Saliency Methods for Object Discovery Based on Image and Depth Segmentation

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Germán Martín García
aus
Madrid

Bonn, 2015

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Erstgutachter: Prof. (em.) Dr. Armin B. Cremers, Bonn
Zweitgutachter: PD Dr. Simone Frintrop, Bonn
Tag der Promotion: 28.01.2016
Erscheinungsjahr: 2016

Zusammenfassung

Object Discovery ist ein neues Paradigma in Computer Vision und Robotic Vision, bei dem die Interpretation eines Bildes durch die Vergabe von Candidate Regions beginnt, die möglicherweiser Objekten im Bild entsprechen. Im weiteren Verlauf können diese Kandidaten mit Objekterkennungsmodulen oder durch Interaktion von Robotern validiert werden. In dieser Arbeit schlagen wir eine neue Methode zur Object Discovery vor, bei der einzelne Bilder verwendet werden, mit dem Ziel eine höhere Wiedererkennungsrate mit weniger Objektkandidaten als State-of-the-Art Methoden zu erreichen. Unser Ansatz benutzt Saliency, um den Ort und die Maße der Objekte abzuschätzen und Segmentierung, um die Konturen präzise abzugrenzen. Wir vergleichen vier verschiedene Methoden, die auf Farbe, Tiefendaten sowie einer frühen und späten Fusion dieser Daten basieren und kommen zum Schluss dass die späte Fusion die erfolgreichere ist. Die Sortierung der Objektkandidaten erfolgt durch eine neue Strategie, die auf einer Kombination von Merkmalen wie dreidimensionaler Konvexität und Saliency beruht. Wir evaluieren und vergleichen unsere Methode mit anderen modernen Ansätzen, indem wir überladene, schwierige Sequenzen aus der Wirklichkeit nutzen die aus drei verschiedenen öffentlichen Datenbanken stammen. Die Ergebnisse beweisen, dass unsere Methode die anderen konsistent übertrifft. Im zweiten Teil der Arbeit konzentrieren wir uns auf Sequenzen von Bildern. Unser Ziel hierbei ist es, so wenige Kandidaten pro Bild wie nötig zu generieren und dabei so viele Objekte wie möglich durch die Sequenz hindurch zu erkennen. Um dieses Ziel zu erreichen, erweitern wir unsere Methode mit einem sogenannten Spatial Inhibition of Return Mechanismus, bei dem die Objektkandidaten unterdrückt werden, die den bereits in der Vergangenheit generierten Objekten entsprechen. Die Herausforderung hierbei ist es, dies so auszuführen, dass es konsistent mit einem Perspektivwechsel ist, weshalb wir den Inhibition of Return Mechanismus auf räumlichen Koordinaten fundieren. Im letzten Teil dieser Arbeit wird die Anwendung unserer Object Discovery Methode zur Salient Object Segmentation präsentiert. Auch hier zeigen die Ergebnisse, dass unsere vergleichbare Ergebnisse zu anderen Methoden, die dem aktuellen Stand der Technik entsprechen, erreicht.

Germán Martín García

Short Curriculum Vitae

Education and employment

Feb. 2012 - 2015 **Research Assistant** at the Cognitive Computer Vision Group (PD Dr. Simone Frintrop), at the Institute of Computer Science III (Prof. em. Dr. Armin B. Cremers - Prof. Dr. Stefan Wrobel). Rheinische Friedrich-Wilhelms-Universität Bonn.

July 2011 - Dec. 2011 **Student Assistant** at the Intelligent Vision Systems Group, Institute of Computer Science III (Prof. em. Dr. Armin B. Cremers). Rheinische Friedrich-Wilhelms-Universität Bonn.

Apr. 2010 - March 2011 **Student Assistant** at the Autonomous Intelligent Systems Group, Institute of Computer Science VI (Prof. Dr. Sven Behnke). Rheinische Friedrich-Wilhelms-Universität Bonn.

Sept. 2009 - Feb. 2012 **Master in Computer Science** at the Rheinische Friedrich-Wilhelms-Universität Bonn with focus on Intelligent Systems. Average grade 1.2 (very good).

Sept. 2008 - Sept. 2009 **Software Engineer** at Deimos-Space in Madrid (Spain).

July 2007 - Jan. 2008 **Internship** at Telefónica in Madrid (Spain).

Sept. 2002 - July 2008 **Diploma in Computer Science** at the Universidad Autónoma de Madrid (Spain).

Abstract

Object discovery is a recent paradigm in computer and robotic vision where the process of interpreting an image starts by proposing a set of candidate regions that potentially correspond to objects; these candidates can be validated later on by object recognition modules or by robot interaction. In this thesis, we propose a novel method for object discovery that works on single RGB-D images and aims at achieving higher recall than current state-of-the-art methods with fewer candidates. Our approach uses saliency as a cue to roughly estimate the location and extent of the objects, and segmentation processes in order to identify the candidates' precise boundaries. We investigate the performance of four different segmentation methods based on colour, depth, an early and a late fusion of colour and depth, and conclude that the late fusion is the most successful. The object candidates are sorted according to a novel ranking strategy based on a combination of features such as 3D convexity and saliency. We evaluate our method and compare it to other state-of-the-art approaches in object discovery on challenging real world sequences from three different public datasets containing a high degree of clutter. The results show that our approach consistently outperforms the other methods. In the second part of this thesis, we turn to streams of images. Here, our goal is to generate as few object candidates per frame as necessary in order to find as many objects as possible throughout the sequence. Therefore, we propose to extend our object discovery system with a so called spatial inhibition of return mechanism to inhibit object candidates that correspond to objects that have already been generated in the past. The challenge here is to inhibit the candidates consistently with viewpoint change, and therefore, we root our inhibition of return mechanism in 3D spatial coordinates. In the final part of this thesis we show an application of our object discovery method to the task of salient object segmentation. The results show that our method achieves state-of-the-art performance.

