Visual Salient Object Detection: Interactive, Unsupervised and Supervised Methods

by

Xuebin Qin

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

The human vision system has an effective mechanism for retrieving and localizing the most important information from visual scenes. In computer vision, Salient Object Detection (SOD) algorithms aim at modeling this mechanism by extracting or segmenting these salient targets from given images or video frames. Such algorithms can be used in a wide range of applications such as image segmentation, image editing, visual tracking, robot navigation, *etc.* Three categories of salient object detection methods are studied in this thesis. 1) **Interactive Annotation with ByLabel: A Boundary based Semi-Automatic Image Annotation Tool.** We develop a novel boundary based semi-automatic tool, ByLabel, for accurate image annotation. Given an image, ByLabel first detects its edge features and computes high-quality boundary fragments. Current labeling tools require the operators to accurately click on numerous boundary points. ByLabel simplifies this process to selecting of the automatically detected boundary fragments. To evaluate the performance of ByLabel, 10 volunteers, with no experience of image annotation, were asked to label both synthetic and real images. Compared to the commonly used tool LabelMe, ByLabel reduces image-clicks and time by 73% and 56% respectively, while improving the accuracy by 73%. The results show that our ByLabel outperforms the popular annotation tool, LabelMe, in terms of efficiency, accuracy and user experience. 2) **Unsupervised Salient Closed Boundary Extraction by Perceptual Grouping.** Salient closed boundary extraction aims to automatically identify and connect a subset of detected

fragments to form a closed boundary based on the principles of Gestalt laws. Particularly, we propose a novel method for perceptual grouping of the salient closed boundary, in which the salient closed boundary extraction problem is formulated as a problem of searching for a special cycle from an undirected graph. We propose a novel graph-based optimization algorithm "Bi-Directional Shortest Path (BDSP)" for searching the special graph cycle. In addition, we adapt our new method to different applications including building outline extraction and salient closed boundary tracking. Experimental results show that our methods outperform the state-of-the-art (SOTA) methods of different applications in terms of both robustness and accuracy. 3) **Supervised Salient Object Detection by Deep Convolutional Neural Networks.** Deep Convolutional Neural Networks (DCNN) have been adopted for salient object detection and achieved state-of-the-art performance. Most of the previous works, however, focus on region accuracy but not on the boundary quality. We propose a predict-refine architecture, **BASNet** (348.5 MB, 25 frames per second (FPS)), and a new hybrid loss for Boundary-Aware Salient object detection. Experimental results show that our method outperforms the SOTA methods both in terms of regional and boundary evaluation measures. To achieve lighter models with faster speed, we further design a simple yet powerful deep network architecture, **U^2-Net**, with a two-level nested U-structure for salient object detection. A full size U^2-Net (176.3 MB, 30 FPS) and a small size U^2-Net† (4.7 MB, 40 FPS) are instantiated based on the nested architecture. Both of them achieve very competitive results against the SOTA models.

Preface

Material for this thesis is based on the following papers.

Chapter 3: **Xuebin Qin**, Shida He, Zichen Zhang, Masood Dehghan and Martin Jagersand. ByLabel: A Boundary based Semi-Automatic Image Annotation Tool. *In Proc. of the IEEE Winter Conf. on Applications of Computer Vision* (WACV), March 2018.

Chapter 4.2: **Xuebin Qin**, Shida He, Xiucheng Yang, Masood Dehghan, Qiming Qin and Martin Jagersand. Accurate Outline Extraction of Individual Building from High Resolution Aerial Images. *IEEE Geoscience and Remote Sensing Letters* (GRSL), 2018.

Chapter 4.3: **Xuebin Qin**, Shida He, Camilo Perez Quintero, Abhineet Singh, Masood Dehghan and Martin Jagersand. Real-time salient closed boundary tracking via line segments perceptual grouping. *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS), September 2017.

Chapter 4.4: **Xuebin Qin**, Shida He, Zichen Zhang, Masood Dehghan and Martin Jagersand. Real-time salient closed boundary tracking using perceptual grouping and shape priors, *In Proc. of the 28th British Machine Vision Conference* (BMVC), spotlight, September 2017.

Chapter 5.3: **Xuebin Qin**, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan and Martin Jagersand. BASNet: Boundary-Aware Salient Object Detection. *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), June 2019.

Chapter 5.4: **Xuebin Qin**, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar Zaiane and Martin Jagersand. U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection. *Under review*, 2019.

Acknowledgements

First and foremost I would like to acknowledge the support and help of my research advisor: Prof. Martin Jagersand. Martin is not only a nice advisor but also a reliable collaborator and considerate friend to his students. I am especially grateful for the freedom, encouragement, guidance and academic training he gave me in the past years. His joy and enthusiasm for his work will keep motivating me to pursue my career in research.

I also thank my thesis committee members: Prof. Herb Yang, Prof. Nilanjan Ray, Prof. Pierre Boulanger, and Prof. James Elder. I highly appreciate their time and constructive suggestions on this thesis.

I would like to thank my labmates in Prof. Martin Jagersand's vision and robotics group for their invaluable help in my research and life. They are Dr. Masood Dehghan, Dr. Camilo Quintero, Dr. Dana Cobzas, Zichen Zhang, Shida He, Laura Petrich, Min Tang, Jun Jin, Rong Feng, Mennatullah Siam, Sepehr Valipour, Steven Lu, Chen Jiang, Connor Stephens, Graham Alexander, Rafaella Graña and Daniel-Theodor Plop. I thank my friends: Dr. Yiming Qian, Dr. Gaojian Fan, Dr. Wanxin Gao, Dr. Ziqi Wei, Chenyang Huang, Chao Gao, Siyang Tian, Ali Jahani Amiri, Shing-Yan Loo, Dr. Xiaolong Wang, Dr. Yu Xia, Juehui Fan, Dr. Changjiang Liu, Dr. Guozheng Xu, Lidong Zou, Dr. Chao Chen, Dr. Xiucheng Yang, Dr. Dornoosh Zonoobi, Dr. Abhilash RH, Roberto Vega, Farzane Aminmansour, Siyavash G. Nia, Pouneh Gorji and John Smolley.

Last but not least, I would like to thank my family members for their unconditional support and never-ending love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

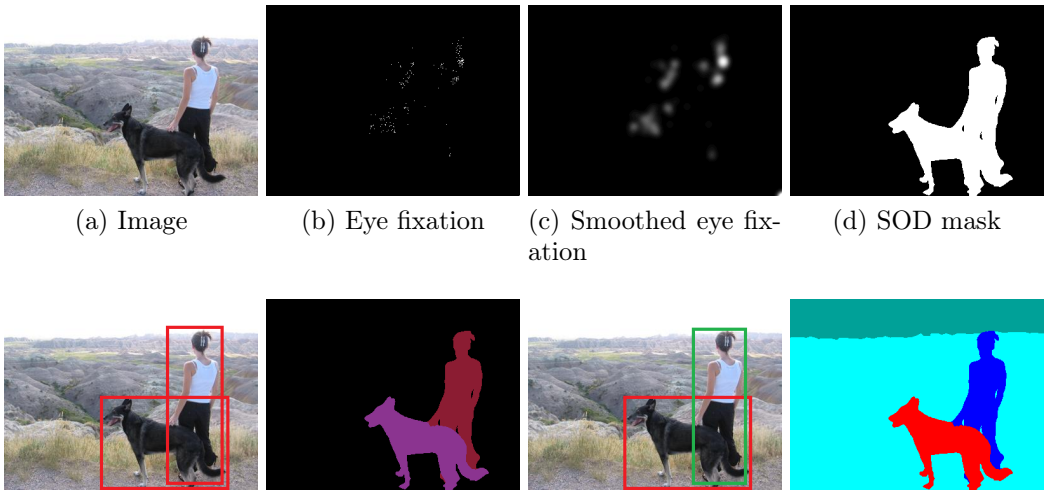
1.1 Background

Large amounts of information including sight, hearing, smell, taste, touch and so on, are perceived by our perception systems. Among all the information obtained by different sensory organs, visual information is the richest. There are $10^8 - 10^9$ bits of visual data entering our eyes every second [27], [104], [123]. Due to the limited perceptual and cognitive resources, it is difficult to process all the data in real-time without the help of effective mechanisms for filtering out the noise and irrelevant parts [27]. The visual attention mechanism, as one of the most efficient and effective mechanisms, plays an important role in our vision system. It helps in quickly retrieving and localizing the most important parts from the visual data and guiding our perception systems to focus on the selected small amount of salient data [269]. In the past decades, the visual attention mechanism has already been widely studied in psychology, neurophysiology, computational neurosciences as well as robotics and computer vision from different perspectives [27]. Psychologists [141], [167], [227] aim at studying the relationship between human behavior, *e.g.* gaze, head and eye movements, and visual attention. Neurophysiologists [240], [243] are interested in explaining how attention mechanism works on neurons. Computational neuroscientists [33], [210] target at studying the attention mechanism by numerical analysis and computer simulation based on the real biophysical models of neurons. Robotics and computer vision scientists [27], [258] mainly focus on modeling the visual attention mechanism by visual saliency detection

from the given image or video input.

The terms "visual attention" and "visual saliency" are closely related and sometimes can be used interchangeably [27]. As described above, "visual attention" refers to a set of cognitive operations that select the conspicuous parts from a visual scene [178]. While "visual saliency" stands for the distinct quality of each part in a visual scene. This quality measures the ability of each part in standing out from its neighbors and grabbing the attention [27], [105]. Instead of modeling the exact cognitive processes of "visual attention" mechanism, most computer vision researchers choose to simulate the "visual attention" mechanism by computing and analyzing the "visual saliency" of given input images and videos. Saliency detection dates back to the saliency-based bottom-up visual attention model built by Itti *et al.* [106] based on the cognitive theories, *e.g.* Feature Integration Theory (FIT) stated by Treisman and Gelade [239] and Guided Search Model proposed by Wolfe *et al.* [268], and early computational frameworks [17], [179], [241], *e.g.* Computational Attention Architecture by Koch and Ullman [122]. This model [106] is the first complete implementation and verification of the Koch and Ullman model [122], which introduces the concept of saliency map to represent the conspicuousness of scene locations [27]. It combines multi-scale image features into a single **topographical saliency map** and facilitates the complex scene understanding by selecting the attentive locations in the order of decreasing saliency. Later, to study the relationship between visual attention and human eye movements, researchers [31], [193] start to predict **eye fixation**. The predicted eye fixation maps are similar to the topographical saliency maps, which have no clear boundaries. To facilitate semantic image understanding, more region information is then introduced into visual saliency estimation [153], [169], [248]. Inspired by the above studies, Liu *et al.* [159], [160] and Achanta *et al.* [1] directly formulate the saliency detection as a binary segmentation problem, which is usually named as **Salient Object Detection (SOD)**.

In addition to the visual stimulus (bottom-up cues) and the human instincts, high-level human knowledge (top-down influences) including experiences, expectations, rewards and so on [27], also plays an important role in



(a) Image (b) Eye fixation (c) Smoothed eye fixation (d) SOD mask

(e) SOD instances (f) SOD instances (g) Object detection (h) Semantic segmentation

Figure 1.1. Outputs of closely related tasks [279]: (a) shows the input image; (b) is the ground truth of eye fixation prediction and (c) is its continuous version; (d) is the ground truth, which is a binary mask, of salient object detection; (e) and (f) are the SOD instances represented by bounding boxes and region masks respectively; (g) and (f) show the ground truth of object detection and semantic segmentation respectively. Compared with (e) and (f), the bounding boxes and region masks in (g) and (h) are labeled with values for indicating the object classes.

"visual attention". Land and Hayhoe [135] illustrate that the visual attention is closely related to the hand movements in extended food preparation tasks and prove that the eye movements in these tasks are mainly related to the task-relevant objects and less dependent on the 'intrinsic salience' of objects. Another related area in computer vision is active vision, in which the viewpoint of the camera can be manipulated to better investigate the environment based on the goal of the active vision system other than the 'intrinsic salience' of the visual stimulus [6], [16]. As a matter of fact, in our real life, both bottom-up and top-down visual saliency play important roles in our vision systems and perception processes. The definitions of visual saliency vary from different stimulus as well as different scenarios and tasks. Therefore, many computer vision related problems and applications can be formulated and solved as visual salient object detection problems.

1.2 Related Research Areas

Given an input image, the saliency related tasks can be categorized into the following groups according to their outputs:

- Eye fixation prediction;
- Salient object detection;
- Salient instance detection;
- Salient object subitizing.

In addition, other tasks including *object proposal* [45], [312], *object detection* (see. Fig. 1.1(g)) [79], [80], [207], *image segmentation* (see Fig. 1.1(h)) [42], [152], [163], *image annotation* [37], [216] are also closely related to salient object detection.

1.2.1 Eye Fixation Prediction

Eye fixation prediction aims to quantitatively estimate the human eye attended locations on the given input image [158]. It has been widely used in many applications, *e.g.* image segmentation [260], video summarization [69], eye tracker calibration [156], gaze estimation [235], video retargeting [214], *etc.* To evaluate the performance of an eye fixation model, eye fixation data *e.g.* DUT-OMRON [279], are collected by eye tracking apparatus, which records eye fixations (a set of image coordinates) when the participant focuses on the input image shown in the monitor (for 2 seconds). The initial eye fixation ground truth is shown in Fig. 1.1(b). The continuous eye fixation maps (see. Fig. 1.1(c)) generated by convolving across each fixation location using a Gaussian filter are usually utilized to evaluate the accuracy of eye fixation models.

In early conventional eye fixation prediction models, the contrast and the rarity of local image patches *e.g.* [30], [31], [89], [106], [193] as well as the global perception mechanism *e.g.* figure-ground segregation [26], [293], which is likely to be influenced by factors including size, surroundedness, convexity

and symmetry [191], are often employed to compute the visual saliency. In recent years, deep convolutional neural networks, *e.g.* SALICON [101], DeepNet [192], Mr-CNN [156], DeepFix [129], DVA [259], greatly raise the bar of eye fixation prediction thanks to large amounts of training data and the strong learning ability of deep networks [134].

1.2.2 Salient Object Detection

In general, salient object detection (or salient object/region segmentation) refers to the process of segmenting pixels belonging to the salient objects or regions from input images or videos [23]. The outputs of salient object detection are usually binary masks (see Fig. 1.1(d)), in which value one denotes the salient object and zero stands for the non-salient background. Compared with the outputs of eye fixation prediction, the binary saliency masks produced by salient object detection keep more geometrical details of the detected salient objects so that there are clear boundaries between salient and non-salient regions (see Fig. 1.1(d)). Thanks to the high spatial accuracy, salient object detection methods can be used in various applications, *e.g.* object recognition [217], scene understanding [25], object detection [224], image editing and manipulating [47], content-aware image resizing [12], [83], [290], image retrieval [84], visual tracking [76], robotics [39], *etc.*

Early salient object detection methods [1], [2], [44], [159], [160] first estimate the saliency of each image pixel/patch based on *priors of local or global contrast* computed by a set of handcrafted features *e.g.* edge contrast, center-surround histogram, color spatial distribution, *etc.* They then achieve the segmentation results by conducting Conditional Random Field (CRF) [133] or clustering methods *e.g.* K-means [171], on the computed saliency maps and the original images. Later on, the *image boundary and connectivity priors* [262], [279], [294], [310], which assume that the image backgrounds are usually homogeneous and connected to the image borders, are introduced to highlight the entire salient object uniformly. With the development of deep neural networks [94], [100], [128], [137], [228], especially the emergence of fully convolutional network (FCN) [163], the deep methods [58], [97], [147], [154], [157], [258]

for salient object detection achieve significant improvements both in terms of accuracy and robustness. Meanwhile, the new solutions of salient object detection provide valuable references for other binary segmentation problems.

1.2.3 Salient Instance Detection

The results produced by salient object detection are binary masks with clear boundaries. Sometimes, there are more than one salient objects in an input image. However, binarized saliency masks are not able to differentiate different instances of those detected multiple salient objects. To address this problem, salient instance detection models are developed. There are two ways for representing detected salient instances: (1) bounding boxes [4], [41], [45], [242], [295], [312] (see Fig. 1.1(e)) and (2) segmentation masks [72], [136], [144] (see Fig. 1.1(f)). Each bounding box indicates that there is one salient object enclosed by the bounding box, but no semantic class information is included, which is different from the bounding boxes output by general object detection methods, *e.g.* R-CNN [80], Fast R-CNN [79], Faster R-CNN [207], *etc.* Compared to the bounding boxes, the salient instance segmentation masks provide accurate shape description of each salient instance, which is similar to Mask R-CNN [92].

1.2.4 Salient Object Ranking and Subitizing

Salient Object Subitizing (SOS) is proposed to simulate the mechanism that the human vision system can immediately and precisely identify the number (1, 2, 3, 4) of items by a simple glimpse [95], [291], [292]. Zhang *et al.* [292] build a subitizing training dataset and try to directly predict the number of salient objects from input images. He *et al.* [95] construct a multi-task learning network to learn salient object detection and subitizing simultaneously. In addition, multiple salient objects detected from the same image are usually assigned with the same saliency level. In many scenarios, multiple salient objects are drastically different in terms of size, shape or appearance so that their saliency qualities are different [291]. Furthermore, different people may

have different views on the saliency of the same object. Therefore, Amirul [11] propose to simultaneously detect, rank and count the salient objects.

1.2.5 Summary

The outputs of the four saliency categories: eye fixation prediction; salient object detection; salient instance detection; salient object subitizing and ranking, have different characteristics and play important roles in many real-world applications. Sometimes, there are no clear boundaries among these four categories. For example, eye fixation prediction models can be used to improve the performance of salient object detection. The salient instance detection and subitizing can be conducted on the results of salient object detection. With the development of computer vision and the related applications, more and more accurate segmentation results are required. However, the results of eye fixation prediction are with low geometric accuracy so that their applications are limited. On the contrary, the salient object detection is essentially a binary segmentation problem and is able to produce high accuracy segmentation masks. In addition, with the development of deep learning, salient object detection models based on deep convolutional neural networks can be easily generalized to other related tasks, *e.g.* medical image segmentation [211], satellite/aerial image segmentation [181], by just replacing the training data. **Therefore, this thesis mainly focuses on salient object detection methods related to different applications and scenarios.**

1.3 Objective and Motivation

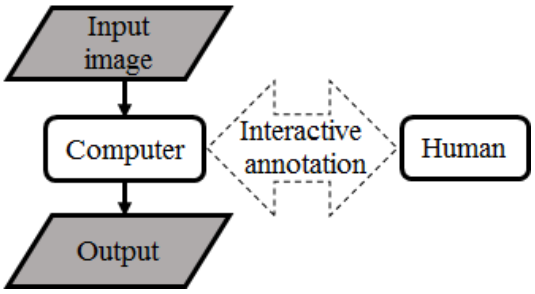
Salient object detection is an important yet challenging topic in computer vision. Generally, there is no unified mathematical definition of "salient" and the definition varies with different applications. Broadly speaking, "salient object" can be low-level features, *e.g.* corners [90], [212] and blobs [18], [164], middle-level elements, *e.g.* edges [32], [238] and line segments [3], [244], or higher-level targets represented by closed boundaries (contours) [253] or region masks [154]. Although the development of computer vision techniques achieves significant

(a) Centroid (b) Bounding box (c) Quadrilateral (d) Polygon (e) Region mask

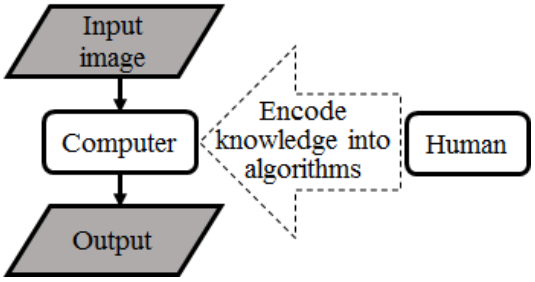
Figure 1.2. Different ways of image annotation.

improvements in the past decades, it is hard to find a general salient object detection method which is able to handle all of the different scenarios and applications. A salient object can be described by different representations, such as centroids [49], bounding boxes [80], quadrilaterals [302], polygons [37], [216], closed boundaries (contours) [35], [253] and region masks [151], [152], [163], as shown in Fig. 1.2. Among these representations, closed boundaries and region masks are the most accurate ones. In describing the two-dimensional (2D) targets, the closed boundaries and region masks are mutually determined and have the same accuracy. That means we can obtain the region masks from the given closed boundaries and vice versa. In addition, it is easy to derive the low accuracy representations, *e.g.* convex hull, bounding box and centroid, of a target from its extracted highly accurate closed boundary or region mask. **Therefore, we are motivated to develop methods for detecting salient objects and producing high-accuracy closed boundaries or region masks.**

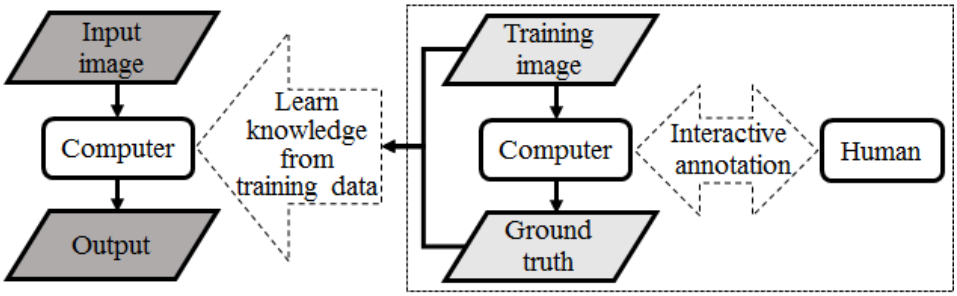
As described above, many methods [23], [24], [27], [258] have been proposed for salient object detection. They are mainly categorized into two groups: bottom-up and top-down. Bottom-up saliency models rely on characteristics of a visual scene [189], while top-down models [27] take more cognitive cues, *e.g.* knowledge, experiences, expectations and reward, into consideration. Recently, to apply the visual saliency to real applications, more and more methods [114], [281] employ both bottom-up and top-down saliency cues. Therefore, there comes the most frequently encountered problem in many computer vision tasks: **how to integrate the characteristics of the visual scenes (bottom-up cues) and the human knowledge (top-down cues) for**



(a) Direct use of human knowledge



(b) Explicit formulating of human knowledge



(c) Implicit encoding of human knowledge

Figure 1.3. Different ways of employing human knowledge for salient object detection.

improving the performance? In general, there are mainly three ways of employing human knowledge into salient object detection methods: (1) directly use human knowledge in the interactive process; (2) explicitly formulate human knowledge as the objective functions and the optimization processes and (3) implicitly encode human knowledge into the manually labeled training data. All of these three ways of employing human knowledge have both pros and cons. They can be utilized in different applications to satisfy diversified demands. **Therefore, the objective of this thesis is to explore new methods for salient object detection from the perspectives of these**

three ways:

- **Direct use of human knowledge.** Human knowledge can be directly applied to salient object detection by interactive annotation (see Fig. 1.3(a)). This way is usually utilized in those applications that require high accuracy and stable results. Many works have been proposed to improve the efficiency and accuracy, while decreasing the workloads and time costs. In this thesis, we first analyze the pros and cons of the existing image annotation tools. Then, we present a novel boundary based semi-automatic image annotation tool that we developed, ByLabel, to improve the annotation efficiency and accuracy by further simplifying the annotation operations.

- **Explicit formulating of human knowledge.** Human knowledge can also be explicitly formulated as the objective functions or the optimization processes in unsupervised salient object detection methods. In a narrow sense, unsupervised methods mainly refer to those methods like principal component analysis (PCA) methods [267] and clustering methods *e.g.* K-means [171], mean shift [46], etc. Broadly speaking, most of the traditional salient object methods can also be categorized as unsupervised methods. The detection processes of these methods are fully automatic, which require neither human intervention nor training data (see Fig. 1.3(b)). Hence, this kind of methods is possible to satisfy the automatic and high speed requirements of certain applications. In this thesis, the unsupervised methods for salient closed boundary extraction by perceptual grouping are studied and then they are adapted to different applications including building outline extraction and salient closed boundary tracking.

- **Implicit encoding of human knowledge.** Encoding human knowledge into the training data is another popular way, which is widely used in supervised methods. With the supervision of the training data, deep networks or other machine learning models can learn the embedded human knowledge for detecting salient objects (see Fig. 1.3(c)). Compared

with unsupervised methods, supervised salient object detection methods, especially deep models, are usually more robust and accurate thanks to the strong fitting capability of deep networks and large amounts of training data. However, there is still a large room for improvement in boundary quality, memory cost, speed, etc. Motivated by these factors, supervised salient object detection methods based on deep convolutional neural networks are also studied in this thesis.

1.3.1 Interactive Annotation with ByLabel: A Boundary based Semi-Automatic Image Annotation Tool

Interactive annotation is the most direct and easiest way of employing human knowledge for salient object detection. In many computer vision applications, human interventions usually produce more accurate and robust results. Therefore, interactive image annotation and labeling still play very important roles in various applications, *e.g.* cartography, accurate image/video editing, generation of segmentation ground truth, *etc.* However, interactive operations, especially those requiring high accuracy, are usually burdensome and time-consuming. For example, to label a given image with salient objects of complex shape, the operator has to localize and click a large number of control points to formulate the polygons for describing the object contours. Hence, a user-friendly annotation tool is necessary for reducing the workload and improving the efficiency. Motivated by that, we develop a novel boundary based semi-automatic image annotation tool, ByLabel. ByLabel enables operators to select and group detected high-quality edge fragments to formulate closed boundaries for describing objects with simple (one closed boundary) or complex (multiple closed boundaries) shapes. To be more general and flexible, our ByLabel is designed as a general image labeling and annotation tool, which is able to not only label binary class salient object masks but also produce multi-class and multi-instance segmentation masks.

1.3.2 Unsupervised Salient Closed Boundary Extraction by Perceptual Grouping

Although our ByLabel greatly reduces the workload of image annotation, it still requires many human interventions. Applications like building outline extraction and closed boundary visual tracking are relatively insensitive to few times of failures and minor spatial errors while requiring automatic processes and fast speed. That means human interactive salient object detection is not able to handle these kinds of applications. Therefore, faster automatic salient closed boundary extraction is needed. To automatically extract salient closed boundaries, we propose a novel method for unsupervised salient closed boundary extraction and reduce the problem into searching for a special cycle from an undirected graph. The key component of this method is a novel flexible graph-based optimization algorithm "Bi-Directional Shortest Path (BDSP)", which is able to search for the (approximated) "optimal" cycle from an undirected graph regardless of the formats of the objective functions.

Furthermore, we adapt this method to address the following two practical applications:

- Building outline extraction is an important issue in remote sensing and geomatics. Straight-line segment is one of the most salient middle-level features in aerial and satellite images. Closed boundaries (or polygons) formulated by multiple straight line segments are good at describing the building outlines in aerial or satellite images. Therefore, our closed boundary extraction method is adapted to extract building outlines by grouping the detected line segments based on our newly proposed grouping cost.

- Several targets like the rim of bowl or mug and cables are too thin to be described by regional representations *e.g.* bounding boxes or region masks. However, they can be described by boundaries. To track this kind of targets, we first adapt our salient closed boundary extraction workflow and optimization algorithm for salient closed boundary tracking via line segments perceptual grouping. Additionally, we apply this tracker for

real robot pouring task to validate its robustness. To further improve the accuracy and robustness of the salient closed boundary tracker, we replace the line segments by edge fragments, which are more accurate in describing smooth boundaries. Particularly, we study the method of fast high-quality edge fragments extraction including edge detection, edge breaking and edge filtering. In addition, the shape prior is encoded into a newly proposed grouping cost and both perimeter and area variation constraints are introduced to guide the grouping process for the salient closed boundary tracking.

1.3.3 Supervised Salient Object Detection by Deep Convolutional Neural Networks

Another way of utilizing human knowledge is to encode that into training data implicitly and then develop machine learning models or deep networks to learn the knowledge for salient object detection from the annotated training data. In recent years, the development of Deep Convolutional Neural Networks (DCNN) greatly raise the bar of salient object detection [168]. These methods formulate the salient object detection problem as a binary class image segmentation problem. Although DCNN greatly improves the accuracy and robustness, there is still a large room for improvement. One of the drawbacks of the existing deep salient object detection models is that their outputs usually have low boundary qualities. However, with the development of applications and demands, more and more accurate results are needed. Although the introduction of Conditional Random Field (CRF) [126] improves the overall segmentation performance, the improvement on boundary quality is limited and the CRF doubles the processing time. To address this problem, we propose a boundary-aware network, BASNet, and a hybrid loss for detecting salient objects with high-quality boundaries.

On the other hand, almost all of the deep salient object detection methods [147], [256], [289], [297] including our BASNet are built upon the backbones adapted from image classification networks, *e.g.* AlexNet [128], VGG [228], ResNet [94], DenseNet [100], *etc.* These backbones are originally designed for

image classification tasks and pre-trained on ImageNet [57] dataset. However, the logic of image segmentation and classification is different. For example, the network for image classification tasks prefer to extract more semantic information from deeper layers by sacrificing the resolutions of feature maps. In the segmentation tasks, both multi-level deep information and high resolution feature maps are essential to accurate segmenting. Besides, these backbones usually limit the running speed of the salient object detection models built on them because of their relatively large number of filters. Therefore, there is a need for novel flexible architectures specifically designed for salient object detection. Motivated by that, we design a deep architecture with two-level nested U-structure for real-time salient object detection without using any existing backbones adapted from image classification. It can be trained from scratch and is flexible to be configured to have different model sizes for training on datasets with different sizes or running on different working environments.

1.4 Contributions

In this thesis, we develop a set of different methods for salient object detection. Our contributions are summarized as follows.

First, we develop an interactive image annotation tool, ByLabel. It allows people to sequentially select detected edge fragments to formulate closed boundaries. One or multiple closed boundaries can be easily grouped to formulate the region masks. Compared with those polygon-based annotation tools, our ByLabel simplifies the burdensome control points localizing and clicking operations as edge fragments selecting operations. It greatly reduces the workload and time costs of labeling, especially for those targets with complex shapes. Besides, ByLabel is able to produce smooth and pixel-wise accurate region masks. Because the boundary fragments are one pixel-width pixel chains detected automatically and they are better than straight line segments of polygons in describing smooth curves.

Second, we develop a flexible graph-based optimization algorithm "Bi-Directional Shortest Path (BDSP)" for solving the problem of searching for

a special cycle from an undirected group. We further adapt the algorithm to two different applications:

- We define a novel saliency cost function which combines the proximity and continuity principles of Gestalt Laws for extracting individual building outlines from high resolution aerial images. Combined with our graph-based optimization algorithm BDSP, our method is able to extract individual building outlines with different shapes and achieves state-of-the-art performance against other unsupervised saliency detection methods.

- We adapt our BDSP for salient closed boundary tracking by line segments perceptual grouping. Particularly, we encode the area variation constraint into the objective saliency function and solve that by our graph-based optimization method BDSP. To validate the performance, we build a salient closed boundary tracking dataset. The experimental results show that our method is able to effectively handle those extremal cases, where stable and enough regional texture information is unavailable. Our method runs in real-time. In addition, we adapt our tracker to a real robot pouring task, which further demonstrates the reliable performance of our method. Since one pixel-width edge fragments are more accurate than line segments in describing smooth curves, we then further improve the above tracker by replacing the line segments with the high-quality edge fragments generated by our newly proposed edge breaking and filtering algorithms. Besides, we propose a novel grouping cost, which integrates a distance difference item and the area normalized total gap length item, and introduce both area and perimeter variation constraints to achieve more robust performance. The experimental results show that our edge-based new tracker greatly improves the tracking accuracy and robustness while keeping the real-time running speed.

Third, we develop two deep neural networks, **BASNet** and **U^2-Net**, for salient object detection.

- BASNet, as a boundary-aware salient object detection network, is designed to have a predict-define architecture. The predict-refine architecture consists of two encoder-decoder modules. The first encoder-decoder module is designed to take the images as inputs and output the coarse salient object probability maps. The refine module is a relatively smaller encoder-decoder with a residual connection, which is used to refine the coarse saliency probability maps. In addition, we define a novel hybrid loss that is the summation of binary cross entropy, structural similarity loss and intersection over union loss. These three components of the loss are used to perceive the pixel-wise, patch-wise and map-wise structures respectively. By using the newly designed predict-refine architecture and the novel hybrid loss, our BASNet (348.5 MB, 25 FPS) is able to produce robust and accurate salient object detection results with high quality boundaries.

- To achieve more flexible deep networks for salient object detection, we design a novel deep architecture with two-level nested U-structure, U^2-Net. Particularly, we first develop a residual U-block, which is able to be configured for extracting multi-scale information from the feature maps with different resolutions. Instead of stacking multiple residual U-blocks sequentially as U$\times n$-Net, we build our novel architecture by a nested way, U^n-Net (n=2). In this thesis, we provide two instances of this architecture: a full size U^2-Net (176.3 MB, 30 FPS) and a small size U^2-Net† (4.7 MB, 40 FPS), to serve for different scenarios. The full size model achieves state-of-the-art performance on three of the six public datasets and achieves very competitive performance on other three, which demonstrate the strong capability of our nested U-structure. The competitive results of the small version U^2-Net†, which is currently the smallest salient object detection model, further proves the flexibility and capability of our nested U-structure.

1.5 Organization

The rest of this thesis is organized as follows.

Chapter 2: Related Works

This chapter summarizes the related works in interactive image and video annotation, unsupervised salient closed boundary extraction and supervised deep salient object detection.

Chapter 3: Interactive Annotation with ByLabel: A Boundary based Semi-Automatic Image Annotation Tool

This chapter presents our interactive image annotation tool, ByLabel. We first introduce the basic types of different objects in terms of boundary based annotation. We then describe the design and implementation details of the newly developed annotation tool. Finally, we compare our ByLabel with a popular image annotation tool in terms of clicks, time costs, errors and user experiences.

Chapter 4: Unsupervised Salient Closed Boundary Extraction by Perceptual Grouping

In this chapter, we first introduce our newly proposed method for salient closed boundary extraction, especially a graph-based optimization algorithm for the optimal graph cycle searching. We then introduce the applications and detailed adaptations of this method in different areas including individual building outline extraction from aerial images, salient closed boundary tracking via line segments and edge fragments perceptual grouping. The experimental results and comparisons are presented in each of those application sections respectively.

Chapter 5: Supervised Salient Object Detection by Deep Convolutional Neural Networks

This chapter describes our newly designed deep networks for salient object detection. We first introduce our boundary-aware salient object detection network, BASNet, including the network design, the newly proposed hybrid loss and the ablation studies. Then, we describe our newly proposed deep nested U-structure, U^2-Net, and present its ablation studies. In the end, comprehensive experiments and comparisons with the state-of-the-art methods are illustrated.

Chapter 6: Conclusions and Future Work

This chapter discusses the strengths and limitations of our proposed methods. Some possible future works are presented in the end.

Chapter 2

Related Works

2.1 Interactive Salient Object Annotation

Interactive annotation is currently the most reliable and accurate approach for obtaining detection and segmentation results from given images or videos. As mentioned in Chapter 1, salient object detection (SOD) is essentially a binary class image segmentation problem, which is a sub-area of image segmentation. That means all of the image annotation tools designed for labeling image segmentation masks are able to be utilized for salient object annotation.

Interactive image annotation can be used in various applications, such as image/video editing, satellite/aerial image based mapping and so on. They can also be used in generating image and video ground truth for performance evaluation and comparison of different methods in various vision related tasks [113]. Besides, in machine learning and deep learning methods, images and their ground truth are indispensable to supervised training. Although many ground-truth datasets [51], [70], [143], [151], [196], [199], [215], [266] have been published, they are still few compared with the diversity of images and applications of interest in the real world. Fast and accurate image annotation remains an open problem in computer vision and related fields.

Image annotation tools seek to maximize labeling accuracy while minimizing human workloads and time costs [257]. Popular existing annotation tools can be categorized into three main classes: (1) bounding box/quadrilateral based labeling; (2) pixel-wise labeling; (3) boundary based labeling. This thesis focuses on the boundary based annotation.

A simple bounding box, defined by two corners (top-left and bottom-right), is usually used in object recognition [70] and two degree of freedom (DoF) tracking [127]. In registration based high DoF tracking, a quadrilateral, which is defined by four corners, describes planar geometric transformations, such as rotation, affine and homography [77], [132], [150], [213]. Bounding boxes labeling is easy to implement. Doermann and Mihalcik develop ViPER, a video annotation tool which allows users to perform annotation frame by frame [60]. Vondrick *et al.* design a crowdsource video labeling tool, VATIC, which introduces inter frames interpolation to generate bounding boxes semi-automatically [245].

Compared to bounding boxes, pixel-wise labeling provides detailed shape descriptions of target objects [266]. Graph cuts [28], watershed segmentation [121], active contour [257], partition trees [78] and even mixed techniques [174], [177] have been introduced to decrease the need for user intervention. When to-be-annotated images have salient foregrounds and relatively flat backgrounds, these methods perform well. Otherwise the region masks produced are often inaccurate and noisy.

Closed boundaries are used in many annotation tools, such as KAT [218], PhotoStuff [86], M-Ontomat Annotizer [198] and iVAT[20]. However, closed boundaries are usually approximated by polygons in these tools, which reduces the accuracy in describing the smooth curves. Russell [216] develop a web-based image annotation tool, LabelMe, based on manual polygon drawing. Lluis *et al.* [37] use Recurrent Neural Networks (Polygon-RNN) to reduce human intervention in polygon annotation. The annotation accuracy depends on the number of the sampled control points and their localization errors. First, densely sampled control points are required to describe curved boundaries accurately. Second, human labelers have to localize each boundary pixel very accurately [28]. These two factors limit the annotation efficiency. To address these problems, Yang *et al.* [282] propose a constrained random walk algorithm, which combines the unified combinatorial user inputs, to obtain relatively smooth segmentations. Maji *et al.* [173] refine the manually labeled coarse polygons using random maximum a posteriori (MAP) perturbations.

However, their results usually have similar problems as the aforementioned pixel-wise annotation methods. Their accuracy is easy to be influenced by noisy intensities and complex local patterns.

Another category of image annotation tools takes advantage of the image boundary features. They allow users to select those automatically detected boundaries for formulating contours of to-be-annotated targets. Mortensen and Barrett [182] develop an interactive segmentation tool, intelligent scissors, which allow users to select piece-wise boundaries generated by graph searches on the over-segmented images from toboggan-based method. Elder *et.al.* [64] build an interactive tool for image editing in the contour domain, which allows users to select specific boundary pixels and then formulates the intact to-be-annotated boundaries by one-dimensional (1D) and two-dimensional (2D) grouping on those automatically detected edges. These tools are more accurate than the polygon-based labeling tools in describing curved boundaries. Besides, both of them introduce shortest path search algorithm to reduce the human intervention for boundary selection.

To achieve highly accurate annotations while minimizing user interventions, we present an edge fragment based annotation tool, ByLabel. Given an image, edges are detected and split into high-quality fragments. Compared with manually sampled polygon control points, those detected edge fragments describe the curved boundaries more accurately. Then, users are allowed to create closed boundaries by selecting subsets of those edge fragments sequentially. The selection operation requires no careful localization of boundaries, hence it greatly reduces the annotation workloads. One or multiple boundaries are grouped to describe different types of objects including simple objects with single boundary, complex objects with holes or divided by occlusions. Finally, region masks are generated and output based on the corresponding closed boundary groups.

2.2 Salient Closed Boundary Extraction

Closed Boundary (CB) is one of the most common elements in natural images. Object contours are usually comprised of one or multiple closed boundaries. Each closed boundary consists of a set of sequentially connected pixels with high gradient magnitude. Salient Closed Boundary (SCB) is defined as the most attractive closed boundary in certain scenarios. Although the "most attractive" is sometimes confusing and not unique, it can be determined by some additional conditions and constraints. Given an image with/without additional conditions, we refer the process of identifying the most attractive closed boundary as salient closed boundary extraction.

As mentioned in Chapter 1, closed boundaries and regions are mutually determined. Hence, both boundary extraction methods and image segmentation methods are able to produce closed boundaries. Many approaches have been proposed for solving this problem. These methods can be mainly categorized into the following groups: (1) active-contour approaches; (2) shape deformation approaches; (3) graph-theoretic approaches; (4) perceptual grouping approaches.

Active-contour approaches extract the target closed boundaries by evolving the initialized closed boundaries based on image energy minimization. Kass *et al.* [117] propose the snakes method, in which snake is an energy-minimized spline guided by external constraints and image forces that enforce the spline to evolve toward the local lines and edges. Amini *et al.* [8] introduce dynamic programming that tries to guarantee the global optima of the variational energy minimization process for active contour problem. Williams and Shah [265] propose a fast active contour method based on a greedy algorithm which is faster than the dynamic programming based method in [8]. Caselles *et al.* [34] propose an active contour model based on a geometric partial derivative equation (Level-set method), which is independent to curve parameterization and enables the extraction of multiple contours at the same time. Caselles *et al.* [35] develop a geodesic active contour method which builds the connection between classical "snakes" based on energy minimization and geometric active

contours based on the theory of curve evolution. Xu and Prince [276] improve the active contour methods by introducing an external force called gradient vector flow (GVF), which is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image. Although the active contour methods achieve significant progress in the past decades, their results heavily depend on the initialization and the global optima is not guaranteed in most of the applications [253].

Shape deformation approaches are developed to achieve more robust performance against noises and partial occlusions. Cootes *et al.* [50] propose to learn patterns of variability from annotated training images and then combine the learned patterns with active contour models to achieve robust results by iterative refinement algorithm. Staib [234] introduce a probabilistic deformable model to describe the flexible constraints for representing diversified and irregular boundaries (shapes). The boundary finding problem is then solved by optimizing a maximum a posteriori objective function. Jain *et al.* [110] develop an object matching method using deformable templates, which consist of representative contours or edges. They utilize a Bayesian scheme to find a match between the deformed template and objects in the image based on this prior knowledge and the edge information in the input image. Han *et al.* [87] propose a topology preserving Level Set method for geometric deformable models, in which topology preservation is obtained by applying the simple point concept from digital topology. Leventon *et al.* [140] present a method for incorporating shape information into the image segmentation process. They predict the target location by estimating the maximum a posteriori position globally and refine the shape of the object based on the prior shape information and the local image information. Although these shape deformation approaches are able to handle certain noises and specifical types of partial occlusions, their performance is still sensitive to the initialization and there is no guarantee that the optimal boundary will be obtained [253].

Graph-theoretic approaches are developed to avoid dependence on initialization and guarantee the production of global optima. Wu *et al.* (Minimum Cut) [273] propose to represent to-be-clustered data (image pixels) by an

undirected adjacency graph G, in which the arcs represent similarities between two data points. The vertices (pixels) of G are then clustered by minimizing the largest inter-subgraph maximum flow. Cox *et al.* (Ratio Regions) [53] define the segmentation objective function as the combination of an exterior boundary cost and an interior benefit associated with it. They optimize the function by a graph partitioning algorithm. Shi and Malik (Normalized Cut) [225] formulate the image segmentation problem as a graph partitioning problem and solve this problem by their newly proposed normalized cut criterion, which measures both the total inter-group dissimilarity as well as the total intra-group similarity. Sarkar and Soundararajan (Average Cut) [220] propose a supervised strategy to learn the relative importance of the basic salient relationships and the grouping parameters for perceptual grouping. Jermyn and Ishikawa [111] define a new form of energy functional on the space of boundaries for modeling and identification of regions of images. This functional is able to incorporate general combinations of information both from the boundary and from the interior of the region. The global optima is searched by two polynomial-time digraph algorithms. Wang and Siskind (Ratio Cut) [254] propose a new image segmentation cost named as cut ratio, which is defined as the ratio of the corresponding sums of two different weights of edges along the cut boundary and models the mean affinity between the segments separated by the boundary per unit boundary length. This cost guarantees that the segments produced by bipartitioning are connected and aligned with image edges. The above graphs constructed in most of those methods are based on vertices correspond to pixels or small regions so that it is difficult to incorporate several Gestalt laws [246], [253]. In addition, most of the graph optimization algorithms are specifically designed for solving certain types of objective functions so that they are limited to be adapted to different applications.

