HUMAN ATTENTION SIMULATION ON NATURE SCENES IN COMPUTER VISION

by

Nianyi Li

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Fall 2017

© 2017 Nianyi Li
All Rights Reserved

HUMAN ATTENTION SIMULATION ON NATURE SCENES

IN COMPUTER VISION

by

Nianyi Li

Approved: _____

Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: _____

Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____

Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Jingyi Yu, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chandra Kambhamettu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Li Liao, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Haibin Ling, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor, Prof. Jingyi Yu for the continuous support of my Ph.D study and related research. He is one of the smartest persons I have ever met in my life. Not only have I learnt the way to do research from him, but also the positive attitude towards life. His guidance helped me in all the time of research and writing of this thesis. Thanks for making my Ph.D. experience a productive and interesting one. I would also like to thank the rest of my thesis committee members: Professor Chandra Kambhamettu, Professor Li Liao, and Professor Haibin Ling, for their insightful suggestions and comments.

My sincere thanks also goes to the PLEX-VR company and ShanghaiTech University, which provided me an opportunity to join their group as intern, and access to the laboratory and research facilities to finish my research on the personalized saliency detection. Typically, many thanks to Professor Shenghua Gao, Yanyu Xu, and Junru Wu in ShanghaiTech for their insight in deep learning methods and their help on gathering the vast amount of individual gaze data. Without their precious support, it would not be possible to conduct this research. I would like to thank Dr. Scott McCloskey, who gave me the intern chance to join the image processing lab in Honeywell, and for his help in my light field and super-resolution research. He is definitely one of the best mentor I have ever known.

I thank my fellow labmates for bearing my endless questions and proving generous help on my research (Jinwei Ye, Yu Ji, Haiting Lin, Wei Yang, Bilin Sun, Can Chen, Mingyuan Zhou, Xinqing Guo, Yang Yang, and Zhong Li), for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years. Also I thank my friends Ningjing Tian, Matthew Saponaro, Dainan Zhang, and Xinyu Liu, for the laughter and happiness they bring over me.

I gratefully acknowledge the funding sources that made my Ph.D. work possible. My work is partially supported by National Science Foundation under grands IIS-1218156, IIS-1218177, RI-1422477 and CRI-1513031.

Last but not the least, I would like to thank my parents for supporting me spiritually throughout writing this thesis and my life in general.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Human Attention Simulation is a long-standing problem in computer vision area. Researchers either attempt to locate the most interesting objects for human beings in images or predict where people will look at in nature scenes. Accurate and reliable human attention detection can benefit numerous tasks ranging from tracking and recognition in vision to image manipulation in graphics. For example, successful saliency detection algorithms facilitate automated image segmentation, more reliable object detection, effective image thumbnailing and retargeting. However, state-of-the-art techniques neglect some essential problems that limit saliency models from precisely simulating human attention mechanism. On the input front, images used for saliency detection tasks always fail to preserve the high dimensional information of the scene; on the task front, it is inevitable to observe inconsistency among ground truth provided by different persons, resulting in uncertainty on prediction performances.

Regarding the input data, existing saliency detection approaches using images as inputs are sensitive to foreground/background similarities, complex background textures, and occlusions. I explore the problem of utilizing light fields as input for saliency detection. The proposed technique is enabled by the availability of commercial plenoptic cameras that capture the light field of a scene in a single shot. I show that the unique refocusing capability of light fields provides useful focusness, depths, and objectness cues. I further develop a new saliency detection algorithm tailored for light fields. To validate the approach, I acquire a light field database of a range of indoor and outdoor scenes and generate the ground truth saliency map. Experiments show that the saliency detection scheme can robustly handle challenging scenarios such as similar foreground and background, cluttered background, complex occlusions, *etc*., and achieve high accuracy and robustness.

As for methods using high-dimensional data beyond regular images as saliency input, they are tailored for different data types. Those techniques adopt very different solution frameworks, in both types of features and procedures on using them. In this dissertation, I present a unified saliency detection framework for handling heterogeneous types of input data. The proposed approach builds dictionaries using data-specific features. Specifically, I first select a group of potential foreground superpixels to build a primitive saliency dictionary. I then prune the outliers in the dictionary and test on the remaining superpixels to iteratively refine the dictionary. Comprehensive experiments show that the proposed approach universally outperforms the state-of-the-art solution on all 2D (regular image), 3D (image with depth information) and 4D (light field) data.

Regarding the saliency detection task, tremendous efforts have been focused on exploring a universal saliency model across users despite their differences in gender, race, age, *etc*. Yet recent psychology studies suggest that saliency is highly specific than universal: individuals exhibit heterogeneous gaze patterns when viewing an identical scene containing multiple salient objects. In this dissertation, I show that such heterogeneity is common and critical for reliable saliency prediction. The conducted study also produces the first database of personalized saliency maps (PSMs). I model PSM based on universal saliency map (USM) shared by different participants and adopt a multi-task CNN framework to estimate the discrepancy between PSM and USM. Comprehensive experiments demonstrate that the new PSM model and prediction scheme are effective and reliable.

Chapter 1

INTRODUCTION

Human Attention Simulation is a long-standing problem in computer vision area. Researchers either attempt to locate the most interesting objects for human beings in images or predict where people will look at in nature scenes. Saliency refers to a component (object, pixel, person) in a scene that stands out relative to its neighbors and has been considered key to human perception and cognition. Traditional saliency detection techniques attempt to extract the most pertinent subset of the captured sensory data for predicting human visual attention. Applications are numerous, ranging from compression [46] to image re-targeting [100], and most recently to virtual reality and augmented reality [19].

By far, nearly all previous approaches have focused on using RGB low-dimensional data as input. In the meantime, the majority algorithms attempt to explore a universal saliency model, i.e., to predict potential salient regions common to users while ignoring their differences in gender, race, age, personality, etc. In this dissertation, I thoroughly explore utilizing the light field data as input of the salient object detection frameworks. I further study the inconsistency eye fixation problems among individuals and propose a novel framework that can encode the personalized gaze discrepancy into the prediction model.

1.1 Challenging problems in Saliency Prediction

Regular RGB Image as Input

State-of-the-art solutions [8] have focused on integrating low-level features (pixels or superpixels) and high-level descriptors (regions or objects). However, existing solutions have many underlying assumptions, *e.g.*, the foreground should have a different color from the background, the background should be relatively simple and smooth, the foreground is occlusion free, *etc*. In reality, many real images violate one or multiple assumptions, and

existing saliency algorithms are inherently different from how human visual system detects saliency. Human eyes have two unique properties that are largely missing in existing saliency solutions on regular images. First, human eye can conduct dynamic refocusing that enables rapid sweeping over different depth layers. Hence, for humans, the input is a focal stack instead of a single. Second, human uses two eyes to infer scene depth, e.g., via stereo, for more reliable saliency detection whereas most existing approaches assume that the depth information is largely unknown.

Heterogenous Types of Input Data

There is an emerging interest in using high-dimensional datasets beyond regular images (2D) in saliency detection. For instance, the using of image with depth information (3D) [88, 67] and more recently 4D light field data [75]. Despite their effectiveness by adopting high dimensional information, saliency detection algorithms based on 2D, 3D and 4D data have adopted completely different frameworks, due to the heterogenous low-level features directly from the data. A unified saliency prediction methods, that can handle different types of data simultaneously, benefits the related several vision applications, such as object detection [5, 32] and retargeting [97].

Universal Saliency Prediction

By far, nearly all previous approaches have focused on exploring a universal saliency model, i.e., to predict potential salient regions common to users while ignoring their differences in gender, race, age, personality, etc. Such universal solutions are beneficial in the sense they are able to capture all "potential" saliency regions. Yet they are insufficient in recognizing heterogeneity across individuals. Examples in Fig. 5.1 illustrate that while multiple objects are deemed highly salient within the same image (eg, *human face* (first row), *text* (last tow rows) and object of (*high color contrast*), different individuals have very different fixation preferences when viewing the image. For the rest of the dissertation, I use term *universal saliency* to describe salient regions that incur high fixations across all subjects and term *personalized saliency* to describe the heterogeneous ones.

1.2 Dissertation Statement

In this dissertation, I first explore how to conduct salient object detection using light field as input and further present a universal saliency detection framework for handling heterogenous types of input data. I then explore the personalized saliency prediction problem, which encode the incongruent gaze pattern of individuals into the prediction model.

Saliency Detection with High-dimensional Input.

We first thoroughly discuss the benefits, modeling and the adopted features of utilizing the high-dimensional scene data as input for the salient object detection tasks. I construct the first light field data sets for saliency detection tasks and provide a unified solution for handling regular RGB (2D), image with depth information (3D), and 4D light field data. Experiments show that our saliency detection scheme can robustly handle challenging scenarios such as similar foreground and background, cluttered background, complex occlusions, *etc*., and achieve high accuracy and robustness on different dimensional datasets.

Personalized Saliency Prediction.

I then present a comprehensive analyze of the inconsistent gaze pattern problems among different persons. I show that such heterogeneity is common and critical for reliable saliency prediction. Our study also produces the first database of personalized saliency maps (PSMs). We model PSM based on universal saliency map (USM) shared by different participants and adopt a multi-task CNN framework to estimate the discrepancy between PSM and USM. Comprehensive experiments demonstrate that our new PSM model and prediction scheme are effective and reliable.

1.3 Contributions

This dissertation makes the following contributions in computer vision.

Saliency Detection on High-dimensional datasets:

- The first light field dataset for salient object detection is proposed. We acquire a light field database of a range of indoor and outdoor scenes and generate the ground truth

saliency map. We have already shared this database, *i.e.*, Light Field Saliency Detection (LFSD) Dataset, to community online[1].

- The first saliency algorithm tailored for light fields input. The key advantage of using a light field instead of a single image is that it provides both focusness and depth cues. Our solution echoes these observations and also provides an alternative and more robust method to extract these cues through the analysis of light fields. Experiments show that our technique can handle many challenging scenarios that cast problems on traditional single-image-based algorithms.
- The first unified framework for different dimensional data types. I present a novel saliency detection algorithm that is applicable to 2D image data, 3D stereo/depth data, and 4D light field data without modifying the processing pipeline. We first develop a data-specific feature vector descriptor. For 2D data, it corresponds to color and textures. For 3D, we append depth information. For 4D, we further append focusness measures. We show that two types of feature descriptors are complimentary to each other for handling variational types of texture/color scene compositions. Compared with state-of-art techniques that commonly adopt different solution frameworks for handling different data inputs, our technique does not require modifying the algorithm but only the input descriptor. Comprehensive experiments have shown that it outperforms previous tailored solutions for different data types.

Personalized Saliency Prediction:

- The first dataset for personalized saliency prediction. We present the first dataset of personalized saliency maps (PSMs) that consists of 1600 images viewed by 20 human subjects. To improve reliability, we ensure that each image is viewed by every subject for 4 times over about one week interval. We use the '*Eyegaze Edge*' eye tracker to track gaze and produce a total of 32,000 (1600×20) fixation maps. To correlate the acquired PSMs and the image contents, we manually segment each image into a collection of objects and semantically label them. Our annotated dataset provides

[1] http://www.eecis.udel.edu/~nianyi/LFSD.htm

fine-grained semantic analysis for studying saliency variations across individuals. For example, we observed that certain types of objects such as watches, belts would introduce more incongruity (possibly due to gender differences) whereas other types such as faces would lead to more coherent fixation maps.

- Encoding the gaze inconsistency into the prediction model. We further present a computational model towards this personalized saliency detection problem. Notice that saliency maps from different individual still share certain commonality via the USM. Hence, we model the PSM as a combination of USM and a residual map which is related to the identity and the image contents. We adopt a multi-task convolutional neural network (CNN) to identify the discrepancy between PSM and USM for each person. Experimental results demonstrate the effectiveness of our framework.

1.4 Blueprint of the Dissertation

This dissertation is organized as follows. I give a more thorough review of the saliency detection models and related features to this thesis in Chapter 2. In Chapter 3, we explore the problem of using light fields as input for saliency detection. Specifically, I first show that the unique refocusing capability of the light fields provides useful focusness, depth and objectness cues and then introduce our acquired light field datasets and our tailored algorithm on this dataset. Chapter 4 discusses a unified saliency detection framework for handling heterogenous types of input data. Chapter 5 presents our framework for predicting the fixation maps for individuals. We introduce how we construct the image dataset for personalized saliency prediction and then discuss our multi-task CNN model for computing the discrepancy map between the universal and personalized eye fixation prediction. Chapter 6 concludes the thesis and discusses future extensions.

Chapter 2

BACKGROUND AND PREVIOUS WORK

This chapter discusses the background and the previous work on human attention simulation (Saliency prediction) in computer vision. We first explore the classical computational models to simulate human visual attention and then outlining two divisions of models based on their architectures. We then discuss the most related features/cues that we use in this dissertation in Section 2.2. Next, we summarize the state-of-the-arts deep learning methods in modeling visual attention in Section 2.3.

2.1 Simulating human visual attention in computer vision

There is an increasing interest in locating the objects/regions/gaze points that attract human beings' attention in computer vision, robotics, computational photography, computer graphics and design, human-computer interaction. An efficient human attention mechanism can help to assign priorities to different image parts and thus directs the analysis process to exam more interesting locations first. Simulating the human visual attention can also help researchers to understand human perception, and to improve vision system. The highly effective attention simulation algorithms have been studied extensively from the psychological theories of the human visual system. The computer vision community uses the term saliency, borrowed from cognitive psychology, for two different tasks, *i.e.*, the attention-motivated saliency and the saliency in local feature detection [6, 59, 58]. The antention-motivated saliency aims to assign a probability value to each pixel in the image or the scene to indicate the likelihood of attending to every location in an image/scene, thus achieving more efficient analysis. In local feature detection tasks, saliency refers to a relatively large number of points (or small regions), whose location are stable under pose and illumination changes w.r.t. the

(a) Regular image (b) Salient object (c) Eye fixation

Figure 2.1: Illustration of the ground truth of two categories of saliency models. (a) A nature image. (b) Salient Object detection methods aim at detecting the whole salient objects in the scene. (c) Fixation prediction models tend to simulate the gaze pattern (red indicate higher possibility for human looking at this point).

objects in the scene. In this dissertation, we only discuss the first category, and the term "*saliency*" in the following content are all indicating the attention-motivated saliency.

The literature for saliency prediction is huge and existing solutions can be roughly classified into two categories by their targeting tasks: *Salient Object Detection* and *Eye Fixation Prediction*. In salient object tasks, image labelers are asked to annotate an image by drawing either bounding box or pixel-accurate contour lines of the objects that they believe to be salient in the given images. The goal of a salient object method is to generate a map that matches the annotated salient object mask. Unlike the salient object detection algorithms that tend to highlight specific objects/regions in an image, the saliency in fixation prediction experiment is defined by pixel-wise eye gaze points. Specifically, participants are asked to view each image/scene for seconds while their eye fixations are recorded by a eye tracker. The eye fixation prediction algorithms aim to compute a probabilistic map of an image to indicate the actual human gaze patterns. Generally, for salient object detection task, the test saliency maps are binary maps obtained by first averaging the individual segmentations from the ground-truth subset, and then threshold with a fix value Th to generate the binary masks for each subset. In the fixation task, the ground-truth map for each image is generated by first plotting all the fixation points from either all or individual participants, and then filter the fixation points map by a 2D Gaussian kernel with a fixed σ of the image width.

The salient object detection models aim to *identify salient regions/objects* in the image/scene. Generally, a salient object detection model should, first detect the salient attention-grabbing objects in a scene, and second, segment the entire object [8]. The saliency map generated by the algorithm highlight the pixels which are more likely belonging to the salient object. A precisely salient object detection methods benefits various computational applications, *i.e.*, image processing and understanding [45, 59, 104, 82]. Saliency detection is related to many vision applications, e.g., object detection/recognition [5, 32],image/ video compressing [12, 36], effective image thumbnailling [103] and retargeting [97].

The goal of the fixation based saliency is to compute a probabilistic map (saliency map) to simulate the *eye movement behaviors* of human. Typically, pixel/region with higher saliency value indicate larger possibility that human will look at it when free-viewing the images. After Itti *et al*. [45] introducing the first computational models for fixation prediction to the computer vision community, numerous models have been proposed during the recent decades to predict both the fixation and salient objects in images. Utilizing the pixel-based [40, 34] or region-based [45, 13, 50] features, these fixation models compute the *pixel-wise* saliency map by a local or global interaction step that combines the re-weighted or re-normalized feature saliency values.

Recent studies have shown that the two tasks of salient object detection and eye fixation prediction are correlated [120, 78]. Specifically, based on the fact that the locations of salient objects in the scenes providing guidance to human eye fixations, Li *et al*. [78] utilize a simple eye fixation based model for segmenting salient objects in an image and achieved state-of-the-art results. Nevertheless, it is usually hard to apply the algorithms tailored for salient object detection to predict eye fixation and vice versa. It is because that unlike salient object models, which generate smooth connected areas, the fixation prediction models often pop-out sparse blob-like salient regions. Typically, detecting large salient areas are doomed to cause severe false positives for fixation prediction. Moreover, popping-out only sparse salient regions causes massive misses in detecting salient regions and objects. However, the developing of deep learning methods makes it possible to simultaneously address the related aspects of eye fixations and object saliency, a more detailed discussion of the deep learning

application in saliency detection is in the Section 2.3 of this thesis.

2.1.1 Bottom-up and top-down saliency models

According to whether the detection procedure requires human interaction or not, existing methods are divided into two categories: bottom-up and top-down approaches. The first category usually determines the saliency of a pixel based on low-level stimuli-driven features without any prior of the salient region or object [45, 33, 68]. On the contrary, the second one often describes the saliency by the visual knowledge constructed from the training process, and then use such knowledge for saliency detection on the test images [35, 80, 108]. During normal human perception, both mechanisms interacts.

State-of-the-arts computational models that are merely based on bottom-up methods have shown high performance in large scale datasets, due to the successfully application of several effective low-level features, *e.g.*, color contrast feature [48, 63, 80, 20], background features [112, 117], focusness features [55, 75], and sparse coding [74]. Results from perceptual research [28, 91] and previous approaches [49, 93] indicate that the most influential factor in bottom-up visual saliency is contrast. The definition of contrast in previous works is mainly based on different types of image features, such as color variation, edges and gradients [45], spectral analysis [39], histograms [21], multiscale descriptors [80], or combinations thereof [9]. Both types of methods tend to rely solely on the local center-surround contrast [45, 80] or the global contrast [1, 20] with respect to the entire scene for estimating the saliency.

Methods using the bottom-up architecture are usually built on the scheme proposed by Koch *et al*. [63]. In their method, an image is represented by various low-level attributes such as color, intensity, and orientation across several spatial scales which are then linearly or non-linearly normalized and combined to form a master saliency map. Another major contribution of their work is the idea of the center-surround contrast framework, which define saliency as distinctiveness of an image region to its immediate surroundings. This scheme also proposes a solution for both object detection and fixation models. Base on their model, Itti *et al*. [45] adopt a Difference of Gaussians (DoG) approach to process the captured

color, intensity and orientation features, and linearly combine the generated feature maps to produce the final saliency maps. However, [21] point out that the resulting saliency maps by [45] are often blurry and overemphasize the small, purely local features, leading to this the method less useful for applications such as segmentation and detection. Ma and Zhang [82] propose an alternate local contrast analysis for computing saliency maps, which is then extended utilizing a fuzzy growth model. Harel *et al*. [50] normalize the features of [45] to assign high saliency values to distinct regions and enable the combination with other essential maps to further generate the final saliency. This simple framework is biologically plausible and allowing parallelization. Liu *et al*. [80] usea Gaussian image pyramid to linearly combine multi-scale contrast. Cheng *et al*. [21] define saliency of an object by its local and global color uniqueness and spatial distribution. They use super-pixel to classify the nearby pixels into small regions, and generate 3D histograms to represent each super-pixel. The saliency in their work refers to the incongruity between the histogram bin.

Top-down approaches [83, 116] use visual knowledge commonly acquired through learning to detect saliency. Approaches in this category are highly effective on task-specified saliency detection, *e.g*., identifying human activities [84]. For this kind of algorithms, it is essential to effectively learn the differences between salient objects and background from images. Some top-down factors are already well known, and some is still waiting for being further explored. Einhäuser *et al*. [29] propose that objects are better indicators for fixation than the bottom-up saliency. [18] show that faces and text attract human gaze. [95] demonstrate that objects with specific emotion, *e.g*. the angry bird, and object with action motivation are more likely to attract attention. Judd *et al*. [57] point out that human figures, faces, cars, text, and animals attract visual attention most. Alongside, cultural and characteristic factors, age, and experience will also affect the gaze pattern of human being [22]. It has been increasingly popular to use deep networks for saliency detection, because that the top layers of the neural networks contains rich high-level information of images. Huang *et al*. [42] propose to fine-tune CNNs pre-trained for object recognition via a new objective function based on saliency evaluation metrics such as Normalized Scanpath Saliency (NSS), Similarity, or KL-Divergence,*etc*. Pan *et al*. [90] propose to use a shallow convnet

trained from scratch and fine-tune a deep convnet that trained for image classification on the ILSVRC-12 dataset, we will discuss the deep learning methods in detail in Section 2.3.