Acknowledgements

I came to Bonn in the Autumn of 2009 willing to learn new exciting things that would prevent me from getting bored at work. There was a person that incommensurably helped me find my way in the years before I took that decision. I will always be grateful to Concha and to my family for their support during that time.

I would like to especially thank Prof. Armin B. Cremers for supervising this thesis and for his wise feedback throughout the process. I'm deeply grateful to Simone Frintrop for giving me the chance to do a PhD in the first place, for sharing my excitement with the work we did and for being a great boss. Thanks a lot to the colleagues with whom we cooperated in the past two years, from whom I learnt so much: Esther Horbert and Bastian Leibe, and Ekaterina Potapova. Many thanks to all my colleagues for the fruitful discussions and support during the last three years, especially to Jens Behley, Volker Steinhage, Dominik A. Klein, Stefan Mehner, Jürgen Gall, Helmut Grohne and Alexander Richard. And many many thanks to all the students that worked in our group for their valuable work, especially Mircea Pavel, Thomas Werner and Johannes Teutrine.

I am mostly grateful to Sophia for her love and care, and to my friends Mohammed, Manus, Khaled and Aljosa. And finally to the city of Bonn for the wonderful past 6 years of my life!

Contents

1. Introduction

Localising and recognising the objects in the environment is an essential ability for any mobile vision system that needs to understand the world around it and interact with it —in this general category of mobile vision systems we include robots, wearable devices, and any computational system that uses a camera as its primary means for obtaining information about the world. In the field of computer vision, this is known as object detection: given an image, an object detector has to localise the instances of an object of a particular class. A common way to solve this task was to train object detectors for several categories (cars, faces, persons, trees, etc.) and then scan the image with each detector at each possible scale and location, as it was successfully done by [Viola and Jones, 2004] to detect faces in images and quickly became a standard practise in object detection —see, for example, [Dalal and Triggs, 2005, Harzallah et al., 2009, Felzenszwalb et al., 2010]. However, such an exhaustive search does not scale well as soon as the number of object classes grows. Particularly, if the scale and/or the aspect ratio of the objects is not known in advance, the number of windows to be evaluated can be in the order of 10^6 to 10^7 [Hosang et al., 2014, Hosang et al., 2015]. A recent trend in computer vision has been, instead of evaluating potentially millions of image subwindows for each object class, to generate a smaller set of so called set of object proposals —which depending on the method might be in the order of thousands or tens of thousands— and evaluate the object classifiers on them: the works of [Cinbis et al., 2013, Girshick et al., 2014, He et al., 2014] are examples of current object detection methods that rely on a previous object proposal generation step. Fig. 1.1 illustrates the two paradigms.

A parallel path has been followed in the robotic vision community, where several methods have been developed to segment the objects in the scene prior to knowing their category, in what is commonly known as object discovery. Despite being developed in separate scientific communities, the object proposal generation methods and the object discovery methods in robotics aim at the same goal: producing a set of candidate regions in the image that potentially correspond to objects; i.e., detecting the presence of an object before its category is known.

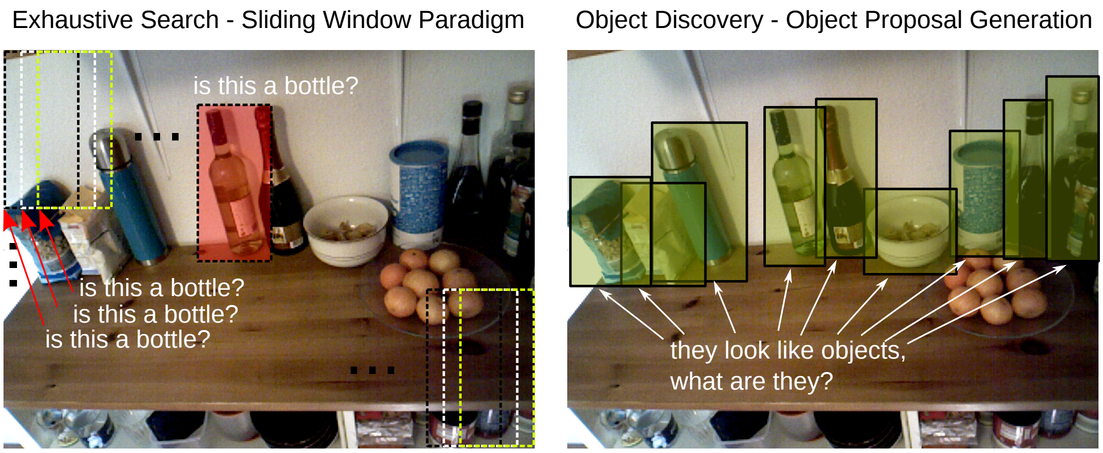


Figure 1.1.: Left: the sliding window/exhaustive search paradigm for object detection. Right: the newer object discovery paradigm.

Objects vs Things But what is typically meant by objects in the computer vision literature? In a recent article, Alexe et al. [Alexe et al., 2012] define objects as

> "standalone things with a well-defined boundary and center, such as cows, cars, and telephones, as opposed to amorphous background stuff, such as sky, grass, and road."

This is a rather practical definition that relates to the one offered by Forsyth, Malik and colleagues [Forsyth et al., 1996]:

> "A material (e.g. skin) is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape. An object (e.g. a ring) has a specific size and shape."

They remark the distinction between materials and objects, or "Stuff" and "Things". They continue by pointing out the different approaches that might lead to the successful recognition of both categories:

> "Indeed, materials with particularly distinctive color or texture (e.g. sky) can be successfully recognized with little or no shape analysis, and objects with particularly distinctive shapes (e.g. telephones) can be recognized using only shape information."