Perceptual grouping approaches focus on identifying and grouping low-level features *e.g.* pixels, line segments or edge fragments, to formulate high-level targets. Most of the perceptual grouping approaches are designed based on the principles of Gestalt Laws [115], [124], [246]. One class of the grouping approaches [9], [36], [38], [54]–[56], [186], [208] targets at grouping dis-

joint points/edges into relatively longer open boundaries. Another class aims at generating closed boundaries to facilitate the detection or segmentation of certain targets. This class of algorithms [10], [66], [68], [109], [172], [233], [253] is relatively more applicable in real-world applications, such as object detection, image segmentation, *etc.* Elder and Zucker [68] demonstrate that contour closure plays an important role in the rapid discrimination of two-dimensional shapes. Jacobs [108], [109] propose an algorithm for extracting salient convex collections of line segments by encoding proximity and convexity into distance constraint and angle consistent. Elder and Zucker [66] propose an algorithm for extracting highly closed bounding contours. They use tangent vectors and image intensity to represent the contours and build a sparsely-connected graph based on this representation. Then, they pose the problem of computing closed contours as the computation of shortest-path cycles in this graph. Amir and Lindenbaum [10] introduce a generic method for perceptual grouping of various types of data features. This method first constructs a graph based on available grouping cues and then conducts graph partition to find feature groups using known statistical tools such as Wald's SPRT algorithm [247] and the Maximum Likelihood criterion. Mahamud *et al.* [172] develop a real-time image segmentation method using a saliency measure based on the proximity and good continuity principles of Gestalt laws. Specifically, the contour closure is incorporated by finding the eigenvectors and eigenvalues of a transition matrix, which describes a Markov process. Each element (**i**,**j**) of the matrix denotes the conditional probability that edge **j** belongs to a contour and edge **i** will also be a part of the contour. To achieve more robust results, Elder *et al.* [65] propose to group contours by combining prior probabilistic knowledge of the object appearance with probabilistic contour grouping models. Particularly, they first search for a set of closed boundary candidates and then evaluate these candidates by computing the maximum a posteriori based on figure-ground and prior probabilities. Wang *et al.* (Ratio Contour, RC) [253] formulate the salient closed boundary problem as a problem of searching for a graph cycle from an undirected graph. They solve this problem by reducing it into a problem of finding a Minimum-Weight Perfect Matching (MWPM) in

the same graph. Stahl and Wang [233] adapt Ratio Contour method by implicitly encoding the continuity principle into the area of the region enclosed by the target closed boundary. To solve the problem, they specifically designed a method for attaching the area information to each line segments.

As described above, many algorithms first construct a graph based on the principles of Gestalt laws and then search for the closed boundaries from the graph. The key component of these perceptual grouping methods is the optimization algorithm. Since applications of perceptual grouping are diversified, a well-designed optimization algorithm will enable the development of more flexible and accurate grouping objective functions according to the requirements of different applications. In this thesis, a simple yet flexible graph-based optimization algorithm named as "Bi-Directional Shortest Path (BDSP)" is developed. This algorithm first generates a set of cycle candidates by searching for the shortest paths from two endpoints of a graph edge to a third common vertex. Then, it searches for the optimal cycle from these generated cycle candidates by evaluating them based on the grouping cost functions and priors. This algorithm is able to handle different objective functions and flexible to be used in different perceptual grouping related applications.

2.3 Supervised Salient Object Detection by Deep Convolutional Neural Networks

In addition to extracting salient closed boundaries, labeling pixels belonging to the salient object is another way of salient object detection. This problem is usually formulated as a binary image segmentation problem. Early traditional methods detect salient objects by searching for pixels according to a pre-defined saliency measure computed based on handcrafted features [231], [279], [294], [310]. Borji *et al*. [24] provide a comprehensive survey and analysis of these traditional methods in. In recent years, many deep salient object detection networks [258] have been proposed. Compared with traditional handcrafted features based methods [24], deep salient object detection networks show more competitive performance. These deep methods can be

categorized into the following groups:

(1) Patch-wise Deep Methods: Encouraged by the advancement of image classification of Deep Convolutional Neural Networks (DCNNs) [128], [228], early deep salient object detection methods search for salient objects by classifying image pixels or super pixels into salient or non-salient classes based on the local image patches extracted from single or multiple scales [145], [147], [158], [250], [307]. However, these methods usually generate coarse outputs most probably because spatial information is lost in the fully connected layers.

- Zhao *et al.* (**MCDL**) [307] propose to extract both local and global context information of a superpixel by feeding the superpixel centered image patches with different sizes into two unified networks. Then each superpixel is predicted to be salient or not.

- Wang *et al.* (**LEGS**) [250] develop a two stages saliency detection framework. Local context information are extracted in the first stage from image patches by a deep neural network and the global saliency scores are predicted based on these local features via a fully connected network.

- He *et al.* (**SuperCNN**) [96] propose to predict per-superpixel saliency by feeding superpixel based multi-scale color uniqueness and distribution sequences into convolutional networks.

- Lee *et al.* (**ELD**) [139] try to predict per-superpixel saliency by concatenating the deep image feature vectors extracted by VGG [228] and the feature vectors of encoded low-level distance map of the super-pixel query region.

- Kim and Pavlovic (**SCSD**) [119] develop a shape-preserving saliency prediction method, which uses a convolutional neural network to learn the correspondences between image patches and a set of shape classes from a pre-defined shape dictionary.

- Li *et al.* (**MDF**) [147] propose to feed multi-scale image patches around

a target pixel to a neural network and then obtain a feature vector for describing the saliency of this pixel.

(2) FCN-based Methods: Salient object detection methods based on Fully Convolutional Network (FCN) [130], [146] achieve significant improvement compared with patch-wise deep methods, presumably because FCN is able to capture richer spatial and multi-scale information. Most of the methods in this category usually focus on developing new strategies or methods for better aggregating or integrating multiple deep layers' feature maps extracted by backbone networks adapted from image classification, such as VGG, ResNet [94], DenseNet [100] and so on.

- Zhang *et al.* (**UCF**) [301] develop a reformulated dropout and a hybrid upsampling module to reduce the checkerboard artifacts of deconvolution operators as well as aggregating multi-level convolutional features in (**Amulet**) [300] for saliency detection.

- Hu *et al.* (**DLS**) [98] propose to learn a Level Set [190] function by a deep convolutional neural network to output accurate boundaries and compact saliency.

- Luo *et al.* [168] design a network (**NLDF**+) with a 4×5 grid structure based on VGG network to combine local and global information and utilize a fusing loss of cross entropy and boundary IoU inspired by Mumford-Shah [185] to achieve clear saliency boundaries.

- Hou *et al.* (**DSS**+) [97] adopt Holistically-Nested Edge Detector (HED) [275] by introducing short connections to its skip-layers for saliency prediction.

- Chen *et al.* (**RAS**) [43] adopt HED by refining its side-output iteratively using a reverse attention model.

- Zhang *et al.* (**LFR**) [299] predict saliency with clear boundaries by proposing a sibling architecture and a structural loss function.

- Zhang *et al.* (**BMPM**) [297] design a controlled bi-directional passing of features between shallow and deep layers to obtain accurate saliency predictions.

- Wu *et al.* (**MLMS**) [270] improve the saliency detection accuracy by developing a novel Mutual Learning Module for better leveraging the correlation of boundaries and regions.

- Wu *e.g.* (**CPD**) [272] propose to use Cascaded Partial Decoder framework for fast and accurate salient object detection.

(3) Deep Recurrent and Attention Methods: Instead of designing feature aggregation strategies or integration approaches, methods in this category aim to extract richer local and global context information from the feature maps generated by backbone networks adapted from image classification tasks. Hence, different multi-scale feature extraction modules are designed and integrated with those backbone networks.

- Kuen *et al.* (**RACDNN**) [131] learn global saliency information by introducing a recurrent network to iteratively perform feature extraction and refinement on selected image sub-regions.

- Zhang *et al.* (**PAGRN**) [303] develop a recurrent saliency detection model that transfers global information from the deep layer to shallower layers by a multi-path recurrent connection.

- Hu *et al.* (**RADF**+) [99] recurrently concatenate multi-layer deep features to capture richer local and global context information for salient object detection.

- Wang *et al.* (**RFCN**) [252] design a recurrent FCN-based encoder-decoder for iteratively correcting the saliency prediction errors.

- Liu *et al.* (**PiCANet**) [157] predict the pixel-wise attention maps by a contextual attention network and then incorporate it with U-Net architecture to detect salient objects.

- Zhang *et al.* (**CapSal**) [298] design a local and global perception module to extract both local and global saliency information from features extracted by the backbone network.

- Zeng *et al.* (**MSWS**) [289] design an attention module to predict the spatial distribution of foreground objects over image regions meanwhile aggregate their features.

- Feng *et al.* (**AFNet**) [74] develop a global perception module and attentive feedback modules to better explore the structure of salient objects.

- Liu *et al.* (**PoolNet**) [154] develop an encoder-decoder architecture for salient object detection by introducing a global guidance module for extraction of global localization features and a multi-scale feature aggregation module adapted from pyramid pooling module for fusing global and fine-level features.

(4) Coarse to Fine Deep Methods: To capture finer structures and more accurate boundaries, numerous refinement strategies including progressive refinement modules and hybrid losses have been proposed.

- Liu *et al.* (**DHSNet**) [155] propose a deep hierarchical saliency network which learns various global structured saliency cues first and then progressively refine the details of saliency maps.

- Wang *et al.* (**SRM**) [255] propose to capture global context information with a pyramid pooling module and a multi-stage refinement mechanism for saliency maps refinement.

- Inspired by [195], Amirul *et al.* (**CARNet**) [103] propose an encoder-decoder network that utilizes a refinement unit to recurrently refine the saliency maps from low resolution to high resolution.

- Li *et al.* (**C2S**) [148] develop a two branches contour-to-saliency network, in which contour and saliency features are encoded in the same bottom layers. The contour detection and SOD branches are trained iteratively to progressively refine each other.

- Deng *et al.* (**R**^{3}**Net**+) [58] develop a recurrent residual refinement network for saliency maps refinement by incorporating shallow and deep layers' features alternately.

- Wang *et al.* (**DGRL**) [256] propose to localize salient objects globally and then refine them by a local boundary refinement module.

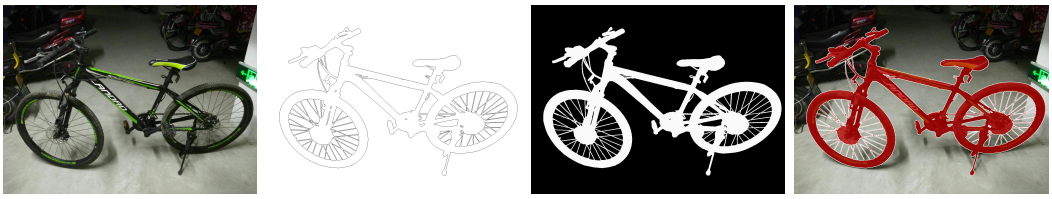
Although these methods raise the bar of salient object detection greatly, there is still a large room for improvement in terms of the fine structure segment quality and boundary recovery accuracy. Besides, almost all of the above deep salient object detection methods are built upon the existing backbone networks adapted from image classification tasks like ImageNet [57]. On one hand, the usage of these backbones does lower the risks of developing new SOD models and improve the overall performance of SOD. On the other hand, those backbones were originally designed for image classification tasks and can not be directly generalized for SOD. To overcome this drawback, more and more strategies and modules are introduced and result in overly complicated SOD models. The aforementioned two main problems motivate us to develop new deep convolutional networks for accurate, fast and robust salient object detection. Therefore, we first present a new deep method, **BASNet**, for boundary-aware accurate salient object detection. Then, a computation and memory efficient deep architecture **U**^{2}**-Net** is proposed for real-time salient object detection.

Chapter 3

Interactive Annotation with ByLabel: A Boundary based Semi-Automatic Image Annotation Tool

3.1 Overview

This chapter presents a novel boundary based semi-automatic tool, ByLabel, for accurate image annotation, as shown in Fig. 3.1. Given an image, ByLabel first detects its edge features and computes high-quality boundary fragments. Current labeling tools require the human to accurately click on numerous boundary points. ByLabel simplifies this to just selecting among the boundary fragment proposals that ByLabel automatically generates. To evaluate the performance of ByLabel, 10 volunteers, with no experience of annotation, were asked to label both synthetic and real images with both our ByLabel and another commonly used labeling tool, LabelMe [216]. Compared to LabelMe, our ByLabel reduces image click operations and time costs by 73% and 56% respectively, while improving the accuracy by 73% (from 1.1 pixel average boundary error to 0.3 pixel). The results show that our ByLabel outperforms the state-of-the-art annotation tool in terms of efficiency, accuracy and user experience. The tool is publicly available: `http://webdocs.cs.ualberta.ca/~vis/bylabel/`.



(a) Input image (b) Edge map (c) Region mask (d) Image and mask overlap

Figure 3.1. Annotation of a bike. Given an input image (a), ByLabel computes boundary proposals and allows humans to select and group closed boundaries. Then, the region masks (c) are generated from these closed boundaries (b).

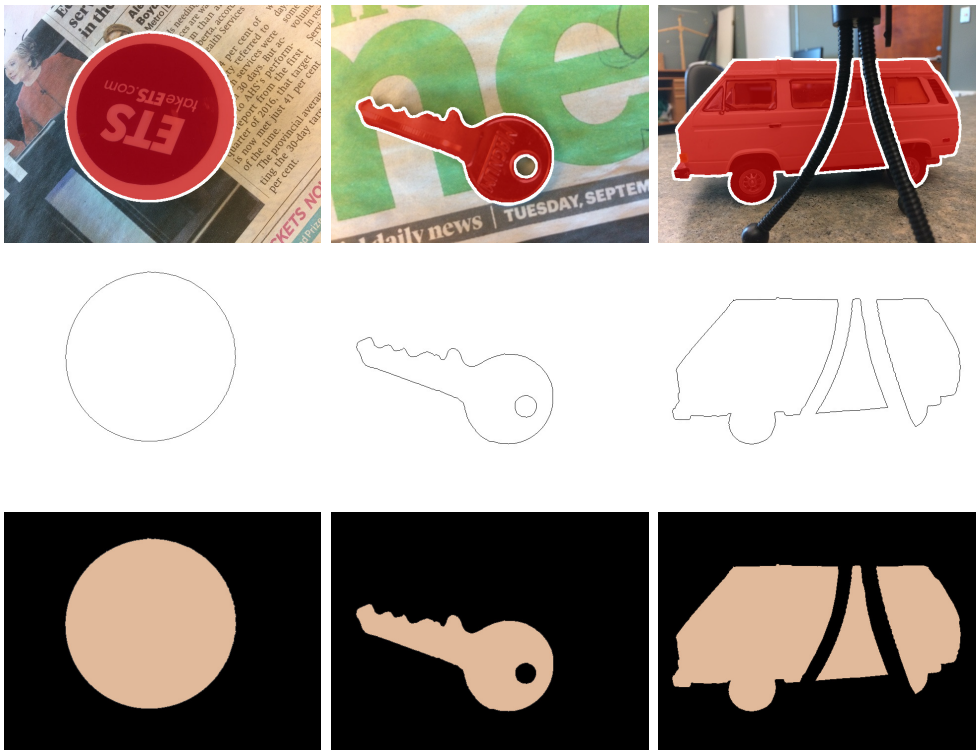
3.2 Basic Types of To-Be-Annotated Objects

Boundary maps and region masks are the two most commonly used image ground truth types. The goal of ByLabel is to produce these ground truth annotations semi-automatically by interacting with a human user. We categorize those to-be-annotated objects into three basic types:

- simple objects, which can be defined by one closed boundary or a piece of the contiguous region, as shown in Fig. 3.2(a).
- objects with holes, which can be described by several nested closed boundaries or a piece of the contiguous region, as shown in Fig. 3.2(b).
- objects divided by occlusions, which can be determined by multiple closed boundaries or isolated regions, as shown in Fig. 3.2(c).

As we can see in Fig. 3.1 and Fig. 3.2, holes and occlusions often exist. Multiple boundaries or regions are necessary to describe these kinds of complicated targets. We take closed boundaries and the regions enclosed by them as "dual" representations of targets, since they can be determined by each other. Hence, we can label either boundaries or regions and generate the other one automatically. ByLabel is designed to label boundaries. Region masks are generated from the labeled boundaries.

To achieve high annotation accuracy and efficiency, we propose to use automatically detected edge fragments instead of the manually sampled control



(a) Circular box (b) Key (c) Toy van

Figure 3.2. Three basic types of objects. Column (a) is a circular box (one closed boundary). Column (b) is a key with a hole (two boundaries). Column (c) is a partially occluded toy van (three boundaries). The top, middle and bottom row are their original images, edge maps and region masks respectively.

points to describe object boundaries. Given an image, annotating a new object involves three steps, as illustrated in Fig. 3.3. First, the edge features of the image are detected and split properly. Then, users are allowed to annotate boundaries interactively with a well-designed keyboard and mouse interface. Finally, grouped boundaries, generated region masks and input class names are organized and formatted in an output fold to represent the objects.

3.3 Feature Detection

Edge fragments (EF) are basic elements in our annotation process. They are obtained by splitting detected edge segments (ES). There are many algorithms have been proposed for edge detection. One the most famous edge detectors is Canny [32]. But the edges detected by Canny are often jittered and dis-

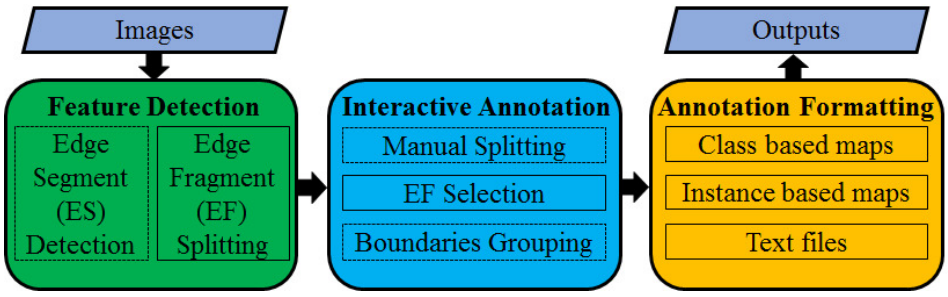
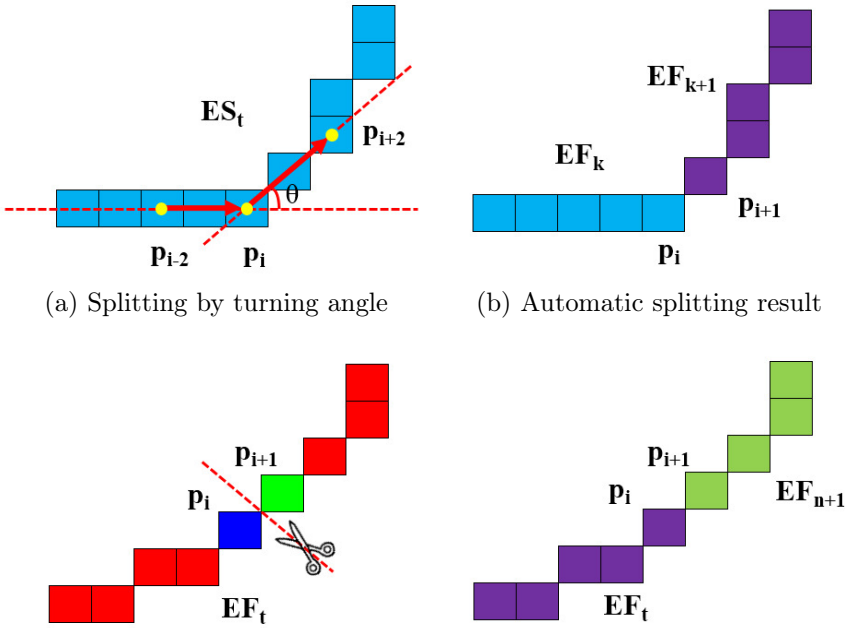


Figure 3.3. Annotation workflow of ByLabel.

continuous. Therefore, many multi-scale [67] or learning based [61], [275] edge detectors were proposed to obtain better edge detection results by taking image scale and structure information into consideration. However, the outputs of these methods are usually binary masks (or probability maps) and the edge pixels are independent. To have continuous pixel chains, these independent edge pixels are usually grouped into edgels by additional grouping processes. That means specific grouping costs and optimizers have to be developed. Conducting these two steps separately avoids "edge breaking" but introduces other problems, such as how to design a reliable grouping costs and guarantee the smoothness of the locally grouped edges; how to exclude noisy edge pixels and avoid erroneous grouping; etc. In this work, Edge Drawing (ED) [238], which is able to produce well-localized, clean, contiguous and one-pixel wide edge segments, are utilized. Given an image, ED first detects its anchor edge pixels and then searches for other edge pixels and groups them into edge segments simultaneously. The integration of edge pixels extraction and grouping is able to produce relatively smoother and less noisy edge segments. Each (chain-wise) edge segment detected by ED is output in vector form as an array of pixels. Compared with manual control points, these pixel chains are smoother and more accurate in fitting object boundaries. However, in the automatically detected fragments some foreground and background edge pixels are often improperly identified as one long and complex edge segment. Our annotation tool employs two "edge breaking" ways for splitting an edge segment into multiple well-organized edge fragments, as illustrated in Fig. 3.4: (1) automatic splitting using turning angle rules; (2) manual splitting.



(a) Splitting by turning angle (b) Automatic splitting result

(c) Manual splitting (d) Manual splitting result

Figure 3.4. Edge segment splitting. (a) and (b) show the automatic splitting of an Edge Segment (ES) based on turning angle. As illustrated in (a), if the turning angle of the pixel $\mathbf{p_i}$ is larger than a threshold (35 degree), the edge segment will be split between pixel $\mathbf{p_i}$ and $\mathbf{p_{i+1}}$. (c) and (d) illustrate the manual splitting of an Edge Fragment (EF). The edge fragment is split between the blue and green pixels.

Pixels which have large curvatures are more likely to be incorrect connections of foreground and background edges. Here, we search for these pixels using a simple measure, turning angle θ, which is computed as:

$$\theta_{\mathbf{p_i}} = arccos(\frac{(\mathbf{p_i} - \mathbf{p_{i-2}}) \cdot (\mathbf{p_{i+2}} - \mathbf{p_i})}{|\mathbf{p_i} - \mathbf{p_{i-2}}||\mathbf{p_{i+2}} - \mathbf{p_i}|}). \tag{3.1}$$

where $\mathbf{p_i}$ is the current pixel, $\mathbf{p_{i-2}}$ and $\mathbf{p_{i+2}}$ are two pixels sampled near $\mathbf{p_i}$. $\theta_{\mathbf{p_i}}$ is the angle between vector $\overrightarrow{\mathbf{p_{i-2}p_i}}$ and $\overrightarrow{\mathbf{p_ip_{i+2}}}$, as illustrated in Fig. 3.4(a). All of the detected edge segments are split at pixels where turning angles are larger than a certain threshold (set to 35 throughout our experiments), as shown in Fig. 3.4(b). Compared with line fitting-based edge splitting methods, this method retains relatively long smooth edge curves, and hence prevents over splitting.

Sometimes, foreground and background edges also have smooth connec-

tions, leading to a boundary that cannot be inferred from the turning angle. To handle this case, ByLabel allows manual splitting. Users can split an edge fragment by moving the mouse cursor over the expected splitting position and pressing key "b". As shown in Fig. 3.4(c) and Fig. 3.4(d), the expected splitting position is indicated by two pixels: the blue pixel $\mathbf{p_i}$, which is the closest one to the mouse cursor, and the green pixel $\mathbf{p_{i+1}}$, which is the next one of $\mathbf{p_i}$ in the vector of edge fragment $\mathbf{EF_t}$.

3.4 Interactive Annotation

In ByLabel, one or multiple closed boundaries are employed to describe an object of arbitrary shape. Closed boundaries are defined by sequentially connected multiple edge fragments. In the annotation process, detected edge fragments are superimposed on the original image, and users sequentially select a subset of them to form the object boundaries. If there is a small gap, two successively selected edge fragments are connected with their two closest endpoints by a short straight line segment that is automatically added. Sometimes, not all of the object boundaries can be detected successfully. To address this problem, ByLabel provides a "drawing" mode. Users can switch the labeling mode between "selecting" and "drawing" by press key "a". Similar to LabelMe, the "drawing" mode allows users to manually enter multiple control points to fit those missing arcs. After selecting all necessary fragments, users can click the middle button of the mouse (mouse wheel) to finish the labeling and close the boundary. The simple process of annotation is shown in Fig. 3.5.

An object can consist of multiple boundaries. To group these boundaries into an object entity, users have to label them successively. The annotation processes in ByLabel are sequentially ordered. After completing a closed boundary, there will be a pop-up window asking whether the labeled boundary is the last one of the current object. The input of "n" means the object has more boundaries to label. The input of "y" means the current object labeling is finished and there will be another pop-up window for inputting the object's

(a) Edge Fragments (EF) (b) Labeled boundary (c) Region mask

Figure 3.5. Interactive annotation. (a) shows the detected edge fragments. (b) shows the labeled boundary: red edges are selected EF; green edges are generated connections; pink edges are drawn segments. (c) is the region mask generated from the labeled boundary.

class name or identity.

3.5 Annotation Formatting

There are mainly seven types of annotation outputs stored in their corresponding folders: (1) *color_im_overlap*, (2) *edge_map_classes*, (3) *edge_map_instances*, (4) *region_map_classes*, (5) *region_map_instances*, (6) *text_EF_pixels*, (7) *text_shape_pixels*. The pixel coordinates of detected edge fragments are output into *text_EF_pixels*. The direct results of interactive annotation are boundaries represented by multiple edge fragments. These boundaries are all one-pixel-wide. They are written in text files (*text_shape_pixels*) and drawn as boundary maps. In addition to label boundaries, ByLabel is also able to generate region masks according to labeled boundaries. Here, both edge maps and region masks are output as two types of color images: class based (*edge_map_classes*, *region_map_classes*) and instance based (*edge_map_instances*, *edge_map_instances*). Users are free to choose any type of

output depending on the application. More details and instructions can be found: `https://github.com/NathanUA/ByLabel`.

3.6 Experiments

To demonstrate the advantages of our boundary based annotation tool ByLabel, we conduct a set of user tests to quantitatively compare its performance with that of a popular web-based annotation tool LabelMe. Additionally, we show the results of some typical annotation cases to demonstrate the effectiveness of ByLabel qualitatively.

3.6.1 User Tests and Evaluation

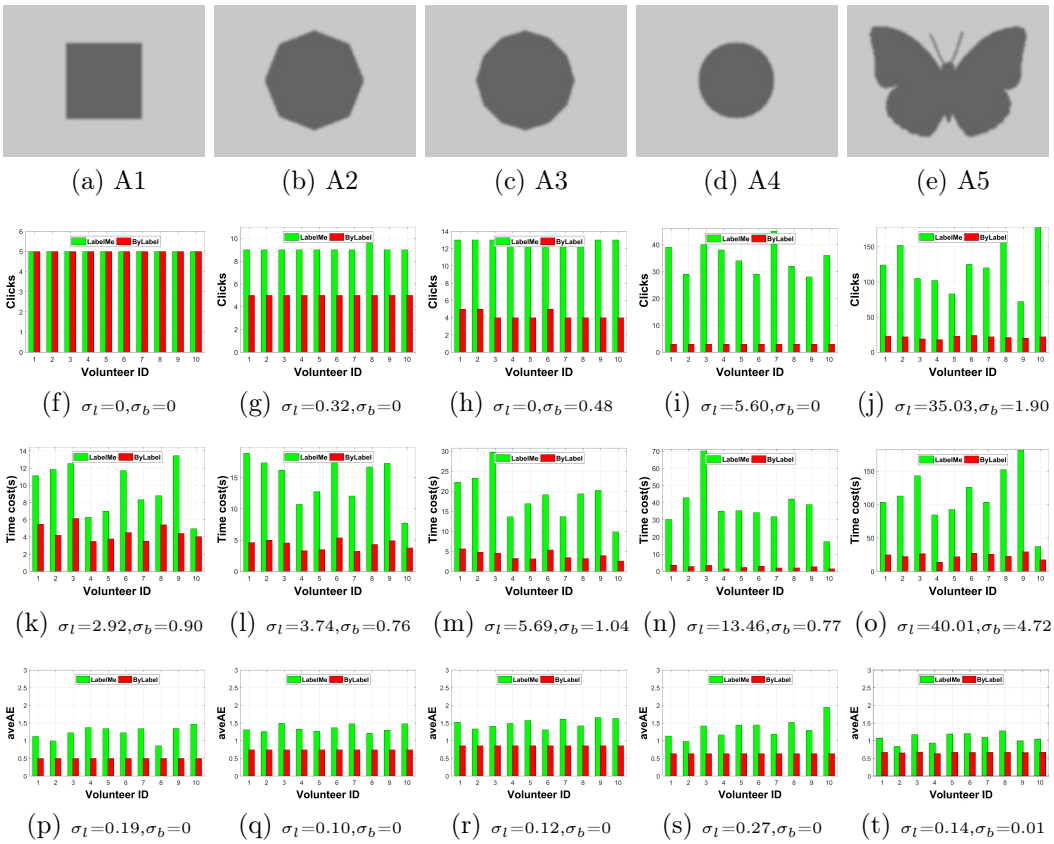
In our user experiments, ten volunteers are asked to annotate five synthetic images (Fig. 3.6), and ten real images (Fig. 3.7 and Fig. 3.8), as fast and accurate as possible using both LabelMe and ByLabel. These ten volunteers have no prior experiences on image annotation. To reduce the total work load of the annotation test, each testing image contains only one target object, which is defined by a single boundary.

We evaluate ByLabel and compare it with LabelMe on the following four aspects: *clicks*, *time costs*, *error* and *user experience*. Mouse clicks are the most commonly used operation in image annotation. Generally, more *clicks* produce more detailed annotation results and higher geometric accuracy. However, a large number of clicks require more patience and time. Therefore, *clicks* and *time costs* reflect the annotation work load directly. The geometric annotation *error* is defined by the average Alignment Error ($aveAE$) [204]

$$aveAE = \frac{B_{usr} \otimes Dist_{gt}}{P_{usr}} \tag{3.2}$$

where $\otimes$ indicates the summation of element(pixel)-wise multiplication, B_{usr} is the annotated binary boundary map, $Dist_{gt}$ is the distance map of the ground truth, P_{usr} is the edge pixels' number of B_{usr}.

In our tests, we recorded users' *clicks*, *time costs* and *errors* on each testing image, as shown in Fig. 3.6, Fig. 3.7 and Fig. 3.8. After annotating, they



(a) A1 (b) A2 (c) A3 (d) A4 (e) A5

(f) σ_l=0,σ_b=0 (g) σ_l=0.32,σ_b=0 (h) σ_l=0,σ_b=0.48 (i) σ_l=5.60,σ_b=0 (j) σ_l=35.03,σ_b=1.90

(k) σ_l=2.92,σ_b=0.90 (l) σ_l=3.74,σ_b=0.76 (m) σ_l=5.69,σ_b=1.04 (n) σ_l=13.46,σ_b=0.77 (o) σ_l=40.01,σ_b=4.72

(p) σ_l=0.19,σ_b=0 (q) σ_l=0.10,σ_b=0 (r) σ_l=0.12,σ_b=0 (s) σ_l=0.27,σ_b=0 (t) σ_l=0.14,σ_b=0.01

Figure 3.6. The results of user tests on synthetic images. The top row shows the synthetic images. The second to fourth row show the *clicks*, *time costs* and *average Alignment Error (aveAE)* respectively. σ_l and σ_b are standard deviations of LabelMe and ByLabel.

were also asked to fill the forms of NASA Task Load Index (TLX) [91] for *user experience* evaluation (see the results in Fig. 3.10).

1) Tests on synthetic images

Image annotation are usually simplified as polygon drawing. As mentioned above, there are two factors that affect the annotation work load. One is the number of boundary control points, which determines the required *clicks* along the boundary. The other one is the boundary saliency. Here, the boundary saliency denotes the intensity change rate (image gradient) along the perpendicular direction of local boundary. It affects the difficulty of key points localization and therefore their accuracy.

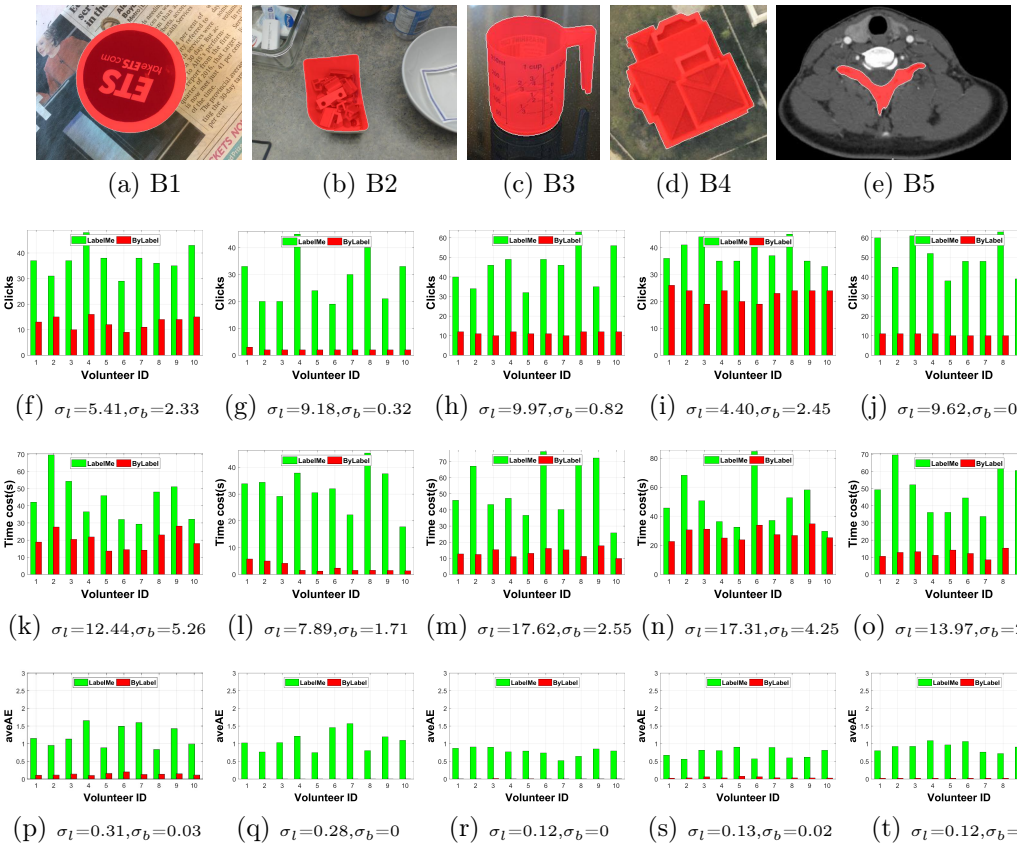
To study the influences of the two factors on image annotation, we first

conduct our user tests on five synthetic images of increasing complexity. These five images are square, octagon, dodecagon, circle and a shape of butterfly respectively (see the first row of Fig. 3.6). They are generated by blurring corresponding binary shape maps using a Gaussian filter (kernel size = 5, $\sigma = 1$). The Gaussian filter blurs the binary map to simulate the appearance of real image boundaries. The binary shape maps are retained as ground truth of synthetic images.

In LabelMe, to annotate regular polygons whose boundaries are straight line segments as shown in Fig. 3.6(a), 3.6(b) and 3.6(c), users just have to click through the corners sequentially. But as the number of corners goes up, more *clicks* and *time costs* are required (see Fig. 3.6(i) and 3.6(n)). The extreme case of the regular polygon is a circle, as shown in Fig. 3.6(d). Theoretically, there are infinitely many corners along its boundaries. The users would need to click many times to obtain an accurate annotation. Compared with LabelMe, ByLabel annotates targets simply by selecting the detected edge fragments, which is more efficient for smooth boundary annotation. As can be seen in Fig. 3.6(f) - 3.6(i), using ByLabel reduces *clicks* greatly and therefore saves the annotation time (see Fig. 3.6(k) - 3.6(n)). Although ByLabel requires the same number of clicks as LabelMe in annotating A1, it costs less time because selecting detected edge fragments is easier than localizing the exact control points. The butterfly shape in Fig. 3.6(e) is comprised of many corners and smooth arcs. It is designed as a comprehensive annotation test. The results show that ByLabel achieves significant improvement in terms of *clicks* and *time cost*, see Fig. 3.6(j), Fig. 3.6(o) and Fig. 3.9.

Overall, as the shape complexity increases, more *clicks* and *time costs* are required to obtain relatively detailed annotations using LabelMe. However, with ByLabel, the number of clicks and the time costs stayed at a low level, as shown in the second and third row of Fig. 3.6.

As shown in Fig. 3.6(p) - 3.6(t), users produce smaller errors (0.5 - 1 pixel) when using ByLabel than using LabelMe (over 1 pixel). Besides, different users achieve almost the same error on each testing image when using ByLabel which suggests that ByLabel is able to reduce the annotation uncertainties. The



(a) B1 (b) B2 (c) B3 (d) B4 (e) B5

(f) σ_l=5.41,σ_b=2.33 (g) σ_l=9.18,σ_b=0.32 (h) σ_l=9.97,σ_b=0.82 (i) σ_l=4.40,σ_b=2.45 (j) σ_l=9.62,σ_b=0.53

(k) σ_l=12.44,σ_b=5.26 (l) σ_l=7.89,σ_b=1.71 (m) σ_l=17.62,σ_b=2.55 (n) σ_l=17.31,σ_b=4.25 (o) σ_l=13.97,σ_b=2.80

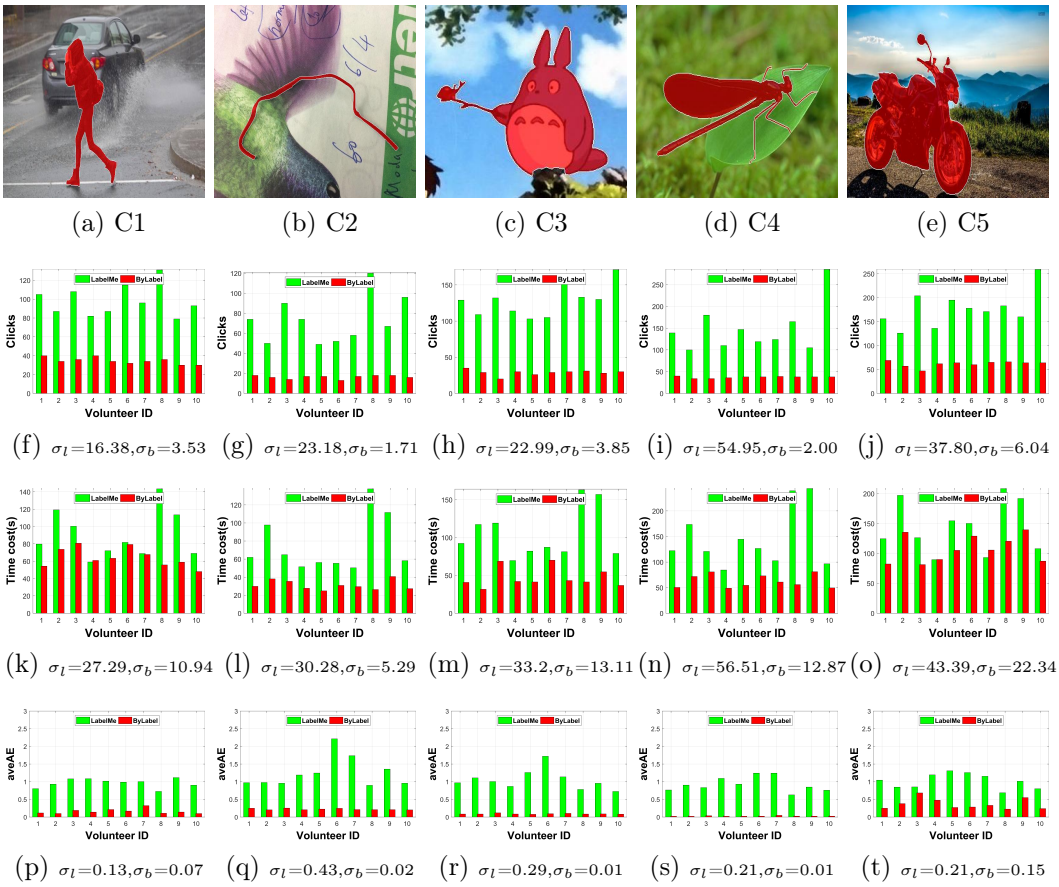
(p) σ_l=0.31,σ_b=0.03 (q) σ_l=0.28,σ_b=0 (r) σ_l=0.12,σ_b=0 (s) σ_l=0.13,σ_b=0.02 (t) σ_l=0.12,σ_b=0

Figure 3.7. The results of user tests on the first group of real images. The top row shows the test images. The highlighted red regions are to-be-annotated targets. The second to fourth rows show the *clicks*, *time costs* and *average Alignment Error (aveAE)* respectively. σ_l and σ_b are standard deviations of LabelMe and ByLabel.

quantitative evaluation of the uncertainties is the standard deviation σ.

2) Tests on real images

The 10 real images including nature images, satellite image, medical image, and manga are selected for user tests. Their ground truth masks are obtained by averaging the labeling results of three experienced annotators. In the ground truth masks labeling, different image processing algorithms, such as adaptive image filtering, image enhancement, gradient computation and so on, are allowed to obtain auxiliary (qualitative and quantitative) information for achieving relatively more accurate results. These images are divided into two groups according to their targets' shape complexity. The targets in the first group are relatively simple, as shown in Fig. 3.7(a) - 3.7(e). Those in the



(a) C1 (b) C2 (c) C3 (d) C4 (e) C5

(f) σ_l=16.38,σ_b=3.53 (g) σ_l=23.18,σ_b=1.71 (h) σ_l=22.99,σ_b=3.85 (i) σ_l=54.95,σ_b=2.00 (j) σ_l=37.80,σ_b=6.04

(k) σ_l=27.29,σ_b=10.94 (l) σ_l=30.28,σ_b=5.29 (m) σ_l=33.2,σ_b=13.11 (n) σ_l=56.51,σ_b=12.87 (o) σ_l=43.39,σ_b=22.34

(p) σ_l=0.13,σ_b=0.07 (q) σ_l=0.43,σ_b=0.02 (r) σ_l=0.29,σ_b=0.01 (s) σ_l=0.21,σ_b=0.01 (t) σ_l=0.21,σ_b=0.15

Figure 3.8. The results of user tests on the second group of real images.

second group are more complicated and challenging, see Fig. 3.8(a) - 3.8(e).

B1, B2, B4 and B5 are circular box, rounded rectangular container, measure cup and MRI image of human neck respectively. Their boundaries are all smooth curves. Fig. 3.7(f), 3.7(g), 3.7(h), 3.7(j) and Fig. 3.7(k), 3.7(l), 3.7(m), 3.7(o) show that ByLabel reduces both *clicks* and *time costs* on this kind of targets. B3 (see Fig. 3.7(d)) is an aerial image with a building roof whose boundary is comprised of multiple straight edges and sharp corners. Our algorithm split its boundary into many edge fragments at its sharp corners. As a result, the annotation *clicks* and *time costs* of ByLabel and LabelMe are similar. But using ByLabel achieves almost zero errors while the errors when using LabelMe are all close to 1 pixel, as shown in Fig. 3.7(p) - 3.7(t).

Targets in Fig. 3.8(a) - 3.8(e) are pedestrian, cable, manga totoro, insect and motorcycle respectively. Compared with the targets in the first group, this

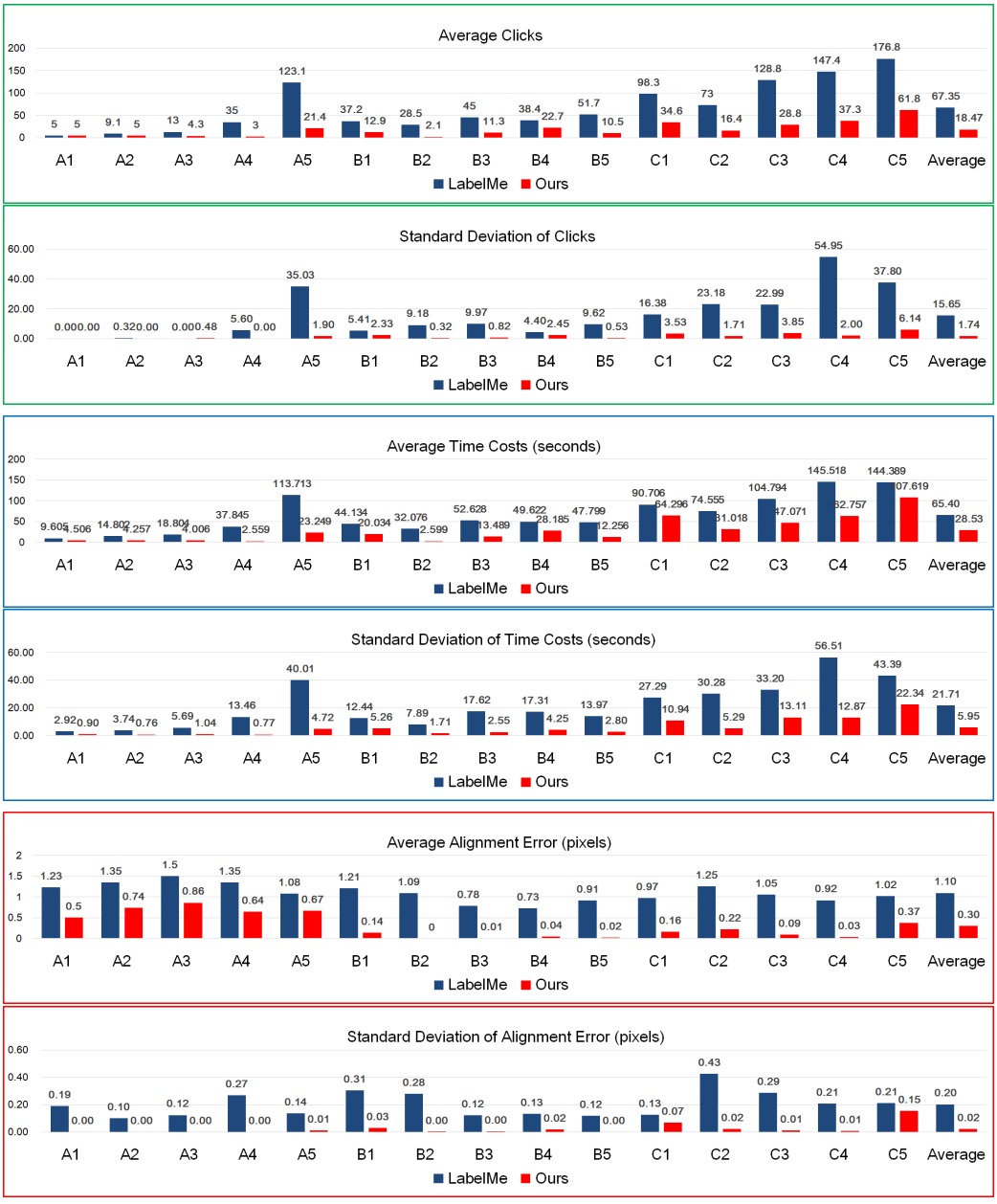


Figure 3.9. Averages and standard deviations of different measures.