Integrating the top-down information into the bottom-up framework is proved to be more efficient than merely using one type of features [18, 95, 29, 57]. Phycological research [16, 122, 106] on human gaze pattern indicate that, at the first few hundred milliseconds of the early stages of free viewing, the core factor to determine visual attention is the image-based conspicuity. As the viewing time increasing, high-level factors, *e.g*. the image context, will take charge of the attention mechanism. However, these high-level factors may not necessarily translate to saliency boosted by low level-features, such as color, intensity, and orientation, and should be considered separately. For example, a face of a human or an animal in the image may not attract the most of attention of people compared with other object in the scene, and people may still notice the faces in the scene. Cerf *et al*. [18] refined the bottom-up model by Itti and Koch [63], and add a conspicuity map indicating the location of faces and text and show impressive improvement on the detection performance. They also add a human defined object-map to further refine the saliency result. Moosmann *et al*. [85] propose an iterative algorithm based on the online estimation of the position of object parts. They treat saliency as the set of attributes that distinguishes a concept (object category) the most from others. Goferman *et al*. [35] employ the object-specific information in their detection framework and use its detection results to generate the binary map. Then also take a max operation to combine the bottom-up and top-down result.

2.1.2 Salient detection on high dimensional data

A main contribution of this dissertation is that we build computational models to predict saliency in high dimensional data, and extent the saliency detection to new areas. In this section, we category state-of-the-arts saliency models based on the data type they adopt as input. To avoid duplicate statement with Section 2.3, we do not include the deep learning based algorithms in this section.

2D saliency. Human vision system is particularly sensitive to high-contrast stimulus [44, 89, 96] and traditional approaches have focused on applying this model to 2D images.

Most contrast-based methods measure saliency by feature (color, texture,gradient, shape, etc.) difference between pixels/superpixels. The performance of 2D image-based techniques depend highly on the choice of feature descriptors. For example, if the color difference between foreground and background is small, methods based on color feature descriptors can lead to poor performance. To address this issue, recent algorithms incorporate high-level reasoning into the solution framework. For example, additional cues that emulate human vision systems such as focusness, objectness, location of specific types of object (e.g., faces) [54, 93, 113] have been added onto the feature descriptor.

3D saliency. More recent approaches acknowledge that 2D images do not completely represent how human eyes perceive the world [88, 67]. In particular, depth perception provided by two eyes has been largely ignored in saliency detection. Therefore, several new approaches have been proposed to incorporate 3D depth information into saliency detection. In [88] work, a disparity map is first inferred from a stereo pair and later used to enhance saliency detection. The results are promising. For example, the depth map can help distinguish foreground from background even if they have similar appearance. One major challenge in those approaches is how to effectively combine traditional features with depth features without modifying the solution framework.

4D saliency. There is also emerging interest on using datasets beyond 3D such as the light field towards the scene [75]. A unique feature of light field is that it enables dynamic refocusing through light field rendering. In [75], the focal stack is used to infer focusness and objectness of superpixels for more reliably selecting the background candidates and foreground saliency candidates. It then integrates other cues based on color and texture contrast. This 4D saliency method eliminates the need of 3D depth maps and shows impressive results on challenging scenarios including similar foreground and background, clustered background, complex occlusions, etc. Nevertheless, the solution framework is significantly different from previous 2D and 3D solutions.

2.2 Related Features/Cues

The features/cues used in saliency detection are various. Here, we only introduce the most related ones to our proposed models in this thesis.

2.2.1 Center vs. Background Priors.

Many saliency detection schemes exploit contrast cues, *i.e.*, salient objects are expected to exhibit high contrast within certain context. Koch and Itti [45] are the first to use center-surround contrast of low level features to detect saliency. Motivated by their work, many existing approaches compute the center-surround contrast either locally or globally. Local methods compute the contrast within a small neighborhood of pixels by using color difference [13], edge orientations [80], or curvatures [107]. Global methods consider statistics of the entire image and rely on features such as power spectrum [39], color histogram [21], and element distributions [101].

Although the center-surround approaches are proven highly effective, Wei *et al*. [112] suggested that background priors are equally important. In fact, one can eliminate the background to significantly improve foreground detection. Yang *et al*. [117] observed that connectivity is an important characteristics of background and used a graph-based ranking scheme to measure patch similarities. Since most existing approaches rely on color contrast, when the foreground and background have similar color, these approaches can easily fail. Our approach resolves this issue by combining color contrast, background prior, and focusness prior w.r.t. different depth layers obtain from the light field.

2.2.2 Focusness and Objectness Cue.

Jiang *et al*. [55] proposed that objects of interest in an image are often photographed in focus. This naturally associates the focusness with that saliency. They estimated the focusness by the scale of edges using scale-space analysis. In addition, they also proposed an objectness estimation which utilized the probability of a region belongs to a complete object in some local windows to measure. Regarding our techniques vs. [55], we want to emphasize that our scheme is advantageous over [55] in several ways. First, our focusness

cue is extracted directly from a complete focal stack produced by the 4D light field whereas the cue has to be inferred from a single image in [55]. Therefore, our technique is more robust and reliable especially on the images that contain similar foreground/background and/or lack defocus cues. Second, the availability of light fields facilitates easier an effective extraction of location, contrast and foreground cues. These cues, in many ways, serve the similar purpose of uniqueness and objectness cues in [55] but are more robust. Third, our objectness cue is concerned as to an focus stack slice not to a certain region or pixels, which accelerates the computational speed.

2.2.3 Depth Cue.

Recent studies on human perception [67] have shown that depth cue plays a important role in determining salient regions. However, only a handful of works incorporate depth maps into saliency models. Maki *et al*. [84] used depth cue to detect human motions. Their depth features are highly task-dependent and the detection is performed in a top-down fashion. Niu *et al*. [88] computed saliency based on the global disparity contrast in a pair of stereo images. Lang *et al*. [67] used a Kinect sensor to capture the scene depth. Ciptadi *et al*. [23] used 3D layouts and shape features from depth maps. Peng *et al*. [92] detected saliency taking account of both depth and appearance cues derived from low-level feature contrast, mid-level region grouping and high-level priors enhancement.

2.3 Deep learning methods in saliency prediction

Even though by combining the hand-crafted bottom-up and top-down features saliency models have achieve relative high performance on the popular dataset, recent advances in deep learning and the availability of large datasets have enabled models to perform end-to-end learning.

The seminal work of Krizhevsky *et al*. [65] introduce the Deep Convolutional Networks into the computer vision community, and bring a paradigm shift in vision research from hand-crafting features to learning them directly from data. Motivated by the functioning of cells in visual cortex of primates, this early deep networks target at providing solutions

to image classification. By using this network, Krizhevky *et al*. successfully captured rich visual features with semantical meanings in a hierarchical fashion. They also show high performance in the related pixel-level image processing tasks such as semantic object segmentation [81] and depth estimation [27].

Vig *et al*. [109] propose the first framework to model saliency with deep convolutional networks (DCNs), where feature maps from different layers in a 3-layer ConvNet are fed into a simple linear classifier for distinguishing salient from non-salient regions. However, due to the limited number of image data online, this architecture fail to reach the state-of-the-arts performance. This method facilitates another popular deep network architecture, *i.e*. the *DeepGaze* [62]. This deep visual attention neural network use the existing *AlexNet* [65], which is trained for image classification, to predict fixation maps. Particularly, they remove the fully connected layers of *AlexNet* network and generate a high dimensional feature space, which is then linearly combined to predict the fixation saliency. Based on the *DeepGaze* framework, Srinivas *et al*. propose a *DeepFix* network kruthiventi2015deepfix, which adopts very deep networks to capture semantic features at multiple scales, to predict saliency. This method uses Location Biased Convolution filters to allow the network to exploit location dependent patterns. The *SALICON* model [42] use a new objective function based on the saliency evaluation metrics (Normalized Scanpath Saliency, Similarity, and KL-Divergence *etc*.) to fine-tune the CNNs, which is pre-trained for object recognition (*AlexNet* [65], *VGG-16* [102] and *GoogLeNet* [105]). This model incorporate multiple scales to select attention at different resolutions. Pan *et al*. [90] proposed to use a shallow CNN trained from scratch and another deep CNN where the weights of its first 3 layers was adapted from *VGG_CNN_M* trained for image classification. Liu *et al*. [79] proposed a multi-resolution CNNs where three final fully connected layers are combined to form the final saliency map.

Other methods focused on saliency object detection. Zhao *et al*. [123] collect both local and global context information by employing two parallel network to detect the salient object in the scene. The input data of their framework contains a pre-processed superpixel-centered window to feed the two ConvNets separately. They use a fully connected layers to combine the outputs of the two ConNets and generate the final saliency map. Li and

Yu [73] feed three nested windows to three different ConvNets at different scales, and then fuse the three ConvNets to predict the salient object. Wang *et al*. [110] propose a two-step deep network for integrate local and global information from the image. Specifically, they first learn local saliency by local features detected from a deep neural network, *i.e*. DNN-L. Next, they train another deep network (DNN-G) to assign values to each object based on the local saliency, global contrast feature and geometric information. Kruthiventi *et al*. [66] proposed a unified framework to predict eye fixation and segment salient objects. This multi-task CNN shares the initial network layers to capture the objet level semantics and the global contextual aspects of saliency. Then, they feed the captured low-level features to two separate CNNs to address the task specific aspects.

Chapter 3

SALIENCY DETECTION ON LIGHT FIELD

In this chapter, I explores the problem of using light fields as input for saliency detection. Specifically, I first show that the unique refocusing capability of the light fields provides useful focusness, depth and objectness cues and then introduce our acquired light field datasets and our tailored algorithm on this dataset.

3.1 Motivation

State-of-the-art solutions [8] have focused on integrating low-level features (pixels or superpixels) and high-level descriptors (regions or objects). However, existing solutions have many underlying assumptions, *e.g.*, the foreground should have a different color from the background, the background should be relatively simple and smooth, the foreground is occlusion free, *etc*. In reality, many real images violate one or multiple assumptions as shown in Fig. 5.1.

By far, nearly all existing saliency detection algorithms utilize images acquired by a regular camera. In this dissertation, we explore the salient object detection problem by using a completely different input: the light field of a scene. A light field [14] can be essentially viewed as an array of images captured by a grid of cameras towards the scene. Commercial light field cameras can now capture reasonable quality light fields in a single shot. Lytro, for example, mounts a lenslet array in front of the sensor (as shown in Fig. 3.2 (a)) to acquire a light field at a 360×360 (upsampled to 1080×1080) spatial resolution and 10×10 angular resolution. The Raytrix R11 camera can produce a higher spatial resolution at the cost of lower angular resolution. The multi-view nature of the light field has enabled new generations of stereo matching [60] and object segmentation algorithms [111]. In this paper, we explore how to conduct salient object detection using a light field camera.

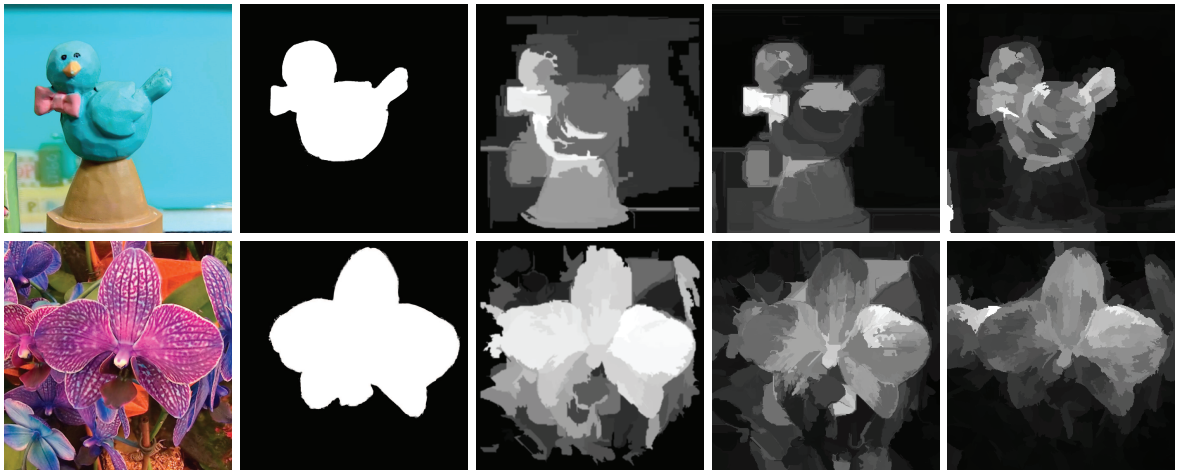


Figure 3.1: Light field vs. traditional saliency detection. Similar foreground and background or complex background imposes challenges on state-of-the-art algorithms (e.g., RC [20], DRFI [52]). Using light field as inputs, our saliency detection scheme is able to robustly handle these cases.

Human vision system has the refocusing ability which can help us pay more attention to the interesting objects, since the other objects are blurred when our eye focus on certain object [47]. Due to above reason, we can easily distinguish the interesting object, *i.e.*, the salient object, regardless the texture or the color of other objects in the scene, *i.e.*, the background. When it comes to detect the salient object from images, several problems will be arisen if the objects in the image have similar color or texture appearance, as shown in Fig. 5.1.

Conceptually, the light field data can benefit saliency detection in a number of ways. First, the light field has a unique capability of post-capture refocusing [87], *i.e.*, it can synthesize a stack of images focusing at different depths. As shown in Fig. 3.2 (b), we can always find right layers which focus on the salient object within focus stack. If we can pick out the right layers which only focus on foreground, the salient object detection problem will be equal to the focus measures algorithms. The availability of a focal stack is inline with the recently proposed "focusness" metric [55]. It is the reciprocal of blurriness and can be estimated in terms of edge scales via scale-space analysis. Second, a light field provides an approximation to scene depth and occlusions. In saliency detection, even a moderately

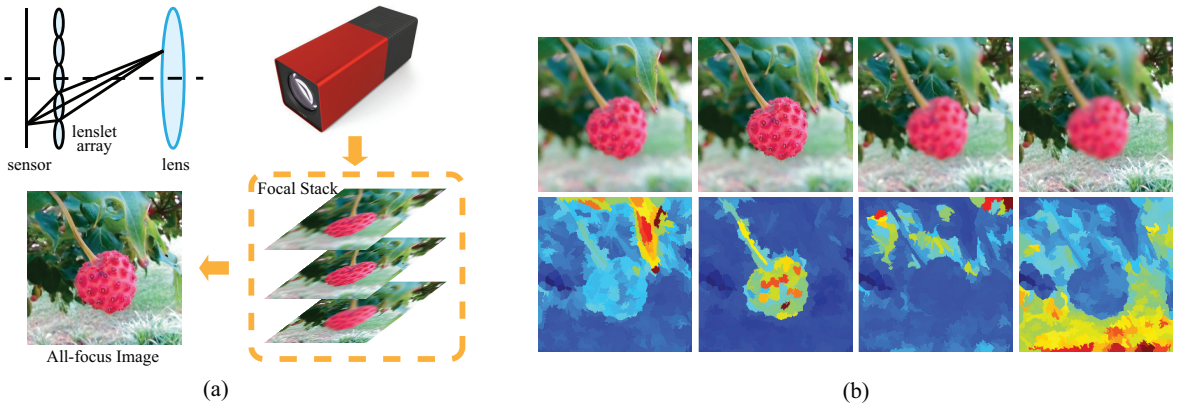


Figure 3.2: (a) A Lytro light field camera can capture a light field towards the scene in a single shot. The results can be then used to synthesize a focal stack and further a all-focus image. (b) Focus stack(the first row) and its corresponding focus regions (second row).

accurate depth map can greatly help distinguish the foreground from the background. This is also inline with the "objectness" [55], *i.e.*, a salient region should complete objects instead of cutting them into pieces.

In addition to focusness and objectness, we also exploit the recent background prior [112]. Instead of directly detecting salient regions, such algorithms aim to first find the background and then use it to prune non-salient objects. Robust background detection, however, is challenging, especially when the foreground and background have similar appearance or the background is cluttered. To resolve this problem, we utilize the *focusness and objectness* to more reliably choose the background and select the foreground saliency candidates. Specifically, we compute a *foreground likelihood score (FLS)* and a *background likelihood score (BLS)* by measuring the focusness of pixels/regions. We select the layer with the highest BLS as the background and use it to estimate the background regions. In addition, we choose regions with a high FLS as candidate salient objects. Finally, we conduct contrast-based saliency detection on the all-focus image and combine its estimation with the detected foreground saliency candidates.

For validation, we acquire a light field database of a range of indoor and outdoor scenes and generate the ground truth saliency map. We have already shared this database, *i.e.*,

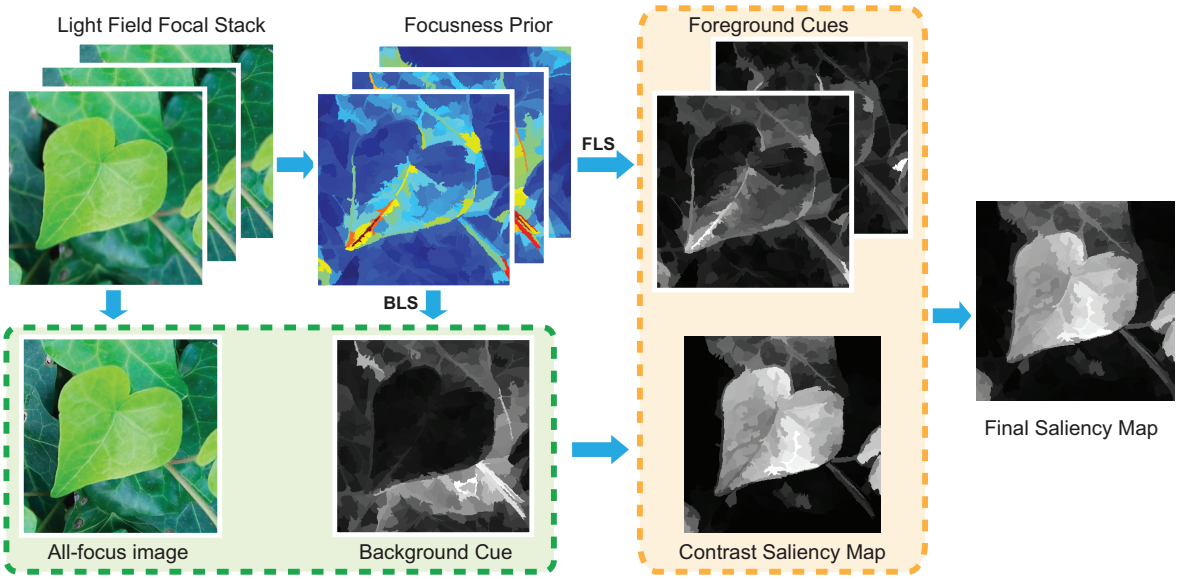


Figure 3.3: Processing pipeline of our saliency detection algorithm for light fields.

Light Field Saliency Detection (LFSD) Dataset, to community online[1]. Experiments show that our saliency detection scheme can robustly handle challenging scenarios such as similar foreground and background, cluttered background, and images with multiple depth layers and with heavy occlusions, *etc*., and achieve high accuracy and robustness. In addition, the comparison results show that our focusness cues using light field are more effective than or equally as good as other state-of-arts depth cues.

Recent studies on human perception [67] have shown that depth cue plays a important role in determining salient regions. However, only a handful of works incorporate depth maps into saliency models. Maki *et al*. [84] used depth cue to detect human motions. Their depth features are highly task-dependent and the detection is performed in a top-down fashion. Niu *et al*. [88] computed saliency based on the global disparity contrast in a pair of stereo images. Lang *et al*. [67] used a Kinect sensor to capture the scene depth. Ciptadi *et al*. [23] used 3D layouts and shape features from depth maps. Peng *et al*. [92] detected saliency taking account of both depth and appearance cues derived from low-level feature contrast, mid-level region grouping and high-level priors enhancement. In this chapter, we

[1] http://www.eecis.udel.edu/~nianyi/LFSD.htm

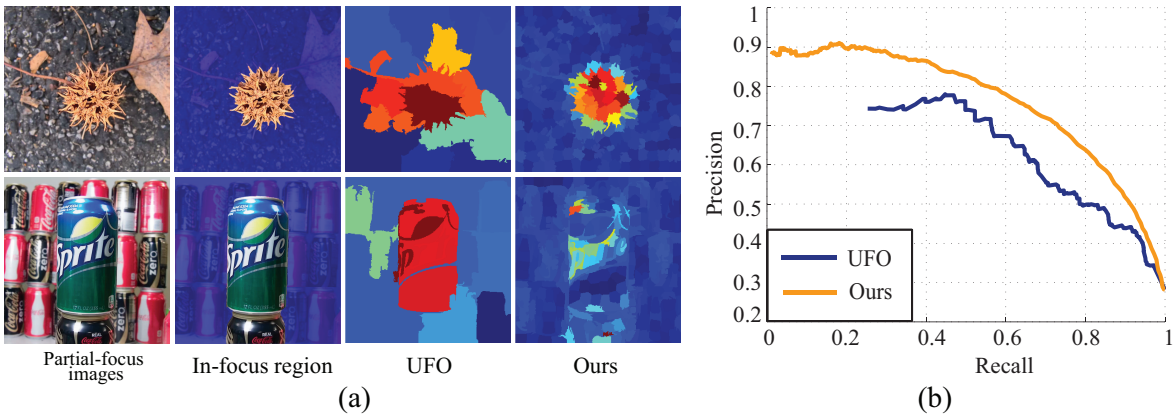


Figure 3.4: Foucsness detection comparison of UFO[55] vs. ours. (a) Focusness detection results comparsion. (b) PRCs comparison.

exploit rich depth information embedded in the light field. Specifically, we use coarse depth information embedded in a focal stack to guide saliency detection. To achieve more accurate result, most depth cue based schemes need relatively accurate depth maps. In reality, depth estimation from images (*e.g.*, stereo) remains challenging in both computational cost and accuracy on real scene. One can alternatively resort to active sensing (*e.g.*, structured light or time-of-flight). However, such schemes also have their limitations such as limited depth range and interference with environment lighting. Focus stack rendering, on the other hand, is more intuitive and precise. Also, isolating different objects by depth map is more likely to break object into pieces, as we have no prior depth information of each object. Our proposed scheme aims to resemble human perception using eye: the eyes can dynamically refocus at different slices to determine saliency. This can be done by constructing focus stack using light field rendering approach. Detecting on focus stack, on the other hand, is more likely to preserve better objectness of salient object than depth map, if salient objects have narrow depth range compared with the depth range of complete scene. More detailed discussion can be found in Section 3.2.4.