Such a distinction between objects and materials fine grains on the way that an algorithm should accomplish its task of finding what is where in images in a successful manner. The fields of object discovery —in robotics— or object proposal generation —in computer vision— are concerned with finding candidate regions in images that correspond to the objects/things, rather than the materials/stuff. Such a definition of objects coincides with the ones we find in the psychological literature; for [von Hofsten and Spelke, 1985], objects are

> "manipulable units with internal coherence and external boundaries."

Let us now informally define what is meant by object discovery or object proposal generation.

Object Discovery Given an input image I, the goal of object discovery is to generate a set of image regions —which can be pixel precise or bounding boxes— that potentially correspond to the objects that are present. Each of these regions is what we call an object candidate or object proposal. Furthermore, the object candidates need not partition the image nor be mutually exclusive; some can even hierarchically contain others as it happens with real objects: the cap of a bottle is part of the bottle and an object itself. Finally, the generated object candidates should be ranked according to a measure of their objectness so that good candidates can be evaluated first by a recognition algorithm.

Computer and robotic vision algorithms are typically evaluated on benchmarks. Specific benchmarks exist for most vision tasks: optical flow [Baker et al., 2011], object detection [Everingham et al., 2007], etc. In the case of object detection or object discovery, the ground truth needs to be manually annotated, i.e., someone has to go through the images and decide what pixels correspond to the objects. A fundamental difference between the robotic and computer vision communities is the type of data — or benchmarks— used to evaluate the performance of algorithms. In the case of object discovery, computer vision algorithms are evaluated on the available object detection benchmarks [Alexe et al., 2012, Manén et al., 2013]. However, the kind of images that appear in those benchmarks does not make them ideal for testing vision methods for robotics applications: often the images include one or a few objects that occupy most of the image; these are situations that a service robot will rarely encounter, instead, it will often have to deal with cluttered scenes where many objects are present. Object discovery approaches in the robotic vision community are typically tested in indoor scenarios [Karpathy et al., 2013, Potapova et al., 2014] where the robots are expected to operate on; however, no common realistic challenging benchmark exists for this task. Part of the work that was carried out in this thesis was to provide a common benchmark for object discovery consisting of sequences recorded in realistic indoor scenes, where the present objects were annotated consistently on regular intervals.

Detecting the objects in a scene is a scientific challenge per se, but it also has many applications in technical systems, many of which we might not know yet, because the methods are not mature enough. Such methods are essential for service robots to interpret their environment, since they are expected to perform their tasks autonomously: cleaning up a house, preparing lunch, etc. The car industry is strongly interested in automatically finding pedestrians, cars and other objects that might populate the environment of a car using cameras. Examples are the Image Understanding Group of Daimler[1] or the Cognitive Systems & Representation group at the Honda Research Institute Europe.[2] Up to now, laser methods have been a common practise (see the self driving car of Google [Guizzo, 2011]). However, images provide a very rich source of information, and furthermore, they are passive sensors far cheaper than the standard laser ones. One critical requirement for such computer vision systems is that they are fast. Therefore, the use of object proposal generation methods can be a boost in the speed of the whole detection pipeline.

In this thesis, we propose a novel method for object discovery that aims at finding the objects in challenging realistic images, such as those that would be encountered by robots in their *everyday life*. The core of our method is an attention system that helps us localising the objects, and the use of segmentation to find their precise boundaries. Our goal will be to recall more objects than current state-of-the-art approaches with fewer candidates in realistic cluttered scenes. In the second part of this thesis, we move from single images to sequences, and propose a mechanism —inspired by findings from the cognitive science— for establishing correspondences between object candidates over frames; this will let us find most of the objects in the sequence with a very small number of candidates per frame by inhibiting candidates that have already been produced in the past. In the last part of this thesis we show an application of the object discovery algorithm for the task of salient object segmentation.

1.1. Contributions

This work was done in the context of the project “Situated Vision to Perceive Object Shape and Affordances” funded by the Deutsche Forschungsgemeinschaft (German Research Foundation). The project was carried out in cooperation with three other research institutes with whom we did two successful cooperations [Martín García et al., 2015b, Horbert et al., 2015]: the group of Bastian Leibe at RWTH Aachen, Markus Vincze at TU Wien, and Barbara Caputo at IDIAP, Mar-

[1] Group led by Markus Enzweiler `http://www.markus-enzweiler.de/index.html`

[2] `http://www.honda-ri.de/tiki-index.php?page=CognitiveSystemAndRepresentation`

tigny.[3] The purpose of the project was to provide robots with the methods to visually explore a scene, identify the potential objects and classify them according to their affordances. Our contributions in this project are in the areas of object discovery, visual scene exploration and salient object segmentation:

- Object Discovery. We propose a novel method for generating object candidates in RGB-D (colour and depth) images. Our method is based on a combination of saliency computation and segmentation and is especially well suited for complex cluttered scenes. We investigate the use of four different approaches for segmentation: colour, depth, an early fusion of colour and depth, and a late fusion of colour and depth. We also propose a new ranking method based on several objectness features such as 3D convexity. This work is presented in [Martín García et al., 2015b]. Furthermore, we improved the candidate generation process by using a novel multi-scale saliency approach in [Horbert et al., 2015]. A complete benchmark for object discovery was also presented in [Horbert et al., 2015] where several challenging real world sequences were manually annotated to indicate the presence of objects.

- Computational Visual Attention for 3D Scene Exploration. We show how the proposed method for object discovery can be used to sequentially explore a visual scene. We propose a novel method for inhibiting already attended candidates that is rooted in 3D coordinates, which lets us strongly reduce the number of generated object candidates without significantly affecting the recall of the system. The idea of the IOR mechanism was originally published in [Martín García and Frintrop, 2013] and [Martín García et al., 2013]. The system has been improved with respect to the original idea to 1) inhibit object candidates instead of salient regions, and 2) by incorporating the newer object discovery method. This improved version has been submitted to the Cognitive Processing Journal [Martín García et al., 2015a].