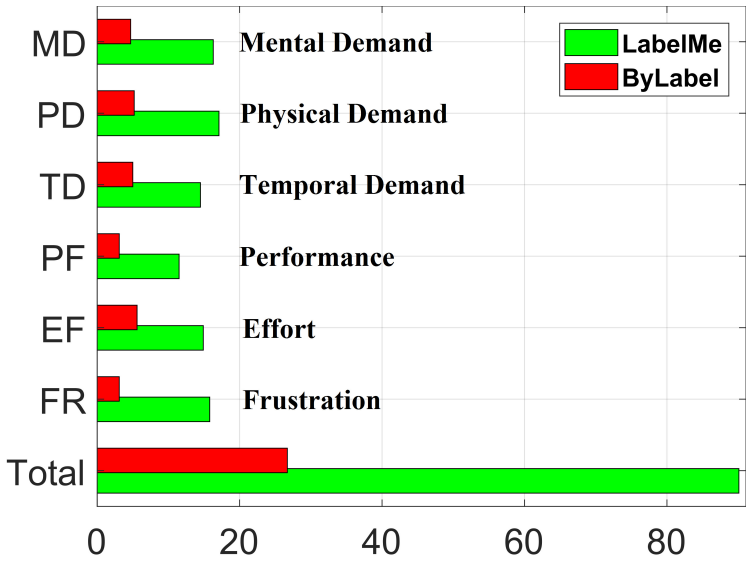


Figure 3.10. The average value of NASA Task Load Index (TLX) produced by those 10 volunteers.

five images have finer structures, which are challenging to annotate. As shown in Fig. 3.8(f) - Fig. 3.8(j), more than 100 clicks are required to annotate each of these targets by LabelMe. ByLabel reduces *clicks* to close to or fewer than 50. Fig. 3.8(k) - 3.8(o) illustrate the advantage of ByLabel in terms of *time costs*.

As can be seen in Fig. 3.8(k) and Fig. 3.8(o), some volunteers spend the same or even more time in annotating object C1 and C5 when using ByLabel compared to LabelMe. The reason is that some part of the boundary in C1 and C5 are missing. Volunteers have to draw the missing parts using "drawing" mode of ByLabel. A lot of the time are spent in figuring out how to switch between "selecting" and "drawing" mode due to their limited experience with the tool. This can be reduced for experienced users.

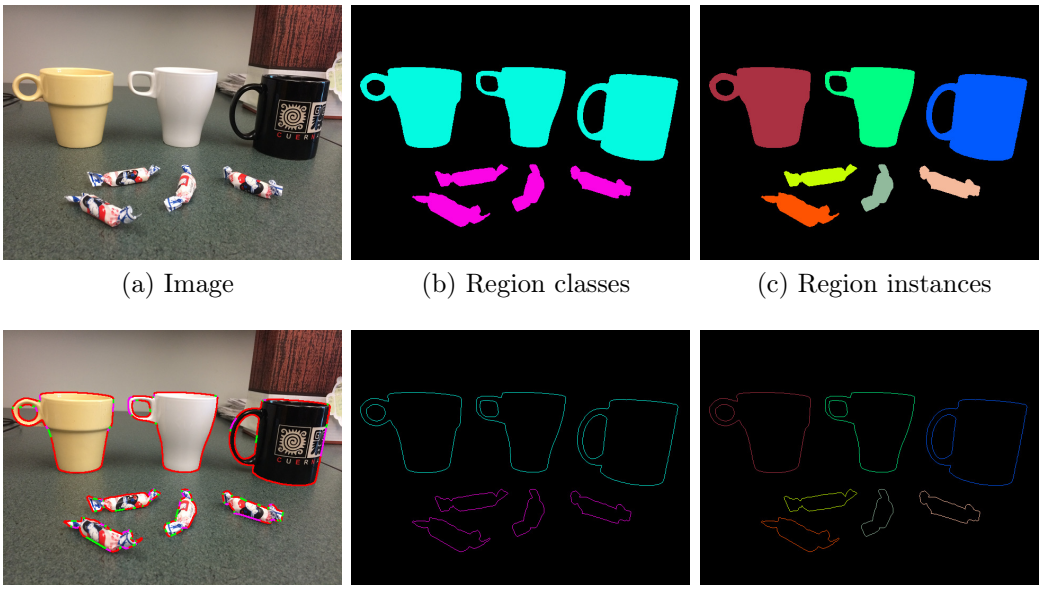
3) Overall Evaluation

We summarize the averages and standard deviations of *clicks*, *time costs* and *errors* of 10 volunteers on each testing image in Fig. 3.9. Our ByLabel reduces 72.58% of *clicks* (from 67.35 to 18.47), meanwhile saves 56.38% *time costs* (from 65.04 s to 28.53 s). The bar graph shows that our ByLabel achieves an overall average Alignment Error (ave_AE) of 0.30. Compared with that

Figure 3.11. Annotation of objects with multiple boundaries.

(1.10 pixels) of using LabelMe, the annotation error is decreased by 72.73%. The standard deviation of each measure indicates that our ByLabel is more reliable than LabelMe.

Fig. 3.10 shows the NASA Task Load Index results generated by these 10 volunteers. Lower scores denote more friendly *user experience*. As shown in Fig. 3.10, ByLabel achieves lower scores than LabelMe in all six aspects.



(a) Image (b) Region classes (c) Region instances

(d) Annotated image edge (e) Edge classes (f) Edge instances

Figure 3.12. Classes and instances.

3.6.2 More Annotation Examples

In section 3.1, the testing targets are all defined by single boundaries. To further illustrate the capability of ByLabel, Fig. 3.11 shows annotations of some commonly used objects which are defined by multiple boundaries.

Additionally, ByLabel is able to output both class based and instance based annotations. Fig. 3.12 depicts a scene with three cups and four candies. Fig. 3.12(b) and Fig. 3.12(e) show the class based annotations. Objects belonging to the same class are encoded in the same color. Fig. 3.12(c) and Fig. 3.12(f) show the instance based annotations. Each object instance has a unique color. The color codes and their corresponding classes and instances are output into the additional text files.

3.7 Summary

In this chapter, we develop a novel semi-automatic boundary based image annotation tool, ByLabel. Instead of annotating images directly, ByLabel introduces edge detection and splitting algorithms to assist annotation, which greatly improves the annotation efficiency and accuracy. The results of user tests show that ByLabel outperforms the state-of-the-art annotation tool LabelMe in terms of *time costs*, *accuracy* and *user experience*. Additionally, ByLabel can also be used to annotate video streams frame by frame.

Chapter 4

Unsupervised Salient Closed Boundary Extraction by Perceptual Grouping

4.1 Overview

In the previous chapter, we described our novel interactive method, ByLabel, for semi-automatic image annotation. The interactive method is able to produce very accurate and reliable results. However, its accuracy and robustness still rely on human interventions so that it hard to be used in some applications, which require fully automatic processing or real-time speed. It motivates us to explore fully automatic and unsupervised methods.

4.2 Problem Formulation

Salient Closed Boundary (SCB) extraction refers to the process of extracting the most salient object or target represented by a closed boundary from a given input image or video frame. The problem of salient closed boundary extraction here is formulated as a fragments based perceptual grouping problem. The fragments based perceptual grouping refers to the process of identifying and connecting a subset of extracted edge fragments or line segments from the given image to formulate a complete closed boundary. The idea of perceptual grouping is originally derived from psychology based on the observation that humans naturally perceive objects as organized patterns and objects [264]. The

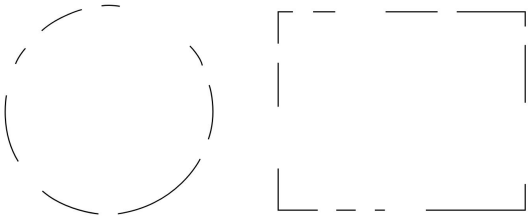


Figure 4.1. Principle of closure [264].

perceptual grouping methods are usually based on a set of principles, Gestalt Laws [115]. The Gestalt laws mainly consist of six aspects including proximity, closure, continuity, similarity, common fate and good form [264]. The Gestalt laws correspond to enforcing specified properties of a boundary. The Gestalt laws enforce the to-be-grouped boundary to have specific properties [253]. For example, proximity enforces to group the neighboring fragments with small gap lengths. Closure requires the target boundary to be a cycle. Continuity corresponds to the regulation on the smoothness of the target boundary. Fig. 4.1 shows a set of disconnected fragments. Human vision systems are prone to perceive them as one circle on the left and one rectangle on the right due to the principles of proximity, closure and continuity in psychology.

4.3 Workflow of the Perceptual Grouping

To simulate the perceptual grouping process of the human psychology system, we adopted a general workflow mainly based on the principles of proximity and closure in the Gestalt laws. The general workflow can be divided into the following five steps: (1) fragments extraction; (2) gap filling; (3) graph construction; (4) objective function definition; (5) optimization.

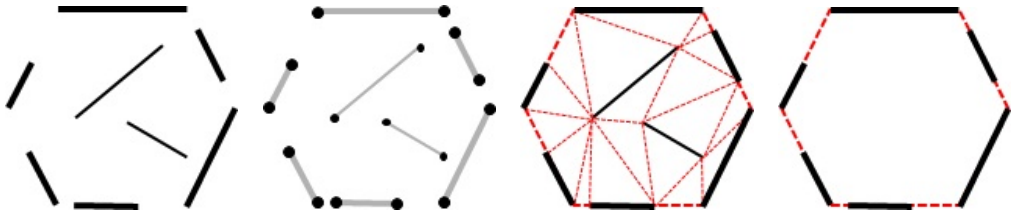
(1) **Fragments extraction** is the basic step for obtaining the to-be-grouped elements. The fragments here can be either edges or line segments. It is worth noting that edges directly detected by commonly used edge detectors like canny [32], edge drawing [238] have to be split into fragments with proper length that excludes cases where background and foreground pixels belong to the same edge. Hence, the output of this step is a set of disconnected fragments.

(2) **Gap filling** is to connect the extracted fragments. The naive way of building the connection is to generate line segments between endpoints of different fragments based on a distance threshold. However, this connection building method is sensitive to scale changes and noisy fragments. Hence, Delaunay Triangulation [200] is used here to generate the connection segments.

(3) **Graph construction** aims at reducing the salient closed boundary perceptual grouping problem into the problem of searching for a cycle from an undirected graph. The graph vertices represent the endpoints of extracted fragments and the graph edges correspond to the extracted fragments and the generated connecting line segments. Each edge is assigned a weight to describe its properties.

(4) **Objective function definition** is one of the most important steps in solving many vision related problems. The objective function should provide an accurate mathematical description of to-be-solved problem and it determines the theoretical upper bound of the final performance. In this work, the objective functions are all defined based on the grouping cues encoded in the undirected graphs.

(5) **Optimization** is usually closely related to the definition of the objective function. Some of the objective functions, especially those defined on graphs, are often NP-hard problems and are difficult to be solved in polynomial time. Hence, many optimization algorithms are specifically designed for certain objective functions. Meanwhile the design of objective functions has to follow specific rules to guarantee the solvability. In this work, a novel simple graph-based optimization algorithm, "Bi-Directional Shortest Path (BDSP)", is developed to solve the salient closed boundary extraction problem. It is flexible and is able to solve different types of objective functions defined on the undirected graph, which enables the salient closed boundary extraction workflow to be adapted to other different applications.



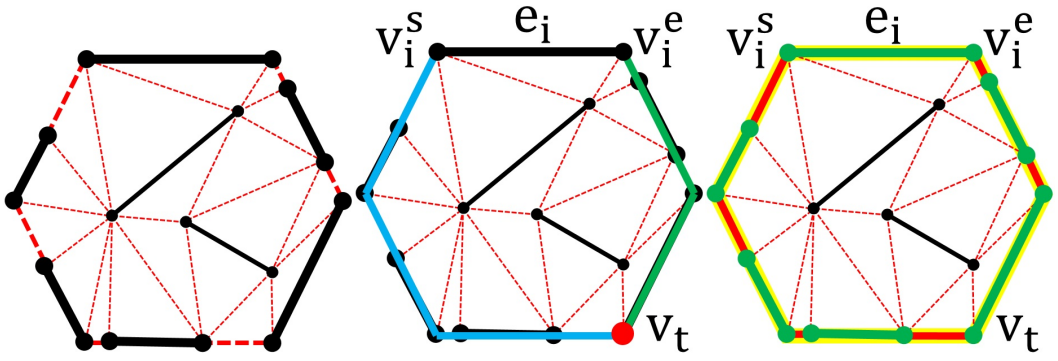
(a) Detected fragments (b) Fragments endpoints (c) Connected fragments (d) Grouped hexagon

Figure 4.2. Perceptual grouping of a hexagon: (a) shows detected fragments of a target hexagon; thick and thin fragments denote hexagon boundary fragments and noises; (b) shows the endpoints of the detected fragments; (c) dashed lines are connection segments generated by Delaunay Triangulation [200] conducted on fragments endpoints; (d) illustrates the grouped hexagon.

For example, Fig. 4.2(a)-(d) shows the typical process of grouping a salient hexagon target. Given an image with a salient target, the edge fragments or line segments will be detected first (see Fig. 4.2(a)). These detected fragments are usually disconnected, non-crossing and noisy (thin fragments in Fig. 4.2(a)). The goal of perceptual grouping is to extract the most salient intact closed hexagon as shown in Fig. 4.2(d). To find closed boundaries, the potential connections have to be built by connecting their endpoints (see Fig. 4.2(b)). Fig. 4.2(c) shows the gap filling results produced by conducting the Delaunay Triangulation on the endpoints of extracted fragments. Then, an undirected graph G(V,E) is built by mapping the fragments (both detected and generated) and their endpoints to graph edges E and vertices V respectively. Each graph edge is assigned a weight value to describe its property. Now the problem is reduced to searching for the optimal cycle from the undirected graph. An objective function and a graph-based optimization algorithm are developed to search for the optimal cycle from the undirected graph and produce the final grouping result (see Fig. 4.2(d)).

4.4 The Proposed Optimization Algorithm

As described above, the problem of salient closed boundary extraction is formulated as a problem of searching for the optimal cycle from an undirected graph G(V,E) (see Fig. 4.3(a)). To solve this problem, both objective func-



(a) Graph (b) Cycle (c) Objective

Figure 4.3. Graph based optimization: (a) an undirected graph where solid black lines and dashed red lines corresponds to the extracted fragments and generated gap filling fragments respectively; (b) a graph cycle candidate $C_{v_t}^{e_i}$ comprised of edge e_i, the shortest path $P_{v_i^s}^{v_t}$ from vertex v_i^s to v_t and the shortest path $P_{v_i^e}^{v_t}$ from v_i^e to v_t; (c) the cycle $C_{v_t}^{e_i}$ consists of the graph edges corresponding to extracted fragments (solid green lines) and those corresponding to generated gap filling fragments (solid red lines).

tion and the optimization algorithm play important roles. Since many graph related optimization problems are NP-hard, the objective functions defined on the undirected graph usually have to follow certain formats, such as the fraction style objective functions in [253] and [233], to guarantee the existence of polynomial-time solutions. This constraint reduces the flexibility of the objective function designing. Therefore, there is a need for a simple and flexible optimization algorithm which is able to optimize objective functions with different formats. Generally, a promising optimization algorithm usually has following characteristics: (1) guarantee the global (or approximately global) optimal solution; (2) run in polynomial time; (3) flexible to be generalized to solve objective functions with different formats. In this section, a novel graph-based optimization algorithm called "Bi-Directional Shortest Path" (BDSP) with above characteristics is presented.

To illustrate the newly proposed graph-based optimization algorithm BDSP, a simple objective function defined upon the closure and proximity principles of Gestalt laws will be demonstrated as an example. As mentioned above, the principle of closure enforces the boundary to be close which is explicitly

reflected by searching for a cycle from the undirected graph. The principle of proximity is encoded as the fraction of the total gap length $\oint_{\mathcal{B}} w(s)ds$ along the to be extracted closed boundary $\mathcal{B}$ and the perimeter $\oint_{\Gamma} ds$ of this boundary:

$$\Gamma(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds}{\oint_{\mathcal{B}} ds}. \tag{4.1}$$

As shown in Fig. 4.3(c), the total gap length $\oint_{\mathcal{B}} w(e)ds$ is the summation of the length of the solid red lines and the perimeter $\oint_{\mathcal{B}} ds$ is the total summation of the length of the solid red and green lines. The length and the property of each graph edge e_i are encoded into a weight value:

$$w(e_i) = \begin{cases} 0 & e_i \text{ corresponds to a } \textit{detected} \text{ fragment } s_i \\ |P_1^i P_2^i| & e_i \text{ corresponds to a } \textit{generated} \text{ fragment } s_i \end{cases} \tag{4.2}$$

where $|P_1^i P_2^i|$ is the length of the corresponding fragment s_i of the graph edge e_i.

The difficulty of optimizing (4.1) on the undirected graph G(V,E) is that it is hard to compute the total gap length and the perimeter before determining the cycle. On the other hand, without computing the total gap length and the perimeter it is impossible to determine the optimal cycle. Therefore, it is a chicken and egg problem. Exhaustive search is able to guarantee the global optima. But it is time-consuming and unfavorable when the graph is relatively large. Wang *et al.* [253] transfer the optimization problem to a Minimum Ratio Alternate (MRA) cycle problem and solve that by a polynomial-time algorithm, Minimum-Weight Perfect Matching (MWPM). However, to be solved by MWPM the objective function has to follow or can be transferred into the fraction-style format like (4.1), which is unavailable in some other applications.

Given an undirected graph G(V,E) with n edges corresponding to *detected* fragments and $2n$ vertices, our BDSP algorithm has two steps: (1) *Cycle Candidates* generation, (2) optimal cycle retrieval. First, several possible cycle candidates will be extracted in the cycle candidates generation step. As shown in Fig. 4.3(b), BDSP searches two shortest paths from two vertices (v_i^s, v_i^e) related to edge e_i to a commonly third vertex v_t by Dijkstra [59]. To avoid generating of two overlapping shortest paths, the weight of edge e_i is

temporarily set to an infinite value when conducting the Dijkstra. The edge e_i, the shortest path $P_{v_i^s}^{v_t}$ from vertex v_i^s to v_t and the shortest path $P_{v_i^e}^{v_t}$ from v_i^e to v_t construct a graph cycle candidate $C_{v_t}^{e_i}$. For each edge e_j, BDSP traverses all the other $2 \times (n-1)$ vertices to achieve $2 \times (n-1)$ cycle candidates. For the whole graph, there are $2 \times n \times (n-1)$ cycle candidates in total. Second, the objective function value of each cycle candidates are computed with (4.1). The cycle candidate with the minimal value is then selected as the final result of the salient closed boundary. The hypothesis of BDSP is that a graph cycle with smaller total weight is more likely to be the optimal boundary. Although the final results are not theoretically guaranteed to be global optima, they are usually perfect in practice. Since BDSP collects the cycle candidates first, those determined graph cycles enable the computation of objective functions with different formats. Furthermore, BDSP can be adapted to achieve a trade-off between accuracy and speed according to different applications. For example, instead of traversing all of the n edges (corresponds to *detected* fragments) and $2 \times (n-1)$ vertices in the graph, bi-directional shortest paths can be conducted on fewer sampled edges, *e.g.* $n/2$, and vertices, *e.g.* $(n-1)$.

4.5 Application I: Building Outline Extraction

4.5.1 Overview

This section presents a novel approach for extracting accurate outlines of individual buildings from Very-High-Resolution (VHR, 0.1-0.4 m) optical images. Building outlines are defined as polygons here. Our approach operates on a set of straight line segments that are detected by a line detector. It groups a subset of detected line segments and connects them to form a closed polygon. Particularly, a new grouping cost is defined firstly. Second, a weighted undirected graph *G(V,E)* is constructed based on endpoints of those extracted line segments. The building outline extraction is then formulated as a problem of searching for a graph cycle with the minimal grouping cost. To solve the graph cycle searching problem, the "Bi-Directional Shortest Path (BDSP)" method is utilized. Our method is validated on a newly created dataset which contains 123 images of various building roofs with different shapes, sizes and intensities. The experimental results with average Intersection-over-Union of 90.56% and average alignment error of 6.56 pixels demonstrate that our approach is robust to different shapes of building roofs and outperforms state-of-the-art method.

4.5.2 Motivation

Accurate building outline extraction is important to urban planning, cadastral surveying and other related applications. Building outlines extraction from very high resolution (VHR) images has become a popular yet challenging topic in the fields of photogrammetry, remote sensing, geographic information system (GIS) and computer vision.

There already exist many methods for building outlines extraction from aerial and satellite images. Image segmentation based methods [116], [190], [40], [142], [308] is a popular class of methods. Cote and Saeedi [52] combine distinctive corners detection with level-set method to fit the best possible boundaries of building rooftop. Song and Shan [230] adopt active contour models and intensity based cluster to extract building boundaries from satel-

lite images. Yang and Wang [280] extract building contours using shape priors constrained level set method. These methods are sensitive to initialization, local minimum and noise. In addition, perceptual grouping [249] [283] and deep learning based approaches [288] have also been proposed. However, the approach proposed in [249] is not able to extract non-rectangular building outlines. Because its grouping rules are designed according to the basic attributes of rectangles: four sides with right angles. Learning based building extraction methods [288] require large number of training data and their results are usually region masks with relatively coarse contours. Little attention has been paid to obtaining accurate and detailed outlines of individual buildings.

This work focuses on extracting accurate outlines of individual buildings with different shapes. The accurate outline extraction is formulated as a salient object detection problem. There are mainly two classes of salient object detection methods: intensity based and edge based.

The intensity based methods detect objects using the saliency measure defined on the difference or contrast between foreground and background pixels. Yang *et al*. [278] utilize super-pixels of a given image as nodes to construct a close-loop graph. The saliency maps are obtained by ranking these nodes according to their similarities to background and foreground queries based on affinity matrices. Zhu *et al*. [310] propose a background measure to characterize the spatial layout of image regions with respect to image boundaries. They integrate multiple low level cues and the background measures into their optimization problem. Srivatsa and Babu [232] estimate the foreground regions using objectness proposals and then other pixels/regions are weighted by their proposed saliency measure. They integrate these weights into an optimization framework to obtain the final saliency map. Zhang *et al*. [296] develop a salient object detection method by solving an approximate Minimum Barrier Distance (MBD) Transform, which achieves 100X speedup over the exact MBD algorithm. The outputs of these methods are usually saliency maps with coarse or uneven boundaries of the target objects.

The edge based methods usually define saliency measure based on the properties of to-be-extracted object boundaries, such as curvature, gap, length,

enclosed region area, etc. Gestalt laws [115] and perceptual grouping are their theoretical basis.

Kokkinos [125] proposes a fractional-linear programming approach to find the most salient boundary. Lu *et al.* [165] create a contour saliency measure subject to completeness and tightness constraints, and optimize it using dynamic programming in polar coordinate system. However, the transformation between Cartesian coordinate system and polar coordinate system decreases the accuracy of boundaries. Wang *et al.* [249] extract rectangular building outlines by grouping straight line segments according to their distances and angles, but only simple rectangular buildings can be extracted and several parameters have to be tuned carefully. Wang *et al.* [233], [253] develop a graph-based method for salient closed boundaries detection. This method selects and connects a subset of edge fragments sequentially to form a closed boundary with the saliency maximum. Although they can handle more irregular shapes than [249], the cost function in their optimization is not adapted for building outline extraction.

As buildings are artifacts with rich straight line features, our method is based on grouping detected straight line segments, similar to [233], [253]. The main contributions of this work are: (1) a novel grouping cost for individual building outline extraction; (2) a new dataset with manually labeled ground truth; (3) state-of-art performance on the new dataset both in terms of region and boundary measures.

4.5.3 Proposed Method

Given a VHR image, our goal is to group a subset of detected line segments and form a polygon which describes the accurate outline of a target building. First, a novel grouping cost, which is composed of the completeness (or closure) and smoothness (or continuation) [115], is defined. Then the straight line segments are detected from the given image to construct a weighted undirected graph $G(V, E)$. Finally, the grouping process is formulated as searching for the optimal cycle in the weighted undirected graph.

(1) Derivation of Grouping Cost

1) Completeness

An intuitive definition of completeness requires the ratio of non-edge pixel number over the total pixel number along a closed curve to be sufficiently small (the optimum is zero) [165]. Given a closed curve $\mathcal{B}$, the completeness cost $C(\mathcal{B})$ is defined:

$$C(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds}{\oint_{\mathcal{B}} ds} \tag{4.3}$$

where $w(e)$ is defined in (4.2).

2) Smoothness

The smoothness cost $S(\mathcal{B})$ of a closed curve $\mathcal{B}$ is given by the Elastic prior [125], [223]:

$$S(\mathcal{B}) = \frac{\oint_{\mathcal{B}} |\kappa(s)|ds}{\oint_{\mathcal{B}} ds} \tag{4.4}$$

where $\kappa(s)$ is the curvature of edge pixel. The numerator denotes the total absolute curvature and measures how far the curve is from a convex curve [29].

Our method operates on a set of straight line segments that are extracted by a line detector. As the curvature at straight line intersections is not well defined, we describe the relative magnitude of the total absolute curvature term in (4.4) by the ratio of contour perimeter and area:

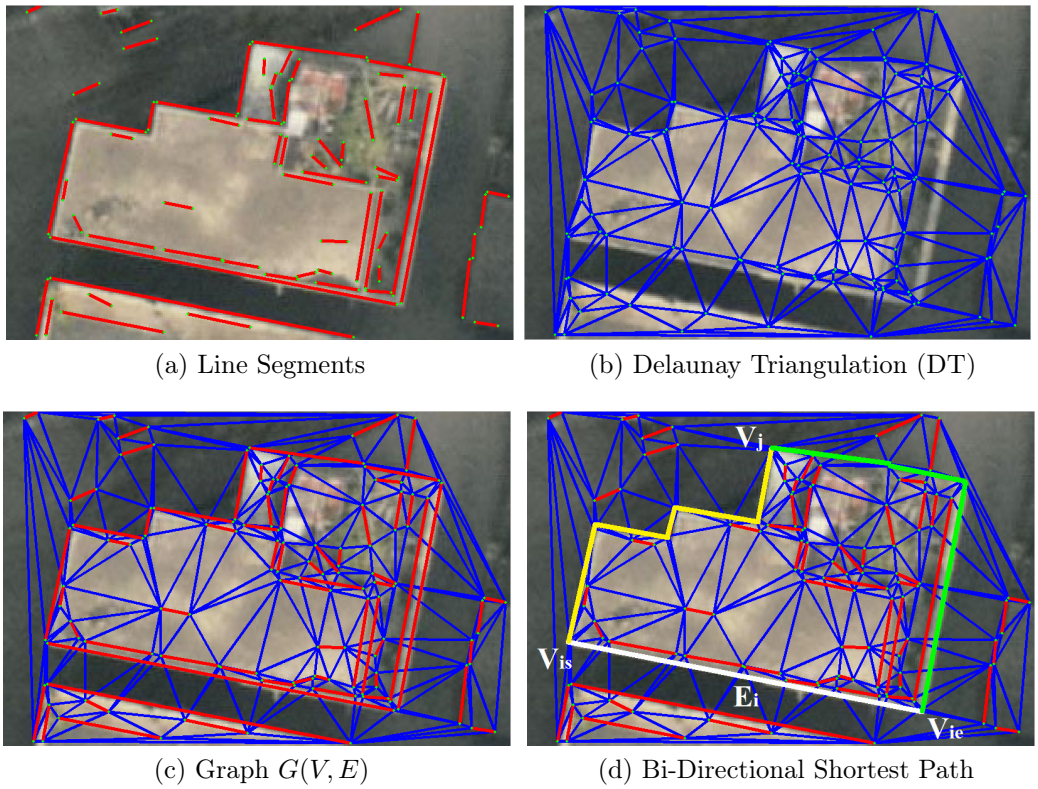
$$\oint_{\mathcal{B}} |\kappa(s)|ds \propto \frac{\oint_{\mathcal{B}} ds}{\iint_{\mathcal{R}} dA} \tag{4.5}$$

where $\mathcal{R}$ is the region enclosed by contour $\mathcal{B}$ and $\iint_{\mathcal{R}} dA$ denotes the area of region $\mathcal{R}$. The smoothness term can be further simplified using (4.4) and (4.5) to:

$$S(\mathcal{B}) \propto \frac{1}{\iint_{\mathcal{R}} dA} \tag{4.6}$$

In other words, this term is reciprocal to the area of region $\mathcal{R}$ enclosed by $\mathcal{B}$. Literally, it indicates the preference for extracting objects with larger areas. In fact, it implicitly guarantees the smaller grouping costs of contours with smoother and less concave shapes.

3) Grouping Cost



(a) Line Segments (b) Delaunay Triangulation (DT)

(c) Graph $G(V, E)$ (d) Bi-Directional Shortest Path

Figure 4.4. Graph construction and optimization.

Grouping cost is typically formulated as a weighted summation of completeness and smoothness [165]:

$$\Gamma(\mathcal{B}) = C(\mathcal{B}) + \lambda S(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds}{\oint_{\mathcal{B}} ds} + \lambda \cdot \frac{1}{\iint_{\mathcal{R}} dA} \tag{4.7}$$

where λ is a weight that balances the completeness and smoothness. However, (4.3) and (4.6) are two incomparable measures with different units. This makes the tuning of λ a hard process for obtaining good results. In fact, the choice of optimal λ is shape-dependent and differs for different images. This will be further elaborated in the experimental results.

This motivate us to propose a new Grouping Cost $\Gamma(\mathcal{B})$ as the multiplication of these two terms:

$$\Gamma(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds}{\oint_{\mathcal{B}} ds} \cdot \frac{1}{\iint_{\mathcal{R}} dA}. \tag{4.8}$$

In this work, the building outlines are depicted by polygons. The Grouping Cost $\Gamma(\mathcal{B})$ of a polygon, comprised of n detected line segments and k gaps, is

defined as:

$$\Gamma(\mathcal{B}) = \frac{\sum_j^k w(e_j)}{(\sum_i^n l_i + \sum_j^k w(e_j))} \cdot \frac{1}{A} \tag{4.9}$$

where e_j denotes the length of a gap between two sequentially connected line segments and $\sum_j^k w(e_j)$ is the total gap length along polygon, l_i indicates the length of a line segment and $(\sum_i^n l_i + \sum_j^k w(e_j))$ is the perimeter of the polygon and A is the area of the polygon. The polygon with the smallest $\Gamma(\mathcal{B})$ is taken as the final optimal outline.

(2) Graph Construction

Given a VHR image, its straight line segments are first detected by EDLines [3]. Fig. 4.4(a) shows the detected line segments of a VHR image. Each line segment is represented by a pair of endpoints. The line segments are fit from the one pixel-width edges extracted by Edge Drawing (ED) [238], in which edge pixels are searched based on the magnitudes and directions of image gradients. As detected line segments are not connected, Delaunay Triangulation (DT) [200] is utilized to fill the gaps between them (see Fig. 4.4(b)). Then detected line segments are superimposed on the results of Delaunay Triangulation to construct an undirected graph $G(V, E)$ (see Fig. 4.4(c)). Each node in the graph corresponds to an endpoint. A graph edge correspond to either a generated edge (blue line) or a detected line segment (red lines). Note that there are two types of graph edges and their weights are set differently similar to [233]. For a generated edge, the weight is set to its geometric length which is in fact the gap length. The weight of a line segment is set to zero. This means that line segments are more likely to be a part of building outlines.

(3) Optimal Outline Search

A building outline corresponds to a special cycle, which has the minimal grouping cost (4.9), in the weighted undirected graph $G(V, E)$. Minimum Ratio Weight Cycles (MRWC) [112], [253] are most commonly used algorithms for the grouping cost optimization. However, the denominator of our grouping cost (4.9) is multiplication of two terms which cannot be solved with MRWC methods. This section describes our algorithm for finding the optimal cycle based on the grouping cost (4.9). As mentioned above that our BDSP al-

gorithm has two steps: (1) *Cycle Candidates* generation; (2) optimal outline retrieval.

In the first step, a set of cycle candidates are generated. As shown in Fig. 4.4(d), a cycle is considered as a candidate if it consists of the following three components: 1) a zero-weight edge E_i (white edge); 2) a shortest path from the start node V_{is} of the edge E_i to a third node V_j (yellow path); 3) a shortest path from the end node V_{ie} of the edge E_i to the same third node V_j (green path). For each zero weighted edge, we first set its weight to infinity and then traverse all of the third nodes on the graph to generate cycle candidates by BDSP. This means that for a graph with n zero-weighted edges, there are $n \times 2(n-1)$ cycle candidates.

In the second step, the grouping cost (4.9) for each cycle candidate is computed. The cycle candidate with the minimal grouping cost is taken as the final optimal building outline.

4.5.4 Experiments

(1) Dataset and Error Metrics

To assess the performance of our approach, a new dataset for building outline extraction on VHR aerial and satellite images of urban areas is built using [203]. The resolution of these images varies from 0.1-0.4 m. The dataset[1] contains 123 images of buildings with different shapes. The accurate *Ground Truth* (GT) is manually labeled for all the images.

To quantify building outline extraction results, two accuracy measures are used: a region-based and an edge-based measure. The region-based measure is *Intersection-over-Union* (IoU) which is defined as the relative region coincidence [253]:

$$IoU = \frac{|R \cap R_{GT}|}{|R \cup R_{GT}|} \times 100\% \tag{4.10}$$

where R_{GT} and R are the regions enclosed by the ground truth and extracted building outline, respectively. $|R|$ indicates the area of R.

The edge-based measure used is the average *Alignment Error* (E_{aveAl})

[1] `https://webdocs.cs.ualberta.ca/~xuebin/building_extraction.html`

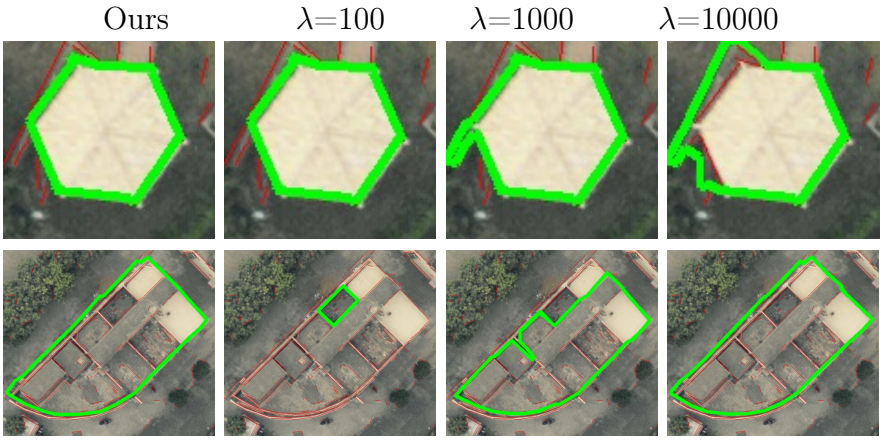


Figure 4.5. Results based on grouping cost (4.7): the size of top row image is 116×108, the size of bottom row image is 501×439.

which is defined as

$$E_{aveAl} = \frac{Dist_{GT} \otimes I_{\mathcal{B}}}{|I_{\mathcal{B}}|} \tag{4.11}$$

where $Dist_{GT}$ is the distance transform map of the ground truth region boundary. $I_{\mathcal{B}}$ is the extracted binarized outline map and $|I_{\mathcal{B}}|$ is the perimeter of the outline $\mathcal{B}$.

(2) Evaluation and Comparison

Both summation-based grouping cost (4.7) and multiplication-based grouping cost (4.9) can be optimized using our "BDSP" algorithm. However, in (4.7), the choice of the weight λ has to be tuned carefully for different images, especially when these buildings have big differences in their shapes and sizes. This is further illustrated in Fig. 4.5, where the results of the algorithm for summation-based cost is shown for different values of λ. It can be seen that it is not possible to achieve a good accuracy on both images using the same λ. Our multiplication-based grouping cost (4.9), on the other hand, resolves this problem. It works well on both of these two buildings without parameters tuning, see the first column in Fig. 4.5.

Our method is compared with other five state-of-the-art methods: (i) a regional information combined ratio contour method (RRC) [233], (ii) a minimum barrier salient object detection method (MB+) [296], (iii) a saliency detection method via graph-based manifold ranking (MR) [278], (iv) an ob-

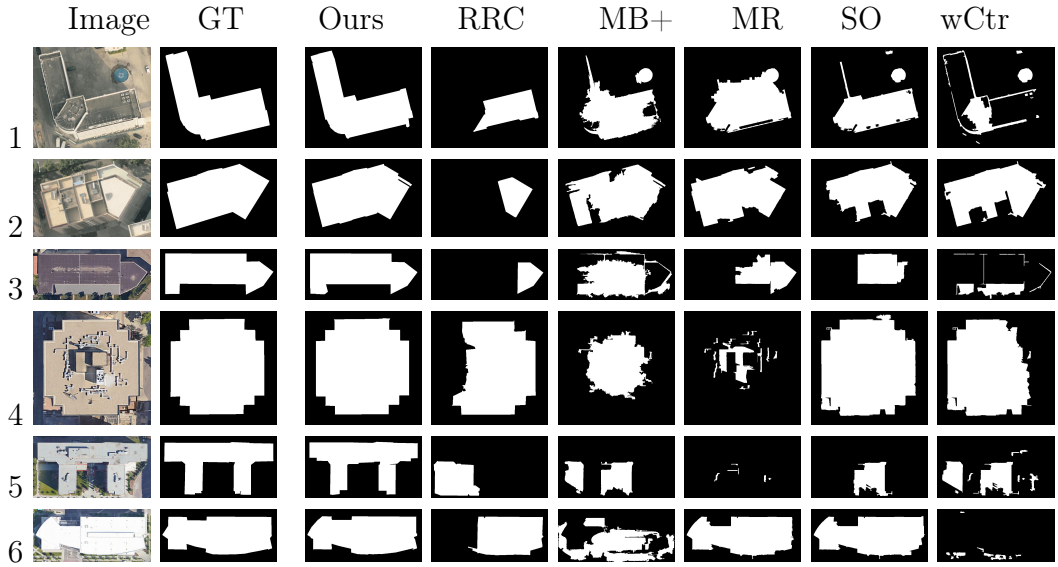
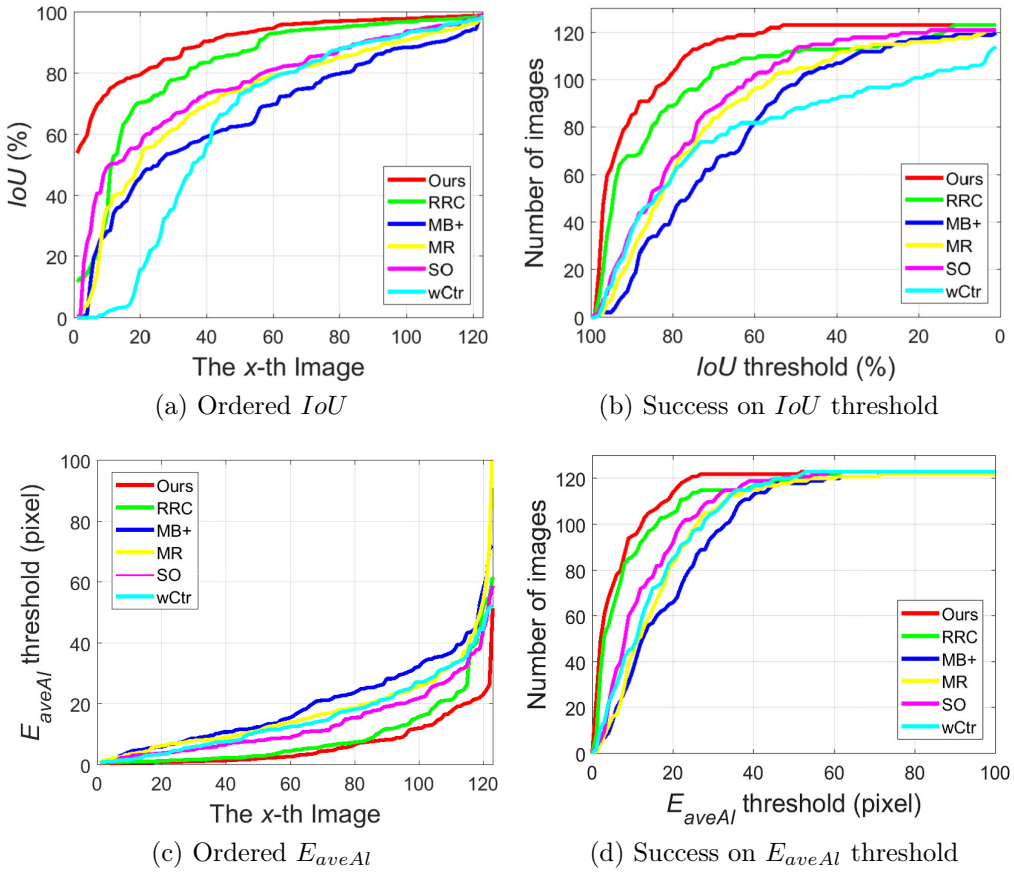


Figure 4.6. Sample results of different methods: the first column is the original image, the second column is the ground truth, the third column is the result of our method, column four to column eight are results produced by RRC, MB+, MR, SO and wCtr.

jectness measure based salient object detection method (SO) [232], and (v) a saliency optimization based method (wCtr) [310]. The RRC method is the state-of-the-art line-based grouping method. The direct outputs of our method and RRC are grouped polygons. To facilitate comparison, they are presented as binarized region maps in Fig. 4.6. Methods (ii)-(v) are regional intensity-based methods. Their original results are saliency maps represented by normalized gray scale images (0-255). In our experiments, these saliency maps are thresholded (the threshold is set to 125 because it provides almost the best overall performance of these methods according to our thresholding tests.) to obtain binarized buildings' regional segmentation. The building outlines are then obtained via edge extraction from those binarized images.

Fig. 4.6 shows sample results of different methods. The key challenge of building outlines extraction by using grouping based method is to resist the impacts of many detected noisy line segments which are close to the target building. RRC fails easily in complex and concave shapes, see row 1-6. It is sensitive to noisy line segments and prone to group shorter boundaries. Our



(a) Ordered IoU (b) Success on IoU threshold

(c) Ordered E_{aveAl} (d) Success on E_{aveAl} threshold

Figure 4.7. Region and edge based evaluations. (a), (b) and (c), (d) are region based and edge based evaluations respectively. (a) and (c) are ordered IoU and E_{aveAl} of the testing images. (b) shows the number of images where the IoU is greater than certain threshold. (d) shows the number of images where the E_{aveAl} is less than certain threshold.

method is more robust than RRC to this kind of noisy line segments. MB+, MR, SO, wCtr are more dependent on homogeneous colors or intensities. They are good at extracting building roofs with unified intensities. Theoretically, they are more robust to complex shapes. However, VHR images contains very detailed structures of building roofs and these structures usually have different colors, which confuse the intensity based methods. Detecting building roofs with heterogeneous intensities is difficult for these intensity based methods, see results in row 2 and 4. They are also prone to take salient background as the building roofs, see row 5. It can be seen that our method is robust to different shapes and intensity variations compared to other methods.

