seven subjects. They were asked to (d)-(e) identify the salient objects by merging smaller segments and to rank objects based on their saliency. (b) The state-of-the-art salient object detectors only used binary ground-truth maps by thresholding object segmentations similar to SED-100.

2.1.2 Limitations of the Datasets

The well-known datasets consist mostly of simple images with a single, prominent object as illustrated in Figure 2.5. When there is a single salient object in an image, the collective human attention approaches to a consensus, which minimizes the subjectivity. On one hand, having a single salient object per image helps us understand the low-level visual cues that affect object saliency. On the other hand, it oversimplifies the salient-object detection problem, as natural images can be complex and can include multiple objects. In Chapter 3, we introduce the COS dataset that is more representative of natural images, compared to the well-known datasets.

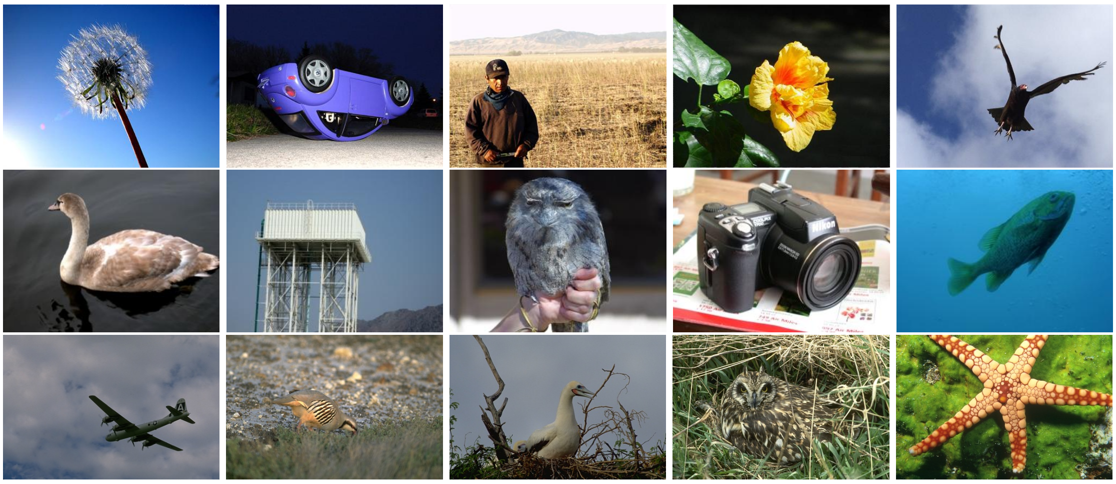
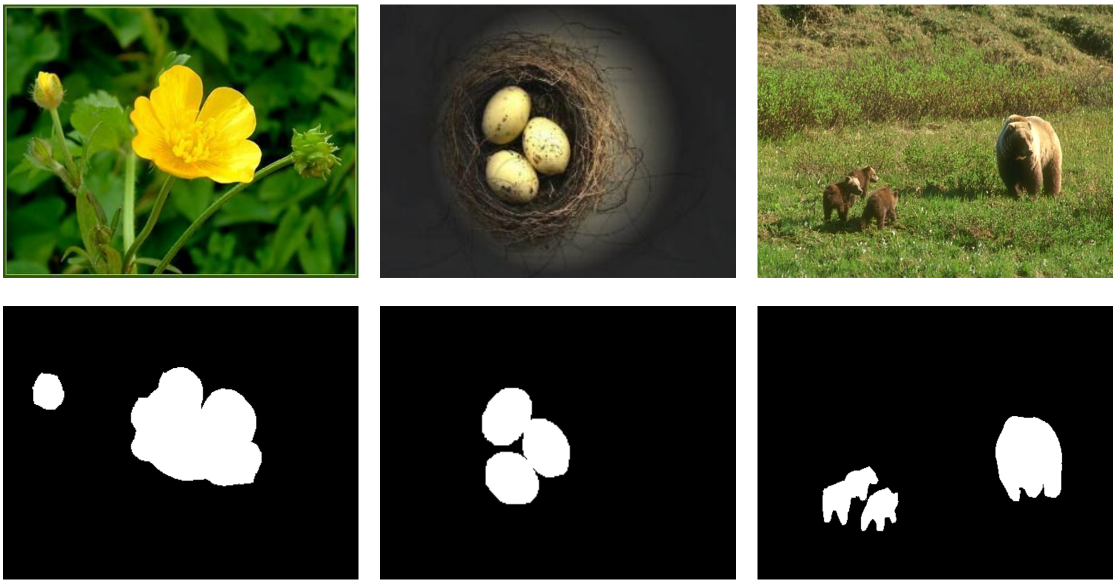


Figure 2.5: Typical image examples from the well-known datasets, MSRA-1000 (first row), SED-100 (second row), SOD (third row).

A small number of images in the datasets include multiple objects that could be used for investigating multi-level object saliency. However, in the MSRA-1000 and the SED-100 datasets, subjective data on multi-level object saliency were only used to identify the most prominent object and are not informative as far as the saliency level of an object is concerned. Although the subjects in the SOD dataset ranked the multiple objects with respect to their saliency, there are three problems with the data collection. First, the experiments are task-driven (sort the salient objects), which involves high-level cognitive functions and can bias the saliency measurements. The obtained saliency rankings might be overridden by semantic context rather than low-level visual cues. Second, the object rankings only show if an object is more salient than another one. It does not quantify the perceptual difference on saliency. Finally, when collecting the saliency data, subjects form salient objects from already sub-segmented objects. This could lead to a bias on what constitutes a salient object.

The experimental data in all datasets are used to form a binary ground-truth map as illustrated in Figure 2.6. As the state-of-the-art salient-object detectors are designed to divide an image into foreground and background regions, they were evaluated using these binary maps.

When there are multiple objects in an image, due to their relative importance to each other and the background, their saliency values might differ. As we discussed in Section 1.1 and 1.2, this difference is evident when the saliency of these objects is evaluated by different people, which creates subjectivity on object saliency. In Chapter 3, instead of removing the subjectivity, we introduce a new dataset that retains it in order to measure multi-level object saliency.



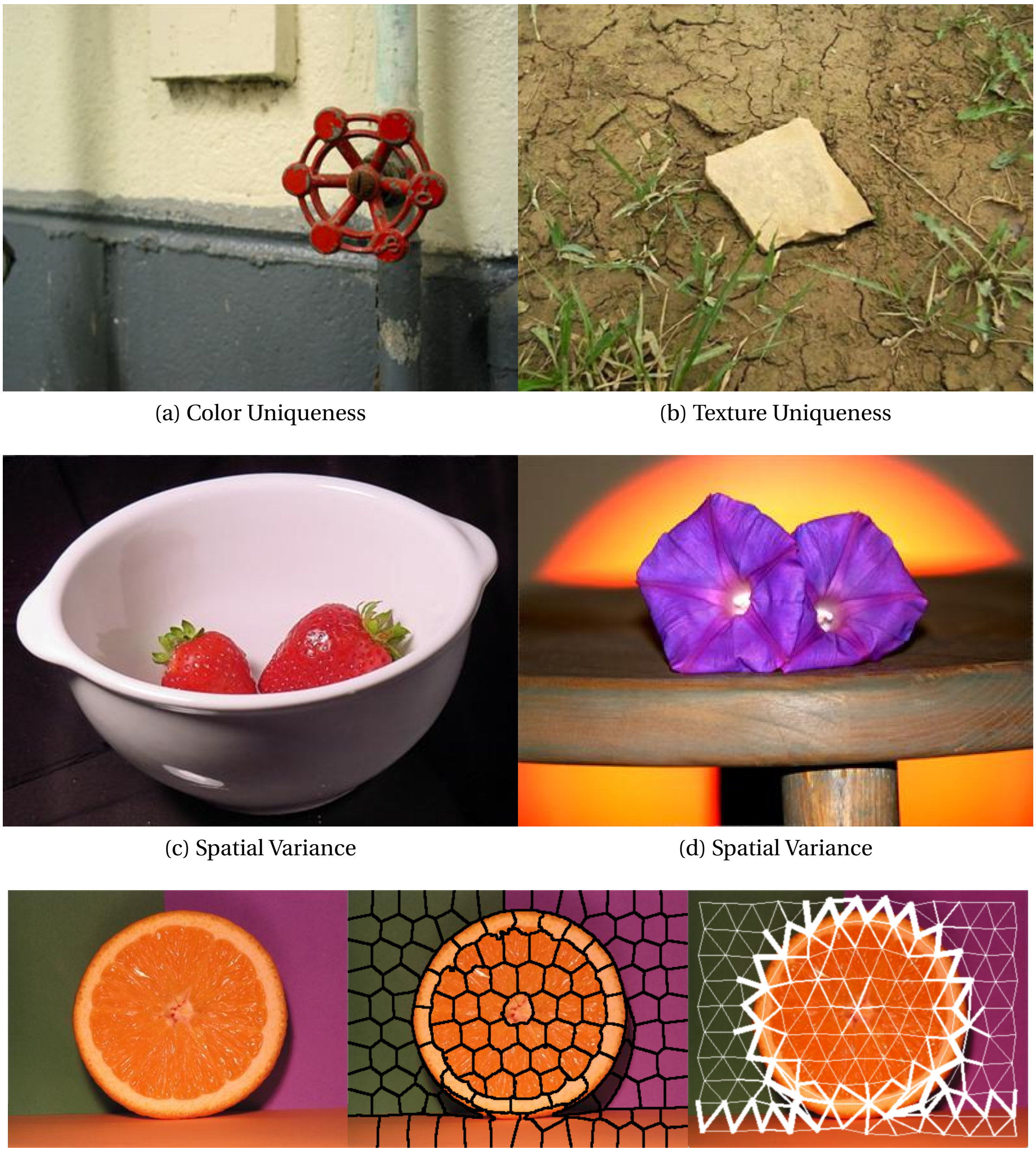
(a) MSRA-1000 (b) SED-100 (c) SOD

Figure 2.6: Images with multiple objects (first row) and their corresponding binary ground truth maps (second row).

2.2 Salient Object Detectors

Detecting salient regions in images has been comprehensively studied under the context of duplicating human fixation prediction [3, 25–28, 39–43]. Estimated eye-fixation maps indicate the salient objects as similarly sized blobs, instead of as with their accurate sizes and boundaries, as shown in Figure 1.3(e). These maps are sufficient for psychophysical studies [44–49]. However, for computer vision and image processing applications, such as object segmentation, image re-targeting, warping, and compression [13, 50–55], accurate object boundaries are very important. Therefore, salient-object detection has replaced fixation predictions in computer vision.

Salient-object detectors calculate a salient-object map that uniformly highlights salient object(s) and they suppress background pixels on a pixel-level accuracy. In order to achieve an accurate map, we can exploit various image characteristics that involve different cognitive levels. Current salient-object detectors usually focus on low-level features, such as color, texture, shape, and image edges. There has been an extensive amount of work on designing salient-object detectors [10,13–20,52,53,56–70]. Here, we choose 10 methods [10, 13–20] that are conceptually comparable to the techniques we propose in Chapter 4 and have publicly available codes. We group these 10 state-of-the-art salient-object detectors under three *mechanisms*, each of which uses low-level features in a different way: uniqueness, spatial variance, and spatial connectivity. These mechanisms are illustrated in Figure 2.7. For legibility, we refer to the state-of-the-art methods by their acronyms in Table 2.1



(a) Color Uniqueness

(b) Texture Uniqueness

(c) Spatial Variance

(d) Spatial Variance

(e) Image

(f) Segmentation

(g) Segmentation Graph

Figure 2.7: Uniqueness refers to how distinctive an object is, in terms of (a) color, (b) texture, or many other cues, compared to its surroundings. Spatial variance usually favors spatially compact objects as salient. Although the bowl in (c) and the sun in (d) have unique colors, their spatial variances are larger than that of the actual salient objects (strawberries and flowers). In spatial connectivity, (e) an image is divided into (f) segments that can be used to form (g) a graph. This graph can be used to robustly identify the salient object, even if there are multiple unique colors or spatial variance is ineffective (white lines indicate the edge weights of the graph).

Table 2.1: The acronyms of the state-of-the-art methods.

Mechanism	Reference	Acronym
Uniqueness	Achanta et al. [10]	FT
	Cheng et al. [13]	HC
	Cheng et al. [13]	RC
	Shen et al. [14]	LR
Spatial Variance	Perazzi et al. [15]	SF
	Cheng et al. [16]	GC
Spatial Connectivity	Jiang et al. [17]	AMC
	Yang et al. [18]	GMR
	Yan et al. [19]	HSD
	Li et al. [20]	CH

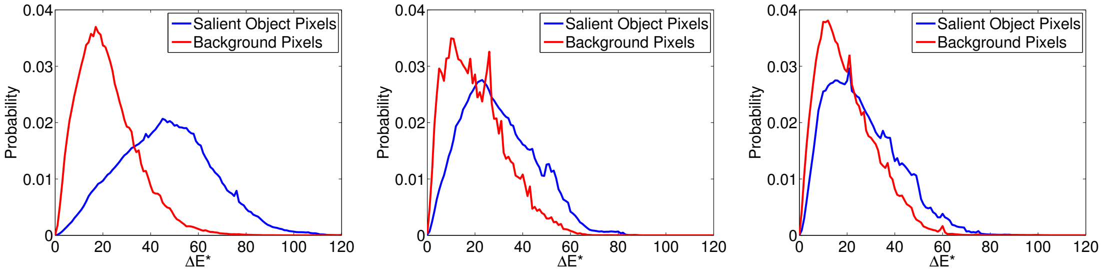
2.2.1 Uniqueness

In general, a salient object has a unique quality that makes it stand out, relative to the rest of the image. The most frequently used uniqueness measure is the perceptual difference between the color of a central and a surrounding region, i.e. color contrast. In the very early human vision, to detect the regions of interest of a visual input, the color-opponent receptive fields of our ganglion cells provide the initial information to our brains. Inspired by this, the color-contrast computations are often performed in a perceptually uniform opponent color space such as CIELa*b*, where color difference is measured by the Euclidean distance metric ΔE^*.

One of the pioneering works that used center-surround difference is by Itti et al. [3], where they computed a visual-importance map of an input image by combining color-, intensity-, and orientation-contrast values. This model mimics the HVS by outputting an estimated eye-fixation map that does not indicate the size and the shape of a salient object. Itti et al.'s idea is modified by Achanta et al. [10] (FT) to produce object-level maps, i.e. salient-object maps, by computing the saliency value of a pixel as follows:

$$\mathbf{S}(x,y) = ||\mathbf{I}_\mu - \mathbf{I}_\sigma(x,y)|| \tag{2.1}$$

Here, $\mathbf{S}(x,y)$ is the estimated saliency value at coordinates (x,y), $\mathbf{I}_\mu$ is the average color of the image, and $\mathbf{I}_\sigma(x,y)$ is the image filtered with a small Gaussian filter that eliminates noise and object texture. In this method, color contrast is measured between a small central region and a surrounding region that covers the whole image. This approach assumes that the average color of an image is perceptually closer to the color(s) of the background pixels than the color(s) of the salient object. In order to validate this assumption, in Figure 2.8, we illustrate the probability of ΔE^* between image pixels and the average color of the images in well-known datasets. When the ΔE^* distributions of the salient object and the background pixels are separate, this means that color-contrast feature can perform better in salient-object detection. According to Figure 2.8, the MSRA-1000 dataset consists of salient objects with unique colors. The color uniqueness seems to decrease for the SED-100 and the SOD datasets, indicating that color-contrast based salient-object detectors might perform worse on these datasets, which we demonstrate in Section 2.3.



(a) MSRA-1000 (b) SED-100 (c) SOD

Figure 2.8: The probability distributions of ΔE^* values computed between image pixels and the average image color in respective datasets. Blue and red lines correspond to salient object and background pixels, respectively. The object pixels in the MSRA-1000 dataset are more likely to have ΔE^* values larger than that of the background pixels.

One slightly better approach for computing the color contrast is to calculate,for each pixel, the average value of color distances of a pixel to the rest of the image pixels as follows:

$$\mathbf{S}(x,y) = \frac{1}{N}\sum_{x'}\sum_{y'} ||\mathbf{I}(x,y) - \mathbf{I}(x',y')|| \tag{2.2}$$

Here, N is the number of pixels in the image $\mathbf{I}$. The main drawback of this method is that we need to perform this computation for each pixel in the image, giving us a $O(N^2)$ computational complexity. In order to estimate a salient-object map in a rapid fashion, (2.2) is slightly modified by Cheng et al. [13] (HC) with a color-quantization step. The salient-object map using the quantized colors are calculated by this method as follows:

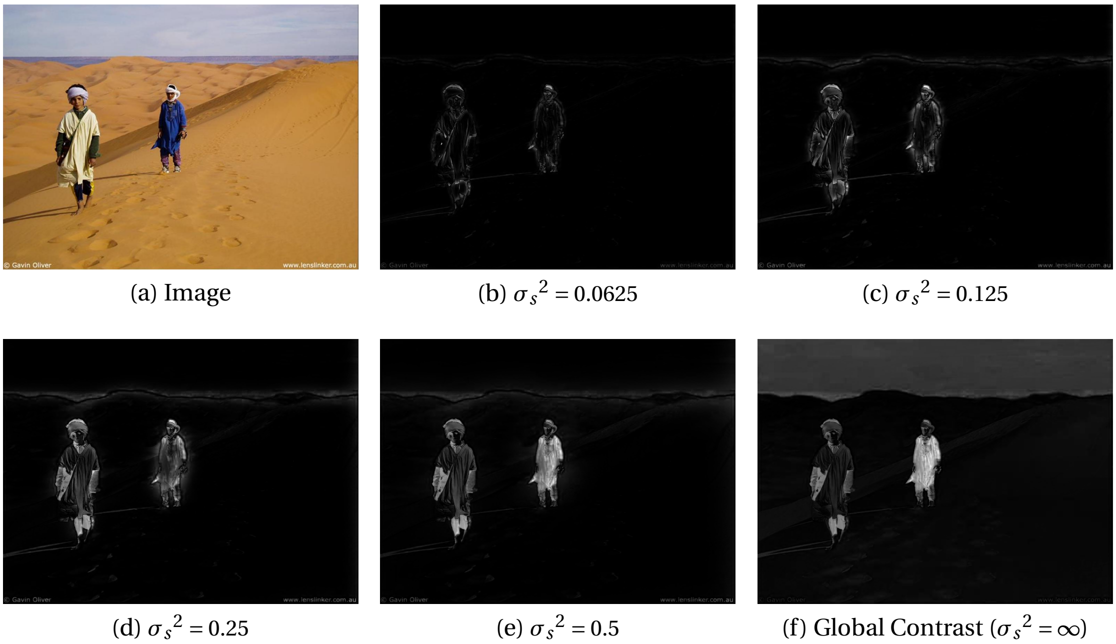
$$\mathbf{S}(\mathbf{q}) = \frac{1}{N_Q}\sum_{c=1}^{N_Q} h_c \cdot ||\mathbf{q}_c - \mathbf{q}|| \tag{2.3}$$

Here, N_Q is the number of quantized colors, $\mathbf{q}_c$ is a quantized color, h_c is the number of pixels that are quantized into $\mathbf{q}_c$, and $\mathbf{q}$ is the quantized color for which global-color-contrast is computed. The color-quantization step reduces the number of color distance computations to a few tens of colors, thus decreasing the computational complexity to $O(N_Q^2)$.

Global color-contrast measures how unique a color is throughout the whole image. When an object has a color that is only distinctive compared to its local surroundings, global methods can fail. Therefore, in the same paper Cheng et al. [13] (RC) propose a second method that compromises between local and global color-contrast as follows:

$$\mathbf{S}(\mathbf{q}) = \frac{1}{N_Q} \sum_{c=1}^{N_Q} h_c \cdot ||\mathbf{q}_c - \mathbf{q}|| \cdot \exp\left(- ||\mathbf{p}_c - \mathbf{p}||^2 / {\sigma_s}^2 \right) \tag{2.4}$$

Here, $\mathbf{p}_c$ and $\mathbf{p}$ are the position vectors of the quantized colors $\mathbf{q}_c$ and $\mathbf{q}$, respectively, and σ_s adjusts the balance between local and global contrast. Although in RC the adjusting parameter is fixed to ${\sigma_s}^2 = 0.4$ (in terms of normalized image coordinates), the optimal σ_s value depends on the size of the salient object(s) and how the color is distributed along the image. An example image with different σ_s values are given in Figure 2.9. A small σ_s value acts as an edge detector and the optimal value that maximizes the saliency of an object depends on its size and the shapes surrounding it. In order to minimize the problems of optimal-parameter selection, in Section 4.2 we vary σ_s and combine the results and in Section 4.4, we use a machine learning technique on a hierarchical representation of an input image.



(a) Image

(b) ${\sigma_s}^2 = 0.0625$

(c) ${\sigma_s}^2 = 0.125$

(d) ${\sigma_s}^2 = 0.25$

(e) ${\sigma_s}^2 = 0.5$

(f) Global Contrast (${\sigma_s}^2 = \infty$)

Figure 2.9: The color contrast based salient object maps of (a) the image with (b)-(f) different σ_s values. The saliency value of the objects on the left and right are maximized in (e) and (f), respectively.

In addition to the uniqueness in color, Shen et al. [14] (LR) investigated the effect of uniqueness in image edges and their orientations on object saliency. For this purpose, first they oversegment an image into superpixels, a group of neighboring pixels that are similar in color. Then the feature vectors (color, edge, and orientation) of all superpixels are concatenated in a feature matrix. In their method, this feature matrix is assumed to be the sum of two matrices. The first one is a low-rank matrix that represents the highly correlated background superpixels. The second one is a sparse matrix that represents the unique, i.e. salient, superpixels in the image. In Figure 2.10, we visually compare the salient-object maps that are estimated by the state-of-the-art methods, which use the uniqueness as the main premise during a salient-object detection operation.

Due to its important role in the very early human vision, color contrast is a fundamental way of identifying the unique regions in an image. The optimal way to use color contrast in salient-object detection is to adjust the size and shape of the center and surrounding regions with respect to the size and the shape of the object we want to detect. Obviously, this creates a circular reference between salient-object detection and object characteristics. One way to break the loop is to have an algorithm that does **not** know, but is *aware* of the spatial characteristics of the salient objects. The state-of-the-art methods achieve this by computing the "spatial variance" of colors. We extend this idea in Section 4.3 to detect the position and size of the salient objects and in Chapter 5 to move from binary to multi-level object saliency via Object-Awareness Model (OAM).

2.2.2 Spatial Variance

In a natural image, in general, the pixels of a salient object are concentrated in a certain part of the image, whereas the background pixels are distributed in the image. Salient-object detectors exploit this property by calculating the spatial variance of colors and by favoring colors with small variations when calculating salient-object maps. One way to compute the spatial distribution of a color is as follows:

$$\begin{aligned} V_i &= \sum_j ||\overline{\mathbf{p}}_i - \mathbf{p}_j||^2 \cdot \exp\Big(- \frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_c}^2} \Big) \\ \overline{\mathbf{p}}_i &= \sum_j \mathbf{p}_j \cdot \exp\Big(- \frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_c}^2} \Big) \end{aligned} \tag{2.5}$$

Here, $\mathbf{c}_i$ is a color in the image, $\mathbf{p}_i$, $\overline{\mathbf{p}}_i$, and V_i are the spatial position, color-weighted spatial position, and color-weighted spatial variance of that color, respectively. The parameters σ_c and σ_s control the effect of the color contrast and spatial distance. In Figure 2.11, we illustrate the probability of V_i values, calculated using the average colors of the salient objects and the background regions, in the well-known datasets. Here, salient objects are more likely to have a small spatial variance, i.e. salient objects are spatially smaller than the background regions. In Figure 2.11, as in Figure 2.8, we can see that spatial variance feature is more beneficial for the MSRA-1000 dataset and the benefits decrease for the SED-100 and the SOD datasets.

Spatial variance is employed by Perazzi et al. [15] (SF) and by Cheng et al. [16] (GC) under similar setups with different names. They both oversegment an image into groups of pixels using superpixel segmentation and Gaussian Mixture Models, respectively. Both methods compute the uniqueness and the spatial variance of image colors. In order to estimate a final salient-object map, in SF these two measures are

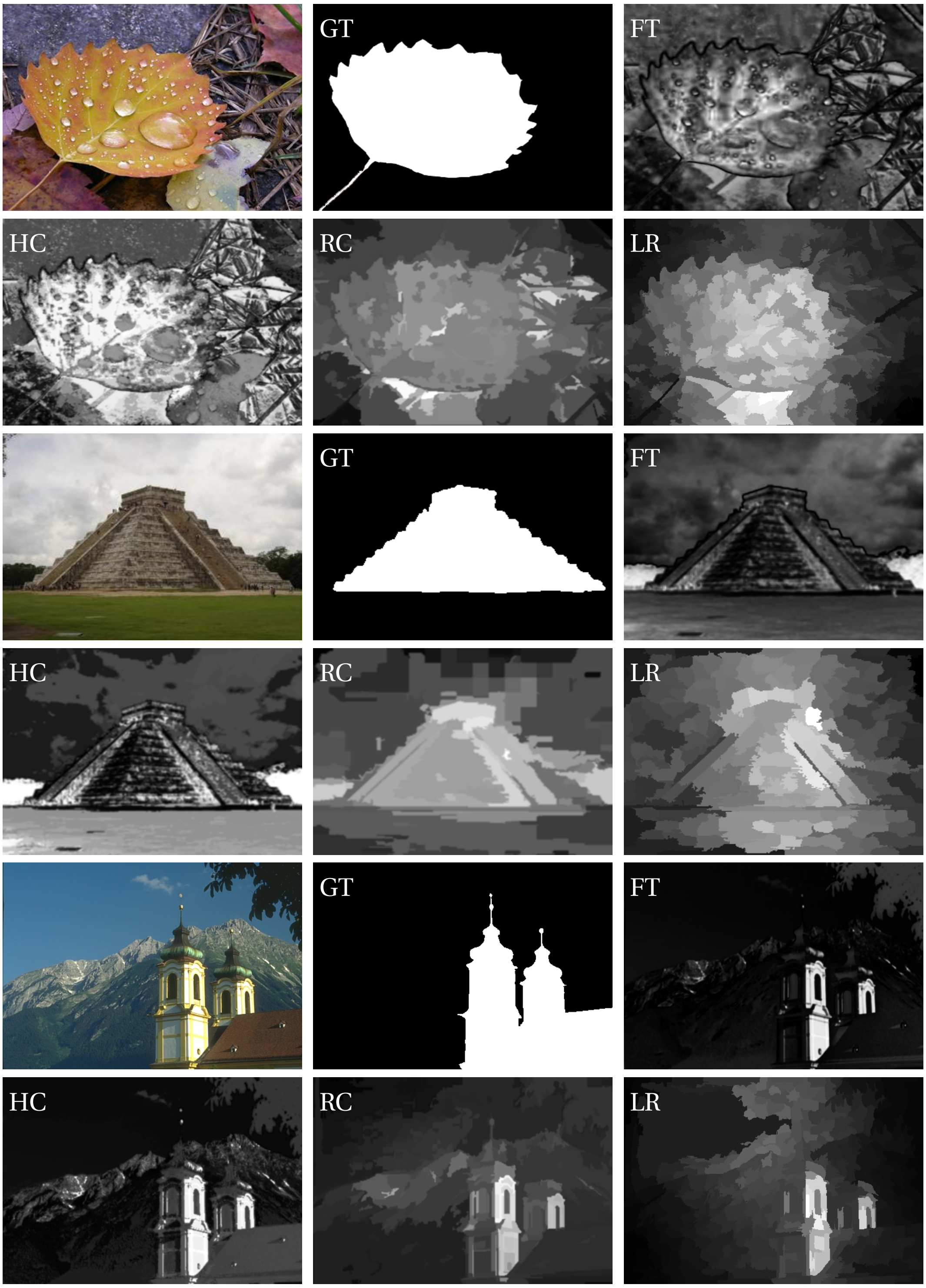
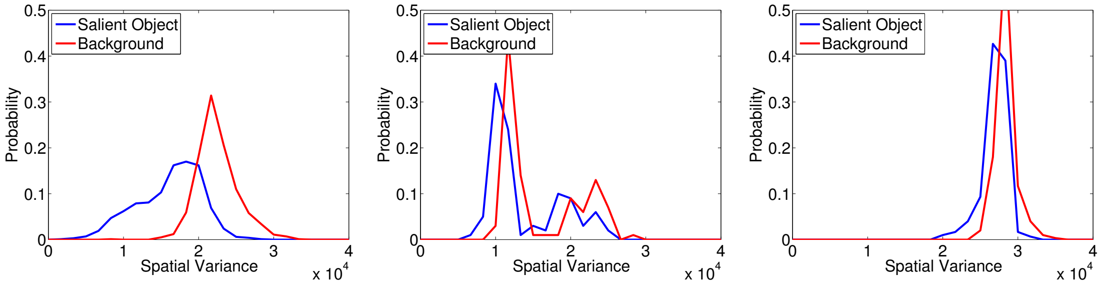


Figure 2.10: The visual comparison of salient-object maps of the uniqueness-based methods (GT: Ground-Truth Map).



(a) MSRA-1000 (b) SED-100 (c) SOD

Figure 2.11: The probability of color spatial variances of the well-known datasets. Blue and red lines correspond to the spatial variance of the salient and background regions, respectively.

combined. Whereas in GC, one of them is selected based on a spatial salient-object map-compactness criterion.

During the uniqueness calculations, both of the methods use a fixed σ_s value (see (2.4)), which is not optimal if the salient-object size widely varies. However, in order to compensate for this, they use the spatial variance measure. This variance is assumed to be inversely related to the saliency of an object. Therefore, distributed background pixels are suppressed and a bias is introduced towards smaller salient objects, because their spatial variances are small. The salient-object maps that are estimated by SF and GC are visually compared in Figure 2.12.

Unlike SF and GC, instead of favoring only small objects, in Section 4.3, we propose a way to use this variance to compute the position and the size of a salient object, and we incorporate this method into a probabilistic framework for deciding the saliency value of that object.

LR, SF, and GC cluster image pixels into segments based on their color and spatial position, which facilitates the computation of a salient-object map. Another layer of information can be explored through connecting these clusters based on their neighborhood properties, forming a graph from the image segments, and analyzing the spatial connectivity of this graph for salient objects.

2.2.3 Spatial Connectivity

Image over-segmentation assists salient-object detection in three ways. First, image segments usually follow the boundary between an object and a background region, which provides medium-level context-aware information. Second, it can reduce the effect of noise and texture on the detected salient-object map. Third, it generates a much smaller abstraction of the image (around a few hundred superpixels), compared to hundreds of thousands of pixels; this can be used to exploit spatial connectivity of image regions. Therefore, the most recent methods [17–20] use an over-segmentation algorithm as a pre-processing step to salient-object detection. The main approach in exploiting the spatial connectivity of superpixels is to form a graph of the image,

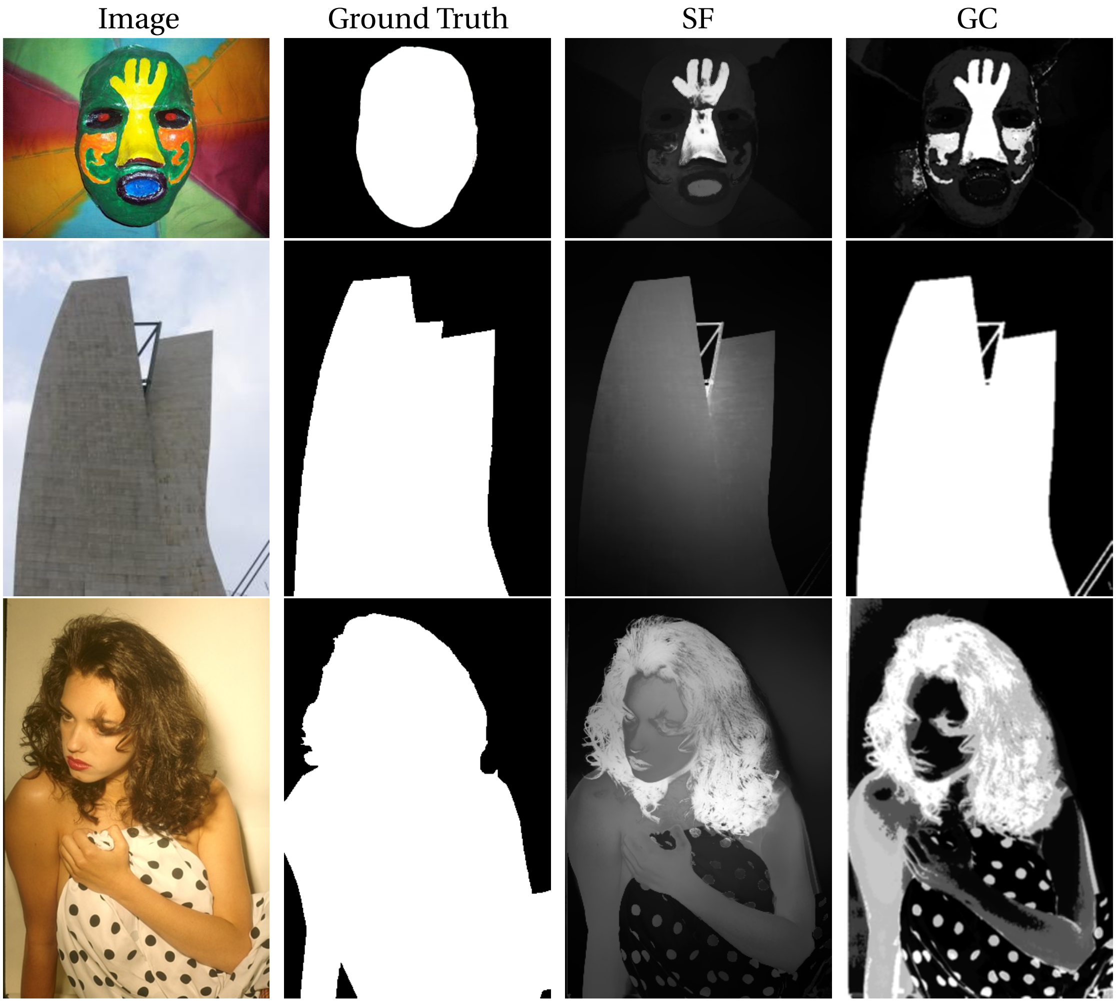
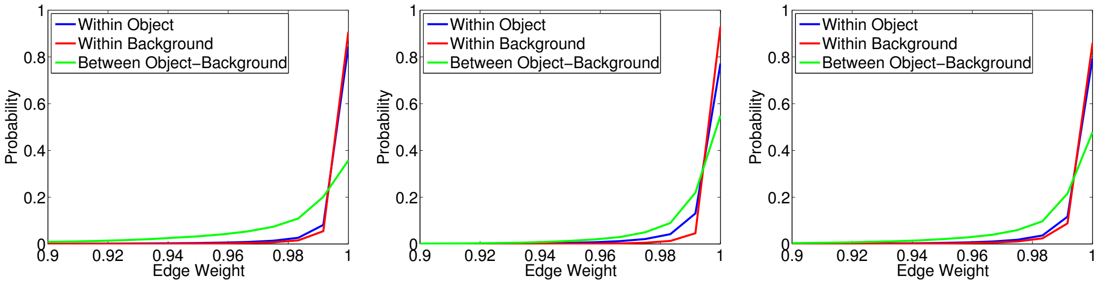


Figure 2.12: The visual comparison of salient object maps that are estimated by the methods, which employ spatial variance.

where each superpixel is a node, superpixels that share boundary pixels are connected with edges, and the weight of the edges are calculated using the average colors of the superpixels as follows:

$$w_{ij} = \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_c}^2}\right) \tag{2.6}$$

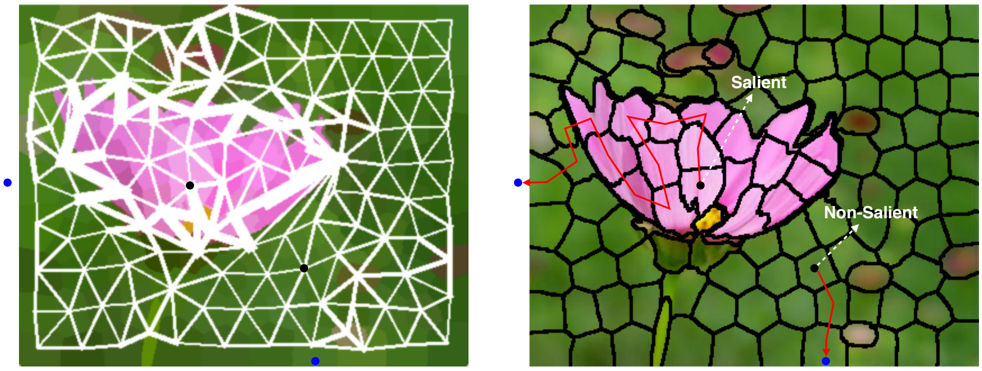
Here, w_{ij} is the weight of the edge between i^{th} and j^{th} superpixels, $\mathbf{c}_i$ and $\mathbf{c}_j$ are the average color of these superpixels. The effect of the color difference is controlled via σ_c. In Figure 2.13, we illustrate the probability of edge weights within salient objects, within background regions, and between objects and backgrounds. Here, we can see that within objects and background regions, the probability of having a large edge weight is approximately equal to one, i.e. the average colors of the superpixels are similar within a structure, whether it be an object or a background. On the contrary, the weights of the edges, which connect an object superpixel to a background superpixel, are likely to have values smaller than 1.



(a) MSRA-1000 (b) SED-100 (c) SOD

Figure 2.13: The probability of edge weights between neighboring superpixels for the images in well-known datasets.

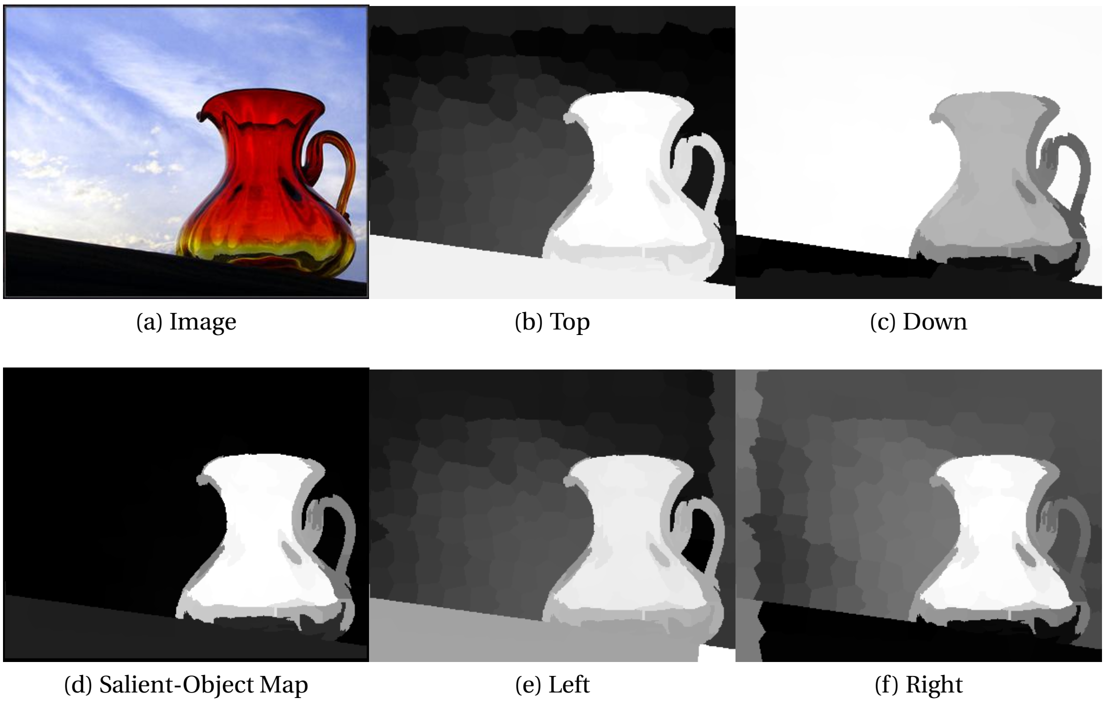
A salient object detector can benefit from the spatial connectivity of a superpixel graph in various ways. In [17], Jiang et al. (AMC) forms a superpixel graph with edge weights that are calculated using (2.6). This graph is assumed to be an absorbing Markov chain, where the saliency value of a superpixel is directly related to the absorption time of that superpixel by an absorbing node that resides outside the image boundary. An illustration of this method is shown in Figure 2.14. The nodes that are at the edge of the image are mirrored and constitute the absorbing nodes. The edge weights are considered to be the probability of transmission from one node to the other one, i.e., it is more probable for algorithm to jump between superpixels that are similar in color. Therefore, a background pixel jumps to an absorbing node on a smaller average time. Whereas, a salient superpixel takes a longer path before being absorbed.



(a) Superpixel graph (b) Example absorption path

Figure 2.14: (a) A superpixel graph is formed and (b) the average time of a node (black dots) to reach an absorption node (blue dots) through a possible absorption path is calculated as the saliency of that superpixel.

In [18], Yang et al. (GMR) formed a superpixel graph of an image that is similar to AMC, whereas this time, manifold rankings of superpixels with four different background initial conditions are combined into a final salient-object map. An illustration of this method is given in Figure 2.15. Here, each initial condition assumes that the superpixels at the border of the image belong to the background. Therefore, when the "down" border of the image in Figure 2.15 (c) was assumed to be background, as the salient object is very close to the lower boundary of the image, the generated salient-object map was inaccurate. In order to avoid problematic cases, the method combines four conditions into one final salient-object map in Figure 2.15 (d).


(a) Image

(b) Top

(c) Down

(d) Salient-Object Map

(e) Left

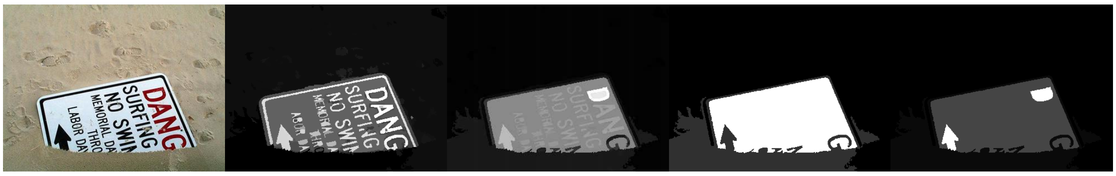
(f) Right

Figure 2.15: (a) An image is processed with (b, c, e, f) four initial conditions that assume that side of the image to be a background region. The maps that are obtained using these conditions are then combined into (d) a final salient object map.

In AMC and GMR, a single-layer graph is formed from the superpixels. In [19] and [20], a hierarchical graph with multiple layers is formed by merging superpixels into larger ones.