- Salient Object Segmentation. We show how our method for object discovery can also be applied for salient object segmentation with minor modifications. The task in salient object segmentation is to detect the most salient object[s] in a given image. The results are published in [Frintrop et al., 2015] and are state-of-the-art in salient object segmentation.

[3] Barbara Caputo changed her affiliation in the course of the project and is now Associate Professor at the University of Rome La Sapienza.

1.2. Outline

The structure of this thesis is the following. We start by introducing some basic concepts and definitions that will appear throughout the text in Chapter 2. We continue with an introduction to the related work in computational visual attention systems and object discovery as well as their cognitive background in Chapter 3. We then follow to our contributions: In Chapter 4, we propose a method for generating visual object candidates based on saliency and segmentation. There, we also propose three different strategies for ranking the object candidates according to their “objectness”. In Chapter 5, we situate the attention system in its 3D environment and develop a mechanism that lets us inhibit the processing of already attended candidates. Finally, in Chapter 6, we show how our visual object candidates can be combined for the task of salient object segmentation. We evaluate all our contributions and show that they outperform several other state-of-the-art methods in object discovery, and that they are state-of-the-art in the task of salient object segmentation. We conclude in Chapter 7 and give a brief overview of future work.

2. Basic Concepts and Definitions

The reader is assumed to have some familiarity with basic image processing —digital filters, convolution— and computer vision ideas and problems —segmentation, object detection. Otherwise, we point to basic text books in these topics: [Gonzalez and Woods, 2006], [Forsyth and Ponce, 2003] or the on-line available [Szeliski, 2010]. In order to make this thesis as much self contained as possible, we will define the main ideas that will appear throughout the text from now on.

In this thesis, we use an ASUS Xtion PRO sensor,[1] which like the famous Kinect camera from Microsoft provides RGB-D —colour and depth— images. An example of a colour and depth image obtained from the ASUS Xtion PRO sensor is shown in Fig. 2.1: note how in the depth map, bright pixels correspond to locations that are far away, and darker pixels correspond to shorter distances. Also note that, because of the physical properties of the sensor, black objects or transparent ones pose a difficulty and depth measurements are often missing for them. Finally, it is also worth mentioning that since the focal lengths of the colour and the depth sensors are different, not every pixel which has a colour value has a depth measurement as well (see the black band around most of the depth map where no measurements are present).

Image A digital image is a two dimensional function $f(x, y)$. The values that f takes can be scalars for the case of gray-scale images or vectors in the case of colour images. The function is indexed by its row and column pixel coordinates x and y which are positive integers. Its origin, $(0, 0)$, is the upper left corner of the image.

Range Image Often referred to as **depth map**, $d(x, y)$, it is a two dimensional array of pixels containing range —depth— measurements. Range measurements are often encoded as either floating point or 16 bit precision unsigned integers.

Logical Operations between Images In the following chapters, we will sometimes refer to logical operations between images. These normally take place between binary images, that is, images whose values are either 1 or 0. Here, an alternative representation for a binary image $f(x, y)$ is to consider it as a set of pixels whose value is 1:

[1] The specifications of the sensor can be found at `https://www.asus.com/Multimedia/Xtion_PRO/`

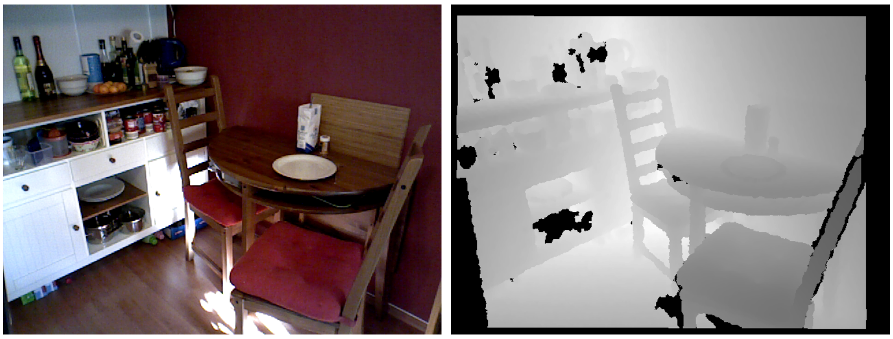


Figure 2.1.: A colour image and its corresponding depth map obtained from the ASUS Xtion PRO sensor.

$I = \{(x, y) \mid f(x, y) = 1\}$, where x and y are the image's row and column coordinates. Then we can talk about common logical operators such as:

- the intersection, or AND operation between two binary images, : $I_1 \cap I_2 = \{(x, y) : (x, y) \in I_1 \wedge (x, y) \in I_2\}$. A graphical example of the AND operation is given in Fig. 2.2;
- the union, or OR operation between two binary images, : $I_1 \cup I_2 = \{(x, y) : (x, y) \in I_1 \vee (x, y) \in I_2\}$.

More details about logical operations between images can be found in the text book of [Gonzalez and Woods, 2006].

Digital Filters The basic image processing mechanism for extracting information from images is through digital filters. They are arrays of a certain size (typically much smaller than the size of the image) which are applied to images by means of convolution. They can be used, e.g., for extracting information about where the edges in an image are, where the blobs are, or where a certain template correlates.