3.2 Computing Light Field Saliency Cues

Fig. 4.2 shows our saliency detection approach using the light field. We first generate a focal stack and an all-focus image through light field rendering. For each image in the

focal stack, we detect the in-focus regions and use them as the focusness measure. Next, we combine the focusness measure with the location prior to extract the background and the foreground salient candidates. We further couple the background prior with contrast-based saliency detection for detecting saliency candidates in the all-focus image. Finally, we use the objectness as weights for combining the saliency candidates from the all-focus image and from the focal stack as the final saliency map.

3.2.1 Focal Stack and All-Focus Images

A unique capability of light field is after-capture refocusing. Here we briefly reiterate its mechanism. A light field stores regularly sampled views looking towards the scene on a 2D sampling plane. These views form a 4D ray database and new views can be synthesized by querying existing rays. Given the light field of a scene, one can synthesize a Depth-of-Field (DoF) effects by selecting appropriate rays from the views and blending them, as shown in Fig. 3.2 (a). Isaksen *et al*. [43] proposed to render DoF by reparameterizing the rays onto the focal plane and blending them via a wide aperture filter. Ng *et al*. [87] proposed a similar technique in the Fourier space and the solution has been adopted in the Lytro light field camera. Using the focal stack, we can fuse an all-focus image, *e.g*., through photomontage [3]. We refer the readers to the comprehensive survey on light field imaging [70, 118] for more details about the refocusing algorithm.

In this paper, we use the Lytro camera as the main imaging device to acquire the light field. The Lytro camera uses an array of 360×360 microlenses mounted on an 11 megapixel sensor, where each microlens resembles a pinhole camera. It can produce the refocused results at a resolution of 360×360.

We compose an all-focus image by focus fusion using existing online-tools [2] from the focal stack so that the all-focus image has the same resolution as the focal stack. In addition, it is worth noting that DoF effect is not significant in Lytro focal stack due to small microlens baseline. As a result, each slice is just slightly defocused. Therefore, brute-force

[2] http://code.behnam.es/python-lfp-reader/

approaches such as applying saliency detection on each slice and then combine the results are not directly applicable since all slices will produce similar results.

Before proceeding, we explain our notation. We denote $\{I^i\}$, $i = 1, ..., N$ as the focal stack synthesized from the light field and I^* the all-focus image by fusing the focused regions of $\{I^i\}$. Our goal is to compute a saliency map w.r.t. I^*. We segment each slice $\{I^i\}$ and I^* into a set of small non-overlapping regions (superpixels) using the mean-shift algorithm [24]. This segmentation helps to preserve edge consistency and maintain proper granularity. We use (x, y) index a pixel and r to index to a region.

3.2.2 Focusness Measure

We start with detecting the in-focus regions in each focal stack image I^i and use them as the focusness prior. In the recent focusness-based saliency detection work, Jiang *et al*. [55] measured focusness via edge sharpness. However, edge-based in-focus detection is only reliable when the out-of-focus regions appear severely blurred. In our case, the DoF of Lytro is not as shallow as the one in DSLR. Therefore, edges in out-of-focus regions are not as blurred as in the classical datasets, as shown in Fig. 3.4 (a). It is hence difficult to use spatial algorithms to separate the in-focus/out-of-focus regions. Our approach is to analyze the image statistics in the frequency domain. In Fig. 3.4, we compare the saliency detection resutls vs. the focusness measure both visually and quantitatively. Specifically, we select 80 focus slices that have a clear boundary between in-focus and defocused regions. We then segment out the in-focus region. Fig. 3.4 illustrates that the in-focus regions are often quite different from the actual saliency maps. It is also worth noting that [55] attempts to segment the complete in-focus object whereas our algorithm handels the focusness measures at region level. Consequently, [55] is more likely to over-segement in-focus regions, *i.e*., it will cut into part of the out-of-focus regions, as shown in Fig. 3.4 (a). Our method, on the other hand, processes superpixels and prevents over-segmentation.

Given an $n \times n$ image I, we first transform I into frequency domain by the Discrete Cosine Transform (DCT)

$$\mathcal{D}(u,v) = \sum_{x=0}^{n-1}\sum_{y=0}^{n-1} \cos(\frac{\pi u}{2n}(2x+1)) \cos(\frac{\pi v}{2n}(2y+1)) I(x,y). \tag{3.1}$$

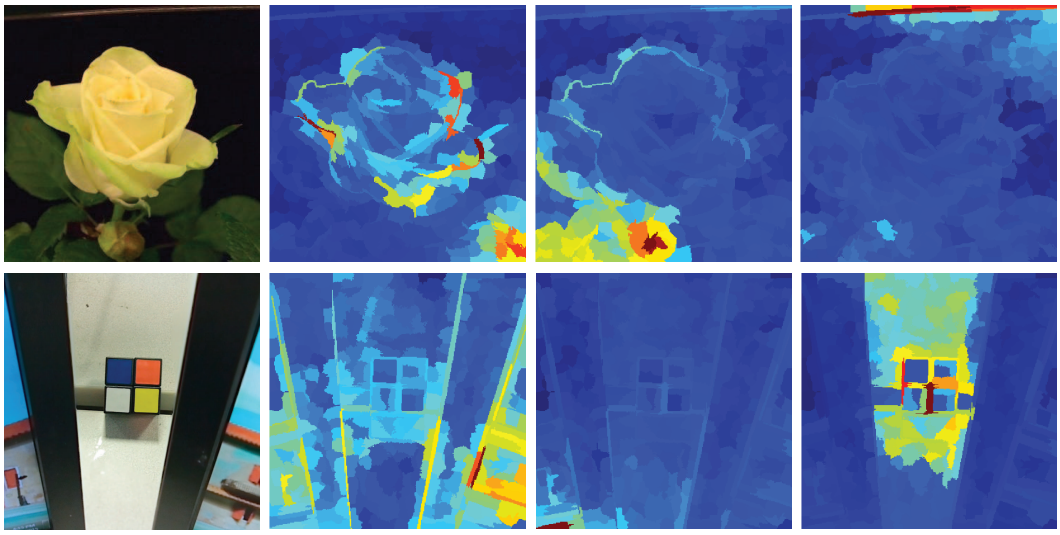
Next, we compute the image's response with respect to different frequency components. We first apply a series of M bandpass filters $\{P_m\}$, $m = 1, ..., M$ on $\mathcal{D}(u,v)$ for decomposing the signal and then transform the decomposed results back via the inverse DCT. Recall that out-of-focus blurs will remove certain high frequency components. Therefore, only regions with a sharp focus will have high responses at all frequencies. In our implementation, we use a sliding window of 8×8 pixels and compute the variance τ_m within each patch with respect to filter P_m. To ensure reliable focusness measurements, we use the harmonic variance [72] to measure the overall variance over all M filters:

$$\mathcal{F}(x,y) = \left[\frac{1}{M-1} \sum_{m=1}^{M} \frac{1}{\tau_m^2(x,y)} \right]^{-1}. \tag{3.2}$$

We use $\mathcal{F}(x,y)$ as the focusness measure at pixel (x,y). Under this formulation, only when the response of all filters are high, the harmonic variance $\mathcal{F}(x,y)$ will be high. Any small τ_m will result in low $\mathcal{F}$. Therefore, this formulation ensures that only local windows preserving all frequency components would be deemed as in-focus. Since both DCT and harmonic variance computations are effective, we compute $\mathcal{F}$ for every pixel in the image. Finally, to measure the focusness of a region, we simply compute the average of all pixels within a region r

$$\mathcal{F}(r) = \sum_{(x,y)\in r} \frac{\mathcal{F}(x,y)}{A_r}, \tag{3.3}$$

where A_r is the total number of pixels in r. We will use this region-based focusness prior $\mathcal{F}(r)$ for selecting background and saliency candidates in Section 3.2.3 and 3.2.4. It is worth noting that more sophisticated focusness estimation techniques such as scanning through the focal volume can be used. In practice, our measure is sufficient for the task of saliency detection and is much faster. Notice that harmonic variance would fail at detecting regions with single-directed edges or with homogeneous color, as can be seen from Fig. 3.4. However,



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

Figure 3.5: Foucsness detection result on focus stack. (a) All focus image. (b) Focusness map on the nearest objects. (c) Focuseness map on objects at the middle of depth range. (c) Focusness map on the furthest objects.

like the blue bottle case in Fig. 3.14, the wrongly suppressed regions could be correctly highlighted as salient region in the final saliency map by incorporating color contrast cue. More details are discussed in Section 3.3.2.

3.2.3 Background Selection

Next, we set out to find the background slice. Notice that the background slice is *not* equivalent to the farthest slice in the focal stack. Recall that we synthesize the focal stack without any knowledge on scene depth range. Therefore, the farthest slice may not contain anything in focus and hence provides little cues, as shown in the first row of Fig. 3.5. Second, the slice that have the farthest object in focus does not necessarily translate to the background slice, like what the second example in Fig. 3.5 shows, the object may be isolated from majority of the background and should be treated as an outlier.

Our approach is to analyze both the distribution of the in-focus objects with respect to their locations in the image: if the majority of in-focus objects (pixels) lies near the border of the image, then they are more likely to belong to the background. Further, if the corresponding depth layer is far away, its in-focus objects are also more likely to be background.

We therefore scan through all focal slices. For each slice I^i, we integrate (project) the focusness measure $\mathcal{F}$ of all pixels along the x and y axes respectively to form two 1D focusness distributions as

$$D_x = \frac{1}{\alpha}\sum_{y=1}^{h}\mathcal{F}(x,y), \quad D_y = \frac{1}{\alpha}\sum_{x=1}^{w}\mathcal{F}(x,y), \tag{3.4}$$

where w and h are the width and height of the image and $\alpha = \sum_x \sum_y \mathcal{F}(x,y)$ is the normalization factor.

A common assumption in saliency detection is that an salient object is more likely to lie at the central area surrounded by the background [112]. If a focal slice corresponds to the background, its D_x and D_y should be high near the endpoints but low in the middle. To quantitatively measure it, we define a "U-shaped" 1D band suppression filter

$$\mathcal{U}(x,w) = \left(\frac{1}{\sqrt{1+(x/\eta)^2}} + \frac{1}{\sqrt{1+((w-x)/\eta)^2}}\right), \tag{3.5}$$

where η controls the suppression bandwidth in $\mathcal{U}$ depending on the image size/resolution, *i.e.*, a high resolution image should have a high η. The Lytro focal stack images have a uniform resolution of 360×360 and we use $\eta = 47$ in all experiments.

Finally, we scale the focusness distribution by the suppression filter to compute a Background Likelihood Score (BLS) for each focal slice I^i

$$BLS(I^i) = \rho \cdot \left[\sum_{x=1}^{w} D_x^i(x)\cdot\mathcal{U}(x,w) + \sum_{y=1}^{h} D_y^i(y)\cdot\mathcal{U}(y,h)\right], \tag{3.6}$$

where $\rho = \exp(\frac{\lambda \cdot i}{N})$ is the weighting factor of layer i in terms of depth, N is the total number of slices in the focus stack and $\lambda = 0.3$. We choose the slice with the highest BLS as the background slice I^B. It is important to note that each focal slice has a corresponding BLS even though it is not chosen as I^B.

3.2.4 Objectness and Foreground Measures

Alexe *et al.* [4] suggested that a salient object should be complete instead of being broken into pieces and refer to this property as the objectness. Given a focal stack image I^i,

we measure the objectness of its focused region using a 1D gaussian filter with mean μ and variance σ as

$$\mathcal{G}(x) = \exp(-\frac{x-\mu}{2\sigma^2}), \tag{3.7}$$

where μ corresponds to the centroid of the object and σ as its size. Recall that we have already computed the focusness distributions D_x or D_y. Therefore, we can directly obtain $\mu = x_p$ or y_p, that corresponds to the peak location of D_x or D_y respectively. If multiple peaks exist, we simply take their average.

Next we estimate σ as the size of the object. If σ is too small, isolated small super-pixels would be treated as an object. If σ is too large, *i.e.*, it would treat the entire image as an object. In our implementation, we choose $\sigma = 45$, *i.e.* 50% Gaussian covers half of the D_x or D_y. We compute the objectness score (OS) for each focal slice

$$OS(I^i) = \sum_{x=1}^{w} D_x^i(x) \cdot \mathcal{G}(x, w) + \sum_{y=1}^{h} D_y^i(y) \cdot \mathcal{G}(y, h). \tag{3.8}$$

Conceptually, if an object in a given slice is salient, it should have a low BLS and high OS, indicating it belongs to the foreground. We therefore define a foreground likelihood score (FLS) as

$$FLS(I^i) = OS(I^i) \cdot (1 - BLS(I^i)). \tag{3.9}$$

Same as how we select the background slice I^B, we choose the foreground slices $\{I^F\}$ as one with the higher FLS ($FLS > 0.7 \times max(FLS)$). Fig. 3.6 illustrates the process of finding the background and foreground slices on a sample image. Notice that salienct object can be separated into several layers, which might result in inaccurate FLS/BLS score for some focusness layers. For instance, the first layer $\mathcal{F}^1$ in Fig. 3.6 focuses on the salient object, but our algorithm regarded it more likely to be background layer. We would like to point out that not all slices focusing on foreground are good choices for $\{I^F\}$. In the $\mathcal{F}^1$ case, even though its highlighted regions belong to salient object, they are scattered around image boundary whereas our goal is to detect salient object as a whole. In reality, saliency objects have narrow depth range in regard to the depth range of the complete scene. This indicates that within a focal stack, there generally exists a slice where the entire salient object/region

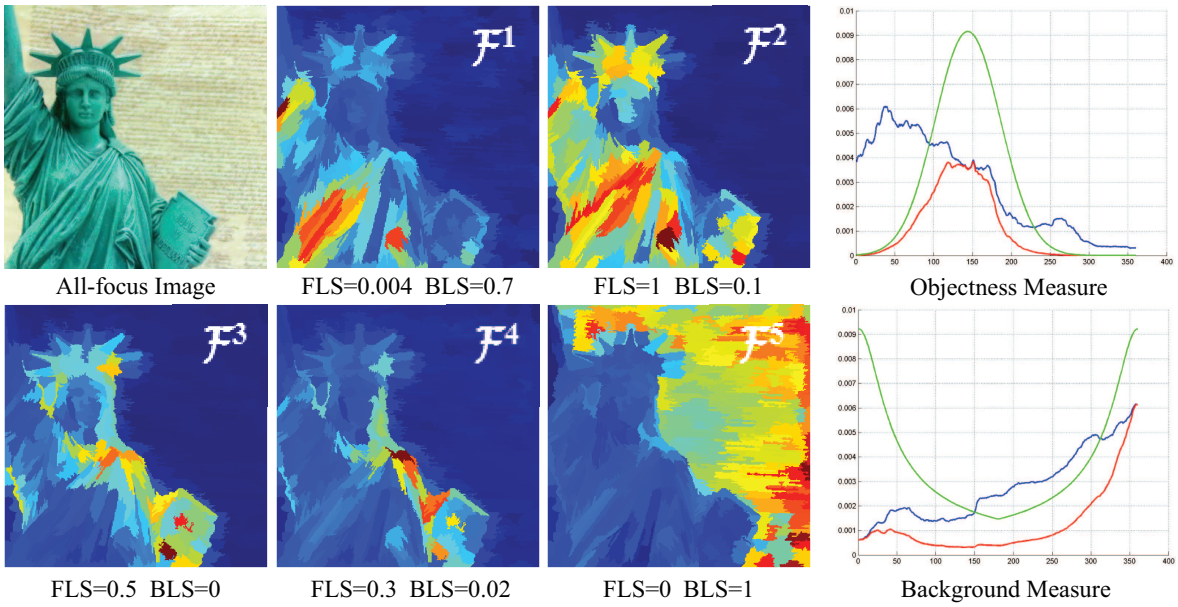


Figure 3.6: Separating the foreground and background using focusness cues. Left: the computed foreground likelihood score (FLS) and the background likelihood score (BLS) computed on different focal slices. Right: Examples on computing objectness measure (up) and background measure (bottom). Green curve is corresponding filter (U-shape or Gaussian); blue curve is sample D_x/D_y; red curve is the scaled distribution by the filter.

exhibits high sharpness, such as the second layer $\mathcal{F}^2$ in Fig. 3.6. Once we are able to select the correct candidate slices, *i.e.*, slices with high FLS/BLS value, the inclusion of incorrect FLS/BLS will not greatly affect the final saliency result. In fact, inaccurate FLS will affect saliency detection only when the salient object has a large depth range in the complete scene.

3.3 Saliency Detection

Finally, we combine the cues obtained from the light field focal stack to detect saliency in the all-focus image I^*.

3.3.1 Location Cues.

We first locate the background regions in I^* using the focusness measure $\mathcal{F}^B(r)$ of the estimated background slice I^B. To incorporate the location prior [101], we scale the

focusness measure for each region R_r in terms of its distance to the center of the image and use it as a new background cue

$$BC(r) = \frac{1}{\gamma}[\mathcal{F}^B(r) \cdot ||\mathbf{p}_r - \mathbf{c}||^2], \tag{3.10}$$

where γ is a normalization factor, $\mathbf{p}_r$ is the centroid of r and $\mathbf{c}$ is image center. We further threshold the BC for determining the background regions $\{B_{r'}\}$, $r' = 1, ..., K$ in I^* (where K is the total number of background regions). We can then compute the Location cue as:

$$LC(r) = \exp(-\beta \cdot BC(r)). \tag{3.11}$$

In our experiment, we use $\beta = 8$.

3.3.2 Contrast Cues.

Once we obtain the background regions, we apply the color-contrast based saliency detection on the non-background region. For each non-background region r and background region r' in I^*, we calculate their color difference $\delta(r, r')$ w.r.t. r' as $\delta(r, r') = \max\{|red(r) - red(r')|^2, |green(r) - green(r')|^2, |blue(r) - blue(r')|^2\}$. To improve robustness, we use compute the harmonic variance of all $\delta(r, r')$ for r

$$HV(r) = \left[\frac{1}{K}\sum_{r'=1}^{K}\frac{1}{\delta(r, r')}\right]^{-1}. \tag{3.12}$$

Combining the harmonic variance of color difference HV with location cue LC, we obtain a color contrast based saliency map as

$$S_C(r) = HV(r) \cdot LC(r). \tag{3.13}$$

3.3.3 Foreground Cues.

From the detected foreground salient candidates $\{I_j^F\}$, $j = 1, ..., L$ via focusness analysis (where L is the total number of foreground slices), we compute the foreground cues the combining the focusness maps $\mathcal{F}_j^F(r)$ and the location cue LC:

$$S_F^j(r) = \mathcal{F}_j^F(r) \cdot LC(r). \tag{3.14}$$

3.3.4 Combine.

Finally, We use the objectness measure as weight for combining the contrast based salience map $S_C(r)$ and foreground maps $S_F^j(r)$ as:

$$S(r) = \sum_{j=1}^{L} \omega_j \cdot S_F^j(r) + \omega_C \cdot S_C(r), \tag{3.15}$$

where $\omega_j = OS(S_F^j)$ and $\omega_C = OS(S_C)$ are the objectness weights calculated by Eqn. 3.8.

3.4 Experiments

Recall that most previous approaches use a single image as input whereas our approach uses the light fields. Since a light field captures much richer information of the scene than a single image, our comparisons do not intend to show that our technique outperforms the state-of-the-art as any such comparisons would be unfair. Rather, our goal is to show that the additional information provided by the light field can greatly improve saliency detection tasks.

3.4.1 Dataset

Traditional benchmark data sets [80, 1] are all single images and cannot be used to test our solution. Most online light field datasets, on the other hand, are not suitable for the purpose of saliency detection. For example, several datasets are either too simple: they only contain a single foreground object again a plain background, or too complex: foreground too cluttered. Further, most light field datasets are captured by large baseline light field cameras, to enhance the DoF effect in refocusing. Consequently, the rendered focus stacks are more likely to break salient objects into smalls pieces, which would impact the final saliency map as we discussed in Section 3.2.4.

We therefore first collect a dataset of 100 light fields using the Lytro light field camera. The dataset consists of 60 indoor scenes and 40 outdoor scenes.

For each data, we ask three individuals to manually segment the saliency regions from the all-focus image. The results are deemed ground truth only when all three results are consistent (*i.e.*, they have an overlap of over 90%).

3.4.2 Evaluations on different Superpixel Algorithms

We first evaluate the impact of superpixel algorithms on our scheme. We compare the most widely used two superpixel-generating algorithms in saliency detection, *i.e.*, Mean-Shift Clustering (MS)[24] and simple linear iterative clustering (SLIC)[2]. The rest parameters were kept the same. It is worth noting that MS would generate more regions than SLIC if they have same original superpixel number N. Consequently, we set the N of SLIC and MS to 300 and 200 respectively.

To quantitatively compare different methods, we use the canonical precision-recall curve (PRC) to evaluate the similarity between the detected saliency maps and the ground truth. Precision corresponds to the percentage of salient pixels that are correctly assigned and recall refers to the fraction of detected salient region w.r.t. the ground truth saliency. Fig. 3.8 shows the PRC comparison result on our light field dataset. Our experiment follows the settings in [21], *i.e.*, we binarize the saliency map at each possible threshold within $[0, 255]$. Fig. 3.9 is a visual comparison between the saliency maps of different schemes. We can see that the saliency results adopting SLIC (LFS_SLIC) resemble MS (LFS_MS) whereas SLIC is about 2.5 times faster, as validated in Fig. 3.8 and Fig. 3.10.