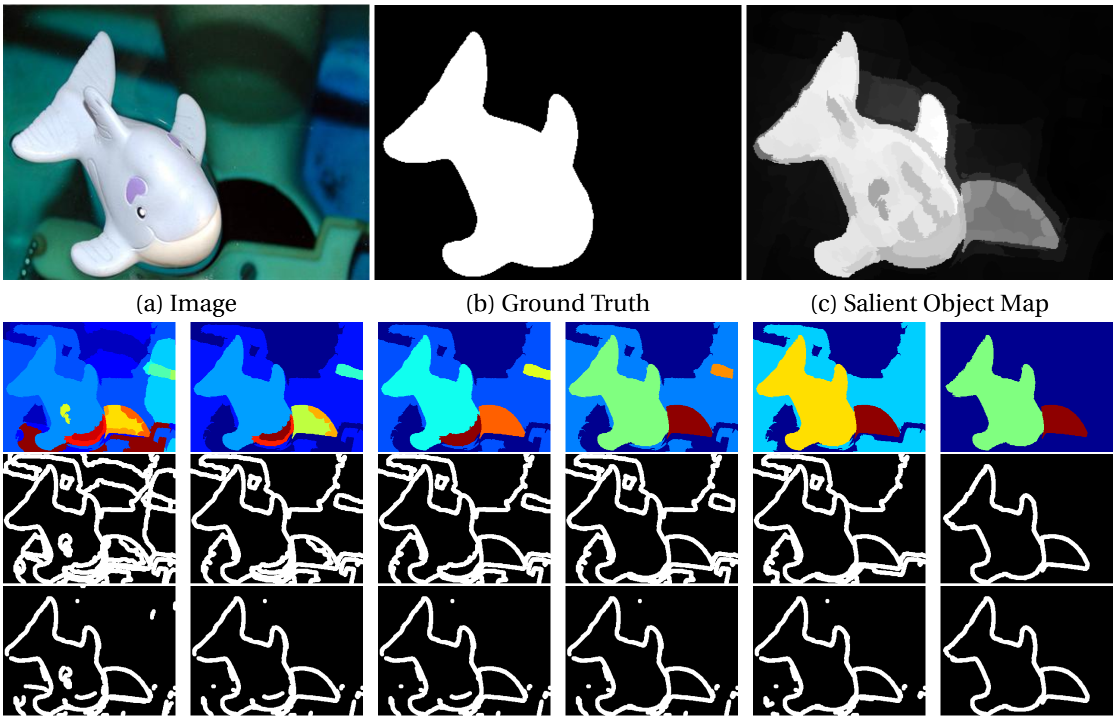
In [19], Yan et al. (HSD) form a three-layer graph from an over-segmented image and compute the color contrast at each layer as separate salient-object maps. These maps are then combined into a final salient-object map via belief propagation. The illustration of the layers and the final map is given in Figure 2.16.

In [20], Li et al. (CH) create a graph from image superpixels. They then form multiple hierarchical layers and larger superpixels by merging superpixels of previous layers. The saliency value of a superpixel is related to number of image edges on its perimeter



(a) Image (b) Layer-1 (c) Layer-2 (d) Layer-3 (e) Salient-Object Map

Figure 2.16: (a) An image is separated into (b, c, d) three layers that compute color contrast. These layers are combined into (e) a final salient-object map.



(a) Image (b) Ground Truth (c) Salient Object Map

Figure 2.17: The superpixels of (a) the image are merged to form larger superpixels of the multi-layer graph (second row). The image edges are multiplied with superpixel edges (third row) in an element-wise fashion in order to determine the contextual clues (fourth row).

(see the last row of Figure 2.17) and the number of image-boundary pixels inside it. An illustration of this method is given in Figure 2.17.

The salient object detectors that use spatial connectivity of superpixels are visually compared in Figure 2.18.

Spatial connectivity, along with the hierarchical representation of an image with multiple layers, are very helpful for understanding the image on an object level that is independent from the position and the size of the object. Therefore, in Section 4.4 and Chapter 5, we employ spatial connectivity in our machine-learning-based methods.

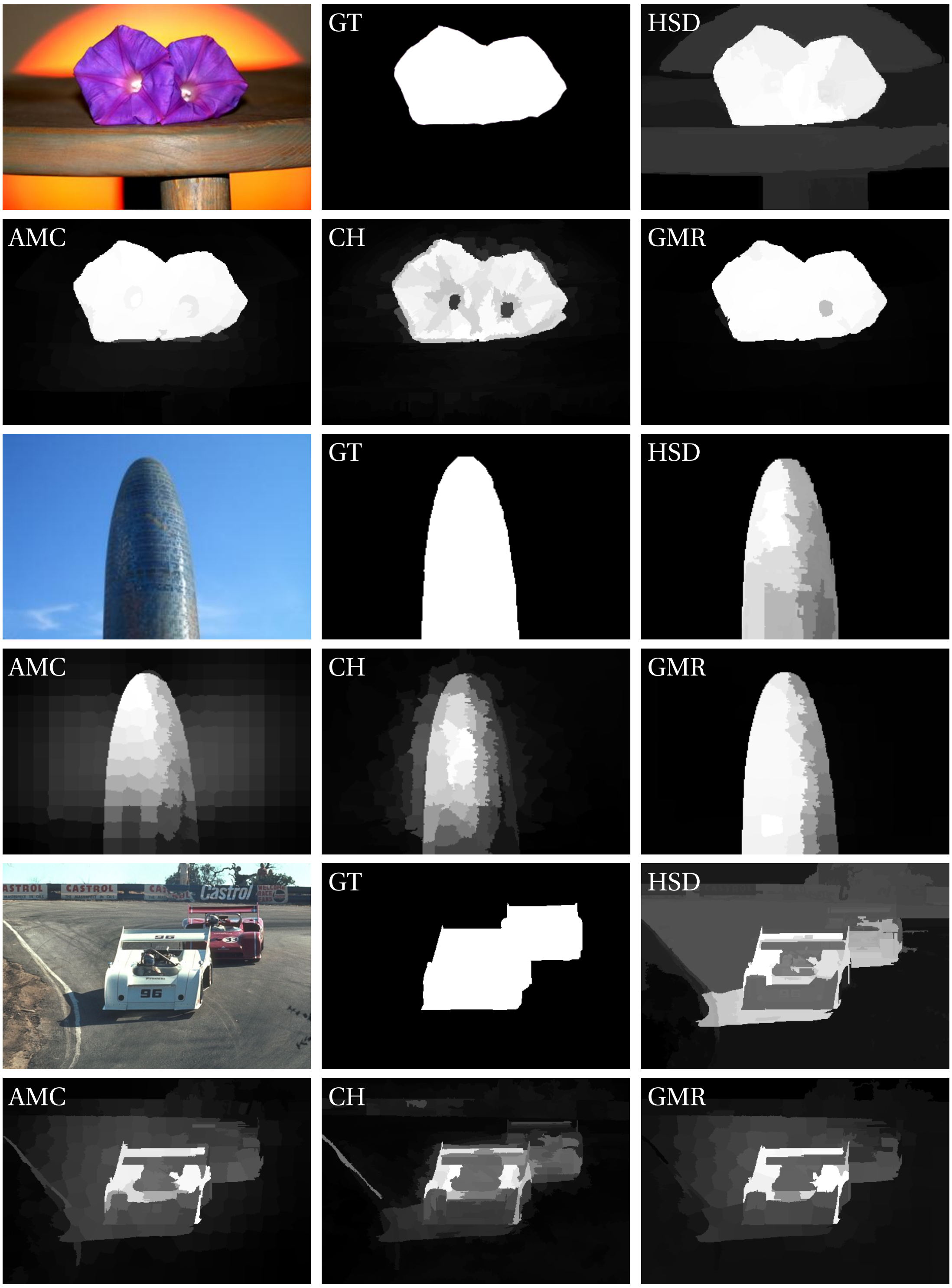


Figure 2.18: The visual comparison of salient object maps that are estimated by the methods, which employ spatial connectivity (GT: Ground-Truth Map).

2.3 Performance of Salient-Object Detectors

In this section, we evaluate the performance of the state-of-the-art salient-object detectors on the well-known datasets with binary ground-truth maps through three metrics: precision-recall curves, F-measure, and mean absolute error. As we mentioned before, these methods are selected based on their conceptual similarity to our proposed methods in Chapter 4 and their publicly available codes.

2.3.1 Precision-Recall Curves

The accuracy of a binary salient-object map is affected by both correctly estimating the foreground object and suppressing the background pixels. The precision and recall values measure the trade-off between foreground and background performance as follows:

$$\begin{aligned}\text{Precision} &= \frac{t_p}{t_p + f_p}\\ \text{Recall} &= \frac{t_p}{t_p + f_n}\end{aligned} \tag{2.7}$$

Here, t_p is the number of true positives, f_p is the number of false positives, and f_n is the number of false negatives, all of which are illustrated in Figure 2.19. Here, the number of true positive pixels (in yellow) is equal to 18.

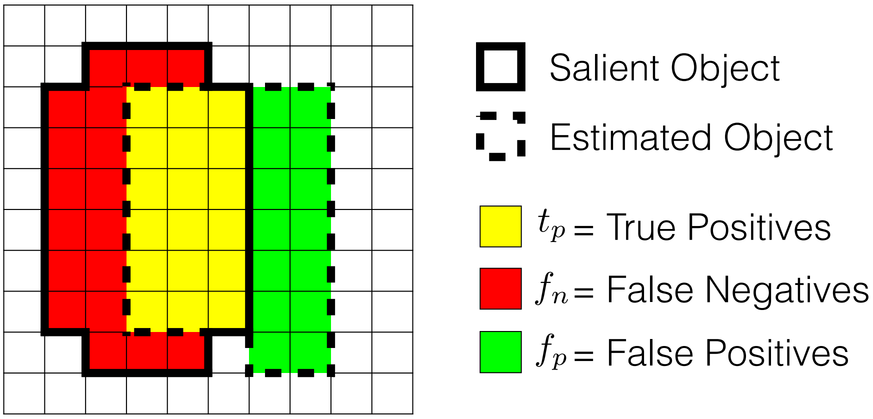


Figure 2.19: The representation of true and false positives, and false negatives.

Precision can be considered as background-suppression accuracy, whereas recall is related to salient-object detection accuracy. Precision and recall value pairs can be estimated by comparing two binary masks. Even though salient-object detectors attempt to estimate a binary salient-object map, in order to avoid binarization problems, they produce grayscale maps. Therefore, instead of measuring the performance of a map using a single precision-recall pair, we binarize the estimated salient-object maps with several thresholds and plot the curve of the resulting precision-recall pairs. In Figure 2.20, we calculate the precision-recall curves of the state-of-the-art salient-object detectors on the well-known datasets by varying the threshold from 0 to 255 with steps of one. Here, an ideal curve forms a square with a unit area underneath.

The performance of the algorithms increase from uniqueness to spatial connectivity as the methods add more salient-object detection mechanisms.

RC performs better than global color-contrast methods, such as FT and HC, because, unlike them, it relies on a balance between local and global color-contrast. As LR adds more aspects of uniqueness to color, such as image edge and orientation features, it surpasses the color-only methods FT, HC, and RC.

The methods that add spatial variance on top of uniqueness, such as SF and GC, outperform the uniqueness-only methods. There is no significant difference between SF and GC, due to the similar algorithmic structure.

The best performing algorithms are those that exploit the spatial connectivity through superpixel graphs, such as AMC, GMR, CH, and HSD. When we compare the methods using single and multi-layer graphs, unexpectedly, single-scale graph-based methods AMC and GMR perform slightly better than CH and HSD. CH computes the final salient-object map by linearly combining (with fixed weights) individual maps obtained at each layer. Even though one of the layers is really accurate, linear combination of several salient-object maps could lead to a map that is worse than the individual maps. In HSD, only the color-contrast feature is used, which is not enough when we look at the performance of the color-contrast methods such as FT, HC, and RC.

2.3.2 F-Measure

In Section 2.3.1, precision-recall curves are obtained by varying the binarizing threshold. If a computer vision application requires a foreground map, we need to adaptively choose a threshold that depends on the map. This forces us to choose an operating point on the precision-recall curve and the accuracy of this point is evaluated using F-measure. In order to choose an adaptive operating point for all of the state-of-the-art methods, we use Otsu's widely-used method [71]. We then calculate the precision-recall pair of the salient-object map that is binarized via the adaptive threshold. Finally, the F-measure can be computed as follows:

$$\text{F-Measure} = \frac{(\beta^2+1)\cdot \text{Precision}\cdot \text{Recall}}{\beta^2\cdot \text{Precision}+\text{Recall}} \tag{2.8}$$

Here, $\beta^2 = 1$ is the coefficient of trade-off between precision and recall. The average F-Measure values of the salient object detectors on the well-known datasets are shown in Figure 2.20.

The performance more or less follows the same trend in Section 2.3.1, where it increases from uniqueness to spatial connectivity-based methods. The only exceptions are the uniqueness-based methods LR and RC, both of which perform only slightly better than the spatial variance-based methods.

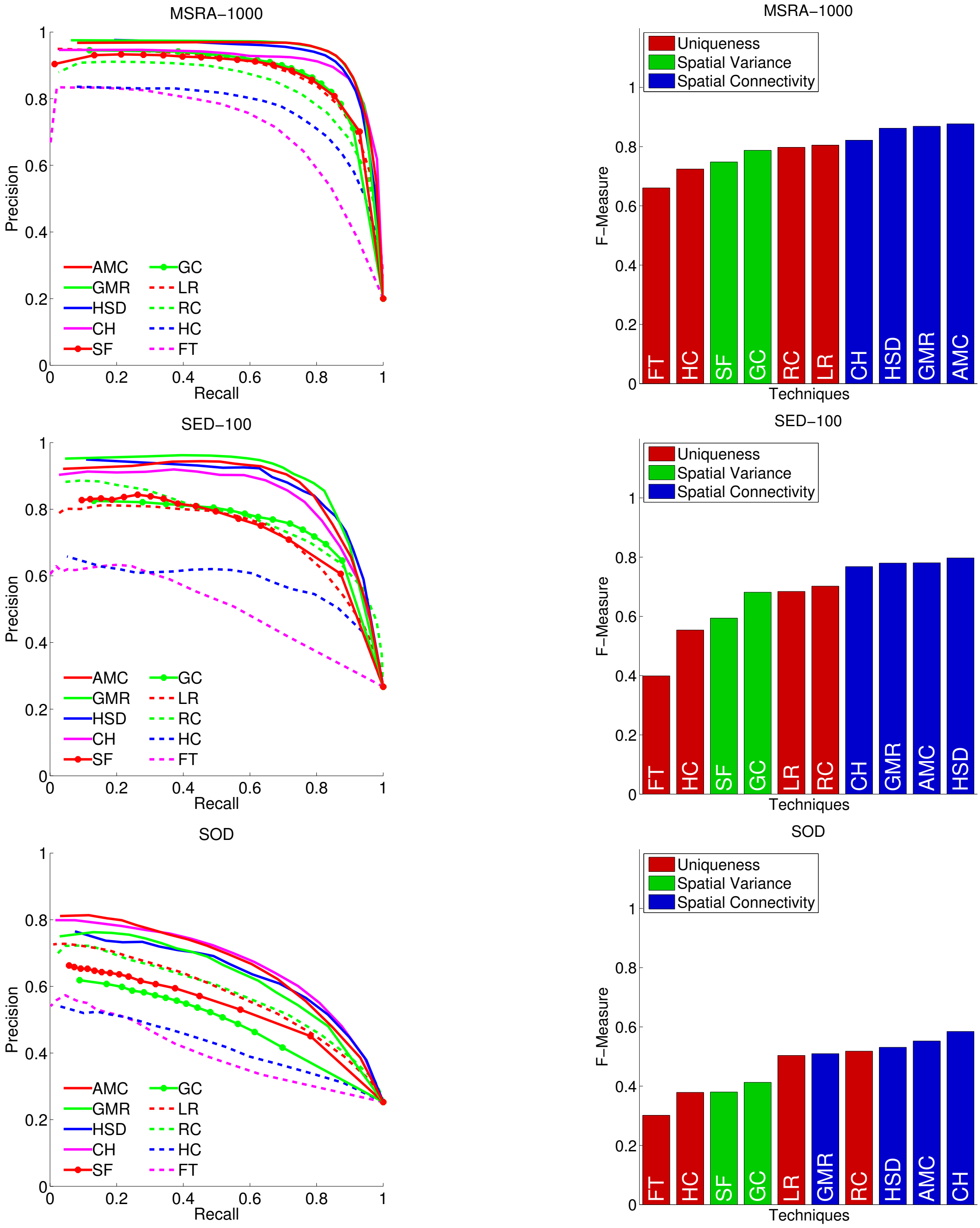


Figure 2.20: Precision-Recall curves of the salient object detectors on the well-known datasets (on the left). The uniqueness, spatial variance, and spatial connectivity-based methods are shown with dashed, circle markered, and solid lines, respectively. The average F-Measure values of the salient object detection methods on the well-known datasets (on the right).

2.3.3 Mean Absolute Error

Mean Absolute Error (MAE) measures the absolute pixel-wise error between an estimated salient object map and a ground truth map as follows:

$$
\begin{aligned}
\text{Mean Absolute Error (Foreground)} &= \frac{1}{N_F} \sum_{(x,y)\in F} |\mathbf{S}(x,y) - \mathbf{G}(x,y)| \\
\text{Mean Absolute Error (Background)} &= \frac{1}{N_B} \sum_{(x,y)\in B} |\mathbf{S}(x,y) - \mathbf{G}(x,y)|
\end{aligned}
\tag{2.9}
$$

Here, the foreground error is computed using the foreground pixels (F) and N_F is the total number of foreground, i.e. salient pixels in the image. A similar calculation is performed for the background error. As mean absolute error is a pixel-wise absolute error metric, we do not need to binarize the salient-object map. The mean absolute errors of the salient-object detectors on the well-known datasets are shown in Figure 2.21. As far as the foreground error is concerned, spatial-connectivity-based methods perform better than the other mechanisms. The main reason for this might be their robustness against image textures, which causes a more uniform salient-object map and thus lower mean absolute errors. Spatial-variance-based methods have a comparable performance to the spatial-connectivity-based methods in terms of background error, because favoring small spatial variances provides good background elimination and low background errors. As the mean absolute error metric does not require a binary ground-truth map, we use this metric in Chapter 5 when we compare various methods on multi-level ground truth maps.

2.4 Discussion

There are many low-level visual cues that affect object saliency including but not limited to color, texture, and orientation, which are illustrated in Figure 1.1. The feature that is common in all of the state-of-the-art salient-object detectors is color contrast. Texture and orientation cues are used only by LR and CH. Although LR is one of the best uniqueness-based methods, spatial-connectivity-based methods that only use color, such as GMR, AMC, and HSD, perform better. We draw two conclusions from these comparisons. First, the mechanism that is used for salient-object detection is as important, if not more, than the visual features. Second, the well-known datasets do not include a variety of aspects of visual saliency; there are not sufficient number objects that are distinctive in terms of texture. For example, the precision-recall curves of the best performing methods on the MSRA-1000 and the SED-100 datasets in Figure 2.20 are already very close to the ideal curve. In Chapter 3, we introduce our own dataset and in Chapter 3 that represents natural images better than the well-known datasets. We then compare the state-of-the-art methods on this dataset in Chapter 5. We show that not only the mechanism, but also the variety of visual features is important for accurate salient-object detection.

The limitations of the state-of-the-art salient-object detectors are similar to those

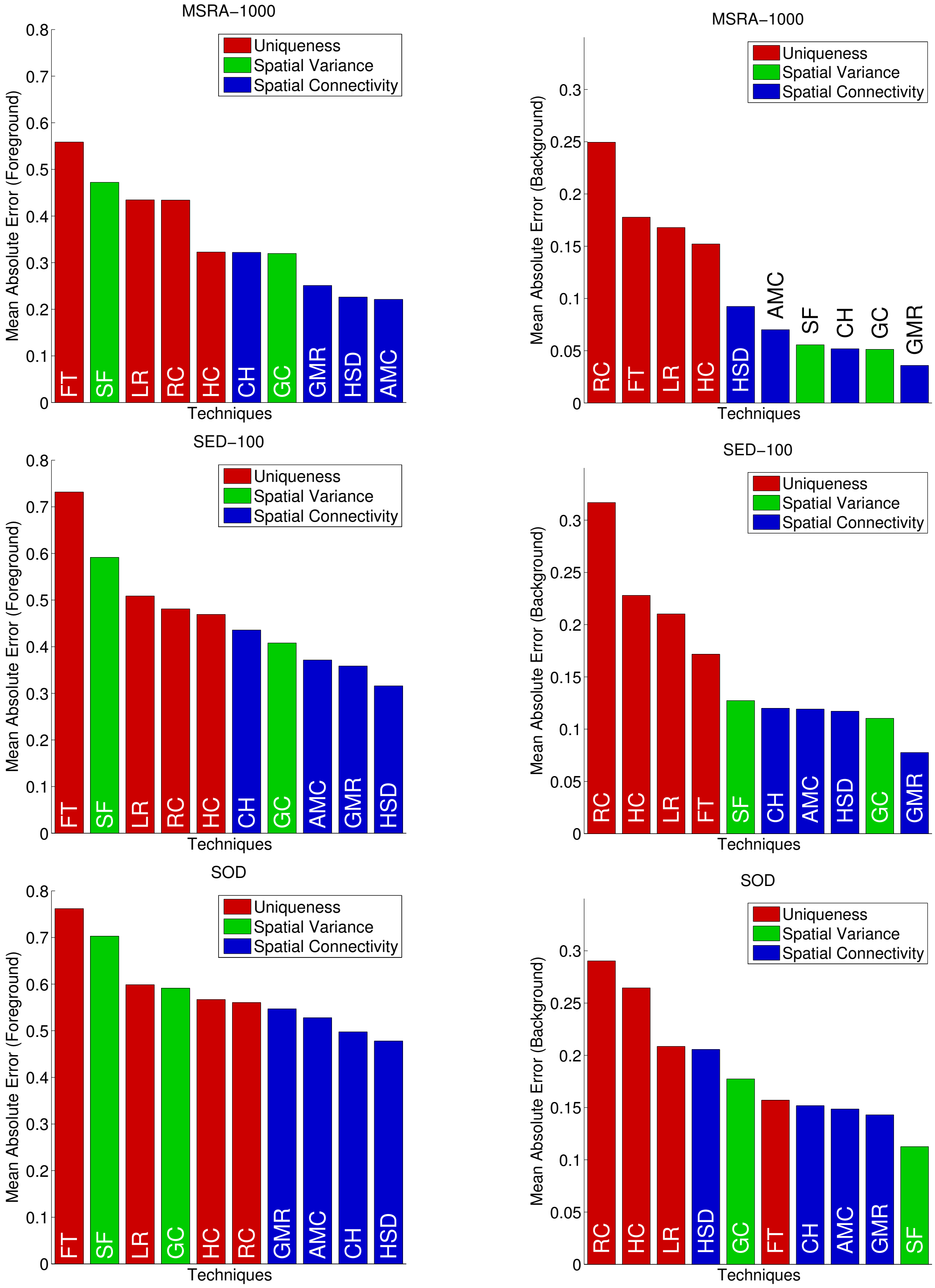


Figure 2.21: The mean absolute error values of the salient object detectors on the well-known datasets.

of the well-known datasets, which are explained in Section 2.1. Global uniqueness methods FT, HC, and LR, and the methods that use a fixed local-global contrast adjusting parameter, such as RC, SF, and GC, assume that there is only a single salient object. Although, this assumption is avoided by using spatial connectivity, HSD, CH, AMC, and GMR still do not take the varying saliency of objects into account, i.e. they assume all salient pixels are equally important. In Chapter 5, we show that these assumptions cause a limited performance in multi-level object saliency by evaluating their performance on our new dataset, which is introduced in Chapter 3.

2.5 Summary of the Chapter

In this chapter, we have reviewed the well-known datasets with binary ground-truth maps, the state-of-the-art salient-object detectors, and their performances. We have discussed that the well-known datasets have a limited representation of the natural images, because the majority of the images in these datasets include a single salient-object and varying saliency of objects is overlooked, i.e. all salient pixels are assumed to be equally important. We have grouped the state-of-the-art salient object detectors with respect to the way they use image information as uniqueness, spatial variance, and spatial connectivity. We have explained the advantages and limitations of these detectors in estimating object saliency. Finally, we have evaluated the performance of the salient object detectors on the well-known datasets. We have showed that these methods are designed to operate on simple images and on binary object saliency. Therefore, even though they can be modified to estimate multi-level object saliency, as we show in Chapter 5, they have a limited performance with their current state.

3 Comprehensive Object-Saliency Dataset

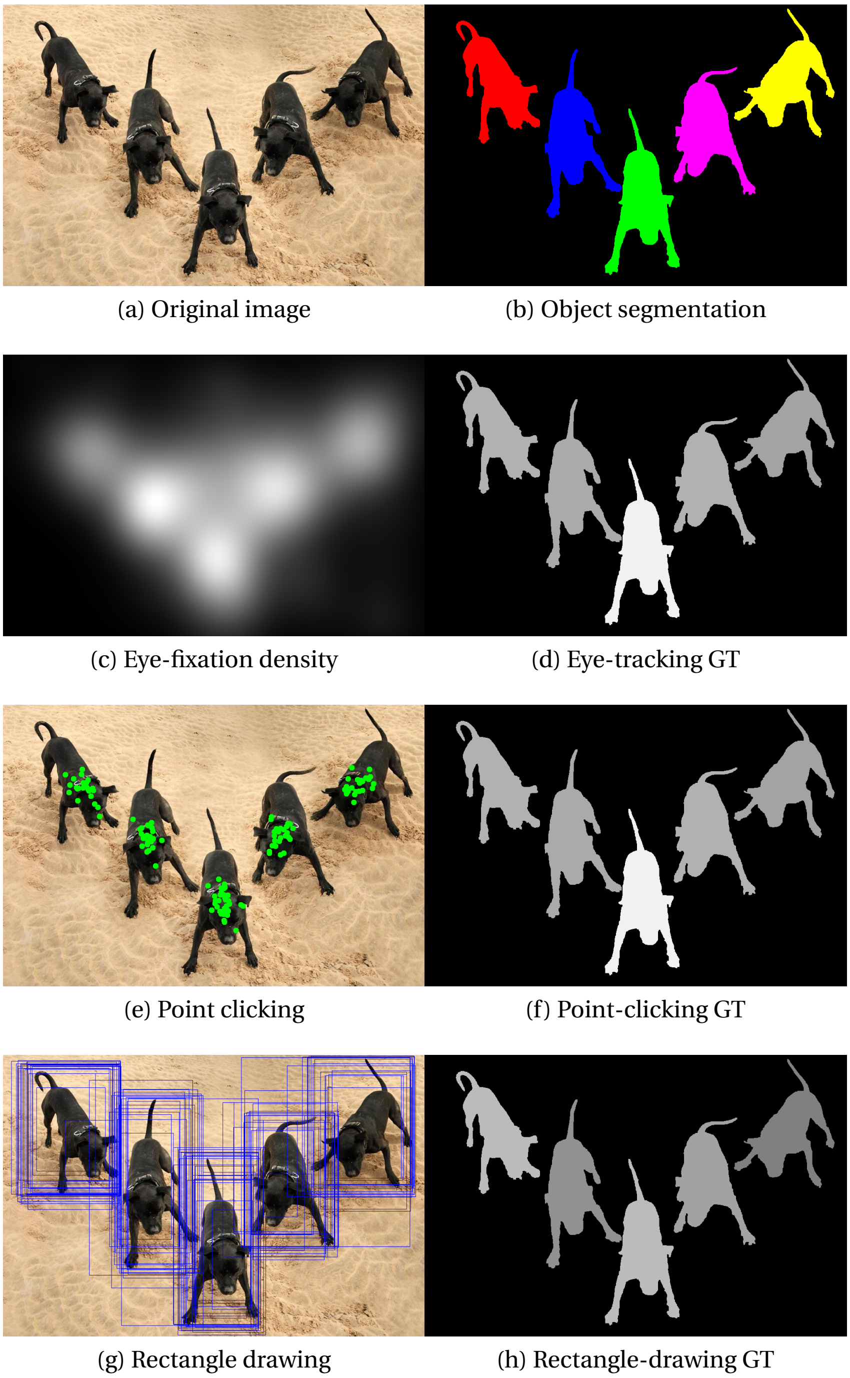
In Section 2.1, we showed that the well-known evaluation datasets, such as the MSRA-1000 [10], the SED-100 [11], and the SOD [12], have limitations in correctly measuring the multi-level saliency of objects in images. The ground-truth maps of these datasets mark only the most prominent object(s) as salient and the surrounding background as non-salient, which results in two saliency levels. This approach assumes that the objects in an image are either of equal saliency, or not salient at all, which is not representative enough for natural images.

In contrast, as shown in Figure 3.1(a), the natural images we take every day have multiple visually-distinctive objects that have different levels of saliency depending on their characteristics and respective surrounding contexts. In Section 1.1 and 1.2, we discussed the subjectivity of visual saliency and, to measure object saliency, we indicated the need for experimental data. Therefore, we collected a Comprehensive Object Saliency (COS) dataset, which contains 588 images with multiple salient objects per image (2434 objects in total), and we conducted three subjective experiments on the dataset for measuring multi-level object saliency. An overview of our dataset is illustrated in Figure 3.1. The subjective experiments and the corresponding experimental data are described as follows:

- **Eye-tracking experiments:** Subjects were asked to freely view the images on a monitor and their eye fixations are recorded using eye-tracker equipment. An example fixation map is illustrated in Figure 3.1(c). This experiment was conducted to measure the effect of *fixation duration* on object saliency, which was discussed in Section 1.1.

- **Point-clicking experiments:** Subjects were asked to click on the objects that they notice at first glance. A set of clicked points are illustrated in Figure 3.1(e).

- **Rectangle-drawing experiments:** Subjects were asked to draw tight rectangles around the objects that they notice at a first glance. A set of rectangles are illustrated in Figure 3.1(g). We conducted the point-clicking and rectangle-drawing experiments to measure the *collective human attention* on object saliency, which was discussed in Section 1.2.



(a) Original image

(b) Object segmentation

(c) Eye-fixation density

(d) Eye-tracking GT

(e) Point clicking

(f) Point-clicking GT

(g) Rectangle drawing

(h) Rectangle-drawing GT

Figure 3.1: In order to measure the saliency level of an object, (a) for each image in our dataset, we collected (c,e,g) three different types of experimental data and (b) we manually segmented the objects. The experimental data and the segmentations are used to generate (d,f,h) multi-level ground-truth maps for each subjective experiment (GT: Ground Truth).

In order to accurately measure object saliency, we require pixel-level information on the objects in our dataset. Therefore, we manually and **separately** segment out each object that were attended (fixated, clicked, or drawn) during the subjective experiments. Object segmentation is represented in Figure 3.1(b) with different colors on each separate object. Manual segmentations and measured multi-level saliency values are used to generate multi-level ground-truth maps in Figure 3.1(d)(f) and (h).

The tasks in the subjective experiments represent different levels of involvement from human attention. Eye-tracking experiments are not associated with any task and conducted in a free-viewing fashion. Point clicking introduces a clicking task and involves positional awareness of an object. The rectangle drawing brings in the object size/scale concept, as it is necessary to tightly fit a rectangle. Our experiments measure multi-level object saliency at various cognitive levels as shown in Table 3.1.

Table 3.1: Involvement of human attention in the subjective experiments in the COS dataset.

	Subconscious ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ Focused		
	Free Viewing	Object Position	Object Size
Eye Tracking	✓		
Point Clicking	✓	✓	
Rectangle Drawing	✓	✓	✓

In this chapter, we present a new image dataset, where each image has multiple salient objects, three types of experimental data, separate segmentation masks for the salient objects, and multi-level ground-truth maps. The rest of the chapter is divided into two parts. First, we explain the dataset collection, the object segmentation, and the subjective experiments in detail and we define a method for measuring object saliency. Second, we show that object saliency is multi-level, i.e. all objects are not equally important, and we investigate the visual characteristics that can affect the saliency level of an object.

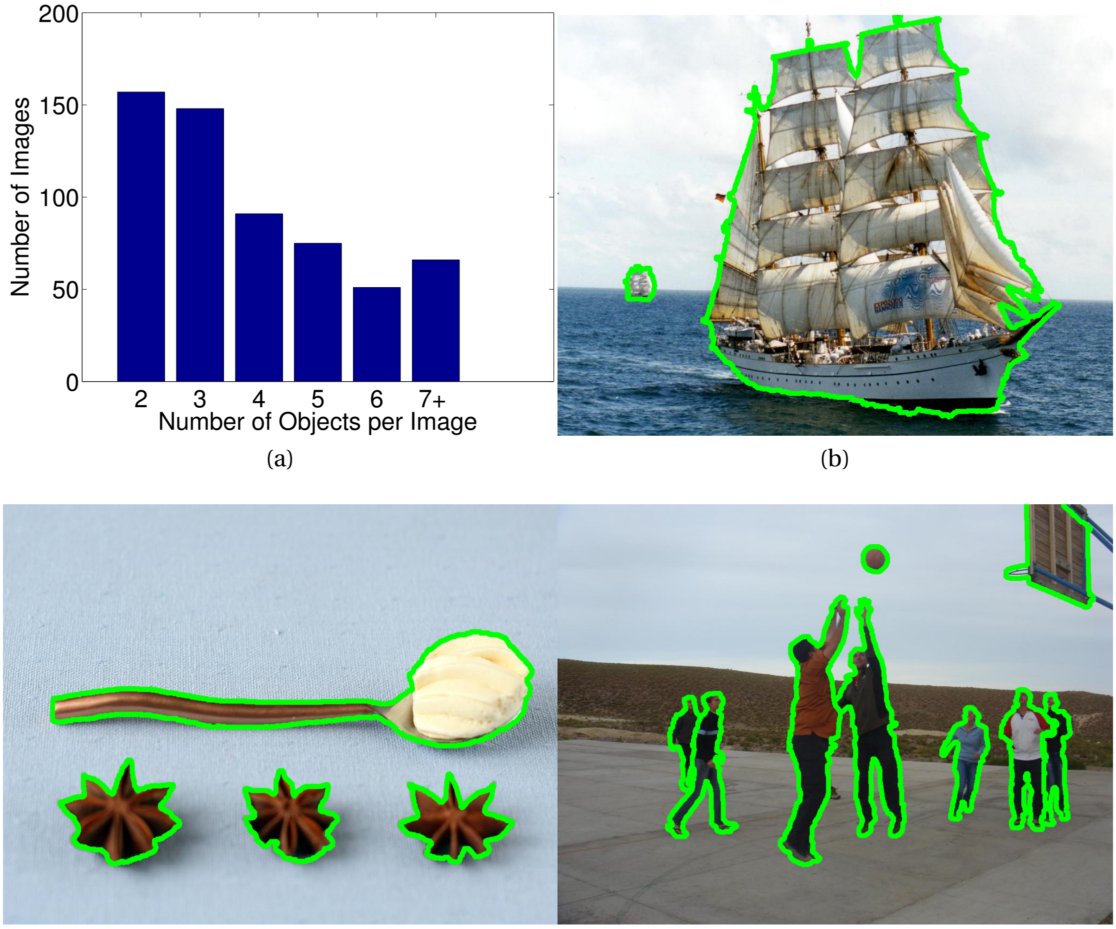
3.1 The Data Collection

In order to form our dataset, we go through the ImageNet [35] database, where the object bounding boxes are provided for a certain number of images. Among them, we selected 588 natural images that have multiple objects of interest and retrieved their original high-resolution versions[1] from Flickr[2]. Each image in our dataset includes more than one salient object, which gives us a total of 2434 objects. In Figure 3.2(a), we illustrate the distribution of object count in our dataset.

The images in our dataset are selected based on a variety of object sizes, shapes, and aspects of saliency. In Figure 3.3, we illustrate the distribution of objects with respect to several parameters. These parameters are explained in Section 3.1 and 3.2.

[1] Larger dimension of all of the images in our dataset is 1024 pixels.

[2] http://www.flickr.com



(a)

(b)

(c)

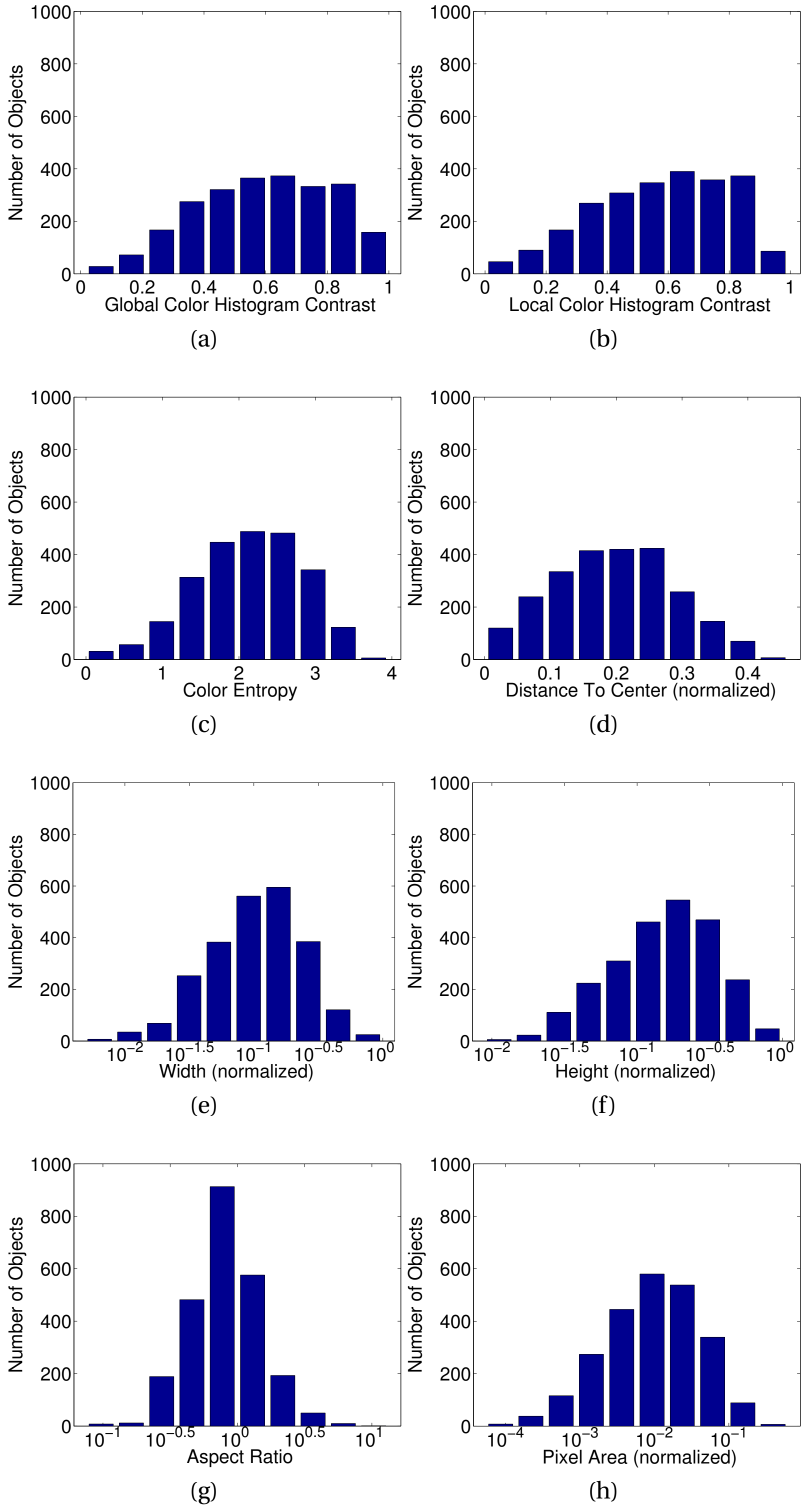
(d)

Figure 3.2: (a) Number of images with given number of objects and some example images from our dataset with (b) 2, (c) 4, and (d) 7+ salient objects (Salient objects are enclosed with a green border for clarity).

Object saliency is subjective. In order to correctly measure the saliency value, we first segment the objects, which were attended by subjects during our experiments, with their precise boundaries. We then use the results of the three subjective experiments: eye tracking, point clicking, and rectangle drawing. The naming convention we use for subjective experiments is summarized in Table 3.2.

Table 3.2: Naming convention that is used for subjective experiments.

Experiment Name	**Alias**	**Data**	**Saliency Value**
Eye-Tracking	et	$\mathbf{D}_{et}$	s_{et}
Point-Clicking	pc	$\mathbf{D}_{pc}$	s_{pc}
Rectangle-Drawing	rd	$\mathbf{D}_{rd}$	s_{rd}



(a)

(b)

(c)

(d)

(e)

(f)

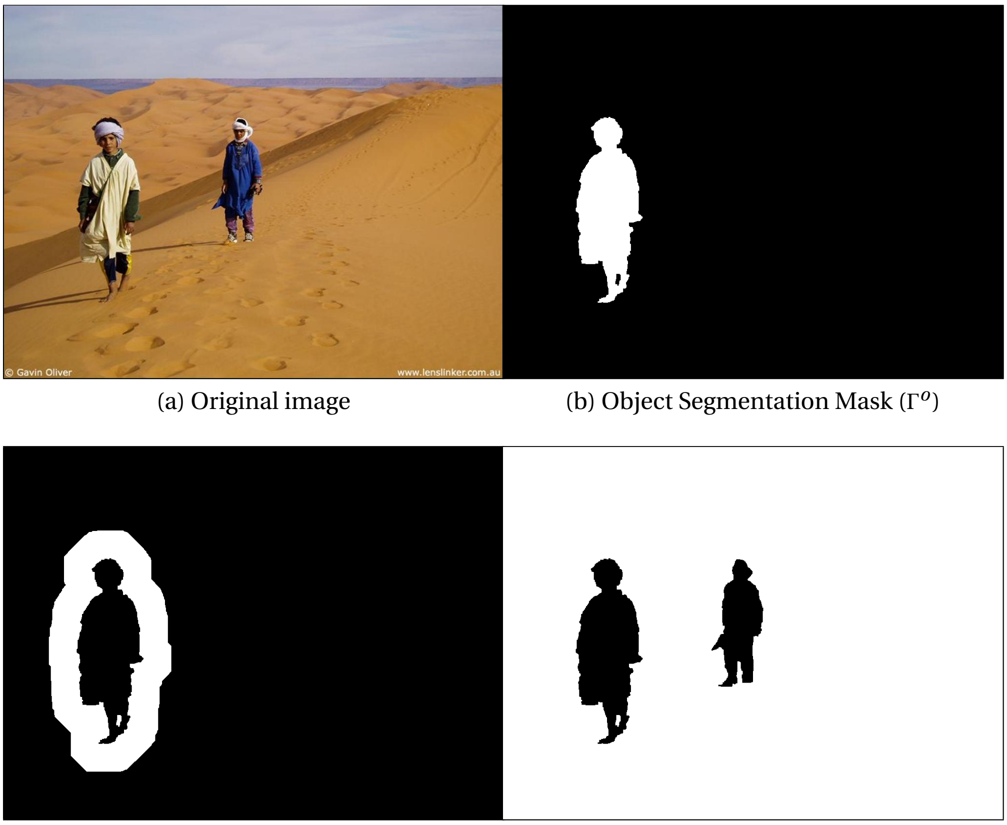
(g)

(h)

Figure 3.3: Distribution of the objects in our dataset with respect to several parameters.

3.1.1 Object Segmentation

As we investigate object saliency, we need to know which pixels belong to which object or if they belong to background. In order to find the objects of interest, we analyze the experimental data (eye-tracking, point-clicking, and rectangle-drawing), determine the objects that received satisfactory amount of attention[3] and separately segment them with pixel-precise outlines using an interactive segmentation tool [72]. The segmentation mask Γ^o of an object o, as well as the masks for local and global neighborhoods are illustrated in Figure 3.4. The local neighborhood mask Γ^o_l is obtained using morphological dilation and binary operations on Γ^o so that it covers the same area and has a shape similar to Γ^o. The global mask Γ_g is obtained by removing all salient object masks from the image. We use these masks to measure object saliency and compute object-based features for our analyses. We follow the same mask naming convention throughout the thesis.



(a) Original image

(b) Object Segmentation Mask (Γ^o)

(c) Local Neighborhood Mask (Γ^o_l)

(d) Global Neighborhood Mask (Γ_g)

Figure 3.4: For each object in (a) an image, we compute segmentations masks for (b) the object, (c) its local surrounding, and (d) its global surrounding (black = 0, white = 1). Note that, global surrounding excludes other salient objects.

[3] An object is segmented if more than three persons attended.

3.1.2 Eye-Tracking Experiments

We performed our eye-tracking experiments using RED250[4] infrared eye-tracking device with a sampling frequency of 250 Hz. In total, we collected eye-tracking data from 95 people, (48% women, 52% men) within ages 18-34. Each person in the experiment was asked to freely view 200 images (two experiments with 100 images each) on a screen with a resolution of 1680×1050 pixels. The experiment subjects were given five seconds to view each image followed by an empty gray screen for two seconds. Each image was viewed by 24 subjects on average (min: 13, max: 34).