Table 4.1. Overall average IoU and E_{aveAl}.

Method	Ours	RRC	MB+	MR	SO	wCtr
IoU	**90.57**	82.67	66.67	72.23	76.36	64.32
E_{aveAl}	**6.54**	9.52	20.17	17.76	13.87	15.87

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

Figure 4.8. Illustration of failure cases: Red and green outlines are ground truth and our results, respectively. (a) failure caused by complex structures inside the roof outline. (b) failure due to non-building noisy line segments. (c) failure due to missing edges. (d)-(e) failure due to large surroundings area that includes specific textures or structures; (d) is of size 281×277 and (e) is of size 589×611.

Table 4.1 summarizes the average IoU and E_{aveAl} of all the methods over the dataset. Our method achieves 90.56% of the average IoU and 6.56 pixels of E_{aveAl} (more than 30% improvement in E_{aveAl}), which outperform other methods. The overall IoU and E_{aveAl} trends of each method are shown in Fig. 4.7(a) and 4.7(c). To further highlight the robustness of our method, the curves of successfully extracted image numbers with respect to different IoU and E_{aveAl} thresholds are shown in Fig. 4.7. It can be seen that our method outperforms all of the others in terms of both IoU and E_{aveAl}. Although our method outperforms state-of-the-art grouping methods in most cases, there are still extreme cases that result in failure of accurate outline extraction, some of which are shown in Fig. 4.8.

4.5.5 Summary

This chapter addresses the problem of accurate extraction of complex building outlines from VHR aerial and satellite images. A new outline grouping cost is proposed in terms of a ratio that is normalized relative to outline length and

area. Then, a novel and simple framework is introduced for graph construction and outline searching. The results on our newly built dataset demonstrate that our method is robust to buildings with different intensities and shapes. Currently, our method can only extract the most salient building outline from a given image. Hence, the input image has to be roughly cropped around the target building. This prerequisite somehow limits the applications of our method. Future work will focus on extending our method to multiple building outlines extraction from large scale images by integrating oriented object detection methods.

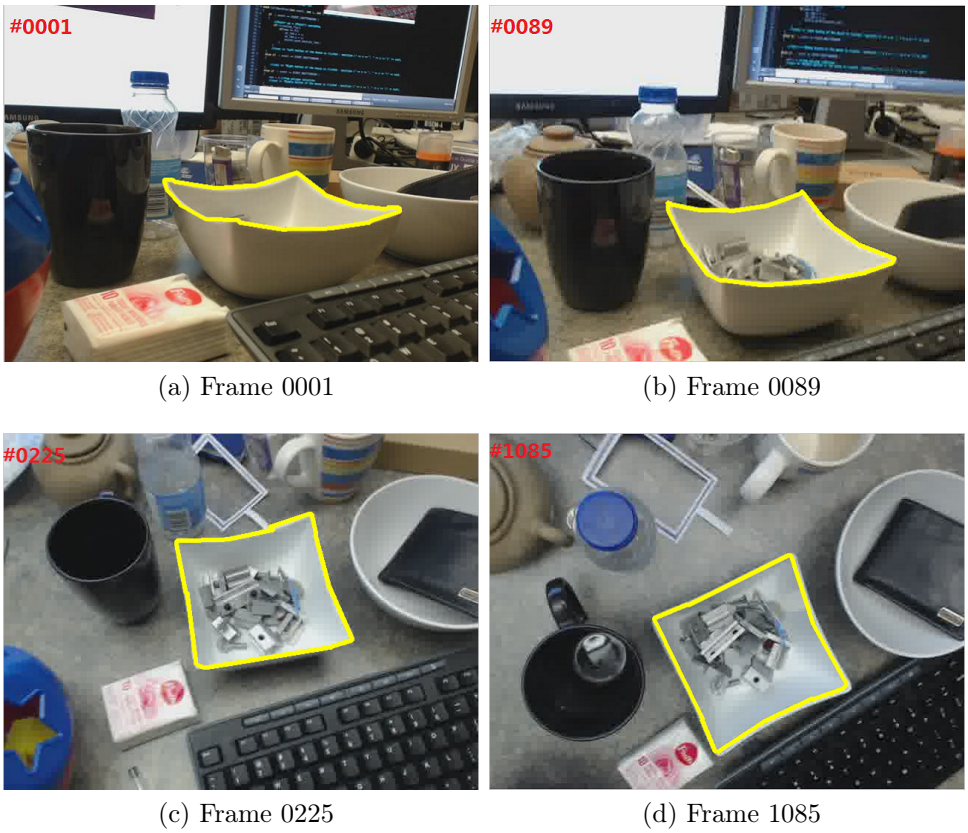
4.6 Application II: Salient Closed Boundary Tracking

Visual tracking is an important yet challenging issue in computer vision and related areas. Many different trackers [48], [82], [166], [309] have been developed to handle different targets and scenarios. Fully automatic processing and relatively fast (e.g. real-time) speed are two important requirements of the most tracking applications. In this section, our two newly developed salient closed boundary tracking methods are presented. The first method is built with line segments perceptual grouping, in which a grouping cost adopted from closed boundary extraction [233] combined with an area variation constraint are utilized and optimized by our BDSP algorithm to track the target closed boundary. To further improve the tracking accuracy and robustness, we replace the line segments by the high quality edge fragments generated by Edge Drawing [238] and our simple splitting technique. In addition, we encode the prior contour information into a distance difference term and integrate it with the area normalized total gap length to formulate a novel prior shape constrained grouping cost. This grouping cost combined with both area and perimeter variation constraints is also optimized by our BDSP algorithm. This edge fragments perceptual grouping method for salient closed boundary tracking provides high quality tracking performance in terms of both accuracy and robustness while keeping the real-time speed.

4.6.1 Salient Closed Boundary Tracking via Line Segments Perceptual Grouping

4.6.1.1 Overview

This section presents a novel real-time method for tracking salient closed boundaries from video frame sequences. This method operates on a set of straight line segments that are produced by line detection. The tracking scheme is coherently integrated into a perceptual grouping framework in which the visual tracking problem is tackled by identifying a subset of these line segments and connecting them sequentially to form a closed boundary with the



(a) Frame 0001 (b) Frame 0089

(c) Frame 0225 (d) Frame 1085

Figure 4.9. Tracking the rim boundary of a bowl with the following characteristics: (1) the rim of the bowl is non-planar, (2) the bowl itself has no salient stable textures, (3) the viewpoint changes dramatically.

largest saliency and a certain similarity to the previous one. Specifically, we define a new tracking criterion which combines a grouping cost and an area similarity constraint. The proposed criterion makes the resulting boundary tracking more robust to local minima. To achieve real-time tracking performance, we use Delaunay Triangulation to build a graph model with the detected line segments and then reduce the tracking problem to finding the optimal cycle in this graph. This is solved by our "Bidirectional Shortest Path" algorithm. The efficiency and robustness of the proposed method are tested on real video sequences as well as during a robot arm pouring experiment.

4.6.1.2 Motivation

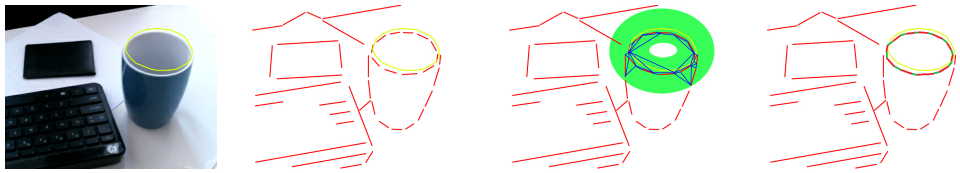
Closed boundaries are common elements in real world scenes. Hence, real-time closed boundary tracking is important in robot vision. As an example, con-

sider Fig. 4.9, where the rim contour of a bowl is tracked. In real-world indoor images and robot applications, conventional trackers can fail, e.g. in the presence of texture-less target regions, non-rigid deformations, non-Lambertian surfaces, changing lighting conditions, cluttered background and drastic changing of supportive regions.

Template-based high DOF trackers can estimate image transformations such as affine and homography of a planar object region (or contour) from one frame to the next [285]. Many template trackers using different appearance models [166], [221], [62] and search methods [14], [19] have been proposed and achieve good performance. However, they are highly dependent on stable textures and sensitive to the presence of local minima. Keypoints based approaches [226], [77] are relatively robust to local minima, but they require many accurate feature points to be detected that can be difficult to achieve in practice.

Non-planar contours tracking can be approached as a pose estimation problem [48] when 3D models are available. In unstructured, natural environments 3D models are seldom available. In these cases non-planar contours tracking can be approached as non-rigid tracking. Pixel-wise segmentation based [82] and contour evolving based [206], [21], [236] methods are usually employed to track those objects. However, segmentation based methods do not work well in tracking targets whose appearances change significantly or targets which are comprised of several regions with great differences. Given an initial contour from the previous frame, contour based tracking is performed by searching the target contour based on minimizing a suitable energy [285]. These methods are more likely to trap in local minima in cluttered environments [201].

Another promising technique for closed boundary tracking is perceptual grouping, which has been widely used in salient closed boundary extraction from static images [66], [253], [233]. Schoenemann and Cremers [222] perform contour tracking by integrating pixel-wise perceptual grouping and elastic shape matching into one solvable optimization problem. Unfortunately, this method is not real-time without using GPU processing, ruling out light weight or low power robotics applications. In [162], rough contour tracking



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

Figure 4.10. Illustration of closed boundary tracking: (a) Current image and the tracked boundary (B_p) from the last frame (yellow polygon), (b) Detected line segments, (c) Gap filling among line segments in the buffer (green) region of boundary B_p, (d) Tracked boundary (B_c) of the current frame.

and shape context matching are performed separately to improve time efficiency and handle cluttered background. However, the use of a fixed shape template restricts the ability to track non-planar closed boundaries with out-of-plane motion. Despite all different attempts, tracking of boundary targets in unstructured environment is still a challenging problem. This paper address this problem by presenting a novel line segments grouping based method for tracking salient closed boundaries. Our key contributions are:

- We adapt the salient closed boundary extraction workflow for real-time visual tracking.
- We define a salient closed boundary tracking criterion by combining a boundary grouping cost [233] and a regularization constraint on the boundary's area variation, which improves the tracking robustness greatly.
- To evaluate the performance of our tracking scheme, we collected and annotated nine video sequences (9598 frames) of typical closed boundaries, which are challenging to track in real robot applications.

We implement our tracking method[2] and test it on our newly collected real world video sequences[3], and compare it against state-of-the-art trackers: RKLT [302], ESM [19], HoughTrack [82] and a tracker adapted from the contour grouping method RRC [233]. We also test the performance of our method

[2] `https://github.com/NathanUA/SalientClosedBoundaryTracking`
[3] `https://github.com/NathanUA/SalientClosedBoundaryTrackingDataset`

during a real robot arm experiment following a moving bowl and pouring cereal into it.

4.6.1.3 Proposed Method

(1) Problem Formulation

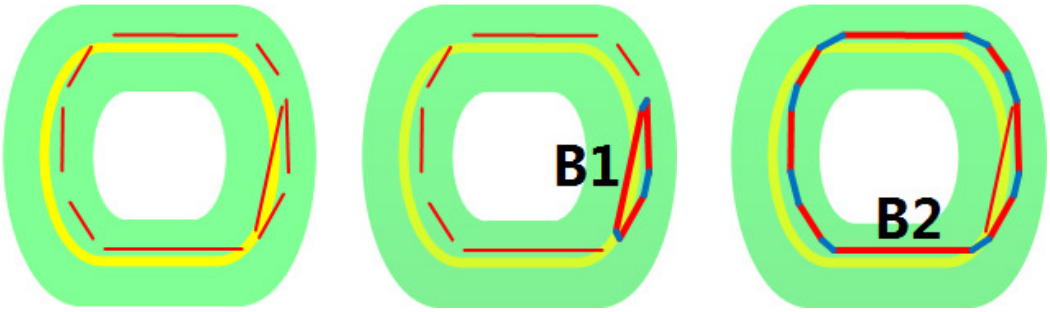
Given a video sequence, we refer to the process of extracting corresponding salient closed boundaries from sequential video frames as boundary tracking. The target closed boundary is usually initialized by selecting a coarse polygon manually in the first frame. The main idea of the coming frames' boundary tracking is identifying a subset of line segments produced by line detector and connecting them to form a closed boundary which corresponds to the boundary tracked in the previous frame, see Fig. 4.10.

Therefore, the closed boundary tracking problem can be reduced to a prior shape constrained perceptual grouping problem. We define a tracking criterion which takes both grouping cost and shape constraint into consideration. The grouping cost Γ, introduced in [233], is given by:

$$\Gamma(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds}{\iint_{\mathcal{R}} dA}, \tag{4.12}$$

where $\oint_{\mathcal{B}} w(e)ds$ denotes the summation of the gap filling segments (blue segments in Fig. 4.10(c)) length along the boundary $\mathcal{B}$. The denominator $\iint_{\mathcal{R}} dA$ is the area of region $\mathcal{R}$ which is enclosed by the boundary $\mathcal{B}$.

Although the distance based filtering (the green region shown in Fig. 4.10(c)) excludes many unrelated line segments, the above grouping cost is still not able to handle grouping illusions caused by noisy segments. As shown in Fig. 4.11, it is clear that $\Gamma(\mathcal{B}_1) < \Gamma(\mathcal{B}_2)$ which results in an incorrectly grouped boundary $\mathcal{B}_1$. To eliminate those obviously erroneous cycle candidates without significantly increasing the time complexity, we propose to introduce a simple area similarity $S_{\mathcal{B}p\_\mathcal{B}c} < S_e$ (e.g. 0.9, this threshold is set based on experiences. The overall tracking performance is relatively insensitive to the small variations on this threshold because the final optimal cycle candidate is selected based on the grouping cost.) between the searched boundary ($\mathcal{B}_c$) and the prior shape



(a) Lines within buffer region of prior boundary (b) Wrong grouping (c) Correct grouping

Figure 4.11. Illustration of wrong boundary grouping: Yellow contours are prior boundaries. Line segments noise often results in wrong grouping of boundary (B_1) without using area constraint.

($\mathcal{B}_p$) to constrain the grouping process as follows:

$$S_{\mathcal{B}p\_\mathcal{B}c} = min(\frac{\iint_{\mathcal{R}p} dA}{\iint_{\mathcal{R}c} dA}, \frac{\iint_{\mathcal{R}c} dA}{\iint_{\mathcal{R}p} dA}), \tag{4.13}$$

where $\iint_{\mathcal{R}p} dA$ is the area of region $\mathcal{R}p$ which is enclosed by the prior shape boundary $\mathcal{B}p$. $\mathcal{B}p$ is the initialization or the tracked boundary from the last frame. ($\mathcal{B}c$ is the to-be-tracked boundary).

To solve the grouping problem, we map the detected and generated (fill-in) line segments and their endpoints to an undirected graph $G = (V, E)$. Line segments and endpoints correspond to graph edges and graph vertices respectively, and thus closed boundaries correspond to graph cycles. Now, the problem of closed boundary grouping is converted into an optimal graph cycle searching problem. We develop a novel graph based optimization method to find the optimal boundary (quasi-optimum) in real-time.

The workflow of our salient closed boundary tracking method is shown in Fig. 4.12. First, we introduce the line detection and filtering. Then, the details of gap filling are presented, followed by graph modeling and optimization.

(2) Line Detection and Filtering

Straight line segments are fundamental elements in our tracking method. Considering the multiple steps of the tracking method shown in Fig. 4.10, the selected line segment detector is expected to produce relatively high quality line segments and run faster than real-time. There are many different straight

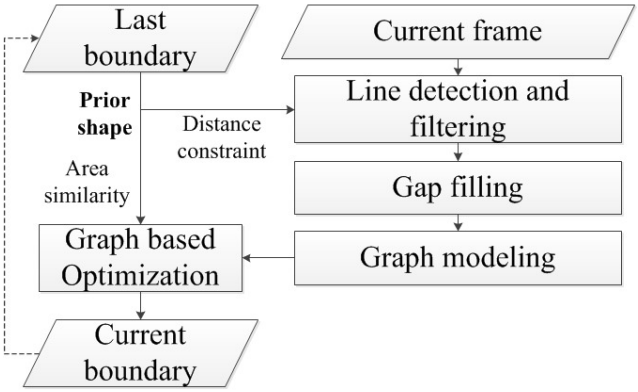


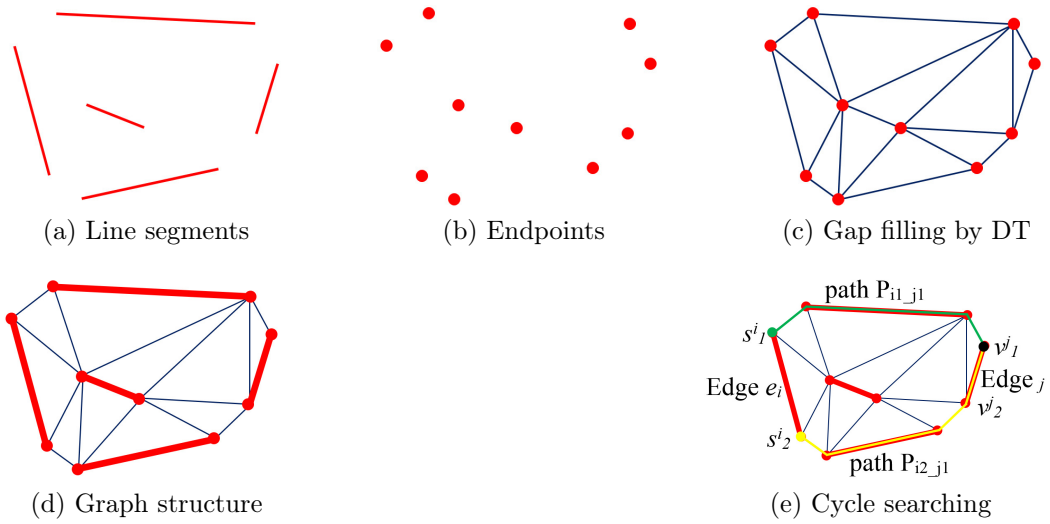
Figure 4.12. Workflow of boundary tracking.

line segment detectors, such as Hough Transform method [15], LSD [244], EDLines [3], MCLSD [5] and so on. Both the performance and running speed of Hough Transform method are not able to satisfy our requirements. Compared to Hough Transform method, LSD and MCLSD are able to produce higher quality line segments. However, they are still slower than real-time in our settings. Therefore, we use EDLines, a faster-than-real-time line segments detector in our settings, for automatic detection of boundary line segments. Detected line segments are represented by pairs of endpoints. In each incoming frame, line detection is conducted on the whole frame. Lines of interest are filtered by a distance constraint (with similar effect of the green buffer region shown in Fig. 4.10(c)) from the previous boundary.

Specifically, three distances of a line from its two end-points and mid-point to the prior boundary are computed. If the average of these distances is smaller than certain threshold (e.g. 20 pixels), it will be retained. Lines of interest are not directly detected from the frame subregion defined by the green buffer because masking an image with an irregular buffer region takes more time.

(3) Gap Filling

After obtaining the line segments of interest, Delaunay Triangulation (DT) [200] is introduced to generate virtual fragments and fill the gaps among disconnected segments, as illustrated in Fig. 4.13. First, line segments are simplified as endpoints (see Fig. 4.13(a) and Fig. 4.13(b)). Then, DT is conducted on these endpoints (Fig. 4.13(c)). Finally, we superimpose the detected line



(a) Line segments (b) Endpoints (c) Gap filling by DT

(d) Graph structure (e) Cycle searching

Figure 4.13. Graph structure construction by gap filling (a)-(d) and Cycle candidates searching by BDSP (e): Edge e_i is the current edge and v_1^j is the third vertex. Edge e_i, shortest paths P_{i1_j1} and path P_{i2_j1} construct a closed cycle candidate.

segments in Fig. 4.13(a) over the generated DT edges in Fig. 4.13(c). Generated DT edges that overlap the detected line segments are removed. The final result of gap filling is an undirected graph structure as shown in Fig. 4.13(d). To distinguish two kinds of line segments, we refer to the detected line segments (the red lines in Fig. 4.13(d)) as *detected* segments and the gap filling segments (the blue lines in Fig. 4.13(d)) as *generated* segments.

(4) Graph Construction

The gap filling process constructs the structure of an undirected graph $G = (V, E)$, which maps endpoints and segments (both *detected* and *generated*) to graph vertices V and edges E respectively. Then, we define the edge-weight function for each edge $e \in E$ similar to [233]. Given a graph edge e_i, its weight is set by (4.2) as follow:

$$w(e_i) = \begin{cases} 0 & e_i \text{ corresponds to a } \textit{detected} \text{ fragment } s_i \\ |P_1^i P_2^i| & e_i \text{ corresponds to a } \textit{generated} \text{ fragment } s_i \end{cases} \tag{4.14}$$

where $|P_1^i P_2^i|$ is the length of the corresponding fragment s_i of the graph edge e_i.

(5) Graph based Optimization

Now, our goal is to find the optimal graph cycle which has the minimum boundary cost, based on (4.12), and satisfies the similarity constraint in (4.13) simultaneously. The key problem is that both the boundary cost and the similarity constraint cannot be determined when the boundary itself is unknown. Furthermore, they are difficult to be integrated into one cost function. So we utilize our novel BDSP method to obtain the optimal cycle. Similar to the last application, this method here still has two steps: generating boundary candidates and finding the optimal one from these candidates.

Given a graph which contains n *detected* line segments, we sample half $n/2$ of them and search $(n-1)$ (the current segment is excluded) cycle candidates for each sampled (*detected*) segment. As shown in Fig. 4.13(e), for each sampled *detected* edge e_i(edges with odd or even indices), we search shortest paths from its two vertices s_1^i and s_2^i to the same third vertex v_1^j using Dijkstra [59]. The weight of e_i is set to infinity other than its original weight zero during searching. The edge e_i, shortest paths P_{i1_j1} and P_{i2_j1} construct a cycle candidate C_{i_j1}. Vertex v_1^j and Vertex v_2^j belong to the same edge e_j and the weight of e_j is zero, thus, taking v_1^j or v_2^j as the third vertex usually produce the same cycle. Hence, the total number of the cycle candidates is $n(n-1)/2$. Having obtained these cycle candidates, we can search for the optimal closed boundary by computing their boundary costs (4.12) and similarity constraints (4.13) easily. The optimal boundary searching algorithm is shown in Algorithm 1.

4.6.1.4 Experimental Results

We validate our tracking scheme using a number of comparative experiments and a real robot arm experiment.

(1) Dataset and Evaluation Measure

We collected nine video sequences of salient closed boundaries, as shown in Fig. 4.15. Each sequence is about 30 sec (30 fps) and the frame size is 640×480 (width×height). There are 9598 frames in total. In each sequence, different motion styles such as translation, rotation and viewpoint changing are all performed. We annotate them by drawing polygons which are well

Algorithm 1 Optimal cycle searching

Input: Undirected graph $G = (V, E)$ and a shape prior represented by a set of ordered points

Output: The optimal cycle C_{opt}

for $i = 0; i < n; i+ = 2$ **do**
  Use BDSP to search cycle candidates $C_{i\bullet}$
  for $j = 0; j < 2(n-1); j+ = 2$ **do**
    if $i == 0 \&\& j == 0$ **then**
      $C_{opt} = C_{ij}$
    end if
    if $S_{C_{ij}\_Bp} > S_e$ **then**
      if $\Gamma(C_{ij}) < \Gamma(C_{opt})$ **then**
        $C_{opt} = C_{ij}$
      end if
    end if
  end for
end for
return C_{opt}

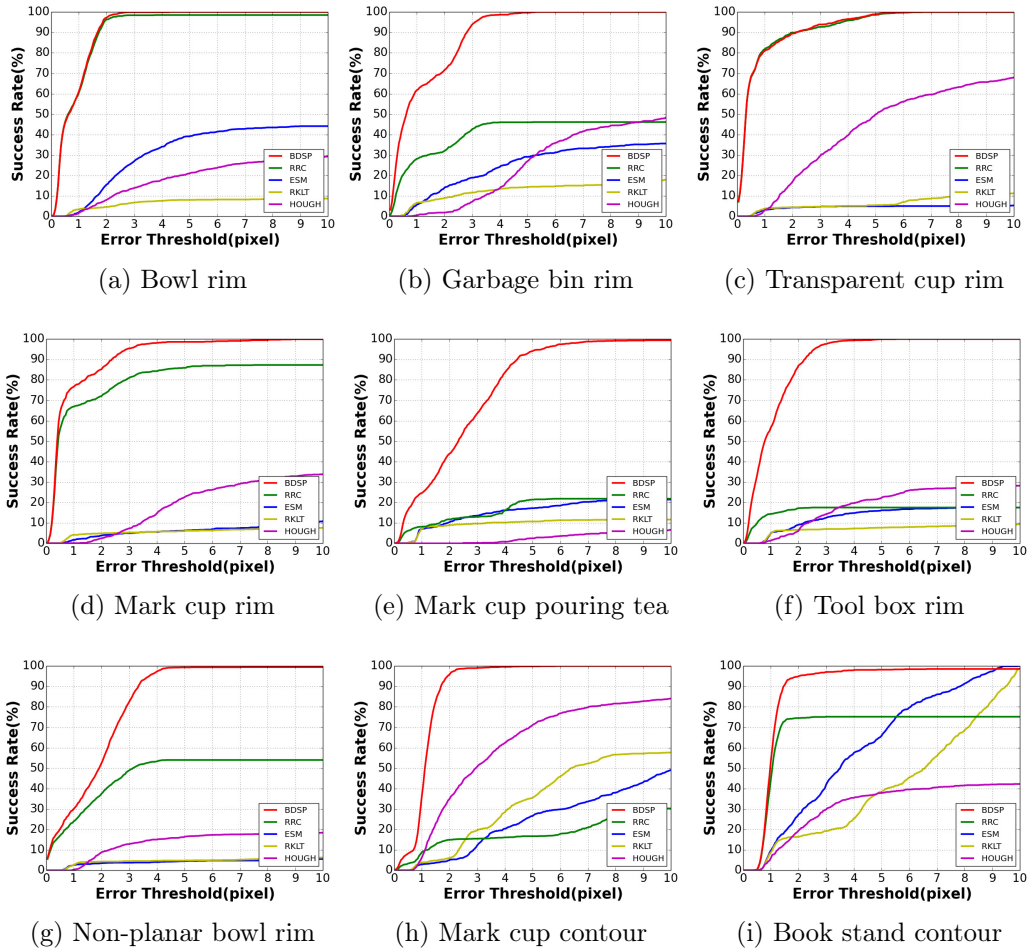
Notes: $C_{i\bullet}$ are the $(n-1)$ cycle candidates related to edge e_i.

matched with the salient closed boundaries in human vision.

To evaluate our proposed method quantitatively, we define the error metric as the alignment error (E_{AL}) of tracked closed boundary and ground truth (B_{gt}) as: $E_{AL} = max\{\frac{B_i \otimes Dist_{B_{gt}}}{P_{B_i}}, \frac{B_{gt} \otimes Dist_{B_i}}{P_{B_{gt}}}\}$, where $\otimes$ denotes convolution, B_i is the boundary binary image, $Dist_{B_i}$ is the distance transform image of B_i and P_{B_i} indicates the perimeter of boundary B_i. We use the success rate to measure a tracker's overall accuracy [229]. The success rate on a sequence is defined as the ratio of frames where the tracking error E_{AL} is less than a threshold of e_p pixels and the total frames.

(2) Qualitative and Quantitative Comparison

We compare our tracking method BDSP against following methods: ESM [19] which is a popular registration based homography tracker, RKLT [302] which is a cascade registration based tracker that can handle partial appearance changing and occlusion by RANSAC, HoughTrack [82] which is a state-of-the-art segmentation based tracker that can provide us accurate contour and a tracker adapted from edge grouping method RRC [233]. Both ESM and RKLT are tested with appearance model NCC, which are implemented in a



(a) Bowl rim (b) Garbage bin rim (c) Transparent cup rim

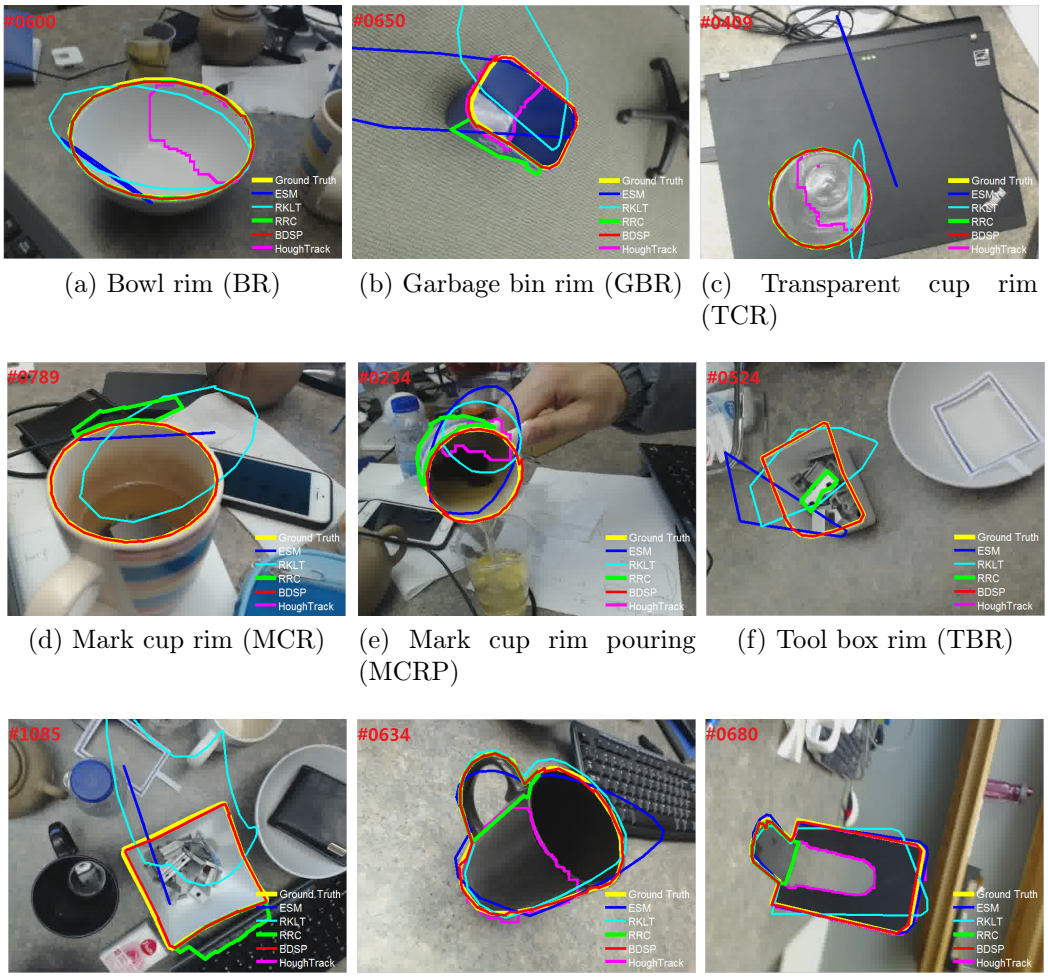
(d) Mark cup rim (e) Mark cup pouring tea (f) Tool box rim

(g) Non-planar bowl rim (h) Mark cup contour (i) Book stand contour

Figure 4.14. Success rates of BDSP, RRC, ESM, RKLT and HoughTrack on the video sequences of Fig. 4.15.

modular tracking framework (MTF) [229]). We initialize them by selecting a quadrilateral which encloses the target boundary at the first frame. Then boundaries of following frames are computed by homography transformations with respect to the first frame (all boundaries are assumed to be planar). We modified RRC by substituting its line detector for EDlines [3] and added a buffer search region as shown in Fig. 4.10(c).

The success rate curves of the above four methods and our method are illustrated in Fig. 4.14. As we can see, the proposed method (BDSP) performs better than others in almost all of these sequences. Both registration based trackers ESM and RKLT fail quickly because they are sensitive to appearance changing, as blue and cyan boundaries shown in Fig. 4.15(a) to Fig. 4.15(g).



(a) Bowl rim (BR) (b) Garbage bin rim (GBR) (c) Transparent cup rim (TCR)

(d) Mark cup rim (MCR) (e) Mark cup rim pouring (MCRP) (f) Tool box rim (TBR)

(g) Non-planar bowl rim (NBR) (h) Mark cup contour (MCC) (i) Book stand contour (BSC)

Figure 4.15. Tracked boundaries on typical frames.

Although HoughTrack performs better than registration based trackers thanks to its model updating, it corrupts quickly when the content of target region are heterogeneous as pink boundaries shown in Fig. 4.15. Fig. 4.15(a) and Fig. 4.15(c) show an empty bowl and an empty transparent cup with relative clean background. The corresponding tracking results illustrated in Fig. 4.14(a) and Fig. 4.14(c) show that the RRC tracker produces almost the same success rate with our method. But it is very susceptible to noise, as the tracked green boundaries shown in Fig. 4.15. The intact video results are included in the supplementary video[4].

[4] `https://youtu.be/RXjDOyHkukI`

Table 4.2. Graph scale and time efficiency.

Video	BR	GBR	TCR	MCR	MCRP	TBR	NBR	MCC	BSC
Edges	445	508	550	672	753	362	419	505	286
Nodes	80	88	96	112	124	62	70	89	53
lt(ms)	4.34	8.78	4.15	4.69	5.17	4.33	5.09	5.07	3.79
gt(ms)	14.45	18.85	19.00	26.10	32.13	8.68	16.93	17.72	6.56
fps	54.12	37.55	43.21	32.48	26.80	76.81	59.74	43.88	96.57

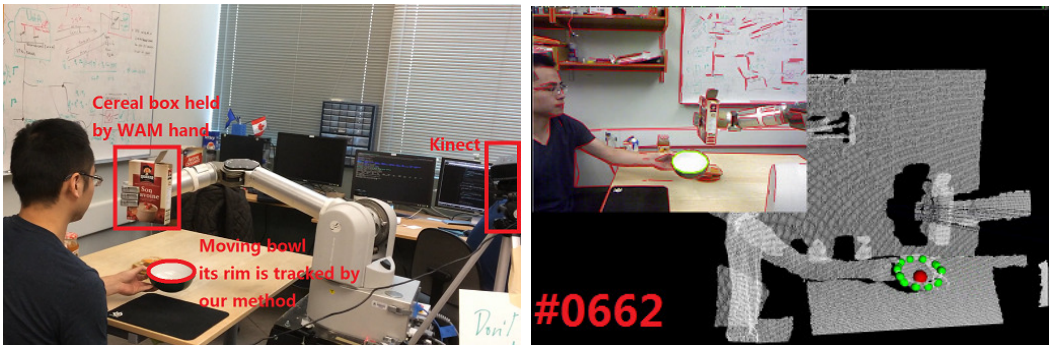
Notes: ms denotes milliseconds.

We also measured the average processing speed of our method for each of the nine video sequences on a machine with a quad core 3.10 GHz Intel Core i5 processor, 16 GB RAM and Ubuntu 14.04 64-bit OS. Our method is implemented in C++ using OpenCV and Boost library. Table 4.2 illustrates the average graph size, which is indicated by numbers of edges and nodes, the average time costs of line detection (lt), grouping time (gt) and the average frequency per second (fps). The total tracking time per each frame contains line detection time and grouping time. We compute the instantaneous fps of each frame and then average them over the whole sequence to get the average fps, as shown in the last row of Table 4.2. As we can see our method is acceptable for real-time tracking.

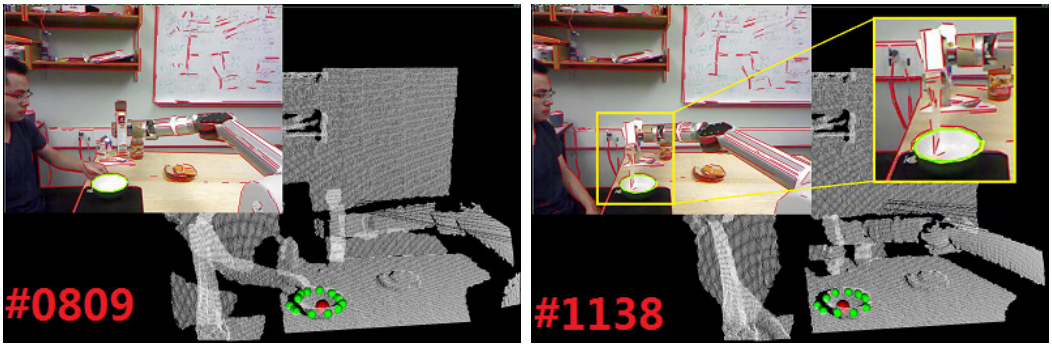
(3) Robot Arm Poursing Experiment

Our salient closed boundary tracker has been used successfully in a real robot arm pouring experiment. The task is to track and follow a moving bowl and then pour cereal into it. The difficulties of this experiment are that the bowl is non-Lambertian and has no salient textures.

The setup of our experiment is shown in Fig. 4.16(a). The system includes a set of WAM arm and a Kinect. The 3D coordinates of the WAM arm and the Kinect are registered. We initialize the bowl rim and track it in RGB video stream captured by the Kinect. Meanwhile, we map the tracked closed boundary, which is represented by a set of 2D image points, to 3D points acquired by the Kinect depth sensor, as shown in Fig. 4.16(b) and Fig. 4.16(c). The centroid of the bowl is computed based on these mapped 3D contour points and is taken as the pouring target position. We pre-compute



(a) Experiment setup (b) Following



(c) Following (d) Following and accurate pouring

Figure 4.16. Robot pouring experiment: (a) A human is moving the bowl continuously under the surveillance of a Kinect. (b)(c) Our algorithm tracks the bowl rim and maps its 2D image points to 3D points in the robot coordinates through the Kinect which is registered with the robot coordinates, then, makes the robot arm follows the moving bowl. (d) The robot hand pours the cereal into the bowl accurately according to our tracking result.

the shifting of the WAM hand to the bowl centroid. When the distance $Dist_{r\_b}$ between the centroid of the tracked bowl and the robot hand satisfies certain thresholds($e_{low} < Dist_{r\_b} < e_{high}$), the WAM arm will pour the cereal into the bowl, as shown in Fig. 4.16(d). Without having the tracking points of the contour provided by our tracker it will be very difficult to find the 3D center of the bowl with any other types of trackers. The experiment shows that our tracker is stable and efficient in real robot application. A demonstration of the pouring task can be seen in the accompanying video.

4.6.1.5 Summary

This chapter presents a novel real-time method for salient closed boundary tracking. By combining a saliency measure and an area constraint as tracking criterion, the proposed method improves the tracking performance greatly. A bidirectional shortest path based boundary candidates searching algorithm enables the real-time solvability of the combined tracking criterion. We validated it quantitatively on various real-world video sequences. Since it is robust and fast enough, it has been used successfully in real robot pouring experiment where other trackers have failed. Our future work will focus on addressing the problem of tracking boundaries, which are hard to be described by straight line segments, as well as the problem of self-occlusions.

4.6.2 Salient Closed Boundary Tracking via Edge Fragments Perceptual Grouping

4.6.2.1 Overview

In this section, we propose a real-time method for accurate salient closed boundary tracking via a combination of shape constraints and perceptual grouping on edge fragments. Particularly, we encode the Gestalt law of proximity and the prior shape constraint in a novel ratio-form grouping cost. The proximity and prior constraint are depicted by the relative gap length and average distance difference along the to-be-tracked boundary with respect to its area. We build a graph using the detected edge fragments and in-between gaps. The grouping problem is formulated as searching for a special cycle in this graph with a minimum grouping cost. To reduce the search space and achieve real-time performance, we propose a set of novel techniques for efficient edge fragments splitting and filtering. We evaluate this method on a public real-world video dataset against other methods. The average alignment errors of different sequences achieved by our method are mostly less than 1 pixel, an improvement over state-of-the-art methods.

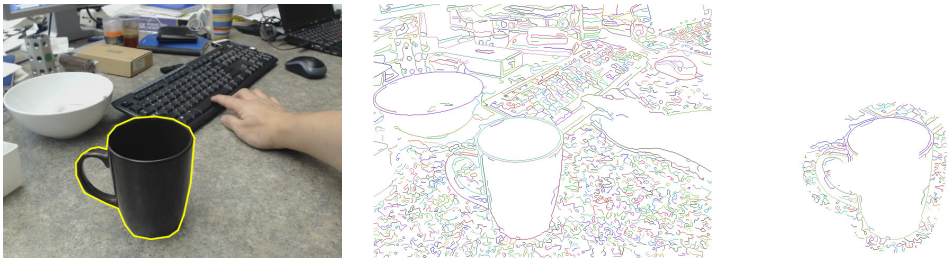
4.6.2.2 Motivation

Real-time salient closed boundary tracking is an important yet challenging issue in the computer vision community. It is also an important technique for vision guided robot applications. Pertinent methods can be categorized into three main classes: 1) *template based methods*, which estimate the geometric transformations, such as affine and homography, of planar rigid target contours from one frame to the next one [166], [271], 2) *region segmentation based methods*, which determine boundaries on each frame by segmenting images into foreground and background using techniques like graph-cuts [82], level sets [284], etc. 3) *perceptual grouping based methods*, which search for a special cycle of low level primitives (e.g. edge fragments, line segments) forming the closed boundaries [184]. The proposed tracking method of this paper belongs to the third class.

Perceptual grouping algorithms have been widely used in contours completion [118], [180], [208] and salient closed boundaries extraction [7], [66], [172], [184], [233], [253]. Their basic principles draw from Gestalt laws [263] including proximity, good continuation, closure, etc. However, most of these methods focus on extracting contours by exploring grouping cues from the current image and cannot be used for closed boundary tracking directly.

Tracking a salient closed boundary through a video sequence can be formulated as a prior shape constrained perceptual grouping problem. Elder *et al.* [65] developed a framework of combining prior probabilistic knowledge of the target appearance with probabilistic models for contour grouping. Schoenemann and Cremers [222] chose the tangent angles of curves as a prior shape constraint term and combined it with pixel gradients, penalties for shape stretching and shrinking to formulate a pixel-wise elastic shape matching and grouping model. Neither of these methods run in real-time without GPU acceleration. Qin *et al.* [202] formulated a boundary tracking criterion by combining an area variation constraint and a grouping cost [233], which is defined as the ratio of gap length and region area enclosed by the boundary. They also developed a shortest path [59] based algorithm to search for the optimal boundary. However, the area variation constraint is weak and does not preserve fine structures. In addition, the line segments used in their methods can not fit curved boundaries well.

In this chapter, we develop a perceptual grouping method that combines Gestalt saliency and prior shape information. We demonstrate a real-time CPU implementation. Contributions include: 1) We define a new grouping cost by adding a distance difference based prior shape constraint to the grouping cost. 2) We propose a novel technique for fast edge segments splitting to speed up the grouping process. It generates high quality edge fragments in less than one milliseconds per frame. Redundant edge fragments are removed according to their lengths and average distance differences. 3) We implement this edge fragments based closed boundary tracking method and test it on a public real-world video dataset. It achieves state-of-the-art performance, outperforming a method adapted from [233] and the method proposed in [202].



(a) (b) (c)

(d) (e) (f)

Figure 4.17. Processes of our salient closed boundary tracking. (a) current frame and the manually initialized boundary in yellow (or that tracked from the previous frame); (b) edge segments detected by Edge Drawing; (c) filtered edge segments pixels according to their distances to the prior boundary (yellow boundary in (a)); (d) edge fragments generated from edge segments splitting and length based filtering; (e) edge fragments filtered by the distance difference over length $\frac{DD_i}{L_i}$; (f) edge fragments selected for grouping the current salient closed boundary.

4.6.2.3 Proposed Method

(1) Tracking via Prior Shape Constrained Grouping

Given a video sequence and a salient closed boundary defined in the first frame, as shown in Fig. 4.17(a), our goal is to track the closed boundary over the entire sequence by identifying and sequentially connecting a set of edge fragments in each frame, Fig. 4.17(f). Compared with grouping the salient closed boundaries from single image, the closed boundary tracking from video sequences can take advantage of both saliency properties derived from Gestalt laws and prior shape constraints from the tracked boundary on previous frame.

We divide prior shape constraints into three levels. First, intact boundary oriented coarse constraints, such as perimeter and area variations. Constraints of this level restrict significant shape variations other than fine structures. Sec-

ond, primitives oriented middle level constraints such as curvature, direction, length and distance properties of edge fragments and line segments. These constraints provide relatively stronger restrictions to shape variations. Third, pixel-wise constraints that can even produce very fine and accurate pixel-by-pixel matching between to-be-tracked boundary and the prior boundary. In this paper, we use the first and second level constraints because the pixel-wise constraints are usually time consuming and hard to be solved in real-time.