3.4.3 Evaluations on Regular Images

Next, we show our light field saliency detection results and the results using a range of unsupervised schemes on regular images. These include algorithms based on spatiotemporal-cues (LC[119]), graph-based saliency (GB [50]), frequency-tuning (FT [1]), spectral residual (SS [40]), global-contrast (HC [21] and RC [20]), Low Rank Matrix Recovery (LRMR [101]), Graph-Based Manifold Ranking (GBMR [117]), focusness-based (UFO [55]), Hierarchical Saliency (HS [116]) and Discriminative Regional Feature Integration (DRFI [52]). Most these methods have open source code and we use the default parameter.

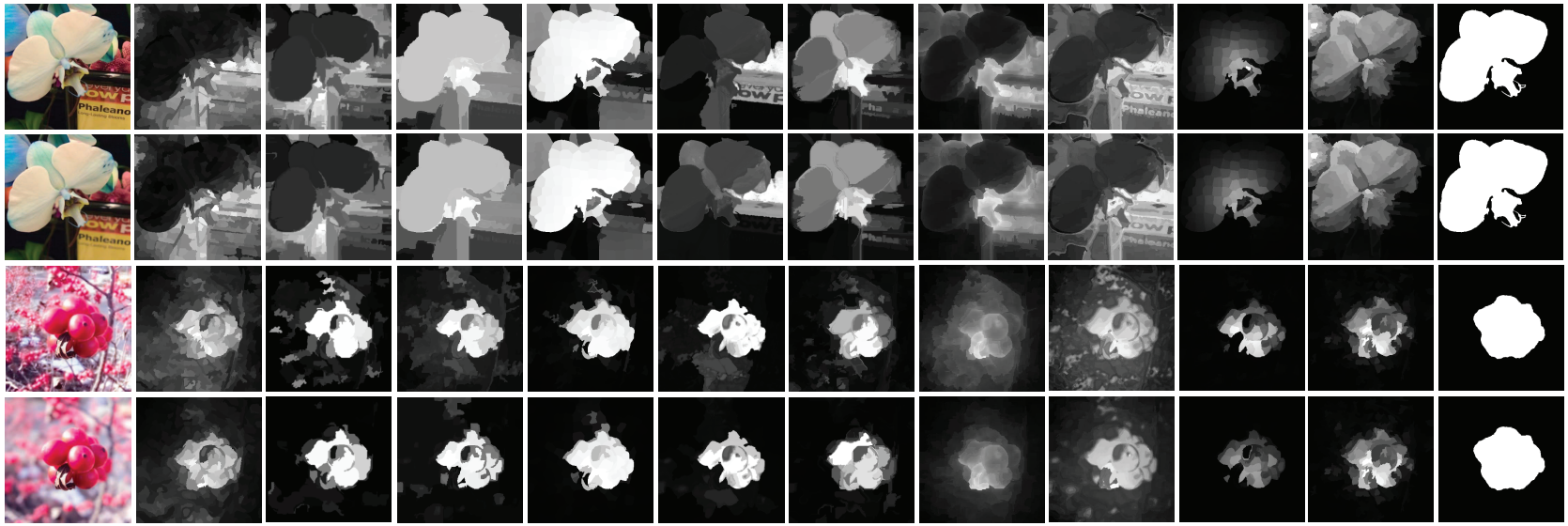
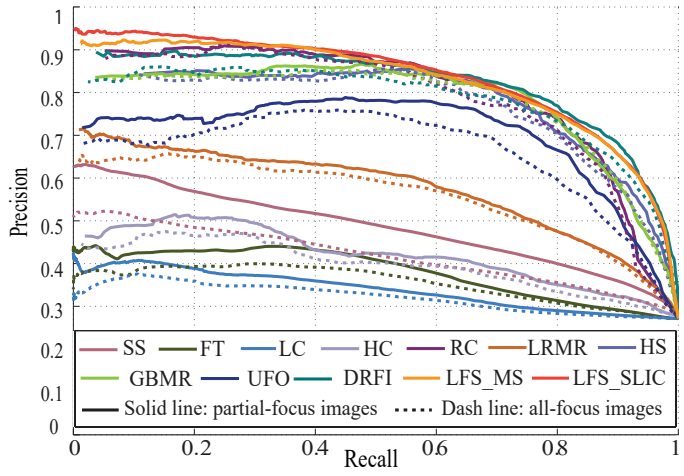
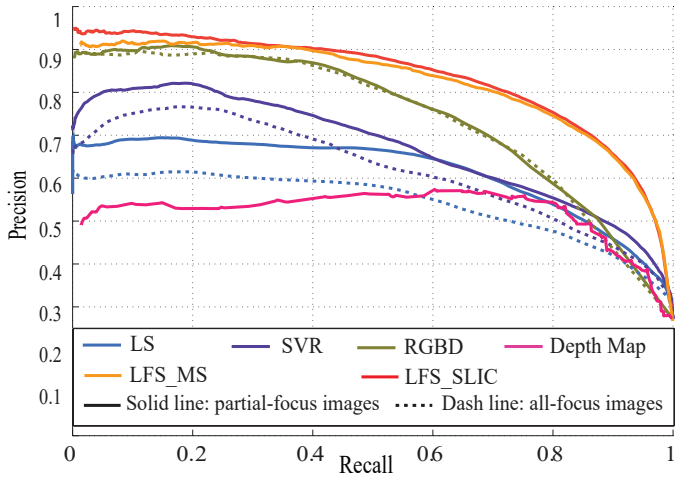


Figure 3.7: Saliency results using all-focus images (the first and third rows) and partial-focus images (the second and forth rows)

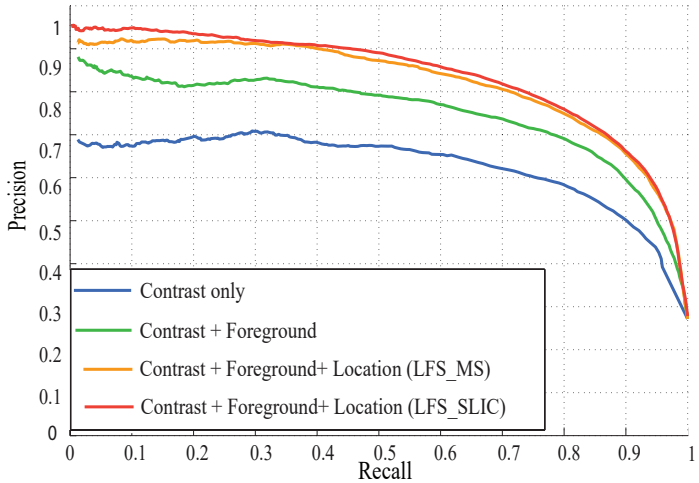
We first evaluate the performance of above methods on all-focus images. In Fig. 3.9, we show the saliency detection results for visual comparisons. For very challenging scenes such as the blue bird (second row) , our approach produces much better results than regular image based techniques. It is important to note that all-focus image will degrade the sharpness contrast between salient object and background, which would impact the performance of algorithms based on sharpness/focusness cues. To ensure fairness, we then compare the performance on partial-focus images, *i.e.*, the image layer focusing at a fixed depth layer. If there are several layers that focus on the same foreground object, we simply pick out the one that produces the sharpest image of the salient object. Fig. 3.8 (a) provides the PRCs comparison. In Fig. 3.7, we show a visual comparison between the resulting saliency maps of various single-image based state-of-the-art schemes. We observe that only in cases where a partial-focus image exhibits a severely defocused background, partial-focus slice would produce better performance than an all-focus image, as shown in Fig. 3.7 (the blue flower scene vs. the fruit scene). Notice though that the results using the complete light field still outperforms the ones using either the best partial slice or the all-focus image. This illustrates the significant advantage of using the light field as inputs for saliency detection.



(a)



(b)



(c)

Figure 3.8: PRC comparisons on our light field dataset. (a)Results of regular image based algorithms. (b) Results of depthmap based algorithms.(c) Using different cues in our approach.

Figure 3.9: Visual Comparisons of different saliency detection algorithms vs. ours on our light field dataset.

We would like to point it out that the PRCs are less smooth than they appear in traditional saliency works. This is due to the small amount of data in our dataset (100 light field sets vs. 1000 images in classical benchmarks), although the curves still provide useful insights on the performance. Also note that a large number of scenes in our light field dataset is highly challenging to previous techniques, *i.e.*, many have complex background or similar foreground and background. Fig. 3.9 shows sample in-focus images of these difficult scenes. We observe that recently proposed RC [20], HS [116] and DRFI [52] can still achieve reasonable performance. This is partially due to the background prior refinement and color space smoothing methods used in RC, the multi-scale features used in HS and the supervised feature vector mapping approach used in DRFI. Results using our technique produces the highest precision in the entire recall range. This illustrates the importance of focusness and objectness prior provided by the light field.

Fig. 3.10 evaluates the running time of each methods. We implemented all methods with open source code and list their average running time for one scene. Notice that even though our algorithm needs processing much larger data (about 10 times) than others, the average processing time is still comparable to those regular-image-based techniques using the same programming platform.

Method	FT	LC	HC	SS	RC	LRMR	HS	GBMR	DFRI	Ours_MS	Ours_SLIC
Time (ms)	8.11	3.30	231.16	102.59	918.91	12629.7	393.69	674.91	9104.18	9830.5	3755.1
Code	C++	C++	C++	Matlab	C++	Matlab	C++	Matlab&C++	Matlab	Matlab	Matlab& C++

Figure 3.10: Comparison of average time taken for different saliency detection methods.

3.4.4 Evaluations on Images with Depth information

We further choose three recent proposed depthmap-based methods, *i.e.* SVR[26], LS[23] and RGBD[92], to compare their performances with our model. The depth maps of LFSD are generated directly from Lytro desktop. Fig. 3.8 (b) shows the PRC comparisons among above mentioned algorithms, which illustrates that the focusness cues utilized in our technique are equally or more useful than depth cue. Fig. 3.9 shows the visual comparisons

on several LFSD images. We observe that LS and SVR may produce low precision results, since they treat depth cues independently for saliency detection, while ignoring the strong complementarity between appearance and depth cues and utilize depth cues as an independent image channel for saliency detection. It is important to note that directly using depth maps as saliency cues is not reliable. For example, simple thresholding on the depth maps would produce large errors on images in row 3, 4, and 6 of Fig. 3.9 where both salient and non-salient objects lie at the similar depth. In fact, in Fig. 3.8 (b), we have plotted the PRC performance by using merely depth maps as saliency cues and the results show that it is inferior to depth-based approaches.

We also evaluate their performance on both all-focus and partial-focus images. It is noteworthy that all-focus images also degrade the performance of those depthmap-based techniques. This is because that all these three methods incorporate depth saliency with regular saliency models to obtain the final saliency maps. Moreover, the larger the regular saliency features weigh, the more evident improvement will show.

3.4.5 The Effect of Camera-to-Object Distance

Recall that a Lytro camera has small baseline. In order to enlarge the infocusing and defocusing contrast between foreground and background, most of the salient objects in our dataset are placed near the Lytro camera. When the foreground object is faraway from the camera, the change of depth-of-field when switching the focus from the foreground to the background would be less significant. To test whether our algorithm is robust to the camera-to-object distance d, we capture 50 more light fields where salient objects are located at diverse d. Notice that the maximum d making the object notable is proportional to the object size. In our experiment, instead of exploring the connection of performance vs. d, we analyze the relevance of performance vs. objects' depth-to-size ratio $\mathcal{R}$:

$$\mathcal{R} = \frac{d \cdot max(depth\ range)}{Height(Object) \cdot Width(Object)}. \tag{3.16}$$

Typically, the range of $\mathcal{R}$ in our testing set is between $[14, 170]$. To plot the performance vs. $\mathcal{R}$ curve, we divide the 50 light fields into 5 subsets according to their $\mathcal{R}$ value.

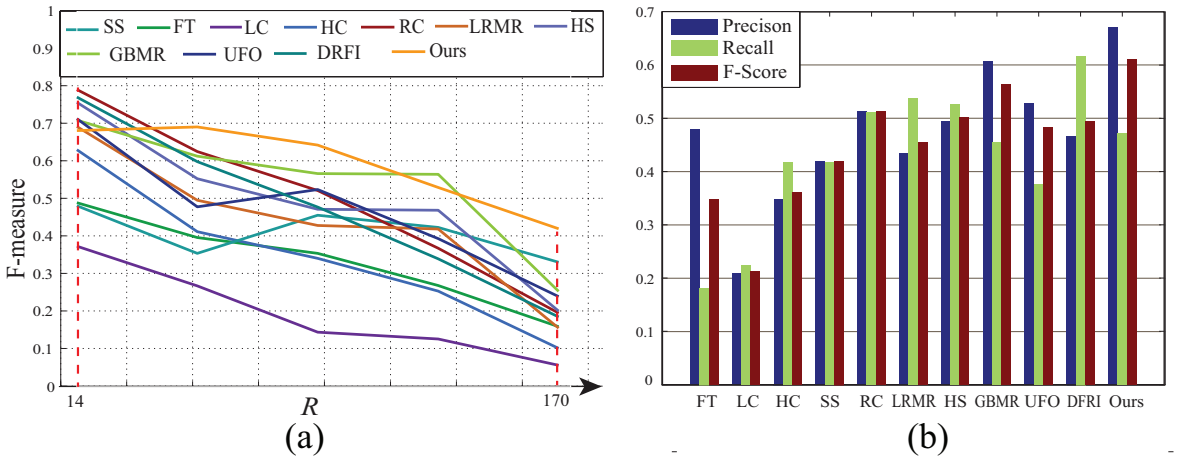


Figure 3.11: (a) Performance comparisons of F-score regarding $\mathcal{R}$. (b) Average precision, recall and F-score on 50 testing light fields.

Each set contains about 10 light fields. Here, we adopt the F-score methodology:

$$F_\beta = \frac{(1+\beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}. \tag{3.17}$$

It is concluded in [80] that precision rate is more essential than recall in attention detection. Accordingly, we choose $\beta^2 = 0.3$ to weigh precision more than recall. For each light field set, we calculate its average precision and recall rate. The average F_β is derived by Eqn. 3.17. Fig. 3.11 (a) shows the $F_\beta - \mathcal{R}$ curves of different methods and Fig. 3.11 (b) presents the comparison of average precision, recall and F-score on this 50 light fields set. Fig. 3.12 provides the visual comparison of different methods when changing d. We can tell that as $\mathcal{R}$ goes larger, the performance of most algorithms decrease. This is because that when the object lies far away from the camera, in the slice where the object is in-focus, the background appears nearly focused as well. Such a scenario resembles the classical all-focus saliency detection case where the usefulness of most focusness cue is reduced. Due to the effectiveness of our focusness detection algorithm, our method shows the best robustness in $F_\beta - \mathcal{R}$ curves and also achieves the highest average F-score.

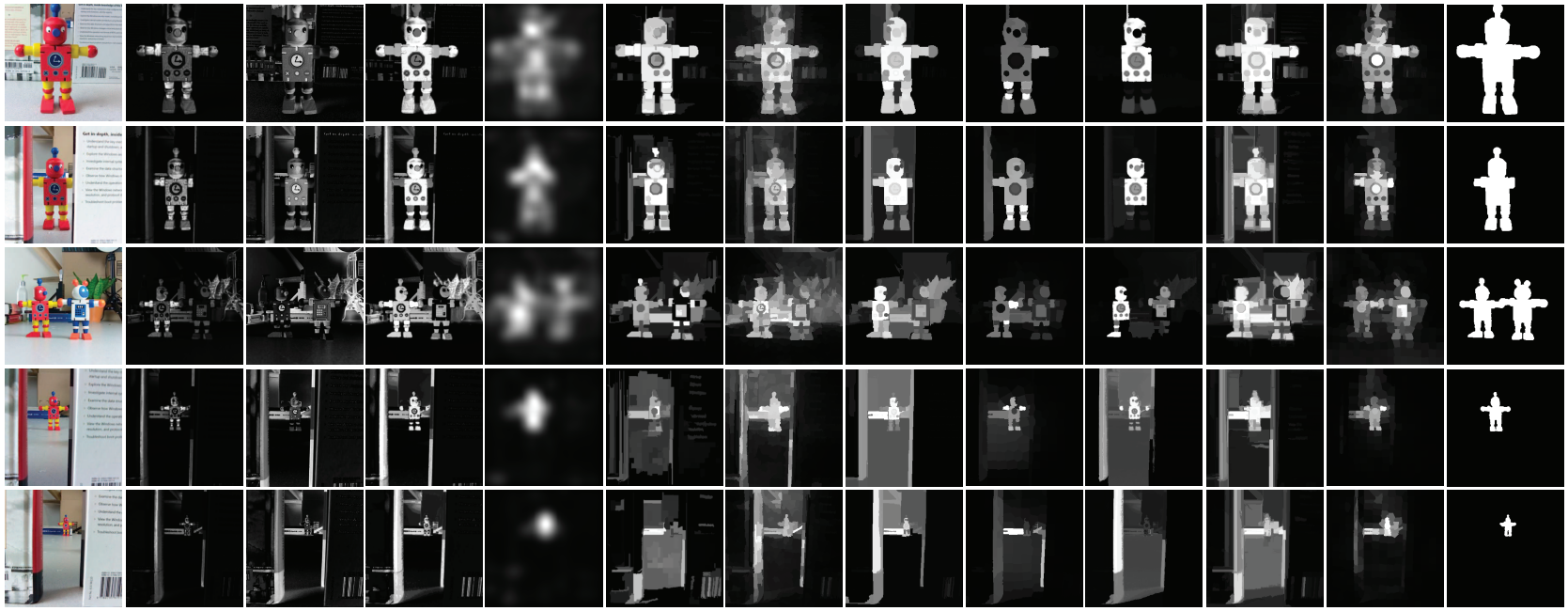


Figure 3.12: Saliency maps of red robot at different $\mathcal{R}$. From top to down, $\mathcal{R} = 14, 53, 92, 131, 170$

3.4.6 The Effect of Parameters

For all the experiments described above, the parameters were kept fixed, *i.e.*, no user fine-tuning was done. To test the robustness of our algorithm to the parameters and to analyze their effect, we repeated the experiments, while varying η from Eqn. 3.5, σ from Eqn. 3.7, N (the number of superpixels), β from Eqn. 3.11, and λ from Eqn. 3.6. To quantitatively show these impact, we follow the F-score methodologies described in Section 3.4.5 to evaluate the accuracy of the detected saliency when varying η, σ , N, β and λ, as shown in Fig. 3.13.

Parameter η controls U-shape filter while parameter σ controls the shape of Gaussian. Since we have normalized both U-shape and Gaussian filter to a very small scale, those two parameters causes barely modifications when changing values. N denotes the number of superpxiels. It can be observed that our algorithm is very robust as well, due to the performance is dominantly affected by the objectness of superpixels and less by the number of superpixels. Certainly when the number of superpixels is too small, the salient and non-salient regions will be merged and the performance of our approach will be inferior. It is noticeable that the above three parameters are varying among different ranges. η should vary between 0 and $min(w, h)$, where w and h are the width and height of the image. Recall that the Gaussian filter has relative large response between $[\mu-\sigma, \mu+\sigma]$, during which the salient objects should be located. Therefore, we normally keep σ between 0 and $min(w, h)$. As for N, we range it from 20 to 1000.

Parameter β effects the highlight extend of regions located at the center of images and parameters λ controls the probability of picking the back layer as background slice. β and λ of exponential functions, on the other hand, have much smaller ranges, *i.e.*, [0,100], to prevent from out of memory issues. Unlike η, σ and N, changing the values of β and λ has specific impact on our final results, even though slightly.