The data collection setup is illustrated in Figure 3.5. We convert the fixation points of a person δ for an image k into a fixation-density map $\mathbf{D}_{et}^{\delta,k}$ using a Gaussian filter. The σ parameter, which is equal to the radius of the circle of fovea, of this filter is calculated using (3.1).

$$\sigma = d_v \cdot \frac{r_v}{h_m} \cdot \big(\tan(\alpha + \eta + \theta) - \tan(\theta)\big) \tag{3.1}$$

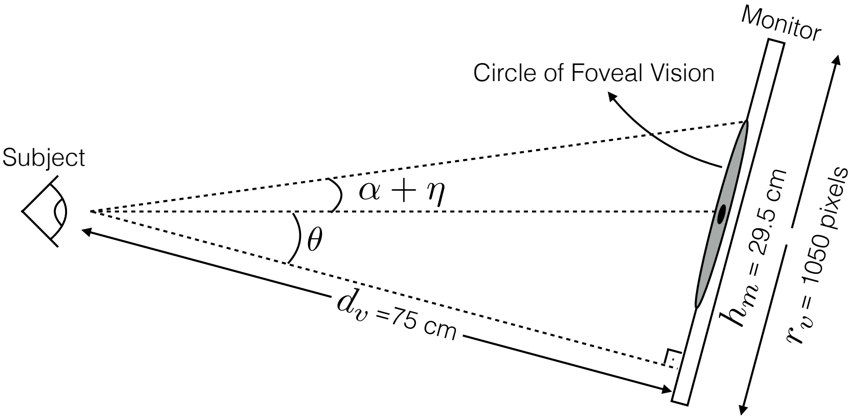


Figure 3.5: The configuration of a subject and the monitor during the eye-tracking experiments.

Here, $\alpha = 1°$ is the half size of the human fovea, $\eta = 0.4°$ is the accuracy of the eye tracker, $d_v = 75$ cm is the viewing distance, $r_v = 1050$ pixels is the vertical resolution of the monitor, and $h_m = 29.5$ cm is the height of the monitor. In our experiments $\sigma \approx 66$. This value slightly changes from person to person, depending on the viewing distance output of the eye-tracker device. After the Gaussian filtering, we obtain the fixation-density maps $\mathbf{D}_{et}^{\delta,k}$.

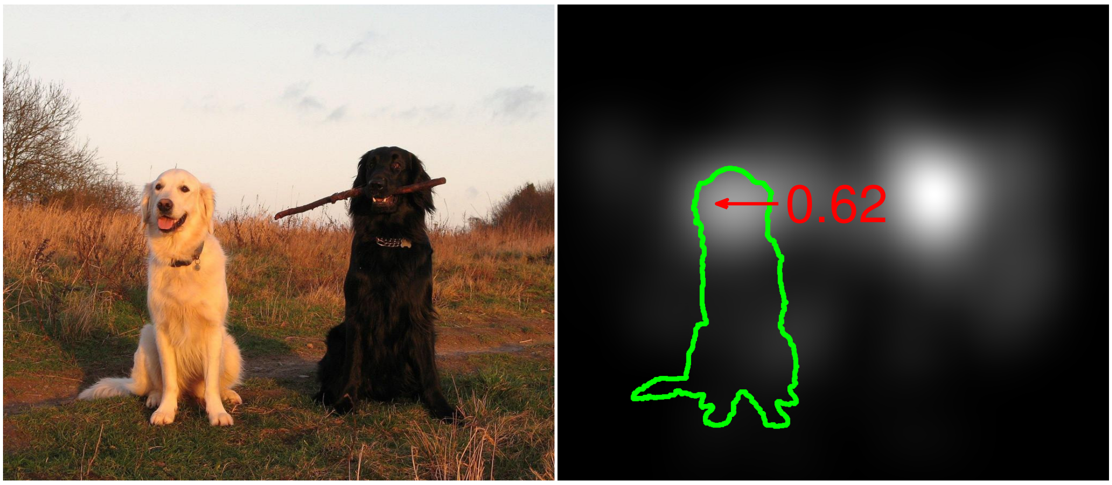
Fixation-density maps indicate the visual-saliency level of the object in an image through fixation duration (see Section 1.1). We use these maps to measure the "eye-tracking saliency value", specifically s_{et}^{o}, of an object o in an image. It is possible that the experiment subjects were fixated different parts of the same object. In order

[4] http://www.smivision.com/en/gaze-and-eye-tracking-systems/products/red-red250-red-500.html

to correctly measure s_{et}^{o}, we use the maximum fixation-density values within the boundary of object o as follows:

$$s_{et}^{o} = \frac{\sum_{\delta} \max_{i \in \Gamma^{o}} \left(\mathbf{D}_{et}^{\delta,k}(i) \right)}{N_{et}^{k}} \tag{3.2}$$

Here N_{et}^{k} is the number of people who viewed the image k during the eye-tracking experiments and i is a pixel inside the object boundary ($i \in \Gamma_o$). An illustration of this operation is given in Figure 3.6.



(a)

(b)

Figure 3.6: (a) Original image and (b) the corresponding fixation-density map for a person δ. The maximum value inside the boundary (green line) of the white dog, namely $\max_{i \in \Gamma^{o}} \left(\mathbf{D}_{et}^{\delta,k}(i) \right)$, is equal to 0.62. The eye-tracking saliency of the white dog is equal to the average of maximum values of all subjects who viewed the image.

3.1.3 Point-Clicking Experiments

The second subjective experiment we conducted is called “point-clicking". In order to perform these experiments, we used a crowd-sourcing web site[5]. As the meaning of the word “saliency" can be different under scientific context, we asked people to click on the objects that they “notice at first glance", without analyzing the image for a long time. The task duration was limited to 30 minutes and at each task, 42 images were shown one at a time. Crowd-sourcing experiments leave approximately 45 seconds to the subjects per image, which is sufficient as far as analyzing the image, clicking, and internet connection speed is concerned. In this experiment, each image was viewed by 33 people on average (min: 24, max: 38).

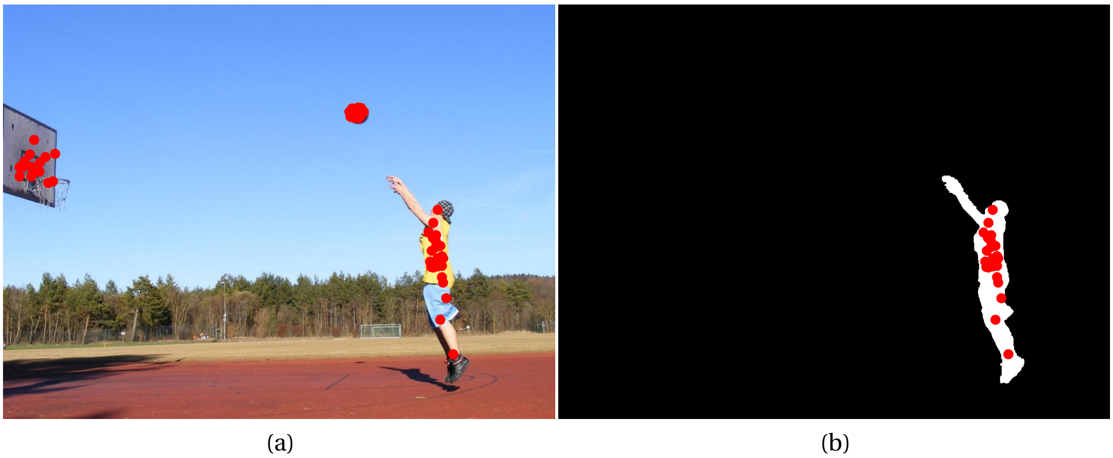
We represent the set of points where person δ clicked on image k as $\mathbf{D}_{pc}^{\delta,k}$. In order

[5] http://www.shorttask.com

to measure the “point-clicking saliency value", namely s^o_{pc}, of an object o, we count the number of people who clicked an object and normalized it with the number of subjects who viewed the image. Formally, it can be calculated as follows:

$$
\begin{aligned}
s^o_{pc} &= \frac{\sum_\delta f(\Gamma^o, \mathbf{D}^{\delta,k}_{pc})}{N^k_{pc}} \\
f(\Gamma^o, \mathbf{D}^{\delta,k}_{pc}) &= \begin{cases} \exists i \in \mathbf{D}^{\delta,k}_{pc} \quad | \quad i \subset \Gamma^o, & 1 \\ \text{else}, & 0 \end{cases}
\end{aligned}
\tag{3.3}
$$

Here, N^k_{pc} is the number of people who viewed the image k during point-clicking experiments. An illustration of this operation is given in Figure 3.7.



(a)

(b)

Figure 3.7: (a) The point clicks (in red dots) are overlaid on the original image and (b) the points inside the object boundary are selected. There are 23 subjects who clicked on this object and 30 subjects viewed this image, which makes the point-clicking saliency value of the human in the image $s^o_{pc} = 23/30 \approx 0.76$.

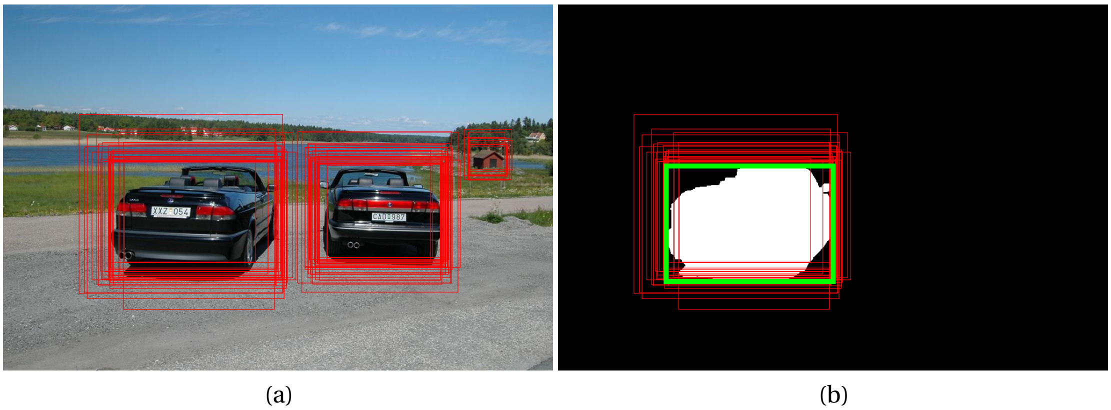
3.1.4 Rectangle-Drawing Experiments

Similar to point-clicking experiments, we performed rectangle-drawing experiments using crowd sourcing. We asked people to draw a tight rectangle on the objects that they notice at first glance, without analyzing the image for a long time. The task duration was limited to 30 minutes and at each task 42 images were shown. This experiment is referred to as “rectangle drawing" in the rest of the paper. In this experiment, each image was viewed by 32 people on average (min: 15, max: 50).

We represent the set of rectangles where person δ drew on image k as $\mathbf{D}^{\delta,k}_{rd}$. In order to measure the “rectangle-drawing saliency value", namely s^o_{rd}, of an object o, we count the number of people, who could draw a rectangle on an object with an intersection-over-union [73] scores greater than 0.3 and normalize it using the number of subjects who viewed the image. We can calculate this value as follows:

$$
\begin{aligned}
s_{rd}^{o} &= \frac{\sum_p f(\Gamma^o, \mathbf{D}_{rd}^{p,k})}{N_{pc}^{k}} \\
f(\Gamma^o, \mathbf{D}_{rd}^{p,k}) &= \begin{cases} \exists r \in \mathbf{D}_{rd}^{p,k} | g(r, r^o) \geq 0.3, & 1 \\ \text{else}, & 0 \end{cases}
\end{aligned}
\tag{3.4}
$$

Here, N_{pc}^{k} is the number of people who viewed the image k during rectangle-drawing experiments, r is a rectangle in $\mathbf{D}_{rd}^{p,k}$, $g(.,.)$ is a function that computes the intersection-over-union score, and r^o is the reference rectangle that tightly encloses the object we segmented before. An illustration of this operation is given in Figure 3.8.



(a)

(b)

Figure 3.8: (a) The subjective rectangles, $\mathbf{D}_{rd}^{p,k}$, (in red frames) are overlaid on the original image and (b) the rectangles that have a intersection-over-union score greater than 0.3 with respect to the reference rectangle r^o (green) are selected. In this case, 27 subjects drew rectangles and 35 subjects viewed this image, which makes the rectangle-drawing saliency value of the car on the left in the image $s_{rd}^{o} = 27/35 \approx 0.77$.

3.1.5 Discussion on Saliency Values

In Section 1.1, we discussed that humans fixate on the objects that are more informative about the scene for a longer period of time compared to other parts of an image. Motivated by this, in our eye-tracking experiments, we take fixation duration as a measure of saliency, which expects saliency and informativeness to be correlated. In Chapter 1, we defined saliency as low-level distinctiveness of an item compared to its surrounding area. We thus can use informativeness as a measure of saliency, as long as the low-level vision controls the attention. In [74], Alers et al. compare the eye-fixation trends of subjects under two conditions: free-viewing and task-driven. They observe that subjects have a tendency to fixate more on the objects of interest, when they are not given a certain task (free-viewing condition). Moreover, this effect continues for more than five seconds. Therefore, as low-level factors are more prominent in a free-viewing condition, we can use our eye-tracking saliency value s^o_{et} to represent the effect of fixation duration.

Eye-tracking experiments have been extensively studied [3, 25–28, 39–43] and have been used in psychophysics [44–49] as an indicator for human attention. Point-clicking experiments measure visual saliency with a different approach. It requires a subject to voluntarily move the mouse and to perform the click. This significantly reduces the strong center bias in eye-tracking data and eliminates noise due to the various factors (involuntary eye-movements, experiment fatigue etc.). Note that, point-clicking data cannot replace eye-tracking data, because there is no duration information in clicked points. However, it can be very useful for resolving the object-level attention ambiguity, especially when two or more objects are spatially very close to each other in an image.

As we discussed in Section 1.2, object saliency is subjective. Even though a single person finds certain objects to be equally salient, the collective idea of a group of people would significantly differ. When we measure object saliency, we take the subjectivity of saliency into account by evaluating collective human attention. In our point-clicking experiments, we asked a group of people to click on the noticeable objects. The subjects of this experiment followed various strategies, such as clicking on all of the objects, only on the most prominent object, or on the objects that are closer to the camera. We measure the overall trend on object saliency by considering the ratio of the number of people who clicked on an object to the number of people who viewed an image as the collective human attention.

In another crowd-sourcing experiment, we asked subjects to draw rectangles on the noticeable objects. Although we use both the point-clicking and the rectangle-drawing experiments to measure the collective human attention, there are two major differences between them. First, the rectangle-drawing experiments have a higher-level of involvement in the experimental task, because subjects need to judge the size of the object. Second, the rectangle data specifies which object is considered salient more clearly compared to clicked points. For example, a point on a human face can imply the saliency of both the face and the body. This ambiguity can be resolved by examining the rectangle data.

We use point-clicking s^o_{pc} and rectangle-drawing s^o_{rd} saliency values as a measure of collective human attention. Note that, the standard deviation of the collective opinion is correlated to the collective human attention. When the majority of the subjects agrees on the significance (or insignificance) of an object in an image, saliency value of that object approaches to 1 (or 0), which minimizes the variance of opinions. Whereas, for a saliency value of 0.5, the subjects are equally divided into two groups, which implies a strong disagreement on object saliency.

One might argue that the order of attention, i.e. the order a subject looked at, clicked on, or drew on an object, can be another measure of multi-level object saliency. Although this can be a driving factor for certain cases, such as a significantly large or very uniquely-colored objects, the order of attention is considerably affected by the defined task in a subjective experiment. For example, in our point-clicking experiments, we asked subjects to click on the objects that they notice at a first glance. The clicked points of five persons and their clicking orders are superimposed on an image in Figure 3.9. Here, subjects usually clicked in a horizontal order: from left to right. One reason for this could be the convenience of a subject to minimize the spatial distance between mouse clicks. The order of attention introduces an undesired bias in saliency measurements, such that the objects on the left seem to be more salient than the objects on the right. It is possible to remove this bias by asking subjects to click on the objects in the order of saliency. However, this creates a problem similar to that of the SOD dataset (see Section 2.1.2), where high-level semantic context overrides the effect of low-level visual cues and thus creates another bias. In our dataset, we did not consider the order of attention as a measure of multi-level object saliency.

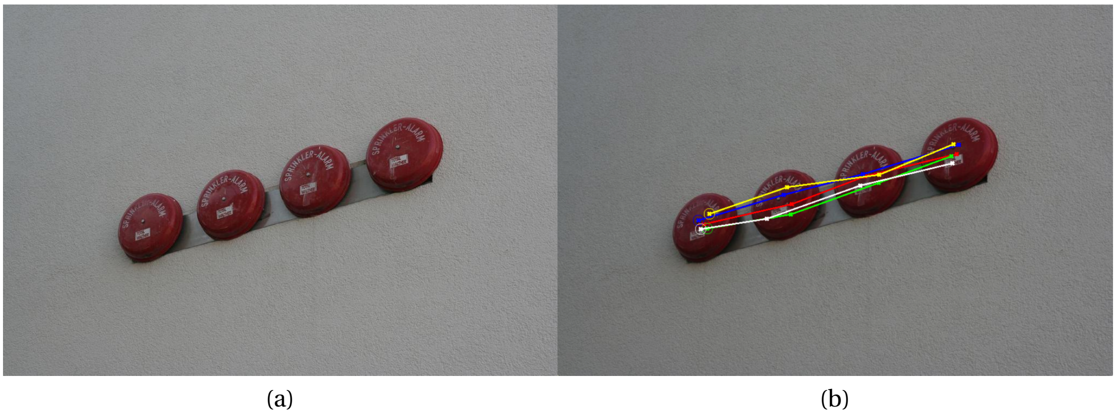


Figure 3.9: In a subjective experiment, subjects are asked to click on the noticeable objects (a) in images. (b) Their clicks have a tendency to follow an order from left to right, which might introduce an undesired bias when we measure object saliency.

3.1.6 Multi-Level Ground-Truth Maps

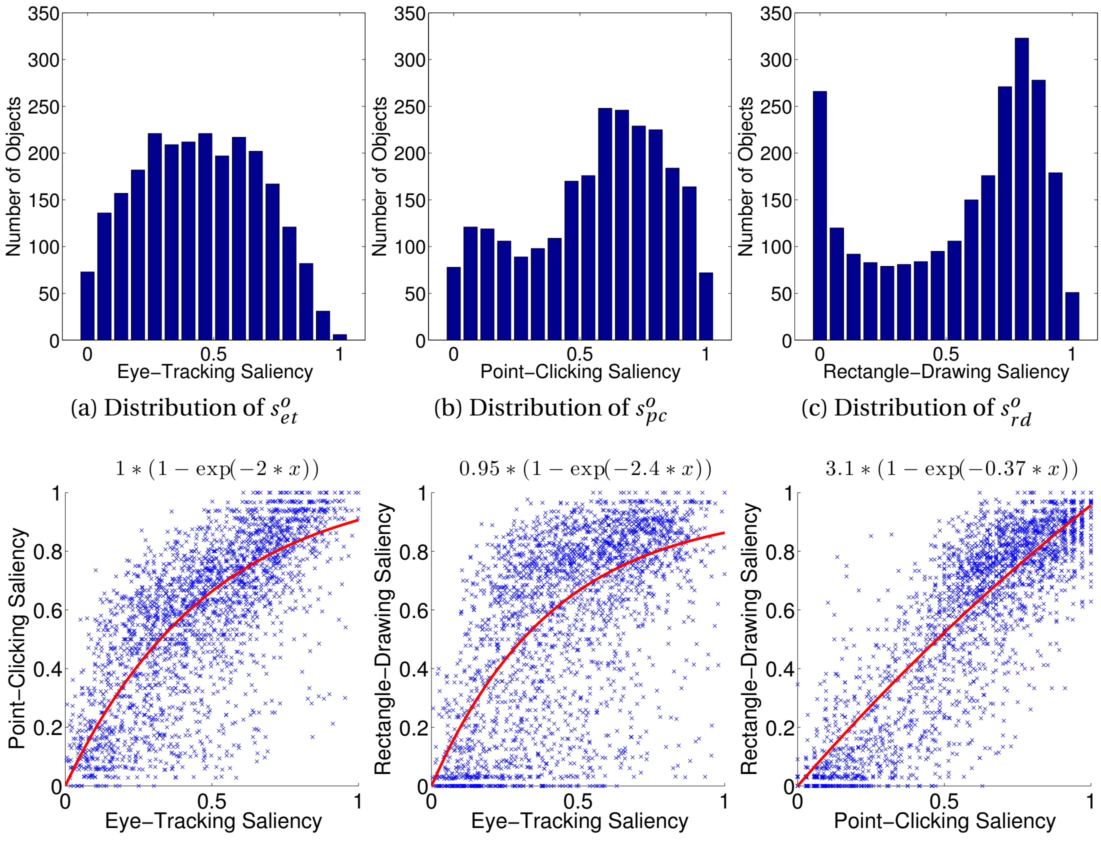
After we calculate the eye-tracking (s^o_{et}), point-clicking (s^o_{pc}), and rectangle-drawing saliency (s^o_{rd}) values of all objects, we can generate multi-level ground-truth maps for an image as follows:

$$\mathbf{M}^k_\gamma = \sum_{\forall o \text{ in image } k} s^o_\gamma \cdot \Gamma^o \tag{3.5}$$

Here, $\mathbf{M}^k_\gamma$ is the multi-level ground-truth map of the image k using γ-type saliency values, where $\gamma = \{et, pc, rd\}$ is one of the subjective experiments. Consequently, the CSD dataset includes three different ground-truth maps for each image.

3.2 Visual Saliency of Objects

We use the experimental data and manual object segmentations for measuring the saliency levels of objects. Each object in the COS dataset has three types of multi-level object-saliency values between 0 and 1 ($s^o_{et}, s^o_{pc}, s^o_{rd}$). The distributions of these saliency values are illustrated in Figure 3.10. When we take the informativeness of fixation duration and collective human attention into consideration, saliency values of objects are not restricted to binary values, 0 (non-salient) and 1 (salient). In natural images with multiple objects, the importance of salient objects vary, i.e. all objects are not equally salient.



(a) Distribution of s^o_{et} (b) Distribution of s^o_{pc} (c) Distribution of s^o_{rd}

(d) $R^2 = 0.64$ (e) $R^2 = 0.48$ (f) $R^2 = 0.79$

Figure 3.10: (a) The distributions of the eye-tracking (s^o_{et}), point-clicking (s^o_{pc}), and rectangle-drawing saliency (s^o_{rd}) values of the objects in our dataset. (d-f) The coefficient of determination between different saliency values. Red lines indicate the gamma non-linearity.

The measured saliency values of an object are different for each experiment. In Figure 3.10(d)-(f), we show the correlation between eye-tracking, point-clicking, and rectangle-drawing saliency-values of the objects in our dataset in pairs. For each pair, we fit a gamma non-linearity that maps the values from the x-axis to the y-axis. The coefficients of determination (R^2) for these functions are illustrated in Figure 3.11. The hierarchical order of human involvement in our the subjective experiments, which was shown in Table 3.1, is reflected to the correlation values between saliency types.

The saliency values of objects strongly depend on their low-level visual characteris-

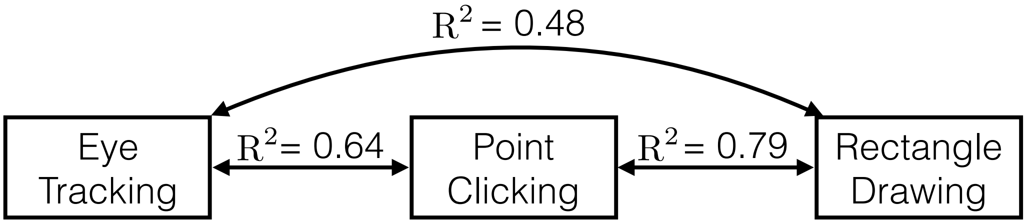


Figure 3.11: The values of the coefficients of determination is in accordance with the ordering of attention involvement in subjective experiments.

tics, such as color, shape, and texture. Here, we analyze the relationship between these features and object saliency.

3.2.1 Color

Color is one of the very prominent factors that plays a role in the visual saliency of an object. In order to see the distribution of colors of salient objects and their surrounding regions, we calculate their average color in CIELAB color space as follows:

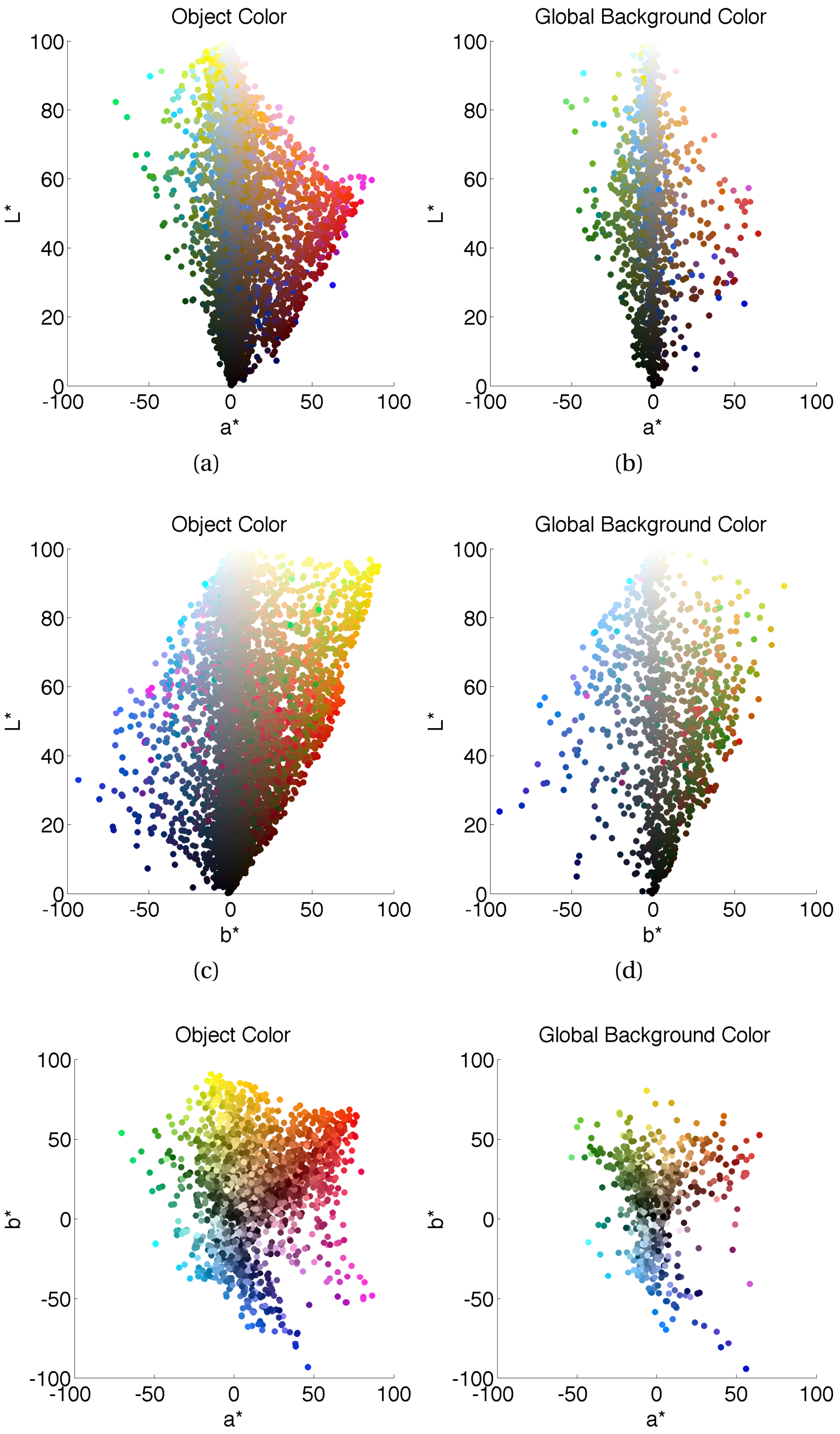
$$\begin{aligned}
\mathbf{c}^o &= \frac{1}{|\Gamma^o|} \sum_{i \in \Gamma^o} \mathbf{c}(i) \\
\mathbf{c}_l^o &= \frac{1}{|\Gamma_l^o|} \sum_{i \in \Gamma_l^o} \mathbf{c}(i) \\
\mathbf{c}_g &= \frac{1}{|\Gamma_g|} \sum_{i \in \Gamma_g} \mathbf{c}(i)
\end{aligned} \tag{3.6}$$

Here, $\mathbf{c}^o, \mathbf{c}_l^o$ and $\mathbf{c}_g$ are the average colors of the object o, and its local and global surroundings, respectively; and $\mathbf{c}(i)$ is the color of the i^{th} pixel. The average colors are computed using the segmentation masks, $\Gamma^o, \Gamma_l^o, \Gamma_g$, in Figure 3.4.

We can see from Figure 3.12, the average color of the salient objects are spread out over the color space, including very saturated colors. We benefit from this result in salient-object detection by learning a saturation-based feature in Chapter 5. In addition to saturation, certain colors, such as red and yellow, appear more frequently compared to blue and green. This difference has been employed in salient-object detection as "warm color prior" in [14].

In contrast, if we look at the average global-background colors, we can see that the color variety is much smaller compared to average object-colors. In addition, certain colors imply a priori semantics about the non-salient regions, such as sky, sea (tones of blue), vegetation (tones of green), soil (tones of brown), rock, and cloud (tones of gray). A semantic image-segmentation algorithm could be a useful pre-processing step to discarding non-salient image regions. Note that, depending on the image context, a puddle of water, a patch of sky or grass could also be salient.

In an image, the objects that have perceptually very different colors compared to



(a) (b)

(c) (d)

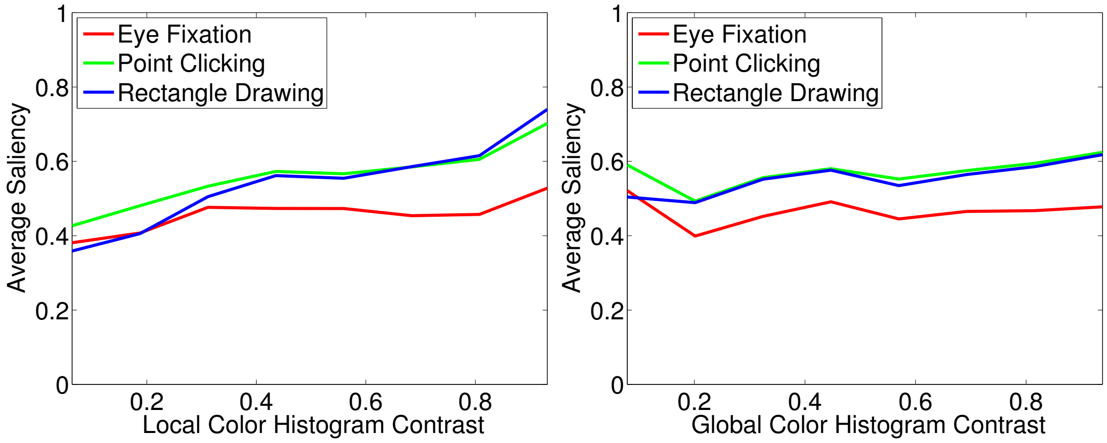
(e) (f)

Figure 3.12: (a,c,e) The colors of the salient objects and (b,d,f) the colors of the background regions in CIEL*a*b* color space.

their surroundings stand out and are easily noticed by humans. Therefore, we analyze how local and global color-histogram contrast affect the visual saliency of objects. The color-contrast values are calculated as follows:

$$\begin{aligned} \text{Global Color-Contrast} &= \chi^2(\mathbf{h}^o, \mathbf{h}^o_l) \\ \text{Local Color-Contrast} &= \chi^2(\mathbf{h}^o, \mathbf{h}^g) \end{aligned} \tag{3.7}$$

Here, χ^2 computes the Chi-Squared distance between two histograms, $\mathbf{h}^o$, $\mathbf{h}^o_l$, and $\mathbf{h}^g$ are $4 \times 4 \times 4$ CIELa*b* color histograms of the object, its local and global surroundings, respectively. In Figure 3.13, we can see that local contrast is directly related to the visual saliency. Whereas, global contrast has a weaker effect on saliency, due to the inclusion of a wider variety of colors.



(a) (b)

Figure 3.13: The relationship between the multi-level saliency values of the objects in our dataset and their (a) local and (b) global color contrast.

3.2.2 Shape and Position

The shape and position of an object can significantly affect its distinctiveness. In order to show this, we extract the shape and position features in Table 3.3. These features are illustrated in Figure 3.14.

Table 3.3: Shape and position-based features.

Feature Name	**Description**
Aspect Ratio	The ratio between width and height of an object
Compactness	The concavity of an object
Distance to Center	The distance of an object from the image center
Pixel Area	The number of pixels in an object

In Figure 3.15, we illustrate how shape and position features affect the saliency of

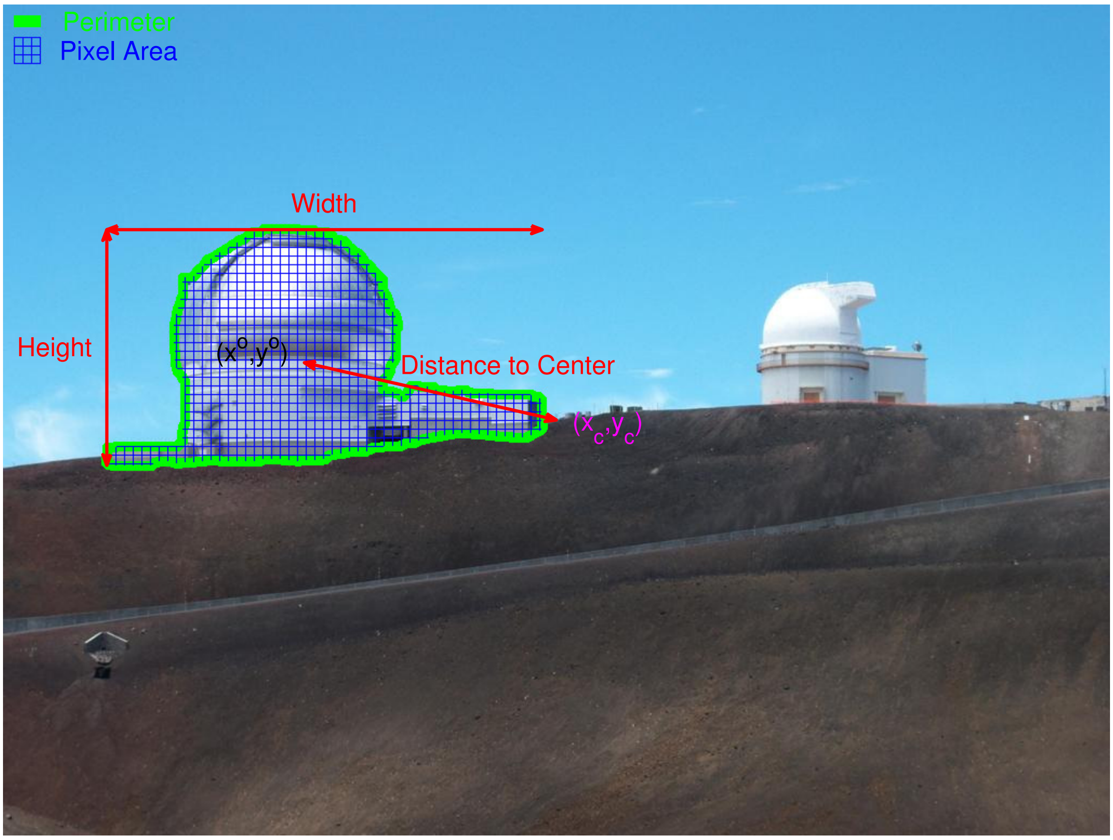


Figure 3.14: Representation of the features that are related to the position and the shape of a salient object.

an object. From these results, we draw the following conclusions:

- Objects with aspect ratios that are close to one tend to be visually more salient compared to objects that are elongated in horizontal or vertical directions.
- The features that are related to the size of the objects (perimeter, width, height, and pixel area) are directly related to the object saliency. However, as shown in Figure 3.16, there is a drop in saliency when an objects gets too large, which indicates that for very large objects, instead of the object itself, its smaller features become more salient.
- In terms of compactness, very concave or very compact objects are less salient than other objects.
- As expected from the center-bias phenomenon [25, 75, 76], objects that are far away from the image center are less likely to be salient, which is illustrated in Figure 3.17.

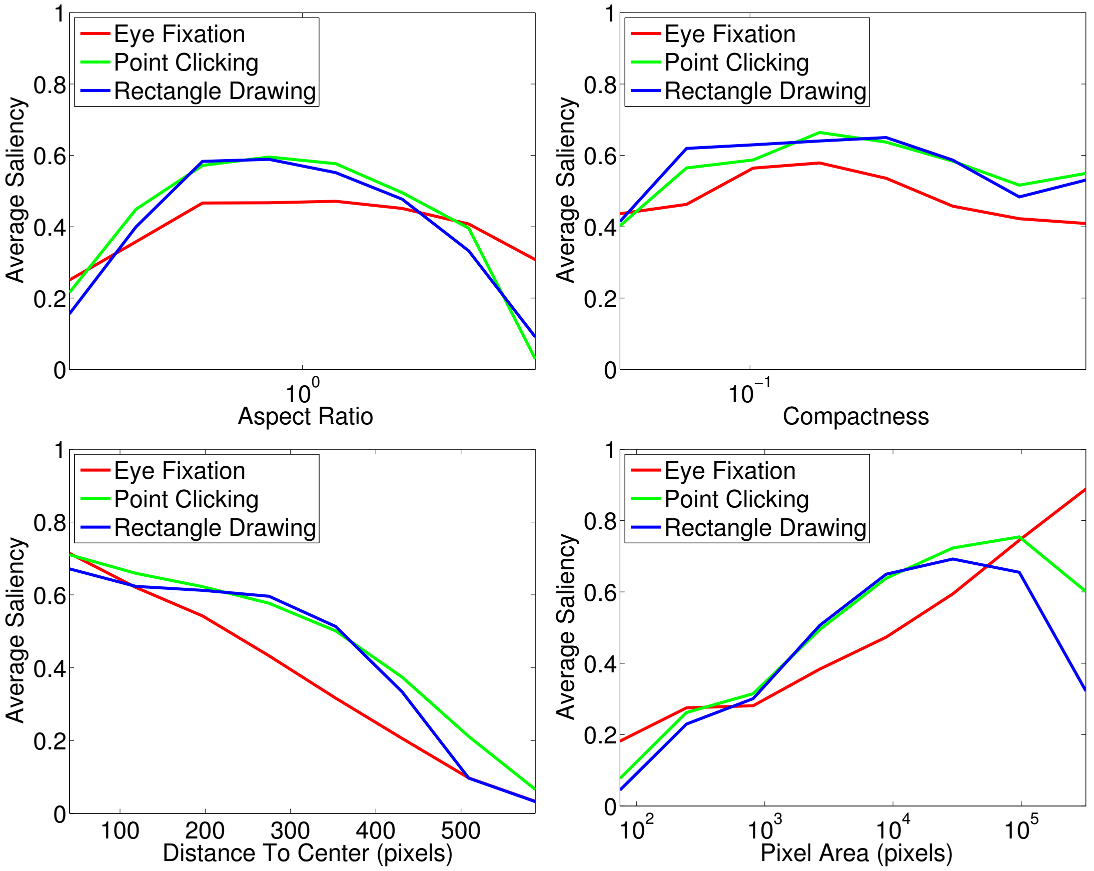


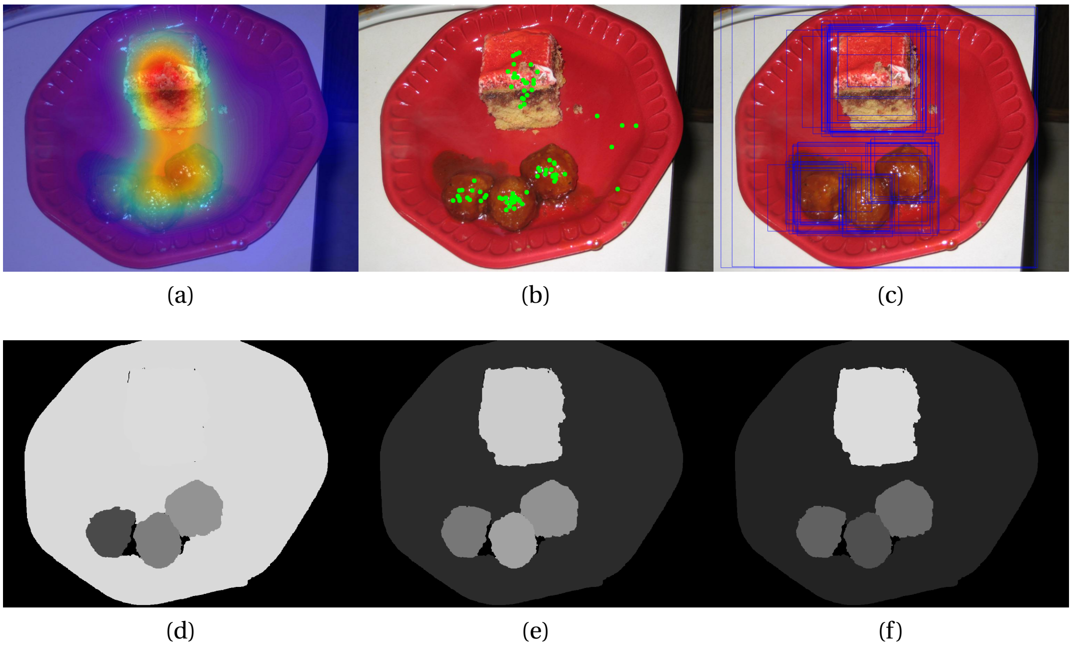
Figure 3.15: The relationship between several shape features of objects and their average saliency values.

3.2.3 Texture

Texture can be an important factor in distinguishing an object from its surroundings. Similar to [3], one way to compute the texture contrast is to look at the edge orientations. For this purpose, we compute the histogram of oriented gradients (HOGs) [77] for objects, and its local and global surroundings and illustrate the effect of the texture contrast in Figure 3.18. We can see that the saliency value has an increasing trend with local HOG contrast. Whereas, global HOG-contrast is not correlated to the saliency, because global histograms most likely include multiple textures.

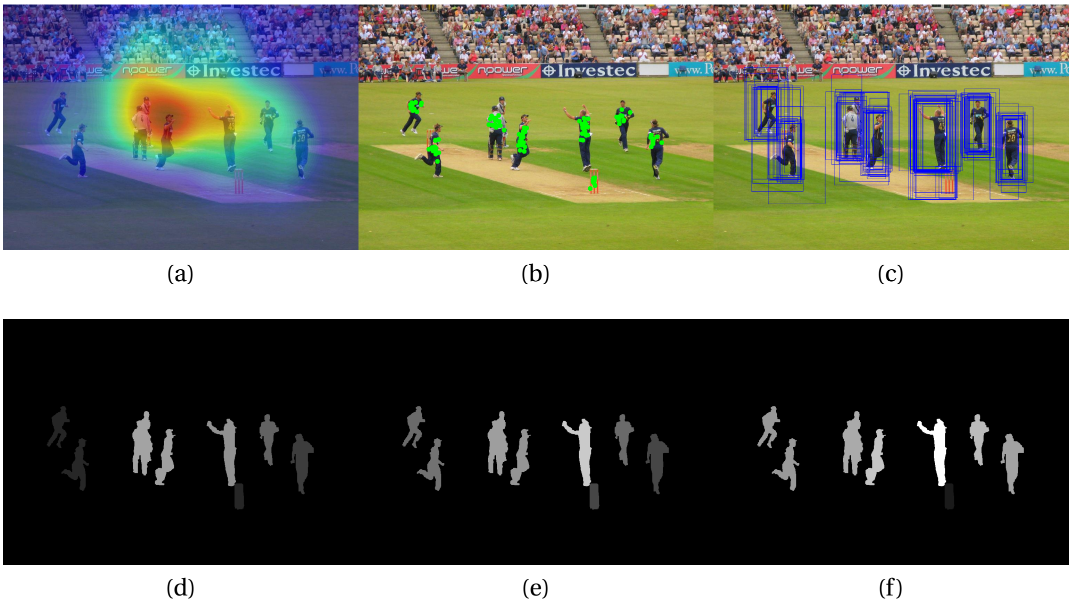
3.3 Summary of the Chapter

In this chapter, we have provided an image dataset, which has three types of multi-level ground truth maps and multiple objects for each image. In order to measure the saliency level of an object, we conducted three subjective experiments: eye tracking, point clicking, and rectangle drawing. Based on these experiments, we have calculated saliency values of the objects. We have shown that saliency levels of multiple objects in complex natural images in fact differs, i.e. object saliency is multi-level.