Difference-of-Gaussians (DoG) Filter The filter values of a DoG filter approximate a function that is obtained by subtracting two Gaussians of different standard deviations σ_1 and σ_2:

$$DoG(x, y) = \frac{1}{2\pi\sigma_1^2} \cdot e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} \cdot e^{-\frac{x^2+y^2}{2\sigma_2^2}}. \tag{2.1}$$

An example is shown in Fig. 2.3: the blue and the red curves correspond to Gaussians of standard deviation $\sigma_1 = 3$ and $\sigma_2 = 7$ respectively. The yellow curve is the

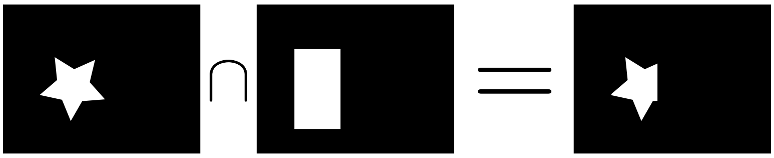


Figure 2.2.: Example of an AND operation between two binary images: black values represent 0 and white values represent 1.

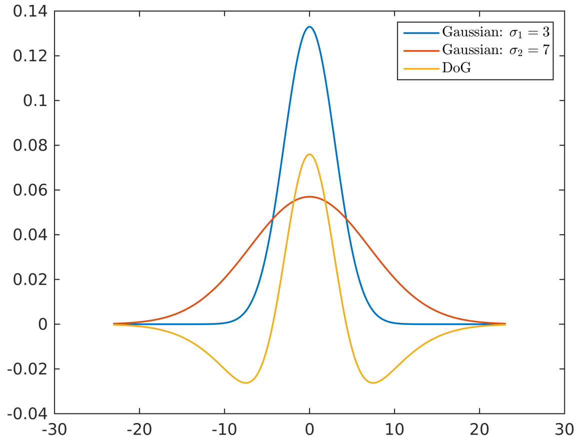


Figure 2.3.: Blue: Gaussian function with standard deviation $\sigma_1 = 3$. Red curve: Gaussian function with standard deviation $\sigma_2 = 7$. Yellow: the difference of the two Gaussians.

corresponding DoG (Eq. 2.1). Convolving an image with a DoG filter tends to highlight regions that contrast with its surround. As we will see in Chapter 4, it is a filter commonly used to compute saliency.

For more details about basic image processing tools and digital filters, the reader can refer to [Gonzalez and Woods, 2006].

Segmentation If not properly specified, the task of segmentation can be ambiguous on whether semantic information is used in the process or not. In the following, we will refer to segmentation algorithms as to what is also known as perceptual grouping: a partition of an image into regions that are perceptually coherent (some examples of such algorithms are [Shi and Malik, 2000, Felzenszwalb and Huttenlocher, 2004]). We will call such regions segments or superpixels.

We refer to the formal definition of segmentation given by [Felzenszwalb and Huttenlocher, 2004]. The authors define a graph where the nodes are the pixels in the image to be segmented, and the edges define neighbour-

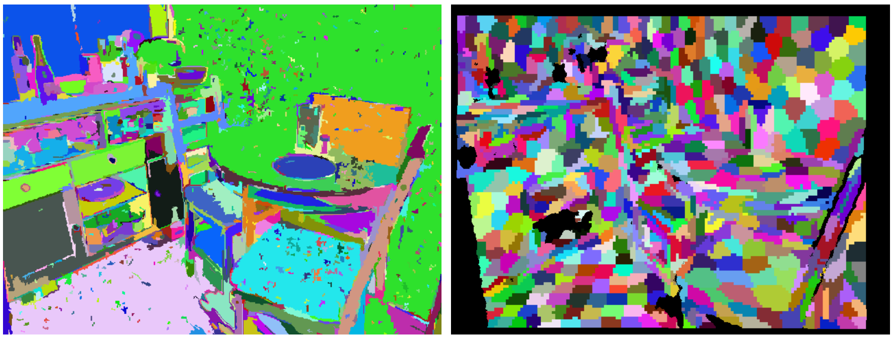


Figure 2.4.: Left: an example of segmentation of the colour image in Fig. 2.1 with the method of [Felzenszwalb and Huttenlocher, 2004]. Right: segmentation of the same image using colour and depth with the method of [Papon et al., 2013].

hood/adjacency relations between them:

> "We take a graph-based approach to segmentation. Let $G = (V, E)$ be an undirected graph with vertices $v_i \in V$, the set of elements to be segmented, and edges $(v_i, v_j) \in E$ corresponding to pairs of neighboring vertices. Each edge $(v_i, v_j) \in E$ has a corresponding weight $w((v_i, v_j))$, which is a non-negative measure of the dissimilarity between neighboring elements v_i and v_j. In the case of image segmentation, the elements in V are pixels and the weight of an edge is some measure of the dissimilarity between the two pixels connected by that edge (e.g., the difference in intensity, color, motion, location or some other local attribute)."

A segmentation of the image would be equivalent to a partition of the graph into connected components:

> "In the graph-based approach, a segmentation S is a partition of V into components such that each component (or region) $C \in S$ corresponds to a connected component in a graph $G' = (V, E')$, where $E' \subseteq E$."

There are segmentation algorithms that operate on pixel difference in intensity or colour [Felzenszwalb and Huttenlocher, 2004], on depth [Richtsfeld et al., 2012], as well as on colour and depth [Papon et al., 2013]. Two examples of image segmentations are shown in Fig. 2.4.