Fig. 3.13 (b) reveals that when β goes large, the performance of our approach will degrade. Notice that low recall occurs when highlighted regions in saliency maps are all of high value. In our case, the larger β is, the higher values will be assigned to the central regions by the location cue, *i.e.*, LC. F-score, therefore, will be decreased consequently. When λ is large (> 2), the performance will degrade slightly (about 0.03). This is because

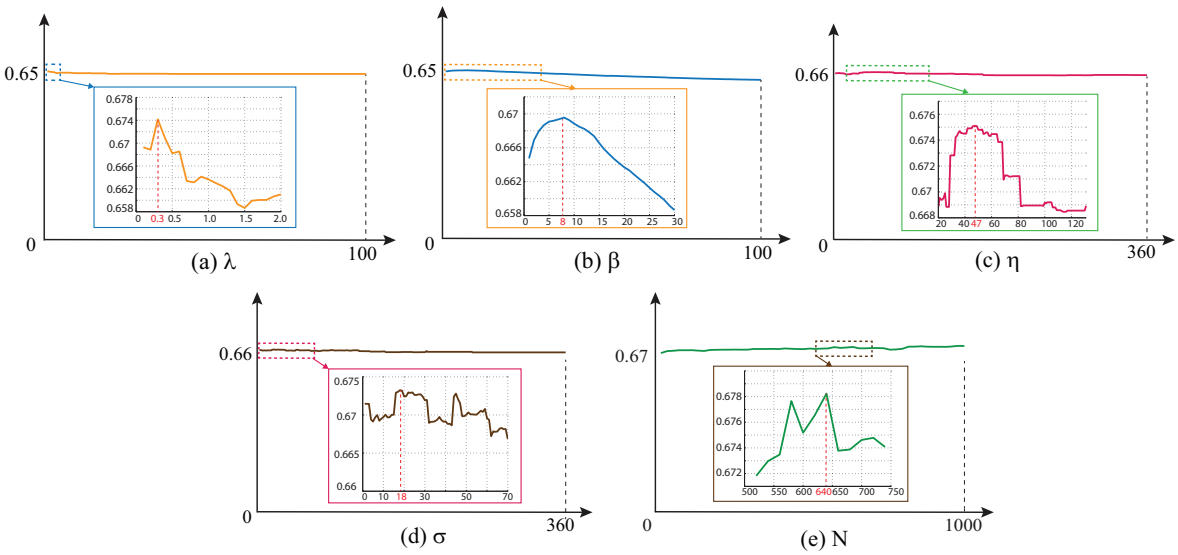
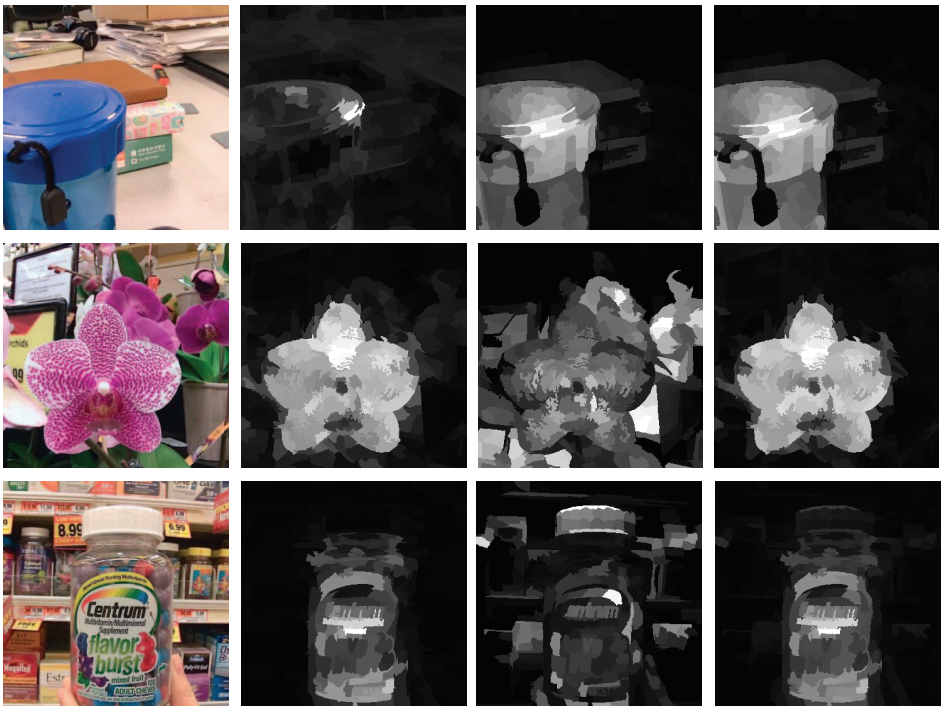


Figure 3.13: F-score curvers when varying λ, β, η, σ, N.

that large λ value will enforce picking the furthest layer as background slice, which may fail at cases we discussed in Section 3.2.3.

We also compare the saliency components obtained using different cues, *i.e.*, color contrast, location and focusness cues. Fig. 3.8(c) shows the PRC comparisons using individual vs. combined cues. The plot illustrates that each cue has its unique contribution to saliency detection, although in some cases, an image can be dominated by a specific cue as shown in Fig. 3.14. In the first row, color contrast provides most valuable cues and the estimated saliency from it resembles the final one. This is mainly because the blue mug lacks texture and hence is not robustly detected as the foreground object to provide focusness cues. In contrast, in the flower scene in the second row, the color contrast result treats both the foreground flower and the background clutter as saliency. The focusness cue, however, manages to correct the errors by removing the background. In the last example, the color contrast result misses the foreground bottle and the focusness cue manages to add it back.



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

Figure 3.14: Saliency detection using different cues. (a) All-focus images; (b) Detected saliency using focusness cues; (c) Detected saliency using color contrast. (d) Saliency results by combining (b) and (c).

3.4.7 Limitations.

The performance of our algorithm is largely dependent on the quality of the acquired light field. Lytro, however, has a much narrow Field-of-View than regular cameras. Therefore, objects in our light fields generally appear "bigger" than in other benchmarks. With emerging interest on light field camera designs, we expect next-generation models to overcome this limitation. There are also alternative approaches to use the light field for saliency detection. For example, one can potentially first construct a depth map using stereo matching. However, the quality of stereo matching depends largely on scene composition. Nevertheless, even a low quality depth map may provide useful cues comparable to the focusness cue. Furthermore, it is also possible to first conduct saliency detection on the all-focus image and then use the results to improve the quality and speed of light field stereo matching.

3.5 Discussion

We have presented a saliency detection algorithm tailored for light fields. We believe this is the first light field saliency detection scheme. The key advantage of using a light field instead of a single image is that it provides both focusness and depth cues. In recent works [88, 55], these new cues have shown great success in improving accuracy and robustness in saliency detection. Our solution echoes these observations and also provides an alternative and more robust method to extract these cues through the analysis of light fields. Experiments show that our technique can handle many challenging scenarios that cast problems on traditional single-image-based algorithms. Another contribution of our work is the construction of the light field saliency dataset which consists of the raw light field data, the synthesized focal stacks and all-focus images, and the ground truth saliency maps. Our immediate future work is to build a much larger and comprehensive database and share it with the community.

Chapter 4

A WEIGHTED SPARSE CODING FRAMEWORK FOR SALIENCY DETECTION

In this chapter, I provide a unified saliency detection framework for handling heterogenous types of input data.

4.1 Motivation

Existing 2D saliency algorithms are inherently different from how human visual system detects saliency. Human eyes have two unique properties that are largely missing in existing 2D saliency solutions. First, human eye can conduct dynamic refocusing that enables rapid sweeping over different depth layers. Hence, for humans, the input is a focal stack instead of a single, fixed-focus or all-focus image as has been used in traditional approaches. Second, human uses two eyes to infer scene depth, e.g., via stereo, for more reliable saliency detection whereas most existing approaches assume that the depth information is largely unknown.

Recently there has been an emerging interest in emulating these the properties of human eyes. For example, light field saliency uses the Lytro camera as the acquisition apparatus and then synthesize a focal stack via light field rendering [71]. The focusness cues are then extracted from the focal stack and integrated with color, location, and contrast cues. Preliminary results seem promising although the image resolution is generally low due to tradeoff between spatial-angular sampling. Several schemes have been proposed to incorporate stereo vision. Niu et al.[88] employed the disparity maps to better extract better foreground/background separations. Lang et al.[67] used the Kinect sensor to acquire scene depth and integrate the results with regular 2D saliency via a Gaussian mixture model. Despite their effectiveness, saliency detection algorithms based on 2D, 3D and 4D data have adopted completely different frameworks. In particular, the features used for distinguishing

Figure 4.1: Our method vs. the latest feature-matrix-based DSR algorithm [77] on different data inputs. From top to bottom: we show results on 2D images, 3D stereo data, and 4D light field data.

saliency candidates and more importantly the procedures for utilizing them differ significantly.

In this chapter, we present a universal saliency detection framework for handling heterogenous types of input data. We set out to build saliency/non-saliency dictionaries using data-specific features. Specifically, we first select a group of potential foreground superpixels to build the saliency dictionary. We then prune the outliers and test on the remaining superpixels to iteratively refine the dictionaries. A major advantage of our technique is that it provides a universal framework for all different types. The only variation to the algorithm is input features: for 2D images, we use color, texture and focusness characteristics; for stereo data, we add depth/disparity cues; and for the 4D light field data, we add focusness cues on focus stack. Comprehensive experiments on a broad range of datasets (MSRA-1000 [80] and SOD [86] for 2D, SSB [88] for 3D, and the light field saliency dataset[75] for 4D) show that our technique outperforms state-of-the-art solutions.

The literature of saliency detection is huge and we only discuss the most relevant ones. For a comprehensive survey state-of-the-art algorithms, we refer the readers to [11].

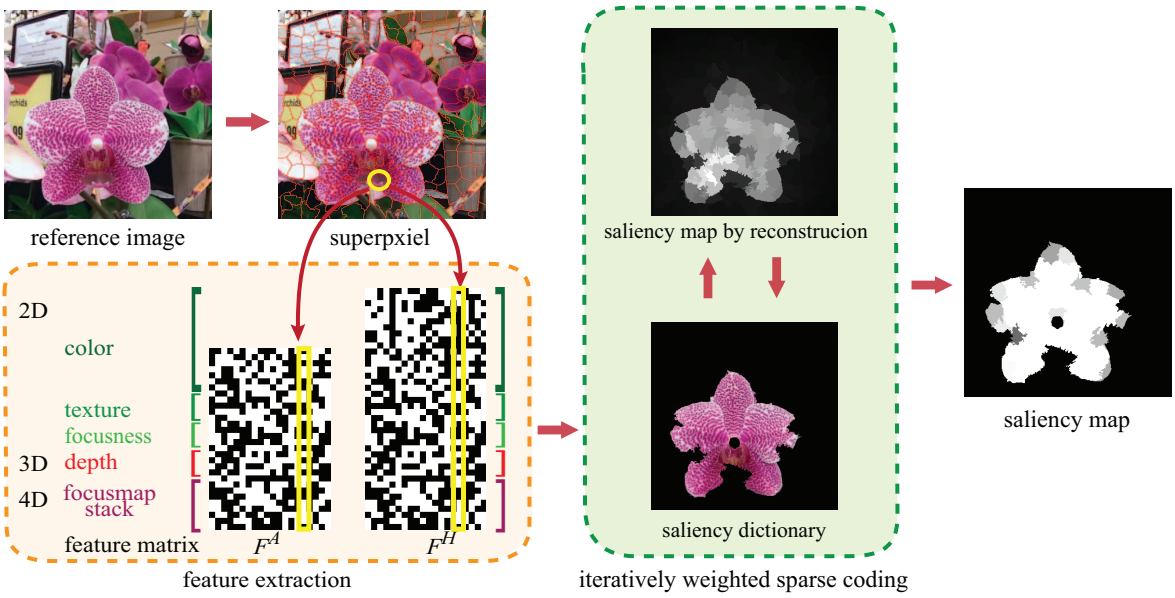


Figure 4.2: Processing pipeline of our dictionary-based saliency detection algorithm.

4.2 Feature Selection

Our approach is based on building saliency/non-saliency dictionaries and our approach is generic to 2D, 3D and 4D datasets. The dictionaries are built for superpixels. Regarding different segmentation schemes, we use the widely adopted simple linear iterative clustering (SLIC) algorithm [2] for its high efficiency, compared with other schemes, *e.g.*, mean-shift. We use SLIC to segment the reference image I into a set of small non-overlapping regions/superpixel $R = \{r_1, r_2, ...r_N\}$. For stereo pair data, the reference image refers to the one used for generate the disparity map. For light field data, the reference is the all-focus image. We use p to index pixel and r to superpixel. The ultimate goal is to assignment each superpixel r a saliency value $Sal(r)$.

4.2.1 Feature Extraction

For each pixel, we set out to associate with a feature vector. A good feature descriptor should exhibit high contrast between saliency objects and background.

2D feature.

Color is the most intuitive feature to distinguish two regions. As shown in[10], coupling RGB and Lab color spaces improves the accuracy of saliency maps. Here, we choose

both RGB and Lab color spaces as color descriptors. For texture, Gabor filters have been shown as an effective measure [31]. When using Gabor filters as orientation and scale tunable edge detectors, we can characterize the intrinsic texture information using the statistics of microfeatures within the superpixel. We use the Gabor filter responses with 12 orientations and 3 scales as texture descriptors.

For focusness, we utilize the mean distance to its 8-neighbors in the RGB space:

$$\sigma_f(p) = \frac{1}{8}\sum_{m=1}^{8}\delta_m(p, p_m) \tag{4.1}$$

where $\delta(p, p_m) = \|p^{rgb} - p_m^{rgb}\|_2^2$ and p^{rgb} is the color vector of p in RGB color space.

3D feature.

3D data further provides depth/disparity information for each points in the scene. In [88], disparity is used as a unique feature to distinguish objects from background. When disparity/depth is available, we directly append it to the features vector.

4D feature.

For 4D light field, we can further synthesize a focal stack. We use the in-focus measure at each focal slice to derive an additional light field feature descriptor. For instance, if a focus stack has L different focus slices, we calculate L focusness values $\sigma_f^l(p), l = 1, 2...L$ by applying Eqn. 4.1 on each slice. After appending them to the feature vector, we get the stacked vector $\boldsymbol{f_p} = [\sigma_1\sigma_2...\sigma_C]^T$ of p.

4.2.2 Feature Matrix

From the feature vectors of all pixels, we generate two feature matrices for all superpixels.

Averaging.

The simplest approach to convert per-pixel feature vector to per-superpixel feature vector is through averaging [113, 77]. We use the $C \times N$ matix F^A to represent the result feature matrix. Notice that F^A is expected to perform well if the scene is composed of objects with simple color and textures but will be less robust if the foreground and background

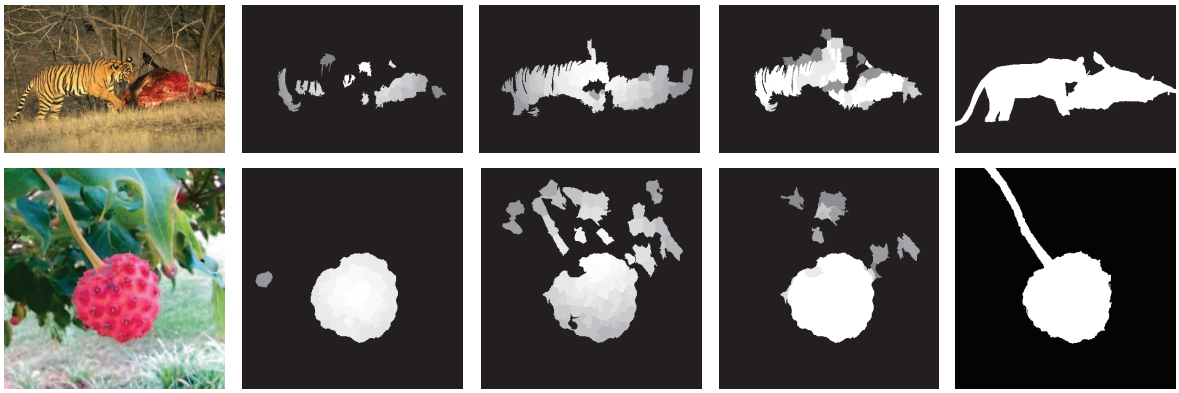


Figure 4.3: Saliency detection results using our approach on individual and combined feature matrices.

contain highly complex textures, as shown in Fig. 4.3. This is because that averaging over all pixels loses information that characterizes color variations within each superpixel.

Color Histogram.

To handle textures, our second scheme computes the histogram over three color channels. Specifically, we treat color in terms of ratios $\{\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}\}$ and compute the histogram of the two channels (the third is dependent of the other two). Specifically, we use the R and G channel and we discretize the two channels into 32×32 bins for computing the histogram. Consequently the color components of the feature vector for a superpixel becomes $\{\sigma_1^{r_i}, \sigma_2^{r_i} ... \sigma_{1024}^{r_i}\}$. The other feature components such as focusness and depth remain the same as the averaging scheme. We use the $C' \times N$ matix F^H to represent the resulting feature matrix. Notice that F^H is suitable for handling scenarios where the scene contains highly textured objects. However, it is fragile for the textureless cases, which is because that color histogram will introduce inner-region noises when images consist of smooth foreground and background, as shown in Fig. 4.3. Notice that the two schemes are complementary to each other and we can apply our saliency detection scheme (Section 4.3) on each matrix and combine the results. Fig. 4.4 (a) shows some sample results using individual matrices and their combined result.

4.3 Dictionary Based Saliency Detection

From F^A and F^H, we develop a sparse coding framework: saliency superpixels correspond to the ones that yield to low/high reconstruction error from the saliency/non-saliency dictionary. Our solution is based on recent studies that show non-saliency regions can be represented by a sparsely coded dictionary [77, 117]. We use the error measure to refine the foreground superpixels and to identify foreground saliency ones.

In classical (unweighted) sparse coding scheme [77], the goal is set to find a sparse code $\boldsymbol{\alpha_i}$ that can achieve the maximum/minimum reconstruction error. The coefficients should encode the saliency value, if the template D denotes the set of K potential non-saliency/saliency regions respectively:

$$\boldsymbol{\alpha_i} = \arg\min_{\boldsymbol{\alpha_i}} \|\boldsymbol{f_i} - \boldsymbol{D\alpha_i}\|_2^2 + \lambda\|\boldsymbol{\alpha_i}\|_1 \tag{4.2}$$

For saliency detection, we adopt the weighted sparse coding scheme [41]:

$$\boldsymbol{\alpha_i} = \arg\min_{\boldsymbol{\alpha_i}} \|\boldsymbol{f_i} - \boldsymbol{D\alpha_i}\|_2^2 + \lambda\|\boldsymbol{diag(\omega_i)} \cdot \boldsymbol{\alpha_i}\|_1 \tag{4.3}$$

where the jth value of ω_i is the penalty for using the jth member in template D to encode f_i and we set λ=0.01 in our implementations.

Notice that large ω_i will suppress nonzero entries α_i and force the solution α to concentrate on indices where ω_i is small. Therefore, the weight (penalty) ω_i for saliency detection should be inversely proportional to the similarity between the feature vector f_i and template members D. In other words, if the f_i is similar to some template in D, the penalty ω_i should be small and vice versa. Fig. 4.4(b) shows that, by adding this penalty weight ω into the framework, the performance of saliency detection is significantly improved.

4.3.1 Weighted Sparse Coding Saliency

Fig. 4.2 shows our framework. Given a set of superpixels $S = \{r_1, r_2 ... r_K\}$, which consists of indices of a certain subset of superpixels, we use their corresponding feature vectors (of superpixels) $A = \{F^A_{r_1}, F^A_{r_2} ... F^A_{r_K}\}$ and $H = \{F^H_{r_1}, F^H_{r_2} ... F^H_{r_K}\}$ to construct two dictionaries.

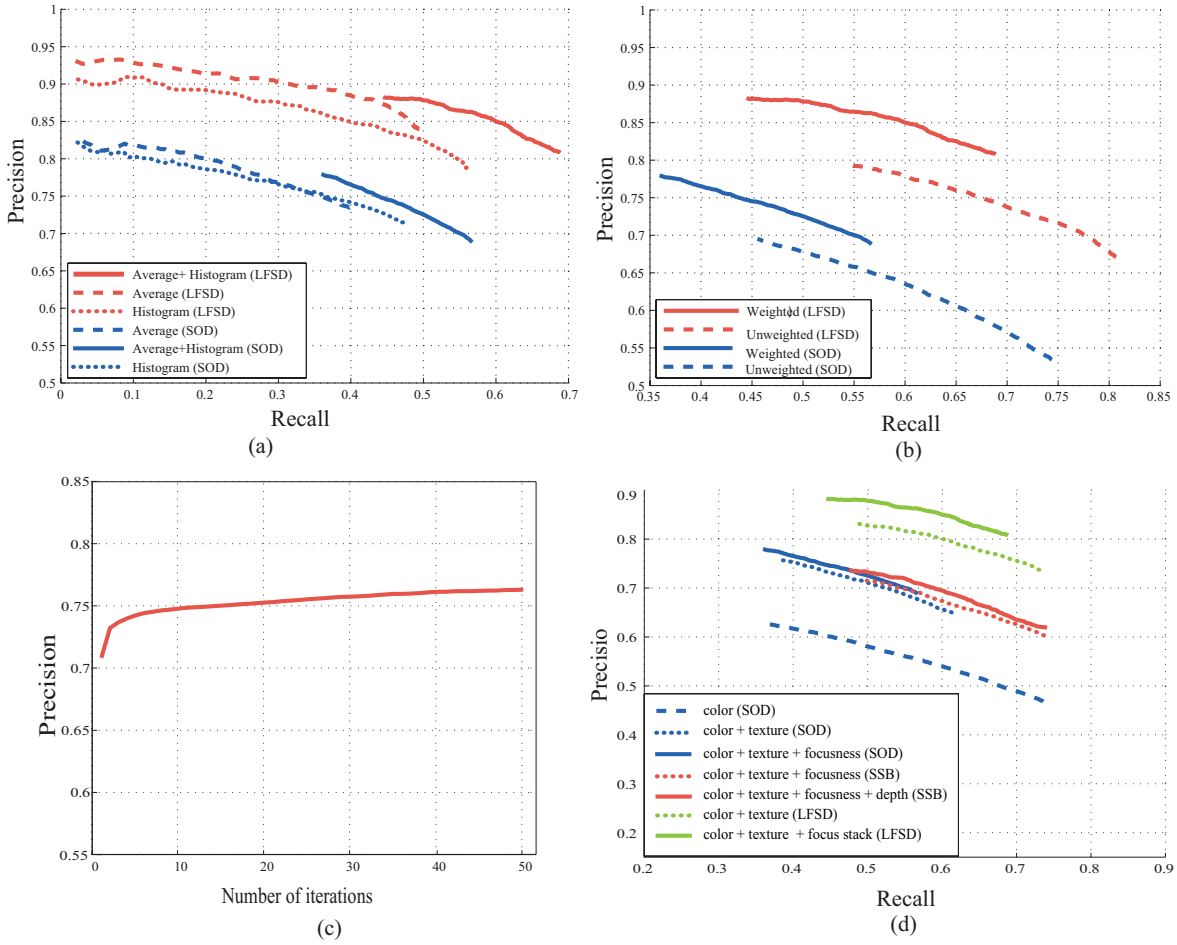


Figure 4.4: (a) PRC comparisons using our weighted approach with individual and combined feature matrices on the SOD and LFSD datasets; (b) PRC comparisons using weighted and unweighted dictionary frameworks on the SOD and LFSD datasets; (c) Precision improvement with more iterations; (d) PRC comparison using different features.