(a) (b) (c)

(d) (e) (f)

Figure 3.16: (b,c) The smaller objects (desert) on a very large object (plate) might become more salient. In (e) and (f), corresponding ground-truth maps are given, respectively. For (a) eye-tracking data, this is not true because of the inherent ambiguity between eye-fixations and the corresponding objects.


(a) (b) (c)

(d) (e) (f)

Figure 3.17: As the objects get closer to the image center, they are more likely to be attended by the experiment subjects in (a) eye tracking, (b) points clicks and (c) rectangle drawings. In (d)-(f) corresponding multi-level ground-truth maps are given.

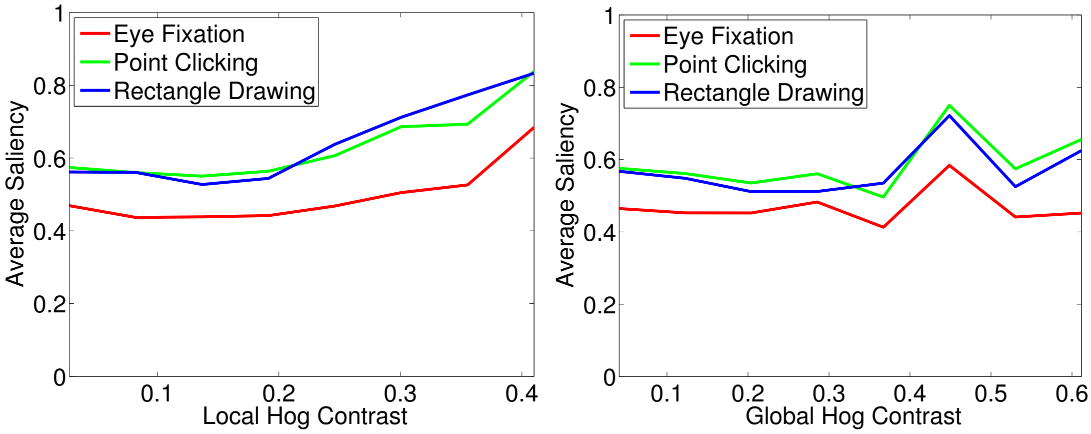


Figure 3.18: The relation between the multi-level saliency values of the objects in our dataset and their local and global texture-contrast.

4 Finding Salient Objects

According to the Zoom-Lens Model of human attention [22], the allocation of processing resources is adjusted via focus and fringe regions, illustrated in Figure 4.1. The focus is the central region of attention, where most of the processing resources are reserved. The size of the focus region represents a trade-off between a larger processing area and the processing efficiency. The fringe is the low-resolution region surrounding the focus.

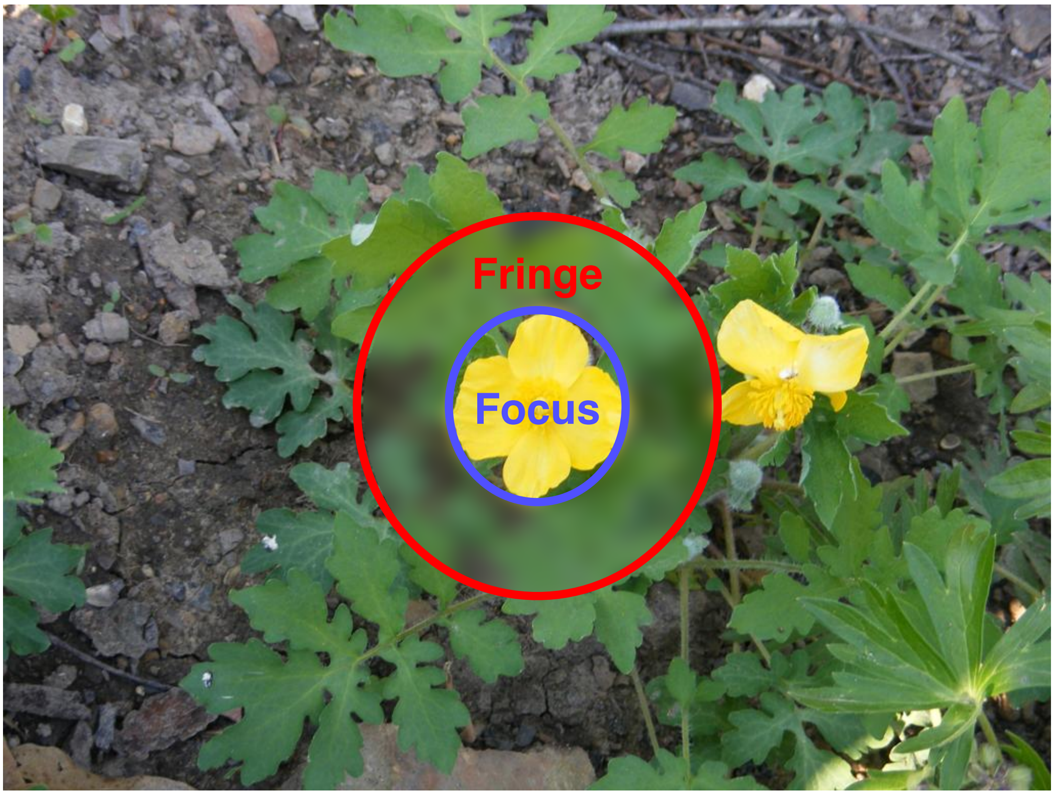


Figure 4.1: The Zoom-Lens Model for visual attention.

An object can be considered distinctive or salient, if the focus and the fringe regions are visually different from each other. The focus-fringe contrast is often calculated using low-level visual cues such as color, texture, orientation, and edge. In salient-object detection literature, this contrast is taken as a measure of visual saliency and is investigated, albeit under different names such as center-surround difference, in [3, 6, 21].

The size and the shape of focus-fringe pairs can affect the quality of an estimated salient object map in terms of background suppression, uniformity, and resistance to noise and texture. In this chapter, we discuss four methods that use focus-fringe pairs with different sizes and shapes as shown in Figure 4.2. The single-scale filtering method is explained in Section 2.2.1 and constitutes a baseline for finding salient objects. The remaining three methods are proposed in this thesis and successively tackle the problems in salient-object detection.

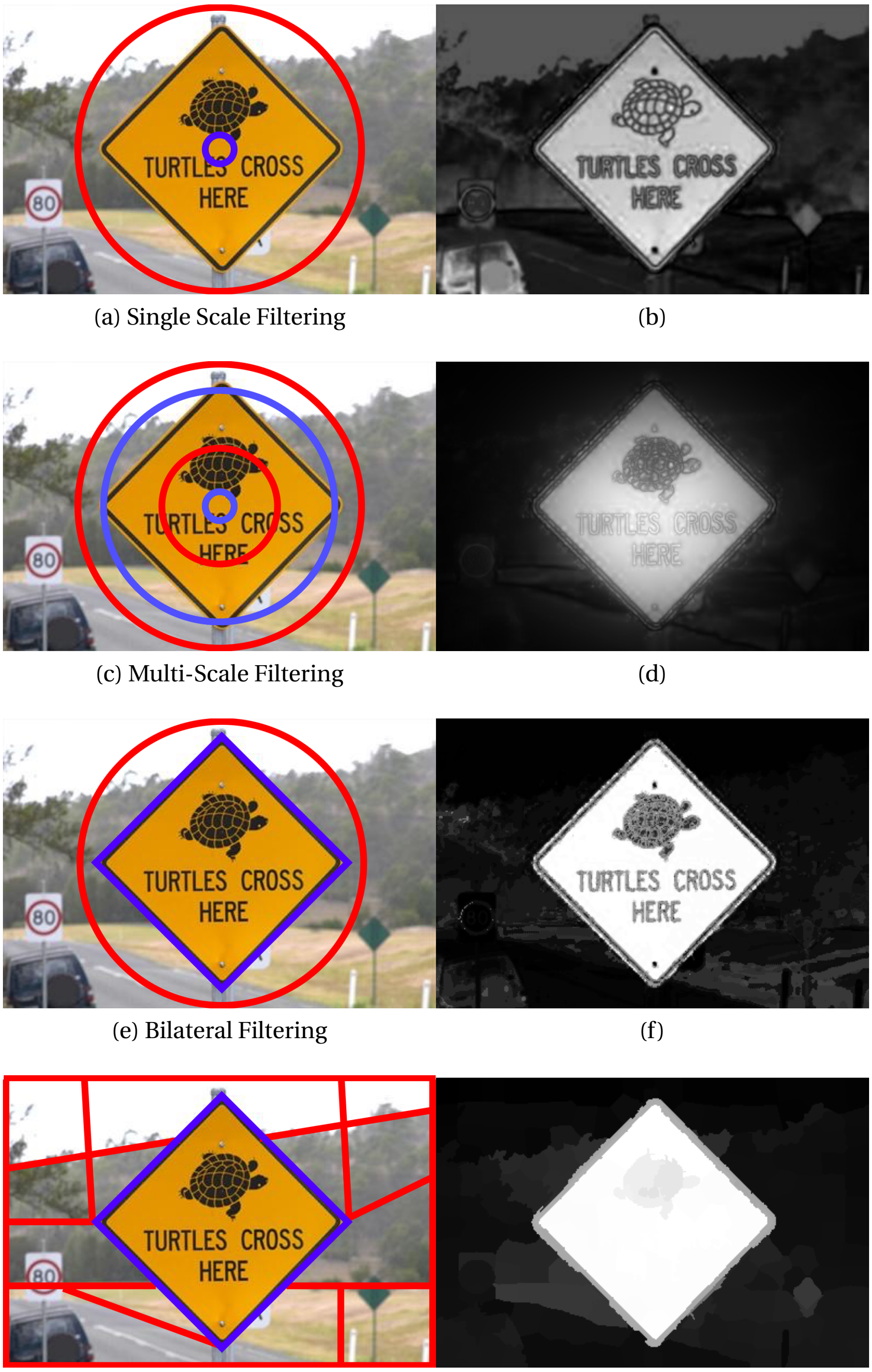
The single-scale filtering method is FT [10]. This method only employs color uniqueness using circular focus and fringe regions with fixed sizes via Gaussian filtering. Although the salient object is highlighted in Figure 4.2(b), most of the background pixels are not properly suppressed.

In our first method, we propose a solution for the background problem with multi-scale filtering. This approach uses multiple circular focus-fringe region pairs (Gaussian filtering) and combines the results based on a compactness measure. The multi-scale nature of this method overcomes the difficulty of choosing an optimal local-global contrast-adjusting parameter mentioned in Section 2.2.1. As shown in Figure 4.2(d), multi-scale filtering labels background pixels as non-salient. However, the saliency values of the object pixels are not uniform, because the focus-fringe pairs are always circular and are independent of the shape of the salient object.

In order to generate uniform salient object maps, in our second method, we introduce an adaptive approach via bilateral filtering. As the coefficients of bilateral filtering are modified with local image-data, the shape of the focus region adapts to the boundary between object and background. Therefore, the bilateral-filtering approach generates more uniform salient-object maps as shown in Figure 4.2(f). This method also regards the spatial variance (see Section 2.2.2) by measuring the position and the size of the salient objects.

The bilateral-filtering approach is only color based and is not able to eliminate textured regions. Therefore, we propose a third method that divides an image into segments and performs salient-object detection using machine learning on said segments. This leads to a uniform salient-object map that can properly suppress the background and eliminate object textures. This method employs the spatial connectivity idea that is discussed in Section 2.2.3, on multiple hierarchical layers. In addition, due to the edge-aware image segmentation step, it is able to adjust the shape and the size of both focus and fringe regions.

The salient-object detection mechanisms that are explained in Section 2.2 and that are used by the methods we propose in this chapter are shown in Table 4.1. The performances of salient-object detectors not only depend on the mechanisms, but also are affected by visual feature types and local-global balance as shown in Section 2.3.



(a) Single Scale Filtering

(b)

(c) Multi-Scale Filtering

(d)

(e) Bilateral Filtering

(f)

(g) Segmentation Based

(h)

Figure 4.2: The focus-fringe differences of the four methods explained in this chapter. Red and blue curves correspond to the focus and fringe regions, respectively. The estimated salient-object map of each method is given in (b,d,f,h).

Table 4.1: The salient-object detection mechanisms of our proposed methods.

	Uniqueness	Spatial Variance	Spatial Connectivity
Multi-Scale Filtering	✓		
Bilateral Filtering	✓	✓	
Segmentation Based	✓	✓	✓

In this chapter, we start the discussion about finding salient object with the single-scale filtering method. We then successively introduce three salient-object detectors and elaborate on their advantages and disadvantages. Finally, we compare their performance with the state-of-the-art methods in Section 2.2 by using the well-known datasets in Section 2.1. We show that our methods perform comparable or better in their respective categories (uniqueness, spatial variance, spatial connectivity) with different measurement metrics that are explained in Section 2.3.

4.1 Single-Scale Filtering Approach

Color uniqueness is the most prominent and easy-to-use visual cue that makes an object salient. We can compute color contrast using a circular focus-fringe pair (two Gaussian filters) at each pixel of an input image. This numerical color difference is often regarded as the saliency value of that pixel. One good example of this approach is presented by Achanta et al. [10] (FT), where the focus region is considered to be a 3×3 Gaussian filter and the fringe region is assumed to be the whole image. Note that using a single focus-fringe pair is regarded as a single-scale filtering. There are two major drawbacks of single-scale filtering approaches:

- Because the filtering uses only one scale, large, non-salient structures with unique properties (e.g. color, texture, etc.) are not properly suppressed.
- The large fringe-sizes highlight only globally-unique objects. Although some methods [13, 15] use a spatial-distance term (see (2.4)) to balance between local and global contrast, the effect of this term is fixed, because fringe size is constant.

In order to eliminate background pixels and to detect locally- and globally-salient objects, the size of focus-fringe pairs should be varied using multiple scales.

4.2 Multi-Scale Filtering Approach

In our first method, we propose a multi-scale filtering (MSF) approach. Our method is a more general version of Achanta et al. [10] (FT), where focus-fringe differences are globally computed by using single-scale filtering. Whereas, in our algorithm, we vary both focus and fringe filter sizes. The output of each filter pair is considered as a *weak* salient-object map. in order to generate the final salient-object map, we combine these weak maps by weighting them with two adaptive sub-modules based on object compactness and on center prior. The flowchart of our method is shown in Figure 4.3.

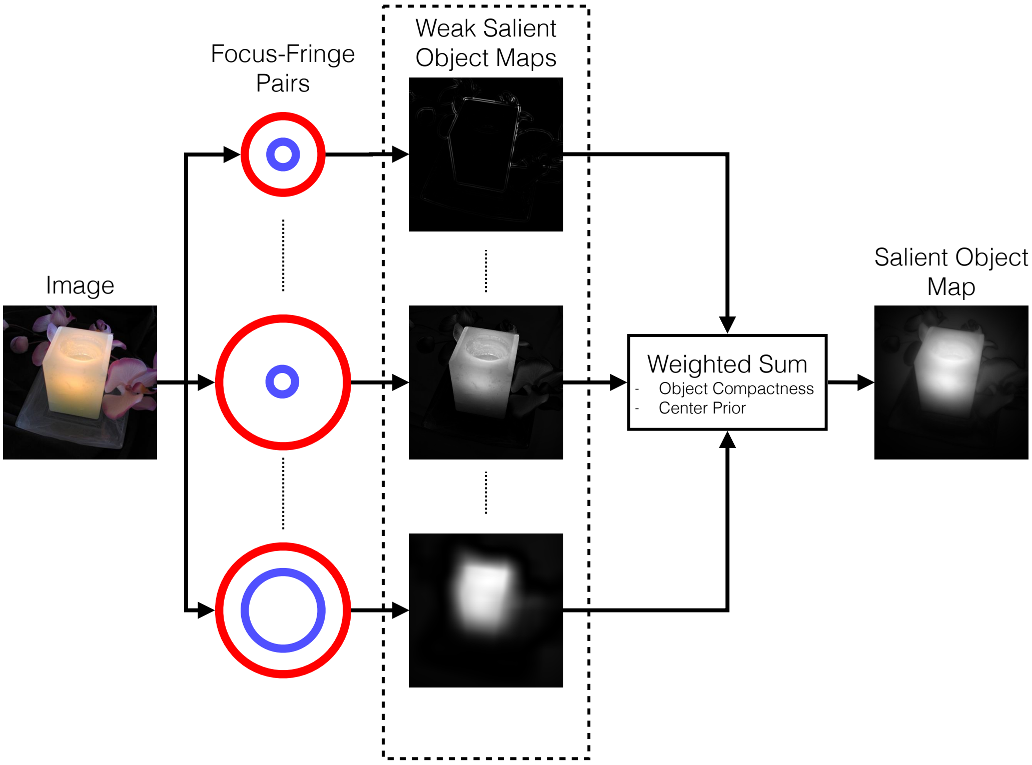


Figure 4.3: The flowchart of our multi-scale filtering based salient object detector.

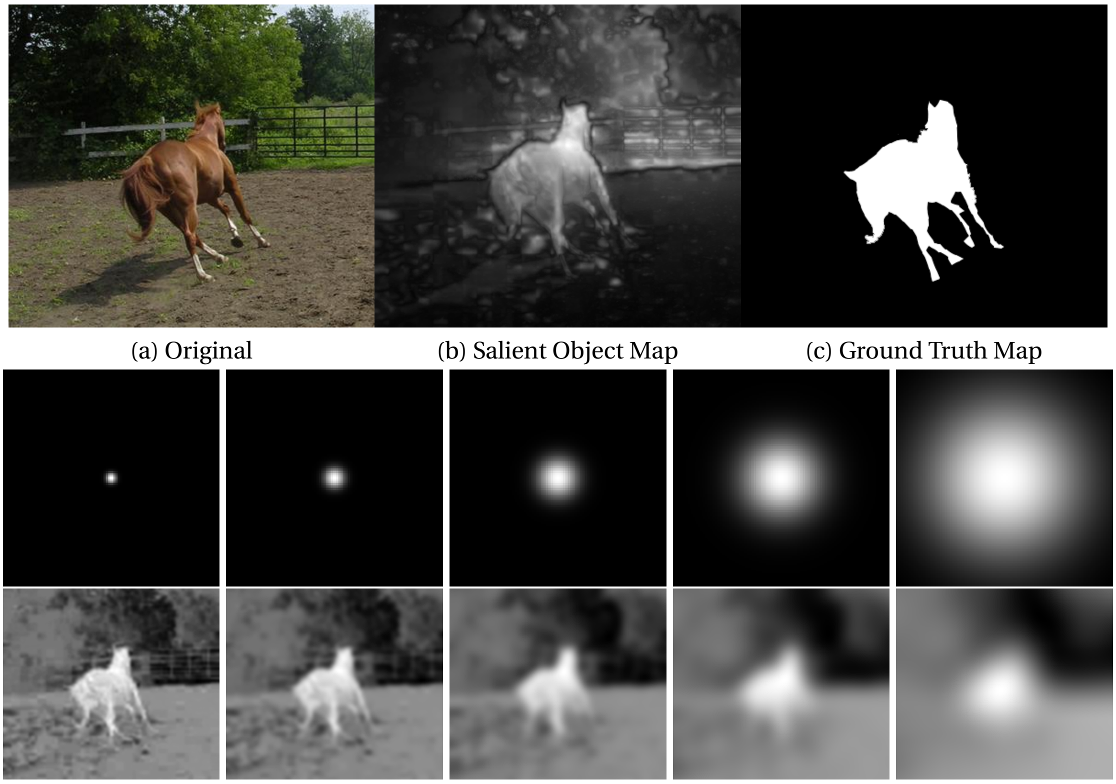
We perform the filtering operations in perceptually uniform CIELa*b* color space. An example of multi-scale filtering output is illustrated in Figure 4.4. A weak salient-object map is the pixel-wise difference between any two filter outputs (both in the same color channel). Each weak salient-object map represents a distinct portion of spatial-frequency spectrum.

4.2.1 Analysis of the Frequency Domain

Spatial-frequency content and its relation to visual saliency is extensively analyzed in [10]. Here, we present a multi-scale extension of their work by providing further analysis on the importance of frequency content in salient-object map extraction. We illustrate the analysis using one-dimensional Gaussian filters defined as follows:

$$F_G{}^s = \frac{1}{\sqrt{2\pi\sigma_s{}^2}} \exp\left(-\frac{x^2}{2\sigma_s{}^2}\right) \tag{4.1}$$

Here, $F_G{}^s$ is a Gaussian filter of size $2^s + 1$, and σ_s is related to the filter size $\sigma_s = 2^{s-1}$. We can generate band-pass filters using the Difference-of-Gaussian (DoG) from 2D versions of these filters; this gives us a very flexible method capable of processing the whole frequency spectra with adjustable filters. An illustration of the band-pass DoG filter outputs on a 64×64 image are shown in Figure 4.5. The salient object is outlined



(a) Original

(b) Salient Object Map

(c) Ground Truth Map

Figure 4.4: In the first row, (a) an example image from MSRA-1000 dataset, (b) our estimated salient object and (c) ground truth map is given. Second row shows the multi-scale filters. Third row shows filtered a* channel of the image.

when a small focus and a large fringe filter, such as 1 and 129, are combined. This corresponds to the single-scale filtering result in FT. If both focus and fringe are small, filtering corresponds to an edge detector, which in turn helps us preserve the object boundaries in the estimated salient object map. Combining large focus and fringe filters, such as 65 and 129, detects the salient object as a blob and provides texture resistance on salient regions.

4.2.2 Low-Level Salient-Object Detection

Our model includes three main steps for salient-object detection. First, we discuss our multi-scale filtering framework. We then continue with two adaptive improvements, compactness and center prior measure.

Filtering Framework

In order to compute the color contrast, we first convert an image into CIELa*b* color space. We then filter all the channels of an input image **I** using a series of 2D Gaussian filters ${F_G}^s$ ($s = 0, 1, 2, ...N$) with different sizes. F_0 represents an all-pass filter (i.e. image itself). The remaining filters have a size of $(2^s + 1)$. The σ_s value of the filters are equal

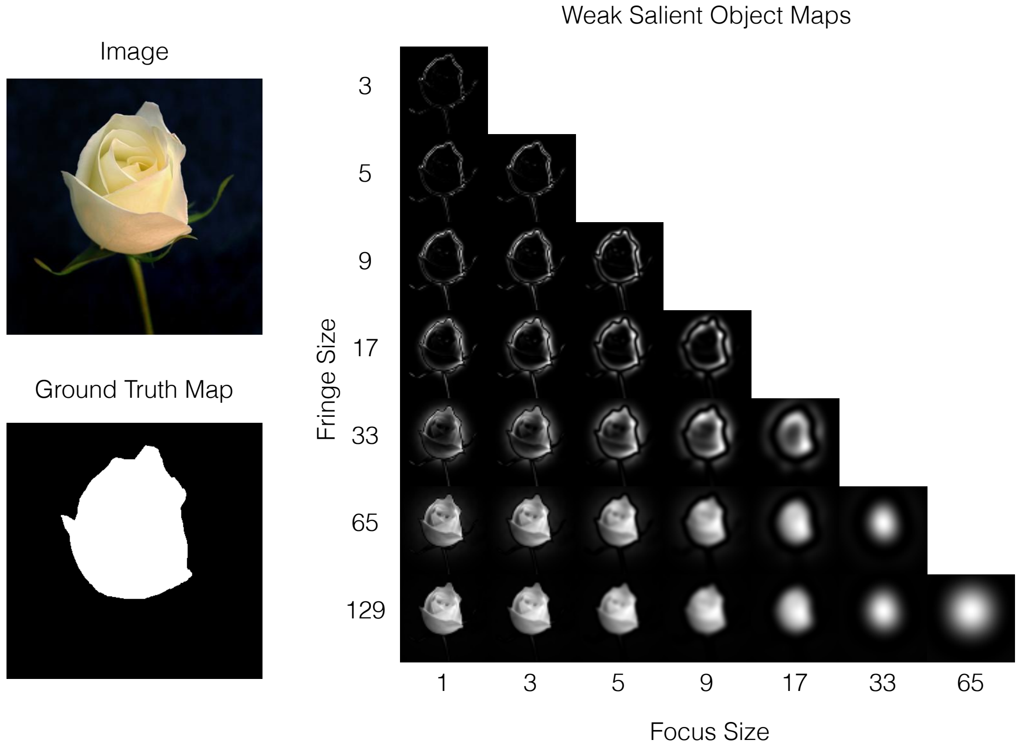


Figure 4.5: Muti-scale filters are applied to a natural image (Here, only L channel of CIELab color space is illustrated). The numbers represent the size of the focus and fringe filters. For example the weak salient object map on bottom-left corner is obtained by taking difference of a focus size of 1 pixel and a fringe size of 129 pixels.

to 2^{s-2}. The number of filters (N) should depend on the input image size, because fixed-size ${F_G}^N$ cannot cover whole frequency spectrum $[0,\pi)$ for images larger than ${F_G}^N$. In addition, a larger image requires a finer resolution in frequency domain, due to the increased amount of detail. ${F_G}^N$ is the smallest filter that is larger than the image (i.e., ${F_G}^N$ has a size of 513, if the input image is 400×300). As the last filter is very large, we should carefully filter image borders. Assuming that the border pixels belong to background regions in general, we replicate the border value of the image for correct filtering. All filtering operations are performed in frequency domain for efficient computation.

Pixel-wise squared differences of every possible filtered image-pairs are computed (see Figure 4.5) for the initial image abstraction. Each pair that uses filters ${F_G}^i$ and ${F_G}^j$ gives us a weak salient-object map $\mathbf{Z}^C_{ij}$, where C is either L, a* or b*. A weak salient-object map is an image representing a certain band-pass spatial-frequency interval for a channel. The final salient-object map is a weighted combination of these weak maps. In order to calculate the weights, we introduce two different adaptive measures, compactness K (scalar) and center prior $\mathbf{P}$ (same size as the image). They are explained in Section 4.2.2 and 4.2.2, respectively. The flow of this algorithm is as follows:

```
Initialize salient object map $\mathbf{S} = \mathbf{0}$
for $C = L, a^*, b^*$ do
    for $i = 1, 2, 3, ...N$ do
        for $j = i+1, i+2, ...N$ do
            $\mathbf{I}_i$ = filter$(C, {F_G}^i)$;
            $\mathbf{I}_j$ = filter$(C, {F_G}^j)$;
            $\mathbf{Z}^C_{ij} = (\mathbf{I}_i - \mathbf{I}_j) \bullet (\mathbf{I}_i - \mathbf{I}_j)$
            $K$= compactness($\mathbf{Z}^C_{ij}$); (Section 4.2.2)
            $\Theta$= centerPrior($\mathbf{Z}^C_{ij}$); (Section 4.2.2)
            $\mathbf{S} = \mathbf{S} + \Theta \bullet K \cdot \mathbf{Z}^C_{ij}$
        end
    end
end
```

Algorithm 1: Multi-scale filtering algorithm

Here • represents an element-wise multiplication.

Adaptive Compactness Measure

Our method combines all weak salient-object maps to get a final saliency estimation (see Algorithm 1). A naive summation of these filter pairs could cause noisy salient-object maps with many false positives. In order to avoid this, we introduce a compactness measure that evaluates the distribution of salient pixels around the image.

In order to compute the compactness, we first normalize a weak salient-object map $\mathbf{Z}^C_{ij}$ between 0 and 1 and get $\overline{\mathbf{Z}}^C_{ij}$. We then calculate the center of mass (μ_x, μ_y) (used in Section 4.2.2) and the spatial distribution (σ^2_x, σ^2_y) of a weak map along the x and y image dimensions as follows:

$$\begin{aligned} \mu_x &= \frac{1}{|T|}\sum_x\sum_y x \cdot \overline{\mathbf{Z}}^C_{ij}(x,y) \\ \sigma^2_x &= \frac{1}{|T|}\sum_x\sum_y (x-\mu_x)^2 \cdot \overline{\mathbf{Z}}^C_{ij}(x,y) \end{aligned} \tag{4.2}$$

Here, T is the sum of all values in $\overline{\mathbf{Z}}^C_{ij}$. Similar equations are used for the computation of μ_y and σ^2_y. These variables measures the position and the compactness of a salient-object map along each dimension as illustrated in Figure 4.6.

We also calculate the same variables using the inverted salient-object map $1-\overline{\mathbf{Z}}^C_{ij}$ and call them $\tilde{\mu}_x, \tilde{\mu}_y, \tilde{\sigma}^2_x, \tilde{\sigma}^2_y$. These variables represent the compactness of the background pixels. Final compactness value is computed as follows.

$$K = \exp\Big(-k * \Big(\frac{\sigma^2_x \sigma^2_y}{\tilde{\sigma}^2_x, \tilde{\sigma}^2_y}\Big)\Big) \tag{4.3}$$

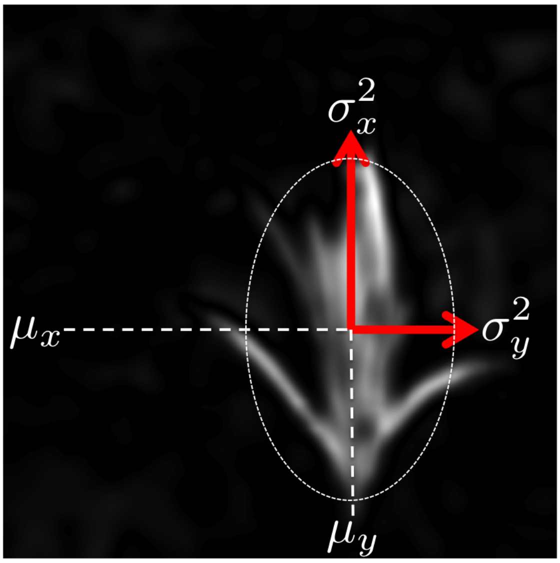


Figure 4.6: The spatial meanings of each term in compactness computation. μ_x and μ_y are related to the center of mass of the object, and σ_x^2 and σ_y^2 are related to the 2D size of the object.

Here K is the compactness measure and k is an adjustment parameter. In our experiments, we find that $k = 4$ gives the best performance. It can be seen from (4.3) that, at low σ_x^2,σ_y^2 values (compact object) and high $\tilde{\sigma}_x^2$,$\tilde{\sigma}_y^2$ values (distributed background), compactness approaches to 1 and vice versa. An example of the compactness measure computation is given in Figure 4.7, where non-compact weak maps, such as in Figure 4.7(b) are suppressed thus resulting a better salient-object map estimation. Note that, although the equations (4.2) and (2.5) are similar, the adaptive compactness value does not measure the spatial variance of a visual feature. It instead calculates the spatial distribution of saliency values.

Adaptive Center-Prior

Humans tend to look at the center of an image [25]. We can benefit from this property and assume a center prior in salient object map computations. Even though center prior eliminates false positives near the boundary of the image, applying a non-adaptively might also eliminate salient objects that are not close to the image center, which is not desirable. Therefore, we introduce an adaptive center prior for each weak salient-object map by using the image compactness statistics we compute in Section 4.2.2.

$$\Theta(x, y) = \exp\left(-\frac{(x-\mu_x)^2}{n*\sigma_x^2} - \frac{(y-\mu_y)^2}{n*\sigma_y^2}\right) \tag{4.4}$$

Here, n is an adjustment factor, and $n = 12$ is used throughout our experiments. The adaptive center prior Θ has the same size with the weak salient-object map and multi-

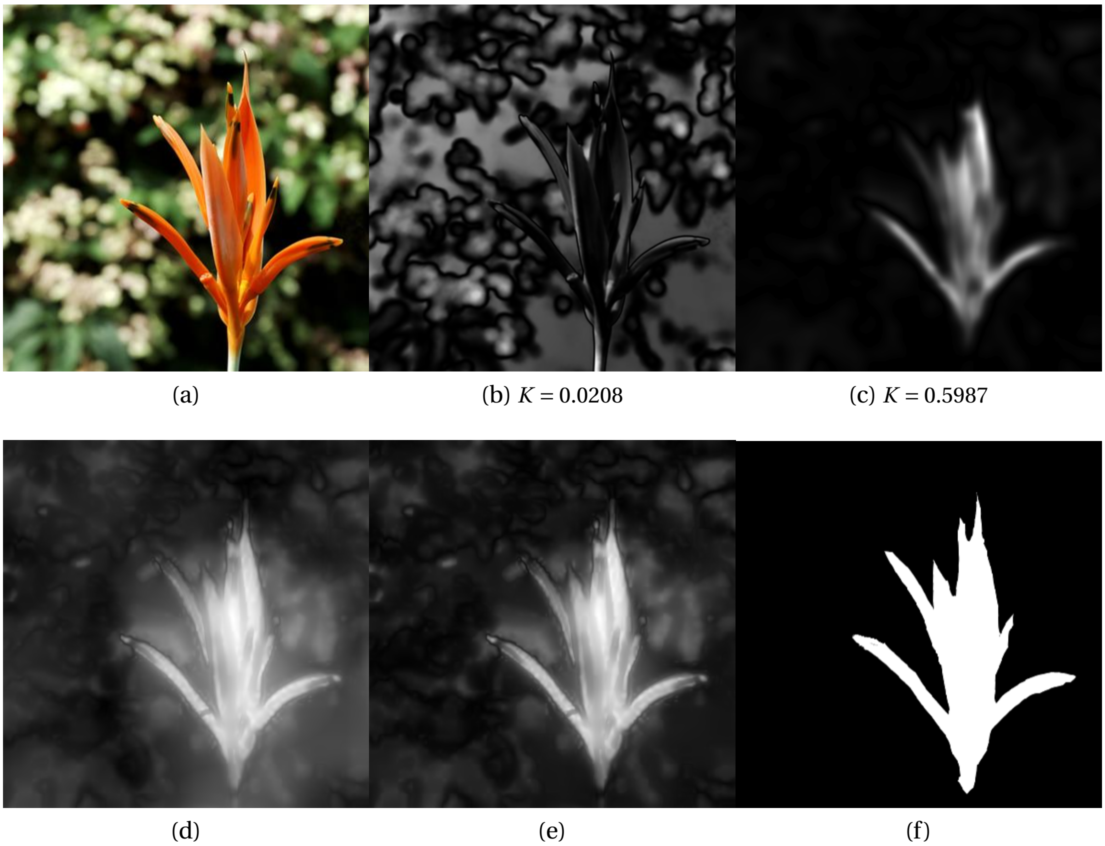


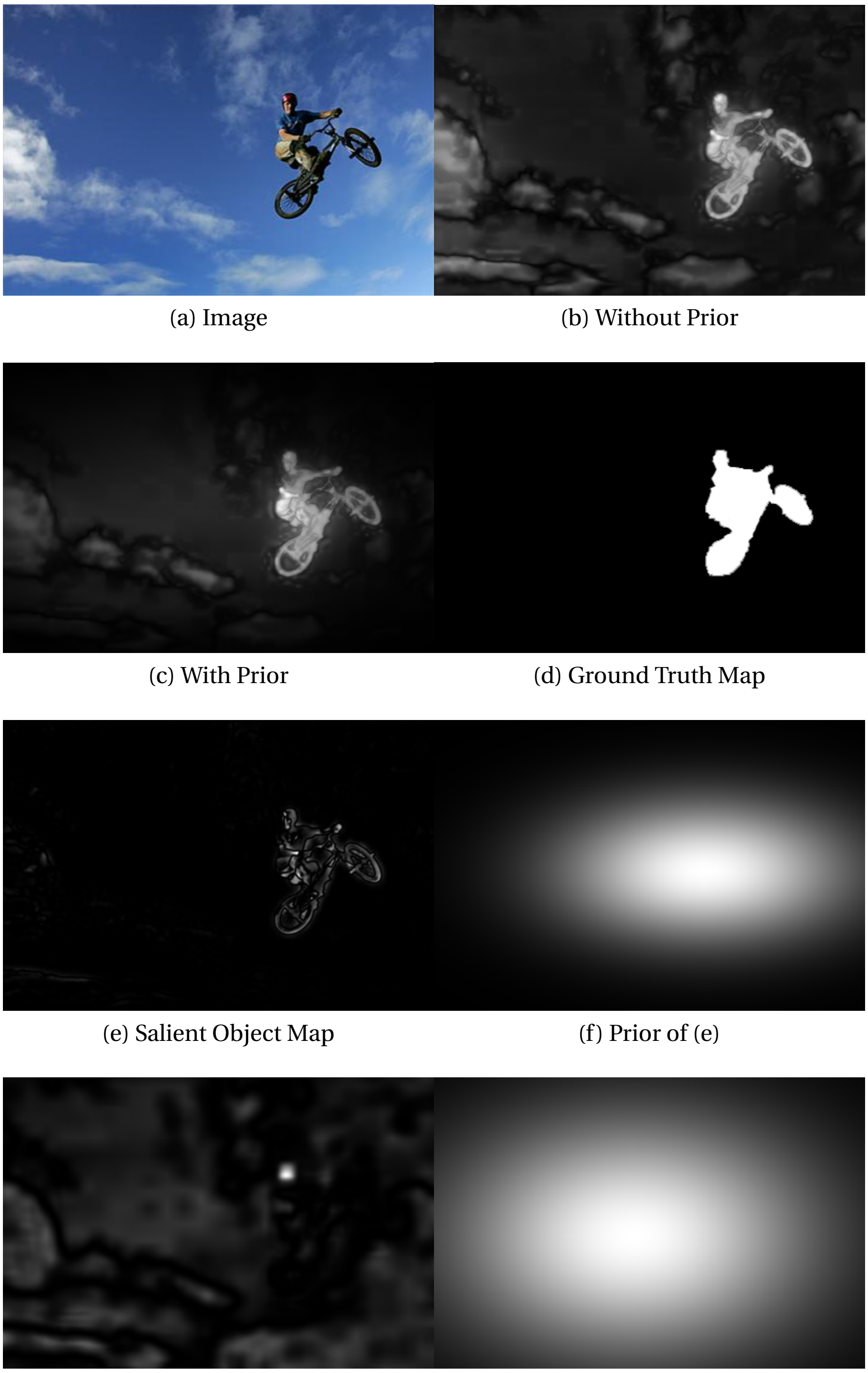
Figure 4.7: Example weak maps an their compactness values. (a) Original image (b,c) weak saliency maps with compactness values (d) output without compactness (e) output with compactness (f) ground truth

plies it element-wise.

The state-of-the-art methods, such as that of Shen and Wu [14], also employ center prior. Our method differs from them by taking distribution of salient pixels into account and adaptively shifting the prior mask. In addition, as center prior is computed for each weak salient-object map, non-centered saliency information is not lost. In Figure 4.8, an illustration of center prior is given. As the same statistics are used for both compactness and center prior measure, their effect on the image are not completely independent from each other. However, compactness eliminates undesired weak salient-object maps in a global sense, which works locally for adaptive center-prior.

4.2.3 Discussion

Our first method was based on combining multi-scale filtering outputs by using compactness and adaptive center prior modules. Varying focus and fringe size in this framework has several advantages compared to single-scale filtering, such as better background suppression and detection of locally salient objects. However, as illus-



(a) Image

(b) Without Prior

(c) With Prior

(d) Ground Truth Map

(e) Salient Object Map

(f) Prior of (e)

(g) Salient Object Map

(h) Prior of (g)

Figure 4.8: Example weak maps an their adaptive center priors. (a) Original image (b) output without prior (c) output with prior (d) ground truth (e-h) two weak saliency maps and their priors

trated in Figure 4.2(d), due to the circular shape of the linear Gaussian filter, saliency values of the object pixels are not uniform and decrease at the object-background boundary.

4.3 Bilateral-Filtering Approach

In our second method, we introduce a bilateral-filtering approach. This method has two major advantages over multi-scale filtering. First, the shape of the focus region is adaptively changed with the image data. Therefore, salient-object maps are uniform. Second, with the aid of a spatial variance step, it can detect the position and size of objects. An example result of this method is given in Figure 4.10. Bilateral filtering is a computationally heavy operation. As salient-object detection is a pre-processing step, it should process the image in an efficient and accurate manner and provide as much information as possible for the successive step. Therefore, in our second method, we satisfy the efficiency, accuracy, and information criteria by introducing a Fast, Accurate, and Size-Aware (FASA) salient-object detector.

4.3.1 Overview of the Method

Our second method, FASA, combines a probability of saliency with a global-contrast map. Figure 4.9 provides a scheme illustrating our method. For computational efficiency, our algorithm first quantizes an image to reduce the number of colors. Then, in order to estimate the position and the size of the salient object, the spatial center and variances of the quantized colors are calculated. These values are put in an object model to compute the probability of saliency. The same quantized colors are used to generate global contrast values as well. Finally, the saliency probabilities of the colors and the contrast values are fused into a single salient-object map.

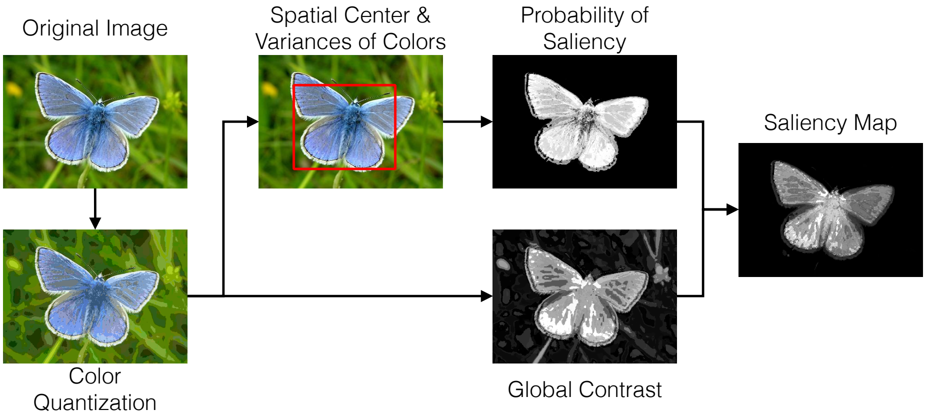
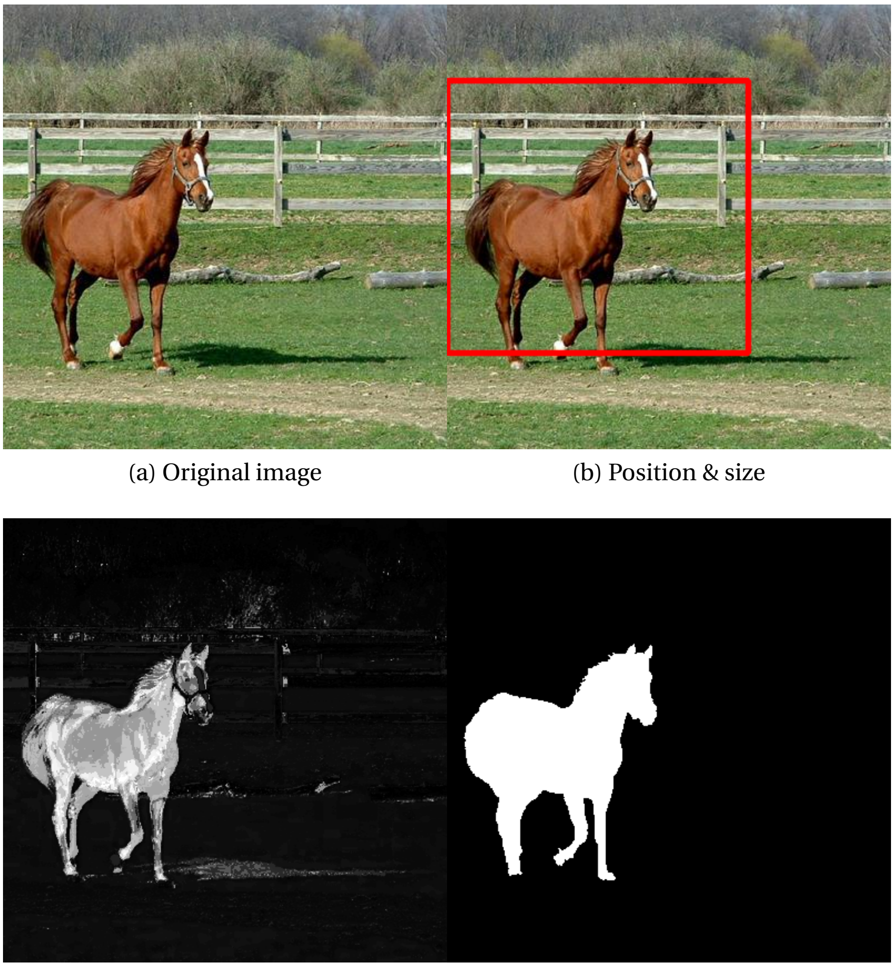


Figure 4.9: Scheme of our bilateral-filtering-based method.