To combine the Gestalt cues of the current frame and prior shape information, we first define a grouping cost $\Gamma(\mathcal{B})$ of a closed boundary $\mathcal{B}$ based on the middle level constraints as:

$$\Gamma(\mathcal{B}) = \frac{\oint_{\mathcal{B}} w(e)ds + |DD_{\mathcal{B}}|}{\iint_{\mathcal{R}} dA} \tag{4.15}$$

where $\iint_{\mathcal{R}} dA$ is the area of the region $\mathcal{R}$ which is enclosed by the boundary $\mathcal{B}$. $\oint_{\mathcal{B}} w(e)ds$ denotes the total length of the edge fragments' in-between gap segments along the boundary and $\frac{\oint_{\mathcal{B}} w(e)ds}{\iint_{\mathcal{R}} dA}$ depicts the proximity (saliency) of the boundary [233]. $|DD_{\mathcal{B}}| = \sum_{i \in EF} DD_i$ is the total absolute distance difference. Fig. 4.18 illustrates the distance difference (DD_i) of an edge fragment EF_i which is:

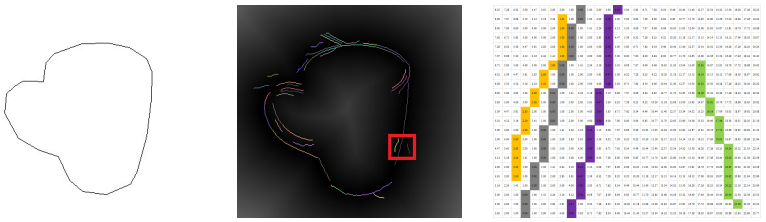
$$DD_i = \sum_{j=1}^{p-1} |dist(P_{j+1}) - dist(P_j)| \tag{4.16}$$

where p is the number of the edge fragment pixels, P_j denotes the j-th pixel in the fragment pixel array and $dist(P_j)$ is its corresponding value on the distance transform map [73] of the prior shape.

In addition to the area constraint used in [202], this paper introduces a perimeter variation constraint to reduce the search space and improve the tracking robustness as follows:

$$\begin{cases} v(P) = min(\frac{P_{prior}}{P_{cur}}, \frac{P_{cur}}{P_{prior}}) < e_P \\ v(A) = min(\frac{A_{prior}}{A_{cur}}, \frac{A_{cur}}{A_{prior}}) < e_A \end{cases} \tag{4.17}$$

where $v(P)$ and $v(A)$ are perimeter and area variations. P_{prior} and P_{cur} are the perimeters of prior and current closed boundaries. A_{prior} and A_{cur} are the



(a) (b) (c)

Figure 4.18. Distance Difference (DD): (a) prior shape; (b) detected edge fragments superimposed on the distance transform map of the prior shape; (c) zoom-in view of region enclosed by the red box in (b), gray pixels belong to the prior shape, colorized edges are extracted from the current frame, the value of each pixel is the Euclidean distance of the pixel to the closest pixel of the prior shape. The effect of DD is similar to the tangent angle [222] but easier to compute in real time.

areas of regions that enclosed by prior and current boundaries. e_P and e_A represent the perimeter and area variation constraints respectively. In fact, these two constraints are used to exclude obviously erroneous closed contour candidates. The thresholds e_P and e_A are set based on the experiences. The final tracking performance is insensitive to the small variations on these two thresholds.

(2) Edge Fragments Detection

We use a real-time edge segment detector Edge Drawing [238] to extract high quality edge segments from incoming frames. Each of the resulting edge segment is a linear pixel chain with one-pixel width. The result of ED is outputted in vector form as an array of chain-wise edge segments, as shown in Fig. 4.17(b). These detected edge segments cannot be used for real-time salient closed boundary grouping directly due to the following reasons: 1) large number of redundant edge segments make the grouping search space huge; 2) foreground and background edge pixels are often improperly identified as one long edge segment. Hence, we propose a novel edge segments splitting technique combined with redundant edge pixels and fragments filtering to obtain a set of well-identified high quality edge fragments.

1) Edge Pixels Filtering

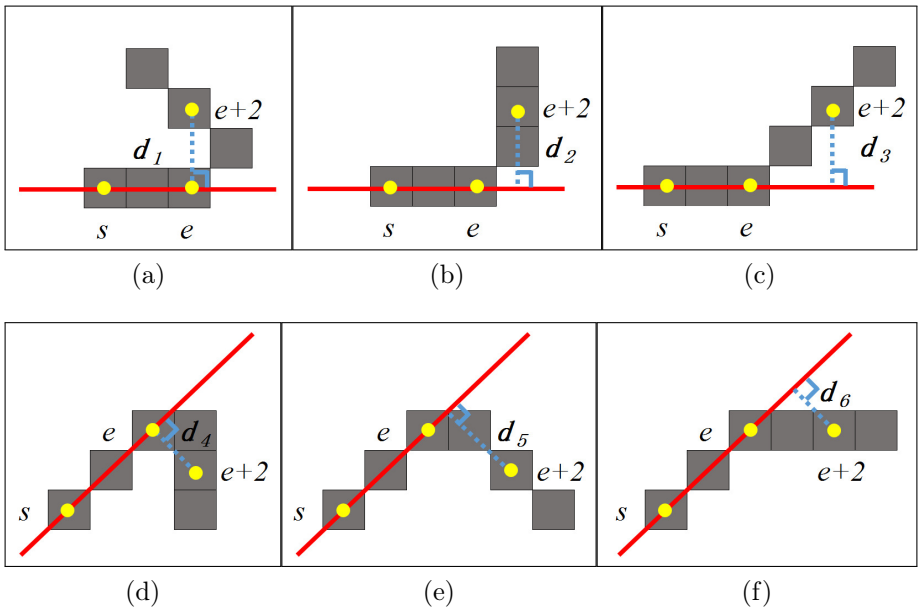


Figure 4.19. Illustration of six basic ways of edge turns. The turning distance is the distance from the extending pixel (e+2) to the l_{se} line: $d_1 = d_2 = d_3 = 2$, $d_4 = d_6 = \sqrt{2}$, $d_5 = \frac{3\sqrt{2}}{2}$. According to these turning distances, we set the splitting threshold to $\sqrt{2} \approx 1.4$ pixels. The threshold of 1.4 is chosen because the turning distances above this threshold indicates that their corresponding turning angles are in the edge splitting range of 45 degrees to 135 degrees. Theoretically, if the turning angle is smaller than 45 degrees, the edge has also to be split. But that barely happens because the edge detector, Edge Drawing, prefers to produce split edges by starting a new edge searching process in that case.

The edge segments detected by ED contains a lot of redundant edge pixels. We filter these pixels according to their distances to the prior shape. Pixels with absolute distances smaller than a threshold are retained, as shown in Fig. 4.17(c). The distance threshold (e.g. 30 pixels) depends on the relative motion speed and the frame rate (or frames per second) of the video.

2) Edge Segments Splitting

Curvature [253], turning angle [208] and line fitting [233] based approaches have been proposed to split detected edge segments into multiple fragments. The curvature based method used in [253] requires spline fitting which is time consuming. Although the line fitting based method [233] is more efficient, it often misses short edge fragments. We develop a novel splitting method based

on an analysis of the six basic ways of edge turns, as shown in Fig. 4.19. Given an edge segment, we split it into one or multiple fragments by traversing it with a step size of two pixels. Particularly, we start a fragment searching by sampling two pixels with indices of s (starting pixel) and $e = s + 2$ (ending pixel). Then, we compute the distance from the extending pixel $(e + 2)$ to the line l_{se} to decide whether to split or not. Based on Fig. 4.19 if the distance is greater than a threshold (1.4 pixels), we split the segment. Another splitting criterion is the middle pixel deviation. If the distance from the middle pixel $(m = \frac{s+e}{2})$ to l_{se} is greater than a threshold (5 pixels) we also split the segment. Otherwise, we extend the current fragment by $e = e + 2$. The edge splitting algorithms used in the previous chapters based on turning angle or least square line fitting cost several milliseconds to process one frame. This turning distance based edge splitting algorithm reduces the processing time to less than one millisecond.

3) Edge Fragments Filtering

Edge segments splitting produces a large number of edge fragments. However, some of them are redundant. To reduce grouping search space and achieve real-time performance, we propose to use their length L_i and average distance difference $\frac{DD_i}{L_i}$ to filter redundant edge fragments. The length based filtering is aiming at excluding those tiny spurious edge fragments, as shown in Fig. 4.17(d). On the other hand, we assume that the to-be-tracked boundary in the current frame is almost parallel to the prior one. The average distance difference depicts the parallelism between fragments and the prior boundary. It filters those edge fragments which are more likely to be perpendicular to the prior boundary, as shown in Fig. 4.17(e).

(3) Graph Modeling and Optimization

In this section, we construct an undirected graph $G = (V, E)$ [202] data structure of the remaining edge fragments and assign weights to the graph edges according to our tracking cost function. First, to model the graph structure, we take the endpoints of those disconnected edge fragments as graph vertices, as illustrated in Fig. 4.20(b). Then, Delaunay Triangulation (DT) [200] is used to generate gap filling segments and connect filled vertices, Fig.

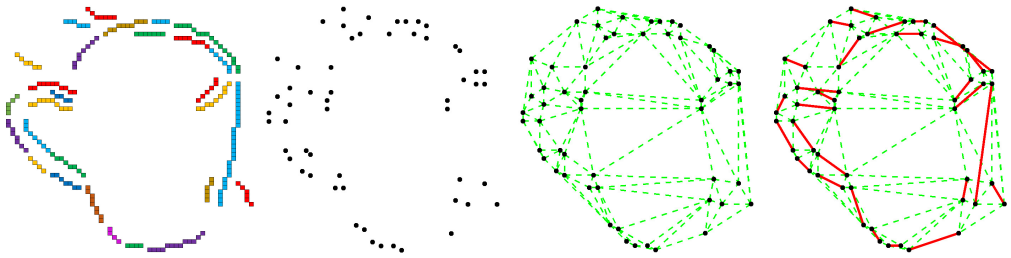


Figure 4.20. Illustration of graph modeling using detected edge fragments: (a) detected and filtered edge fragments and there are no intersections and common endpoints; (b) endpoints of those detected edge fragments; (c) Delaunay Triangulation (DT) of the endpoints; (d) graph structure constructed by the union of DT edges (c) and the corresponding degenerated edges from (a).

4.20(c). Fig. 4.20(d) shows the graph structure whose edges are the union of the green dotted lines (DT edges) and the red solid lines (edges corresponding to the edge fragments in Fig. 4.20(a)).

Each graph edge is assigned two weights $w_L(e_i)$ and $w_{DD}(e_i)$ as follows:

$$w_L(e_i) = \begin{cases} 0 & e_i \text{ corresponds to an edge fragment} \\ |P_1^i P_2^i| & e_i \text{ corresponds to a DT edge} \end{cases} \tag{4.18}$$

where $|P_1^i P_2^i|$ is the length of corresponding gap filling line segment $P_1^i P_2^i$ of the graph edge e_i.

$$w_{DD}(e_i) = DD_i \tag{4.19}$$

where DD_i is the distance difference of the edge e_i from (4.16).

The grouping cost in (4.15) is computed as:

$$\Gamma(B) = \frac{\sum_{i \in E(\mathbf{C})} w_L^i + \sum_{i \in E(\mathbf{C})} w_{DD}^i}{Area(\mathbf{C})} \tag{4.20}$$

where $\mathbf{C}$ is a graph cycle that corresponds to a closed boundary B, $E(\mathbf{C})$ is the set of edges of $\mathbf{C}$. $Area(\mathbf{C})$ is the area of the corresponding polygon of the graph cycle and it approximates the closed boundary area $\iint_{R(B)} dxdy$ in (4.15) to simplify the area computation.

Finding a graph cycle that minimizes the grouping cost $\Gamma(B)$ (4.20) is a minimum-ratio-cycle problem and can be solved by algorithms proposed in

[253] and [233] in polynomial time. However, we have to integrate the length and area variation constraints (4.17) into the optimization problem. To this end, we use a bidirectional shortest path based search strategy of [202] to generate number of graph cycle candidates and find the optimal one according to our tracking measure defined in (4.17) and (4.20).

4.6.2.4 Experimental Results

To evaluate the performance of our tracking method, we tested it on a public dataset [5] [202] which has nine real-world video sequences and 9598 frames in total. Each video sequence is about 30-45 seconds (30 fps, size: 640× 480) and contains a single moving salient closed boundary with different types of motions including translation, rotation, zooming in/out and even out of plane rotation. These to-be-tracked targets includes planar/non-planar, non-Lambertian and irregular closed boundaries with/without clustered backgrounds, see Fig. 4.22 and 4.23.

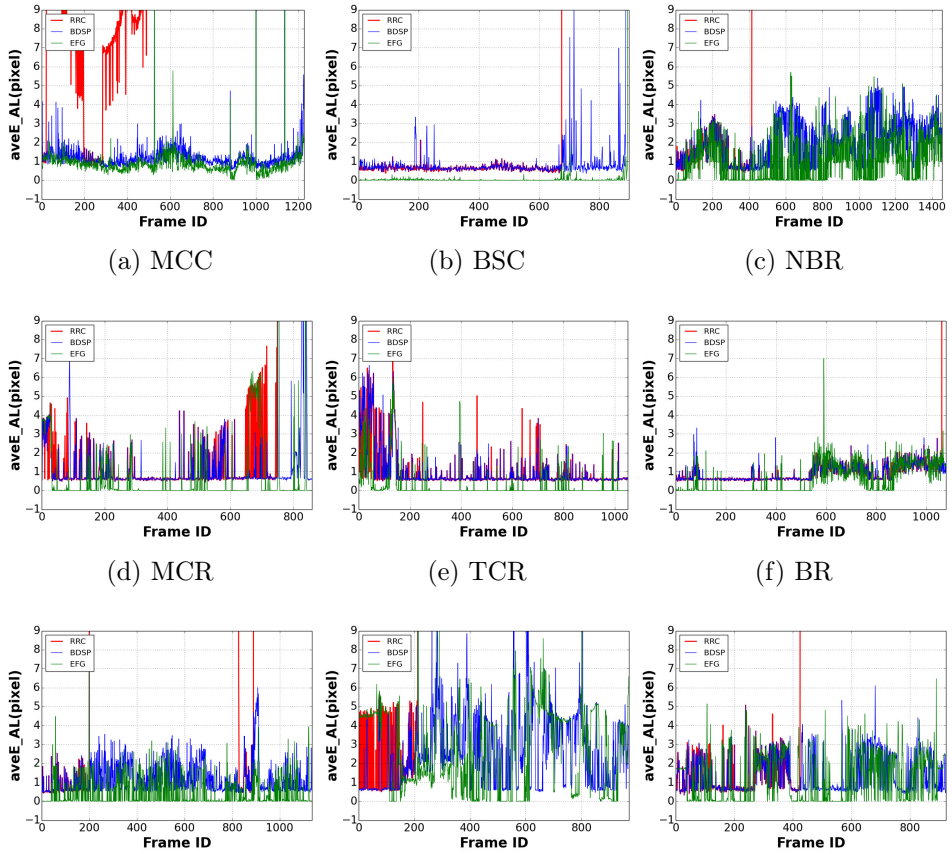
(1) Quantitative and Qualitative Evaluation

To evaluate our tracking method quantitatively, we introduce the maximal average cross alignment error $aveE\_AL = max\{B_i \otimes Dist_{B_{gt}}/P_{B_i}, B_{gt} \otimes Dist_{B_i}/P_{B_{gt}}\}$ where B_i and B_{gt} represent the binary images of the tracked boundary and the corresponding ground truth. $Dist_{B_i}$ and $Dist_{B_{gt}}$ are the distance transform maps of B_i and B_{gt}. $\otimes$ denotes the summation of element(pixel)-wise multiplication. P_{B_i} and $P_{B_{gt}}$ are the number of pixels in the tracked boundary and the corresponding ground truth.

We compared our *Edge Fragments Grouping* (EFG) based method with two others: a real-time tracker which is adapted from the regional information combined ratio contour (RRC) algorithm [233] and a state-of-the-art closed boundary tracking method (BDSP) which only uses a saliency measure and an area constraint [202].

Fig. 4.21 plots the alignment error $aveE\_AL$ of each frame achieved by these three methods. Table 4.3 shows the average $aveE\_AL$ of each sequence. As illustrated in Fig. 4.21 and Table 4.3, the RRC tracker fails quickly on se-

[5] `https://github.com/NathanUA/SalientClosedBoundaryTrackingDataset`



(a) MCC (b) BSC (c) NBR

(d) MCR (e) TCR (f) BR

(g) TBR (h) MCRP (i) GBR

Figure 4.21. Average alignment error $aveE\_AL$ of the closed boundary tracking achieved by trackers RRC, BDSP and our EFG. Mark Cup Contour (MCC), Book Stand Contour (BSC), Nonplanar Bowl Rim (NBR), Mark Cup Rim (MCR), Transparent Cup Rim(TCR), Bowl Rim(BR), Toolbox Rim(TBR), Mark Cup Rim Pouring (MCRP), Garbage Bin Rim(GBR).

quences Mark Cup Contour **MCC** (failure starts from frame: 285/total frame number: 1226), Book Stand Contour **BSC** (676/899), Nonplanar Bowl Rim **NBR** (416/1454), Mark Cup Rim **MCR** (750/859), Transparent Cup Rim **TBR** (200/1135), Mark Cup Rim Pouring **MCRP** (213/971) and Garbage Bin Rim **GBR** (426/924) and produces large average alignment errors. These failures are mainly caused by cluttered backgrounds, see the first two rows of Fig. 4.22, and relatively complex boundaries, as shown in the last two rows of Fig. 4.22. Compared with RRC, the BDSP tracker is more robust because of its area variation constraint, as shown in Table 4.3. However, boundaries

Table 4.3. Average $aveE\_AL$ (pixel) of each sequence.

Video	**MCC**	**BSC**	**NBR**	**MCR**	**TCR**	**BR**	**TBR**	**MCRP**	**GBR**
RRC	56.09	58.06	120.48	19.29	1.01	3.37	74.09	58.18	108.28
BDSP	1.31	1.24	1.99	1.18	1.03	0.94	1.21	**2.73**	1.38
EFG	**0.99**	**0.18**	**1.08**	**0.68**	**0.26**	**0.61**	**0.46**	2.80	**1.22**

tracked by BDSP often have some erroneous wiggles, as illustrated in the last two rows of Fig. 4.22. The reason for these wiggles is that the area variation constraint is too weak to restricts fine structures.

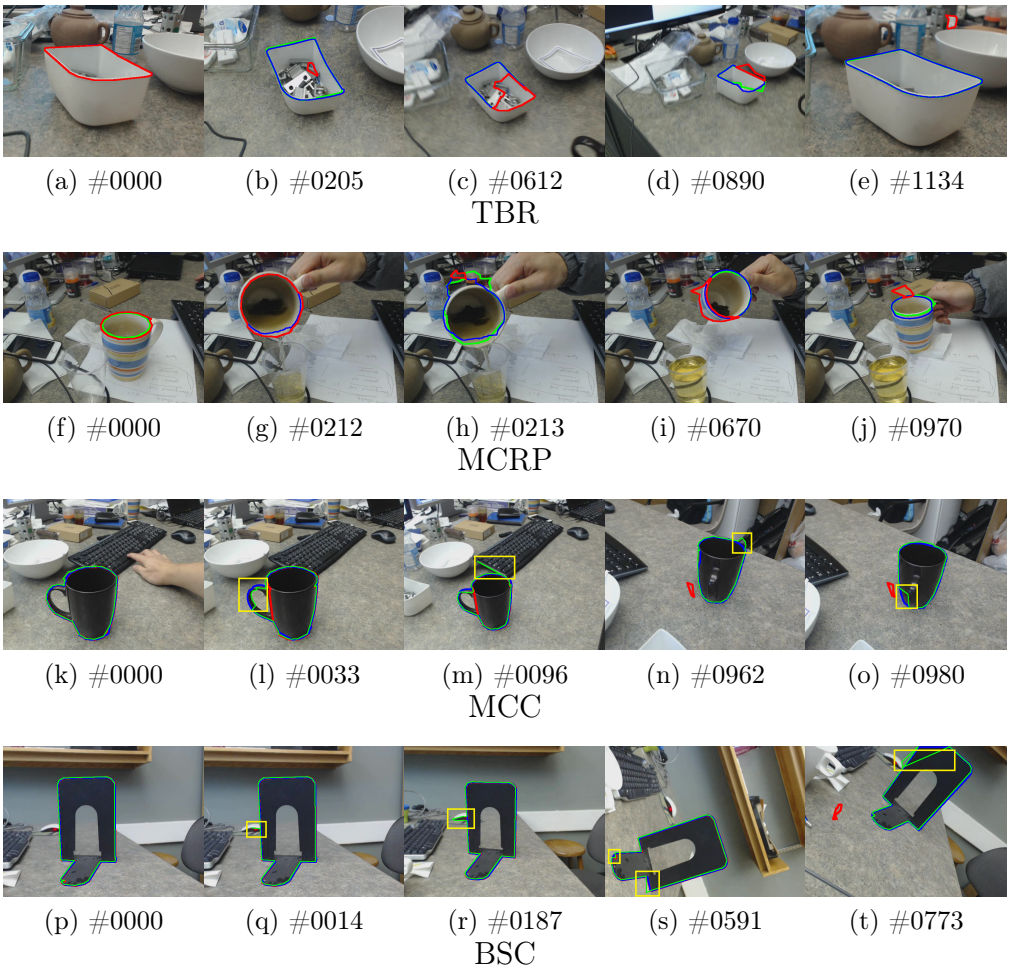
Table 4.4. Average tracking time costs for each video sequence: $aveEFs$ and $aveGFs$ are the average numbers of Edge Fragments and Gap Filling segments. $2 \times aveEFs$ is the number of graph vertices and $aveEFs + aveGFs$ is the number of graph edges. T is the average time cost of each frame tracking in milliseconds. It includes edge segments detection, splitting, filtering, graph construction and optimization. The frame loading time is not included because this process can be implemented in parallel and will not influence the tracking speed significantly.

Video	MCC	BSC	NBR	MCR	TCR	BR	TBR	MCRP	GBR
$aveEFs$	27	23	25	36	30	25	29	35	28
$aveGFs$	145	120	137	200	162	136	155	192	154
$T(ms)$	**28.11**	**20.37**	**21.60**	**36.39**	**26.45**	**22.31**	**26.39**	**35.08**	**30.92**

Our EFG tracker suppresses accidental wiggles effectively through the distance term. As shown in Fig. 4.21 and Table 4.3, almost all of the errors of our EFG tracker on different sequences are smaller than those of RRC and BDSP. Because both RRC and BDSP operate on detected line segments while our EFG splits edge segments into smaller smooth fragments. Fig. 4.23 illustrates the qualitative difference of representing boundaries via line segments and edge fragments. Compared with RRC and BDSP, our EFG tracker fits curved boundaries better and produces higher accuracy.

(2) Run Time

We implemented our tracking method in C++ using OpenCV and Boost library on Ubuntu 14.04 64-bit OS. The time costs are collected by running our tracking method on a quad core 3.10 GHz and 16 GB RAM computer without GPU acceleration. To demonstrate our tracking method runs in real-time, the average tracking time costs of each video sequence are illustrated in Table 4.4.



(a) #0000 (b) #0205 (c) #0612 (d) #0890 (e) #1134
TBR

(f) #0000 (g) #0212 (h) #0213 (i) #0670 (j) #0970
MCRP

(k) #0000 (l) #0033 (m) #0096 (n) #0962 (o) #0980
MCC

(p) #0000 (q) #0014 (r) #0187 (s) #0591 (t) #0773
BSC

Figure 4.22. Failures of RRC and wiggles of BDSP: Red, green and blue boundaries are results of RRC, BDSP and EFG respectively. All of the four rows demonstrate the failure of RRC tracking. The last two rows show the wiggles (within yellow boxes) produced by BDSP.

The average per frame time cost of RRC is in the range of 15 to 30 ms but it fails quickly in most of the sequences. The average time cost of our method is 27.51 ms which is slightly (4.64 ms) more than that of BDSP 22.87 ms, while our method improved the accuracy from 1.45 (BDSP) to 0.92 (see Table 4.4 and 4.3). The 0.53 difference per pixel is usually the result of large wiggles in the boundary (Fig. 4.23, 3rd and 4th rows, yellow boxes), which is suppressed in our method. The results show that our method takes close to or less than 30 ms per frame which means its speed is acceptable for real-time applications.

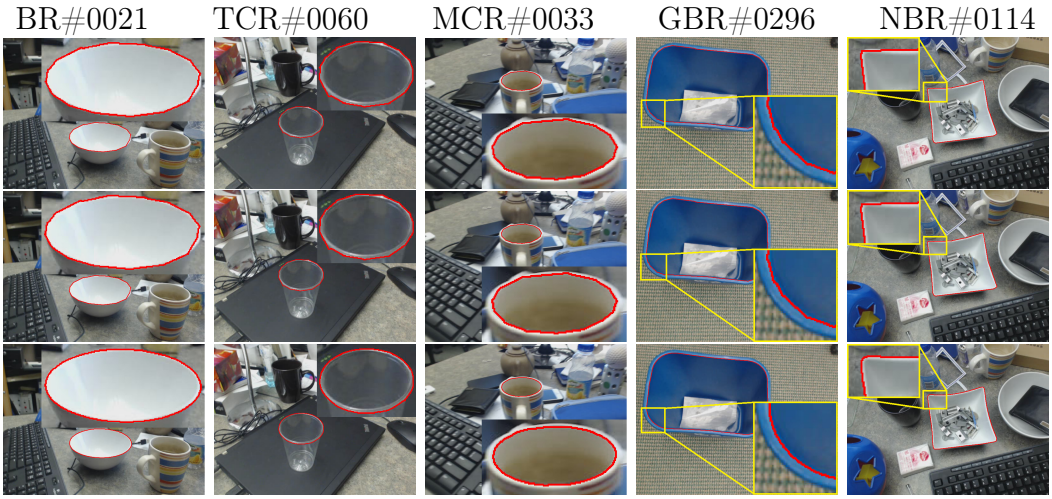


Figure 4.23. Comparison between line segments and edge fragments represented boundaries. From top row to bottom row are results of RRC, BDSP and our EFG respectively.

4.6.2.5 Summary

In this chapter, we proposed a novel real-time edge fragments grouping based method for salient closed boundary tracking. Our tracker encodes the prior shape constraint into the distance difference of deliberately split edge fragments and combines it with the boundary salient measure of relative gap length. It suppresses most of the small erroneous wiggles on the boundaries and improves the tracking accuracy. We validated our method on real-world video sequences and achieve the state-of-the-art results both qualitatively and quantitatively. Currently, complex shape priors and/or shapes with tiny clustered structures (e.g. comb) are hard to track as the grouping process may ignore the tiny structures. Also, the shape prior based algorithm can not handle the large motions in-between frames. To address these problems, we will try to introduce shape templates and more robust shape descriptors in our future works.

Chapter 5

Supervised Salient Object Detection by Deep Convolutional Neural Networks

5.1 Overview

Deep Convolutional Neural Networks have been adopted for salient object detection and achieved state-of-the-art performance. Most of the previous works, however, focus on region accuracy but not on the boundary quality. In this chapter, we propose a predict-refine architecture, BASNet, and a new hybrid loss for Boundary-Aware Salient object detection. Specifically, the architecture is composed of a densely supervised Encoder-Decoder network and a residual refinement module, which are respectively in charge of saliency prediction and saliency map refinement. The hybrid loss guides the network to learn the transformation between the input image and the ground truth in a three-level hierarchy – pixel-, patch- and map- level – by fusing Binary Cross Entropy (BCE), Structural SIMilarity (SSIM) and Intersection-over-Union (IoU) losses. Equipped with the hybrid loss, the proposed predict-refine architecture is able to effectively segment the salient object regions and accurately predict the fine structures with clear boundaries. Experimental results on six public datasets show that our method outperforms the state-of-the-art methods both in terms of regional and boundary evaluation measures. Our method runs at over 25 fps on a single GPU. The code is available [1].

[1] `https://github.com/NathanUA/BASNet`

Besides, we design a simple yet powerful deep network architecture, U^2-Net, for salient object detection (SOD). Compared with U-Net, the architecture of our U^2-Net is a two-level nested U-structure. The proposed design has the following advantages: (1) it is able to capture more contextual information from different scales thanks to the mixture of receptive fields of different sizes in our proposed ReSidual U-blocks (RSU); (2) it increases the depth of the whole architecture without significantly increasing the computation and memory costs because of the pooling operations used in these RSU blocks. Using this novel architecture, we show that we can train a very deep network from scratch that achieves competitive performance on six public SOD datasets. Our U^2-Net runs at 30 FPS on a GTX 1080 Ti GPU. Additionally, to facilitate the usage of our U^2-Net, we provide a small variant U^2-Net† (40 FPS), which is only 4.7 MB. Its competitive performance further demonstrates the flexibility and capability of our U^2-Net architecture.

5.2 Datasets

There are many public salient object detection datasets. We use the most frequently used six benchmark datasets: **DUTS** [251], **DUT-OMRON** [279], **HKU-IS** [145], **ECSSD** [277], **PASCAL-S** [149] and **SOD** [183], to evaluate our models.

- **DUTS** is currently the largest saliency detection dataset. It is comprised of two subsets: DUTS-TR and DUTS-TE. **DUTS-TR** contains 10553 images designed for training and **DUTS-TE** has 5019 images for testing.
- **DUT-OMRON** has 5168 images with one or two objects in each images. Most of the foreground objects are structurally complex.
- **HKU-IS** contains 4447 images. Most of them have more than one connected or disconnected foreground objects.
- **ECSSD** contains 1000 semantically meaningful but structurally complex images.

- **PASCAL-S** consists of 850 images with cluttered backgrounds and complex foreground objects.
- **SOD** contains 300 images which are originally designed for image segmentation. These images are very challenging since most of them contain multiple salient objects either with low contrast or overlapping with the image boundary.

5.3 Evaluation Metrics

The outputs of the deep salient object methods are usually probability maps that have the same spatial resolution with the input images. The value of each pixel in the predicted saliency map is in the range of 0 and 1 (or [0,255]). The ground truth are usually binary masks, in which each pixel is either 0 or 1 (or 0 and 255) where 0 indicates the background pixel and 1 indicates the foreground salient object pixel.

To comprehensively evaluate the quality of those probability maps against the ground truth, seven measures including (1) Precision-Recall (PR) curves, (2) maximal F-measure ($maxF_\beta$), (3) F-measure curves, (4) Mean Absolute Error (MAE), (5) weighted F-measure (F_β^w) [175], (6) structure measure (S_m) [71] and (7) relaxed F-measure of boundary ($relaxF_\beta^b$) are used:

(1) PR curve is a standard way of evaluating the predicted saliency probability maps. The precision and recall of a saliency map are computed by comparing the binarized saliency map against the ground truth mask. Each binarizing threshold results in a pair of average precision and recall over all saliency maps in a dataset. Varying the threshold from 0 to 1 produces a sequence of precision-recall pairs, which is plotted as the PR curve.

(2) F-measure: to have a comprehensive measure on both precision and recall, F_β is computed based on each pair of precision and recall as:

$$F_\beta = \frac{(1+\beta^2)\times Precision\times Recall}{\beta^2\times Precision+Recall} \tag{5.1}$$

where β^2 is set to 0.3 to weight precision more than recall [2]. The maximum F_β ($\boldsymbol{max}F_\beta$) of each dataset is reported in this thesis.

(3) F-measure curve is mainly used to evaluate the robustness of the saliency model against different binary thresholds. F-measure curve is plotted based on data points consists of a set of binary thresholds and their corresponding F-measures F_β.

(4) *MAE* [197] denotes the average absolute per-pixel difference between a predicted saliency map and its ground truth mask. Given a saliency map, its

MAE is defined as:

$$MAE = \frac{1}{H \times W} \sum_{r=1}^{H} \sum_{c=1}^{W} |P(r,c) - G(r,c)| \tag{5.2}$$

where P and G are saliency probability map and its ground truth respectively, H and W represents the height and width of the saliency map and (r, c) denotes the pixel coordinates. For a dataset, its MAE is the average MAE of all the saliency maps.

(5) weighted F-measure (F_β^w) [175] is utilized as a complementary measure to $maxF_\beta$ for overcoming the possible unfair comparison caused by interpolation flaw, dependency flaw and equal-importance flaw [157]. It is defined as:

$$F_\beta^w = (1 + \beta^2) \frac{Precision^w \cdot Recall^w}{\beta^2 \cdot Precision^w + Recall^w}. \tag{5.3}$$

(6) S-measure (S_m) is used to evaluate the structure similarity of the predicted non-binary saliency map and the ground truth. The S-measure is defined as the weighted sum of region-aware S_r and object-aware S_o structural similarity [71]:

$$S = (1 - \alpha) * S_r + \alpha * S_o. \tag{5.4}$$

Given a saliency probability map and its ground truth mask, its region-aware structural similarity S_r is computed as follows: Divide the saliency probability map (P) and the ground truth mask G into K (4, top-left, top-right, bottom-left, bottom-right) subregions based on the centroid of the foreground regions. Each subregion of the saliency map and the ground truth mask can be represented as $x = \{x_i | i = 1, 2, ..., N\}$ and the ground truth $y = \{y_i | i = 1, 2, ..., N\}$ respectively. The region-aware similarity between P and G is then computed as:

$$S_r = \sum_{k=1}^{K} w_k * ssim(k). \tag{5.5}$$

The Structural SIMilarity (SSIM) [261] is defined as:

$$ssim = \frac{2\overline{x}\overline{y}}{\overline{x}^2 + \overline{y}^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \cdot \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{5.6}$$

where $\overline{x}$, $\overline{y}$, σ_x and σ_y are the mean and standard deviations of x and y. σ_{xy} is the covariance of x and y. As we can see, the SSIM computes the similarities

between the ground truth masks and the predicted maps based on their patch-wise mean, standard deviations and covariances, which contain relatively more reliable statistical information of the local structures. Hence, this measure is able to evaluate the structure qualities, which is closer to the human visual evaluation mechanism than pixel-wise errors, of the predicted saliency maps. The object-aware structural similarity is defined as:

$$S_o = \mu * O_{FG} + (1 - \mu) * O_{BG} \tag{5.7}$$

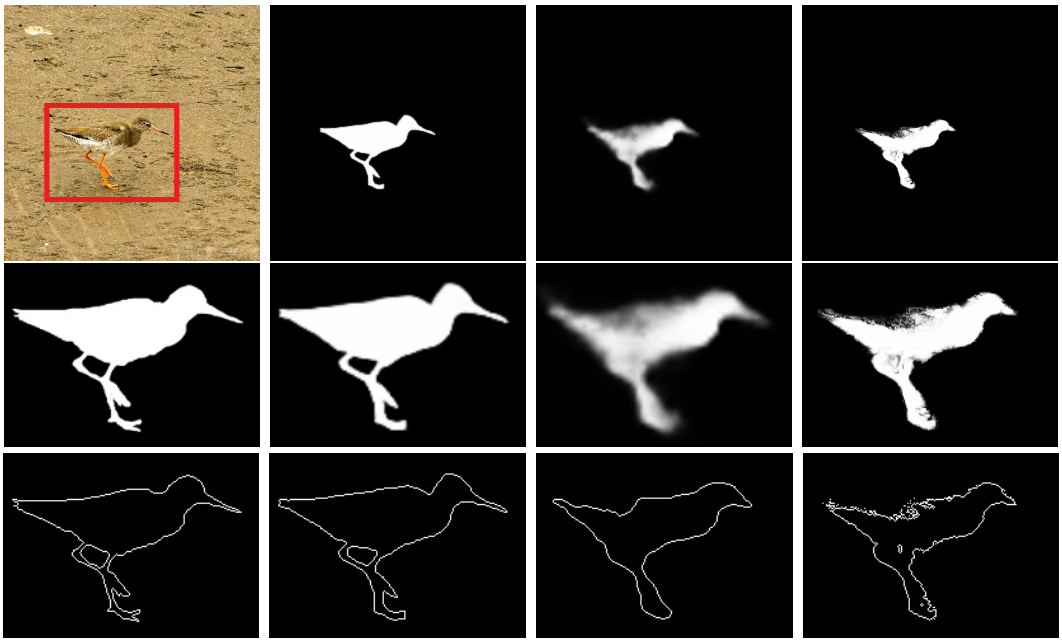
where O_{FG} and O_{BG} are foreground and background comparison respectively:

$$O_{FG} = \frac{2\overline{x}_{FG}}{(\overline{x}_{FG})^2 + 1 + 2\lambda * \sigma_{x_{FG}}}, \tag{5.8}$$

$$O_{BG} = \frac{2\overline{x}_{BG}}{(\overline{x}_{BG})^2 + 1 + 2\lambda * \sigma_{x_{BG}}}. \tag{5.9}$$

$\overline{x}_{FG}$, $\overline{x}_{BG}$, $\sigma_{x_{FG}}$ and $\sigma_{x_{BG}}$ are the mean and standard deviation of the saliency probability map's foreground region and background region.

(7) relax boundary F-measure $relaxF_\beta^b$ [63] is introduced to quantitatively evaluate boundaries of the predicted saliency maps. Given a saliency probability map P, we first convert it to a binary mask P_{bw} using a threshold of 0.5. Then, we obtain the mask of its one pixel wide boundary by conducting an $XOR(P_{bw}, P_{erd})$ operation where P_{erd} is the eroded binary mask [88] of P_{bw}. The same method is used to get the boundaries of ground truth mask. The relaxed boundary precision ($relaxPrecision^b$) is then defined as the fraction of predicted boundary pixels within a range of ρ pixels from ground truth boundary pixels. The relaxed boundary recall ($relaxRecall^b$) measures the fraction of ground truth boundary pixels that are within ρ pixels of predicted boundary pixels. The slack parameter ρ is set to 3 which is similar to the previous studies [181], [219], [304]. The relaxed boundary F-measure $relaxF_\beta^b$ of each predicted saliency map is computed using equation (5.1), in which $Precision$ and $Recall$ are replaced by $relaxPrecision^b$ and $relaxRecall^b$. For each dataset, we report its average $relaxF_\beta^b$ of all predicted saliency maps.



(a) im/GT (b) Our (c) PiCANet (d) PiCANetR

Figure 5.1. Sample result of our method (BASNet) compared to PiCANetR [157]. Column (a) shows the input image, zoom-in view of ground truth (GT) and the boundary map, respectively. (b), (c) and (d) are results of ours, PiCANetR and PiCANetRC (PiCANetR with CRF [126] post-processing). For each method, the three rows respectively show the predicted saliency map, the zoom-in view of saliency map and the zoom-in view of boundary map.

5.4 BASNet: Boundary-Aware Salient Object Detection

5.4.1 Motivation

Recently, Convolutional Neural Networks (CNNs), especially Fully Convolutional Neural Networks (FCN) [252], have been adopted for salient object detection. They take salient object detection problem as a binary image segmentation problem. Although these methods achieve significant results compared to traditional methods, their predicted saliency maps are still defective in details (see Figs. 5.1 (c)-5.1 (d)), reflected in either poor representation of fine structures and/or inaccurate (blurred) boundaries. defective in fine structures and/or boundaries (see Figs. 5.1 (c) -5.1 (d)). Therefore, one of our goal is to achieve accurate salient object detection results.

There are two main challenges in accurate salient object detection:

(1) the saliency is mainly defined over the global contrast of the whole image rather than local or pixel-wise features. To achieve accurate results, the developed saliency detection methods have to understand the global meaning of the whole image as well as the detailed structures of the objects [58]. To address this problem, networks that aggregate multi-level deep features are needed;

(2) Most of the salient object detection methods use Cross Entropy (CE) as their training loss. But models trained with CE loss usually have low confidence in differentiating boundary pixels, leading to blurry boundaries. Other losses such as Intersection over Union (IoU) loss [176], [187], [205], F-measure loss [306] and Dice-score loss [75] were proposed for biased training sets but they are not specifically designed for capturing fine structures. Although Conditional Random Field (CRF) [126] is able to improve the overall performance in terms of region-based measures like IoU and F measure, it is incapable of producing clean and sharp boundaries when the foreground and background pixels are similar (see Fig. 5.1 (d)).

To address the above challenges, we propose a novel Boundary-Aware network, namely **BASNet**, for Salient object detection, which achieves accurate salient object segmentation with high quality boundaries (see Fig. 5.1 (b)):

(1) To capture both global (coarse) and local (fine) contexts, a new predict-refine network is proposed. It assembles a U-Net-like [211] deeply supervised [138], [275] Encoder-Decoder network with a novel residual refinement module. The Encoder-Decoder network transfers the input image to a probability map, while the refinement module refines the predicted map by learning the residuals between the coarse saliency map and ground truth (see Fig. 5.2). In contrast to [58], [103], [195], which use refinement modules iteratively on saliency predictions or intermediate feature maps at multiple scales, our module is used only once on the original scale for saliency prediction.

(2) To obtain high confidence saliency map and clear boundary, we propose a hybrid loss that combines Binary Cross Entropy (BCE) [22], Structural SIMilarity (SSIM) [261] and IoU losses [176], which are expected to learn from ground truth information in pixel-, patch- and map-level, respectively. Rather than using explicit boundary losses (NLDF+ [168], C2S [148]), we implicitly inject the goal of accurate boundary prediction in the hybrid loss, contemplating that it may help reduce spurious error from cross propagating the information learned on the boundary and the other regions on the image.

5.4.2 Contributions

The main contributions of this work are threefold:

- A novel boundary-aware salient object detection network: BASNet, which consists of a deeply supervised encoder-decoder and a residual refinement module,
- A novel hybrid loss that fuses BCE, SSIM and IoU to supervise the training process of accurate salient object prediction on three levels: pixel-level, patch-level and map-level,
- A thorough evaluation of the proposed method that includes comparison with state-of-the-art methods on six widely used public datasets. Our method achieves state-of-the-art results in terms of both regional and boundary evaluation measures.

5.4.3 Architecture of BASNet

(1) Network Architecture

Our proposed BASNet consists of two modules as shown in Fig. 5.2. The prediction module is a U-Net-like densely supervised Encoder-Decoder network [211], which learns to predict saliency map from input images. The multi-scale Residual Refinement Module (RRM) refines the resulting saliency map of the prediction module by learning the residuals between the saliency map and the ground truth.

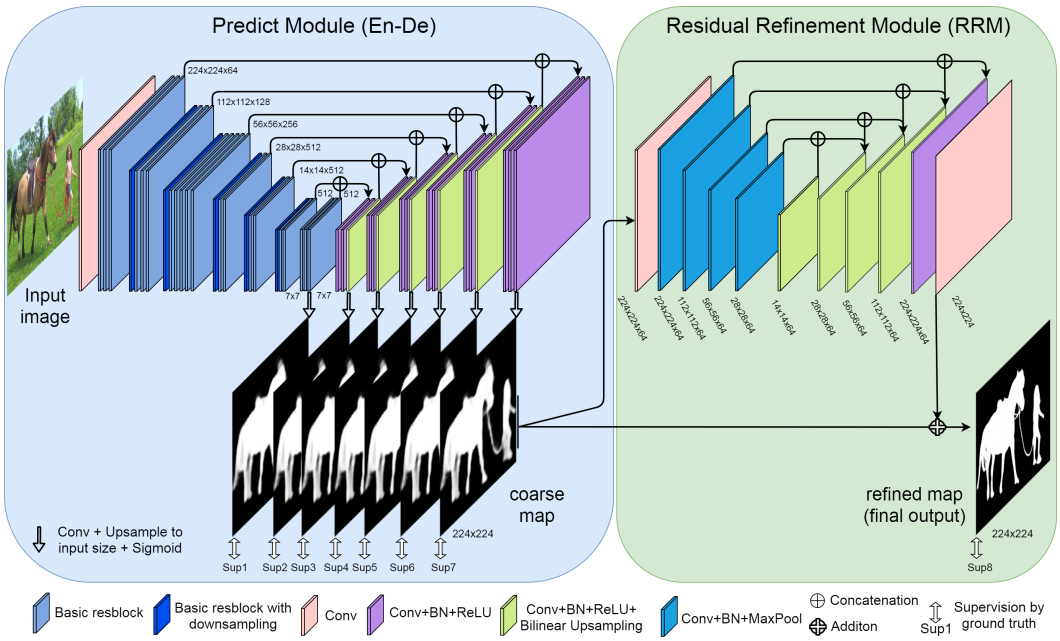


Figure 5.2. Architecture of our proposed boundary-aware salient object detection network: BASNet.