We use $\omega_{r_i}^A$ and $\omega_{r_i}^H$ to represent the weight/penalty for superpixel r_i. Here, the template symbol D can be either A or H, *i.e.*, $D \in \{A, H\}$. $\omega_{r_i}^D$ is a vector that computes the similarity between superpixel r_i (in feature matrix F^D) to all the members in template D:

$$\boldsymbol{\omega}_{\boldsymbol{r_i}}^{\boldsymbol{D}} = [g(r_i, D_1), g(r_i, D_2)...g(r_i, D_K)]^T$$

where $g(r_i, D_j)$ computes the similarity between the superpixel r_i and the jth member of template D:

$$g(r_i, D_j) = e^{\|F_{r_i}^D - D_j\|} \tag{4.4}$$

Next, we use (A, $\omega_{r_i}^A$) and (H, $\omega_{r_i}^H$) as input to Eqn. 4.3 to generate to sparsely coded dictionary $\alpha_{r_i}^A$ and $\alpha_{r_i}^H$ respectively. We then compute the reconstruction error $\epsilon_{r_i}^A$ and $\epsilon_{r_i}^H$ for each r_i:

$$\epsilon_{r_i}^D = \|F_{r_i}^D - D\alpha_{r_i}^D\|_2^2 \tag{4.5}$$

Two saliency value $Sal^A(r_i)$ and $Sal^H(r_i)$ are also computed for r_i:

$$Sal^D(r_i) = Sal^*(\epsilon_{r_i}^D) \cdot Sal^L(r_i) \tag{4.6}$$

where $Sal^L(r_i)$ is the object-bias center prior defined in [77]. $Sal^*(\epsilon_{r_i}^D)$ is the saliency function related to the dictionary's type (saliency or non-saliency). For non-saliency dictionary, it will assign high values to superpixels of a high $\epsilon_{r_i}^D$ value. Similarly, for saliency dictionary, $Sal^*(\epsilon_{r_i}^D)$ will assign high value to superpixels with low $\epsilon_{r_i}^D$.

We define the saliency function for non-saliency dictionary:

$$Sal^*(\epsilon_{r_i}^D) = \epsilon_{r_i}^D \tag{4.7}$$

For saliency dictionary:

$$Sal^*(\epsilon_{r_i}^D) = e^{\beta \cdot \epsilon_{r_i}^D} \tag{4.8}$$

where we set $\beta = -5$ in our implementation.

Finally, we combine $Sal^A(r_i)$ and $Sal^H(r_i)$ to get the saliency value for r_i:

$$Sal(r_i) = Sal^A(r_i) + Sal^H(r_i) \tag{4.9}$$

4.3.2 Dictionary Construction

We define saliency dictionary as a set of superpixels $S = \{r_{s_1}, r_{s_2}, ...r_{s_k}\}$ which are regarded as the potential saliency regions and will be refined through our framework. To get the initial saliency dictionary, we use a non-saliency dictionary to reconstruct the reference image, and patches with high reconstruction error are selected saliency dictionary.

Non-saliency Dictionary.

Non-saliency dictionary is the set of superpixels which are tagged as the non-saliency regions. To obtain it, we first extract two sets of superpixel sets B_1, B_2 where B_1 is the set of superpixels on the reference image boundaries and B_2 is the set of superpixels locate in the out-of-focus regions. For 2D, B_2 correspond to the ones whose focusness response is lower than the average. For 3D data, B_2 correspond to the ones lying far away, i.e., with a small disparity value. For 4D data, we select B_2 by detecting the in-focus regions of the farthest away focal slice in the focal stack. Finally, we combine B_1 and B_2 as the non-saliency set $B = \{B_1, B_2\}$.

To avoid redundancy, we adopt the recently proposed background measure scheme [124]. In [64], similar superpixels are merged into some larger regions $A_m = \{r_1^m, r_2^m, ...r_S^m\}$. A boundary connectivity score, which measures the extent of region A_m connecting to the boundary, is also assigned to each A_m.

$$\omega_{A_m}^{Con} = \frac{K}{\sqrt{Area(A_m)}} \tag{4.10}$$

In our implement, instead of choosing the image boundary to measure the connectivity, we use B to compute the connectivity score $\omega_{A_m}^{Con}$. Superpixels from the merged region have the same connectivity score, namely $\omega_{r_j}^{Con} = \omega_{A_m}^{Con}, r_j \in A_m$. Superpixels whose connectivity scores are non-zero are selected to form non-saliency dictionary.

Saliency Dictionary.

After we obtain the non-saliency dictionary, we use the weighted sparse framework described in Section 4.3.1 to compute a saliency map. For each superpixel r_i, we define the

parameter weight $g(r_i, D_j)$ as:

$$g(r_i, D_j) = e^{\|F_{r_i}^D - D_j\|} + \omega_{r_i}^{Con} \tag{4.11}$$

We choose superpixels whose saliency values are higher than the mean to construct the initial saliency dictionary S^0.

4.3.3 Iterative Refinement

We start with using S^0 as input to the weighted sparse framework. At each iteration, we will refine the saliency dictionary using the estimated saliency map. The algorithm terminates when there is no change to the saliency dictionary. The parameter weight $g(r_i, D_j)$ can be computed using Eqn. 4.4 and the saliency function is computed as Eqn. 4.8.

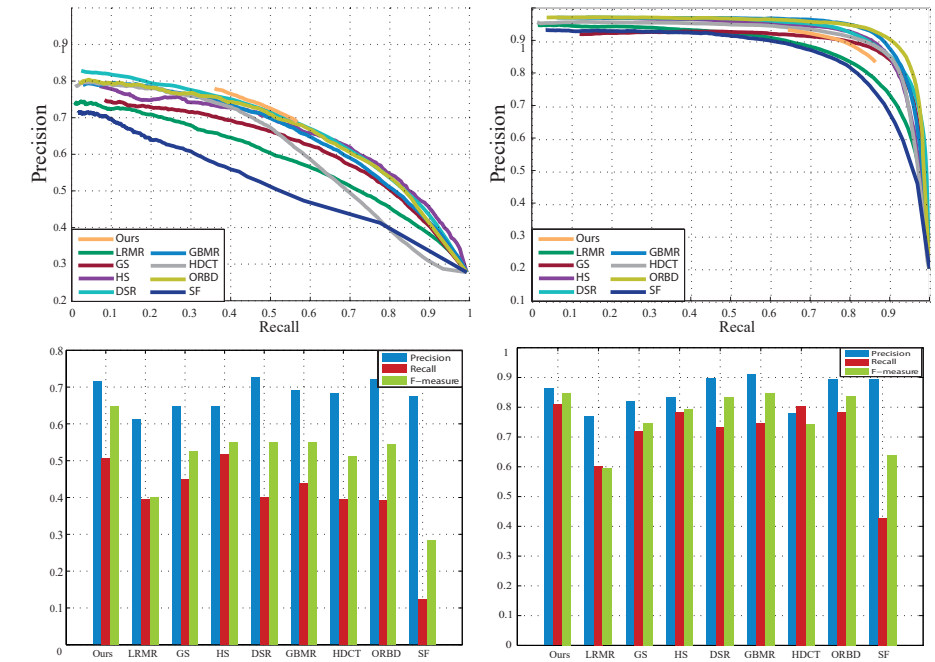
We use superscript to denote iteration number. At the kth iteration, we first classify each superpixels using the saliency dictionary S^k and two template A^k and H^k according to their reconstruction errors (Eqn. 4.3 and Eqn. 4.5). We then compute two saliency maps as Eqn. 4.8. Next, we apply a center cue on two maps to make saliency regions more compact. Finally, we sum the two saliency maps with respect to A and H. A new saliency dictionary S^{k+1} is generated with by using superpixels whose saliency values are higher than the mean. The pseudocode of our iterative refinement is shown in Algorithm 1. Fig. 4.4 (c) illustrates the change of average precision value of SOD dataset with different step lengths in iterative refinement. We can tell that the algorithm converges within 50 iterations.

Notice that we combine two saliency maps to generate the final saliency map, which will cause the ignorable noises on background becoming significant. Hence, we further clamp the low value (¡ 50) to 0.

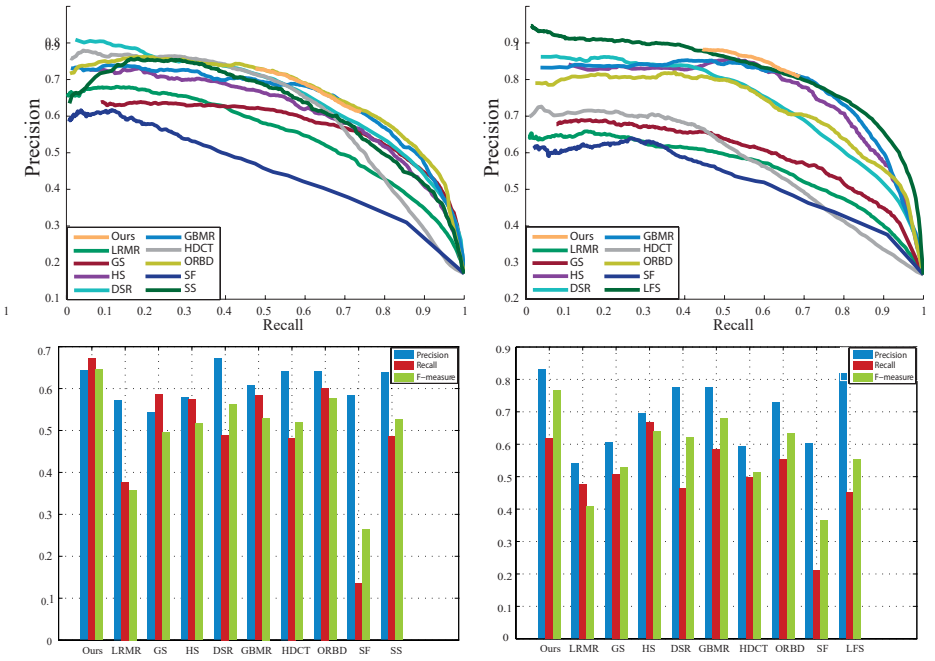
4.4 Experiments

We compare our approach with state-of-the-art techniques tailored for specific dimensional data.

Figure 4.5: Visual Comparisons of different saliency detection algorithms vs. ours on 2D (first two rows: MSRA-1000; last two rows: SOD), 3D and 4D datasets.



(a) SOD (b) MSRA



(c) SSB (d) LFSD

Figure 4.6: Performance comparisons of ours vs. (a) SOD, (b) MSRA, (c) SSB and (d) LFSD. The top row shows the PRC and bottom row uses the bar chart to show the average precision, recall and F-score.(Best view in pdf)

Parameter Setup.

We set the number of superpixels to be 300 in all experiments. Initial backgrounds are extracted from re-clustered segmentation map (re-clustering superpixels) with 2 clustering levels $E = 1$ for feature matrix using original RGB values and $E = 3.5$ for feature matrix for RGB color histogram. E is the matching tolerance value (distance threshold). The reason of choosing a smaller E for feature matrix using original RGB values is that RGB is less representative than color histogram. If E is too high, a superpixel of uniform color and a textural superpixel may be incorrectly merged.

We also test the robustness of our algorithm to the parameters and to analyze their effect. Regarding different superpixels numbers, ranging from 50 to 500, we found that the results are relatively uniform in precision value. We believe it is because the performance is dominantly affected by the choice of the superpixel's feature vectors instead of the number of superpixels. Certainly when the number of superpixels is too small, the salient and non-salient regions will merge and the performance of our approach will ultimatly degrade. Regarding different feature types, we have compared the contribution of individual features to the final performance of our approach on different datasets. We can see from Fig. 4.4 (d) that the addition of the focusness feature better improves 4D light field data than 2D image data. The discrepancies can be attributed to the characteristics of the datasets: 4D light field data provides a more reliable estimation to focusness.

2D databases.

We evaluate the performance of our algorithm vs. DSR [77], GBMR [117], LRMR [113], HS [116], SF [94], GS [112], HDCT[61], ORBD[124] on the MSRA-1000 [80] dataset and the SOD [86] database. MSRA-1000 database contains 1000 images selected from MSRA-5000 with corresponding binary ground truth maps. The SOD database is derived from the Berkeley segmentation database where objects in each image have a consistency score. Objects with high consistency scores are considered salient objects. The SOD database is considered as the most challenge database in saliency detection since the contrast between foreground and background is generally rather small.

3D databases.

The PSU Stereo Saliency Benchmark (SSB) contain 1000 pairs of stereoscopic images and corresponding salient object masks for the left images. All the results are evaluated on the left images of SSB. In addition to the above 2D schemes, we compared our results with SS[88], which is tailored for this dataset. Before running our algorithm, we derive the disparity maps for each left image by SIFT-flow[38]. In order to achieve a more fair comparison, we extend the feature matrix of DSR and LRMR to one more dimension to record the depth information before implementing.

4D databases.

The recently proposed LFSD database contains 100 scenes, where each scenes's light field is recorded by Lytro camera. We compare the results of our algorithm with above 2D methods and LFS[75] (designed for this dataset). For 2D algorithms, we use the all-focus image as input. Again, to avoid unfair comparison, we add the light field features (defined in section 4.2.1) into DRS and LRMR's framework before evaluation.

We follow the canonical precision-recall curve(PRC) and F-measure methodologies to evaluate the accuracy of the detected saliency on databases of different dimension. For details about these two evaluation methods, we refer reader to [45]. The parameters setting in our implement is the same as [21].

Fig. 4.6 shows the result of the two comparison architectures. Experimental results show that the PRC of our unified approach achieves state-of-the-art and the best F-measure in all the databases. It is important to note that our PRC only have values within certain recall range. This is due to the fact that the difference between saliency and non-saliency values assigned by our algorithm is much greater than others. In another word, the saliency maps computed by our algorithms is of the best similarity to ground truth, as shown in Fig. 4.5.

Our approach can handle highly challenging cases such as the blue bird scene in LFSD and the fish scene in SOD where the deemed saliency regions have a similar color/texture to the non-saliency regions. Notice that our recall values are still higher than other methods

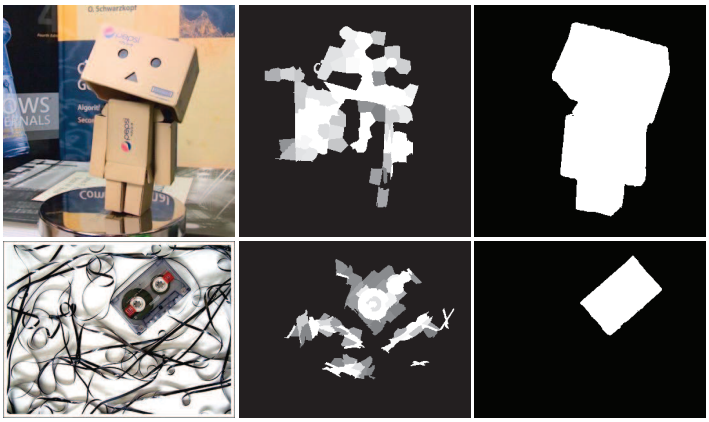


Figure 4.7: Failure cases. Top: our result appears segmented on a 4D light field data due to incorrect focusness estimation. Bottom: our result incurs errors on a 2D image due to high foreground/background similarity.

with favourable precision in most cases. This indicates that our algorithm is capable of locating most saliency regions with a high confidence. Fig. 4.5 shows that our technique also produces more visually pleasing results, e.g., it generates more complete contours and more accurate saliency maps.

4.5 Discussion

We have presented a novel saliency detection algorithm that is applicable to 2D image data, 3D stereo/depth data, and 4D light field data without modifying the processing pipeline. We first develop a data-specific feature vector descriptor. For 2D data, it corresponds to color and textures. For 3D, we append depth information. For 4D, we further append focusness measures. We show that two types of feature descriptors are complimentary to each other for handling variational types of texture/color scene compositions. We have then built a dictionary based framework that constructs saliency and non-saliency dictionaries from the stacked feature vectors. Compared with state-of-art techniques that commonly adopt different solution frameworks for handling different data inputs, our technique does not require modifying the algorithm but only the input descriptor. Comprehensive experiments have shown that it outperforms previous tailored solutions for different data types.

A limitation of our technique is that it does not fully exploit the rich information

embedded in 3D and 4D. By far, we only use the depth value and focusness cues inferred from these data. If they do not provide additional information, our technique falls back to the 2D case, as shown in Fig. 4.7. In the future, we plan to design more effective descriptors, e.g., depth variations and view-dependency features, embedded in 3D and 4D data. Since our approach requires building and refining dictionaries, we also plan to investigate more efficient algorithms to accelerate the process. Finally, we expect other uses of our framework such as tracking and recognition. In particular, there is limited work on using 3D depth and in particular 4D light field data for such tasks. For example, the saliency results can be directly used as inputs to existing tracking or streo matching algorithms, to improve their performance in cluttered scenes.

Algorithm 1 Iterative Refinement

Require: S^0, A^0,H^0,F^A,F^H, $j = 0$
Ensure: Sal

function ITERATIVEOPT(S^0, A^0, H^0, F^A, F^H)
  while $not\ \ converge$ **do**
    for superpixel $r_i = 1 \rightarrow N$ **do**
      $\alpha_{r_i}^{A^j}, \alpha_{r_i}^{H^j} \leftarrow Eqn.\ 4.3$
      $\epsilon_{r_i}^{A^j}, \epsilon_{r_i}^{H^j} \leftarrow Eqn.\ 4.5$
      $Sal^{A^j}(r_i), Sal^{H^j}(r_i) \leftarrow Eqn.\ 4.8$
      $Sal(r_i) \leftarrow Eqn.\ 4.9$
    end for
    $S^{j+1} \leftarrow \{r_i | Sal(r_i) > mean(Sal(r_i))\}$
    $A^{j+1} \leftarrow F^A(S^{j+1})$
    $H^{j+1} \leftarrow F^H(S^{j+1})$
    $j \leftarrow j + 1$
  end while
  $Sal < T \leftarrow 0$
end function

Chapter 5

PERSONALIZED SALIENCY DETECTION

In this section, I present our framework for predicting the fixation maps for individuals. I introduce how we construct the image dataset for personalized saliency prediction and then discuss our multi-task CNN model for computing the discrepancy map between the universal and personalized eye fixation prediction.

5.1 Motivation

By far, nearly all previous approaches have focused on exploring a universal saliency model, i.e., to predict potential salient regions common to users while ignoring their differences in gender, race, age, personality, etc. Such universal solutions are beneficial in the sense they are able to capture all ”potential” saliency regions. Yet they are insufficient in recognizing heterogeneity across individuals. Examples in Fig. 5.1 illustrate that while multiple objects are deemed highly salient within the same image (eg, *human face* (first row), *text* (last tow rows) and object of (*high color contrast*), different individuals have very different fixation preferences when viewing the image. For the rest of the paper, we use term *universal saliency* to describe salient regions that incur high fixations across all subjects and term *personalized saliency* to describe the heterogeneous ones.

In fact, heterogeneity in saliency preference has been widely recognized in psychology: ”Interestingness is highly subjective and there are individuals who did not consider any image interesting in some sequences” [37]. Therefore, once we know a person’s personalized interestingness over each image (personalized saliency), we shall design tailored algorithms to cater to him/her needs. For example, in the application of image retargeting, the texts on the table in the fourth row in Fig. 5.1 should be preserved for observer B and C when resizing the image whereas such texts are less important for observer A. For applications in

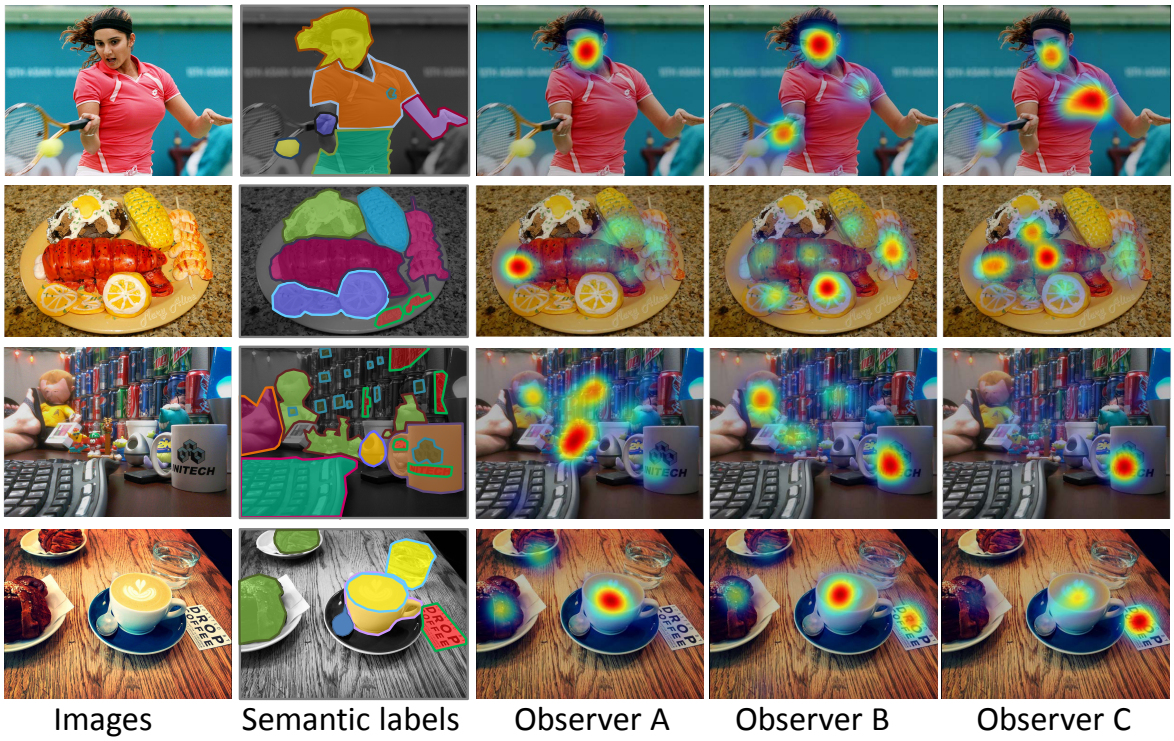


Figure 5.1: An illustration of PSM dataset. Our dataset provides both eye fixations of different subjects and semantic labels. Due to the large amount of objects in our dataset, for each image, we didn't fully segment it and only labelled objects that cover at least three gaze points from each individual. A notable difference between PSM and its predecessors is that each subjects looks 4 times on PSM data to derive solid fixation ground truth maps. Both commonality and distinctiveness exist for PSMs viewed by different participant. This motivates us to model PSM based on USM.