(a) Original image

(b) Position & size

(c) Salient Object Map

(d) Ground Truth Map

Figure 4.10: FASA processes (a) the 400×400 pixel image in 6 miliseconds and outputs (b) the parameters of rectangles that enclose the salient objects and (c) a salient-object map that is comparable to (d) the ground truth.

4.3.2 Spatial Center and Variances of a Color

One of the prominent components of saliency is the spatial variance of a color in a scene [15, 16]. To compute it, we first define a position- and a color-vector notation.

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \mathbf{c}_i = \begin{bmatrix} L^*(\mathbf{p}_i) \\ a^*(\mathbf{p}_i) \\ b^*(\mathbf{p}_i) \end{bmatrix} \tag{4.5}$$

Here, $\mathbf{p}_i$ is the position vector, which represents the coordinates (x_i, y_i) of the i^{th} pixel. $\mathbf{c}_i$ is the color vector of the pixel at position $\mathbf{p}_i$ in CIEL*a*b* color space. The spatial center $\{m_x(\mathbf{p}_i), m_y(\mathbf{p}_i)\}$ and the horizontal and vertical variances $\{V_x(\mathbf{p}_i), V_y(\mathbf{p}_i)\}$ of a color can be calculated using the following equation:

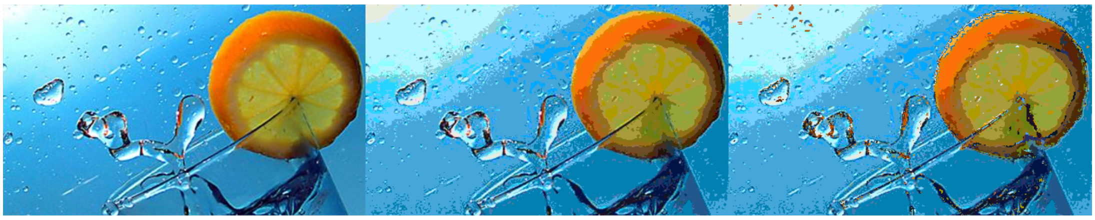
$$
\begin{aligned}
m_x(\mathbf{p}_i) &= \frac{\sum_{j=1}^{N} w^c(\mathbf{c}_i,\mathbf{c}_j)\cdot x_j}{\sum_{j=1}^{N} w^c(\mathbf{c}_i,\mathbf{c}_j)}\\
V_x(\mathbf{p}_i) &= \frac{\sum_{j=1}^{N} w^c(\mathbf{c}_i,\mathbf{c}_j)\cdot \left(x_j - m_x(\mathbf{p}_i)\right)^2}{\sum_{j=1}^{N} w^c(\mathbf{c}_i,\mathbf{c}_j)}
\end{aligned}
\tag{4.6}
$$

Similar calculations can be done for y dimension. Here, N is the total number of pixels in an image, and $w^c(\mathbf{c}_i,\mathbf{c}_j)$ are the color weights and are calculated using a Gaussian function.

$$
w^c(\mathbf{c}_i,\mathbf{c}_j) = \exp\left(-\frac{||\mathbf{c}_i - \mathbf{c}_j||^2}{2{\sigma_c}^2}\right) \tag{4.7}
$$

Here, σ_c is a parameter for adjusting the effect of the color difference. If we look at (4.6), we can notice that w^c in both of the equations depends on the spatial coordinates. These calculations correspond to a bilateral filter with a color kernel, namely $w^c(\mathbf{c}_i,\mathbf{c}_j)$. For computational efficiency, the spatial kernel (or support) is chosen to be the whole image, which turns our algorithm into a global saliency-detection method.

The computational complexity of (4.6) is $O(N^2)$. Here, for efficient bilateral filtering, we follow the approach proposed by Yang et al. [78], in which they quantize the intensity levels of a grayscale image. Here, the colors $\mathbf{c}_i$ of an image are quantized (*i.e.*, a color histogram is created) into a set of colors $\{\mathbf{q}_k\}_{k=1}^{K}$, where K is the number of colors after the quantization. In practice, we can minimize K by assigning certain quantized colors that have very few pixels to the perceptually closest quantized color with a non-zero number of pixels. A similar color quantization in sRGB color space is performed in Cheng et al. [13]. However, we quantize the image in perceptually uniform CIEL*a*b* color space, hence we need fewer quantization bins. An example of the color quantization is given in Figure 4.11.


(a) Original Image

(b) 175 quantized colors

(c) 50 quantized colors

Figure 4.11: The L*a*b* histogram (8 bins in each channel, $8^3 = 512$ bins in total) of (a) the original image contains (b) 175 quantized colors with non-zero histogram bins and (c) 50 quantized colors that can cover 95% of the image pixels.

The operation $\mathbf{c}_i \rightarrow \mathbf{q}_k$ indicates that the color of the pixel at $\mathbf{p}_i$ falls to the k^{th} color histogram bin after the quantization. If we quickly calculate the color histogram of the image and precompute $w^c(\mathbf{q}_k, \mathbf{q}_j)$, we can efficiently estimate the spatial center and variances of the quantized colors as follows:

$$\begin{aligned} m'_{xk} &= \frac{\sum_{j=1}^{K} w^c(\mathbf{q}_k, \mathbf{q}_j) \cdot \sum_{\forall x_i | \mathbf{c}_i \to \mathbf{q}_j} x_i}{\sum_{j=1}^{K} h_j \cdot w^c(\mathbf{q}_k, \mathbf{q}_j)} \\ V'_{xk} &= \frac{\sum_{j=1}^{K} w^c(\mathbf{q}_k, \mathbf{q}_j) \cdot \sum_{\forall x_i | \mathbf{c}_i \to \mathbf{q}_j} (x_i - m'_{xk})^2}{\sum_{j=1}^{K} h_j \cdot w^c(\mathbf{q}_k, \mathbf{q}_j)} \end{aligned} \tag{4.8}$$

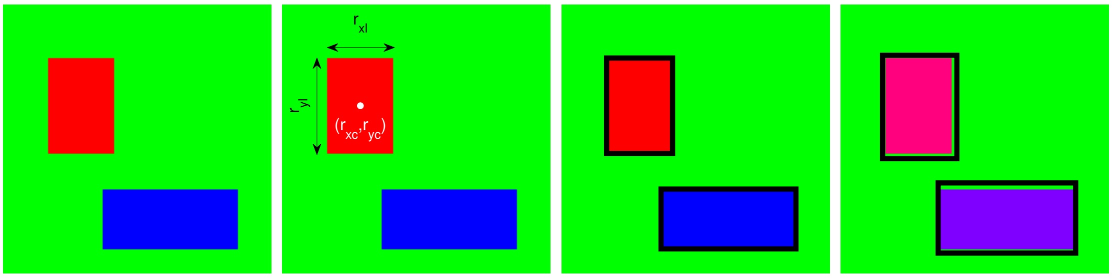
Similar calculations can be performed for y dimension. Here, $\{m'_{xk}, m'_{yk}\}$ is the spatial center and $\{V'_{xk}, V'_{yk}\}$ are the spatial variances of the k^{th} quantized color. $h_k = |\forall x_i | \mathbf{c}_i \to \mathbf{q}_k|$ is the number of pixels in the k^{th} color histogram bin. The spatial center and variances at each pixel in (4.6) can be estimated as follows:

$$\begin{aligned} m_x(\mathbf{p}_i) &\approx m'_{xk} \quad \forall \mathbf{p}_i | \mathbf{c}_i \to \mathbf{q}_k \\ V_x(\mathbf{p}_i) &\approx V'_{xk} \quad \forall \mathbf{p}_i | \mathbf{c}_i \to \mathbf{q}_k \end{aligned} \tag{4.9}$$

Similar calculations can be performed for y dimension. We reduce the complexity of the bilateral filtering in (4.6) to $O(K^2)$ via the color quantization in (4.8). In addition, as explained in Section 4.3.3, $\{m'_{xk}, m'_{yk}\}$ and $\{V'_{xk}, V'_{yk}\}$ provide valuable position and size cues about the salient object.

4.3.3 The Center and the Size of a Salient Object

The spatial center $\{m'_{xk}, m'_{yk}\}$ shows the color-weighted center of mass of k^{th} quantized color of the image. The spatial variances $\{V'_{xk}, V'_{yk}\}$ depict how spatially distributed the same quantized color is within the image. In addition, it also gives us an idea about the "size" of that color. In order to show this relationship, in Figure 4.12 (a), we illustrate a test image that is 256×256 pixels in size and that includes a red and a blue rectangle.



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

Figure 4.12: (a) A test image with two salient rectangles with (b) the center and size parameters of the red rectangle. (c) The estimated position and sizes are shown with black bounding rectangle. (d) The accuracy of the center and the size estimation degrades, when the color of the objects are similar.

In this image, we have three dominant colors, i.e. $k \in \{red, green, blue\}$. As there is sufficient global color contrast between these colors, we can assume $w^c(\mathbf{q}_k, \mathbf{q}_j) \approx 0$ for $k \neq j$ and we know that $w^c(\mathbf{q}_k, \mathbf{q}_k) = 1$. By using this, we can rewrite (4.8) and estimate the center of the objects as follows:

$$m'_{x,rect} \approx \frac{1}{h_{rect}} \sum_{\forall x_i | \mathbf{c}_i \to \mathbf{q}_{rect}} x_i \quad = r_{xc} \tag{4.10}$$

Here r_{xc} is the x coordinate of the center of the red rectangle. As rectangles are symmetrical in both horizontal and vertical dimensions, we can easily compute the center of the red rectangle (r_{xc}, r_{yc}) using (4.10). The size of an object can be calculated as follows:

$$V'_{x,rect} \approx \frac{1}{h_{rect}} \sum_{\forall x_i | \mathbf{c}_i \to \mathbf{q}_{rect}} (x_i - r_{xc})^2 \approx \frac{r_{yl} \cdot \int_{-r_{xl}/2}^{r_{xl}/2} x^2 \cdot dx}{r_{yl} \cdot r_{xl}} = \frac{r_{xl}^2}{12} \tag{4.11}$$

Here, r_{xl} and r_{yl} are the width and the height of the red rectangle, respectively. Similar equations can be derived for the y dimension. As we can see from (4.10) and (4.11), given sufficient color contrast, we are able to estimate the center and the size of both rectangles and ellipses, which is illustrated with black boundaries in Figure 4.12 (c).

Conventionally, a bounding rectangle is used to represent a detected object. However, in some cases, it could be useful to represent the objects by using a bounding ellipse instead. The central position of a bounding ellipse can be computed using (4.10). To estimate the dimensions of an ellipse, we slightly modify (4.11):

$$V'_{x,ellipse} \approx \frac{\pi/4 \cdot e_{yl} \cdot \int_{-e_{xl}/2}^{e_{xl}/2} x^2 \sqrt{1 - \frac{x^2}{(e_{xl}/2)^2}} \cdot dx}{\pi^2/16 \cdot e_{yl} \cdot e_{xl}} = \frac{e_{xl}^2}{16} \tag{4.12}$$

Here, e_{xl} and e_{yl} are the width and the height of an ellipse, respectively. The equation for estimating the height is similar to the one in (4.12).

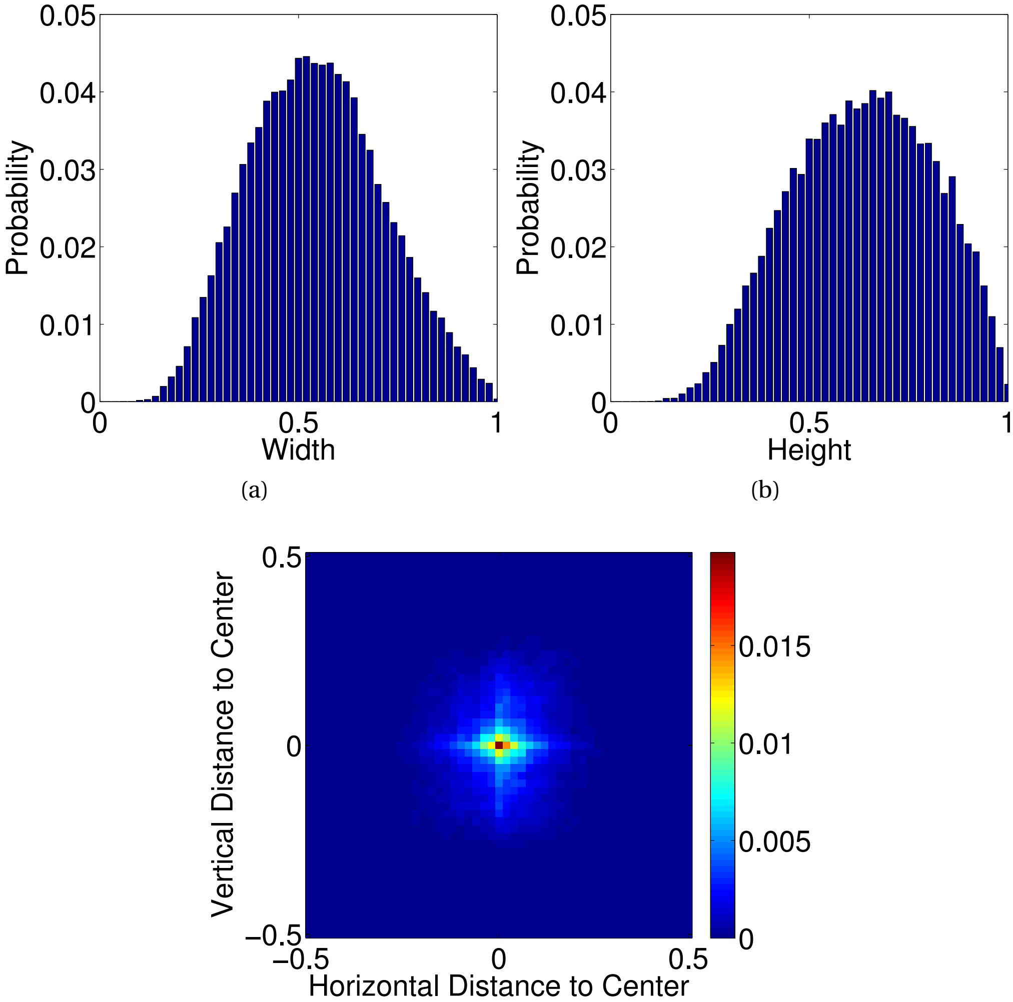
Natural images often contain non-rectangular objects and the color of the objects might interfere with each other as shown in Figure 4.12d. However, the spatial center and variances still give us an idea about the position and the size of an object (or background), so that we can better calculate the saliency value. Moreover, this additional information is beneficial for object-detection applications, as demonstrated in Section 6.2.

4.3.4 Computing the Probability of Saliency

The salient objects tend to be smaller than their surrounding background. As they do not calculate the position and the size of an object, to map the spatial variance to visual saliency, Perazzi et al. [15] and Cheng et al. [16] favor small spatial variations by using an inverting function. This creates a bias towards smaller objects.

In our method, we estimate the position and the size of the salient object, thus we can statistically model a mapping from these variables to a saliency probability.

To generate our model, we use the MSRA-A dataset [37] that includes over 20’000 images with salient objects and their enclosing rectangles marked by three persons. The MSRA-1000 dataset is derived from the MSRA-A dataset. Therefore, to generate unbiased statistics, we exclude the images of the MSRA-1000 from the MSRA-A dataset. In Figure 4.13, we illustrate the probability distributions in terms of the width and the height of the salient objects, as well as their distance to the image center.



(a)

(b)

(c)

Figure 4.13: Distributions of object (a) width (b) height, and (c) distance to image center in the MSRA-A dataset based on the ground truth rectangles. All values are normalized by using the image dimensions.

We can see in Figure 4.13 that all probability distributions resemble a Gaussian distribution. Therefore, we model their joint distribution with a multivariate Gaussian function given as follows:

$$\Lambda(\mathbf{p}_i) = \frac{1}{(2\pi)^2\sqrt{|\mathbf{\Sigma}|}} \exp\Big(-\frac{(\mathbf{g}_i-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{g}_i-\boldsymbol{\mu})}{2}\Big)$$
$$\mathbf{g}_i = \begin{bmatrix} \frac{\sqrt{12\cdot V_x(\mathbf{p}_i)}}{n_w} & \frac{\sqrt{12\cdot V_y(\mathbf{p}_i)}}{n_h} & \frac{m_x(\mathbf{p}_i)-n_w/2}{n_w} & \frac{m_y(\mathbf{p}_i)-n_h/2}{n_h} \end{bmatrix}^T \tag{4.13}$$

Here, $P(\mathbf{p}_i)$ is the probability of saliency of an input image with dimensions n_w and n_h. Note that the factor 12 in $\mathbf{g}_i$ comes from (4.11).

The mean vector and the covariance matrix of the joint Gaussian model that is illustrated in Figure 4.13 are given as follows:

$$\boldsymbol{\mu} = \begin{bmatrix} 0.5555 \\ 0.6449 \\ 0.0002 \\ 0.0063 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 0.0231 & -0.0010 & 0.0001 & -0.0002 \\ -0.0010 & 0.0246 & -0.0000 & 0.0000 \\ 0.0001 & -0.0000 & 0.0115 & 0.0003 \\ -0.0002 & 0.0000 & 0.0003 & 0.0080 \end{bmatrix} \tag{4.14}$$

If we analyze $\boldsymbol{\mu}$ in (4.14), we can see that the average height is larger than the average width. This could be due to a tendency of the photographers to take landscape photographs over portraits, in order to emphasize salient objects. In addition, the average position is very close to the image center, thus validating the well-known center-bias phenomenon [79].

4.3.5 Global Contrast

High color-contrast is widely used as a measure of saliency [3, 10, 13, 15, 16, 18, 80]. Once we have the quantized colors and the color differences $w^c(\mathbf{q}_k, \mathbf{q}_j)$, we can easily compute the global contrast of each quantized color as follows:

$$\xi(\mathbf{p}_i) = \sum_{j=1}^{K} h_j \cdot ||\mathbf{q}_k - \mathbf{q}_j||_2, \quad \forall \mathbf{p}_i | \mathbf{c}_i \to \mathbf{q}_k \tag{4.15}$$

Here, h_j is the number of pixels in the j^{th} histogram bin and $\mathbf{q}_j$ is the quantized color that corresponds to that bin. State-of-the-art methods, such as FT [10], LC [80], and HC [13], rely only on global color-contrast. In order to generate a final saliency map, our method combines global color-contrast with the probability of saliency.

4.3.6 Computing the Final Salient Object Map

In order to combine the probability of saliency and the global contrast into a single salient object map, we use the following approach:

$$s(\mathbf{p}_i) = \frac{\sum_{j=1}^{K} w^c(\mathbf{q}_k, \mathbf{q}_j) \cdot \Lambda(\mathbf{p}_i) \cdot \xi(\mathbf{p}_i)}{\sum_{j=1}^{K} w^c(\mathbf{q}_k, \mathbf{q}_j)}, \quad \forall \mathbf{p}_i | \mathbf{c}_i \to \mathbf{q}_k \tag{4.16}$$

Here, $s(\mathbf{p}_i)$ is the saliency value of the pixel at $\mathbf{p}_i$. All of the computations for $\Lambda(\mathbf{p}_i)$, $\xi(\mathbf{p}_i)$, and $s(\mathbf{p}_i)$ can be done using quantized colors. Therefore, our implementation performs the calculations by using K colors and assigns, based on their color quantization bins, the corresponding saliency values to individual pixels. The final computational complexity of our method is $O(N) + O(K^2)$, where $O(N)$ comes from the histogram computation and $O(K^2)$ comes from the bilateral filtering and other quantization related computations. A color weighting is used in (4.16) for smoother saliency values. After computing the final saliency map, we normalize the map between 0 and 1.

4.3.7 Execution Time

The color quantization step greatly reduces the computational complexity of our method while still retaining the saliency accuracy. Our algorithm estimates the visual saliency of an image in the MSRA-1000 and the SED-100 datasets in, on average, 5.5 and 4.3 ms, respectively. The comparison of execution times is given in Table 4.2. Note that the most time consuming step (superpixel segmentation) in GMR is implemented in C++ and it processes the MSRA-1000 images in approximately 200 ms, on average.

Table 4.2: Average computation time (in miliseconds) for the MSRA-1000 (12×10^4 pixels per image) and the SED-100 (8.7×10^4 pixels per image) datasets.

	Accurate			**Fast**			
	GMR [18]	RC [13]	GC [16]	FT [10]	HC [13]	LC [80]	**FASA**
MSRA-1000	262	180	68	16	12	3	**5.5**
SED-100	214	121	50	13	10	3	**4.3**
Code	C++*	C++	C++	C++	C++	C++	C++

To reduce the computation time, the methods LC, HC, RC, and GC execute a color quantization step that is similar to ours. They quantize the colors in sRGB space by using either 255 bins (LC, independent histograms for each channel) or 12 bins (HC, RC, GC) per channel. Whereas, we perform the quantization in the perceptually more uniform CIEL*a*b* color space and adjust the histogram using the minimum and the maximum values of L*a*b* channels of the processed image. Consequently, we need only 8 bins per channel and obtain a better and faster representation of the image.

4.3.8 Discussion

In the bilateral-filtering approach, we effectively change the shape of the focus regions and we thus generate uniform salient-object maps. In addition, we estimate the size and the position of an object from its spatial variance and use it to compute the probability of saliency. Although the color quantization step speeds up the computation, this method evaluates the salient-object map by using only the colors of the pixels, which is not sufficient for uniformly detecting complex objects.

4.4 Segmentation-Based Approach

The most recent and best-performing salient-object detection algorithms, summarized in Chapter 2, evaluate the final salient-object map by combining biologically-plausible [3] features. However, their performances depend highly on the heuristic function selection and the tuned parameters, because these features do not always imply saliency; the relationship between the parameter values and saliency is not always straightforward. For example, in Figure 4.14(a), we should not assume border pixels are non-salient, because the salient object touches the image boundary. However, in Figure 4.14(b), non-salient sky pixels have high contrast with the rest of the image, and should be classified as background. These handcrafted rules, as in this case, can contradict each other and can lead us to inaccurate salient-object maps.

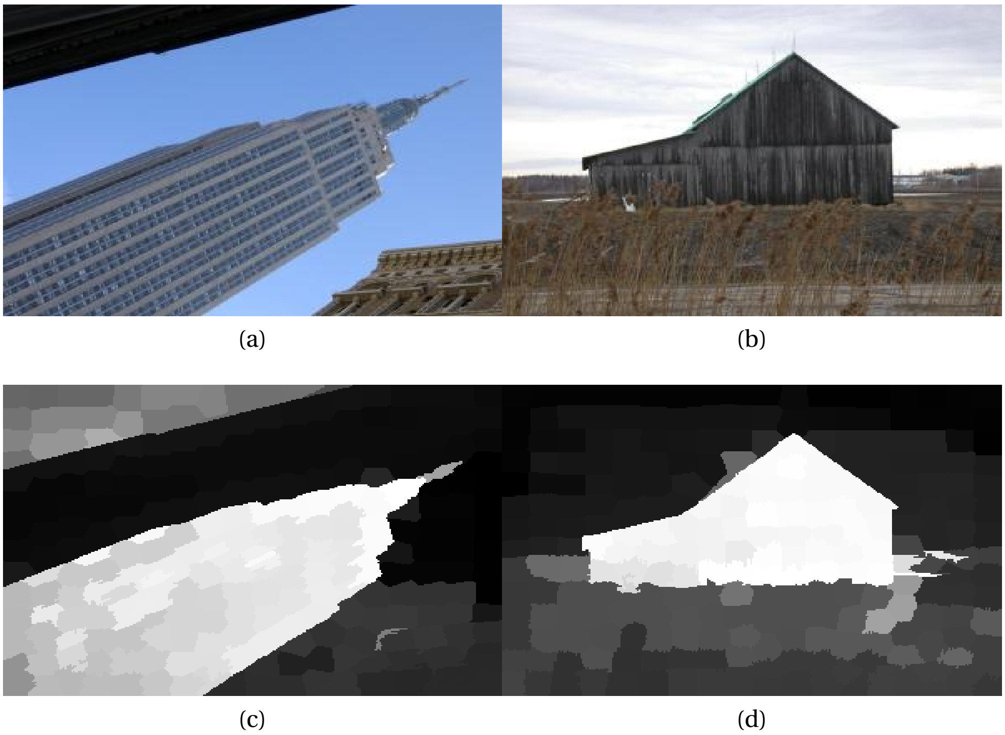


Figure 4.14: Image examples requiring contradicting saliency rules, and outputs of our saliency method.

Instead of crafting heuristic functions for saliency, we can *learn* the contributions of visual features to object saliency. Therefore, in our third method, we propose a segmentation-based approach that uses machine learning for salient object detection. Image segmentation eliminates texture-based inaccuracies in salient object maps. Moreover, as segmentation boundaries coincide with image edges, the shape of the focus and fringe regions becomes context aware. The neighborhood information of image segments provide an opportunity to use spatial connectivity mechanism that was shown to perform well on salient object detection in Section 2.2.3.

4.4.1 Overview of the Method

Our third method employs machine learning in salient-object detection. Therefore, it has a training and test phase, both of which are illustrated in Figure 4.15. In the first step of the training phase, we extract superpixels of an image. Although superpixels provide content-aware focus-fringe pairs, the size of a salient object is still unknown. Therefore, we iteratively merge superpixels using their average color vectors and obtain a hierarchical representation, which is helpful for adjusting the size of the focus-fringe pairs. We then extract visual features from these hierarchical image segments and use them to learn a salient-object detection model.

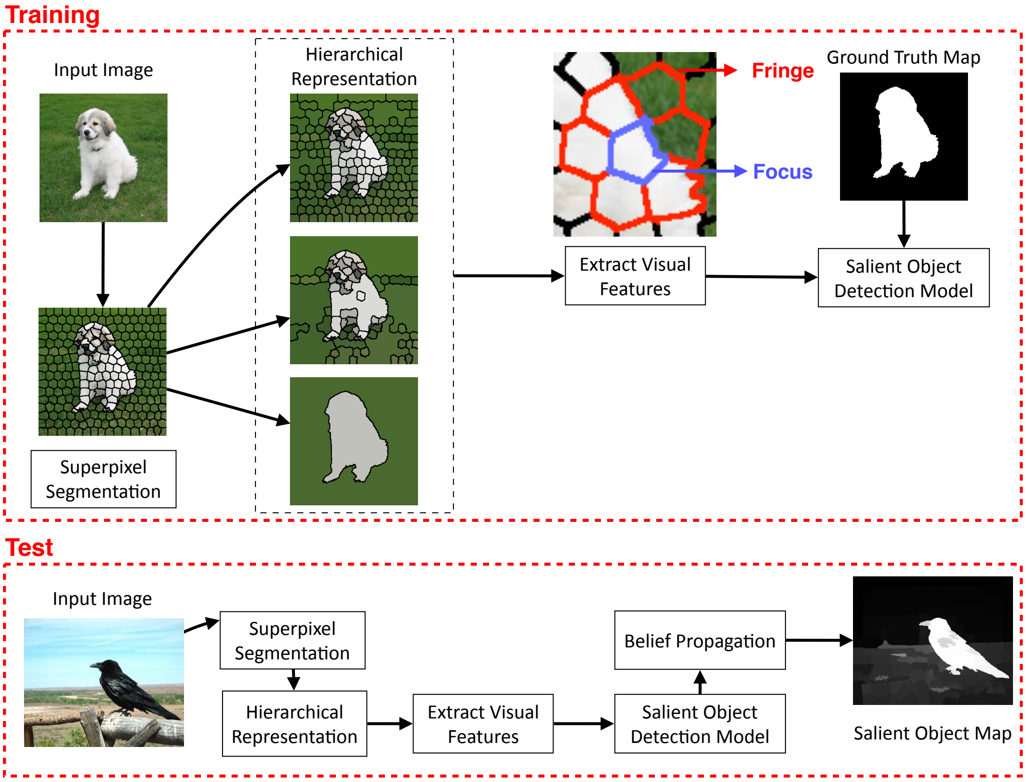
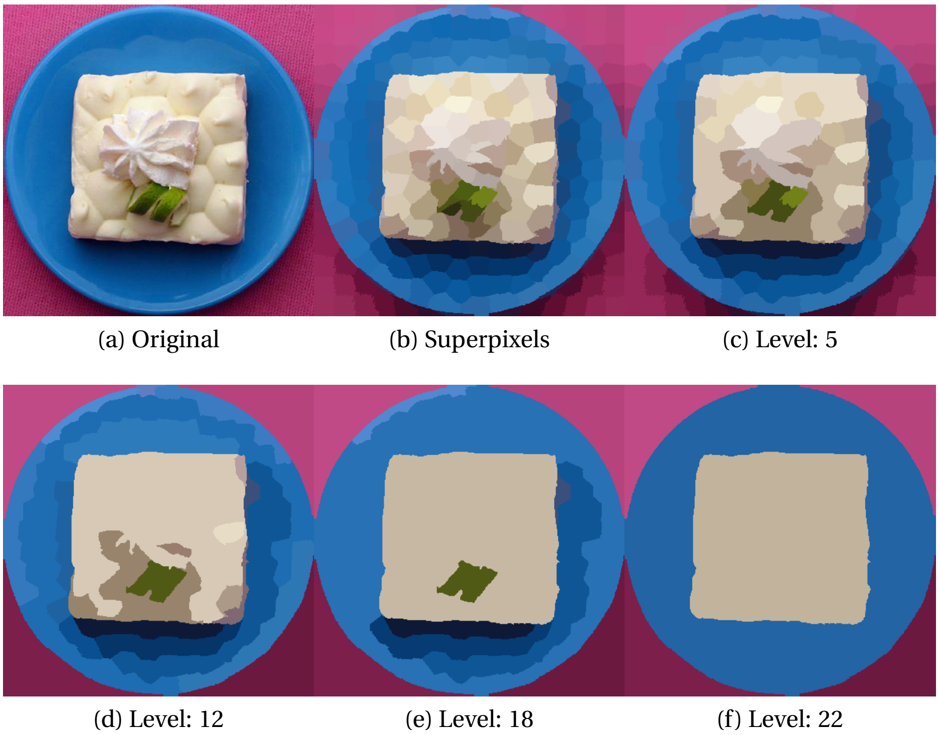


Figure 4.15: The training and text phases of our segmentation-based method. In this approach, the shape of the both focus and fringe regions follow image edges.

In the test phase, we follow the same steps and use our salient object detection model to classify hierarchical image segments as salient or non-salient. We then combine the results into a final salient-object map by using belief propagation on the hierarchical relationship between image segments.

4.4.2 Hierarchical Representation

We oversegment an input image into superpixels by using Achanta et al.'s method [81]. This method produces hexagon-like image segments, which are very useful in forming a well-defined graph representation of an image. The extracted superpixels are iteratively merged into larger segments. For the merging operation, we first find the spatial neighbors of each superpixel. If two spatially neigboring superpixels are each other's nearest neighbors (in terms of mean color vectors) in CIELAB space as well, these superpixels are merged into a larger segment. After all possible superpixel pairs are merged, their properties, such as segment size and average-color vector, are updated and the method moves onto the next level. The algorithm stops if there is nothing left to merge. The merging method is simple and non-parametric, yet it generates an accurate hierarchical representation of the input image. In Figure 4.16, hierarchical levels of an image are shown.



(a) Original (b) Superpixels (c) Level: 5

(d) Level: 12 (e) Level: 18 (f) Level: 22

Figure 4.16: Hierarchical representation of an image.

4.4.3 Image Segment Features

Based on the state-of-the-art methods explained in Chapter 2, we choose the visual features in Table 4.3. For feature extraction, we compute the following variables for each Φ_i^l (the i^{th} superpixel at hierarchy layer l):

- The average color vector in CIELa*b* color space - $\mathbf{c}_i^l$
- Color histogram in CIELa*b* color space - $\mathbf{h}_i^l$
- Histogram of oriented gradients [77] - κ_i^l
- The number of pixels in Φ_i^l normalized by the number of pixels in the image - a_i^l

For the first four rows in Table 4.3, the visual features are computed both locally and globally. Local features are calculated using the set of superpixels Ω_i^l that are spatial neighbors of Φ_i^l. For example, we compute the set of color differences $\left\{||\mathbf{c}_i^l - \mathbf{c}_j^l||_2\right\}_j$, where $j \in \Omega_i^l$. We then calculate four features from this set using four simple functions: $g(.) \in \left\{\text{min}, \text{max}, \text{mean}, \text{median}\right\}$. These functions are helpful to differentiate superpixels from each other. For example, a superpixel at the border of a salient object would have a small minimum color-contrast and a large maximum color-contrast. Whereas, on higher hierarchical layers, if a superpixel covers an entire object, the minimum color-contrast would be large as well. Our machine-learning step can use this difference to distinguish a part of an object from a whole object.

Similar to local features, global features are extracted using a set of superpixels, except this time, the set Ω^l contains all superpixels at layer l. Local-feature extraction regards neighboring superpixels as the fringe region. Whereas, global features consider the fringe region to be the whole image. For average color, color histogram, HOG, and size difference, we extract four local and four global features. We have 41 features in total.

Table 4.3: The visual features that are used in regression trees.

Features for Φ_i^l	Description	Global	Local
Color difference	$g(\lvert\lvert\mathbf{c}_i^l - \mathbf{c}_j^l\rvert\rvert_2)$	$j \in \Omega^l$	$j \in \Omega_i^l$
Histogram difference	$g(\lvert\lvert\mathbf{h}_i^l - \mathbf{h}_j^l\rvert\rvert_2)$		
HOGs difference	$g(\lvert\lvert\kappa_i^l - \kappa_j^l\rvert\rvert_2)$		
Size difference	$g(\lvert a_i^l - a_j^l\rvert)$		
Number of pixels	a_i^l		
Dimensions	X_i^l, Y_i^l		
Color entropy	Entropy($\mathbf{h}_i^l$)		
HOGs entropy	Entropy(κ_i^l)		
Spatial variance	see Section 2.2.2		
Edge fit	see Chapter 5		
Border pixels	see Chapter 5		

4.4.4 Learning Regression Trees

A superpixel Φ_i^l is represented by its 41-dimensional feature vector. The precision of an individual superpixel is calculated as follows:

$$\text{Precision of } \Phi_i^l = \frac{{t_p}_i^l}{{t_p}_i^l + {f_p}_i^l} \tag{4.17}$$

Here, ${t_p}_i^l$ and ${f_p}_i^l$ are the number of true and false positives, respectively, and are computed using Φ_i^l and the ground truth.

In our method, we use the superpixel features in Table 4.3 as input and the precision values as output. The training is performed using the gradient boosting-tree implementation in [58]. Gradient boosting-trees divide the visual-feature space into rectangular (due to thresholding) sub-spaces by adding regression trees via residual fitting (gradient descent). This operation has two main advantages over other learning methods such as SVM. First, by analyzing the sub-spaces, it is possible to understand the relationship between visual features and saliency. For example, let there be a two-dimensional sub-space (high global color-contrast and medium superpixel-size) that contains many salient superpixels and very few non-salient superpixels. Then the specific combination of these features are powerful for detecting saliency. Second, we can progressively verify the regression accuracy of a local minimum.

In order to avoid overfitting, we use a shrinkage factor [82] and trees of depth 2. The shrinkage factor reduces the learning rate and improves the generalization ability of our model. The boosting trees perform at most three threshold operations (from root to leaf), provides balance between a decision stump (one feature) and a strong tree using all 41 features ($\log_2(41) \approx 5.4$). We find 1000 trees to be the saturation point for the training and validation accuracy. We test each image by employing a leave-one-out cross-validation in its dataset.

4.4.5 Feature Analyses

The importance of a feature is correlated to how frequently it is selected for thresholding on the nodes of regression trees. In Figure 4.17, the five most frequently selected features for the MSRA-1000, SED-100, and SOD datasets are illustrated. The “Border pixels" feature is one of the most frequent features. This implies that it is a powerful feature for distinguishing a salient object from a background region. In addition, it is more frequent in the MSRA-1000, because salient objects seldom reside on image boundaries in this dataset. In terms of spatial variance, the X dimension is frequently selected. This can be related to the fact that the field of vision of humans is larger in horizontal dimensions, allowing them to easily identify the horizontal variations (or salient objects). Global “Color difference" features are more frequent than their local counterparts (not in Figure 4.17 due to its frequency), because they give plausible results on every hierarchy level, whereas the Local “Color difference" features are only powerful at the correct level (or scale), where the salient object is properly segmented.

In Figure 4.18, we summarize the effect of certain features on the estimated saliency value. For example, as we mention in Section 4.4, color difference does not directly imply saliency in any of the datasets. The saliency value is correlated to the entropy of the color histogram. This means that segments with low entropy, i.e., regions with small color variations, such as sky, sea, and grass, are less likely to be salient. The correlation decreases from the MSRA-1000 to the SED-100, which makes the MSRA-1000 an easier dataset compared to the others. This is also supported by the results in Section 4.5. Also, small distributions (variances) tend to be salient, where the effect of the X dimension is significantly more prominent than the Y dimension.

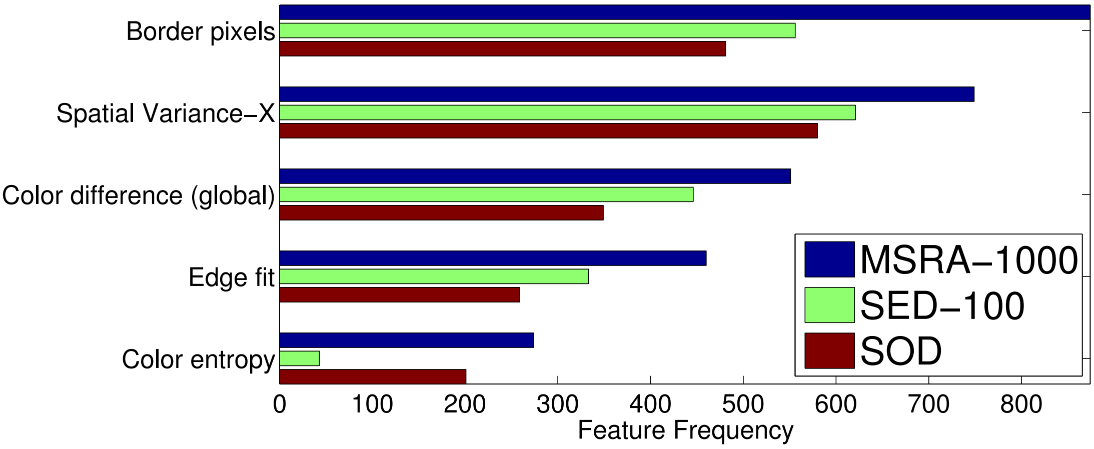


Figure 4.17: The five most frequently selected features.

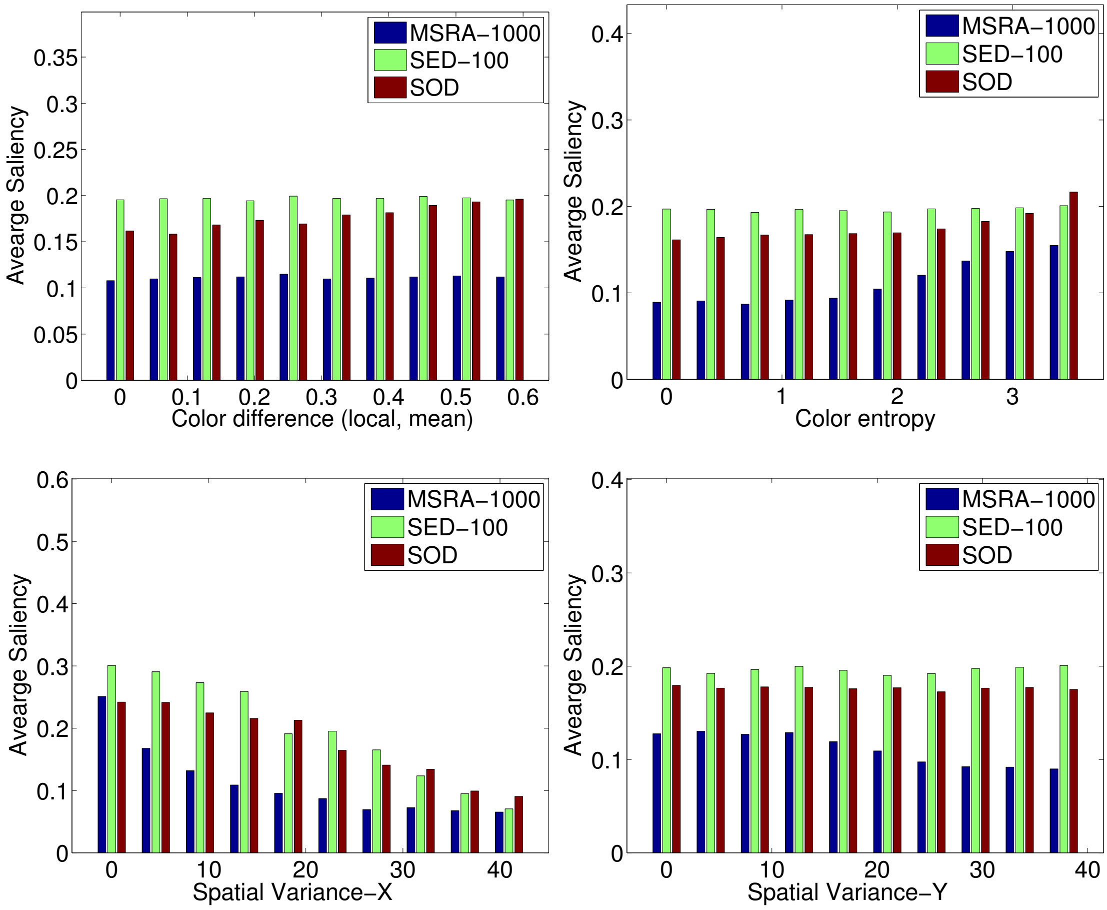


Figure 4.18: The effect of some features to saliency value (see the text for further explanations).