A segmentation of the image is often done in order to reduce the complexity of a given problem: instead of having $N \times M$ pixels, compute a segmentation that produces K superpixels, with $K \ll N \times M$. There are many approaches in various

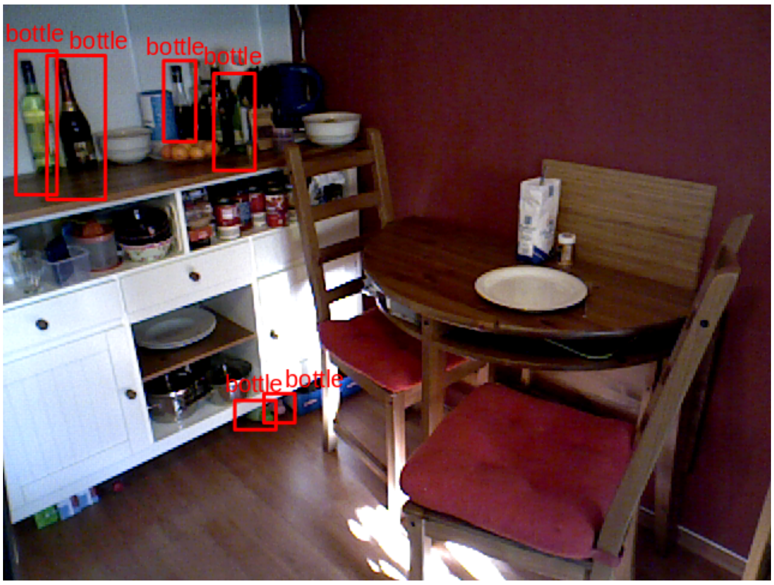


Figure 2.5.: Example of the detection of objects of the class "bottle".

computer vision tasks that use segmentation as a pre-processing step: the object discovery method of [Kootstra and Kragic, 2011], the semantic segmentation method of [Arbeláez et al., 2012] —here the segmentation is semantic, meaning the output segments of the method have a category label attached to them— or the object retrieval method of [Arandjelović and Zisserman, 2011].

Object Detection It is the task of finding the instances of a given object class/category —e.g., car, pedestrian, bicycle— in an image: that means localising them and giving them a category label. The results can be delivered in the form of bounding boxes containing the object class or pixel-precise regions. See [Szeliski, 2010] for more details about object detection algorithms. An example of the detections of objects of the class "bottle" is shown in Fig. 2.5.

Object Proposal Generation or Object Discovery. Given an input image I, the task of object discovery is to generate a set of regions $\{o_i\}$ that correspond to objects, usually called object candidates or object proposals. Each object candidate is thus a set of pixels: $o_i = \{p_j\}$, which are generally connected. No restrictions are applied to the set of object candidates, e.g., they are not required to partition the image or be mutually exclusive: one candidate can contain a part of another candidate. The candidates should be sorted in order of their objectness, so that good candidates can be evaluated first by a recognition algorithm.

Object Proposals and Object Candidates In the rest of this work, we will refer to the object candidates generated by object discovery algorithms as object proposals, object hypotheses or object candidates interchangeably.

3. Related Work

The related work of this thesis spans different fields. We start by introducing the related work in visual attention, pointing out its psychological background as well as the computational models based on them. The second and main part of the related work reviews what has been done in object discovery in computer and robotic vision.

3.1. Related Work in Computational Attention Systems

Many computational attention systems have been built during the last two decades, first for the purpose of modelling and understanding the human visual system [Heinke and Humphreys, 2004], and second to improve technical systems in terms of speed and quality [Frintrop et al., 2010]. The general structure of attention systems is based on psychological models such as the Feature Integration Theory (FIT) [Treisman and Gelade, 1980], which states that visual features are computed in parallel in separate areas of the brain, and by means of focused attention the features are bound together. Two factors are typically distinguished to drive attention: bottom-up and top-down [Frintrop, 2011]. Bottom-up attention is driven by the physical properties of the environment and is typically modelled by saliency —for example, a blinking light can capture our attention. On the contrary, top-down attention is driven by the internal beliefs and goals of the agent —for example, a person looking for his red sock. The first computational model of bottom-up attention, or saliency, was proposed by [Koch and Ullman, 1987]; Koch and Ullman proposed a model where features such as colour and edges are fused into a saliency map that encodes where attention should be allocated. One of the first computational attention system that was implemented based on this model was the renowned Itti-Koch model [Itti et al., 1998], in which feature channels are computed in parallel, image pyramids enable a multi-scale computation, and feature contrasts are computed by Difference-of-Gaussians.

Since the Itti-Koch model, many approaches have been proposed to compute saliency: There are methods based on information theory such as the work of [Bruce and Tsotsos, 2009] and [Klein and Frintrop, 2011]; others rely on machine learning to learn a combination of features to detect salient patterns [Liu et al., 2009].

A recent trend has been to compute saliency on image regions (superpixels) instead of pixels [Yan et al., 2013, Jiang et al., 2013]. Despite the abundance of different methods to compute saliency (see more extensive surveys in [Frintrop et al., 2010, Borji and Itti, 2013]), as it was shown in [Frintrop et al., 2015]: "the underlying method that exists in basically all saliency systems is a contrast computation."

One component of attention systems is the inhibition of return mechanism: a mechanism that inhibits attention from returning to already attended areas. It was discovered by [Posner et al., 1985] as taking place in the human visual system, operating in spatial —not retinotopical— coordinates, and was hypothesised to enable visual exploration. Already [Itti et al., 1998] proposed a computational implementation of IOR that consisted in zeroing values in the saliency map for the regions that had already been the target of attention. Their method however only worked on single images. While IOR is simple on individual images, image sequences introduce the challenge of establishing correspondences between objects over time. In this context, [Backer et al., 2001] performed object-centred IOR by tracking the attended objects. However, their approach operated on simple artificially rendered scenes instead of real world data and on 2D images instead of 3D data as we do. In the work of [Palomino et al., 2011] the authors implemented IOR by visually tracking the objects that are the target of attention.

In Chapter 5, we will introduce an IOR implementation that operates on spatial coordinates —as it was originally discovered by [Posner et al., 1985]— and is object centred: meaning that it is objects that are inhibited by the mechanism.

3.2. Related Work in Object Discovery

Object discovery is the task of finding the objects present in a scene without knowing in advance what their appearance or category will be. Therefore, it is a chicken-and-egg problem: how to look for an object before knowing how it looks like and which features it has?