VR/AR, one can design data compression algorithms that personalized salient regions should be less compressed in order to both improve the users' experience and reduce the size of data in transmission. In addition, we can embed characters/logo/advertisement at those personalized salient regions for different individuals. Despite its importance, very little work has been carried out on studying such heterogeneity, partially due to the lack of suitable datasets and experiments. Further, the problem is inherently challenging as saliency variations across individuals are determined by multiple factors, e.g., gender, race, education, *etc*., as well as the content of the image such as the color, location, size and type of objects.

In this paper, we present the first dataset of personalized saliency maps (PSMs) that consists of 1600 images viewed by 20 human subjects. To improve reliability, we ensure that each image is viewed by every subject for 4 times over about one week interval. We use the '*Eyegaze Edge*' eye tracker to track gaze and produce a total of 32,000 ($1,600 \times 20$) fixation maps. To correlate the acquired PSMs and the image contents, we manually segment each image into a collection of objects and semantically label them. Examples in Fig. 5.1 illustrate how fixations vary across three human subjects. Our annotated dataset provides fine-grained semantic analysis for studying saliency variations across individuals. For example, we observed that certain types of objects such as watches, belts would introduce more incongruity (possibly due to gender differences) whereas other types such as faces would lead to more coherent fixation maps, as shown in Table 5.2.

We further present a computational model towards this personalized saliency detection problem. Notice that saliency maps from different individual still share certain commonality via the USM. Hence, we model the PSM as a combination of USM and a residual map which is related to the identity and the image contents. We adopt a multi-task convolutional neural network (CNN) to identify the discrepancy between PSM and USM for each person, as shown in Fig. 5.4.

The contributions of our paper are two-fold: i) To our knowledge, it is the first work that specifically tackles the personalized saliency and we build the first dataset for personalized saliency detection; ii) We present a USM based PSM detection scheme and a multi-task CNN solution to estimate the discrepancy between PSM and USM. Experimental results

demonstrate the effectiveness of our framework.

Tremendous efforts on saliency detection have been focused on predicting universal saliency. For the scope of our work, we only discuss the most relevant ones. We refer the readers to [7] for a comprehensive study on existing universal saliency detection schemes.

5.2 PSM Dataset

We start with constructing a dataset suitable for personalized saliency analysis.

5.2.1 Data Collection

Clearly, the rule of thumb for preparing such a dataset is to choose images that yield distinctive fixation map among different persons. To do so, we first analyze existing datasets. A majority of existing eye fixation datasets provide the one-time gaze tracking results of each individual human subject. Specifically, we can correlate the level of agreement across different observers with respect to the number of object categories in the image. When an image contains few objects, we observe that a subject tends to fix his/her gaze at locations where objects that have specific semantic meanings, e.g., faces, text, signs [57, 114]. These objects indeed attract more attention and hence are deemed more salient. However, when an image consists of multiple objects all with strong saliency as shown in Fig. 5.1, we observe a subject tends to diverge his/her attention. In fact, the subject focuses attention on objects that attract his/her most personally. We therefore deliberately choose 1,600 images with multiple semantic annotations to construct our dataset for PSM purpose. Among them, 1,100 images are chosen from existing saliency detection datasets including SALICON [53], ImageNet [98], iSUN [115], OSIE[114], PASCAL-S [78], 125 images are captured by ourselves, and 375 images are gathered from the Internet.

5.2.2 Ground Truth Annotation

To gather the ground truth, we have recruited 20 student participants (10 males, 10 females, aged between 20 and 24). All participants have normal or corrected-to-normal vision. In our setup, each observer sits about 40 inches in front of a 24-inches LCD monitor

of a 1920×1080 resolution. All images are resized to the same resolution. We conduct all experiments in an empty and semi-dark room, with only one standby assistant. An eye tracker (‘*Eyegaze Edge*’ eye tracker) records their gazes as they view each image for 3 seconds. We partition 1,600 images into 34 sessions each containing 40 to 55 images. Each session lasts about 3 minutes followed by a half minute break. The eye tracker is re-calibrated at the beginning of each session. To ensure the veracity of the fixation map of each individual as well as to remove outliers, we have each image be viewed by each observer 4 times. We then combine the 4 saliency maps of the same image viewed by the same person, and use the result as the ground truth PSM of the observer. To obtain a continuous saliency map of an image from the raw data of eye tracker, we follow [57] by smoothing the fixation locations via Gaussian blurs.

To further analyze the causes of saliency heterogeneity, we conduct the semantic segmentation for all 1,600 images via the open annotation tool LabelMe [99]. Specifically, we annotate 26,100 objects of 242 classes in total and identify objects that attract more attention for each individual participant. To achieve this, we compare the fixation map with the mask of a specific object and use the result as the attention value of the corresponding object. We then average the result over all images that containing the same object, and use it to measure the interestingness of the object to a specific participant. In Fig. 5.2, we illustrate some representative objects and persons and show the distribution of the interestingness of various objects for a same participant. We observe that all participants exhibit a similar level of interestingness measure on faces where they exhibit different interestingness measures on various objects such as watch, bow tie, *et al*. This validates that it is necessary to choose images with multiple objects to build our PSM data.

5.2.3 Dataset Analysis

Why is each image viewed multiple times for ground-truth annotation?

To validate whether it is necessity for a subject to view each image multiple times, we randomly sample 220 images, and each image is viewed by the same participant 10 times. The time interval for the same person to view the same image ranges from one day to one

	Person 1	Person 4	Person 6	Person 7	Person 8
men bow tie	0.068388	0.046459	0.035015	0.07911	0.025138
women bow tie	0.014818	0.019792	0.078912	0.109666	0.004215
men hand watch	0.034834	0.034573	0.057979	0.036348	0.027059
women hand watch	0.035535	0.04356	0.041277	0.033336	0.022686
men face	0.025989	0.044911	0.04291	0.03387	0.03736
women face	0.027088	0.040768	0.043192	0.037849	0.035902

Figure 5.2: The distribution of the interestingness of various objects for a same participant. The value is calculated as follows: we sum values of the fixation map intersecting with the mask of a specific object, and divide it with the total of fixation maps over the whole image. Thus higher value indicates that the participant puts more attention on the object.

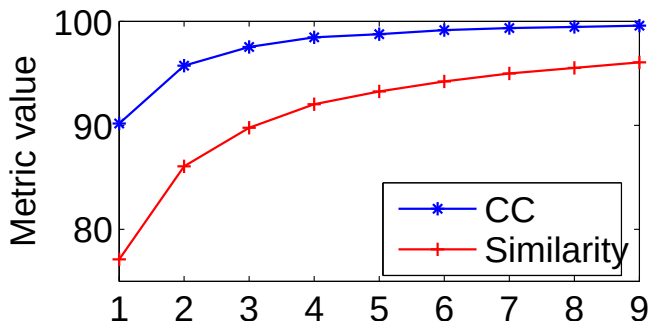


Figure 5.3: The point with $x = n$ measures the differences between ground truth saliency maps generated by viewing the same image n times and n+1 times. This figure shows that when $n \geq 4$, the ground truth saliency map generated by viewing the image n times has little difference with that generated by observing the image $n+1$ times. Thus viewing each image 4 times is enough to get a robust estimation of the PSM ground truth.

week because we want to get the short term memory of the person for the given image. We then calculate the differences of these saliency maps in terms of the commonly used metrics for saliency detection [56]: CC, Similarity. We average these criteria for all persons and all images, and we show the results in Fig. 5.3. We observe that the saliency map obtained by viewing each image only once vs. multiple times exhibit significant differences. Further, the saliency map averaged over 4 or more times is closer to the long term result.

Heterogeneity among different datasets.

To further illustrate that our proposed dataset is appropriate for personalized saliency detection task, we compare the inter-subject consistency, i.e., the agreement among different

viewers, in our PSM dataset and other related datasets. Specifically, for each dataset, we first enumerate all possible subject-pairs, i.e., two different subjects, and then compute the average AUC scores across all pairs. Recall that our PSM dataset consists of images from different datasets, eg, MIT, OSIE, ImageNet, PASCAL-S, SALICON, iSUN *etc*., and only MIT, OSIE, PASCAL-S are designed for saliency tasks[1]. Hence, we only compare the consistency scores among ours and above three datasets, and we show the results in Table 5.1. We observe that our dataset achieves the lowest inter-subject consistency values among all relative ones, indicating that the heterogeneity in our saliency maps are more severe than the others.

AUC judd scores			
Ours	MIT	OSIE	PASCAL-S
79.11	89.34	88.47	88.10

Table 5.1: Inter-subject consistency of different datasets. To compute the inter-subject consistency, we compute AUC judd for pair-wise saliency maps viewed by different observers for each image, then we average the results over all images. For fair comparison, the AUC judd of our method reported here is based on the saliency maps viewed by each observer once.

5.3 Approach

5.3.1 Problem Formulation

[25][90] employed CNN in an end-to-end strategy to predict saliency map and now serves as the state-of-the-art. Intuitively, we can follow the same strategy for PSM prediction, *i.e*. training a separate CNN for each participant to map the RGB images to PSM. However, such strategy is neither scalable nor feasible for a number of reasons. Firstly, it needs a vast amount of training samples to learn a robust CNN for each participant. This requires subjects to view thousands of images with high concentration, which is hard and extremely

[1] Even though SALICON and iSUN are also saliency fixation datasets, the ground truth were annotated based on mouse-tracking and web camera respectively.

time consuming. Secondly, training multiple CNNs for different subjects is computationally expensive and inefficient.

While each participant is unique in terms of their gender, race, age, personality, etc, resulting in their incongruity in saliency preference, different participants still share commonalities in their observed saliency maps because certain objects, such as faces and logos, always seem to attract the attention of all participants as shown in Fig. 5.1.

For this reason, instead of predicting the PSM directly, we set out to explore the difference map between USM and PSM. The discrepancy map $\Delta(P_n, I_i)$ for the given image I_i ($i = 1, \ldots, K$) of the n-th participant P_n ($n = 1, \ldots, N$) is of the form:

$$S_{PSM}(P_n, I_i) = S_{USM}(I_i) + \Delta(P_n, I_i) \tag{5.1}$$

where, $S_{PSM}(P_n, I_i)$ is the desired personalized saliency map and $S_{USM}(I_i)$ is the universal saliency map.

Note that the USMs by traditional saliency method entail the commonality in a saliency map observed by different participants. We convert the problem of predicting PSMs to estimating the discrepancy $\Delta(P_n, I_i)$ and we show it is much more efficient than directly estimating PSMs from RGB images as shown in . This is because that the universal saliency map $S_{USM}(I_i)$ itself already provides a rough estimation of the PSM, and predicting the discrepancy $\Delta(P_n, I_i)$ is actually easier than directly estimating the PSM from an RGB image. In addition, if we take the discrepancy $\Delta(P_n, I_i)$ as an error correction function, the PSM prediction problem can be therefore viewed as a regression task to correct the inaccurate input (USM), which can be implemented in high performance CNN scheme as shown in [17]. Given I_i and $S_{USM}(I_i)$, we propose a Multi-task CNN network to estimate $\Delta(P_n, I_i)$.

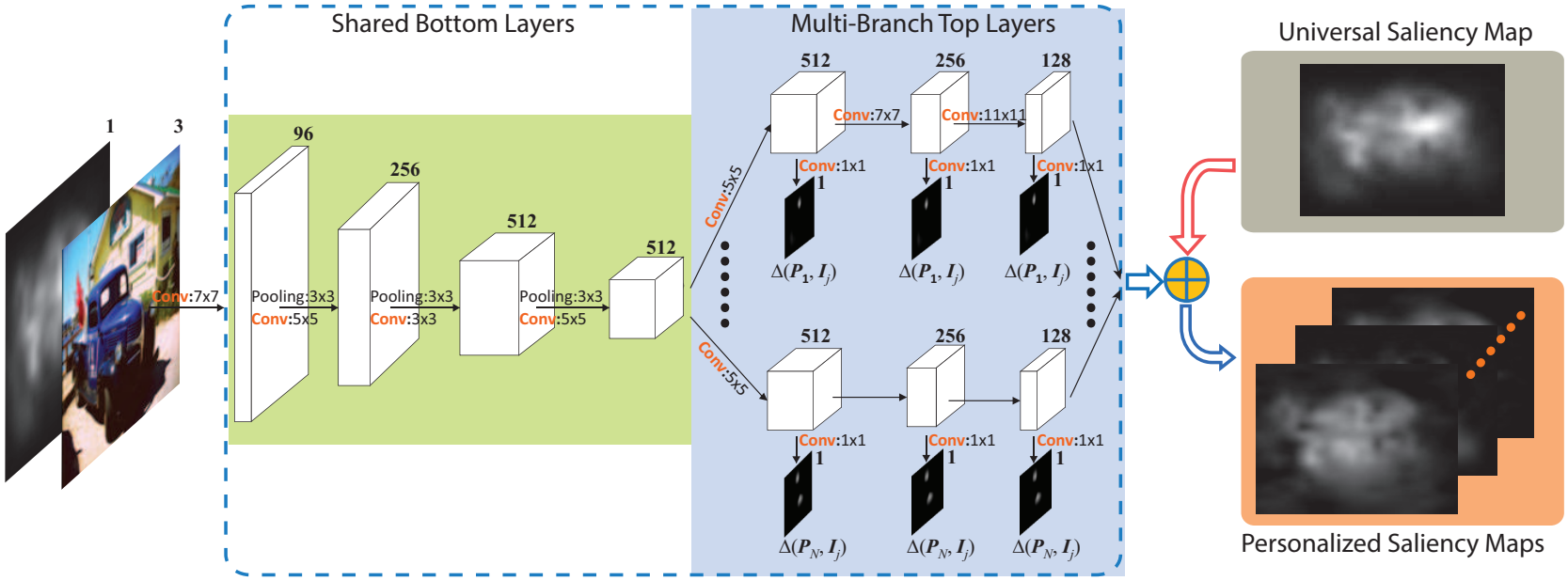


Figure 5.4: The pipeline of our Multi-task CNN based PSM prediction.

5.3.2 Multi-task CNN

Since $\Delta(P_n, I_i)$ is subject-dependent and at the same time dependant to the content of the input image, we construct a Multi-task CNN network to tackle it. The inputs of network are images with their corresponding universal saliency map and our goal is to estimate the discrepancy maps $\Delta(P_n, I_i)$ for n-th participants through n-th task. The network architecture of our Multi-task CNN is illustrated in Fig. 5.4.

Suppose we have N participants in total. We concatenate a 160×120 resolution RGB image with its USM from general saliency models and generate a $160 \times 120 \times 4$ cube as the input of the multi-task network. For image I_i, $\Delta(P_n, I_i)$ is the output of the n-th task corresponding to the discrepancy between PSM and USM for the n-th person. There are four convolutional layers shared by all participants after which the network is then split into N tasks which is exclusive for N participants. Each task has three convolutional layers followed by an ReLU activation function.

[25] and [69] show that by adding the supervision in the middle layers, the features learned by CNN will be more discriminative and can boost the performance of an given task. Consequently, we set an additional Loss Layer on *conv5* and *conv6* layer of the n-th task to impose the middle layer supervision , which can help the prediction of $\Delta(P_n, I_i)$. For the n-th task, $f_\ell^n(S_{USM}(I_i), I_i) \in \mathbb{R}^{h_\ell \times w_\ell \times d_\ell} (\ell = 5, 6, 7)$ is the feature map after the ℓ-th convolutional layer (the first convolutional layer corresponds to the first exclusive convolutional layer, so ℓ starts from 5). For each feature map $f_\ell^n(S_{USM}(I_i), I_i)$, a 1×1 convolutional layer was employed to map it to $S_\ell(S_{USM}(I_i), I_i) \in \mathbb{R}^{h_\ell \times w_\ell \times 1}$, which is the target discrepancy. To make $S_\ell(S_{USM}(I_i), I_i)$ close to $\Delta_\ell(P_n, I_i)$, we set the objective function as:

$$\min \quad \sum_{\ell=5}^{7} \sum_{n=1}^{N} \sum_{i=1}^{K} \| S_k(S_{USM}(I_i), I_i) - \Delta_\ell(P_n, I_i) \|_F^2 \tag{5.2}$$

Then we use mini-batch based stochastic gradient descent to optimize all parameters in our Multi-task CNN.

Remarks: Compared with techniques that use separate CNNs to predict $\Delta(P_n, I_i)$ for different participants, our Multi-task CNN architecture has the two key advantages:

Methods	CC	Similarity	AUC judd
RGB based MultiConvNets	62.24	65.27	77.83
RGB based Multi-task CNN	64.68	66.28	79.98
LDS [30]	65.73	63.34	82.96
LDS + MultiConvNets	70.71	75.65	83.69
LDS + Multi-task CNN	**72.19**	**76.07**	**84.97**
ML-Net [25]	41.35	51.30	71.80
ML-Net + MultiConvNets	65.35	79.42	81.70
ML-Net + Multi-task CNN	**67.53**	**80.17**	**83.45**
BMS [120]	59.59	71.36	80.26
BMS + MultiConvNets	68.68	79.66	83.79
BMS + Multi-task CNN	**70.33**	**80.41**	**85.03**
SalNet [90]	72.66	74.18	84.67
SalNet + MultiConvNets	74.85	77.89	85.09
SalNet + Multi-task CNN	**76.28**	**79.08**	**85.94**

Table 5.2: The performance comparison of difference methods on our PSM dataset.

1. Previous approaches [76] [121] have shown that features extracted by the first several layers can be shared between multiple tasks. In a similar vein, we treat PSMs as some distinct but related regression tasks across different individuals. Different from the multi-task CNN for USM prediction [76], our network shares lots of parameters which reduces the number of parameters and the memory consumption. Therefore, we are able to train these shared parameters using all training samples from all participants.

2. Note that in our architecture, the first few layers are shared and trained by all participants. In the deployment stage, given any unrecorded observer, our model only requires training the last three layers. Thus such a multi-task framework makes the problem scalable for open set settings.

5.4 Experiments

5.4.1 Experimental Setup

Parameters.

We implement our solution on the CAFFE framework [51]. We train our network with the following hyper-parameters setting: mini-batch size (40), learning rate (0.0003),

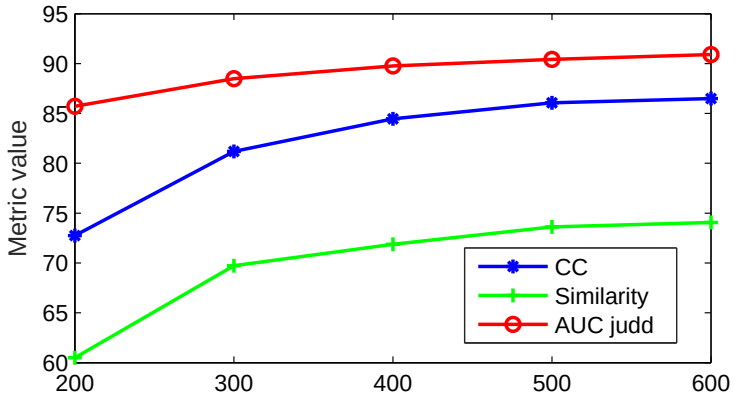


Figure 5.5: The effect of the number of training samples on the accuracy of PSM prediction.

momentum (0.9), weight decay (0.0005), and number of iterations (40,000). In our experiments, we randomly select 600 images ar training data, and use the rest 1,000 images for testing. To avoid over-fitting while improving model robustness, we augment the training data through left-right flip operations.

The parameters corresponding to the universal saliency map channel and 1×1 *conv* layers for middle layer supervision are initialized with 'xavier'. Using the initialization step in [90] and [66], we use the well-trained DeepNet model to initialize the corresponding parameters in our network. The network architecture of our Multi-task CNN is identical to that of DeepNet [90] except that i) the parameters corresponding to tasks of different participants are different; ii) middle layer supervision is imposed by adding 1×1 *conv* layer after *conv5* and *conv6*; iii) a channel corresponding to USM is added in the input.

Baselines.

Based on the performance of existing methods on the MIT saliency benchmark [15] in terms of similarity, we choose LDS [30], BMS [120], ML-Net [25], and SalNet [90] to predict the universal saliency maps on our dataset. The first two methods are based on hand-crafted features, and the latter two are based on deep learning techniques. We use their code provided online to generate USMs.