4.4.6 Combining Levels with Belief Propagation

Naively combining the individual saliency maps of different levels results in an inaccurate saliency estimation [19]. Therefore, we employ a belief-propagation inference method [83]. This method effectively solves the following energy minimization problem:

$$\text{Energy} = \sum_{h=1} \sum_{i} ||\hat{s}_i^l - s_i^l||_2^2 + \sum_{h=1} \sum_{i,\Phi_i^l \subseteq \Phi_j^l+1} ||\hat{s}_j^{l+1} - \hat{s}_i^l||_2^2 \tag{4.18}$$

Here, $\hat{s}_i^l$ is the saliency value after the inference, s_i^l is the initial saliency value that is estimated by the regression trees for Φ_i^l. The two terms in the equation represent the data penalty and the smoothness penalty along hierarchy levels, respectively. The final saliency value $F(.)$ of a pixel at position $\mathbf{p}$ is calculated as:

$$F(\mathbf{p}) = \sum_{h=1} \sum_{j,\mathbf{p} \subseteq \Phi_j^l} \hat{s}_j^l \tag{4.19}$$

4.4.7 Discussion

Single-scale, multi-scale, and bilateral-filtering approaches have certain drawbacks, such as undesired false positives in the background, non-uniform saliency values within object pixels, and susceptibility to image texture. Our segmentation-based approach overcomes these problems with hierarchical image segmentation followed by a machine-learning-based salient-object detection model. Therefore, we use this method as a baseline to our multi-level salient-object detector in Chapter 5.

4.5 Performance of the Proposed Methods

We compare the performances of our proposed methods with the state-of-the-art methods that were discussed in Chapter 2 by using the well-known datasets, such as MSRA-1000, SED-100, and SOD using the evaluation metrics in Section 2.3. The state-of-the-art methods are referred to by their acronyms, which are given in Table 2.1. Our multi-scale filtering, bilateral-filtering, and segmentation-based approaches are referred to as MSF, FASA, and HR, respectively.

According to the precision-recall curves in Figure 4.19(a)-(c), our segmentation-based method (HR) performs better than all other methods. Although AMC, GMR, HSD, and CH employ the spatial-connectivity mechanism on image segments, they use hand-crafted functions that map low-level visual features, such as color contrast and edges, to object saliency. Whereas, in HR, we used machine learning to model this mapping. As we showed in Figure 2.8, 2.11, and 2.13, the MSRA-1000 dataset has salient objects with high color-contrast. Therefore, the performance difference between HR and other methods is not significant. Whereas for the SED-100 and the SOD datasets, the difference is apparent, because these datasets include more variety in terms of aspects of saliency compared to the MSRA-1000 dataset.

The same performance trend is followed in F-measure values in Figure 4.19(d)-(f), where HR outperforms the other methods. Because, F-measure is an adaptive operating point on the precision-recall curve. The methods that employ spatial connectivity produce uniform salient-object maps. When they are binarized with an adaptive threshold, they yield accurate results and high F-measure values. Note that, both precision-recall curves and F-measure use a binarized salient-object map for evaluation. This approach does not take varying importance of objects into account.

The mean absolute-errors of the methods on the foreground pixels are illustrated in Figure 4.20. A lower error-value indicates that a method is able to uniformly highlight an object as being salient. Here, we can see that methods that employ image segmentation surpass the other methods, because the segmentation step makes a salient-object detector robust against noise and texture and provides uniform salient-object maps. HR has the lowest foreground error, because it uses the hierarchical layers to learn and estimate the object saliency, and it combines the results with belief propagation, enforcing top-down uniformity.

When we look at the mean absolute error of the background pixels in Figure 4.20, the order of the methods changes. Although the best method in terms of suppressing the background is GMR, spatial-variance-based methods, SF and GC, also outperformed other techniques, because they assume spatially distributed pixels as background, which can provide accurate background suppression for usually non-salient regions, such as sky, grass, and water. Our method HR averagely performs on mean absolute error on background pixels. This can be due to the mistakes hierarchical representation, such as connecting an object and a background superpixel at the very early layers of the hierarchy. We replace the greedy superpixel merging in HR with a more robust Minimum Spanning Tree (MST) based merging approach in our Comprehensive Salient Object Detection (CSD) method.

Due to its superior performance in salient-object detection, we choose HR method as a baseline to our final Comprehensive Salient Object Detection (CSD) method in Chapter 5, which is extended to estimate their multi-level saliency values.

4.6 Summary of the Chapter

In this chapter, we have presented three binary salient-object detection methods with successive improvements. The first method is based on multi-scale filtering and provides a solution to the background suppression problem in single-scale filtering and optimal scale-parameter selection problem. Our second method applies bilateral filtering to images and uniformly detects salient object maps. It is also very fast and detects the position and the size of the objects by using spatial variance, instead of biasing small objects. In our third method, we use image segmentation for a method that is robust against image textures, and we employ machine learning to minimize hand-crafted salient-object detection functions. This method combines multiple hierarchical layers with spatial connectivity and performs the best compared to our other methods and the state-of-the-art methods.

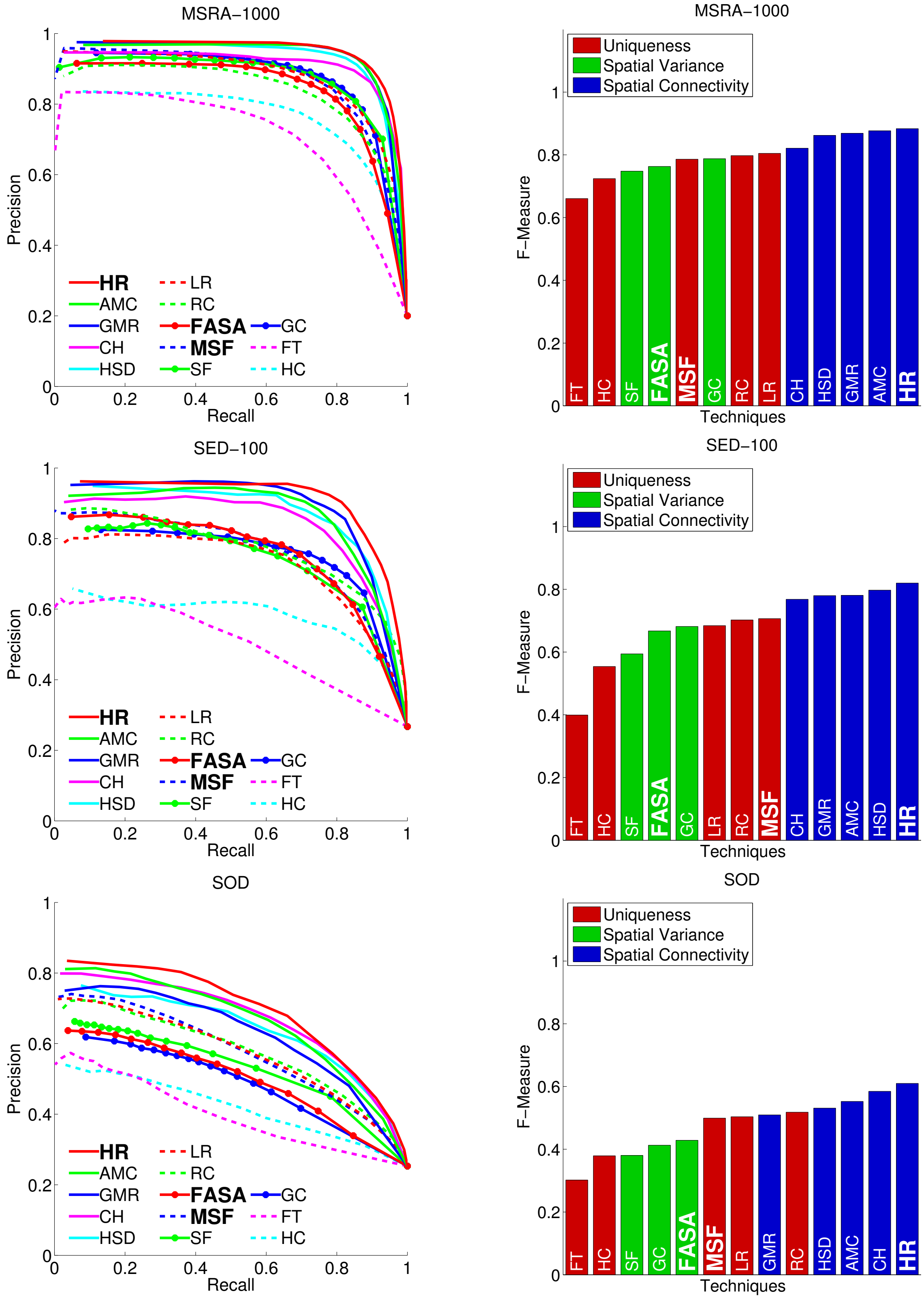


Figure 4.19: Performance comparisons of our methods (highlighted in bold) and the state-of-the-art methods on the well-known datasets using precision-recall curves and F-Measure.

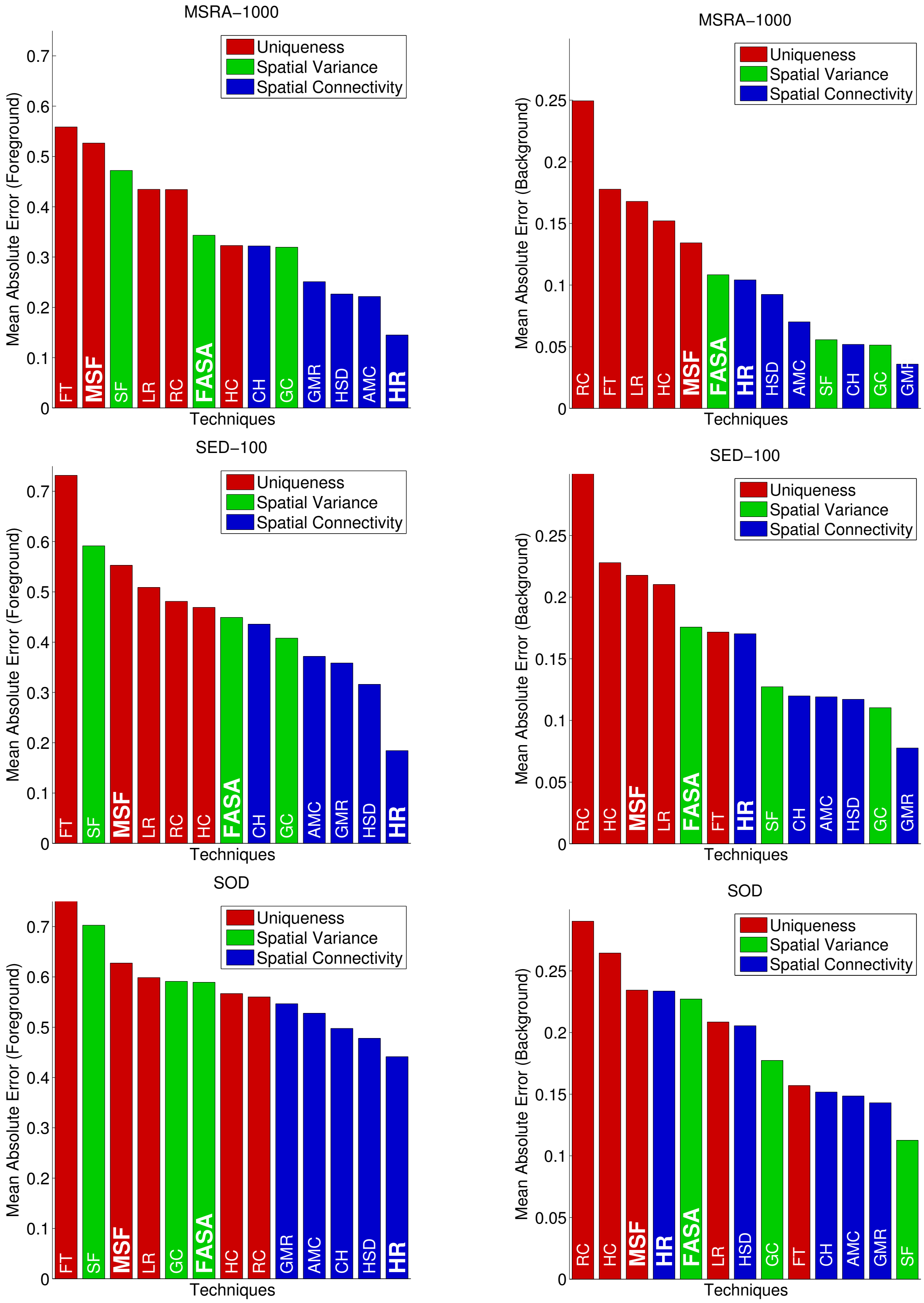


Figure 4.20: Performance comparisons of our methods (highlighted in bold) and the state-of-the-art methods on the well-known datasets using mean absolute error of the foreground and the background regions,.

5 Comprehensive Salient Object Detection

In this chapter, we present our Comprehensive Salient Object Detection (CSD) method that is capable of finding multiple salient objects and estimating their multi-level saliency value. This chapter consists of three parts. First, we discuss the "Object-Awareness" concept, an algorithm step we use to convert binary object-saliency to multi-level. Second, we explain Comprehensive Salient Object Detection (CSD) in detail; it is an extended version of our segmentation-based salient-object detector (HR) in Section 4.4 with the OAM. Third, we compare our method to the state-of-the-art methods on the COS dataset introduced in Chapter 3 and show that CSD significantly outperforms other methods in estimating multi-level object saliency.

5.1 Object Awareness: Binary to Multi-Level Object Saliency

In Chapter 4, we proposed three methods for finding salient objects and generating binary salient-object maps. Here, we want to design a robust post-processing step for these methods; it can estimate the multi-level object saliency values, even when estimated binary salient-object maps are not very accurate.

An example post-processing operation is illustrated in Figure 5.1. Here, we compute the binary salient-object map (non-ideal) using our segmentation-based method in Section 4.4. We then extract low-level visual features, such as the contrast between the average color of an object and its local surroundings (see Figure 5.1). We randomly divide 2434 objects in our COS dataset into training and test sets. We train a multi-level object-saliency estimation model with low-level visual features of the training set using Support Vector Regression (SVR). We then pass the features of the objects of the test set through the model and obtain multi-level object-saliency estimations. In order to evaluate the accuracy of our model, we compute the coefficient of determination (R^2) between estimated and real multi-level object-saliency values.

In Figure 5.2, we illustrate the multi-level object-saliency estimation performance of two different sets of visual features: all features, and only position and size features. We can see that the two sets perform fairly comparably. This is an important result, because the quality of contrast-based features depend highly on the object-boundary accuracy of the salient-object map in Figure 5.1. Whereas, the position and the size features are not significantly affected, even though salient-object boundaries are not

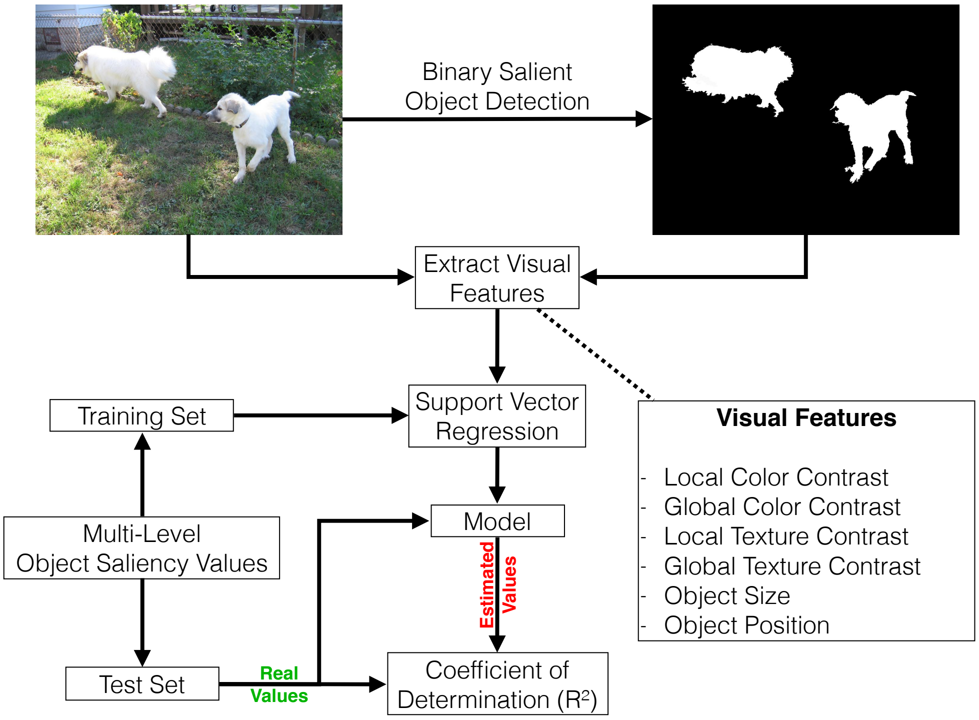


Figure 5.1: The two setups to test how accurate low-level features are in estimate multi-level object saliency, given the object segmentations or rectangles.

correctly detected. This shows that position and size information is sufficient to estimate multi-level object saliency and is robust against segmentation errors. We design our post-processing step, which we call Object-Awareness Model (OAM), by using estimated position and size of an object to determine its multi-level object-saliency values. This simple model can also be used by other salient-object detection techniques, provided that they estimate the position and the size of the salient objects or they compute the hierarchical representation of an input image.

Note that, R^2 value in Figure 5.2 decreases from eye-tracking to rectangle-drawing saliency. This is expected, because low-level visual features are related to the subconscious attention more than the focused attention (see Table 3.1). Because, focused attention, such as guessing the position and the size of an object in rectangle-drawing experiments, involves high-level cognitive functions.

We extend our segmentation-based method (HR) in Section 4.4 with the OAM and obtain a multi-level salient-object detector called Comprehensive Salient Object Detection (CSD) as shown in Figure 5.3.

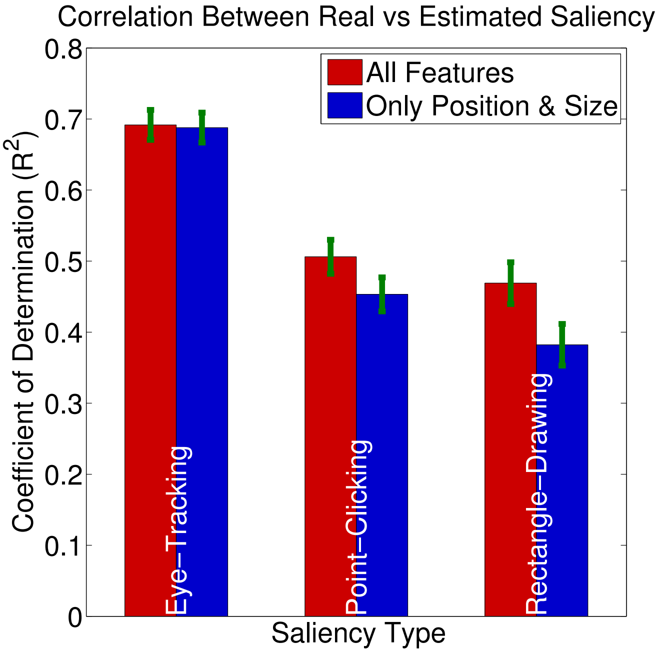


Figure 5.2: The correlation between real and estimated multi-level object saliency values for three different saliency types. The green lines indicate the standard deviation over 10 experiments.

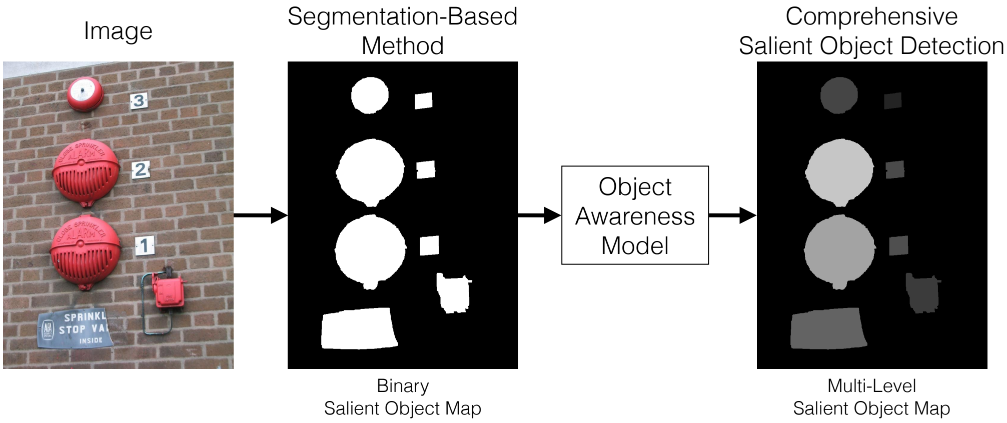


Image
Segmentation-Based Method
Object Awareness Model
Comprehensive Salient Object Detection
Binary Salient Object Map
Multi-Level Salient Object Map

Figure 5.3: The segmentation-based method is extended to CSD using Object-Awareness Model. The resulting a method (CSD) is capable of estimating multi-level object saliency.

5.2 Overview of the CSD Method

Comprehensive Salient Object Detection (CSD) method is capable of finding the salient objects with their multi-level object-saliency values. The flowchart of CSD is illustrated in Figure 5.4. As this method employs machine learning for salient-object detection and multi-level object-saliency estimation, it has training and test phases.

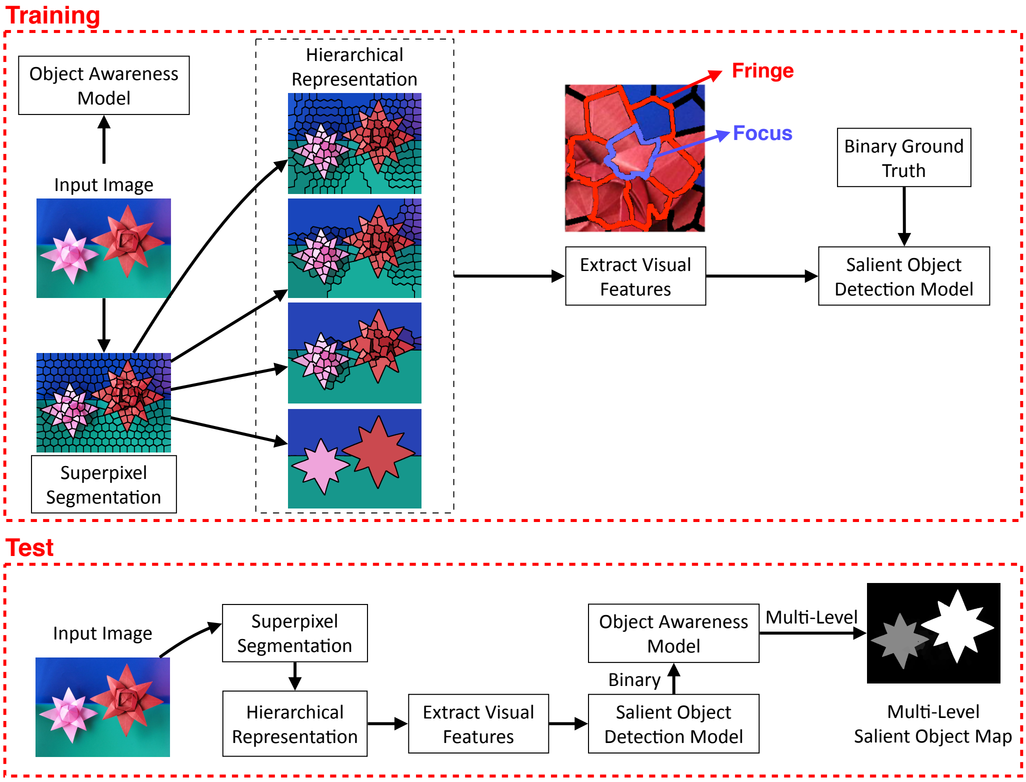


Figure 5.4: Flowchart of Comprehensive Salient Object Detection

5.2.1 Training Phase

In order to perform multi-level salient-object detection, our method oversegments an image into superpixels using the algorithm in [81]. The superpixel segmentation divides an image into a puzzle-like representation and significantly reduces the number of processing blocks from hundreds of thousands pixels to hundreds of superpixels.

Although they usually follow the object-to-object and object-to-background image edges, superpixels do not represent an object as a whole. This representation can be achieved by merging similarly-colored superpixels and by forming hierarchical image segments. The segmentation-based algorithm (HR) we propose in Section 4.4 iteratively merges superpixels into larger ones in a greedy fashion. This strategy might lead to unpredictably large number of hierarchical layers, which might create redundancy and slow down the execution time of the algorithm. In CSD, we instead use a Minimum Spanning Tree (MST) based approach and control the number of hierarchical layers. These layers will be useful later when we employ the Object-Awareness Model (OAM).

After the hierarchical representation step, we extract several low-level visual features from the superpixels, at each hierarchy layer. We train Gradient Boosting Trees by using superpixel features and their saliency values for salient-object detection. Instead of multi-level ground-truth maps, in training, we use binary ground-truth maps for two major reasons. First, as the training and testing are performed on image segments and not on whole objects (which are unknown), in our experiments, classifying object and background segments yielded better performances than regressing their multi-level saliency values did. Because, the feature values of the segments that belong to the same object might significantly vary and cause a noisy regression. Second, separating multi-level salient-object detection into two steps, which are finding salient objects and estimating multi-level object-saliency values, makes the extension of binary saliency to multi-level saliency very modular. Because, an OAM can be added to any salient object detection algorithm, provided that it estimates the position and the size of the object or gets this information from objectness algorithms [84, 85].

In CSD, in order to add the capability of estimating the multi-level object saliency, we train three OAMs for each ground-truth type in our dataset (eye-tracking, point-clicking, rectangle-drawing). In Section 5.2.7, we show that hierarchical representation actually is useful for OAMs.

5.2.2 Test Procedure

When we want to estimate the multi-level salient-object map of an image, we follow a procedure similar to that of the training procedure (superpixel segmentation, hierarchical representation, feature extraction). We then classify image segments using the salient-object detection model. As the position and the size of the objects are unknown, we regard the image segments as "proto-objects" in OAMs and estimate a multi-level object saliency of each segment. The final salient-object map is obtained using max-pooling on hierarchical layers.

5.2.3 Superpixel Segmentation

In Section 2.2.3, we show that the salient-object detectors that use spatial connectivity perform better than the other methods. Therefore, in our method, by using superpixel segmentation [81], we oversegment an input image into similarly colored clusters.

The average size of a superpixel should be small enough to correctly segment the smallest object (7×7 pixels) in COS dataset. If we use a superpixel of size 7×7 pixels, we would get 16'000 superpixels in a typical 1024×768 pixel image, which is redundant and slows down salient-object detection. Therefore, in order to compromise between having a small number of superpixels and low under-segmentation errors, we set the number of superpixels to 1024, which translates to a square of 27×27 pixel. Note that, this square is an initial value and can change size and shape with each iteration during superpixel segmentation [81]; a superpixel can still cover a 7×7 object.

5.2.4 Hierarchical Representation

Superpixel segmentation step divides an image into manageable chunks. However, we do not know which superpixel belongs to which object in the image. Merging superpixels that belong to the same object can create a high-level representation of the image and can increase the accuracy of salient object detection. Therefore, in CSD, we generate hierarchical layers by carefully combining superpixels at multiple layers.

In order to compute hierarchical layers, we first form a superpixel graph, where each superpixel is a node and two neighboring superpixels are connected via a weighted edge. The weights of these edges can be calculated as follows:

$$w_{ij} = ||\mathbf{c}_i - \mathbf{c}_j||^2 \tag{5.1}$$

Here, w_{ij} is the weight of the edge between i^{th} and j^{th} superpixels with the average color vectors $\mathbf{c}_i$ and $\mathbf{c}_j$, respectively. An example superpixel graph is shown in Figure 5.5 (a). In our hierarchical representation of an image, we require two properties: First, we want to merge similarly colored superpixels at the low layers of the hierarchy. Second, we want to control the number of hierarchical layers.

In order to satisfy these requirements, we compute the Minimum Spanning Tree (MST) of the superpixel graph using Kruskal's algorithm [86]. An MST is a tree-shaped graph that spans all the nodes, where the total weight of the edges is minimized. The MST that corresponds to a superpixel graph is given in Figure 5.5 (b). We can use a threshold to divide an MST into multiple sub-trees, each of which merge multiple superpixels in one large superpixel. Note that, a threshold of 0 corresponds to the superpixel segmentation itself (no merging).

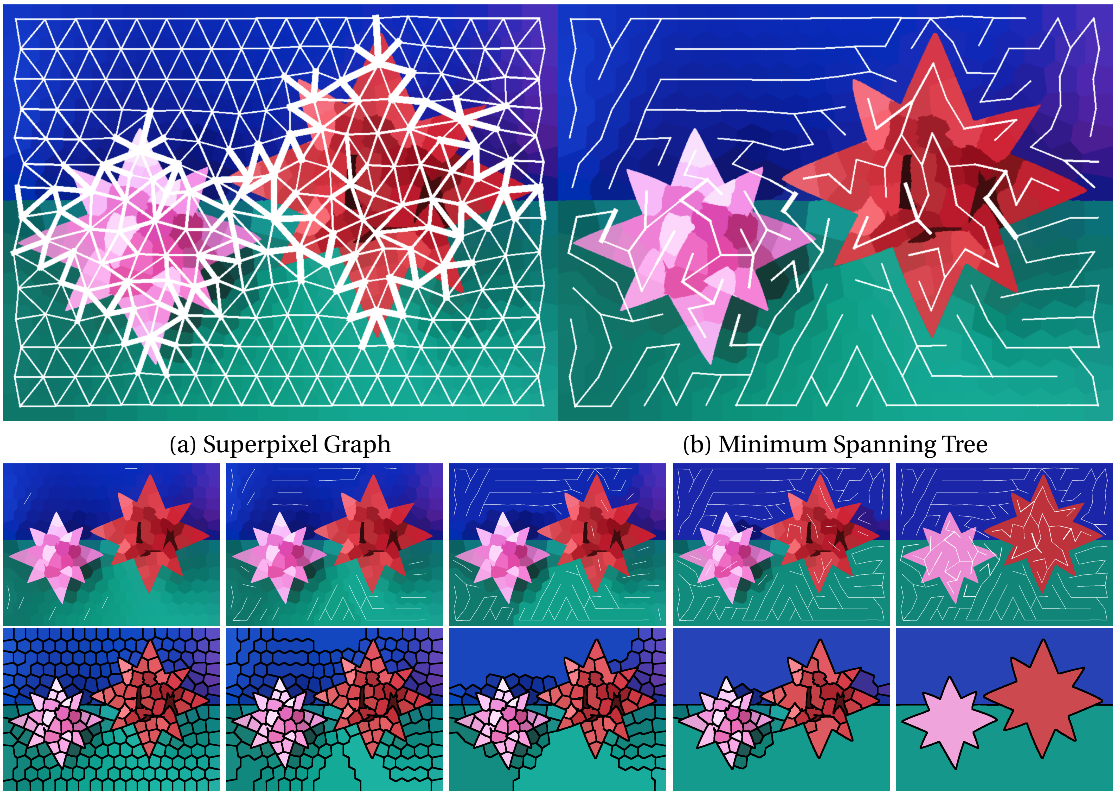
In our framework, as the MST minimizes the total weight of the edges, we merge the similarly colored superpixels in the first couple of layers. In addition, we can vary the threshold from low to high with the desired number of steps, thus control the number of hierarchical layers. Our hierarchical representation step satisfies both of the criteria we mentioned earlier. An example representation is illustrated in the last row of Figure 5.5.

The superpixels in our hierarchical representation have the following properties:

$$\begin{aligned} \Phi_i^l \cap \Phi_j^l &= \emptyset, \qquad \text{for} \quad i \neq j \\ \Phi_i^{l+1} &= \bigcup_n \Phi_n^l \\ \mathbf{I} &= \bigcup_{i=1}^{N_l} \Phi_i^l \end{aligned} \tag{5.2}$$

Here, two superpixels Φ_i^l and Φ_j^l at hierarchical layer l do not share a common pixel, a superpixel at hierarchical level $l+1$ is the union of a set of superpixels at hierarchical layer l, and for any layer l, union of all superpixels (N_l in total) is equal to the image $\mathbf{I}$ itself.

There are two main benefits of hierarchical representation. First, we can employ the OAM for objects of various sizes and shapes. Second, when we combine the multi-level object saliency values of the segments at different layers, the competition between higher and lower layers can correct for errors. In CSD, we use the superpixels at all hierarchy layers in learning the relationship between low-level visual features and object saliency.



(a) Superpixel Graph

(b) Minimum Spanning Tree

Figure 5.5: (a) A graph of superpixels and the corresponding (b) minimum spanning tree. The white lines represent the edges of the graph and the thickness is directly related to the edge weight w_{ij} that connects two nodes. The MST is divided into several components with different thresholds (second row) and superpixels that are connected are merged into bigger ones (third row).

5.2.5 Feature Extraction

In Chapter 3 and Section 4.4, we reviewed various low-level visual features that are related to object saliency. Moreover, in Chapter 4, we discussed the advantages of disadvantages of the size and shape of the focus-fringe pairs in finding salient objects in images. Based on these findings, from each superpixel at each hierarchical layer, we extract the visual features in Table 5.1. We have 37 visual features in total. Here, we represent the feature extraction with a function $F(.)$, where $F(\Phi_i^l)$ corresponds to a feature vector of size 1×37 for superpixel Φ_i^l.

The visual representation of several features are illustrated in Figure 5.6. Here, the local features are extracted by taking the central superpixel (green dot) as the focus and the neighboring superpixels (blue dots) as the fringe region. Whereas, in the global-feature extraction, all superpixels except the central one is used as the fringe region. The edge-fit feature is equal to the ratio between the length of the coinciding edge and the superpixel perimeter. The border pixels feature is equal to the ratio between the number of border pixels and the circumference of the image.

Table 5.1: The low-level visual features that are extracted from superpixels

Name	Size	Explanation
Contrast of Color, Hue, Saturation, Orientation	8	Euclidean distance of the average value of the superpixel to neighboring superpixels (local) or to the whole image (global)
Histogram Contrast of Color, Hue, Saturation, Orientation	8	Chi-squared distance of the histogram of the superpixel to its neighboring superpixels (local) or to the whole image (global)
Orientation Histogram	8	Histogram of oriented gradients [77] of the superpixel
Color Variance	3	Variance of L, a*, and b* values of the pixels in the superpixel
Color Spatial Variance	2	Spatial variance of the average superpixel color (see Section 2.2.2)
Superpixel Position	2	Normalized position (with respect to the image size) of the center of mass of the superpixel
Superpixel Size	2	Normalized size (with respect to the image size) of the size of the superpixel
Aspect Ratio	1	Aspect ratio of the superpixel
Edge Fit	1	The ratio of the pixels at the superpixel perimeter that coincides with image edges to the superpixel perimeter.
Perimeter	1	Normalized perimeter (with respect to the image size) of the superpixel
Border pixels	1	Normalized length (with respect to the image size) of the pixels at the superpixel perimeter that touch the image boundary

5.2.6 Training the Salient-Object Detection Model

In order to train the salient-object detection model, we use Gradient Boosting Trees [87]. The input of the training is the feature vectors of the superpixels $F(\Phi_i^l)$. The target saliency values are the binary numbers that indicate whether more than 90% of the pixels inside the superpixel belong to a salient object; these values can be calculated as follows:

$$\gamma_i^l = \begin{cases} 1, & \text{if} \quad |\mathbf{G}(\Phi_i^l)|/|\Phi_i^l| \geq 0.9 \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

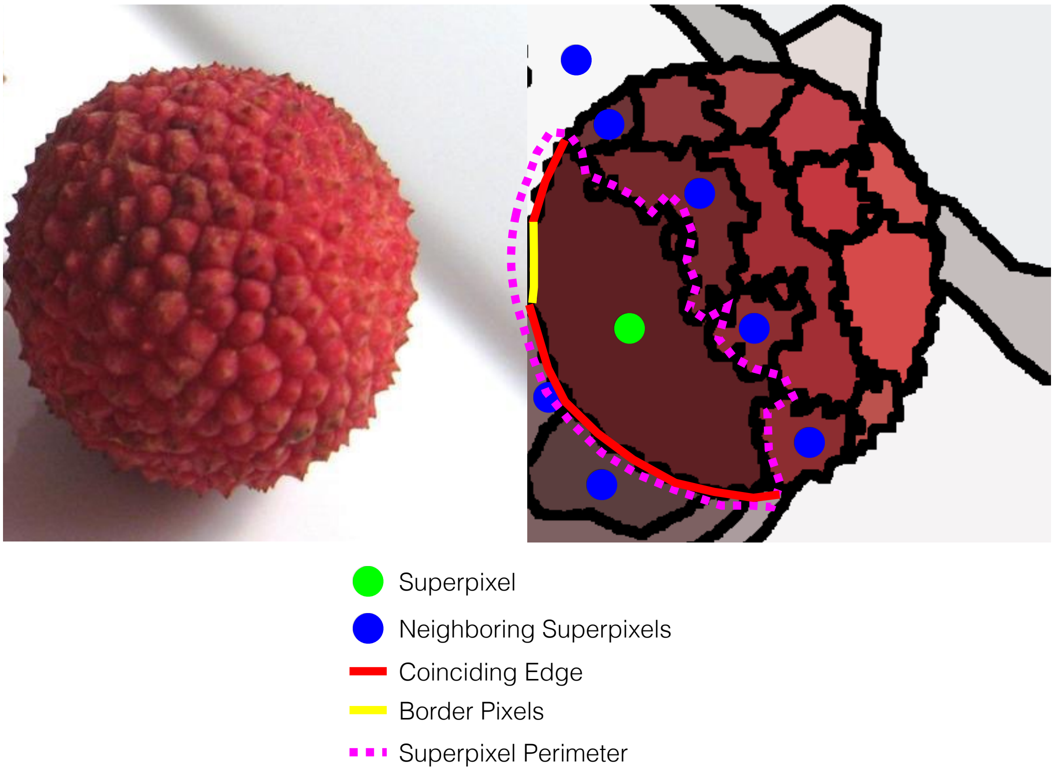


Figure 5.6: The visual representations of the variables that are used in feature extraction.

Here, γ_i^l is the target saliency value of the superpixel Φ_i^l, $\mathbf{G}$ is the ground truth map of an input image, $|\mathbf{G}(\Phi_i^l)|$ is the number of salient pixels in Φ_i^l, and $|\Phi_i^l|$ is the total number of pixels in Φ_i^l. In our experiments, the salient-object detection model is trained by using all of the feature vectors $F(\Phi_i^l)$ and the classification target values γ_i^l.

We apply a 16-fold cross-validation to the images in COS dataset and train 200 gradient boosting-trees with a depth of 3. To avoid the overfitting problems, we keep the depth of the tree at less than $\log_2(37) \approx 5.2$. Note that, the salient-object detection model only classifies the image segments as salient or non-salient. The actual multi-level object saliency will be estimated by the Object-Awareness Model (OAM).

The five most frequently selected features are illustrated in Figure 5.7. Our dataset and the well-known dataset (see Figure 4.17) have two frequently selected features in common: color spatial-variance and global color-contrast. This shows that color is one of the most striking characteristics of an object. The vertical position of an object (Center Y feature) is the second most frequently selected feature. This might be due to the persistent semantic bias. The superpixels that belong to sky and land are usually considered non-salient and reside at the top and the bottom of an image, respectively.

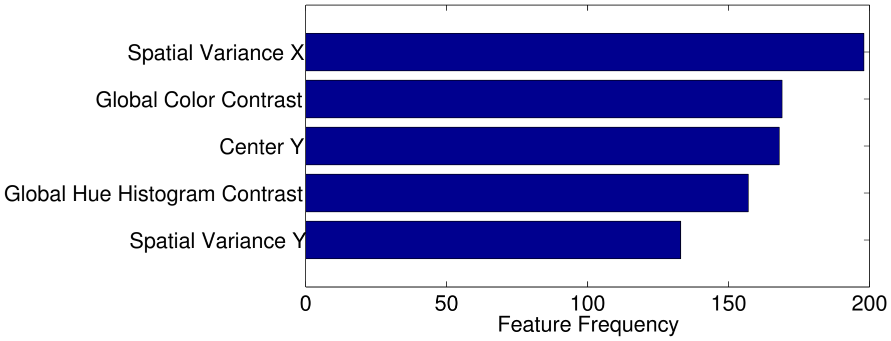


Figure 5.7: The most frequently used visual features in our salient-object detection model.

5.2.7 Training the Object-Awareness Model

As we mentioned in Section 5.1, object awareness has a potential to measure the multi-level saliency values of the objects. Therefore, in our method, we train three OAMs by using gradient boosting trees [87] for regression as shown in Figure 5.8. The inputs of the tree are the position and the size of the salient objects (normalized by the size of the image). The target regression values are the multi-level object-saliency values that are measured using experimental data in Chapter 3. We generate an OAM for each ground-truth map type in our dataset.

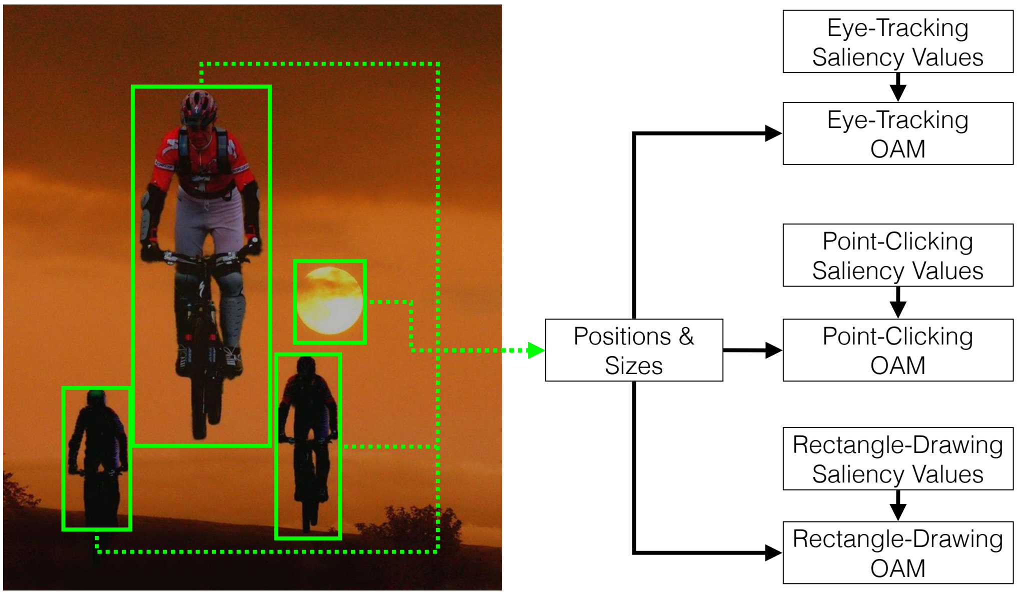

Eye-Tracking Saliency Values
Eye-Tracking OAM
Point-Clicking Saliency Values
Positions & Sizes
Point-Clicking OAM
Rectangle-Drawing Saliency Values
Rectangle-Drawing OAM

Figure 5.8: The Object-Awareness Models (OAM) are trained using the position and the size of the salient objects and their multi-level object saliency values measured with different experiments.

5.3 Finding a Multi-Level Salient-Object Map

After the training procedure, we can use our CSD method to estimate multi-level salient-object map of an input image. Similar to the training, we oversegment an image into superpixels, form the hierarchical layers, and extract low-level visual features from the superpixels. These features are then passed through the Salient Object Detection Model and are classified as salient or non-salient, which generates a binary salient object map for each hierarchical layer.

We do not know the position and the size of the salient objects in a test image. Therefore, we use the size and position of the superpixels in the OAMs instead. Each superpixel is assigned a multi-level object-saliency value, giving us multiple multi-level salient-object maps (one for each hierarchical layer). Finally, these maps are combined into one salient object map via max-pooling as follows:

$$\mathbf{S}_F(x,y) = \max_l \quad \mathbf{S}^l(x,y) \tag{5.4}$$

Here, $\mathbf{S}_F$ is the final salient object map and $\mathbf{S}^l$ is the salient object map at hierarchy level l.