(2) Predict Module

Inspired by U-Net [211] and SegNet [13], we design our salient object prediction module as an Encoder-Decoder network because this kind of architectures is able to capture high level global contexts and low level details at the same time. To reduce over fitting, the last layer of each decoder stage is supervised by the ground truth inspired by HED [275] (see Fig. 5.2). The encoder part has an input convolution layer and six stages comprised of basic res-blocks. The input convolution layer and the first four stages are adopted from ResNet-34 [94]. The difference is that our input layer has 64 convolution filters with size of 3×3 and stride of 1 rather than size of 7×7 and stride of 2. Additionally, there is no pooling operation after the input layer. That means the feature maps before the second stage have the same spatial resolution as the input image. This is different from the original ResNet-34, which has quarter scale resolution in the first feature map. This adaptation enables the network to obtain higher resolution feature maps in earlier layers, while it also decreases the overall receptive fields. To achieve the same recep-

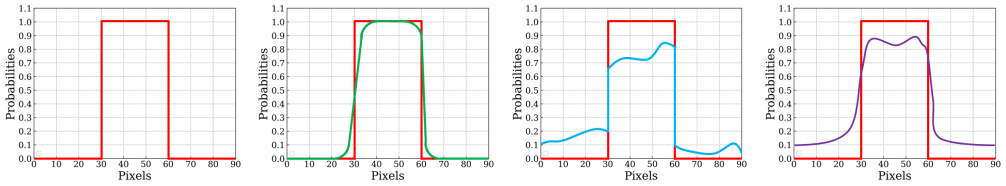


Figure 5.3. Illustration of different aspects of coarse prediction in one-dimension. (a) Red: probability plot of ground truth - GT, (b) Green: probability plot of coarse boundary not aligning with GT, (c) Blue: coarse region having too low probability, (d) Purple: real coarse predictions usually have both problems.

tive field as ResNet-34 [94], we add two more stages after the fourth stage of ResNet-34. Both stages consist of three basic res-blocks with 512 filters after a non-overlapping max pooling layer of size 2.

To further capture global information, we add a bridge stage between the encoder and the decoder. It consists of three convolution layers with 512 dilated (dilation=2) [286] 3×3 filters. Each of these convolution layers is followed by a batch normalization [102] and a ReLU activation function [85].

Our decoder is almost symmetrical to the encoder. Each stage consists of three convolution layers followed by a batch normalization and a ReLU activation function. The input of each stage is the concatenated feature maps of the upsampled output from its previous stage and its corresponding stage in the encoder. To achieve the side-output saliency maps, the multi-channel output of the bridge stage and each decoder stage is fed to a plain 3×3 convolution layer followed by a bilinear upsampling and a sigmoid function. Therefore, given a input image, our predict module produces seven saliency maps in the training process. Although every saliency map is upsampled to the same size with the input image, the last one has the highest accuracy and hence is taken as the final output of the predict module. This output is passed to the refinement module.

(3) Refine Module

Refinement Module (RM) [58], [103] is usually designed as a residual block

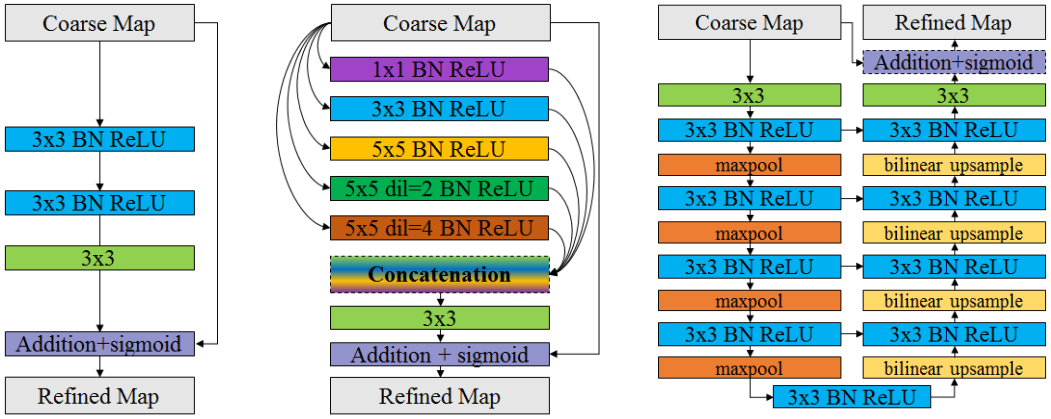
which refines the predicted coarse saliency maps S_{coarse} by learning the residuals $S_{residual}$ between the saliency maps and the ground truth as

$$S_{refined} = S_{coarse} + S_{residual}. \tag{5.10}$$

Before introducing our refinement module, we have to define the term "coarse". Here, "coarse" includes two aspects. One is the blurry and noisy boundaries (see its one-dimension (1D) illustration in Fig. 5.3b). The other one is the unevenly predicted regional probabilities (see Fig. 5.3c). The real predicted coarse saliency maps usually contain both coarse cases (see Fig. 5.3d).

Residual refinement module based on local context (RRM_LC), Fig. 5.4a, was originally proposed for boundary refinement [195]. Since its receptive field is small, Islam *et al.* [103] and Deng *et al.* [58] iteratively or recurrently use it for refining saliency maps on different scales. Wang *et al.* [255] adopted the pyramid pooling module from [93], in which three-scale pyramid pooling features are concatenated. To avoid losing details caused by pooling operations, RRM_MS (Fig. 5.4b) uses convolutions with different kernel sizes and dilations [286], [297] to captures multi-scale contexts. However, these modules are shallow thus hard to capture high level information for refinement.

To refine both region and boundary drawbacks in coarse saliency maps, we develop a novel residual refinement module. Our RRM employs the residual encoder-decoder architecture, RRM_Ours (see Figs. 5.2 and 5.4c). Its main architecture is similar but simpler to our predict module. It contains an input layer, an encoder, a bridge, a decoder and an output layer. Different from the predict module, both encoder and decoder have four stages. Each stage only has one convolution layer. Each layer has 64 filters of size 3×3 followed by a batch normalization and a ReLU activation function. The bridge stage also has a convolution layer with 64 filters of size 3×3 followed by a batch normalization and ReLU activation. Non-overlapping max pooling is used for downsampling in the encoder and bilinear interpolation is utilized for the upsampling in the decoder. The output of this RM module is the final resulting saliency map of our model.



(a) RRM_LC (b) RRM_MS (c) RRM_Ours

Figure 5.4. Illustration of different Residual Refine Modules (RRM): (a) local boundary refinement module RRM_LC; (b) multi-scale refinement module RRM_MS; (c) our encoder-decoder refinement module RRM_Ours.

5.4.4 Bounary-Aware Hybrid Loss

Our training loss is defined as the summation over all outputs:

$$\mathcal{L} = \sum_{k=1}^{K} \alpha_k \ell^{(k)} \tag{5.11}$$

where $\ell^{(k)}$ is the loss of the k-th side output, K denotes the total number of the outputs and α_k is the weight of each loss. As described in above sections, our salient object detection model is deeply supervised with eight outputs, i.e. $K = 8$, including seven outputs from the prediction model and one output from the refinement module.

To obtain high quality regional segmentation and clear boundaries, we propose to define $\ell^{(k)}$ as a hybrid loss:

$$\ell^{(k)} = \ell_{bce}^{(k)} + \ell_{ssim}^{(k)} + \ell_{iou}^{(k)}. \tag{5.12}$$

where $\ell_{bce}^{(k)}$, $\ell_{ssim}^{(k)}$ and $\ell_{iou}^{(k)}$ denote BCE loss [22], SSIM loss [261] and IoU loss [176], respectively.

BCE [22] loss is the most widely used loss in binary classification and segmentation. It is defined as:

$$\ell_{bce} = -\sum_{(r,c)} [G(r,c)\log(S(r,c)) + (1-G(r,c))\log(1-S(r,c))] \tag{5.13}$$

where $G(r,c) \in \{0,1\}$ is the ground truth label of the pixel (r,c) and $S(r,c)$ is the predicted probability of being salient object.

SSIM is originally proposed for image quality assessment [261]. It captures the structural information in an image. Hence, we integrated it into our training loss to learn the structural information of the salient object ground truth. Let $\mathbf{x} = \{x_j : j = 1, ..., N^2\}$ and $\mathbf{y} = \{y_j : j = 1, ..., N^2\}$ be the pixel values of two corresponding patches (size: $N \times N$) cropped from the predicted probability map S and the binary ground truth mask G respectively, the SSIM of $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\ell_{ssim} = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{5.14}$$

where μ_x, μ_y and σ_x, σ_y are the mean and standard deviations of $\mathbf{x}$ and $\mathbf{y}$ respectively, σ_{xy} is their covariance, $C_1 = 0.01^2$ and $C_2 = 0.03^2$ are used to avoid dividing by zero.

IoU is originally proposed for measuring the similarity of two sets [107] and then used as a standard evaluation measure for object detection and segmentation. Recently, it has been used as the training loss [176], [205]. To ensure its differentiability, we adopted the IoU loss used in [176]:

$$\ell_{iou} = 1 - \frac{\sum_{r=1}^{H}\sum_{c=1}^{W} S(r,c)G(r,c)}{\sum_{r=1}^{H}\sum_{c=1}^{W} [S(r,c)+G(r,c)-S(r,c)G(r,c)]} \tag{5.15}$$

where $G(r,c) \in \{0,1\}$ is the ground truth label of the pixel (r,c) and $S(r,c)$ is the predicted probability of being salient object.

We illustrate the impact of each of the three losses in Fig. 5.5. These heatmaps show change of the loss at each pixel as the training progresses. The three rows correspond to the BCE loss, SSIM loss and IoU loss, respectively. The three columns represent different stages of the training process. BCE loss is pixel-wise. It does not consider the labels of the neighborhood and it weights both the foreground and background pixels equally. It helps with the convergence on all pixels.

SSIM loss is a patch-level measure, which considers a local neighborhood of each pixel. It assigns higher weights to the boundary, i.e., the loss is higher

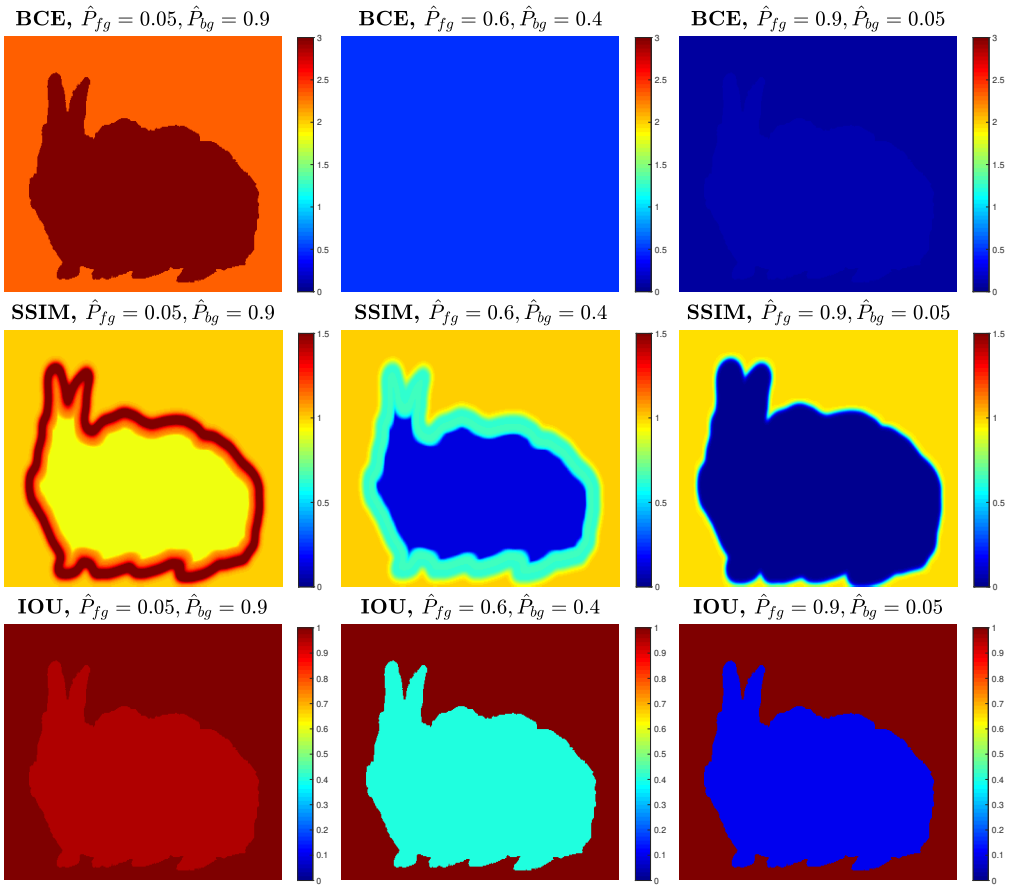


Figure 5.5. Illustration of the impact of the losses. $\hat{P}_{fg}$ and $\hat{P}_{bg}$ denote the predicted probability of the foreground and background, respectively.

around the boundary, even when the predicted probabilities on the boundary and the rest of the foreground are the same. In the beginning of training, the loss along the boundary is the largest (see second row of Fig. 5.5). It helps the optimization to focus on the boundary. As the training progresses, the SSIM loss of the foreground reduces and the background loss becomes the dominant term. However, the background loss does not contribute to the training until when the prediction of background pixel becomes very close to the ground truth, where the loss drops rapidly from one to zero. This is helpful since the prediction typically goes close to zero only late in the training process where BCE loss becomes flat. The SSIM loss ensures that there's still enough gradient to drive the learning process. The background prediction looks cleaner since the probability is pushed to zero.

IoU is a map-level measure. But we plot the per-pixel IoU following Eq. (5.15) for illustration purpose. As the confidence of the network prediction of the foreground grows, the loss of the foreground reduces eventually to zero. When combining these three losses, we utilize BCE to maintain a smooth gradient for all pixels, while using IoU to give more focus on the foreground. SSIM is used to encourage that the prediction respects the structure of the original image, by a larger loss near the boundary.

5.4.5 Implementation and Experimental Setup

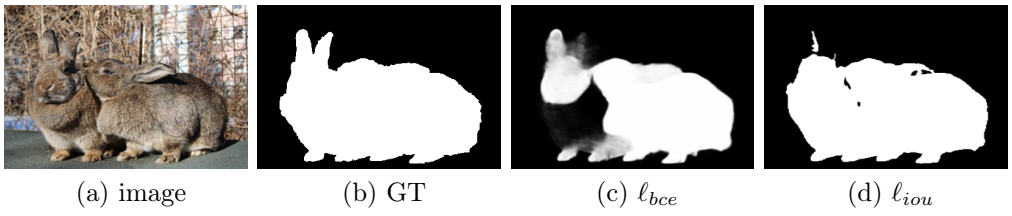
We evaluated our method on six frequently used benchmark datasets as described above: DUT-OMRON [279], DUTS-TE [251], HKU-IS [145], ECSSD [277], PASCAL-S [149], SOD [183].

We train our network using the DUTS-TR dataset, which has 10553 images. Before training, the dataset is augmented by horizontal flipping to 21106 images. During training, each image is first resized to 256×256 and randomly cropped to 224×224. Part of the encoder parameters are initialized from the ResNet-34 model [94]. Other convolutional layers are initialized by Xavier [81]. We utilize the Adam optimizer [120] to train our network and its hyper parameters are set to the default values, where the initial learning rate lr=1e-3, betas=(0.9, 0.999), eps=1e-8, weight_decay=0. We train the network until the loss converges without using validation set. The training loss converges after 400k iterations with a batch size of 8 and the whole training process takes about 125 hours. During testing, the input image is resized to 256×256 and fed into the network to obtain its saliency map. Then, the saliency map (256×256) is resized back to the original size of the input image. Both the resizing processes use bilinear interpolation.

We implement our network based on the publicly available framework: Pytorch 0.4.0 [194]. An eight-core PC with an AMD Ryzen 1800x 3.5 GHz CPU (with 32GB RAM) and a GTX 1080ti GPU (with 11GB memory) is used for both training and testing. The inference for a 256×256 image only takes 0.040s (**25 fps**).

Table 5.1. Ablation study on different architectures and losses: En-De: Encoder-Decoder, Sup: side output supervision; $\ell_{bi} = \ell_{bce} + \ell_{iou}$, $\ell_{bs} = \ell_{bce} + \ell_{ssim}$, $\ell_{bsi} = \ell_{bce} + \ell_{ssim} + \ell_{iou}$.

Ablation	Configurations	$maxF_\beta$	$relaxF_\beta^b$	MAE
Architecture	Baseline U-Net [211] + ℓ_{bce}	0.896	0.669	0.066
	En-De + ℓ_{bce}	0.929	0.767	0.047
	En-De+Sup + ℓ_{bce}	0.934	0.805	**0.040**
	En-De+Sup+RRM_LC + ℓ_{bce}	0.936	0.803	**0.040**
	En-De+Sup+RRM_MS + ℓ_{bce}	0.935	0.804	0.042
	En-De+Sup+RRM_Ours + ℓ_{bce}	**0.937**	**0.806**	0.042
Loss	En-De+Sup+RRM_Ours + ℓ_{ssim}	0.924	0.808	0.042
	En-De+Sup+RRM_Ours + ℓ_{iou}	0.933	0.795	0.039
	En-De+Sup+RRM_Ours + ℓ_{bs}	0.940	0.815	0.040
	En-De+Sup+RRM_Ours + ℓ_{bi}	0.940	0.813	0.038
	En-De+Sup+RRM_Ours + ℓ_{bsi}	**0.942**	**0.826**	**0.037**



(a) image (b) GT (c) ℓ_{bce} (d) ℓ_{iou}

(e) ℓ_{ssim} (f) ℓ_{bi} (g) ℓ_{bs} (h) ℓ_{bsi}

Figure 5.6. Sample results trained with our BASNet on different losses.

5.4.6 Ablation Study

In this section, we validate the effectiveness of each key components used in our model. The ablation study contains two parts: architecture ablation and loss ablation. The ablation experiments are conducted on the ECSSD dataset.

Architecture ablation: To prove the effectiveness of our BASNet, we report the quantitative comparison results of our model against other related architectures. We take U-Net [211] as our baseline network. Then we start with our proposed Encoder-Decoder network and progressively extend it with densely side output supervision and different residual refinement modules in-

cluding RRM_LC, RRM_MS and RRM_Ours. Table 5.3 illustrates the results of this ablation study. As we can see, our BASNet architecture achieves the best performance among these configurations.

Loss ablation: To demonstrate the effectiveness of our proposed fusion loss, we conduct a set of experiments over different losses based on our BASNet architecture. The BCE loss ℓ_{bce}, SSIM loss ℓ_{ssim}, IoU loss ℓ_{iou} and their combinations including BCE-SSIM (BS) loss $\ell_{bs} = \ell_{bce} + \ell_{ssim}$, BCE-IoU (BI) loss $\ell_{bi} = \ell_{bce} + \ell_{iou}$ are compared with our BCE-SSIM-IoU (BSI) loss $\ell_{bsi} = \ell_{bce} + \ell_{ssim} + \ell_{iou}$. The results in Table 5.3 signifies that our proposed hybrid ℓ_{bsi} loss greatly improves the performance, especially for the boundary quality. To further illustrate the qualitative effect of losses, results of our BASNet trained with different losses are shown in Fig. 5.6. It is clear that the proposed hybrid loss achieves superior qualitative results.

5.5 U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection

5.5.1 Motivation

Recently, with the development of deep convolutional neural networks (CNNs), especially the rise of Fully Convolutional Networks (FCN) [163] in image segmentation, the salient object detection has been improved significantly. It is natural to ask, what is still missing? Let's take a step back and look at the remaining challenges.

There is a common pattern in the design of most SOD networks [58], [147], [168], [256], that is, they focus on making good use of deep features extracted by existing backbones, such as Alexnet[128], VGG [228], ResNet [94], ResNeXt [274], etc. However, these backbones are all originally designed for image classification. They extract features that are representative of semantic meaning rather than local details and global contrast information, which are essential to saliency detection. And they need to be pre-trained on ImageNet [57] data which is data-inefficient especially if the target data follows a different distribution than ImageNet.

This leads to our first question: **can we design a new network specifically for SOD, that allows training from scratch and achieves comparable or better performance than those based on existing backbones?**

There are a few more issues on the network architectures for SOD. First, they are often overly complicated [311]. It is partially due to the additional feature aggregation modules that are added to the existing backbones to extract multi-level saliency features from these backbones. Secondly, the existing backbones usually achieve deeper architecture by sacrificing high resolution of feature maps [311]. To run these deep models with affordable memory and computational cost, the feature maps are down scaled to lower resolution at early stages. For instance, at the early layers of both ResNet and DenseNet [100], a convolution with stride of two followed by a maxpooling with stride of two are utilized to reduce the size of the feature maps to one fourth of the input

maps. However, high resolution also plays an important role in segmentation besides the deep architecture [152].

Hence, our follow-up question is: **can we go deeper while maintaining high resolution feature maps, at a low memory and computation cost?**

5.5.2 Contributions

Our main contribution is a novel and simple network architecture, **U^2-Net**, that addresses the two questions above. First, U^2-Net is a two-level nested U-structure that is specifically designed for SOD without using any pre-trained backbones from image classification. It can be trained from scratch to achieve competitive performance. Second, the novel architecture allows the network to go deeper, attain high resolution, without significantly increasing the memory and computation cost. This is achieved by a nested U-structure: on the bottom level, we design a novel ReSidual U-block (RSU), which is able to extract intra-stage multi-scale features without degrading the feature map resolution; on the top level, there is a U-Net like structure, in which each stage is filled by a RSU block. The two-level configuration results in a nested U-structure (see Fig. 5.11). Our U^2-Net (176.3 MB) achieves competitive performance against the state-of-the-art (SOTA) methods on six public datasets, and runs at real-time (30 FPS, with input size of 320×320×3) on a 1080ti GPU. To facilitate the usage of our design in computation and memory constrained environments, we provide a small version of our U^2-Net, called **U^2-Net†** (4.7 MB). The U^2-Net† achieves competitive results against most of the SOTA models (Fig. 5.7) at 40 FPS.

5.5.3 Residual U-blocks

Both local and global contextual information are very important for salient object detection and other segmentation tasks. In modern CNN designs, such as VGG, ResNet, DenseNet and so on, small convolutional filters with size of 1×1 or 3×3 are the most frequently used components for feature extraction. They are in favor since they require less storage space and are computationally

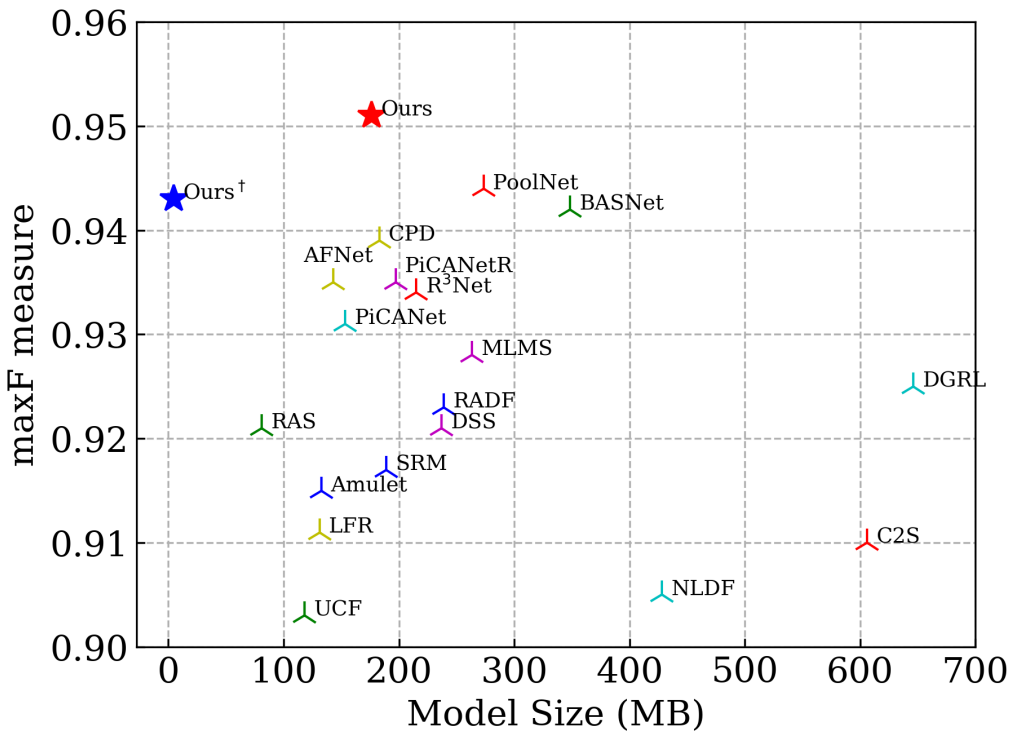


Figure 5.7. Comparison of model size and performance of our U^2-Net with other SOTA salient object detection models. The $maxF_\beta$ measure is computed on dataset ECSSD. The red star denotes our U^2-Net (Ours) (176.3 MB) and the blue star indicates our small version of U^2-Net† (Ours†) (4.7 MB).

efficient. Figures 5.8(a)-(c) illustrates typical existing convolution blocks with small receptive fields. The output feature maps of shallow layers only contain local features because the receptive field of 1×1 or 3×3 filters are too small to capture global information. To achieve more global information at high resolution feature maps from shallow layers, the most direct idea is to enlarge the receptive field. Fig. 5.8 (d) shows an inception like block [237], [297], which tries to extract both local and non-local features by enlarging the receptive fields using dilated convolutions [42]. However, conducting multiple dilated convolutions on the input feature map (especially in the early stage) with original resolution requires too much computation and memory resources. To decrease the computation costs, pyramid pooling [305] and PoolNet [154] adapt this parallel configuration by replacing the dilated convolutions with normal convolutions conducted on downsampled feature maps. But fusion of different

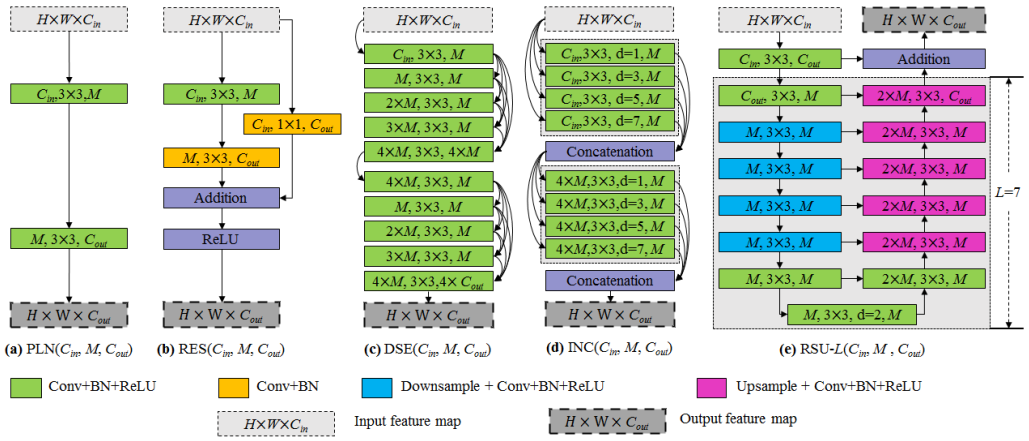


Figure 5.8. Illustration of existing convolution blocks and our proposed RSU block: (a) Plain convolution block PLN; (b) Residual-like block RES; (c) Inception-like block INC; (d) Dense-like block DSE; (e) Our residual U-block RSU.

scale features by direct upsampling and concatenation (or addition) may lead to degradation of high resolution features [311].

Inspired by U-Net [211], we propose a novel **Re**S**i**dual **U**-block, **RSU**, to capture intra-stage multi-scale features. The structure of RSU-$L(C_{in}, M, C_{out})$ is shown in Fig. 5.8(e), where L is the number of layers in the encoder, C_{in}, C_{out} denote input and output channels, and M denotes the number of channels in the internal layers of RSU. Hence, our RSU mainly consists of three components:

(1) an input convolution layer, which transforms the input feature map $\mathbf{x}$ $(H \times W \times C_{in})$ to an intermediate map $\mathcal{F}_1(\mathbf{x})$ with channel of C_{out}. This is a plain convolutional layer for local feature extraction.

(2) a U-Net like symmetric encoder-decoder structure with height of L which takes the intermediate feature map $\mathcal{F}_1(\mathbf{x})$ as input and learns to extract and encode the multi-scale contextual information $\mathcal{U}(\mathcal{F}_1(\mathbf{x}))$. $\mathcal{U}$ represents the U-Net like structure as shown in Fig. 5.8(e). Larger L leads to deeper RSU, more pooling operations, larger range of receptive fields and richer local and global features. Configuring this parameter enables extraction of multi-scale features from input feature maps with

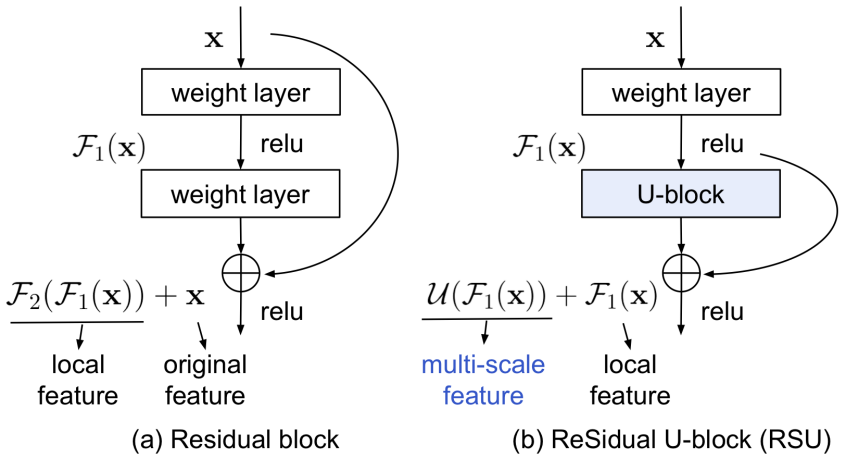


Figure 5.9. Comparison of the residual block and our RSU.

arbitrary spatial resolutions. The multi-scale features are extracted from gradually downsampled feature maps and encoded into high resolution feature maps by progressive upsampling, concatenation and convolution. This process mitigates the loss of fine details caused by direct upsampling with large scales.

(3) a residual connection which fuses local features and the multi-scale features by the summation: $\mathcal{F}_1(\mathbf{x}) + \mathcal{U}(\mathcal{F}_1(\mathbf{x}))$.

To better illustrate the intuition behind our design, we compare our RSU with the original residual block [94] in Fig. 5.9. The operation in the residual block can be summarized as $\mathcal{H}(\mathbf{x}) = \mathcal{F}_2(\mathcal{F}_1(\mathbf{x})) + \mathbf{x}$, where $\mathcal{H}(x)$ denotes the desired mapping of the input features $\mathbf{x}$; $\mathcal{F}_2, \mathcal{F}_1$ stand for the weight layers, which are convolution operations in this setting. The main design difference between RSU and residual block is that RSU replaces the plain, single-stream convolution with a U-Net like structure, and replace the original feature with the local feature transformed by a weight layer: $\mathcal{H}_{RSU}(\mathbf{x}) = \mathcal{U}(\mathcal{F}_1(\mathbf{x})) + \mathcal{F}_1(\mathbf{x})$, where $\mathcal{U}$ represents the multi-layer U-structure illustrated in Fig. 5.8(e). This design change empowers the network to extract features from multiple scales directly from each residual block. More notably, the computation overhead due to the U-structure is small, since most operations are applied on the downsampled feature maps. This is illustrated in Fig. 5.10, where we show

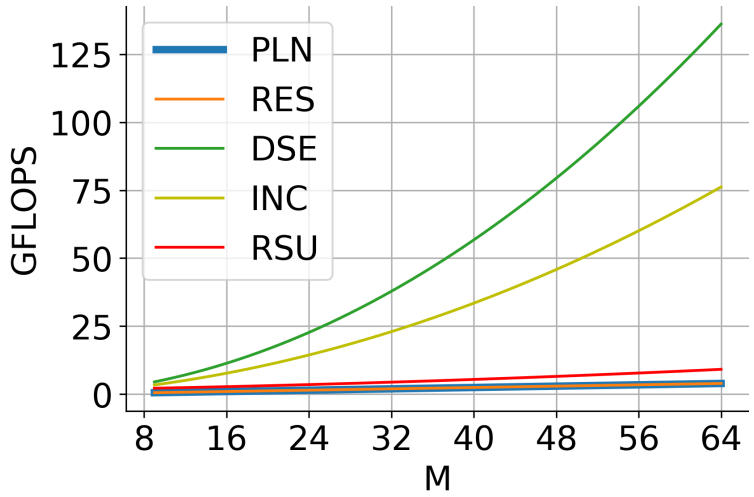


Figure 5.10. Computation costs (GFLOPS Giga Floating Point Operations) of different blocks shown in Fig. 5.8: the computation costs are calculated based on transferring an input feature map with dimension $320 \times 320 \times 3$ to a $320 \times 320 \times 64$ output feature map.

the computation cost comparison between RSU and other feature extraction modules in Fig. 5.8 (a)-(d). The FLOPs of DSE, INC and RSU all grow quadratically with the number of internal channel M. But RSU has a much smaller coefficient on the quadratic term, leading to an improved efficiency. Its computational overhead compared with PLN and RES blocks, which are both linear w.r.t. M, is not significant.

5.5.4 Architecture U^2-Net

Stacking multiple U-Net-like structures for different tasks has been explored for a while, e.g. stacked UNets [188], DocUNet [170], etc. These methods usually stack U-Net-like structures sequentially to build cascaded models and can be summarized as "**(U$\times n$-Net)**", where n is the number of repeated U-Net modules. The issue is that the computation and the memory costs get magnified by n.

In this work, we propose a different formulation, **U^n-Net**, of stacking U-structure for salient object detection. Our exponential notation refers to nested U-structure rather than cascaded stacking. Theoretically, the exponent n can be set as an arbitrary positive integer to achieve single-level or multi-level

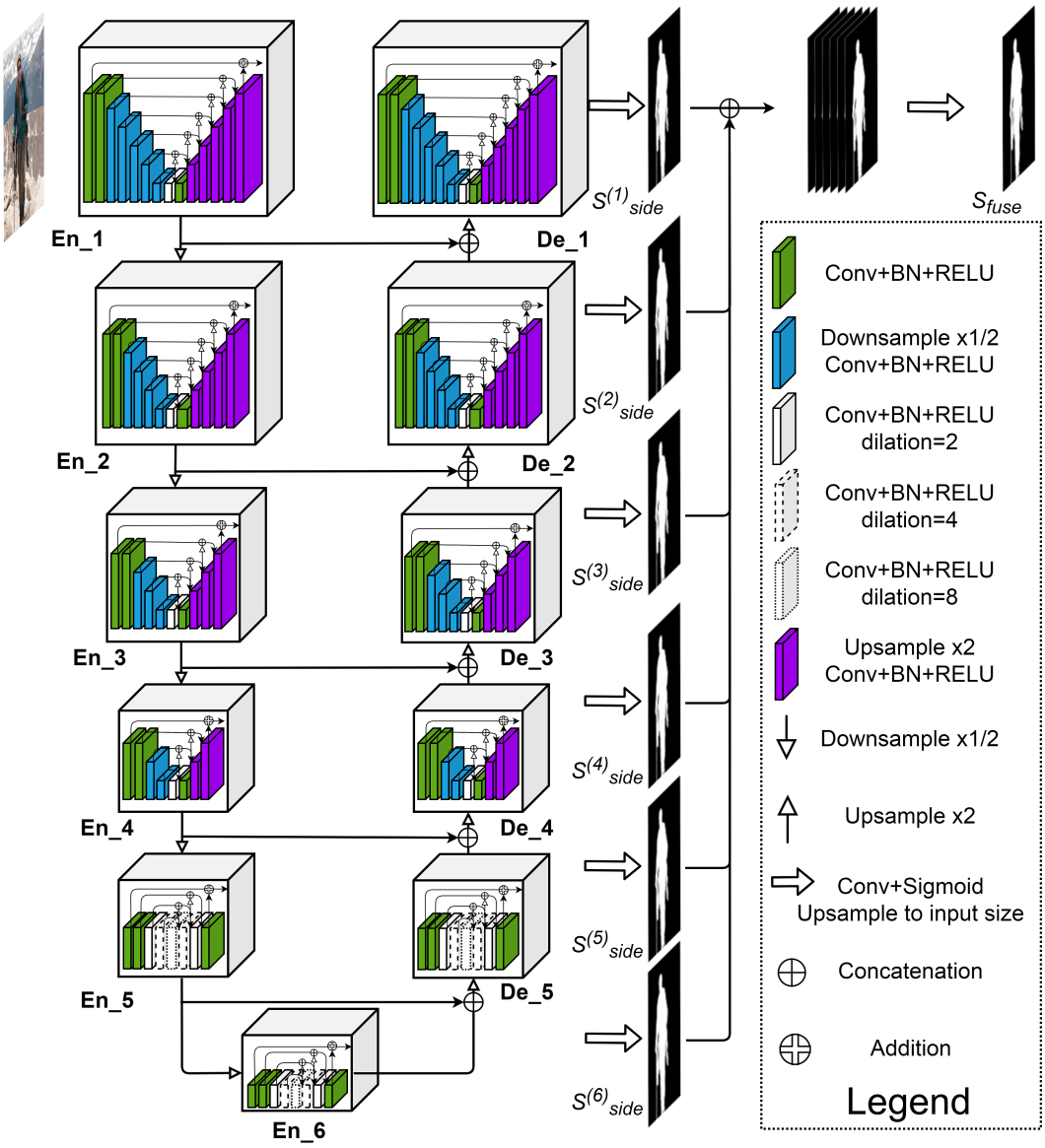


Figure 5.11. Illustration of our proposed U^2-Net architecture. The main architecture is a U-Net like Encoder-Decoder, where each stage consists of our proposed RSU block. For example, **En_1** is based on our RSU block shown in Fig. 5.8(e). Detailed configuration of RSU block of each stage is given in the last two rows of Table 5.2.

Table 5.2. Detailed configurations of different architectures used in ablation study.

Architecture with different blocks	**Stages**										
	En_1	**En_2**	**En_3**	**En_4**	**En_5**	**En_6**	**De_5**	**De_4**	**De_3**	**De_2**	**De_1**
PLN U-Net	PLN (3,64,64)	PLN (64,128,128)	PLN (128,256,256)	PLN (256,512,512)	PLN (512,512,512)	PLN (512,512,512)	PLN (1024,512,512)	PLN (1024,256,256)	PLN (512,128,128)	PLN (256,64,64)	PLN (128,64,64)
RES U-Net	RES (3,64,64)	RES (64,128,128)	RES (128,256,256)	RES (256,512,512)	RES (512,512,512)	RES (512,512,512)	RES (1024,512,512)	RES (1024,256,256)	RES (512,128,128)	RES (256,64,64)	RES (128,64,64)
DSE U-Net	DSE (3,32,64)	DSE (64,32,128)	DSE (128,64,256)	DSE (256,128,512)	DSE (512,128,512)	DSE (512,128,512)	DSE (1024,128,512)	DSE (1024,64,256)	DSE (512,32,128)	DSE (256,16,64)	DSE (128,16,64)
INC U-Net	INC (3,32,64)	INC (64,32,128)	INC (128,64,256)	INC (256,128,512)	INC (512,128,512)	INC (512,128,512)	INC (1024,128,512)	INC (1024,64,256)	INC (512,32,128)	INC (256,16,64)	INC (128,16,64)
U^2-Net (**Ours**)	RSU-7 (3,32,64)	RSU-6 (64,32,128)	RSU-5 (128,64,256)	RSU-4 (256,128,512)	RSU-4F (512,256,512)	RSU-4F (512,256,512)	RSU-4F (1024,256,512)	RSU-4 (1024,128,512)	RSU-5 (512,64,128)	RSU-6 (256,32,64)	RSU-7 (128,16,64)
U^2-Net† (**Ours**†)	RSU-7 (3,16,64)	RSU-6 (64,16,64)	RSU-5 (64,16,64)	RSU-4 (64,16,64)	RSU-4F (64,16,64)	RSU-4F (64,16,64)	RSU-4F (128,16,64)	RSU-4 (128,16,64)	RSU-5 (128,16,64)	RSU-6 (128,16,64)	RSU-7 (128,16,64)

nested U-structure. But architectures with too many nested levels will be too complicated to be implemented and employed in real applications.

Here, we set n as 2 to build our U^2-Net. Our U^2-Net is a two-level nested U-structure shown in Fig. 5.11. Its top level is a big U-structure consists of 11 stages (cubes in Fig. 5.11). Each stage is filled by a well configured RSU block (bottom level U-structure). Hence, the nested U-structure enables the extraction of intra-stage multi-scale features and aggregation of inter-stage multi-level features more efficiently.

As illustrated in Fig.5.11, the U^2-Net mainly consists of three parts: (1) a six stages encoder, (2) a five stages decoder and (3) a saliency map fusion module attached with the decoder stages and the last encoder stage.

- In encoder stages **En_1**, **En_2**, **En_3** and **En_4**, we use RSU-7, RSU-6, RSU-5 and RSU-4, respectively. As mentioned before, "7", "6", "5" and "4" denote the heights (L) of RSU blocks. The L is usually configured according to the spatial resolution of the input feature maps. For feature maps with large height and width, we use greater L to capture more large scale information. The resolution of feature maps in **En_5** and **En_6** are relatively low, further downsampling of these feature maps leads to loss of useful context. Hence, in both **En_5** and **En_6** stages, RSU-4F are used, where "F" means that the RSU is a dilated version, in which we replace the pooling and upsampling operations with dilated convolutions (see Fig. 5.11). That means all of intermediate feature maps of RSU-4F have the same resolution with its input feature maps.

- The decoder stages have similar structures to their symmetrical encoder stages with respect to **En_6**. In **De_5**, we also use the dilated version residual U-block RSU-4F which is similar to that used in the encoder stages **En_5** and **En_6**. Each decoder stage takes the concatenation of the upsampled feature maps from its previous stage and those from its symmetrical encoder stage as the input, see Fig. 5.11.

- The last part is the saliency map fusion module which is used to generate saliency probability maps. Similar to HED [275], our U^2-Net first generates six side output saliency probability maps $\mathcal{S}_{side}^{(6)}$, $\mathcal{S}_{side}^{(5)}$, $\mathcal{S}_{side}^{(4)}$, $\mathcal{S}_{side}^{(3)}$, $\mathcal{S}_{side}^{(2)}$, $\mathcal{S}_{side}^{(1)}$ from stages **En_6**, **De_5**, **De_4**, **De_3**, **De_2** and **De_1** by a 3×3 convolution layer and a sigmoid function. Then, it upsamples these saliency maps to the input image size and fuses them with a concatenation operation followed by a 1×1 convolution layer and a sigmoid function to generate the final saliency probability map $\mathcal{S}_{fuse}$ (see bottom right of Fig. 5.11).

In summary, the design of our U^2-Net allows having deep architecture with rich multi-scale features and relatively low computation and memory costs. In addition, since our U^2-Net architecture is only built upon our RSU blocks without using any pre-trained backbones adapted from image classification, it is flexible and easy to be adapted to different working environments with insignificant performance loss. In this paper, we provide two instances of our U^2-Net by using different configurations of filter numbers: a normal version **U^2-Net** (176.3 MB) and a relatively smaller version **U^2-Net†** (4.7 MB). Detailed configurations are presented in the last two rows of Table 5.2.

5.5.5 Supervision

In the training process, we use deep supervision similar to HED [275]. Its effectiveness has been proven in HED and DSS. Our training loss is defined as:

$$\mathcal{L} = \sum_{m=1}^{M} w_{side}^{(m)} \ell_{side}^{(m)} + w_{fuse} \ell_{fuse} \tag{5.16}$$

where $\ell_{side}^{(m)}$ ($M = 6$, as the Sup1, Sup2, $\cdots$, Sup6 in Fig. 5.2) is the loss of the side output saliency map $S_{side}^{(m)}$ and ℓ_{fuse} (Sup7 in Fig. 5.2) is the loss of the final fusion output saliency map S_{fuse}. $w_{side}^{(m)}$ and w_{fuse} are the weights of each loss term. For each term ℓ, we use the standard binary cross-entropy to calculate the loss:

$$\ell = -\sum_{(r,c)}^{(H,W)} [P_{G(r,c)} log P_{S(r,c)} + (1 - P_{G(r,c)}) log(1 - P_{S(r,c)})] \tag{5.17}$$

where (r, c) is the pixel coordinates and (H, W) is image size: height and width. $P_{G(r,c)}$ and $P_{S(r,c)}$ denote the pixel values of the ground truth and the predicted saliency probability map, respectively. The training process tries to minimize the overall loss $\mathcal{L}$ of Eq. (5.16). It is worth noting that the hybrid loss used in BASNet was also tested with the U^2-Net models. However, the improvement is insignificant most likely because our U^2-Net implicitly captures these multi-scale information and produces accurate results with both high region and boundary quality. In the testing process, we choose the fusion output ℓ_{fuse} as our final saliency map.