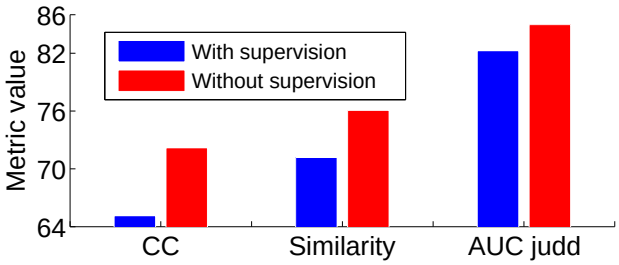


Figure 5.6: The effect of supervision on middle layers in our Multi-task CNN.

To validate the effectiveness of our model, we have compared our scheme with several baseline algorithms:

- **RGB based MultiConvNets**: MultiConvNets are trained to predict $\Delta(P_n, I_i)$ for each participant independently, with RGB images as input.
- **RGB based Multi-task**: Multi-task CNN architecture is trained to predict $\Delta(P_n, I_i)$ for all participants simultaneously, with RGB images as input.
- X**+MultiConvNets**: MultiConvNets are trained to predict $\Delta(P_n, I_i)$ for each participant independently, with RGB images and USM provided by method X as input, where X donates LDS, BMS, ML-Net, and SalNet respectively.

Notice that network architectures of the baseline ones are similar. The major differences are the number of input channels and whether the parameters are shared in the first few layers. For fair comparisons, we have employed the same strategies on data augmentation, middle layers supervision, and parameter initializations.

Measurements.

We adopt the same evaluation metrics in [79], [90] and [66] and choose CC, Similarity, and AUC [56] to measure the differences between the predicted saliency map and ground truth.

5.4.2 Performance Evaluation

The performance of all methods are listed in Table 5.2. We also show some predicted saliency maps for different participants in Fig. 5.7. We observe that our solution achieves the best performance in locating the incongruity fixation among individuals. Furthermore, the discrepancy based personalized saliency detection methods consistently outperform directly predicting PSM from RGB images. This validates the effectiveness of our "error correction" strategy for personalized saliency detection. In addition, the multi-task CNN scheme shows higher performance for fixation prediction for individuals tasks than simply training a CNN for each individual.

The effect of supervision on middle layers

Fig. 5.6 shows the accuracy gain from imposing supervision on middle layers in our Multi-task CNN. We observe that middle layer supervision is helpful for PSM prediction in line with previous findings [69].

The effect of the number of training samples on the PSM prediction accuracy.

Fig. 5.5 shows that increasing the number of training samples from 200 to 600 (the testing data are fixed) helps to improve the testing accuracy. However, training a more robust deep network requires large-scale training samples which would increase the time complexity tremendously.

5.5 Discussion

Our work demonstrates that heterogeneity in saliency maps cross individuals is common and critical for reliable saliency prediction, consistent with recent psychology studies showing that saliency is highly specific than universal. We have built the first PSM dataset and presented a framework to model such heterogeneity in terms of the discrepancy between PSM and USM. We have further presented a Multi-task CNN framework for the prediction of this discrepancy. To our knowledge, this is the first comprehensive study on personalized saliency and it is expected to stimulate significant future research.

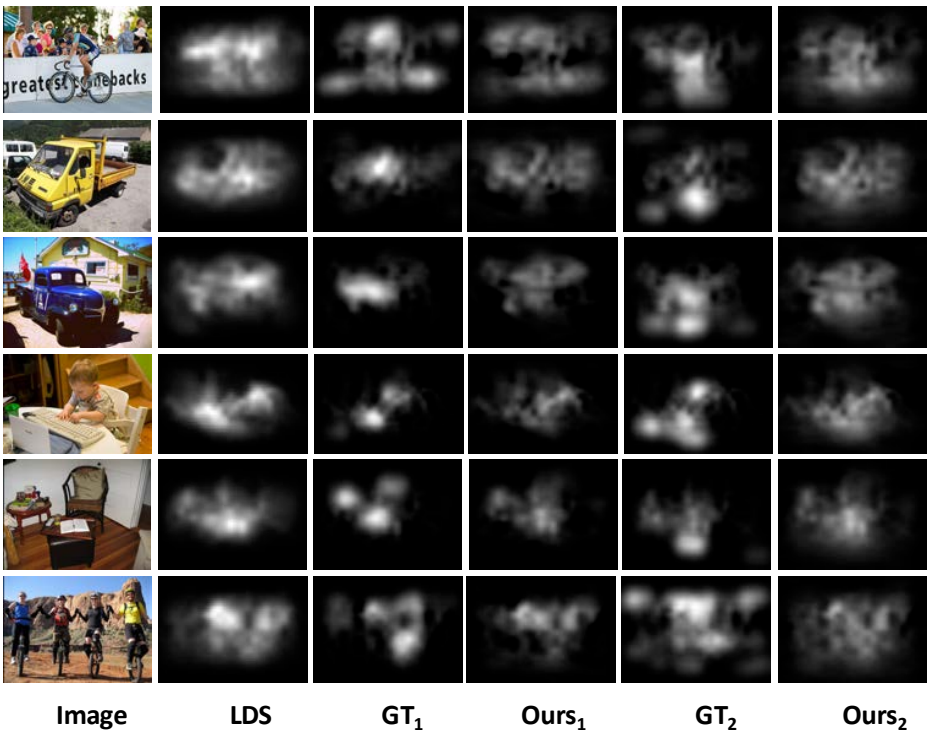


Figure 5.7: Some images, their ground truth PSM for different persons, and PSM predicted by our approach. The subscript indexes the ID of the participant.

In our data collection process, each participant needs to observe thousands of images on a single eye-tracker device, which is a bottleneck to increase both the number of images and participants. Clearly additional eye trackers will greatly improve the PSM collection process and can help build an even bigger dataset. Further, a key finding in our study is that personalized saliency is closely related to the observers' personal information (gender, race, major, *etc*.). If we obtain such information in prior, we can directly incorporate it into the PSM prediction to further improve the accuracy and efficiency.

Chapter 6

CONCLUSION AND FUTURE WORK

Many computational models of attention exist which aim to either detect the salient objects or predict where general people look, but they usually ignore some substantial problems that prevent the computational model accurately simulating real human visual attention on the nature scene. On one hand, the majority of available datasets for saliency detections are regular images, which have already lost the high dimensional information of the real scene. On the other hand, the ground truth annotated by existing datasets overlook the incongruent gaze patterns of individuals, which impact the prediction performances. This dissertation introduces solutions for both of the problems.

We show that by integrating the high dimensional information into the regular images, we can greatly improve the detection performance compared with traditional method. We built the first light field dataset for salient object detection task. This dataset consists of plenty of both indoor and outdoor scenes and ground truth maps for one all-focus image in the scene.

The models using different types of scene data differ in both low-level features and processing procedures. Our second contribution is providing a unified framework to handel different types of input scene data. This novel saliency detection algorithm is applicable to regular image, image with depth information, and light field scene. The prediction performance of our model achieves state-of-the-art.

If the scene consists of several objects, the inconsistency of individual gaze pattern will become indispensable. To address this problem, we present a multi-task CNN framework to encode the discrepancy gaze information between personal gaze and general gaze into the state-of-the-art saliency models, which improve the fixation prediction accuracy for

simulating individual visual attention. We also built the first dataset for personalized saliency prediction tasks. This dataset contain 1,600 images, and 30 observers' gaze fixation are provided. We asked each of the paticipant to free-viewing the dataset four times. We note that certain types of objects such as watches, belts would introduce more gaze incongruity whereas other types like faces and text would lead to more coherent fixation maps.

6.1 Future work

Our immediate future work is using eye gaze information to predict the depth of objects in the scene. We find it possible to do this based on the fact that the pupil shape and gaze pattern of human eyes will change when people look at object on diverse depth. The recent developing of Virtual Reality technique makes it possible to provide enough number of eye images for us to learn the relationship between the depth and the gaze. This work benefits the several computer vision tasks. An intuitive application is that we can get the 3D information of the scene just by an eye tracker. Also, we can generate the depth map for single image when providing the gaze information towards the scene. We are currently working on building an eye tracker system to capture the gaze data and clearly there are still plenty of interesting questions waiting for us to investigate in the future.

BIBLIOGRAPHY

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 1597–1604. IEEE, 2009.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, November 2012.

[3] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. Interactive digital photomontage. In *ACM SIGGRAPH 2004*, 2004.

[4] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[5] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[6] Tamar Avraham and Michael Lindenbaum. Esaliency (extended saliency): Meaningful attention using stochastic image modeling. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):693–708, 2010.

[7] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878*, 2014.

[8] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *Image Processing, IEEE Transactions on*, 24(12):5706–5722, 2015.

[9] Ali Borji and Laurent Itti. Exploiting local and global patch rarities for saliency detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 478–485. IEEE, 2012.

[10] Ali Borji and Laurent Itti. Exploiting local and global patch rarities for saliency detection. In *CVPR*, pages 478–485. IEEE, 2012.

[11] Ali Borji, Dicky N. Sihite, and Laurent Itti. Salient object detection: a benchmark. In *ECCV*, 2012.

[12] Andrew P Bradley and Fred WM Stentiford. Visual attention for region of interest coding in jpeg 2000. *Journal of Visual Communication and Image Representation*, 14(3):232–250, 2003.

[13] N. Bruce and J. Tsotsos. Saliency based on information maximization. In *NIPS*, pages 155–162, 2006.

[14] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 425–432, 2001.

[15] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. Mit saliency benchmark.

[16] Ran Carmi and Laurent Itti. Visual causes versus correlates of attentional selection in dynamic scenes. *Vision research*, 46(26):4333–4345, 2006.

[17] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. *arXiv preprint arXiv:1507.06550*, 2015.

[18] Moran Cerf, Jonathan Harel, Wolfgang Einhäuser, and Christof Koch. Predicting human gaze using low-level saliency combined with face detection. In *Advances in neural information processing systems*, pages 241–248, 2008.

[19] Miko May Lee Chang, Soh Khim Ong, and Andrew Yeh Ching Nee. Automatic information positioning scheme in ar-assisted maintenance based on visual saliency. In *SAIENTO AVR*, pages 453–462. Springer, 2016.

[20] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015.

[21] M.M. Cheng, G.X. Zhang, N.J. Mitra, X. Huang, and S.M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.

[22] Hannah Faye Chua, Julie E Boland, and Richard E Nisbett. Cultural variation in eye movements during scene perception. *Proceedings of the National Academy of Sciences of the United States of America*, 102(35):12629–12633, 2005.

[23] Arridhana Ciptadi, Tucker Hermans, and James M. Rehg. An In Depth View of Saliency. In *British Machine Vision Conference (BMVC)*, 2013.

[24] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[25] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A deep multi-level network for saliency prediction. *arXiv preprint arXiv:1609.01064*, 2016.

[26] Karthik Desingh, K Madhava Krishna, Deepu Rajan, and C Jawahar. Depth really matters: Improving visual salient region detection with depth. In *Proc. BMVC*, 2013.

[27] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[28] Wolfgang Einhäuser and Peter König. Does luminance-contrast contribute to a saliency map for overt visual attention? *European Journal of Neuroscience*, 17(5):1089–1097, 2003.

[29] Wolfgang Einhäuser, Merrielle Spain, and Pietro Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):1–26, 2008.

[30] S. Fang, J. Li, Y. Tian, T. Huang, and X. Chen. Learning discriminative subspaces on random contrasts for image saliency analysis. *TNNLS*, 2016.

[31] Hans G. Feichtinger and T. Strohmer, editors. *Gabor Analysis and Algorithms: Theory and Applications*. Birkhauser Boston, 1st edition, 1997.

[32] D. Gao, S. Han, and N. Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *TPAMI*, 31(6):989–1005, 2009.

[33] Dashan Gao and Nuno Vasconcelos. Bottom-up saliency is a discriminant process. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–6. IEEE, 2007.

[34] Anton Garcia-Diaz, Victor Leboran, Xose R Fdez-Vidal, and Xose M Pardo. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17–17, 2012.

[35] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, 2012.

[36] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *IEEE transactions on image processing*, 19(1):185–198, 2010.

[37] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, Fabian Nater, and Luc Van Gool. The interestingness of images. In *ICCV*, pages 1633–1640, 2013.

[38] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. In *ACM Transactions on Graphics (TOG)*, volume 30, page 70. ACM, 2011.

[39] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, pages 1–8, 2007.

[40] Xiaodi Hou, Jonathan Harel, and Christof Koch. Image signature: Highlighting sparse salient regions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):194–201, 2012.

[41] Jian Huang, Shuangge Ma, and Cun hui Zhang. Adaptive lasso for sparse highdimensional regression. Technical report, University of Iowa, 2006.

[42] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *ICCV*, pages 262–270, 2015.

[43] Aaron Isaksen, Leonard McMillan, and Steven J. Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, 2000.

[44] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.

[45] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *TPAMI*, 20(11):1254–1259, 1998.

[46] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE TIP*, 13(10):1304–1318, 2004.

[47] Laurent Itti. Quantitative modelling of perceptual salience at human eye position. *Visual cognition*, 14(4-8):959–984, 2006.

[48] Laurent Itti and Pierre F Baldi. Bayesian surprise attracts human attention. In *Advances in neural information processing systems*, pages 547–554, 2006.

[49] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[50] Harel J., Koch C., and Pernoa P. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.

[51] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[52] Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Yihong Gong, Nanning Zheng, and Jingdong Wang. Salient object detection: A discriminative regional feature integration approach. *CoRR*, abs/1410.5926, 2014.

[53] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *CVPR*, pages 1072–1080, 2015.

[54] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by ufo: Uniqueness, focusness and objectness. In *ICCV'13*, pages 1976–1983, 2013.

[55] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. Salient region detection by UFO: Uniqueness, Focusness and Objectness. In *ICCV*, 2013.

[56] Tilke Judd, Frédo Durand, and Antonio Torralba. A benchmark of computational models of saliency to predict human fixations. In *MIT Technical Report*, 2012.

[57] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *ICCV*, pages 2106–2113, 2009.

[58] Frédéric Jurie and Cordelia Schmid. Scale-invariant shape features for recognition of object categories. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[59] Timor Kadir and Michael Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[60] Changil Kim, Alexander Hornung, Simon Heinzle, Wojciech Matusik, and Markus Gross. Multi-perspective stereoscopy from light fields. *ACM Trans. Graph.*, 30(6):190:1–190:10, December 2011.

[61] J. Kim, D. Han, Y.-W. Tai, and J. Kim. Salient region detection via high-dimensional color transform. In *CVPR*, 2014.

[62] Matthias Kmmerer, Lucas Theis, and Matthias Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. *Computer Science*, 2014.

[63] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

[64] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, 2000. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[65] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 25(2):2012, 2012.

[66] Srinivas S. S. Kruthiventi, Vennela Gudisa, Jaley H. Dholakiya, and R. Venkatesh Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *CVPR*, pages 5781–5790, 2016.

[67] Congyan Lang, Tam V. Nguyen, Harish Katti, Karthik Yadati, Mohan Kankanhalli, and Shuicheng Yan. Depth matters: influence of depth cues on visual saliency. In *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, ECCV'12, pages 101–115, 2012.

[68] Olivier Le Meur, Patrick Le Callet, Dominique Barba, and Dominique Thoreau. A coherent computational approach to model bottom-up visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 28(5):802–817, 2006.

[69] Chen Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. *Arxiv*, pages 562–570, 2014.

[70] Marc Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.

[71] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM.

[72] Feng Li and Fatih Porikli. Harmonic variance: A novel measure for in-focus segmentation. In *BMVC*, 2013.

[73] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[74] Nianyi Li, Bilin Sun, and Jingyi Yu. A weighted sparse coding framework for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5216–5223, 2015.

[75] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *CVPR*, June 2014.

[76] X. Li, L. Zhao, L. Wei, M. H. Yang, F. Wu, Y. Zhuang, H. Ling, and J. Wang. Deepsaliency: Multi-task deep neural network model for salient object detection. *TIP*, 25(8):3919–3930, 2016.

[77] Xiaohui Li, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via dense and sparse reconstruction. In *ICCV*, 2013.

[78] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[79] Nian Liu, Junwei Han, Dingwen Zhang, Shifeng Wen, and Tianming Liu. Predicting eye fixations using convolutional neural networks. In *CVPR*, pages 362–370, 2015.

[80] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.Y. Shum. Learning to detect a salient object. *TPAMI*, 33(2):353–367, 2011.

[81] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 1337–1342, 2015.

[82] Yu-Fei Ma and Hong-Jiang Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 374–381. ACM, 2003.

[83] Long Mai, Yuzhen Niu, and Feng Liu. Saliency aggregation: A data-driven approach. In *CVPR*, 2013.

[84] Atsuto Maki, Peter Nordlund, and Jan-Olof Eklundh. Attentional scene segmentation: Integrating depth and motion. *Computer Vision and Image Understanding*, 78(3):351–373, 2000.

[85] Frank Moosmann, Eric Nowak, and Frederic Jurie. Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1632–1646, 2008.

[86] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56. IEEE, 2010.

[87] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, April 2005.

[88] Yuzhen Niu, Yujie Geng, Xueqing Li, and Feng Liu. Leveraging stereopsis for saliency analysis. In *CVPR*, 2012.

[89] Hans-Christoph Nothdurft. Salience from feature contrast: additivity across dimensions. *Vision Research*, 40(10C12):1183 – 1201, 2000.

[90] Junting Pan, Elisa Sayrol, Xavier Giroinieto, Kevin Mcguinness, and Noel E. Oconnor. Shallow and deep convolutional networks for saliency prediction. In *CVPR*, pages 598–606, 2016.

[91] Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision research*, 42(1):107–123, 2002.

[92] Houwen Peng, Bing Li, Weihua Xiong, Weiming Hu, and Rongrong Ji. Rgbd salient object detection: a benchmark and algorithms. In *ECCV*, pages 92–109, 2014.

[93] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[94] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

[95] Subramanian Ramanathan, Harish Katti, Nicu Sebe, Mohan Kankanhalli, and Tat-Seng Chua. An eye fixation database for saliency detection in images. *Computer Vision–ECCV 2010*, pages 30–43, 2010.

[96] John H. Reynolds and Robert Desimone. Interacting roles of attention and visual salience in $\{$V4$\}$. *Neuron*, 37(5):853 – 863, 2003.

[97] Michael Rubinstein, Ariel Shamir, and Shai Avidan. Improved seam carving for video retargeting. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):1–9, 2008.

[98] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[99] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[100] Vidya Setlur, Saeko Takagi, Ramesh Raskar, Michael Gleicher, and Bruce Gooch. Automatic image retargeting. In *MUM*, pages 59–68, 2005.

[101] X. Shen and Y. Wu. A unified approach to salient object detection via low rank matrix recovery. In *CVPR*, pages 853–860, 2012.

[102] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[103] Bongwon Suh, Haibin Ling, Benjamin B. Bederson, and David W. Jacobs. Automatic thumbnail cropping and its effectiveness. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, UIST '03, pages 95–104, 2003.

[104] Yaoru Sun and Robert Fisher. Object-based visual attention for computer vision. *Artificial intelligence*, 146(1):77–123, 2003.

[105] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[106] Benjamin 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*, 7(14):4–4, 2007.

[107] R. Valenti, N. Sebe, and T. Gevers. Image saliency by isocentric curvedness and color. In *IEEE International Conference on Computer Vision*, 2009.

[108] Joost Van de Weijer, Theo Gevers, and Andrew D Bagdanov. Boosting color saliency in image feature detection. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):150–156, 2006.

[109] Eleonora Vig, Michael Dorr, and D. Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *CVPR*, pages 2798–2805, 2014.

[110] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2015.

[111] S. Wanner, C. Straehle, and B. Goldluecke. Globally consistent multi-label assignment on the ray space of 4d light fields. In *CVPR*, 2013.

[112] Y. Wei, F. Wen, W. Zhu, and J. Sun. Geodesic saliency using background priors. In *ECCV*, 2012.

[113] Ying Wu. A unified approach to salient object detection via low rank matrix recovery. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 853–860, Washington, DC, USA, 2012. IEEE Computer Society.

[114] Juan Xu, Ming Jiang, Shuo Wang, Mohan S Kankanhalli, and Qi Zhao. Predicting human gaze beyond pixels. *Journal of vision*, 14(1):28–28, 2014.

[115] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[116] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, 2013.

[117] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013.

[118] Jinwei Ye and Jingyi Yu. Ray geometry in non-pinhole cameras: a survey. *The Visual Computer*, pages 1–20, 2013.

[119] Yun Zhai. Visual attention detection in video sequences using spatiotemporal cues. *ACM MM*, 2006.

[120] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *Proceedings of the IEEE international conference on computer vision*, pages 153–160, 2013.

[121] Zhanpeng Zhang, Ping Luo, Change Loy Chen, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pages 94–108, 2014.

[122] Qi Zhao and Christof Koch. Learning visual saliency. In *Information Sciences and Systems (CISS), 2011 45th Annual Conference on*, pages 1–6. IEEE, 2011.

[123] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1265–1274, 2015.

[124] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *CVPR*. IEEE, 2014.

Appendix A

PERMISSIONS

Title: Saliency Detection on Light Field
Author: Nianyi Li
Publication: Pattern Analysis and Machine Intelligence, IEEE Transactions on
Publisher: IEEE
Date: 1 Aug. 2017

Copyright © 2017, IEEE

LOGIN

If you're a copyright.com user, you can login to RightsLink using your copyright.com credentials. Already **a RightsLink user** or want to learn more?

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Copyright © 2017 Copyright Clearance Center, Inc. All Rights Reserved. Privacy statement. Terms and Conditions.
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

Figure A.1: The right link of Chapter 3