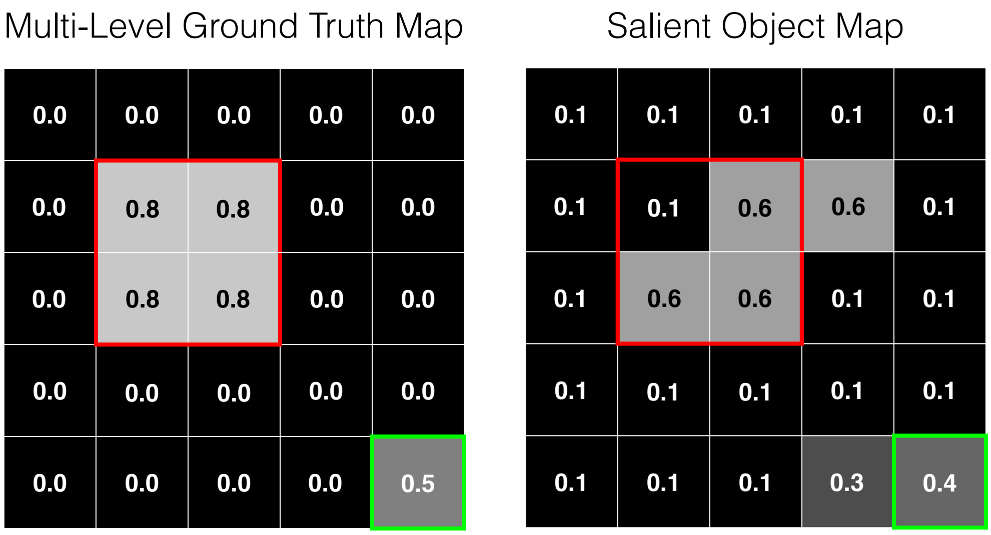
5.4 Performance on Multi-Level Salient-Object Detection

We analyze the multi-level salient object detection performances of the state-of-the-art methods in Chapter 2, our binary salient-object detectors in Chapter 4 and our final CSD method on our Comprehensive Object Saliency (COS) dataset by using two metrics: mean absolute error and salient object ranking. Mean absolute error, which is explained in Section 2.3.3, measures the average pixel-wise absolute difference between a multi-level ground-truth map and a multi-level salient-object map. Whereas, in salient object ranking, we evaluate how well a method can sort objects with respect to their saliency values.

5.4.1 Mean Absolute-Error

Widely-used performance metrics, such as precision-recall curves in Section 2.3.1 and F-measure in Section 2.3.2, are applicable for binary ground-truth maps. Whereas, multi-level salient-object maps consist of continues values. Therefore, we evaluate and compare the performances of the salient-object detectors by using a mean absolute-error metric on salient objects and background regions. In Figure 5.9, an example multi-level ground-truth map and a salient-object map are given. Here, there are two salient objects. The first object, which is outlined with a red line, has a multi-level saliency value of 0.8. Whereas, for the second object, which is shown with a green line, the multi-level saliency value is equal to 0.5.

In order to calculate the Mean Absolute Error (MAE) of a method on a salient object,



Multi-Level Ground Truth Map

0.0	0.0	0.0	0.0	0.0
0.0	0.8	0.8	0.0	0.0
0.0	0.8	0.8	0.0	0.0
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.5

Salient Object Map

0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.6	0.6	0.1
0.1	0.6	0.6	0.1	0.1
0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.3	0.4

Figure 5.9: Two salient objects are illustrated with enclosing red and green lines and multi-level saliency values 0.8 and 0.5, respectively. The same objects are outlined on the salient object map for comparison. Note that the background saliency value of the salient-object map is non-zero.

we use the following formula:

$$\mathrm{MAE}^o = |E^o - s^o| \tag{5.5}$$

Here, MAE^o, E^o, and s^o are the Mean Absolute Error, estimated saliency, and the real saliency value of object o, respectively. We know from Figure 5.9 that $s^1 = 0.8$ and $s^2 = 0.5$. Similarly, we can calculate the estimated saliency values as follows:

$$\begin{aligned} E^1 &= \frac{0.1+0.6+0.6+0.6}{4} = 0.475 \\ E^2 &= \frac{0.4}{1} = 0.4 \end{aligned} \tag{5.6}$$

By using (5.5) and (5.6), we find that $\mathrm{MAE}^1 = 0.325$ and $\mathrm{MAE}^2 = 0.1$. By following a similar procedure, we can also calculate the Mean Absolute Error of the background pixels. For the example in Figure 5.9, $\mathrm{MAE}^b = 2.7/(25-4-1) = 0.135$. Note that, for MAE^b, we subtract the number of object pixels (= 4 + 1) from the total number of image pixels (= 25). We compute the Mean Absolute Error of a salient-object detector by averaging the object errors (over 2434 objects in the COS dataset) and by averaging the background errors (over 588 images in the COS dataset). Note that, we compute the average of the mean absolute-errors over objects not pixels. Therefore, very large objects do not dominate the performance.

In Figure 5.10, we compare the average mean absolute errors of all methods. As we can see, our method CSD has the minimum error in estimating all three types of multi-level object-saliency values (eye-tracking, point-clicking, rectangle-drawing). We incorporated the Object-Awareness Model (OAM) into our HR method and obtain the CSD method. This simple model improves the saliency-estimation performance of CSD when compared to HR, especially for eye-tracking saliency values. Because, there is a strong bias in object position and size in eye-tracking data as illustrated in Figure 3.15. Other methods, such as RC, AMC, and HSD, perform comparably with CSD. However, as illustrated in Figure 5.10(d), their background suppression performances are significantly worse than that of the CSD method.

CH, SF, LR, and MSF seem to have lower background errors compared to our CSD method. However, they are not very accurate in correctly estimating the multi-level object-saliency values. Object and background errors depict a trade-off that is similar to precision-recall curves. An algorithm can be considered better than the others, if it has a low mean absolute error for both objects and background regions. In Figure 5.11, we plot the object and the background errors on a 2D graph to emphasize on the accuracy trade-off. Here, the distance from the origin is inversely related to the overall performance of an algorithm. The CSD method outperforms other methods and unlike the well-known datasets, there is still room for improvement.

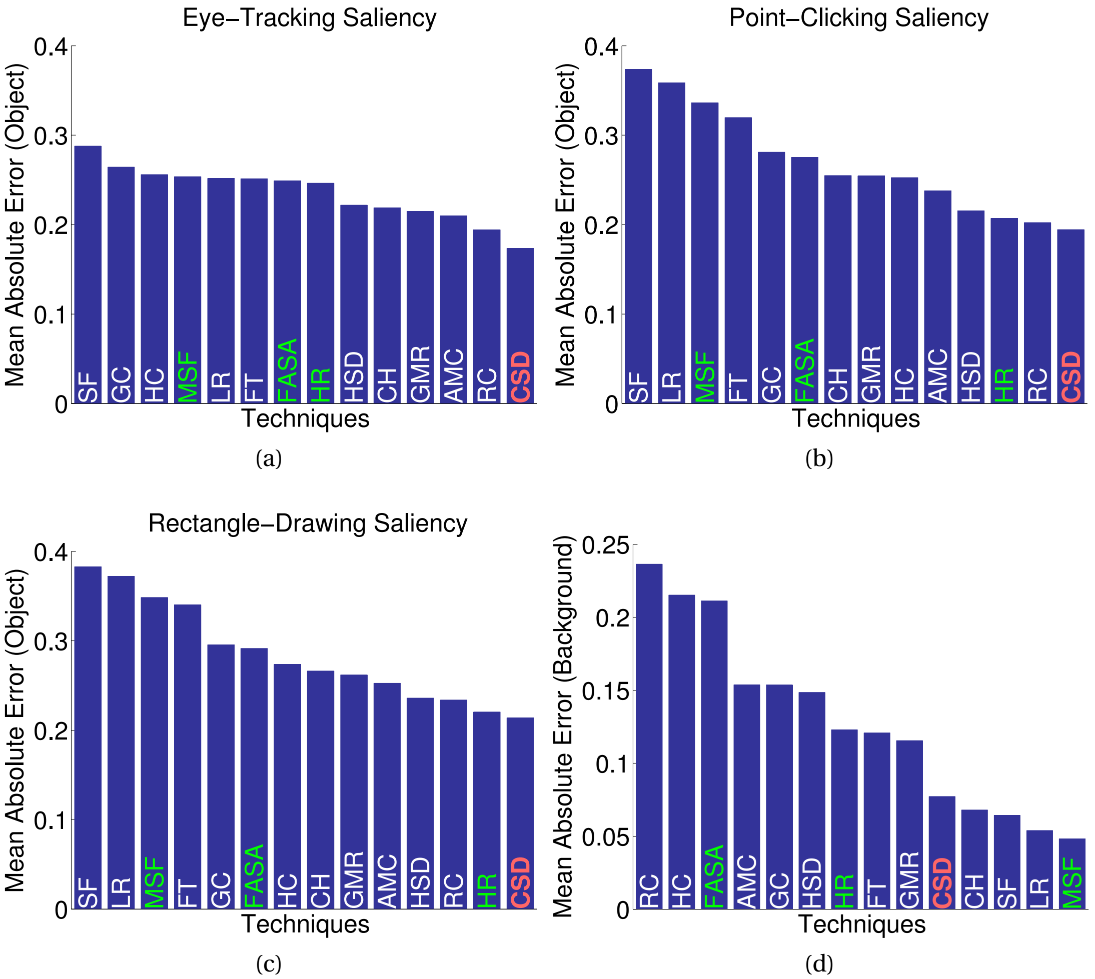
5.4.2 Salient-Object Ranking

Regardless of the absolute saliency values, a method can estimate the relative importance of objects by ranking them. We call this metric as "salient-object ranking", where we compute the Kendall rank-correlation coefficient (Kendall's τ) [88] between the real and the estimated rankings of the objects.

In Figure 5.11, we compare the Kendall's τ of all methods. The spatial-connectivity-based methods, such as GMR, AMC, and HSD, use only color as a visual feature. Although they perform better than the other methods (except CSD), color contrast is not sufficient for estimating multi-level saliency values of the objects in our dataset. Because, our COS dataset includes a variety of aspects of saliency, such as texture, orientation, size. Our CSD method employs multiple visual features and learn to detect object saliency. In addition, the Object-Awareness Model (OAM) provides mutli-level saliency values of objects. Therefore, similar to the results in mean absolute-error metric, our method significantly outperforms all of the state-of-the-art methods in all saliency types.

5.4.3 Discussion on Evaluation Metrics

When we compare the results in Figure 5.10 and 5.11, we can observe that the order of performance of the state-of-the-art methods is significantly different, excluding CSD. Although they are not independent, mean absolute error and salient-object ranking measure very different aspects of multi-level object saliency. In Table 5.2, two hypothetical cases for multi-level salient object detection is presented. In each case, there are two salient objects with different real and estimated saliency values. In the



(a)

(b)

(c)

(d)

Figure 5.10: (a)-(c) The mean absolute error (MAE) of all techniques computed between the estimated and the real multi-level object-saliency values. (d) The mean absolute error of all techniques in suppressing the background pixels.

first case, the mean absolute error metric measures a small error, which indicates a good performance. On the other hand, due to the change in the order of saliency values, the salient-object ranking metric evaluates an inverse correlation, i.e., a bad performance. In the second case, evaluations of the metrics reverse. This result shows that we should use a metric depending on how we apply the multi-level saliency values in an image processing or computer vision task.

Given two salient objects, absolute saliency values not only measure which one is more salient, but also quantify the difference in saliency. This approach can be useful in various applications, such as content-aware image compression in Section 6.3, because absolute saliency values are directly used as an indicator to compression ratio. When absolute measurements are not required, we can rank the salient objects with respect to their saliency.

Salient-object ranking can be used when the ordering of salient objects are more important than their absolute saliency values. For example, in Section 6.1 we demonstrate an image-tagging application, where we use the most salient object for labeling an image. Here, a change in the order of the salient objects would result inaccurate image tags.

Table 5.2: Two hypothetical cases, where evaluation metrics significantly differ.

	Case #1	Case #2
Real Saliency Value #1	0.48	0.3
Real Saliency Value #2	0.52	0.8
Estimated Saliency Value #1	0.51	0.0
Estimated Saliency Value #2	0.49	0.5
Mean Absolute Error	0.03	0.3
Salient-Object Ranking	-1	1

5.4.4 Visual Comparison

Visual comparisons of mutli-level ground truth maps and salient-object maps are given in Figure 5.12 and 5.13. Our CSD method generates uniform and accurate salient-object maps with multi-level saliency values and suppressed backgrounds.

5.5 Summary of the Chapter

In this chapter, we have proposed a salient-object detector that can estimate the multi-level saliency values of objects. We have introduced the Object-Awareness Model (OAM) that robustly converts binary saliency into multi-level by using the position and the size of an object. We have incorporated the OAM into our segmentation-based method (HR) in Chapter 4 and obtain a Comprehensive Salient Object Detection (CSD) method. We have showed that our CSD method outperforms the-state-of-the-art methods on the Comprehensive Object Saliency (COS) dataset in both mean absolute-error and salient-object ranking metrics.

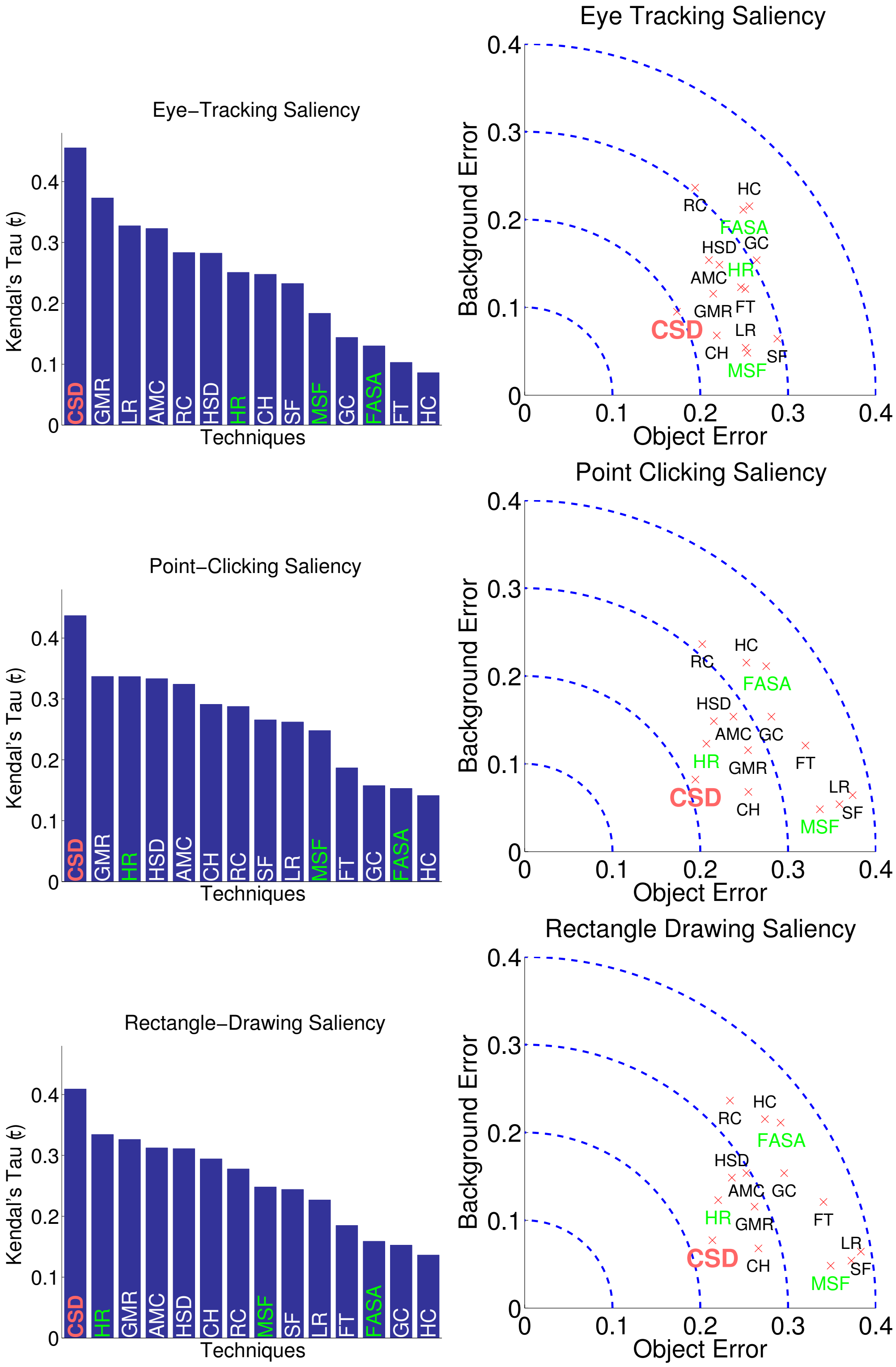
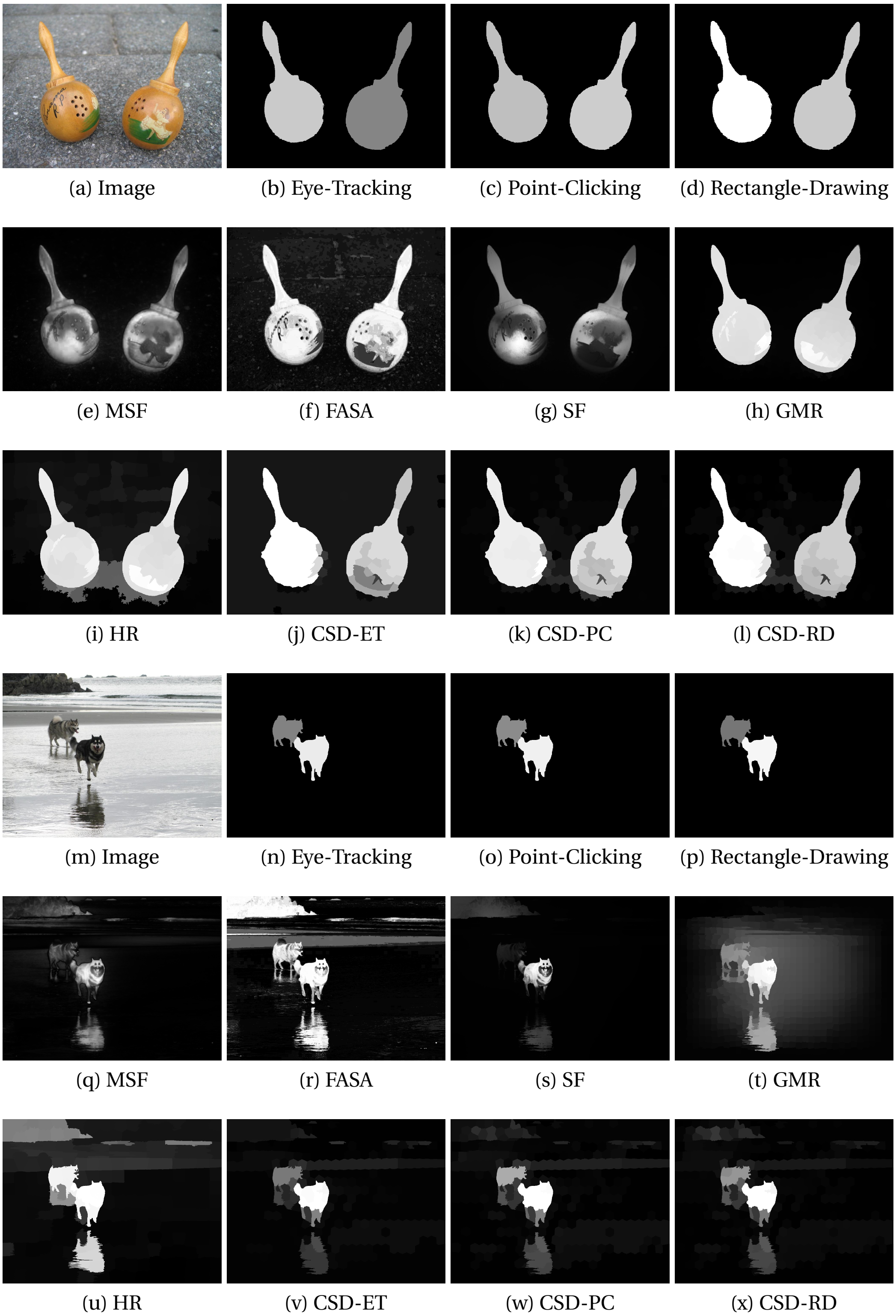


Figure 5.11: The rank correlation coefficients of all methods computed between the estimated and the real salient object rankings (on the left) and 2D representations of Mean Absolute Errors (on the right).



(a) Image (b) Eye-Tracking (c) Point-Clicking (d) Rectangle-Drawing

(e) MSF (f) FASA (g) SF (h) GMR

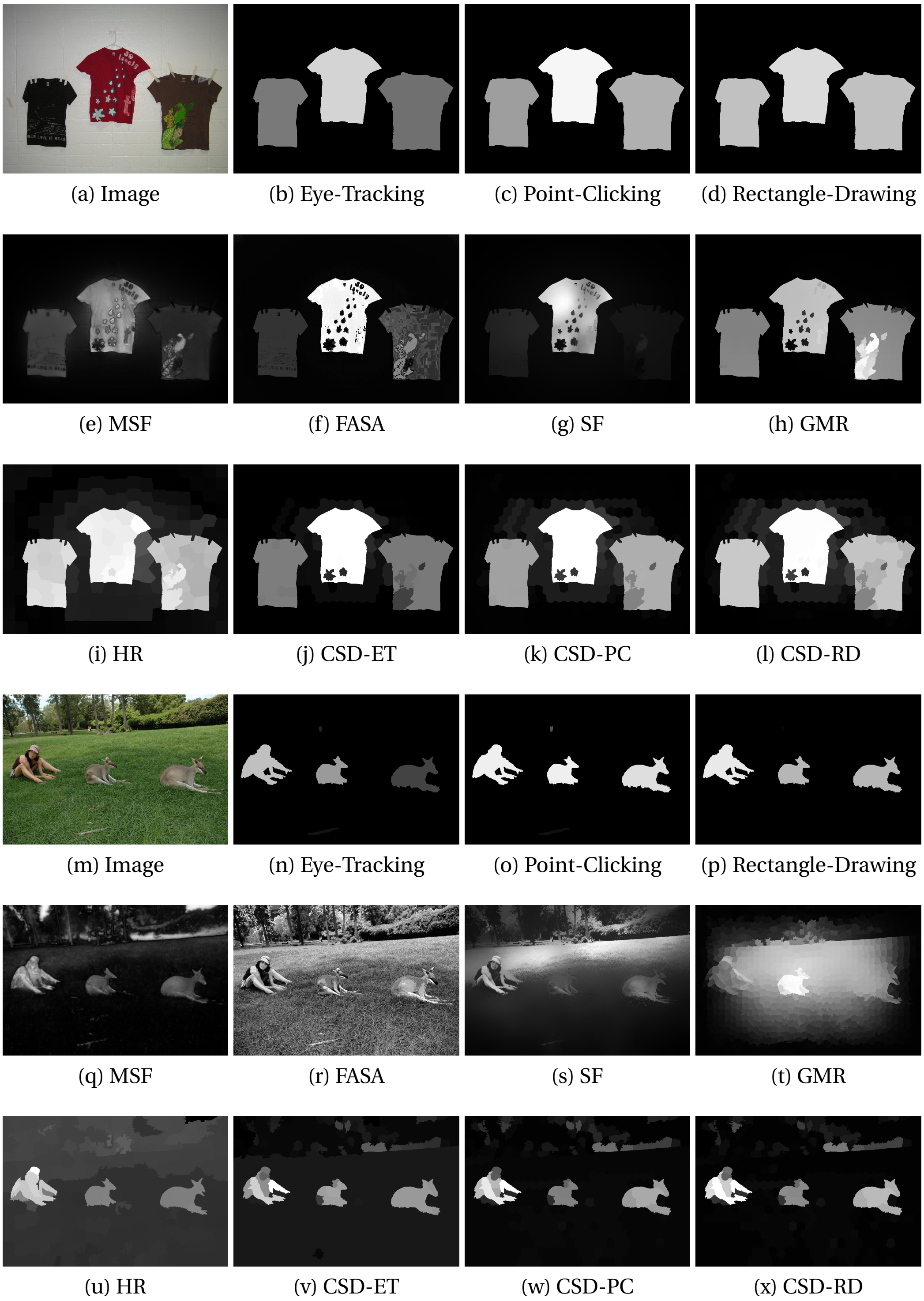
(i) HR (j) CSD-ET (k) CSD-PC (l) CSD-RD

(m) Image (n) Eye-Tracking (o) Point-Clicking (p) Rectangle-Drawing

(q) MSF (r) FASA (s) SF (t) GMR

(u) HR (v) CSD-ET (w) CSD-PC (x) CSD-RD

Figure 5.12: Visual comparison of salient object maps on two images. Our method CSD outputs three different salient-object maps, each of which correspond to a ground truth map type (ET: Eye-Tracking, PC: Point-Clicking, RD: Rectangle-Drawing


(a) Image (b) Eye-Tracking (c) Point-Clicking (d) Rectangle-Drawing

(e) MSF (f) FASA (g) SF (h) GMR

(i) HR (j) CSD-ET (k) CSD-PC (l) CSD-RD

(m) Image (n) Eye-Tracking (o) Point-Clicking (p) Rectangle-Drawing

(q) MSF (r) FASA (s) SF (t) GMR

(u) HR (v) CSD-ET (w) CSD-PC (x) CSD-RD

Figure 5.13: Visual comparison of salient object maps on two images. Our method CSD outputs three different salient-object maps, each of which correspond to a ground truth map type (ET: Eye-Tracking, PC: Point-Clicking, RD: Rectangle-Drawing

6 Applications of Salient-Object Detection

In this chapter, we demonstrate salient-object detection as a pre-processing step in three applications. In the first application, we present an image-tagging framework, where salient object detection assists an object-recognition algorithm in labeling images. We show that image-tagging accuracy is higher when we use a multi-level salient-object detector rather than a binary one. In the second application, we introduce a content-aware approach for compressing images by trading off the quality of salient and non-salient image-regions. In the third application, to propose image windows that are likely to contain an object, we use our bilateral-filtering method (see Section 4.3).

6.1 Image Tagging

Humans are very good at organizing and retrieving visual data, because they summarize the visual content into high-level concepts, such as objects and actions. Whereas for computers, there is a semantic gap between low-level image characteristics, such as color and texture, and the high-level understanding of images. In order to fill this gap, researchers have proposed object-recognition algorithms, which can convert image pixels into text-based information. Text-based knowledge is a good way to abstract concepts. For example, the word "apple" covers a wide range of image objects, regardless of their color, size, and pose, thus making text significantly easier for organizing and retrieving visual data.

Recent developments on deep-learning methods [89] provide comprehensive and accurate object-recognition algorithms. In order to classify a variety of objects, these algorithms are usually trained by using images with a single object-class. Therefore, given a complex image with multiple object types as input, the recognition results might not be satisfactory. Moreover, searching for an object using multi-scale sliding windows is very time consuming. Instead of exhaustively searching an image, or more accurate and rapid image tagging results, we can use a salient-object detector as a pre-processing step and help the object recognizer to focus on salient objects.

We propose a method that labels an image by using salient-object detection and evaluate this method on our dataset (see Chapter 3). As the images in our dataset were downloaded from ImageNet, we have the "real tag" for each image, which constitutes the semantic ground truth. Some of the objects in an image might not necessarily be related to the overall concept of that image. Therefore, we need to be aware of the relative importance or saliency of the objects. We can achieve this by using multi-level salient-object detectors instead of binary ones. We also compare the capabilities of binary and multi-level salient-object maps in image tagging. The procedure for this comparison is illustrated in Figure 6.1. First, we use our method in Chapter 5 and estimate the multi-level salient-object map of an input image. We then extract the most salient object and pass it through an object recognition method, which estimates the tag of the image. Finally, we compare the estimated tag to the real tag and obtain a word similarity score, i.e. tagging accuracy. We follow the same procedure for the binary salient-object map, except, as the binary map does not indicate which object is more salient, to tag the image, we choose a random single object. The main hypothesis of this comparison is that multi-level salient-object maps will help us generate more relevant image tags compared to the binary ones.

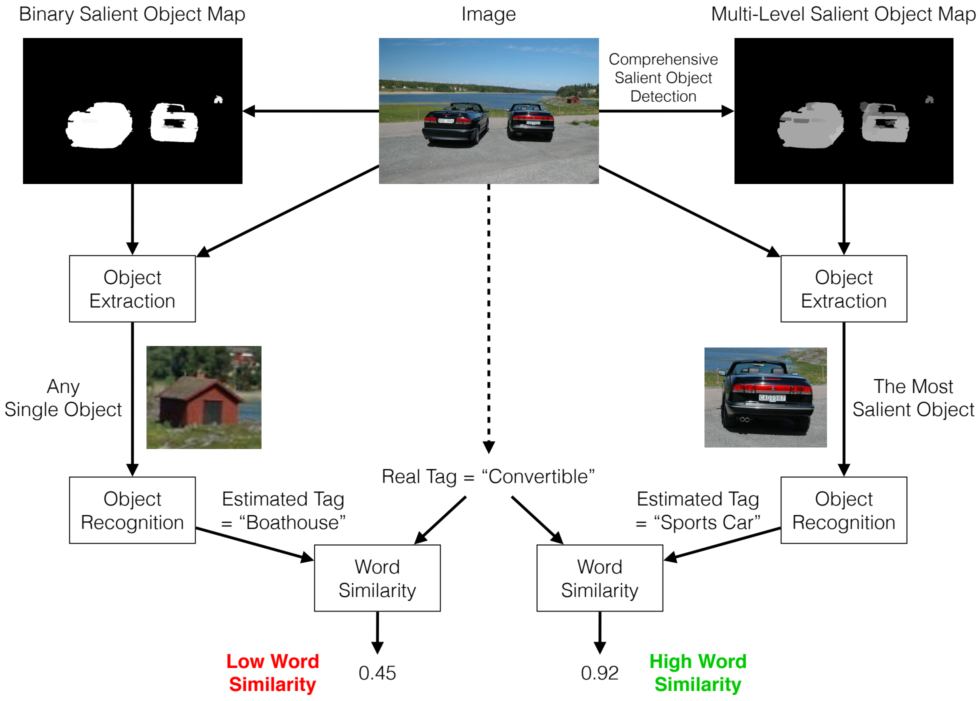


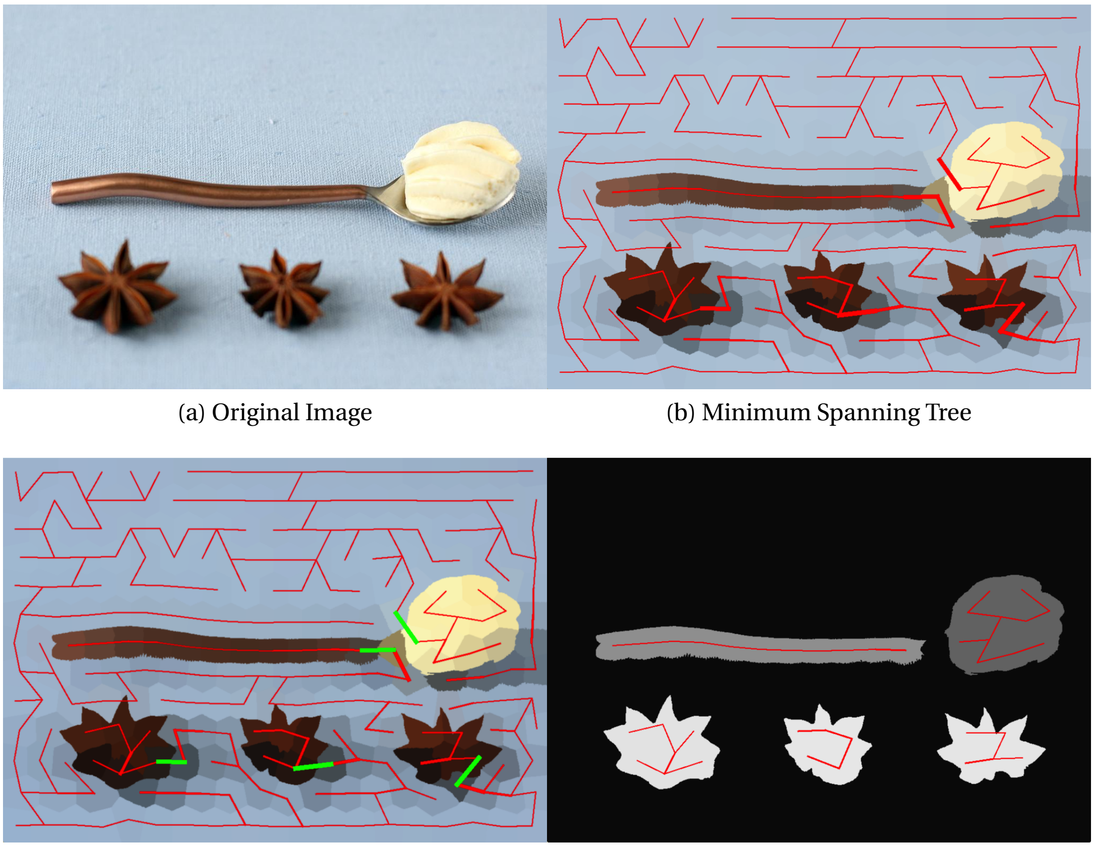
Figure 6.1: Image tagging using binary and multi-level salient object maps.

6.1.1 Object Extraction Using a Salient-Object Map

In Chapter 5, we mentioned that our multi-level salient-object detection algorithm forms a superpixel graph of an input image and calculates its Minimum Spanning Tree (MST), which is illustrated in Figure 6.2 (b). We automatically extract salient objects by using the MST and by following Algorithm 2. We iteratively divide the MST into several sub-trees (Figure 6.2 (c)) and extract 5 objects from an image. We evaluate the average saliency of an object by averaging the saliency values of superpixels that are connected by a sub-tree (Figure 6.2 (d)). We find the most salient object and pass it through the object recognizer for image tagging.

```
Initialize the Minimum Spanning-Tree → $\mathbf{T}$
for $o = 1, 2, ..., 5$ do
    Find the largest edge weight $w_{\max}$ in $\mathbf{T}$
    Set $w_{\max} = 0$, which divides $\mathbf{T}$ into two sub-trees $\mathbf{T}_{small}$ and $\mathbf{T}_{large}$
    Extract the $o^{th}$ object using $\mathbf{T}_{small}$ and compute its average saliency value $s^o$
    $\mathbf{T} = \mathbf{T}_{large}$
end
The most salient object = $\underset{o}{\operatorname{argmax}}(s^o)$
```

Algorithm 2: Algorithm for extracting N objects from an image and for selecting the most salient of them.


(a) Original Image

(b) Minimum Spanning Tree

(c) Object Extraction Edges

(d) Extracted Salient Objects

Figure 6.2: (b) The minimum spanning-tree (in red) of (a) an image is used for extracting objects (c) by removing edges with large weights (in green) (d) We extract the sub-tree that has the largest average saliency value as the most salient object.

6.1.2 Image Tagging Using Object Recognition

In order to estimate the tag of an image, we use Caffe [89], which is an object recognition framework based on convolutional neural networks. It has an object-recognition accuracy of 84.7% in top-5[1] [90] on the ImageNet Large-Scale Visual-Recognition Challenge-2012 (ILSVRC) test set [91] with 1000 classes. This framework re-scales an input image to a size of 256×256 pixels and classifies it in 70 milliseconds (ms) on average. For an image with a single object, the run time of this method is acceptable for practical applications. However, for more complex images with multiple objects, as we do not know the position and the size the objects beforehand, running sliding windows through an image of size 1024×768 for even three scales would take 4.6 hours for the whole image. A salient-object detector cuts the processing time down to order of seconds, as it finds the objects of interest and allocates the processing resources to the recognition of these objects.

6.1.3 Evaluating Word Similarity

Although the Caffe object-recognition framework is highly accurate, it is not always possible to exactly find the real tag of an image. Therefore, we need a way to measure the semantic similarity between two English phrases: the estimated tag and the real tag. For this purpose, we can use the WordNet [92], which is a tree of words in English language with a hierarchical semantic relationship. Note that the ImageNet uses the WordNet tree for image tags and the Caffe framework was trained using ImageNet images. Therefore, both the real tag and the estimated tag exist in the WordNet. The distance of two phrases within the WordNet tree is inversely related to their semantic similarity. In order to compute this distance, we use the Wu-Palmer similarity measure [93] in the Natural Language Processing Toolkit [94], which efficiently searches WordNet tree and computes a similarity measure between 0 and 1 as follows:

$$\text{Word Similarity} = \frac{2d_{msa}}{d_1 + d_2} \tag{6.1}$$

Here, d_1 and d_2 are the depths of the two phrases in WordNet tree, and d_{msa} is the depth of their "most specific ancestor", i.e. the deepest grand-parent node in the WordNet tree.

6.1.4 Performance on Image Tagging

In order to compare the performance of binary and multi-level salient-object maps, we extract one object per map, recognize the object, and evaluate the word similarity between estimated and real tags. We then compute the average word-similarity on our dataset and obtain the results in Table 6.1. Due to its capability for estimating the relative saliency of objects, mutli-level salient-object maps indeed provide image tags that are more relevant to the real tag compared to binary salient-object maps. Note that, as all objects are estimated to be equally important, in binary salient-object

[1] The real class of the object is within the five most likely estimates of the algorithm.

maps, we randomly select an object for image tagging. We repeat the randomized tagging-experiment 10 times and obtain its standard deviation. As show in Table 6.1, the improvement of multi-level salient-object maps is statistically significant.

Table 6.1: Average word similarity values that are evaluated on our dataset for salient object map types.

Salient Object Map Type	Average Word Similarity
Binary	0.602 ± 0.004
Multi-Level	**0.647**

6.2 Object Detection

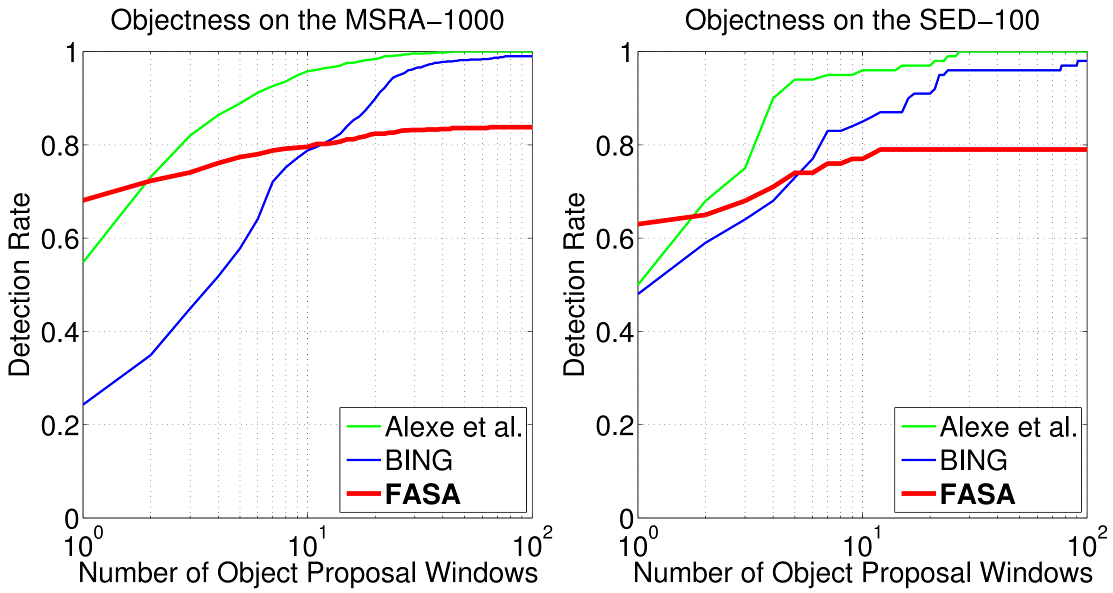
As we state in Chapter 1, salient-object detection is a pre-processing step for successive applications. Therefore, the speed of a salient-object detector can be as essential as its accuracy. Our bilateral-filtering approach (FASA) in Section 4.3 rapidly outputs a salient-object map and the position and the size of salient objects. Therefore, it can be used to propose image windows that might contain an object, regardless of its class. This operation is also called measuring the "objectness" of an image window.

We compare the object-detection capabilities of FASA to well-known objectness measuring methods, such as Alexe et al. [84] and BING [85]. The object-detection rate (probability of detecting an object) versus the number of object-proposal windows for the MSRA-1000 and the SED-100 datasets are illustrated in Figure 6.3 (a) and Figure 6.3 (b), respectively. Our method is more accurate than other methods in the first proposed window. This is logical as our method focuses on (and is optimized for) estimating the salient objects. This property can be useful in an application where a single and accurate window-proposal and an accurate salient-object map are more important than having only multiple window proposals, such as object segmentation.

Our method FASA is fast enough for real-time salient-object detection in videos. Furthermore, it provides the position and size of the salient and non-salient parts of the image. This property can be used in applications such as object tracking in videos [95, 96].

In order to demonstrate the potential of FASA, we estimate the salient-object maps of the publicly available video “Big Buck Bunny"[2]. We are able to process the high-definition (HD) version (1280 × 720) of the video with a speed of 30 frames per second (fps). The computer we used for the tests has an Intel Core i7 2.3GHz processor and 16 GB of RAM. Given that the frame-rate of the video is 24 fps, our method estimates the salient-object map and the center and size of the objects in real time. The fps values under different resolutions are given in Table 6.2. The fps value linearly changes with the number of pixels (N) in a single video frame. In other words, the number of pixels our method can process in a second ($N\times$ fps) is largely independent from the resolution of the image. This shows that our method has a computational complexity

[2] http://www.bigbuckbunny.org/index.php/download/



(a) (b)

Figure 6.3: The objectness detection rate of our method is compared to other methods in (a) and (b).

of $O(N) + O(K^2) \approx O(N)$, where K is the number of colors after quantization and $K^2 \ll N$. Note that our method is optimized to perform salient-object detection on still images and individually processes each video frame. Its performance can be increased if the correlation between consecutive frames is exploited.

Table 6.2: Average processing speed of FASA in frames per second (fps) for different resolutions of the video “Big Buck Bunny".

		Resolution		
		1920 × 1080	1280 × 720	854 × 480
fps	Frames per second	13.7	**30.7**	66.5
N	Number of megapixels	2.07	0.92	0.41
$N\times$ fps	Number of megapixels per second	28.4	28.3	27.2

In Figure 6.4, we illustrate salient-object maps of 10 frames from the same video. We enclose the most salient regions (saliency value > 0.75 after normalization between 0 and 1) with a red rectangle by using their estimated positions and sizes. Due to its global nature, our method is accurate in scenes with a single salient object or multiple salient objects with different colors. It has a limited performance when color interference or a complex scene is present, as shown in the last two frames in Figure 6.4.