The idea of first individuating visual regions and then investigating further their visual properties has a parallel in the cognitive science literature in the work of [Pylyshyn, 2001]: Pylyshyn postulates that the human visual system requires a mechanism that visually individuates the elements in the environment before their properties or categories are known. In the following, we review the related work in object discovery in its development in two different —though related— scientific communities: computer vision and robotics.

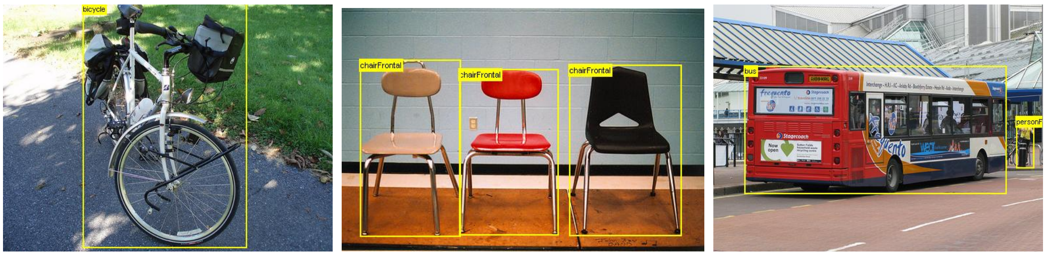


Figure 3.1.: Example images from the Pascal Visual Object Classes Challenge 2012 (`http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/`). On the right, the bounding box is tightly fit to the bus; in the images with the chairs and the bicycle, however, the bounding boxes include much of the background.

3.2.1. Object Discovery in the Computer Vision Community: Object Proposal Generation

The idea of generating object proposals can be traced back to the work of [Malisiewicz and Efros, 2007], when the authors introduced the so called Soup of Segments. As the authors state, up to that point in time, computer vision researchers had largely ignored bottom-up segmentation methods in object detection and recognition. The main paradigm was the sliding window approach of Viola and Jones [Viola and Jones, 2004], where a number in the order of 100 000 to 1 million windows at different locations and scales would scan the image for objects of a given category. Malisiewicz and Efros argued that bottom-up segmentation methods could help to improve the recognition rates by facilitating a more precise fit of the boundaries of objects (or as they said, to improve their spatial support): a bounding box containing an object can include large parts of the background that are going to be part of the training if no precise boundaries are provided. This is where segmentation can be beneficial. Some examples of such situations are shown in Fig. 3.1: the bounding box containing the bus has almost no background; whereas the chairs and the bicycle bounding boxes include a big amount of background pixels.

However —their reasoning continued—, no bottom-up segmentation algorithm can hope to segment out objects “in one piece”, since their texture, colour, or intensity features can vary within the same object: think for example of a person wearing a red sweater and blue jeans; a bottom-up segmentation algorithm would eventually produce a segment for the sweater, one for the jeans, etc. Therefore, the authors propose an approach that uses segments obtained from different segmentation algorithms (also different parameters of the same methods) to determine the regions where to perform object detection. They used three different segmentation methods [Felzenszwalb and Huttenlocher, 2004, Comaniciu and Meer, 2002, Shi and Malik, 2000], each with different parameters and put all the segments together into what they called it a Soup of Segments. Finally, they allowed combinations of up to three neighbouring segments — that were produced by the same segmentation algorithm— to be merged and added to the Soup. They showed that their approach improved the spatial support of objects in object recognition with respect to the sliding window approach of Viola and Jones.

After the original work of [Malisiewicz and Efros, 2007], several other approaches followed. In [Van de Sande et al., 2011], the authors proposed a method for generating object candidates based on a hierarchical segmentation. Their method starts with an over-segmentation using the segmentation algorithm of [Felzenszwalb and Huttenlocher, 2004]. They define a similarity measure between segments, and as the algorithm goes a level up in the hierarchy, every two segments that are most similar according to this measure become merged. The candidates delivered in the end are all the segments across all hierarchy levels; potentially in the order of a thousand.

Two popular approaches are the Objectness measure of [Alexe et al., 2012] and the Randomized Prim Proposals of [Manén et al., 2013]. In [Alexe et al., 2012], the authors propose a method for sampling windows and ranking them using a Naive Bayes framework that combines saliency, color contrast, edge density, and location cues. The method of [Manén et al., 2013] relies on an initial segmentation of the image into superpixels. The algorithm randomly selects superpixels as seeds, and starts on each of them a growing process based on the similarity of the segments. A random component determines when to stop the growing process; this lets the method find objects that might not have homogeneous colour properties.

A very successful approach is the Selective Search of [Uijlings et al., 2013]. It relies on a simple idea and is entirely hand crafted —it does not rely on any machine learning method—, yet the results turned out to be among the best in the comparisons performed in [Hosang et al., 2014]. The core of the idea is to start with an initial over-segmentation of the image, based on which, segments are greedily and iteratively merged according to their similarity. Several segmentation methods [Comaniciu and Meer, 2002, Felzenszwalb and Huttenlocher, 2004], colour spaces and similarity measures are spanned within this hierarchical grouping algorithm to deliver a final set of candidate object locations. The ranking of the proposals is the order in which they were generated in the hierarchical grouping algorithm.

Other popular methods are the category independent object proposals of

[Endres and Hoiem, 2010], the CPMC method [Carreira and Sminchisescu, 2010] or the Geodesic object proposals of [Krähenbühl and Koltun, 2014]. A recent comparison between most of the methods we refer to here has been done in [Hosang et al., 2014] and has been extended with more details in [Hosang et al., 2015].