5.5.6 Implementation and Experimental Setup

Similar to BASNet, we train our network on **DUTS-TR**, which is a part of DUTS dataset [251]. **DUTS-TR** contains 10553 images in total. Currently, it is the largest and most frequently used training dataset for salient object detection. We augment this dataset by horizontal flipping to obtain 21106 training images offline.

In the training process, each image is first resized to 320×320 and randomly flipped vertically and cropped to 288×288. The sizes are larger than BASNet because larger resolution is able to keep more detail information and the low memory and computation cost enable our U^2-Net to accept input images with larger resolutions. We are not using any existing backbones in our network. Hence, we train our network from scratch and all of our convolutional layers are initialized by Xavier [81]. The loss weights $w_{side}^{(m)}$ and w_{fuse} are all set to 1. Adam optimizer [120] is used to train our network and its hyper parameters are set to default (initial learning rate lr=1e-3, betas=(0.9, 0.999), eps=1e-8, weight_decay=0). We train the network until the loss converges without using validation set which follows the previous methods [154], [157], [297]. We were using MSRA-B [160] as the validation set at the beginning. There seems to be overfitting in terms of the validation loss, but the F-measures of the validation set show no overfitting phenomena. Besides, we found that the overall performance on different datasets are more consistent with the training loss than the validation loss or F-measure most likely because these datasets

Table 5.3. Results of ablation study.

Architecture	DUT-OMRON		ECSSD	
	$maxF_\beta$	MAE	$maxF_\beta$	MAE
Baseline U-Net	0.725	0.082	0.896	0.066
PLN U-Net	0.782	0.062	0.928	0.043
RES U-Net	0.781	0.065	0.933	0.042
DSE U-Net	0.790	0.067	0.927	0.046
INC U-Net	0.777	0.069	0.921	0.047
(Ours) RSU U^2-Net	**0.823**	**0.054**	**0.951**	**0.033**
(Ours†) RSU U^2-Net†	**0.813**	**0.060**	**0.943**	**0.041**

have different distributions. Therefore, the validation sets are not used in the training processes of both our BASNet and U^2-Net (Chapter 5.5). After 600k iterations (with a batch size of 12), the training loss converges and the whole training process takes about 120 hours. During testing, the input images ($H \times W$) are resized to 320×320 and fed into the network to obtain the saliency maps. The predicted saliency maps with size of 320×320 are resized back to the original size of the input image ($H \times W$). Bilinear interpolation is used in both resizing processes. Our network is implemented based on Pytorch 0.4.0 [194]. Both training and testing are conducted on an eight-core, 16 threads PC with an AMD Ryzen 1800x 3.5 GHz CPU (32GB RAM) and a GTX 1080ti GPU (11GB memory).

5.5.7 Ablation Study

To validate the effectiveness of our newly designed RSU blocks, we fix the outside Encoder-Decoder architecture of our U^2-Net and replace its stages with other popular blocks including plain convolution blocks (PLN), residual-like blocks (RSE), dense-like blocks (DSE), and inception-like blocks (INC), as shown in Fig. 5.8 (a)-(d). Detailed configurations can be found in Table 5.2. Table 5.3 shows the quantitative results of the ablation study. As we can see that both our U^2-Net and its smaller version U^2-Net† achieves significant improvements against other models. Particularly, our full size U^2-Net improves the $maxF_\beta$ about 3.3% and 1.8%, and decreases the MAE over 12.9% and 21.4% against the second best model on DUT-OMRON and ECSSD datasets,

respectively. Furthermore, compared with the baseline U-Net our full size U^2-Net significantly improves the $maxF_\beta$ by 9.8% and 5.5% on DUT-OMRON and ECSSD as well as greatly decreases the MAE by 34.1% and 50%.

5.6 Comprehensive Comparison against State-of-the-Arts

We compare our methods BASNet (348.5 MB), U^2-Net (full size 176.3 MB) and U^2-Net† (4.7 MB) with 22 state-of-the-art methods including one **AlexNet** based model: **MDF**; 13 **VGG** based models: **UCF**, **Amulet**, **NLDF**, **DSS**, **LFR**, **C2S**, **RAS**, **RADF**, **PAGRN**, **BMPM**, **PiCANet**, **MLMS**, **AFNet**; one **DenseNet** based model **MSWS**; one **ResNeXt** based model: **R^{3}Net**; and six **ResNet** based models: **CapSal**, **SRM**, **DGRL**, **PiCANetR**, **CPD**, **PoolNet**. For fair comparison, we mainly use the saliency results provided by the authors. For the missing results on certain datasets of some methods, we run their released code with their trained models on their suggested environment settings.

5.6.1 Quantitative Comparison

Table 5.4 and Table 5.5 illustrate the comparison of our methods (BASNet, U^2-Net and U^2-Net†) and other state-of-the-art models in terms of five evaluation metrics including maximal F-measure ($maxF_\beta$), Mean Absolute Error (MAE), weighted F-measure (F_β^w), structural similarity measure S_m and relaxed boundary F-measure $relaxF_\beta^b$. Fig. 5.12 and Fig. 5.13 show the Precision-Recall curves and F-measure curves of our methods and others.

As we can see in Table 5.4 and 5.5, although the over all performance of our BASNet is slightly inferior to that of our full size U^2-Net and PoolNet, it outperforms almost all of other state-of-the-art methods. Particularly, our BASNet achieves the best performance on PASCAL-S dataset and the second best performance (our full size U^2-Net is the best) in terms of boundary measure $relaxF_\beta^b$ on datasets DUT-OMRON, DUTS-TE and ECSSD. On dataset HKU-IS and SOD, our BASNet achieves the third best boundary performance just following our U^2-Net (the best) and PoolNet (the second best).

Table 5.4 and 5.5 illustrate that our U^2-Net achieves the best performance on datasets DUT-OMRON, HKU-IS and ECSSD. On DUTS-TE dataset our U^2-Net achieves the second best performance. On PASCAL-S, the performance

Table 5.4. Comparison of our models and 22 SOTA methods in terms of model size, $maxF_\beta$ (↑), MAE (↓), weighted F_β^w (↑), structure measure S_m (↑) and relax boundary F-measure $relaxF_\beta^b$ (↑). Red, Green, and Blue indicate the best, second best and third best performance.

Method	Backbone	Size(MB)	**DUT-OMRON** (5168)					**DUTS-TE** (5019)					**HKU-IS** (4447)				
			$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$	$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$	$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$
MDF$_{\text{TIP16}}$	AlexNet	112.1	0.694	0.142	0.565	0.721	0.406	0.729	0.099	0.543	0.723	0.447	0.860	0.129	0.564	0.810	0.594
UCF$_{\text{ICCV17}}$	VGG-16	117.9	0.730	0.120	0.573	0.760	0.480	0.773	0.112	0.596	0.777	0.518	0.888	0.062	0.779	0.875	0.679
Amulet$_{\text{ICCV17}}$	VGG-16	132.6	0.743	0.098	0.626	0.781	0.528	0.778	0.084	0.658	0.796	0.568	0.897	0.051	0.817	0.886	0.716
NLDF+$_{\text{CVPR17}}$	VGG-16	428.0	0.753	0.080	0.634	0.770	0.514	0.813	0.065	0.710	0.805	0.591	0.902	0.048	0.838	0.879	0.694
DSS+$_{\text{CVPR17}}$	VGG-16	237.0	0.781	0.063	0.697	0.790	0.559	0.825	0.056	0.755	0.812	0.606	0.916	0.040	0.867	0.878	0.706
LFR$_{\text{IJCAI18}}$	VGG-16	131.1	0.740	0.103	0.647	0.780	0.508	0.778	0.083	0.689	0.799	0.556	0.911	0.040	0.861	0.905	0.731
C2S$_{\text{ECCV18}}$	VGG-16	606.0	0.758	0.072	0.661	0.798	0.565	0.807	0.062	0.713	0.819	0.607	0.896	0.048	0.829	0.883	0.717
RAS$_{\text{ECCV18}}$	VGG-16	81.0	0.786	0.062	0.695	0.814	0.615	0.831	0.059	0.740	0.828	0.656	0.913	0.045	0.843	0.887	0.748
RADF+$_{\text{AAAI18}}$	VGG-16	239.0	0.791	0.061	0.723	0.815	0.579	0.821	0.061	0.748	0.814	0.608	0.914	0.039	0.872	0.888	0.725
PAGRN$_{\text{CVPR18}}$	VGG-19	-	0.771	0.071	0.622	0.775	0.582	0.854	0.055	0.724	0.825	0.692	0.918	0.048	0.820	0.887	0.762
BMPM$_{\text{CVPR18}}$	VGG-16	-	0.774	0.064	0.681	0.809	0.612	0.852	0.048	0.761	0.851	0.699	0.921	0.039	0.859	0.907	0.773
PiCANet$_{\text{CVPR18}}$	VGG-16	153.3	0.794	0.068	0.691	0.826	0.643	0.851	0.054	0.747	0.851	0.704	0.921	0.042	0.847	0.906	0.784
MLMS$_{\text{CVPR19}}$	VGG-16	263.0	0.774	0.064	0.681	0.809	0.612	0.852	0.048	0.761	0.851	0.699	0.921	0.039	0.859	0.907	0.773
AFNet$_{\text{CVPR19}}$	VGG-16	143.0	0.797	0.057	0.717	0.826	0.635	0.862	0.046	0.785	0.855	0.714	0.923	0.036	0.869	0.905	0.772
MSWS$_{\text{CVPR19}}$	Dense-169	48.6	0.718	0.109	0.527	0.756	0.362	0.767	0.908	0.586	0.749	0.376	0.856	0.084	0.685	0.818	0.438
R^{3}Net+$_{\text{IJCAI18}}$	ResNeXt	215.0	0.795	0.063	0.728	0.817	0.599	0.828	0.058	0.763	0.817	0.601	0.915	0.036	0.877	0.895	0.740
CapSal$_{\text{CVPR19}}$	ResNet-101	-	0.699	0.101	0.482	0.674	0.396	0.823	0.072	0.691	0.808	0.605	0.882	0.062	0.782	0.850	0.654
SRM$_{\text{ICCV17}}$	ResNet-50	189.0	0.769	0.069	0.658	0.798	0.523	0.826	0.058	0.722	0.824	0.592	0.906	0.046	0.835	0.887	0.680
DGRL$_{\text{CVPR18}}$	ResNet-50	646.1	0.779	0.063	0.697	0.810	0.584	0.834	0.051	0.760	0.836	0.656	0.913	0.037	0.865	0.897	0.744
PiCANetR$_{\text{CVPR18}}$	ResNet-50	197.2	0.803	0.065	0.695	0.832	0.632	0.860	0.050	0.755	0.859	0.696	0.918	0.043	0.840	0.904	0.765
CPD$_{\text{CVPR19}}$	ResNet-50	183.0	0.797	0.056	0.719	0.825	0.655	0.865	0.043	0.795	0.858	0.741	0.925	0.034	0.875	0.905	0.795
PoolNet$_{\text{CVPR19}}$	ResNet-50	273.3	0.808	0.056	0.729	0.836	0.675	0.880	0.040	0.807	0.871	0.765	0.932	0.033	0.881	0.917	0.811
BASNet (Ours)	ResNet-34	348.5	0.805	0.056	0.751	0.836	0.694	0.860	0.047	0.803	0.853	0.758	0.928	0.032	0.889	0.909	0.807
U^2-Net (Ours)	N/A	176.3	0.823	0.054	0.757	0.847	0.702	0.873	0.044	0.804	0.861	0.765	0.935	0.031	0.890	0.916	0.812
U^2-Net† (Ours)	N/A	4.7	0.813	0.060	0.731	0.837	0.676	0.852	0.054	0.763	0.847	0.723	0.928	0.037	0.867	0.908	0.794

Table 5.5. Comparison of our models and 22 SOTA methods in terms of model size, $maxF_\beta$ (↑), MAE (↓), weighted F_β^w (↑), structure measure S_m (↑) and relax boundary F-measure $relaxF_\beta^b$ (↑). Red, Green, and Blue indicate the best, second best and third best performance.

Method	Backbone	Size(MB)	**ECSSD** (1000)					**PASCAL-S** (850)					**SOD** (300)				
			$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$	$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$	$maxF_\beta$	MAE	F_β^w	S_m	$relaxF_\beta^b$
MDF$_{\text{TIP16}}$	AlexNet	112.1	0.832	0.105	0.705	0.776	0.472	0.759	0.142	0.589	0.696	0.343	0.746	0.192	0.508	0.643	0.311
UCF$_{\text{ICCV17}}$	VGG-16	117.9	0.903	0.069	0.806	0.884	0.669	0.814	0.115	0.694	0.805	0.493	0.808	0.148	0.675	0.762	0.471
Amulet$_{\text{ICCV17}}$	VGG-16	132.6	0.915	0.059	0.840	0.894	0.711	0.828	0.100	0.734	0.818	0.541	0.798	0.144	0.677	0.753	0.454
NLDF+$_{\text{CVPR17}}$	VGG-16	428.0	0.905	0.063	0.839	0.897	0.666	0.822	0.098	0.737	0.798	0.495	0.841	0.125	0.709	0.755	0.475
DSS+$_{\text{CVPR17}}$	VGG-16	237.0	0.921	0.052	0.872	0.882	0.696	0.831	0.093	0.759	0.798	0.499	0.846	0.124	0.710	0.743	0.444
LFR$_{\text{IJCAI18}}$	VGG-16	131.1	0.911	0.052	0.858	0.897	0.694	0.801	0.107	0.737	0.805	0.499	0.828	0.123	0.734	0.773	0.479
C2S$_{\text{ECCV18}}$	VGG-16	606.0	0.910	0.055	0.851	0.893	0.708	0.840	0.082	0.766	0.836	0.543	0.823	0.124	0.700	0.760	0.457
RAS$_{\text{ECCV18}}$	VGG-16	81.0	0.921	0.056	0.857	0.893	0.741	0.829	0.101	0.736	0.799	0.560	0.851	0.124	0.720	0.764	0.544
RADF+$_{\text{AAAI18}}$	VGG-16	239.0	0.923	0.049	0.883	0.894	0.720	0.830	0.097	0.755	0.802	0.515	0.838	0.126	0.729	0.757	0.476
PAGRN$_{\text{CVPR18}}$	VGG-19	-	0.927	0.061	0.834	0.889	0.747	0.847	0.090	0.738	0.822	0.594	-	-	-	-	-
BMPM$_{\text{CVPR18}}$	VGG-16	-	0.928	0.045	0.871	0.911	0.770	0.850	0.074	0.779	0.845	0.617	0.856	0.108	0.726	0.786	0.562
PiCANet$_{\text{CVPR18}}$	VGG-16	153.3	0.931	0.046	0.865	0.914	0.784	0.856	0.078	0.772	0.848	0.612	0.854	0.103	0.722	0.789	0.572
MLMS$_{\text{CVPR19}}$	VGG-16	263.0	0.928	0.045	0.871	0.911	0.770	0.855	0.074	0.779	0.844	0.620	0.856	0.108	0.726	0.786	0.562
AFNet$_{\text{CVPR19}}$	VGG-16	143.0	0.935	0.042	0.887	0.914	0.776	0.863	0.070	0.798	0.849	0.626	0.856	0.111	0.723	0.774	-
MSWS$_{\text{CVPR19}}$	Dense-169	48.6	0.878	0.096	0.716	0.828	0.411	0.786	0.133	0.614	0.768	0.289	0.800	0.167	0.573	0.700	0.231
R^{3}Net+$_{\text{IJCAI18}}$	ResNeXt	215.0	0.934	0.040	0.902	0.910	0.759	0.834	0.092	0.761	0.807	0.538	0.850	0.125	0.735	0.759	0.431
CapSal$_{\text{CVPR19}}$	ResNet-101	-	0.874	0.077	0.771	0.826	0.574	0.861	0.073	0.786	0.837	0.527	0.773	0.148	0.597	0.695	0.404
SRM$_{\text{ICCV17}}$	ResNet-50	189.0	0.917	0.054	0.853	0.895	0.672	0.838	0.084	0.758	0.834	0.509	0.843	0.128	0.670	0.741	0.392
DGRL$_{\text{CVPR18}}$	ResNet-50	646.1	0.925	0.042	0.883	0.906	0.753	0.848	0.074	0.787	0.839	0.569	0.848	0.106	0.731	0.773	0.502
PiCANetR$_{\text{CVPR18}}$	ResNet-50	197.2	0.935	0.046	0.867	0.917	0.775	0.857	0.076	0.777	0.854	0.598	0.856	0.104	0.724	0.790	0.528
CPD$_{\text{CVPR19}}$	ResNet-50	183.0	0.939	0.037	0.898	0.918	0.811	0.861	0.071	0.800	0.848	0.639	0.860	0.112	0.714	0.767	0.556
PoolNet$_{\text{CVPR19}}$	ResNet-50	273.3	0.944	0.039	0.896	0.921	0.813	0.865	0.075	0.798	0.832	0.644	0.871	0.102	0.759	0.797	0.606
BASNet (Ours)	ResNet-34	348.5	0.942	0.037	0.904	0.916	0.826	0.856	0.076	0.798	0.838	0.660	0.851	0.113	0.730	0.769	0.603
U^2-Net (Ours)	N/A	176.3	0.951	0.033	0.910	0.928	0.836	0.859	0.074	0.797	0.844	0.657	0.861	0.108	0.748	0.786	0.613
U^2-Net† (Ours)	N/A	4.7	0.943	0.041	0.885	0.918	0.808	0.849	0.086	0.768	0.831	0.627	0.841	0.124	0.697	0.759	0.559

of our U^2-Net is slightly inferior to CPD and competitive against AFNet and PoolNet. On SOD dataset, our U^2-Net achieves the second best overall performance and outperforms other state-of-the-art models except for PoolNet.

Our U^2-Net† is only 4.7 MB, which is currently the smallest model in the field of salient object detection. Compared with the second smallest model MSWS (48.6 MB), our U^2-Net† is 10 times smaller and its performance is much better. With much fewer parameters against other models, Our U^2-Net† still achieves surprisingly competitive results. Particularly, on datasets DUT-OMRON, HKU-IS and ECSSD, the performance of our small size U^2-Net† is only inferior to our full size U^2-Net, our BASNet and PoolNet and it outperforms other state-of-the-art methods, which usually have 10 times or even more parameters. Although the performance of U^2-Net† is not as good as our full size U^2-Net, we believe the difference is acceptable and its small size will facilitates its applications in many computation and memory constrained environments.

Fig. 5.12 and Fig. 5.13 illustrate the precision-recall and F-measure curves of our methods and other state-of-the-art methods. The performance comparison reflected by the precision-recall curves are almost consistent with the $maxF_\beta$ measures. The F-measure curves (see Fig. 5.13) are able to evaluate the robustness of the method against different binarization thresholds. The higher and the more flat the curves are, the better the performance of the model is. As we can see in Fig. 5.13, the curves of our BASNet are relatively more flat compared with others, which means the performance of BASNet is stable and robust against different thresholds partially thanks to our hybrid loss used in BASNet. The F-measure curves of our U^2-Net on different dataset are relatively higher than others, especially on datasets DUT-OMRON, HKU-IS and ECSSD, which means it is more likely to get higher F_β with certain thresholds. In addition, although the F-measure curves of our U^2-Net† are lower and less flat than U^2-Net and BASNet, it still able to achieve a relatively high F_β with certain thresholds.

Although our models achieve competitive results against other state-of-the-art methods, there are still large rooms for improvements. As we can see

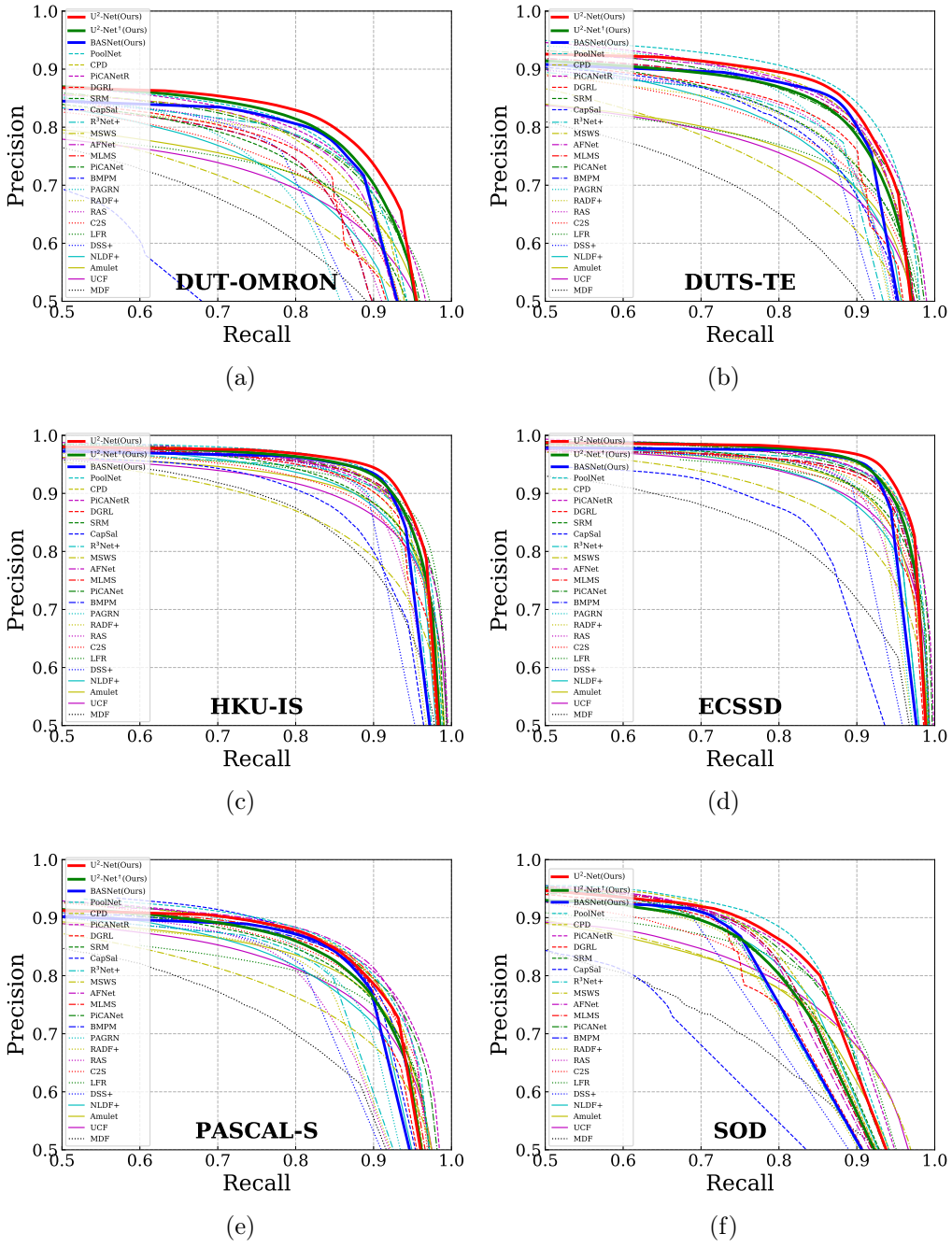


Figure 5.12. Precision-Recall curves of different method on six datasets.

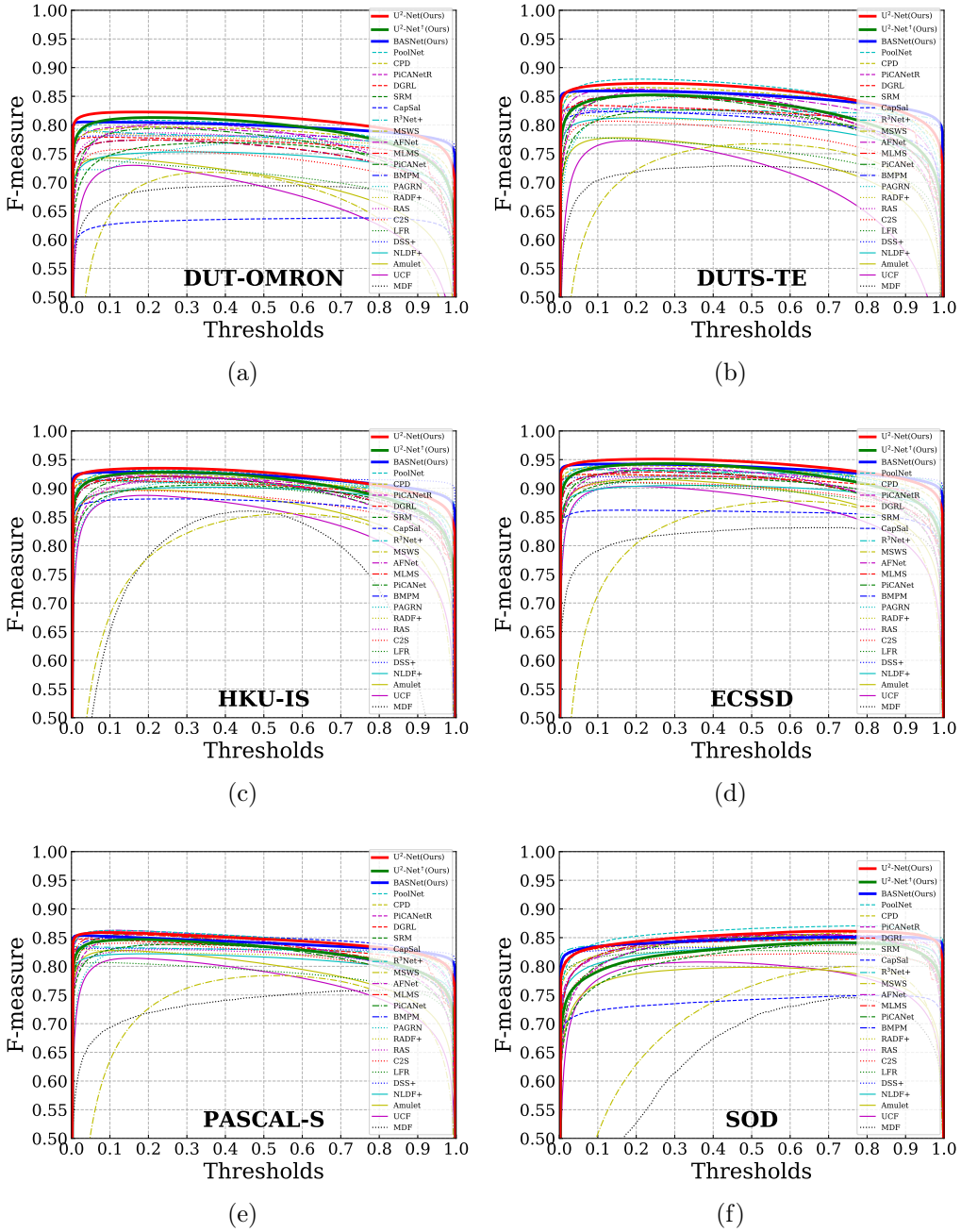


Figure 5.13. F-measure curves of different methods on six datasets.

in Fig. 5.13, our models achieve high F-measures (close to 0.95) on HKU-IS and ECSSD datasets while obtaining relatively low F-measures (around 0.85) on other four datasets including DUT-OMRON, DUTS-TE, PASCAL-S and SOD. In each dataset, there are both high-quality and low-quality predictions. Their different ratios lead to the differences on the final F-measures. There are multiple reasons for resulting the differences. The main reason is that these datasets are collected by different groups so that they have different characteristics including object shapes, object numbers, background features, image contrasts, *etc.* Although the training dataset, DUTS-TR, is relatively large (10553 images), it is still not able to cover all the cases in each testing dataset. Another important reason is that there is no unified mathematic definition of the salient object. Hence, the quality of the "ground truth" itself is imperfect and varies with different annotators. Therefore, although the improvements of our models on certain datasets are relatively insignificant against the remaining errors, their performance should not be underrated while taking the above facts into consideration.

5.6.2 Qualitative Comparison

To further demonstrate the performance of our methods, the qualitative comparisons of our methods and other state-of-the-art models are presented in this section.

Fig. 5.14 shows the comparison on the image with small object. Small object segmentation is a difficult yet key issue in SOD. Because the small targets are easy to be neglected by the downsampling operations in the networks. As we can see that, all of our methods, U^2-Net, U^2-Net†, and BASNet segment the small target correctly. Except for MSWS, other SOTA models either produce erroneously targets like CPD,PAGRN, RADF+, LFR, Amulet, UCF and MDF, or segment the whole image as background.

Fig. 5.15 shows the comparison of results on large object touching image boundaries. Large objects require models have the ability of extracting large scale features while maintaining the fine structures. That means the models have to correctly segment the building wall as well as the crosses on the roof.

Our methods U^2-Net, U^2-Net† and BASNet produce much more accurate segmentation results of the crosses than other SOTA methods. For segmenting the building wall, R^3-Net+ and PiCANet produce the best results. Although our U^2-Net† and BASNet segment a samll part of the wall after/below the fence as background, the overall results are relatively good. Our U^2-Net has the same problem with PoolNet and PiCANetR, which predict the small part of the left wall as background. But its overall prediction is still much better than most of other models like PoolNet, CPD, DGRL, SRM, CapSal, MSWS, AFNet, PAGRN, RADF+, RAS, C2S, LFR, DSS+, NLDF+, Amulet, UCF and MDF.

Fig. 5.16 and Fig. 5.17 demonstrate the ability of our method in handling one or multiple objects with fine structures. In Fig. 5.16, our models U^2-Net, U^2-Net† and BASNet predict the most accurate fins as well as the most clear and smoothest contours of the dolphin body. Fig. 5.17 shows that only our three models are able to correctly segment the legs of these horses, which further demonstrates the strong ability of our methods in handling fine details.

Fig. 5.18 and Fig. 5.19 show the results on targets with complex contours and complex structures respectively. In Fig. 5.18, both our U^2-Net and BASNet outperform other SOTA models. Although the prediction confidence of our U^2-Net† is lower than that of our U^2-Net and BASNet, it still extracts relatively higher quality contours than other SOTA models. As we can see in Fig. 5.19, U^2-Net and U^2-Net† achieves almost the same results with the ground truth and outperforms others with a large margin. As for the BASNet, although it fails in segmenting the top part of the target but it still produces very high quality boundaries compared with other SOTA models.

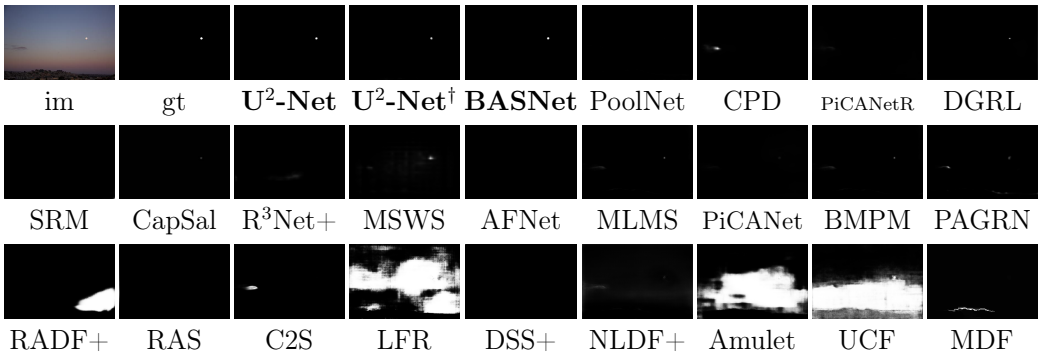


Figure 5.14. Qualitative comparison of results on **SMALL** object. U^2-Net, U^2-Net† and BASNet are our methods.

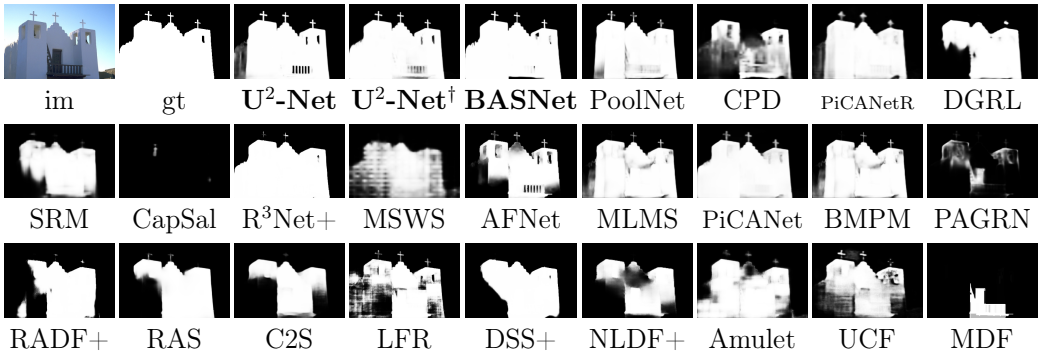


Figure 5.15. Qualitative comparison of results on **LARGE** object. U^2-Net, U^2-Net† and BASNet are our methods.

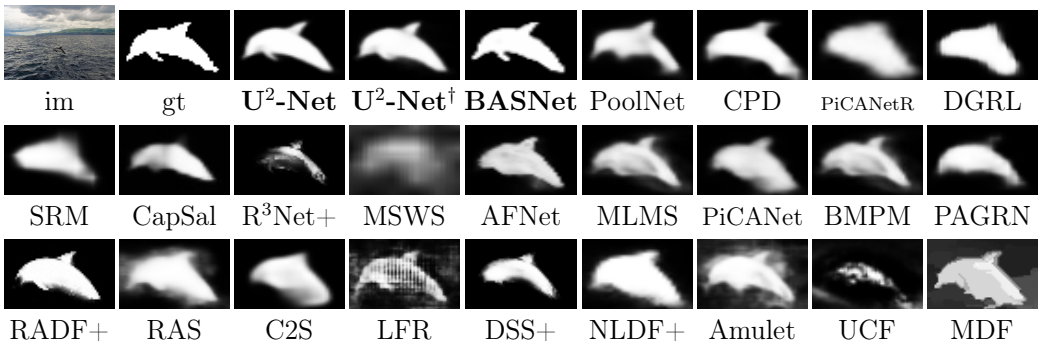


Figure 5.16. Qualitative comparison of results on **SMALL** object **WITH FINE STRUCTURES**. U^2-Net, U^2-Net† and BASNet are our methods.

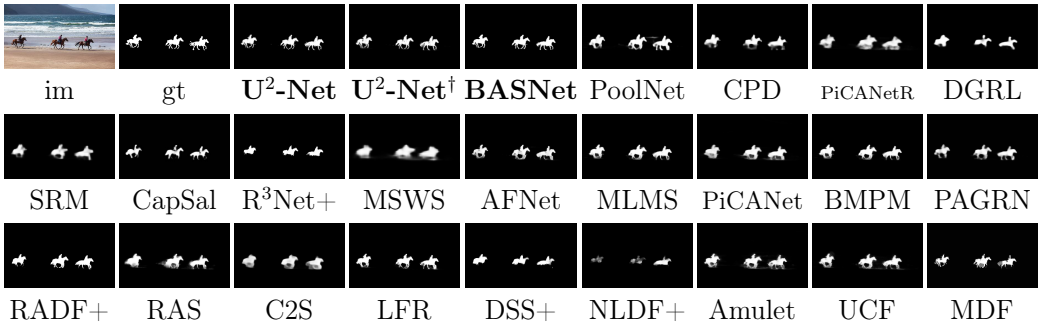


Figure 5.17. Qualitative comparison of results on **MULTIPLE OBJECTS WITH FINE STRUCTURE**. U^2-Net, U^2-Net† and BASNet are our methods.

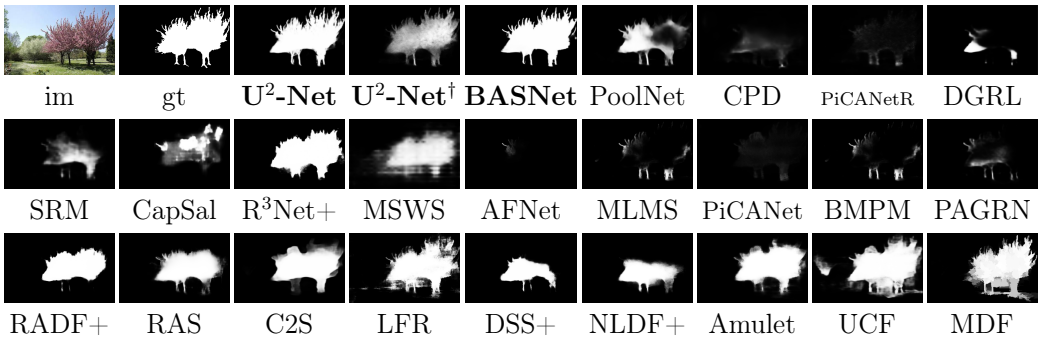


Figure 5.18. Qualitative comparison of results on object with **COMPLEX CONTOUR**. U^2-Net, U^2-Net† and BASNet are our methods.

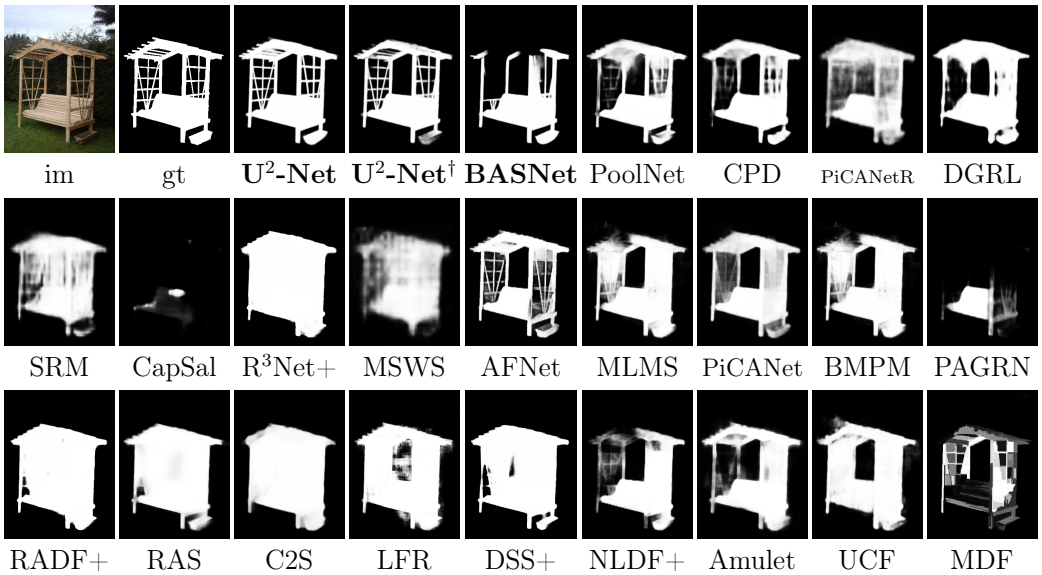


Figure 5.19. Qualitative comparison of results on **COMPLEX** object. U^2-Net, U^2-Net† and BASNet are our methods.

5.7 Summary

To achieve accurate salient object detection results, we proposed a novel end-to-end boundary-aware model, BASNet, and a hybrid fusing loss for accurate salient object detection. The proposed BASNet is a predict-refine architecture, which consists of two components: a prediction network and a refinement module. Combined with the hybrid loss, BASNet is able to capture both large-scale and fine structures, *e.g.* thin regions, holes, and produce salient object detection maps with clear boundaries. Experimental results on six datasets demonstrate that our model achieves very competitive performance against other state-of-the-art methods in terms of both region-based and boundary-aware measures. It runs at 25 FPS (Frame Per Second), which is close to real-time on a single 1080ti GPU. Additionally, our proposed network architecture is modular. It can be easily extended or adapted to other tasks by replacing either the predicting network or the refinement module.

To exclude the dependence on existing backbone networks and achieve more flexible salient object detection models, we proposed a novel deep network: U^2-Net, for salient object detection. The main architecture of our U^2-Net is a two-level nested U-structure. The nested U-structure with our newly designed RSU blocks enables the network to capture richer local and global information from both shallow and deep layers regardless of the resolutions. Compared with those SOD models built upon the combination of existing backbones and additionally developed multi-scale information extraction modules, our U^2-Net is purely built on the proposed RSU blocks which makes it possible to be trained from scratch and configured to have different model size according to the target environment constraints. We provide a full size U^2-Net (176.3 MB, 30 FPS) and a smaller size version U^2-Net† (4.7 MB, 40 FPS) in this thesis. Experimental results on six public saliency detection datasets demonstrate that both models achieve very competitive performance against other 22 state-of-the-art methods in terms of both qualitative and quantitative measures.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize our main contributions. Then, we analyze the limitations of our works and discuss possible improvements and future work.

6.1 Conclusions

In this thesis, we study salient object detection related problems in three different aspects: (1) interactive salient object detection; (2) unsupervised salient closed boundary extraction by perceptual grouping; (3) supervised salient object detection by deep convolutional neural networks.

In Chapter 3, we present a novel boundary based semi-automatic image annotation tool, **ByLabel**, to satisfy the high accuracy and reliability demands of certain applications. Using this tool, human operators are able to manually annotate the targets according to their specific applications. Compared with "automatic" algorithms, our interactive annotation tool, ByLabel, is more reliable, accurate and flexible. In contrast to other interactive methods, ByLabel requires less human intervention and reduces the time costs by simplifying the control points' clicking operations for polygon generation to selecting operations for picking detected edge fragments. This simplification improves both accuracy and efficiency. Because those detected edge fragments are one pixel-width pixel chains which are more accurate than line segments comprised of control points in depicting smooth curves. The reason for the efficiency improvement is that the selecting operations are much easier than

localizing and clicking exact boundary pixels.

In Chapter 4, we propose a perceptual grouping based salient closed boundary extraction method and apply our method to different applications. The key components of our method includes a general workflow and a graph-based optimization algorithm: "Bi-Directional Shortest Path (BDSP)". The general workflow reduces the salient closed boundary extraction problem into the problem of searching for a special cycle from an undirected graph. The BDSP algorithm solves the optimization problem on the undirected graph in polynomial-time. Compared with other optimization methods, BDSP is more tolerant to the formats of the objective functions. In addition, we adapt our salient closed boundary extraction method to following two applications:

- **Building outline extraction via line segments perceptual grouping**: we first adapt the general workflow to reduce the building outline extraction problem into searching for a special cycle from an undirected graph constructed based on the detected line segments. Then, we design a novel saliency objective function by encoding the proximity and continuity principles of Gestalt Laws. Combined with the BDSP algorithm, our method is able to extract building outlines with different shapes and achieves state-of-the-art performance against other methods.

- **Salient closed boundary tracking**: i) *salient closed boundary tracking via line segments perceptual grouping.* We extend the salient closed boundary workflow to a tracking workflow by adding an area changing constraint between the closed boundaries on the current and previous frames. By using the BDSP, we can optimize the saliency objective function with the area constraint easily. Additionally, we build a salient closed boundary tracking dataset including nine video sequences (with 9598 frames in total) with manually annotated ground truth using our ByLabel. We evaluate our method on this dataset. The results show the state-of-the-art performance of our method in tracking closed boundaries without stable or enough supportive region information. To further demonstrate the promising performance of our method, we evaluate on

a real-world robot pouring task that requires continuous tracking of the bowl rim. To achieve more accurate and robust tracking results, we further adapt our method to ii) *salient closed boundary tracking via edge fragments perceptual grouping*. This method replaces the line segments of the above method with edge fragments to achieve higher accuracy for salient closed boundary tracking. We design a new way of edge breaking to obtain high quality edge fragments. To reduce the time costs of optimization, we develop a novel edge filtering approach based on the distance map of the boundary tracked from the last frame. We test our method on the dataset built in the above work. The experimental results show that our edge fragments based tracker outperforms other methods, including our last line segments based tracker in terms of accuracy. In addition, this tracker still runs at real-time speed.

In Chapter 5, we design two deep convolutional neural networks, BASNet and U^2-Net, for high accuracy and fast salient object detection.

- **BASNet**, Boundary-Aware Salient object detection network, is designed with a predict-refine architecture and trained with a novel hybrid loss. The predict module takes an natural image as input and outputs the coarse probability map. The refine module further refines the coarse probability map by feeding it into another small encoder-decoder module. The hybrid loss combines the binary cross entropy, structural similarity and intersection over union to learn the pixel-wise, patch-wise and map-wise structures, respectively. The predict and refine modules combined with the hybrid loss are trained end-to-end. Running at 25 FPS, which is promising to be used in real-time applications, It achieves very competitive performance in terms of both region-based and boundary-based quality.

- Our **U^2-Net** is a nested U-structure built upon our newly designed residual U-block, which is a combination of an U-structure and a residual connection. The residual U-block enables effective extraction of multi-scale information while maintaining relatively low computation and memory

costs. Without using backbones, our U^2-Net is flexible and can be easily configured to work on different training datasets and working environments. We provide two instances of our U^2-Net: a full size U^2-Net (176 MB, 30 FPS on GTX 1080ti GPU with input image size of 320×320) and a light one U^2-Net† (4.7 MB, 40 FPS). Both instances achieve competitive results against the state-of-the-art methods. Although the performance of U^2-Net† is inferior to the full size U^2-Net, the difference is acceptable when the model size is taken into account. Before our **U^2-Net**, almost all of the salient object detection networks including our BASNet are based on the backbones adapted from image classification tasks, which impedes the further adaptation of these networks for different training datasets and working environments. Our nested U-structure can be easily extended to other binary or multiple class segmentation problems.