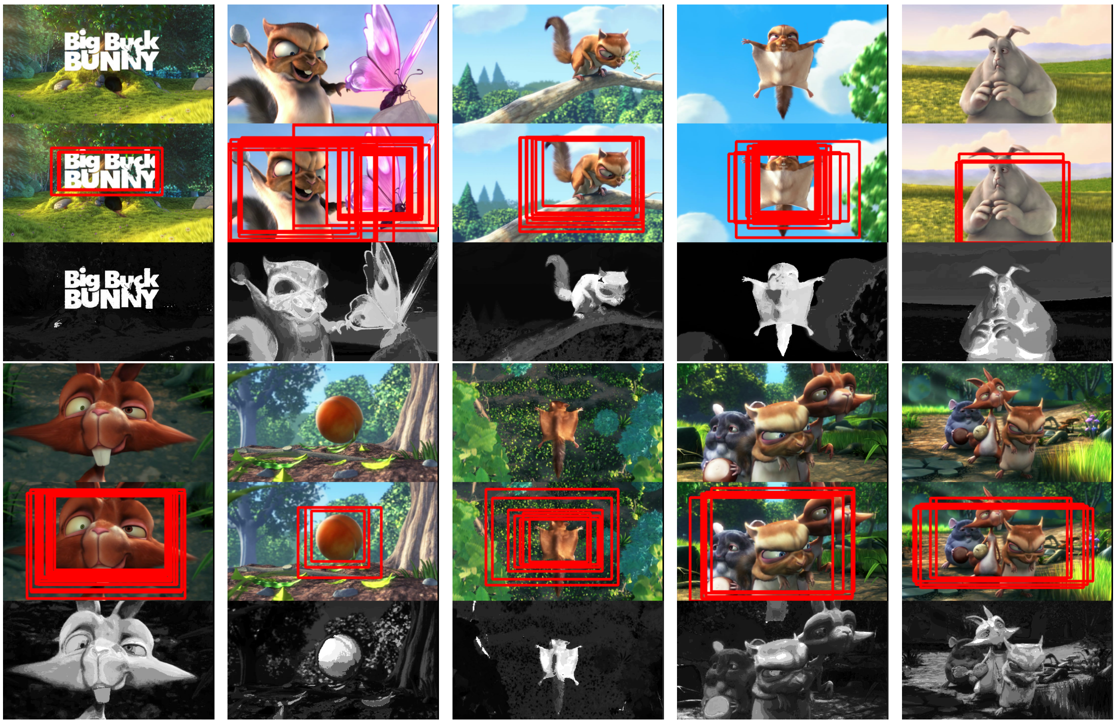


Figure 6.4: 10 frames from the video “Big Buck Bunny". The first row shows the original frame, the second row position and the size of the most salient objects, and the third row the saliency map of the frame.

6.3 Content-Aware Image Compression

Neighboring pixels in natural images are often similar in color. Instead of storing an image in a bitmap format as independent color pixels, we can exploit this redundancy and compress the image into a smaller size. One of the widely-used lossy image-compression methods is the JPEG standard [97], where, in order to save disk space, spectral coefficients of an input image are under-sampled and quantized. JPEG compression attempt to preserve the low-level image details, such as strong edges and textures, without being aware of the high-level content in the image. Here, we bring content-awareness to JPEG image-compression via salient-object detection, which is illustrated In Figure 6.5. In our content-aware compression, we first blur an input image with several Gaussian filters. We then use the estimated multi-level salient-object map and combine blurred images into a final image, where the more salient a region is, the less that region is blurred. This position-dependent blurring trades off the details in non-salient regions with high bit-rate at salient regions and helps us save the image with a medium JPEG quality. Compared to a standard JPEG image with the same file size, it is perceptually better. Preserving the details in the regions of interest is proven to increase the *perceived* image quality for JPEG images [98].

Some example images that are processed with our content-aware compression are compared to the standard JPEG image in Figure 6.6. Here, the background regions of

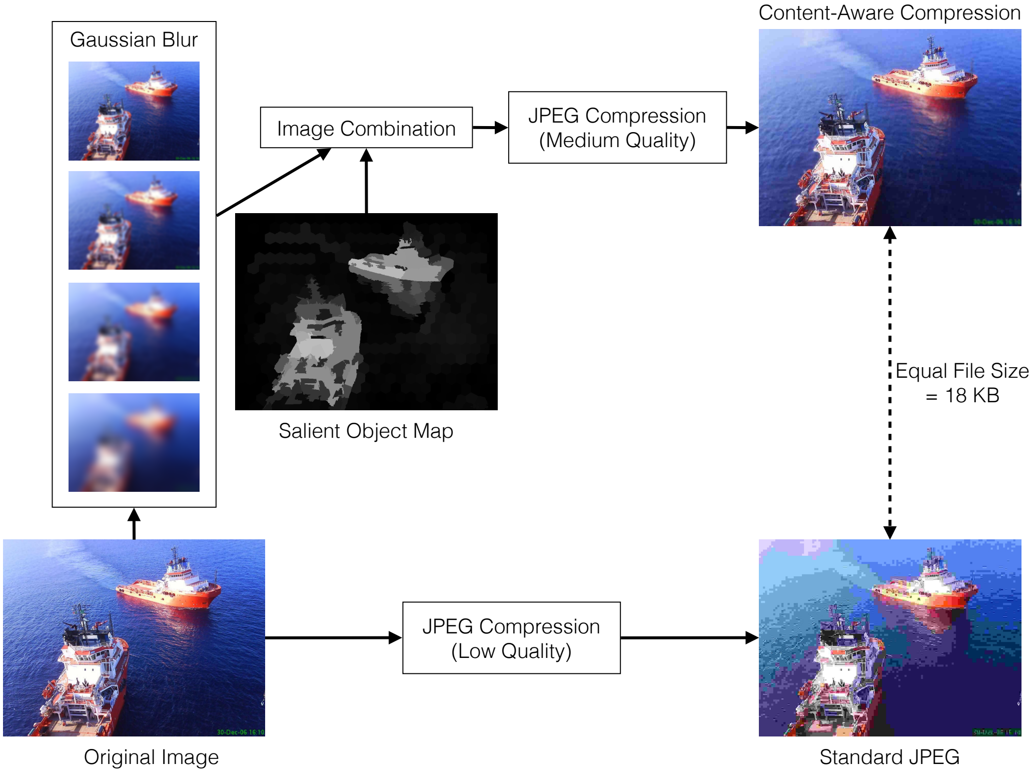


Figure 6.5: In order to preserve the details of the salient objects, content-aware compression trades-off the quality of non-salient regions by blurring the image with several parameters and combining them using a salient object map.

the images that are processed with our method (left) are more blurry than the output of a standard JPEG image (right). However, we benefit from that blurriness on the salient objects, as they are sharper in our results.

6.4 Summary of the Chapter

In this chapter, we have exhibited three applications that employ salient-object detection as a pre-processing step. In the first one, we used salient-object maps to extract an object that is relevant to the real tag of the image. We showed that multi-level salient object maps provide more relevant keywords for image labeling than binary maps. In the second application, we introduced a content-aware image compression technique, which trades off the background and the foreground quality via Gaussian blurring. Our compression method produces visually better results compared to the standard JPEG compression when the file size is fixed. In our final application, we have demonstrated the object-detection potential of our bilateral-filtering approach (see Section 4.3). The first object-proposal window of our method, along with its capability of detection an accurate salient -object map, makes it a good pre-processing step for an object-segmentation algorithm.

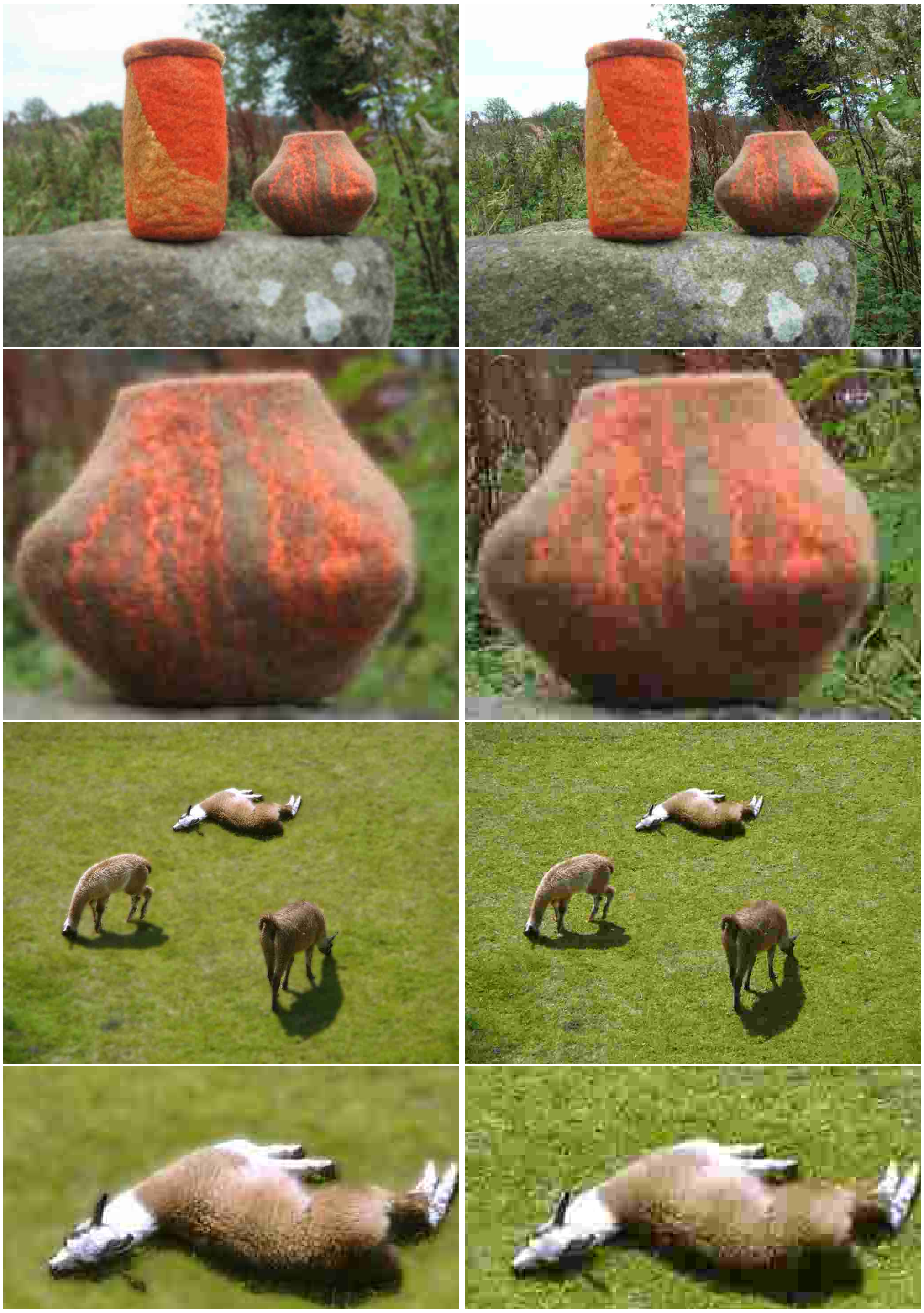


Figure 6.6: Our content-aware compression (left) is compared to the standard JPEG compression (right). In order to preserve the details on the salient objects, we sacrifice background quality.

7 Conclusion

Salient-object detection is the task of finding objects of interest in images or videos. As a pre-processing step to computer vision applications, it enables computers to allocate their processing resources to important regions of visual data. Researchers have hitherto approached salient-object detection as a foreground-background segmentation problem either by assuming a single object per scene or by overlooking the subjectivity of the visual saliency of objects.

The main objective of our thesis was to challenge the conventional methodology of salient-object detection and introduce multi-level object saliency. We completed this objective by accomplishing four goals: First, we provided our Comprehensive Object Saliency (COS) dataset that represents the natural images better than the well-known salient-object dataset. We then conducted three subjective experiments and measured the effect of fixation duration and collective attention on multi-level object saliency. Second, we proposed three salient-object detectors that successively solve more challenging problems of salient-object detection. Our segmentation- and machine-learning-based method performs the best compare to the state-of-the-art techniques. Third, we introduced an Object-Awareness Model (OAM) that maps the object position and size to multi-level saliency values. We incorporated this model into our segmentation-based method and obtain the Comprehensive Salient Object Detection (CSD) method that is capable of multi-level salient-object detection. Our technique significantly outperforms the state-of-the-art methods on the COS dataset. Finally, we used our salient-object detectors as a pre-processing step to three separate image processing and computer vision tasks. We showed that multi-level salient-object detection is preferable over conventional methodology of salient-object detection.

7.1 Summary of the Thesis

We reviewed the previous work that is related to object saliency under two major topics. In Chapter 2.1, we inspected the well-known evaluation datasets for salient-object detectors. We conclude that the images in these datasets do not comprehensively represent natural images, as they mostly consist of images with a single object. Moreover, the subjectivity of visual saliency was not taken into consideration, which is evident from the all-or-nothing approach of the binary ground-truth maps. In Chapter 2.2, we analyzed the characteristics of the state-of-the-art salient-object detectors. These detectors employed three major mechanisms for identifying objects of interest: uniqueness, spatial variance, and spatial connectivity. We showed that spatial-connectivity-based methods perform the best among all methods, because they work with content-aware image segments and estimate more uniform salient-object maps. However, due to the binary object-saliency assumption, we conclude that the state-of-the-art methods cannot measure the relative importance of objects.

The first goal of our thesis was to represent natural images better than the well-known datasets and to show that object saliency is multi-level, i.e., all objects are not equally important. In Chapter 2.1, we mentioned that the well-known datasets do not measure the relative (or subjective) saliency of objects. Therefore, in Chapter 3, we introduced our dataset, which is more comprehensive and representative than its predecessors. Our dataset consists of 588 natural images with multiple objects (2434 objects in total). We retained the subjectivity of visual saliency by carrying out three subjective experiments on each image. Eye-tracking experiments represent the effect of fixation duration on multi-level object saliency. Whereas, point-clicking and rectangle-drawing experiments are related to the collective human attention. We used the experimental data to *measure* the saliency value of objects and we showed that object saliency is not binary, i.e. it is multi-level.

The second goal of our thesis is to *find* multiple salient-objects with unknown positions and sizes. We used the state-of-the-art methods in Chapter 2.2 as a guideline for proposing three novel salient-object detectors in Chapter 4, each of which address a challenging problem. Our first method is based on linear, multi-scale filtering and solves the problem of choosing an optimal local-global contrast adjusting parameter. The second method employs bilateral filtering and provides more uniform salient-object maps compared to linear filtering. In addition, it uses spatial-variance mechanism to estimate the position and the size of salient objects. Our third method adopts hierarchical image segments as building blocks for salient-object detection.The spatial-connectivity mechanism on multiple hierarchical-layers provides a method that is capable of producing accurate salient object maps that are uniform, robust against texture, and capable of finding multiple objects.

The third goal of our thesis was to extend the binary salient-object detection to multi-level by *estimating* the varying importance of objects. In order to achieve this, we used our third method in Chapter 4 as a baseline and improved it with an object-awareness step in Chapter 5. We define object awareness as the ability to estimate the position and the size of the salient objects in an image. We trained an object-awareness model (OAM) for each multi-level ground-truth type in our dataset (Chapter 3) and proposed a method that can estimate three separate multi-level salient -object maps. We showed that our extended algorithm performs significantly better than our previous methods (Chapter 4) and the state-of-the-art methods (Chapter 2.2) in multi-level salient-object detection.

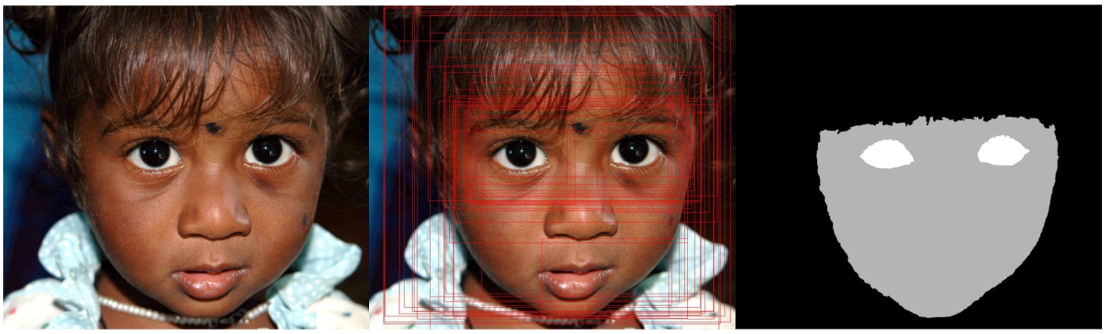
The fourth goal of our thesis was to demonstrate the practical aspects of salient-object detection. In Chapter 6 we accomplished it in three separate applications. In the first one, we used our extended method (Chapter 5) to tag images via object recognition. We showed that estimating the relative importance of objects is more beneficial over assuming all objects to be equally salient by comparing their-image tagging accuracy. In the second application, we introduced a technique for content-aware image compression. We empirically showed that using content-aware image compression via multi-level salient-object detection yields visually better results compared to using the standard JPEG format, especially on low bit-rates. Our final application was to use our bilateral-filtering approach in Chapter 4 for object detection. In addition to estimating an accurate salient-object map, our method performs comparably to the state-of-the-art objectness measures.

7.2 Final Remarks & Future Work

We call the varying importance of objects as multi-level object saliency. It could be argued that, as the saliency value of an object can take any value between 0 and 1 and as this interval is not quantized, we can name the ground truth maps in our dataset as "continuous" rather than multi-level. However, a continuous map would be a misnomer, because in our maps, all pixels of an object have the same saliency value, i.e. multi-level ground-truth maps are not continuous tone images. We could approach a continuous map by conducting subjective experiments not only on individual objects, but also the components and subtle details of them. We performed one such study on human faces [99], where we measured the importance of faces and their components, such as eyes, nose, and mouth as illustrated in Figure 7.1. In this example, we asked subjects to draw rectangles around the objects that they think are important. The rectangles and the corresponding multi-level ground-truth map are shown in Figure 7.1 (b) and 7.1 (c), respectively. Although the face is deemed important by the subjects, there is even a larger interest in the eyes. As face perception is a huge part of social interaction [100], sub-object saliency on human faces could have practical applications. Further research can be done to extend sub-object saliency to generic objects and to achieve real "continuous" salient object maps.

Image over-segmentation is widely used as a pre-processing step to salient-object detection [17–20, 101]. Because, over-segmentation reduces the amount of information from millions of pixels to hundreds of superpixels and leads to the best performance under the context of spatial connectivity, as we showed in Chapter 2.3. After the salient-object detection step, there are several methods [57, 102] that can segment out objects as a whole by using the estimated maps. We can easily see that over-segmentation, salient-object detection, and object segmentation are closely related steps from pixel- to object-level abstraction. Although separately studied, there is still much room for investigating the relationship between these topics, such as "segment importance", where all segments are not equally treated.

As in many computer algorithms, speed-accuracy trade-offs also exist for salient-object detection. Bilateral-filtering approach in Chapter 4.3 estimates a salient-object



(a) Original Image (b) Rectangle Drawing (c) Multi-Level Ground Truth Map

Figure 7.1: (a) In some close-up images, (b) subjects might mark parts of the objects as salient, rather than the object as a whole. As a result, (c) the multi-level ground truth map is represented at sub-object level.

map in the order of milliseconds, making a real-time application possible. However, it does not perform well on complex images. On the other hand, our segmentation-based approach in Chapter 4.4 produces very accurate salient-object maps by sacrificing execution time via hierarchical segmentation, feature extraction, and belief propagation. Although both algorithms can be optimized, a better strategy is to use them in different applications. For example, the bilateral-filtering approach can be used to supply quick salient object proposals to a video-tracking algorithm using particle filters [103]. Whereas, the segmentation-based approach can be useful for significantly reducing the multi-scale sliding window search time of an object recognizer in an image with multiple objects, as we showed in Chapter 6.1.

In addition to computer-vision applications, visual saliency and salient-object detection can be discussed in the context of visual design [104]. There are studies on the saliency in web-page design [105] and the color selection in video games [106]. A promising research topic can be pursued in digital humanities, where salient-object detection can be applied to various visual design and art forms.

Finally, we would like to the attract reader's attention to the machine-learning based salient-object detection. As we showed in Chapter 4.5 and 5.4 our machine-learning-based salient-object detectors outperform the state-of-the-art methods. In recent years, due to the practical implementations of deep-learning architectures, the accuracy of object-recognition algorithms have significantly increased. We can apply the same idea to salient-object detection. One important thing to notice is that, the increased accuracy of object recognition comes from using a huge amount of supervised visual data, because image tagging does not often require any expertise and can be easily performed. However, for salient-object detection, it is very time consuming to generate ground-truth maps with pixel-level precision. Therefore, a very promising direction would be to train an unsupervised deep-learning architecture and modify its output to a salient-object detector.

Glossary

α Half of the human fovea size in degrees. 47, 123

β^2 A scalar value that balances the effect of precision and recall values on F-Measure. 36, 123

$\mathbf{c}$ Three dimensional vector that represents the color of a pixel or the average color of a superpixel. 27, 30, 54, 73–76, 78, 82, 83, 96, 123

$\mathbf{D}$ The data that is obtained via one of the subjective experiments: eye-tracking, point-clicking, or rectangle-drawing. xviii, 44, 47–50, 123

δ Refers to a single subject in a subjective experiment. xviii, 47–49, 123

ΔE^* Euclidean distance between two CIELa*b* color vectors. xvii, 24, 25, 123

η Accuracy of the eye-tracking equipment in degrees. 47, 123

f_n False negative: number of salient pixels that are estimated as non-salient. 35, 123

f_p False positive: number of non-salient pixels that are estimated as salient. 35, 83, 123

F_G Two-dimensional Gaussian Filter. 65–68, 123

γ A binary value that indicates whether corresponding image segment is considered salient or not. 98, 99, 123

$\mathbf{G}$ A ground truth map. 38, 98, 99, 123

$\mathbf{h}$ CIELa*b* color histogram of a superpixel. 56, 82, 83, 123

h A scalar value that represents the number of elements in a histogram bin. 25, 26, 123

$\mathbf{I}$ A matrix with either one channel (grayscale) or three channels (color), which represents a digital image. 24, 25, 66, 68, 96, 123

κ Histograms of oriented gradients of a superpixel. 82, 83, 123

K A scalar value that indicates the spatial compactness of a salient object map. 68, 70, 123

Λ The probability of saliency of a pixel, quantized color, or superpixel. 78, 79, 123

l A single layer at a hierarchical segmentation. 82, 83, 86, 123

Γ A binary masking image, which usually indicate the pixels of an object, a surrounding area, or background. xviii, 46, 48–50, 52, 54, 123

$\mathbf{M}$ A multi-level ground truth map. 52, 123

Φ A superpixel, which is a set similarly-colored image pixels with proximity. 82, 83, 86, 96–99, 123

$\mathbf{p}$ Two dimensional vector that represents the position of a pixel or the center of mass of a superpixel. 26, 27, 73–75, 78, 79, 86, 123

Θ A full-resolution map that applies and adaptive center prior to the salient object map. 68, 69, 123

$\mathbf{q}$ Three dimensional vector that represents a quantized color. 25, 26, 74–76, 78, 123

$\mathbf{R}^2$ Coefficient of determination between two sets of values. 53, 91, 92, 123

$\mathbf{S}$ A salient object map. 24–26, 38, 68, 101, 123

σ_c A scalar value that controls the effect of color contrast. 27, 30, 74, 123

σ_s A scalar value that controls the area of effect of a Gaussian in spatial coordinates. xvii, 26, 27, 65, 66, 123

$\mathbf{T}$ Minimum Spanning Tree of a graph. 111, 123

s Saliency value of a pixel or a superpixel. xviii, xix, 44, 47–53, 78, 79, 86, 111, 123

t_p True positive: number of salient pixels that are estimated as salient. 35, 83, 123

V A scalar value that represents the spatial variance of the color of a pixel or the average color of a superpixel. 27, 123

w A scalar value that represents the weight of an edge between two nodes of a graph. xxi, 30, 96, 97, 111, 123

x Horizontal coordinate of a matrix or an image. 24, 25, 38, 73–76, 123

ξ Global color-contrast value of a pixel, quantized color, or superpixel. 78, 79, 123

y Vertical coordinate of a matrix or an image. 24, 25, 38, 73, 123

$\mathbf{Z}$ A weak salient object map that is later combined with other maps to obtain a final and more accurate salient object map. 67, 68, 123

Acronyms

AMC Absorbing Markov Chain [17]. 24, 36, 38, 40, 123

BSD Berkeley Segmentation Dataset [38]. 19, 123

CH Contextual Hyperhraph [20]. 24, 36, 38, 40, 123

COS Comprehensive Object Saliency. vii, viii, xv, xxiii, 3, 9, 10, 12, 20, 41, 43, 52, 91, 95, 99, 101–103, 105, 119, 123

CSD Comprehensive Salient Object Detection. viii, xv, xxi, 3, 11–14, 52, 87, 91–96, 101, 103, 105, 119, 123

FASA Fast, Accurate, and Size-Aware [107]. 123

FT Frequency-Tuned [10]. 24, 36, 40, 123

GC Global Cues [16]. 24, 36, 40, 123

GMR Graph-based Manifold Ranking [18]. 24, 36, 38, 40, 123

HC Histogram-based Contrast [13]. 24, 36, 40, 123

HR Hierarchical Regression [101]. 123

HSD Hierarchical Saliency Detection [19]. 24, 36, 38, 40, 123

HVS Human Visual System. 2–4, 7, 8, 24, 123

LR Low-Rank matrix recovery [14]. 24, 36, 38, 40, 123

MSF Multi-Scale Filtering. 123

MSRA Microsoft Research Asia Dataset [37]. 18, 123

MSRA-1000 MSRA-1000 Dataset [10]. 18, 123

MST Minimum Spanning Tree. 87, 94, 96, 123

OAM Object-Awareness Model. vii, viii, xxi, 3, 11–13, 27, 91–96, 99–101, 103, 105, 119, 123

RC Region-based Contrast [13]. 24, 36, 40, 123

SED-100 Segmentation Evaluation Dataset [11]. 18, 123

SF Saliency Filters [15]. 24, 36, 40, 123

SOD Salient Object Dataset [12]. 19, 123

SVR Support Vector Regression. 91, 123

Bibliography

[1] M. A. Plaisier, W. M. B. Tiest, and A. M.L. Kappers, "Haptic pop-out in a hand sweep," *Acta Psychologica*, vol. 128, no. 2, pp. 368 – 377, 2008.

[2] L. Lu and H. Zhang, "Automated extraction of music snippets," in *Proceedings of ACM International Conference on Multimedia*, 2003, pp. 140–147.

[3] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on PAMI*, vol. 20, no. 11, pp. 1254–1259, 1998.

[4] M. D. Fairchild, *Color Appearance Models*, 2005.

[5] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of Neural Science*, Third edition, 1991.

[6] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[7] F. Crick, "Function of the thalamic reticular complex: The searchlight hypothesis," in *Neurocomputing: Foundations of Research*, J. A. Anderson and E. Rosenfeld, Eds., pp. 569–575. 1988.

[8] E. Weichselgartner and G. Sperling, "Dynamics of automatic and controlled visual attention," *Science*, vol. 238, pp. 778–780, 1987.

[9] J. K. Tsotsos, "Is complexity theory appropriate for analyzing biological systems?," *Behavioral and Brain Sciences*, vol. 14, pp. 770–773, 1991.

[10] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *Proceedings of IEEE CVPR*, 2009, pp. 1597 – 1604.

[11] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration," in *Proceedings of IEEE CVPR*, 2007, pp. 1–8.

[12] V. Movahedi and J. H. Elder, "Design and perceptual validation of performance measures for salient object segmentation," in *Proceedings of IEEE CVPR Workshops*, 2010, pp. 49–56.

[13] M. Cheng, G. Zhang, N. J. Mitra, X. Huang, and S. Hu, "Global contrast based salient region detection," in *Proceedings of IEEE CVPR*, 2011, pp. 409–416.

[14] X. Shen and Y. Wu, "A unified approach to salient object detection via low rank matrix recovery," in *Proceedings of IEEE CVPR*, 2012, pp. 853–860.

[15] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proceedings of IEEE CVPR*, 2012, pp. 733–740.

[16] M. Cheng, J. Warrell, W. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Proceedings of IEEE ICCV*, 2013.

[17] B. Jiang, L. Zhang, H. Lu, M. Yang, and C. Yang, "Saliency detection via absorbing markov chain," *Proceedings of IEEE ICCV*, 2013.

[18] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of IEEE CVPR*, 2013, pp. 3166–3173.

[19] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of IEEE CVPR*, 2013, pp. 1155–1162.

[20] X. Li, Y. Li, C. Shen, A. Dick, and A. van den Hengel, "Contextual hypergraph modelling for salient object detection," *Proceedings of IEEE ICCV*, 2013.

[21] E. Niebur and C. Koch, "Computational architectures for attention," in *The Attentive Brain*, R. Parasuraman, Ed., chapter 9, pp. 163–186. MIT Press, Cambridge, MA, 1998.

[22] C W Eriksen and J D St James, "Visual attention within and around the field of focal attention: a zoom lens model," *Perception & Psychophysics*, vol. 40, no. 4, pp. 225–240, 1986.

[23] J. M. Henderson and A. Hollingworth, "Eye movements during scene viewing: An overview," in *Eye Guidance while Reading and While Watching Dynamic Scenes*, G. Underwood, Ed., pp. 269–293. Elsevier, 1998.

[24] A. L. Yarbus, *Eye Movements and Vision*, 1967.

[25] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Proceedings of IEEE ICCV*, 2009, pp. 2106–2113.

[26] G. Kootstra, A. Nederveen, and B. de Boer, "Paying attention to symmetry," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 1–10.

[27] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Advances in Neural Information Processing Systems*, pp. 241–248. 2008.

[28] J. Li, M.D. Levine, X. An, X. Xu, and H. He, “Visual saliency based on scale-space analysis in the frequency domain,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, 2013.

[29] J. M. Wolfe, “Guided search 2.0 a revised model of visual search,” *Psychonomic Bulletin & Review*, vol. 1, no. 2, pp. 202–238, 1994.

[30] Q. V. Le, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, “Building high-level features using large scale unsupervised learning,” in *In International Conference on Machine Learning*, 2012.

[31] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the International Conference on Machine Learning*, 2009, pp. 609–616.

[32] P. J. W. Noble, “Self-scanned silicon image detector arrays,” *IEEE Transactions on Electron Devices*, vol. 15, no. 4, pp. 202–209, 1968.

[33] K. Jain, C.G. Willson, and B.J. Lin, “Ultrafast deep UV lithography with excimer lasers,” *IEEE Electron Device Letters*, vol. 3, no. 3, pp. 53–55, 1982.

[34] E. R. Fossum, “CMOS image sensors: electronic camera on a chip,” in *International Electron Devices Meeting*, 1995, pp. 17–25.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *Proceedings of IEEE CVPR*, 2009.

[36] G. E. Moore, “Cramming more components onto integrated circuits,” *Electronics*, vol. 38, no. 8, 1965.

[37] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Shum, “Learning to detect a salient object,” *IEEE Transactions on PAMI*, vol. 33, no. 2, pp. 353–367, 2011.

[38] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *in Proceedisng of IEEE ICCV*, 2001, vol. 2, pp. 416–423 vol.2.

[39] S. Ramanathan, H. Katti, N. Sebe, M. Kankanhalli, and T. Chua, “An eye fixation database for saliency detection in images,” in *Proceedings of ECCV*, 2010, pp. 30–43.

[40] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1304 –1318, 2004.

[41] A. Borji, "Boosting bottom-up and top-down visual features for saliency estimation," in *Proceedings of IEEE CVPR*, 2012, pp. 438–445.

[42] B. Schauerte and R. Stiefelhagen, "Quaternion-based spectral saliency detection for eye fixation prediction," in *Proceedings of ECCV*, 2012, pp. 116–129.

[43] X. Sun, H. Yao, and R. Ji, "What are we looking for: Towards statistical modeling of saccadic eye movements and visual saliency," in *Proceedings of IEEE CVPR*, 2012, pp. 1552–1559.

[44] U. Engelke, Hantao Liu, Junle Wang, P. Le Callet, I. Heynderickx, H. Zepernick, and A. Maeder, "Comparative study of fixation density maps," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1121–1133, 2013.

[45] J. S. Babcock, J. B. Pelz, and M. D. Fairchild, "Eye tracking observers during rank order, paired comparison, and graphical rating tasks," in *in Proceedings of the PICS Digital Photography Conference*, 2003.

[46] A. Toet, "Computational versus psychophysical bottom-up image saliency: A comparative evaluation study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2131–2146, 2011.

[47] N. D. B. Bruce and M. E. Jernigan, "Evolutionary design of context-free attentional operators," in *in Proceedings of IEEE ICIP*, 2003, vol. 1, pp. I–429–32.

[48] N. Ouerhani, R. von 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, no. 1, pp. 13–24, 2003.

[49] P. Le Callet and E. Niebur, "Visual attention and applications in multimedia technologies," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2058–2067, 2013.

[50] Y. Xiang and M. S. Kankanhalli, "Video retargeting for aesthetic enhancement," in *Proceedings of ACM Multimedia*, 2010, pp. 919–922.

[51] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," in *Proceedings of IEEE ICCV*, 2007, pp. 1–6.

[52] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proceedings of IEEE CVPR*, 2010, pp. 2376–2383.

[53] R. Achanta and S. Süsstrunk, "Saliency detection for content-aware image resizing," in *Proceedings of IEEE ICIP*, 2009, pp. 1001–1004.

[54] S. Lee, J. Shin, and M. Lee, "Non-uniform image compression using biologically motivated saliency map model," in *in Proceedings of IEEE Intelligent Sensors, Sensor Networks and Information Processing Conference*, 2004, pp. 525–530.

[55] F. Zund, Y. Pritch, A. Sorkine-Hornung, S. Mangold, and T. Gross, “Content-aware compression using saliency-driven image retargeting,” in *in Proceedings of IEEE ICIP*, 2013, pp. 1845–1849.

[56] P. Jiang, H. Ling, J. Yu, and J. Peng, “Salient region detection by ufo: Uniqueness, focusness and objectness,” ICCV 2013.

[57] S. Kang, H. Lee, J. Kim, and J. Kim, “Automatic image segmentation using saliency detection and superpixel graph cuts,” in *Robot Intelligence Technology and Applications*, vol. 208, pp. 1023–1034. 2013.

[58] R. Sznitman, C. Becker, F. Fleuret, and P. Fua, “Fast object detection with entropy-driven evaluation,” in *Proceedings of IEEE CVPR*, 2013, pp. 3270–3277.

[59] K. Shi, K. Wang, J. Lu, and L. Lin, “PISA: Pixelwise image saliency by aggregating complementary appearance contrast measures with spatial priors,” in *in Proceedings IEEE CVPR*, 2013, pp. 2115–2122.

[60] X. Li, H. Lu, L. Zhang, X. Ruan, and M. Yang, “Saliency detection via dense and sparse reconstruction,” *ICCV 2013*, 2013.

[61] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, “Salient object detection: A discriminative regional feature integration approach,” in *in Proceedings of IEEE CVPR*, 2013, pp. 2083–2090.

[62] Z. Jiang and L.S. Davis, “Submodular salient region detection,” in *in Proceedings IEEE CVPR*, 2013, pp. 2043–2050.

[63] Y. Wei, F. Wen, W. Zhu, and J. Sun, “Geodesic saliency using background priors,” in *Proceedings of ECCV*, vol. 7574, pp. 29–42. 2012.

[64] S. Lu and J. Lim, “Saliency modeling from image histograms,” in *Proceedings of ECCV*, 2012, pp. 321–332.

[65] L. Wang, J. Xue, N. Zheng, and G. Hua, “Automatic salient object extraction with contextual cue,” in *Proceedings of IEEE ICCV*, 2011, pp. 105–112.

[66] H. Jiang, J. Wang, Z. Yuan, T. Liu, and N. Zheng, “Automatic salient object segmentation based on context and shape prior,” *Proceedings of BMVC*, pp. 1–12, 2011.

[67] K. Chang, T. Liu, H. Chen, and S. Lai, “Fusing generic objectness and visual saliency for salient object detection,” in *Proceedings of IEEE ICCV*, 2011, pp. 914–921.

[68] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, “Salient object detection by composition,” in *Proceedings of IEEE ICCV*, 2011, pp. 1028–1035.

[69] Y. Lu, W. Zhang, H. Lu, and X. Xue, "Salient object detection using concavity context," in *Proceedings of IEEE ICCV*, 2011, pp. 233–240.

[70] R. Achanta and S. Susstrunk, "Saliency detection using maximum symmetric surround," in *IEEE International Conference on Image Processing*, 2010, pp. 2653–2656.

[71] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[72] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434–444, 2010.

[73] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, pp. 1–39, 2014.

[74] H. Alers, L. Bos, and I. Heynderickx, "How the task of evaluating image quality influences viewing behavior," in *in Proceedings of International Workshop on Quality of Multimedia Experience*, 2011, pp. 167–172.

[75] 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, no. 7, pp. 1–20, 2008.

[76] B. W. Tatler, "The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions," *Journal of Vision*, vol. 7, no. 14, pp. 1–17, 2007.

[77] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE CVPR*, 2005, vol. 1, pp. 886–893 vol. 1.

[78] Q. Yang, K. Tan, and N. Ahuja, "Real-time O(1) bilateral filtering," in *Proceedings of IEEE CVPR*, 2009, pp. 557–564.

[79] A. Borji, D. N. Sihite, and L. Itti, "Salient object detection: A benchmark," in *Proceedings of the ECCV*, pp. 414–429. 2012.

[80] Y. Zhai and M. Shah, "Visual attention detection in video sequences using spatiotemporal cues," in *Proceedings of ACM Multimedia*, 2006, pp. 815–824.

[81] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[82] T. Hastie, R. Tibshirani, and J. Friedman, "Boosting and additive trees," in *The Elements of Statistical Learning*, pp. 337–387. 2009.

[83] M. Schmidt, K. Murphy, G. Fung, and R. Rosales, “Structure learning in random fields for heart motion abnormality detection,” in *Proceedings of IEEE CVPR*, 2008, pp. 1–8.

[84] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” *IEEE Transactions on PAMI*, vol. 34, no. 11, pp. 2189–2202, 2012.

[85] M. Cheng, Z. Zhang, W. Lin, and P. H. S. Torr, “BING: Binarized normed gradients for objectness estimation at 300fps,” in *IEEE CVPR*, 2014.

[86] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proceedings of the American Mathematical Society*, vol. 7, no. 1, pp. 48–50, 1956.

[87] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[88] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1/2, pp. pp. 81–93, 1938.

[89] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” 2014.

[90] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, pp. 1097–1105. 2012.

[91] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” 2014.

[92] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[93] Z. Wu and M. Palmer, “Verbs semantics and lexical selection,” in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 1994, pp. 133–138.

[94] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st edition, 2009.

[95] K. Zia, T. Balch, and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE Transactions on PAMI*, vol. 27, no. 11, pp. 1805–1819, 2005.

[96] S. Stalder, H. Grabner, and L. Van Gool, “Cascaded confidence filtering for improved tracking-by-detection,” in *Proceedings of ECCV*, 2010, pp. 369–382.

[97] G.K. Wallace, “The JPEG still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. 18–34, 1992.

[98] J. Alers, H.and Redi, H. Liu, and I. Heynderickx, “Studying the effect of optimizing image quality in salient regions at the expense of background content,” *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 043012–043012, 2013.

[99] B. Jin, G. Yildirim, C. Lau, A. Shaji, M. O. Segovia, and S. Süsstrunk, “Modeling the importance of faces in natural images,” in *Proceedings of SPIE Human Vision and Electronic Imaging*, 2015, vol. 9394, pp. 93940V–93940V–11.

[100] A. C. Little, B. C. Jones, and L. M. DeBruine, “The many faces of research on face perception,” *Philosophical Transactions of the Royal Society of London*, vol. 366, no. 1571, pp. 1634–1637, 2011.

[101] G. Yildirim, A. Shaji, and S. Süsstrunk, “Saliency detection using regression trees on hierarchical image segments,” *Proceedings of IEEE ICIP*, 2014.

[102] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, “Salient region detection and segmentation,” in *Proceedings of ICVS*, 2008, pp. 66–75.

[103] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *Proceedings of IEEE Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[104] R. Rosenholtz, A. Dorai, and R. Freeman, “Do predictions of visual perception aid design?,” *ACM Transactions on Applied Perception*, vol. 8, no. 2, pp. 12:1–12:20, 2011.

[105] C. Shen and Q. Zhao, “Webpage saliency,” in *Proceedings of ECCV*, vol. 8695, pp. 33–46. 2014.

[106] R. M. Jiang, A. Bouridane, and A. Amira, “Color saliency evaluation for video game design,” in *Advances in Low-Level Color Image Processing*, vol. 11, pp. 409–425. 2014.

[107] G. Yildirim and S. Süsstrunk, “FASA: Fast, accurate, and size-aware salient object detection,” *Proceedings of ACCV*, 2014.

EPFL/IC/IVRL BC 316
Station 14, 1015, Lausanne, Switzerland
☎ *+41 78 948 12 40*
✉ *gokhan.yildirim@epfl.ch*
🖰 *http://ivrl.epfl.ch/people/yildirim*

Gökhan Yıldırım

Education

2011 - 2015 **Ph.D. in Image and Visual Representation Laboratory**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Switzerland.

2010 - 2011 **Ph.D. Fellowship in School of Computer and Communication Sciences**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Switzerland, *GPA – 5.71/6.00.*

2008 - 2010 **M.Sc. in Electrical and Electronics Engineering**, *Middle East Technical University (METU)*, Turkey, *GPA – 3.71/4.00.*

2004 - 2008 **B.Sc. in Electrical and Electronics Engineering**, *Middle East Technical University (METU)*, Turkey, *GPA – 3.91/4.00 –* 6*th* *among 200.*

Ph.D. Thesis

Title *Are All Pixels Equally Important? Towards Multi-Level Salient Object Detection*

Supervisors Professor Sabine Süsstrunk

Description This thesis explores the varying importance of objects by investigating visual saliency on complex natural images with multiple objects.

M.Sc. Thesis

Title *Antenna Patterns for Detecting Slowly Moving Targets in Two Channel GMTI Processing*

Supervisors Professor Sencer Koc

Description This thesis explores the mathematical requirements for the optimal antenna patterns to detect slowly moving targets by processing radar signals.

Experience

Summer 2013 **Ph.D. Internship**, National University of Singapore (NUS), Singapore.
Research on visual saliency for three months under the supervision of Professor Mohan Kankanhalli.

2008 – 2010 **System Engineer**, ASELSAN INC., Ankara, Turkey.
Full time job as a system engineer, responsible for developing radar-image formation, enhancement, and target detection and identification algorithms and their integration to the whole system.

Computer skills

Programming C++ (including OpenCV), Objective-C

Software MATLAB, Xcode, Microsoft Visual Studio, Latex, Microsoft Office

Professional Interests

Image & video enhancement, restoration, compression, and registration

Object detection & recognition with real-time applications

Software development in C++

Publications

J. Bin, **G. Yildirim**, C. Lau, A. Shaji, S. Süsstrunk, "Modeling the Importance of Faces in Natural Images", *SPIE Human Vision and Electronic Imaging*, 2015

G. Yildirim, S. Süsstrunk, "FASA: Fast, Accurate, and Size-Aware Salient Object Detection", *Asian Conference on Computer Vision*, 2014

G. Yildirim, A. Shaji, S. Süsstrunk, "Saliency Detection Using Regression Trees on Hierarchical Image Segments", *International Conference on Image Processing*, 2014

G. Yildirim, A. Shaji, S. Süsstrunk, "Estimating Beauty Ratings of Videos Using Supervoxels", *ACM International Conference on Multimedia*, 2013

G. Yildirim, S. Süsstrunk, "Rare is Interesting: Connecting Spatio-Temporal Behavior Patterns with Subjective Image Appeal", *ACM International Conference on Multimedia, Workshop on Geotagging and Its Applications in Multimedia*, 2013

G. Yildirim, R. Achanta, S. Süsstrunk, "Text Recognition in Natural Images Using Multiclass Hough Forests", *International Conference on Computer Vision Theory and Applications*, 2013

Upcoming Journal Publications

G. Yildirim, D. Sen, M. Kankanhalli, S. Süsstrunk, "Multi-Level Object Saliency", (title is tentative)

G. Yildirim and S. Süsstrunk, "Multi-Level Salient Object Detection with an Application to Image Tagging", (title is tentative)

Honors and Awards

2013	Outstanding teaching assistant award in EPFL
2004 – 2006	The best student award in Electronics Engineering in METU in three semesters
2004	707th among over 1.5 million students in University Entrance Exam in Turkey

Supervised Projects

Spring 2014	M.Sc. Semester Project, "Image Deblurring Using Motion Sensors", EPFL
Fall 2013	B.Sc. Semester Project, "Race Car Recognition", EPFL
Fall 2012	M.Sc. Semester Project, "Talking to the Camera", EPFL

Languages

Turkish	**Mother tongue**	
English	**Advanced**	*Fluent*
French	**Basic**	*Basic words and phrases only*