6.2 Limitations and Future Work

6.2.1 Interactive Annotation with ByLabel: A Boundary based Semi-Automatic Image Annotation Tool

There are two main limitations of ByLabel: reliable edge fragments generation from multi-scale images and labeling of shared boundaries.

First, the selecting operation of our ByLabel is based on the automatically detected edge fragments. There are mainly two steps for edge fragments generation: i) edge detection, and ii) edge breaking. ByLabel uses the Edge Drawing method for edge detection, with a filter size fixed to 3×3, and it is not able to handle images with high resolutions. Furthermore, the edges detected by EdgeDrawing are usually elongated. As a results, the foreground and background edge pixels are often erroneously tracked and grouped in the same long edge. Although we break these elongated edges into relatively short fragments using turning angles in ByLabel, without taking the semantic meaning of each edge pixel into consideration, the breaking quality is hard to be guaranteed. Therefore, new multi-scale edge detectors would be a valuable direction to explore. Specifically, multi-scale filters or strategies can be introduced to achieve

more reliable edge detection results. Traditional edge detectors only utilize local high frequency informations for edge detection. Introducing more regional low frequency information and combining them with the high frequency information might be another promising way for edge detection and breaking. Furthermore, learning based methods [161], [275], [287] for edge fragments extraction is gaining popularity. Developing lighter, faster and more reliable deep methods for edge detection is a promising direction.

Second, in the labeling of shared boundaries, the annotation logic of our ByLabel is to manually select and group the closed boundaries of target objects. One assumption of this logic is that the annotation of each target object is independent and information is not reusable between two objects annotation processes. Another assumption is that the extracted boundary pixels belongs to the to-be-annotated foreground targets other than the backgrounds. However, these assumptions are potentially limited. For example, if two objects have a shared boundary, it would need to be annotated twice, once for each object. This can be a waste of time. Besides, the boundaries are comprised of pixels with the greatest gradient magnitudes along their gradient directions within the local patches. Theoretically, their ownership is illusive and they could be taken as parts of either the to-be-annotated targets or the backgrounds (or other targets). This will introduce discrepencies to the shared boundaries labeling so that have negative impacts on the accuracy of region masks generated from labeled or extracted boundaries. This problem will get even worse when there are a large number of shared boundaries such as in scene parsing applications where almost every adjacent subregion of the input image has to be labeled. To address this problem, scene parsing annotation logic and the topological representation of annotated boundaries could be introduced. The logic of scene parsing is to divide the whole image into multiple subregions according to their classes. Boundaries should be defined as the continuous paths in-between neighboring pixels other than pixels themselves. Instead of assigning each boundary to specific object, each boundary can be taken as a divider for separating the adjacent image regions. Annotated boundaries can then be saved into a topological format which will avoid

repeated storage of these boundaries.

In summary, interactive image annotation tools still play an important role in many computer vision tasks, especially when building a large number of training datasets. Therefore, it is important for future work to study the standard annotation protocols and web-based platforms. These techniques will enable remote and collaborative annotation for greatly improving annotation accuracy and efficiency.

6.2.2 Unsupervised Salient Closed Boundary Extraction by Perceptual Grouping

In Chapter 3, we present our perceptual grouping method, especially the "Bi-Directional Shortest Path (BDSP)" algorithm, for salient closed boundary extraction and its different applications.

Although the "Bi-Directional Shortest Path (BDSP)" algorithm provides a flexible and efficient solution for the graph-based optimization problems, it is still not able to guarantee the absolute correctness of the to-be-extracted closed boundaries. There are mainly two reasons. First, the results of BDSP are approximately global and biased to certain types of closed boundaries. In BDSP algorithm, its two shortest paths are usually searched based on the local cues, *e.g.* gap length, by Dijkstra. But the grouping costs are often defined with both local and global cues, which are not able to be considered in the shortest paths searching processes. Second, even if the global optima of a grouping cost is guaranteed, the result is still possible to be incorrect. Because the grouping costs are usually designed by trial-and-error based on the human knowledge and experiences on different applications. They are not completely consistent with the essence of the target boundaries. Therefore, the global optima guaranteed optimizers and better strategies of grouping costs design are needed.

The second limitation of our salient closed boundary extraction is similar to that of ByLabel. The performance our method heavily relies on the detected line segments or edge fragments. However, traditional line and edge detectors are often sensitive to variations of illumination, scale, threshold, *etc.* To ad-

dress this problem, more regional information and prior knowledge need to be integrated with these unsupervised detectors. Therefore, there is a need for exploring the methods and strategies of combining unsupervised and supervised techniques for low level feature extraction. To achieve more robust and accurate results, exploring hierarchical perceptual grouping methods, which are able to progressively refine the grouping results, is a promising study direction.

Another drawback of this method is only able to extract a single salient closed boundary. However, many applications such as building outline extraction from large area images and multiple object tracking (MOT) require simultaneous extraction of multiple salient closed boundaries. To address this problem, new problem formulation strategies, loss functions and optimization algorithms should be explored. Furthermore, graph convolutional neural network is an additional technique that shows promise for solving the perceptual grouping problem.

The widely used thresholds in those unsupervised methods potentially impede their practical applications in real-world environments. There are also several thresholds, such as the buffer size and the area variation threshold in Chapter 4.6.1, the turning distance, edge filtering parameters, perimeter and area variation thresholds in Chapter 4.6.2, in our proposed method. Although these parameters are not deterministic to the effectiveness of our method, they still play important roles in guaranteeing the performance of the whole pipeline. The optimal values of these parameters usually vary with different application scenarios or targets. Hence, they are often set based on the experiences or experiments. Exploring novel adaptive methods for automatic parameters tunning by integrating unsupervised, semi-supervised or supervised methods could facilitate the wider applications of our method in real-world scenarios.

In addition to these technical problems, the testing datasets also play important roles in the study of unsupervised methods. Large scale and diversified datasets are able to provide more complete and reliable evaluation on newly developed algorithms. Hence, there is a need for creating larger and more comprehensive datasets.

6.2.3 Supervised Salient Object Detection by Deep Convolutional Neural Networks

Although our BASNet and U^2-Net achieve very competitive results against current state-of-the-art methods on six public datasets, there are still several drawbacks of our methods.

Running time and model size are two important metrics in evaluating a neural network. Both our BASNet and U^2-Net achieve real-time running speed, but they are running on GTX 1080ti GPU and the size of the input images is relatively small (256×256 or 320×320). Although our U^2-Net† is only 4.7 MB and runs at 40 FPS on a GTX 1080ti GPU, it is still difficult to deploy to mobile devices. In addition, with the fast development of image sensors, image resolutions of mobile devices are increasing rapidly. This introduces more challenges to the design of deep networks for real-time applications. Therefore, deep network compression and light network design remain open problems.

The performance of a deep network often heavily relies on the training dataset. The most frequently used training dataset for salient object detection, DUTS-TR, only contains 10553 images. This dataset is good for research purposes, however, if we want to train robust networks for real world applications, more and larger training datasets are needed.

Currently, most of the deep networks for salient object detection are manually designed. The more important issue here is that manually designed "optimal" networks can be biased to a certain datasets. The optimal solution is to search for the network architecture automatically with the given training dataset. Neural architecture searching techniques will be another promising direction for salient object detection and other related fields.

6.2.4 Evaluation Measures

In addition to those methods and datasets related issues mentioned above, evaluation measures also play very important roles in almost all of the studies of computer vision and related areas. The proper usage of the evaluation measures is able to greatly facilitate the development of novel methods and

techniques. There are many evaluation measures have been proposed such as precision, recall, F-measure, Intersection-over-Union, relax boundary F-measure [181], Hausdorff distance (HD) [209], Contour Mapping (CM) [183], etc. However, every evaluation measure is more or less biased to certain types of errors. The selection of evaluation measures relies on the demands of different applications and the characteristics of to-be-evaluated targets. Although several different evaluation measures are used in this thesis for providing relatively comprehensive comparisons, each measure is only able to give an overall quality estimation in terms of certain type of errors. Currently, it is hard to find a unified measure that equally evaluates all types of errors. Therefore, there is a need for developing task-specific novel evaluation measures as well as standard selection criteria of evaluation measures.

References

[1] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *International conference on computer vision systems*, Springer, 2008, pp. 66–75.

[2] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 2009, pp. 1597–1604.

[3] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, 32, no., pp. 1633–1642, 2011.

[4] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE transactions on pattern analysis and machine intelligence*, 34, no., pp. 2189–2202, 2012.

[5] E. J. Almazan, R. Tal, Y. Qian, and J. H. Elder, "Mcmlsd: A dynamic programming approach to line segment detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2031–2039.

[6] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International journal of computer vision*, 1, no., pp. 333–356, 1988.

[7] T. D. Alter and R. Basri, "Extracting salient curves from images: An analysis of the saliency network," *International Journal of Computer Vision*, 27, no., pp. 51–69, 1998.

[8] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 855–867, 1990.

[9] A. Amir and M. Lindenbaum, "Quantitative analysis of grouping processes," in *European Conference on Computer Vision*, Springer, 1996, pp. 369–384.

[10] ——, "A generic grouping algorithm and its quantitative analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, no., pp. 168–185, 1998.

[11] M. Amirul Islam, M. Kalash, and N. D. Bruce, "Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7142–7150.

[12] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," in *ACM Transactions on graphics (TOG)*, ACM, vol. 26, 2007, p. 10.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 2481–2495, 2017.

[14] S. Baker and I. A. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, 56, no., pp. 221–255, 2004.

[15] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," in *Readings in computer vision*, Elsevier, 1987, pp. 714–725.

[16] ——, "Animate vision," *Artificial intelligence*, 48, no., pp. 57–86, 1991.

[17] S. Baluja and D. A. Pomerleau, "Using a saliency map for active spatial selective attention: Implementation & initial results," in *Advances in Neural Information Processing Systems*, 1995, pp. 451–458.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, Springer, 2006, pp. 404–417.

[19] S. Benhimane and E. Malis, "Real-time image-based tracking of planes using efficient second-order minimization," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28 - October 2, 2004*, 2004, pp. 943–948.

[20] S. Bianco, G. Ciocca, P. Napoletano, and R. Schettini, "An interactive tool for manual, semi-automatic and automatic video annotation," *Computer Vision and Image Understanding*, 131, no., pp. 88–99, 2015.

[21] C. Bibby and I. D. Reid, "Real-time tracking of multiple occluding objects using level sets," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010, pp. 1307–1314.

[22] P. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals OR*, 134, no., pp. 19–67, 2005.

[23] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: A survey," *Computational Visual Media*, no., pp. 1–34,

[24] A. Borji, M. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. Image Processing*, 24, no., pp. 5706–5722, 2015.

[25] A. Borji and L. Itti, "Scene classification with a sparse set of salient regions," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 1902–1908.

[26] ——, "Exploiting local and global patch rarities for saliency detection," in *2012 IEEE conference on computer vision and pattern recognition*, IEEE, 2012, pp. 478–485.

[27] ——, "State-of-the-art in visual attention modeling," *IEEE transactions on pattern analysis and machine intelligence*, 35, no., pp. 185–207, 2012.

[28] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *ICCV*, 2001, pp. 105–112.

[29] A. Brook, A. M. Bruckstein, and R. Kimmel, "On similarity-invariant fairness measures," in *Scale Space and PDE Methods in Computer Vision, 5th International Conference, Scale-Space 2005, Hofgeismar, Germany, April 7-9, 2005, Proceedings*, 2005, pp. 456–467.

[30] N. D. Bruce and J. K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of vision*, 9, no., pp. 5–5, 2009.

[31] N. Bruce and J. Tsotsos, "Saliency based on information maximization," in *Advances in neural information processing systems*, 2006, pp. 155–162.

[32] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no., pp. 679–698, 1986.

[33] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer vision, graphics, and image processing*, 37, no., pp. 54–115, 1987.

[34] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische mathematik*, 66, no., pp. 1–31, 1993.

[35] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, 22, no., pp. 61–79, 1997.

[36] R. L. Castaño and S. Hutchinson, "A probabilistic approach to perceptual grouping," *Computer Vision and Image Understanding*, 64, no., pp. 399–419, 1996.

[37] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler, "Annotating object instances with a polygon-rnn," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 4485–4493.

[38] M. Cazorla, F. Escolano, D. Gallardo, and R. Rizo, "Junction detection and grouping with probabilistic edge models and bayesian a," *Pattern Recognition*, 35, no., pp. 1869–1881, 2002.

[39] D. M. Chan and L. D. Riek, "Unsupervised salient object discovery for robots," *Rob.: Sci. and Sys.(RSS) Pioneers*, no., 2019.

[40] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on image processing*, 10, no., pp. 266–277, 2001.

[41] K.-Y. Chang, T.-L. Liu, H.-T. Chen, and S.-H. Lai, "Fusing generic objectness and visual saliency for salient object detection," in *2011 International Conference on Computer Vision*, IEEE, 2011, pp. 914–921.

[42] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, 40, no., pp. 834–848, 2017.

[43] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IX*, 2018, pp. 236–252.

[44] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, no., pp. 569–582, 2014.

[45] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3286–3293.

[46] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, 17, no., pp. 790–799, 1995.

[47] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," in *ACM Transactions on Graphics (TOG)*, ACM, vol. 30, 2011, p. 156.

[48] C. Choi and H. I. Christensen, "3d textureless object detection and tracking: An edge-based approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, 2012, pp. 3877–3884.

[49] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 564–575, 2003.

[50] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, 61, no., pp. 38–59, 1995.

[51] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[52] M. Cote and P. Saeedi, "Automatic Rooftop Extraction in Nadir Aerial Imagery of Suburban Regions Using Corners and Variational Level Set Evolution," *IEEE Transactions on Geoscience and Remote Sensing*, 51, no., pp. 313–328, Jan. 2013.

[53] I. J. Cox, S. B. Rao, and Y. Zhong, "" ratio regions": A technique for image segmentation," in *ICPR*, vol. 2, 1996, pp. 557–564.

[54] I. J. Cox, J. M. Rehg, and S. Hingorani, "A bayesian multiple-hypothesis approach to edge grouping and contour segmentation," *International Journal of Computer Vision*, 11, no., pp. 5–24, 1993.

[55] D. Crevier, "A probabilistic method for extracting chains of collinear segments," *Computer Vision and Image Understanding*, 76, no., pp. 36–53, 1999.

[56] ——, "Bayesian extraction of collinear segment chains from digital images," in *Perceptual Organization for Artificial Vision Systems*, Springer, 2000, pp. 311–323.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[58] Z. Deng, X. Hu, L. Zhu, X. Xu, J. Qin, G. Han, and P.-A. Heng, "R3net: Recurrent residual refinement network for saliency detection," IJCAI, 2018.

[59] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, 1, no., pp. 269–271, 1959.

[60] D. S. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3-8, 2000.*, 2000, pp. 4167–4170.

[61] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, 37, no., pp. 1558–1570, 2014.

[62] N. D. H. Dowson and R. Bowden, "Mutual information for lucas-kanade tracking (MILK): an inverse compositional formulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 30, no., pp. 180–185, 2008.

[63] M. Ehrig and J. Euzenat, "Relaxed precision and recall for ontology matching," in *Proc. K-Cap 2005 workshop on Integrating ontology*, No commercial editor., 2005, pp. 25–32.

[64] J. H. Elder and R. M. Goldberg, "Image editing in the contour domain," in *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, IEEE, 1998, pp. 374–381.

[65] J. H. Elder, A. Krupnik, and L. A. Johnston, "Contour grouping with prior models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, no., pp. 661–674, 2003.

[66] J. H. Elder and S. W. Zucker, "Computing contour closure," in *Computer Vision - ECCV'96, 4th European Conference on Computer Vision, Cambridge, UK, April 15-18, 1996, Proceedings, Volume I*, 1996, pp. 399–412.

[67] J. H. Elder and S. W. Zucker, "Local scale control for edge detection and blur estimation," *IEEE Transactions on pattern analysis and machine intelligence*, 20, no., pp. 699–716, 1998.

[68] J. Elder and S. Zucker, "The effect of contour closure on the rapid discrimination of two-dimensional shapes," *Vision Research*, 33, no., pp. 981–991, 1993.

[69] G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, and Y. Avrithis, "Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention," *IEEE Transactions on Multimedia*, 15, no., pp. 1553–1568, 2013.

[70] M. Everingham, S. M. A. Eslami, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, 111, no., pp. 98–136, 2015.

[71] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4548–4557.

[72] R. Fan, M.-M. Cheng, Q. Hou, T.-J. Mu, J. Wang, and S.-M. Hu, "S4net: Single stage salient-instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6103–6112.

[73] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of Computing*, 8, no., pp. 415–428, 2012.

[74] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1623–1632.

[75] L. Fidon, W. Li, L. C. Herrera, J. Ekanayake, N. Kitchen, S. Ourselin, and T. Vercauteren, "Generalised wasserstein dice score for imbalanced multi-class segmentation using holistic convolutional networks," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries - Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada*, 2017, pp. 64–76.

[76] S. Frintrop and M. Kessel, "Most salient region tracking," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 1869–1874.

[77] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *International Journal of Computer Vision*, 94, no., pp. 335–360, 2011.

[78] X. Giró i Nieto, N. Camps, and F. Marqués, "GAT: a graphical annotation tool for semantic regions," *Multimedia Tools Appl.*, 46, no., pp. 155–174, 2010.

[79] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[80] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[81] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 2010, pp. 249–256.

[82] M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Computer Vision and Image Understanding*, 117, no., pp. 1245–1256, 2013.

[83] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, 34, no., pp. 1915–1926, 2011.

[84] C. Goldberg, T. Chen, F.-L. Zhang, A. Shamir, and S.-M. Hu, "Data-driven object manipulation in images," in *Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 265–274.

[85] R. H. Hahnloser and H. S. Seung, "Permitted and forbidden sets in symmetric threshold-linear networks," in *Advances in Neural Information Processing Systems*, 2001, pp. 217–223.

[86] C. Halaschek-Wiener, J. Golbeck, A. Schain, M. Grove, B. Parsia, and J. A. Hendler, "Annotation and provenance tracking in semantic web photo libraries," in *Provenance and Annotation of Data, International Provenance and Annotation Workshop, IPAW 2006, Chicago, IL, USA, May 3-5, 2006, Revised Selected Papers*, 2006, pp. 82–89.

[87] X. Han, C. Xu, and J. L. Prince, "A topology preserving deformable model using level sets," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE, vol. 2, 2001, pp. II–II.

[88] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE transactions on pattern analysis and machine intelligence*, no., pp. 532–550, 1987.

[89] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Advances in neural information processing systems*, 2007, pp. 545–552.

[90] C. G. Harris, M. Stephens, *et al.*, "A combined corner and edge detector.," in *Alvey vision conference*, Citeseer, vol. 15, 1988, pp. 10–5244.

[91] S. G. Hart, "Nasa task load index (tlx). volume 1.0; paper and pencil package," no., 1986.

[92] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[93] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European conference on computer vision*, Springer, 2014, pp. 346–361.

[94] ——, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[95] S. He, J. Jiao, X. Zhang, G. Han, and R. W. Lau, "Delving into salient object subitizing and detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1059–1067.

[96] S. He, R. W. Lau, W. Liu, Z. Huang, and Q. Yang, "Supercnn: A superpixelwise convolutional neural network for salient object detection," *International journal of computer vision*, 115, no., pp. 330–344, 2015.

[97] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 5300–5309.

[98] P. Hu, B. Shuai, J. Liu, and G. Wang, "Deep level sets for salient object detection," in *CVPR*, vol. 1, 2017, p. 2.

[99] X. Hu, L. Zhu, J. Qin, C. Fu, and P. Heng, "Recurrently aggregating deep features for salient object detection," in *Proceedings of AAAI-18, New Orleans, Louisiana, USA*, 2018, pp. 6943–6950.

[100] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 2261–2269.

[101] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 262–270.

[102] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv: 1502. 03167*, no., 2015.

[103] M. A. Islam, M. Kalash, M. Rochan, N. D. Bruce, and Y. Wang, "Salient object detection using a context-aware refinement network," no., 2017.

[104] L. Itti, "Models of bottom-up and top-down visual attention," PhD thesis, California Institute of Technology, 2000.

[105] ——, "Visual salience," *Scholarpedia*, 2, no., p. 3327, 2007.

[106] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 1254–1259, 1998.

[107] P. Jaccard, "The distribution of the flora in the alpine zone. 1," *New phytologist*, 11, no., pp. 37–50, 1912.

[108] D. W. Jacobs, "Robust and efficient detection of convex groups," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1993, pp. 770–771.

[109] ——, "Robust and efficient detection of salient convex groups," *IEEE transactions on pattern analysis and machine intelligence*, 18, no., pp. 23–37, 1996.

[110] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Transactions on pattern analysis and machine intelligence*, 18, no., pp. 267–278, 1996.

[111] I. H. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, no., pp. 1075–1088, 2001.

[112] I. Jermyn and H. Ishikawa, "Globally optimal regions and boundaries as minimum ratio weight cycles," *IEEE Trans. Pattern Anal. Mach. Intell.*, 23, no., pp. 1075–1088, 2001.

[113] F. D. Julca-Aguilar and N. S. T. Hirata, "Expressmatch: A system for creating ground-truthed datasets of online mathematical expressions," in *10th IAPR International Workshop on Document Analysis Systems, DAS 2012, Gold Coast, Queenslands, Australia, March 27-29, 2012*, 2012, pp. 155–159.

[114] C. Kanan, M. H. Tong, L. Zhang, and G. W. Cottrell, "Sun: Top-down saliency using natural statistics," *Visual cognition*, 17, no., pp. 979–1003, 2009.

[115] G. Kanizsa, *Organization in Vision*. New York: Praeger 1979.

[116] M. Kass, A. P. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, 1, no., pp. 321–331, 1988.

[117] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, 1, no., pp. 321–331, 1988.

[118] R. Kennedy, J. H. Gallier, and J. Shi, "Contour cut: Identifying salient contours in images by solving a hermitian eigenvalue problem," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, 2011, pp. 2065–2072.

[119] J. Kim and V. Pavlovic, "A shape preserving approach for salient object detection using convolutional neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, 2016, pp. 609–614.

[120] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, no., 2014.

[121] B. Klava and N. S. T. Hirata, "A model for simulating user interaction in hierarchical segmentation," in *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, 2014, pp. 4358–4362.

[122] C. Koch and S. Ullman, "Shifts in selective visual attention: Towards the underlying neural circuitry," in *Matters of intelligence*, Springer, 1987, pp. 115–141.

[123] K. Koch, J. McLean, R. Segev, M. A. Freed, M. J. Berry II, V. Balasubramanian, and P. Sterling, "How much the eye tells the brain," *Current Biology*, 16, no., pp. 1428–1434, 2006.

[124] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013.

[125] I. Kokkinos, "Highly accurate boundary detection and grouping," in *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, 2010, pp. 2520–2527.

[126] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[127] M. Kristan, A. Leonardis, and J. Matas, "The visual object tracking VOT2016 challenge results," in *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, 2016, pp. 777–823.

[128] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.

[129] S. S. Kruthiventi, K. Ayush, and R. V. Babu, "Deepfix: A fully convolutional neural network for predicting human eye fixations," *IEEE Transactions on Image Processing*, 26, no., pp. 4446–4456, 2017.

[130] S. S. Kruthiventi, V. Gudisa, J. H. Dholakiya, and R. Venkatesh Babu, "Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5781–5790.

[131] J. Kuen, Z. Wang, and G. Wang, "Recurrent attentional networks for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3668–3677.

[132] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee, "A geometric particle filter for template-based visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 36, no., pp. 625–643, 2014.

[133] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," no., 2001.

[134] Q. Lai, W. Wang, H. Sun, and J. Shen, "Video saliency prediction using spatiotemporal residual attentive networks," *IEEE Transactions on Image Processing*, no., 2019.

[135] M. F. Land and M. Hayhoe, "In what ways do eye movements contribute to everyday activities?" *Vision research*, 41, no., pp. 3559–3565, 2001.

[136] T.-N. Le and A. Sugimoto, "Semantic instance meets salient object: Study on video semantic salient instance segmentation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 1779–1788.

[137] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86, no., pp. 2278–2324, 1998.

[138] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[139] G. Lee, Y.-W. Tai, and J. Kim, "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 660–668.

[140] M. E. Leventon, W. E. L. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contours," in *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.*, IEEE, 2002, 8–pp.

[141] D. T. Levin and D. J. Simons, "Failure to detect changes to attended objects in motion pictures," *Psychonomic Bulletin & Review*, 4, no., pp. 501–506, 1997.

[142] C. Li, C. Xu, C. Gui, and M. D. Fox, "Level set evolution without re-initialization: A new variational formulation," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 430–436.

[143] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, 106, no., pp. 59–70, 2007.

[144] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2386–2395.

[145] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5455–5463.

[146] ——, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[147] ——, "Visual saliency detection based on multiscale deep cnn features," *IEEE Transactions on Image Processing*, 25, no., pp. 5012–5024, 2016.

[148] X. Li, F. Yang, H. Cheng, W. Liu, and D. Shen, "Contour knowledge transfer for salient object detection," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, 2018, pp. 370–385.

[149] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287.

[150] P. Liang, Y. Wu, and H. Ling, "Planar object tracking in the wild: A benchmark," *CoRR*, abs/1703.07938, no., 2017.

[151] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, 2014, pp. 740–755.

[152] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," *arXiv preprint arXiv:1901.02985*, no., 2019.

[153] F. Liu and M. Gleicher, "Region enhanced scale-invariant saliency detection," in *2006 IEEE International Conference on Multimedia and Expo*, IEEE, 2006, pp. 1477–1480.

[154] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, "A simple pooling-based design for real-time salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3917–3926.

[155] N. Liu and J. Han, "Dhsnet: Deep hierarchical saliency network for salient object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 678–686.

[156] N. Liu, J. Han, T. Liu, and X. Li, "Learning to predict eye fixations via multiresolution convolutional neural networks," *IEEE Trans. Neural Netw. Learning Syst.*, 29, no., pp. 392–404, 2018.

[157] N. Liu, J. Han, and M.-H. Yang, "Picanet: Learning pixel-wise contextual attention for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3089–3098.

[158] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 362–370.

[159] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2007, pp. 1–8.

[160] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, 33, no., pp. 353–367, 2010.

[161] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3000–3009.

[162] Z. Liu, H. Shen, G. Feng, and D. Hu, "Tracking objects using shape context matching," *Neurocomputing*, 83, no., pp. 47–55, 2012.

[163] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[164] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, 60, no., pp. 91–110, 2004.

[165] C. Lu, S. Liu, J. Jia, and C. Tang, "Contour box: Rejecting object proposals without explicit closed contours," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 2021–2029.

[166] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81, Vancouver, BC, Canada, August 24-28, 1981*, 1981, pp. 674–679.

[167] S. J. Luck and M. A. Ford, "On the role of selective attention in visual perception," *Proceedings of the National Academy of Sciences*, 95, no., pp. 825–830, 1998.

[168] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, "Non-local deep features for salient object detection," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 6593–6601.

[169] Y.-F. Ma and H.-J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proceedings of the eleventh ACM international conference on Multimedia*, ACM, 2003, pp. 374–381.

[170] K. Ma, Z. Shu, X. Bai, J. Wang, and D. Samaras, "Docunet: Document image unwarping via a stacked u-net," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no., pp. 4700–4709, 2018.

[171] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[172] S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu, "Segmentation of multiple salient closed contours from real images," *IEEE transactions on pattern analysis and machine intelligence*, 25, no., pp. 433–444, 2003.

[173] S. Maji, T. Hazan, and T. S. Jaakkola, "Active boundary annotation using random MAP perturbations," in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, 2014, pp. 604–613.

[174] B. Marcotegui, P. L. Correia, F. Marqués, R. Mech, R. Rosa, M. Wollborn, and F. Zanoguera, "A video object generation tool allowing friendly user interaction," in *Proceedings of the 1999 International Conference on Image Processing, ICIP '99, Kobe, Japan, October 24-28, 1999*, 1999, pp. 391–395.

[175] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps," in *CVPR*, 2014, pp. 248–255.

[176] G. Máttyus, W. Luo, and R. Urtasun, "Deeproadmapper: Extracting road topology from aerial images."

[177] K. McGuinness and N. E. O'Connor, "Toward automated evaluation of interactive segmentation," *Computer Vision and Image Understanding*, 115, no., pp. 868–884, 2011.

[178] S. A. McMains and S. Kastner, "Visual attention," in *Encyclopedia of Neuroscience*, M. D. Binder, N. Hirokawa, and U. Windhorst, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 4296–4302.

[179] R. Milanese, "Detecting salient regions in an image: From biological evidence to computer implementation," *Ph. D Theses, the University of Geneva*, no., 1993.

[180] Y. Ming, H. Li, and X. He, "Connected contours: A new contour completion model that respects the closure effect," in *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, 2012, pp. 829–836.

[181] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *European Conference on Computer Vision*, Springer, 2010, pp. 210–223.

[182] E. N. Mortensen and W. A. Barrett, "Toboggan-based intelligent scissors with a four-parameter edge model," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, IEEE, vol. 2, 1999, pp. 452–458.

[183] V. Movahedi and J. 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*, IEEE, 2010, pp. 49–56.

[184] V. Movahedi and J. H. Elder, "Combining local and global cues for closed contour extraction," in *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.

[185] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications on pure and applied mathematics*, 42, no., pp. 577–685, 1989.

[186] V. Murino, C. S. Regazzoni, and G. L. Foresti, "Grouping as a searching process for minimum-energy configurations of labelled random fields," *Computer Vision and Image Understanding*, 64, no., pp. 157–174, 1996.

[187] G. Nagendar, D. Singh, V. N. Balasubramanian, and C. V. Jawahar, "Neuro-iou: Learning a surrogate loss for semantic segmentation," in *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, 2018, p. 278.

[188] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*, Springer, 2016, pp. 483–499.

[189] H.-C. Nothdurft, "Salience of feature contrast," in *Neurobiology of attention*, Elsevier, 2005, pp. 233–239.

[190] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, 79, no., pp. 12–49, 1988.

[191] S. E. Palmer, *Vision science: Photons to phenomenology*. MIT press, 1999.

[192] J. Pan, E. Sayrol, X. Giro-i-Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 598–606.

[193] D. Parkhurst, K. Law, and E. Niebur, "Modeling the role of salience in the allocation of overt visual attention," *Vision research*, 42, no., pp. 107–123, 2002.

[194] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[195] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters—improve semantic segmentation by global convolutional network," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 1743–1751.

[196] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Computer Vision and Pattern Recognition*, 2016.

[197] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, 2012, pp. 733–740.

[198] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, Y. Kompatsiaris, and S. Staab, "M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features," in *Knowledge-Based Intelligent Information and Engineering Systems, 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006, Proceedings, Part III*, 2006, pp. 633–640.

[199] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv:1704.00675*, no., 2017.

[200] F. P. Preparata and M. I. Shamos, *Computational Geometry - An Introduction*, ser. Texts and Monographs in Computer Science. Springer, 1985.

[201] M. Pressigout and É. Marchand, "Real time planar structure tracking for visual servoing: A contour and texture approach," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, 2005, pp. 251–256.

[202] X. Qin, S. He, C. P. Quintero, A. Singh, M. Dehghan, and M. Jagersand, "Real-time salient closed boundary tracking via line segments perceptual grouping," *arXiv preprint arXiv: 1705. 00360*, no., 2017.

[203] X. Qin, S. He, Z. V. Zhang, M. Dehghan, and M. Jägersand, "Bylabel: A boundary based semi-automatic image annotation tool," in *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, 2018, pp. 1804–1813.

[204] X. Qin, S. He, Z. Zhang, M. Dehghan, and M. Jagersand, "Real-time salient closed boundary tracking using perceptual grouping and shape priors," *28th British Machine Vision Conference, BMVC, London, UK, September 4-7*, no., 2017.

[205] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International Symposium on Visual Computing*, Springer, 2016, pp. 234–244.

[206] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. J. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, 29, no., pp. 1470–1475, 2007.

[207] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[208] X. Ren, C. C. Fowlkes, and J. Malik, "Scale-invariant contour completion using conditional random fields," in *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, 2005, pp. 1214–1221.

[209] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Science & Business Media, 2009, vol. 317.

[210] E. T. Rolls and G. Deco, "Attention in natural scenes: Neurophysiological and computational bases," *Neural networks*, 19, no., pp. 1383–1394, 2006.

[211] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[212] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*, Springer, 2006, pp. 430–443.

[213] A. Roy, X. Zhang, N. Wolleb, C. Perez Quenterio, and M. Jagersand, "Tracking benchmark and evaluation for manipulation tasks," in *International Conference on Robotics and Automation*, IEEE, 2015.

[214] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," in *ACM transactions on graphics (TOG)*, ACM, vol. 27, 2008, p. 16.

[215] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, 115, no., pp. 211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[216] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, 77, no., pp. 157–173, 2008.

[217] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, IEEE, vol. 2, 2004, pp. II–II.

[218] C. Saathoff, S. Schenk, and A. Scherp, "Kat: The k-space annotation tool," no., 2008.

[219] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electronic Imaging*, 2016, no., pp. 1–9, 2016.

[220] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, no., pp. 504–525, 2000.

[221] G. G. Scandaroli, M. Meilland, and R. Richa, "Improving ncc-based direct visual tracking," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI*, 2012, pp. 442–455.

[222] T. Schoenemann and D. Cremers, "A combinatorial solution for model-based image segmentation and real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, 32, no., pp. 1153–1164, 2010.

[223] T. Schoenemann, S. Masnou, and D. Cremers, "The elastic ratio: Introducing curvature into ratio-based image segmentation," *IEEE Trans. Image Processing*, 20, no., pp. 2565–2581, 2011.

[224] H. Shen, S. Li, C. Zhu, H. Chang, and J. Zhang, "Moving object detection in aerial video based on spatiotemporal saliency," *Chinese Journal of Aeronautics*, 26, no., pp. 1211–1217, 2013.

[225] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, no., p. 107, 2000.

[226] J. Shi and C. Tomasi, "Good features to track," in *Conference on Computer Vision and Pattern Recognition, CVPR 1994, 21-23 June, 1994, Seattle, WA, USA*, 1994, pp. 593–600.

[227] D. J. Simons and C. F. Chabris, "Gorillas in our midst: Sustained inattentional blindness for dynamic events," *perception*, 28, no., pp. 1059–1074, 1999.

[228] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, no., 2014.

[229] A. Singh, A. Roy, X. Zhang, and M. Jägersand, "Modular decomposition and analysis of registration based trackers," in *13th Conference on Computer and Robot Vision, CRV 2016, Victoria, BC, Canada, June 1-3, 2016*, 2016, pp. 85–92.

[230] Y. Song and J. Shan, "Building extraction from high resolution color imagery based on edge flow driven active contour and jseg," *IAPRSIS*, 37, no., pp. 185–190, 2008.

[231] R. S. Srivatsa and R. V. Babu, "Salient object detection via objectness measure," in *Image Processing (ICIP), 2015 IEEE International Conference on*, IEEE, 2015, pp. 4481–4485.

[232] R. S. Srivatsa and R. V. Babu, "Salient object detection via objectness measure," in *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, 2015, pp. 4481–4485.

[233] J. S. Stahl and S. Wang, "Edge grouping combining boundary and region information," *IEEE Trans. Image Processing*, 16, no., pp. 2590–2606, 2007.

[234] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 1061–1075, 1992.

[235] Y. Sugano, Y. Matsushita, and Y. Sato, "Appearance-based gaze estimation using visual saliency," *IEEE transactions on pattern analysis and machine intelligence*, 35, no., pp. 329–341, 2012.

[236] X. Sun, H. Yao, S. Zhang, and D. Li, "Non-rigid object contour tracking via a novel supervised level set model," *IEEE Trans. Image Processing*, 24, no., pp. 3386–3399, 2015.

[237] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[238] C. Topal and C. Akinlar, "Edge drawing: A combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, 23, no., pp. 862–872, 2012.

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

[240] S. Treue, "Neural correlates of attention in primate visual cortex," *Trends in neurosciences*, 24, no., pp. 295–300, 2001.

[241] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial intelligence*, 78, no., pp. 507–545, 1995.

[242] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, 104, no., pp. 154–171, 2013.

[243] S. K. Ungerleider and L. G, "Mechanisms of visual attention in the human cortex," *Annual review of neuroscience*, 23, no., pp. 315–341, 2000.

[244] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, 32, no., pp. 722–732, 2008.

[245] C. Vondrick, D. J. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation - A set of best practices for high quality, economical video labeling," *International Journal of Computer Vision*, 101, no., pp. 184–204, 2013.

[246] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt, "A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization.," *Psychological bulletin*, 138, no., p. 1172, 2012.

[247] A. Wald, "Sequential tests of statistical hypotheses," *The annals of mathematical statistics*, 16, no., pp. 117–186, 1945.

[248] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural networks*, 19, no., pp. 1395–1407, 2006.

[249] J. Wang, X. Yang, X. Qin, X. Ye, and Q. Qin, "An efficient approach for automatic rectangular building extraction from very high resolution optical satellite imagery," *IEEE Geosci. Remote Sensing Lett.*, 12, no., pp. 487–491, 2015.

[250] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3183–3192.

[251] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, 2017, pp. 136–145.

[252] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Salient object detection with recurrent fully convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no., 2018.

[253] S. Wang, T. Kubota, J. M. Siskind, and J. Wang, "Salient closed boundary extraction with ratio contour," *IEEE Trans. Pattern Anal. Mach. Intell.*, 27, no., pp. 546–561, 2005.

[254] S. Wang and J. M. Siskind, "Image segmentation with ratio cut," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, no., pp. 675–690, 2003.

[255] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 4039–4048.

[256] T. Wang, L. Zhang, S. Wang, H. Lu, G. Yang, X. Ruan, and A. Borji, "Detect globally, refine locally: A novel approach to saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3127–3135.

[257] T. Wang, B. Han, and J. P. Collomosse, "Touchcut: Fast image and video segmentation using single-touch interaction," *Computer Vision and Image Understanding*, 120, no., pp. 14–30, 2014.

[258] W. Wang, Q. Lai, H. Fu, J. Shen, and H. Ling, "Salient object detection in the deep learning era: An in-depth survey," *arXiv preprint arXiv:1904.09146*, no., 2019.

[259] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Transactions on Image Processing*, 27, no., pp. 2368–2378, 2017.

[260] W. Wang, J. Shen, X. Dong, A. Borji, and R. Yang, "Inferring salient objects from human fixations," *IEEE transactions on pattern analysis and machine intelligence*, no., 2019.

[261] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Ieee, vol. 2, 2003, pp. 1398–1402.

[262] Y. Wei, F. Wen, W. Zhu, and J. Sun, "Geodesic saliency using background priors," in *European conference on computer vision*, Springer, 2012, pp. 29–42.

[263] M. Wertheimer, "Untersuchungen zur lehre von der gestalt," *Psychological Research*, 1, no., pp. 47–58, 1922.

[264] Wikipedia contributors, *Principles of grouping — Wikipedia, the free encyclopedia*, [Online; accessed 16-September-2019], 2019. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Principles_of_grouping&oldid=914154204`.

[265] D. J. Williams and M. Shah, "A fast algorithm for active contours," in *Third International Conference on Computer Vision, ICCV 1990. Osaka, Japan, 4-7 December, 1990, Proceedings*, 1990, pp. 592–595.

[266] J. M. Winn, A. Criminisi, and T. P. Minka, "Object categorization by learned universal visual dictionary," in *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, 2005, pp. 1800–1807.

[267] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, 2, no., pp. 37–52, 1987.

[268] J. M. Wolfe, K. R. Cave, and S. L. Franzel, "Guided search: An alternative to the feature integration model for visual search.," *Journal of Experimental Psychology: Human perception and performance*, 15, no., p. 419, 1989.

[269] J. M. Wolfe and T. S. Horowitz, "What attributes guide the deployment of visual attention and how do they do it?" *Nature reviews neuroscience*, 5, no., p. 495, 2004.

[270] R. Wu, M. Feng, W. Guan, D. Wang, H. Lu, and E. Ding, "A mutual learning method for salient object detection with intertwined multi-supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8150–8159.

[271] T. Wu, X. Ding, and S. Wang, "Video tracking using improved chamfer matching and particle filter," in *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, vol. 3, 2007, pp. 169–173.

[272] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3907–3916.

[273] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no., pp. 1101–1113, 1993.

[274] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, IEEE, 2017, pp. 5987–5995.

[275] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[276] C. Xu, J. L. Prince, *et al.*, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on image processing*, 7, no., pp. 359–369, 1998.

[277] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.

[278] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang, "Saliency detection via graph-based manifold ranking," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 3166–3173.

[279] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3166–3173.

[280] J. Yang and Y.-H. Wang, "Towards automatic building extraction: Variational level set model using prior shape knowledge," in *2012 International Conference on Image Analysis and Signal Processing*, IEEE, 2012, pp. 1–6.

[281] J. Yang and M.-H. Yang, "Top-down visual saliency via joint crf and dictionary learning," *IEEE transactions on pattern analysis and machine intelligence*, 39, no., pp. 576–588, 2016.

[282] W. Yang, J. Cai, J. Zheng, and J. Luo, "User-friendly interactive image segmentation through unified combinatorial user inputs," *IEEE Trans. Image Processing*, 19, no., pp. 2470–2479, 2010.

[283] X. Yang, J. Wang, X. Qin, J. Wang, X. Ye, and Q. Qin, "Fast urban aerial image matching based on rectangular building extraction," *IEEE Geoscience and Remote Sensing Magazine*, 3, no., pp. 21–27, 2015.

[284] A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, 26, no., pp. 1531–1536, 2004.

[285] ——, "Object contour tracking using level sets," in *Asian Conference on Computer Vision*, 2004.

[286] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, no., 2015.

[287] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5964–5973.

[288] J. Yuan, "Automatic building extraction in aerial scenes using convolutional networks," *CoRR*, abs/1602.06564, no., 2016.

[289] Y. Zeng, Y. Zhuge, H. Lu, L. Zhang, M. Qian, and Y. Yu, "Multi-source weak supervision for saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6074–6083.

[290] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin, "A shape-preserving approach to image resizing," in *Computer Graphics Forum*, Wiley Online Library, vol. 28, 2009, pp. 1897–1906.

[291] J. Zhang, "Visual saliency computation for image analysis," PhD thesis, Boston University, 2016.

[292] J. Zhang, S. Ma, M. Sameki, S. Sclaroff, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech, "Salient object subitizing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4045–4054.

[293] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 153–160.

[294] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1404–1412.

[295] ——, "Unconstrained salient object detection via proposal subset optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5733–5742.

[296] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch, "Minimum barrier salient object detection at 80 fps," in *IEEE International Conference on Computer Vision(ICCV)*, 2015.

[297] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1741–1750.

[298] L. Zhang, J. Zhang, Z. Lin, H. Lu, and Y. He, "Capsal: Leveraging captioning to boost semantics for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6024–6033.

[299] P. Zhang, W. Liu, H. Lu, and C. Shen, "Salient object detection by lossless feature reflection," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 2018, pp. 1149–1155.

[300] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 202–211.

[301] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 212–221.

[302] X. Zhang, A. Singh, and M. Jägersand, "RKLT: 8 DOF real-time robust video tracking combing coarse ransac features and accurate fast template registration," in *12th Conference on Computer and Robot Vision, CRV 2015, Halifax, NS, Canada, June 3-5, 2015*, 2015, pp. 70–77.

[303] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 714–722.

[304] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, no., 2018.

[305] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[306] K. Zhao, S. Gao, Q. Hou, D. Li, and M. Cheng, "Optimizing the f-measure for threshold-free salient object detection," *CoRR*, abs/1805.07567, no., 2018. arXiv: `1805.07567`. [Online]. Available: `http://arxiv.org/abs/1805.07567`.

[307] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[308] Y. Zheng, B. Jeon, D. Xu, Q. Wu, and H. Zhang, "Image segmentation by generalized hierarchical fuzzy c-means algorithm," *Journal of Intelligent & Fuzzy Systems*, 28, no., pp. 961–973, 2015.

[309] G. Zhu, Q. Zeng, and C. Wang, "Efficient edge-based object tracking," *Pattern Recognition*, 39, no., pp. 2223–2226, 2006.

[310] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2814–2821.

[311] Y. Zhuge, Y. Zeng, and H. Lu, "Deep embedding features for salient object detection," in *AAAI*, vol. 33, 2019, pp. 9340–9347.

[312] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European conference on computer vision*, Springer, 2014, pp. 391–405.