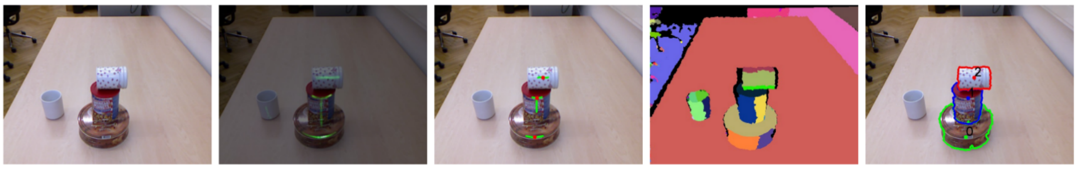
To sum up, what all these methods have in common is their goal of delivering either regions or bounding boxes that can be used by a recognition algorithm to find the objects in an image. Among the common characteristics are that all of them rely purely on visual cues (no use of depth data), they produce a set of proposals in the order of 1000-10 000, and incorporate a method for ranking them according to their objectness. In the following section, we will cover the approaches that have been developed in the robotics scientific community.

3.2.2. Object Discovery in Robotic Vision

Related methods for object discovery in robotic vision have followed a parallel path. Due to the physical limitations of robots, the methods we find here tend to produce a small but precise number of candidates: if a robot needs to get close to the object to examine it, or even interact with it, it is not practical to have a set of 10 000 potential candidates for a given camera frame. Furthermore, these methods tend to rely on depth data (sometimes exclusively [Karpathy et al., 2013]); this modality is seldom used in the computer vision community since the most popular object detection benchmarks only have colour images [Everingham et al., 2010].

One of the first methods we find in this community was proposed by [Mishra and Aloimonos, 2012]. The authors argued that it makes sense for a robot to first segment the potential object and then recognise it, since the recognition module already has the scale, shape and dimensions available. The authors propose a method for segmenting objects that relies on finding attention points —or fixations— that lie in the center of objects, and the use of the border ownership in the contour points (knowing on which side of the contour is the object). Generating fixations on the center of objects was also done in the work of [Kootstra and Kragic, 2011]: their method generates a superpixel segmentation of the image, and fixation points based on a symmetry operator. This is based on the assumption that objects are symmetric, and therefore, the peaks in a symmetry map will lie on the center of objects. The symmetry seeds are used to start a growing process of superpixels based on their similarity. Finally, the authors propose a method to evaluate the objectness of the generated candidates based on several Gestalt principles.

Following these ideas, we find the method [Potapova et al., 2014], which makes use of depth data in order to find symmetry points that fall in the center of objects; see



(a) Original image (b) Saliency Map (c) Attention points (d) Surface patches (e) Segmentation result

Figure 3.2.: Illustration of the method of [Potapova et al., 2014]. Figure from [Potapova et al., 2014].

Fig. 3.2: in image b), the symmetry axes are denoted in green. Around the fixation points, surface patches are clustered to form the final object segmentations/candidates. In contrast to the method of Kootstra [Kootstra and Kragic, 2011], Potapova and colleagues [Potapova et al., 2014] rely on depth in order to compute features such as 3D convexity or symmetry, which are potentially more stable indicators of objectness than their equivalent in colour images. Furthermore, the features are used in the object candidate building process, as opposed to [Kootstra and Kragic, 2011] where they are exclusively used for the candidate ranking.

Other approaches use information about changes over time to segregate objects from background [Herbst et al., 2011] or interact with possible object candidates to determine what is an object [Schiebener et al., 2014]. While these are useful approaches to resolve ambiguities, it is certainly desirable to be able to find objects also without or before interaction, and if possible already from a single view without having to observe the scene over a long time.

In the family of object proposal generation methods we encounter in the computer vision community, a large set of object candidates is generated, and the work of finding those which are reliable objects is left to an object recognition module. In the robotic vision literature, however, a lot of effort has been invested in obtaining a scene segmentation, or a much smaller set of object candidates that with high confidence correspond to actual objects. We think, however, that object recognition has to play an important role in confirming which object hypotheses are actually objects and which not. This could be done automatically or even with human interaction. Recognising the object candidates is not a part of this thesis, but we outline how it could be incorporated in future work in Chapter 7.

4. Generating Object Proposals

In this chapter, we deal with the problem of finding objects in RGB-D scenes without having a priori knowledge of what the objects look like or what class they belong to. Our approach is based on a simple assumption: objects stick out of their environment in some way. This difference can be in colour, intensity, texture, depth, or other features. Such a measure of contrast is what is generally computed by bottom-up attention systems in saliency maps. In our method, salient regions are extracted from the saliency map to estimate the location and extent of objects. Their precise boundaries are obtained by using segmentation in either colour or depth modalities, or both. An overview of our approach is depicted in Fig. 4.1: for a given input image (on the left), the saliency map (middle left) and image segmentation (middle right) are computed. Saliency is used to glue segments together into object candidates (some example candidates are shown on the rightmost image of the figure).

The idea to combine saliency and segmentation was inspired by the psychological work of Rensink [Rensink, 2000]: he argues that in human perception, so-called *proto-objects* are detected by perceptual organization rules, e.g., by segmentation processes that bundle parts of the visual field. These proto-objects are combined by focused attention to form coherent objects. Quoting Rensink: “Attention acts as a hand to grasp proto-objects” [Rensink, 2000]. While in human vision such attention mechanisms consist of bottom-up and top-down parts, top-down information is not always available in technical systems. Thus, we focus on bottom-up attention here, which corresponds to saliency computation. In our approach to object discovery, saliency acts as hand to grasp visual segments.

This chapter is structured as follows. The saliency computation methods are described in Sections 4.1.1 and 4.1.2. We evaluate the saliency system and obtain appropriate parameters for the object discovery task in Section 4.1.3. In Section 4.1.4, we explain how salient regions can be extracted from the saliency map. We cover four segmentation approaches using either colour, depth, or both in Section 4.2. How the segments are bundled together into object candidates with precise boundaries is explained in Section 4.3. Post-processing of the object candidates and three ranking strategies are discussed in Sections 4.4 and 4.5 respectively. Finally, we evaluate our

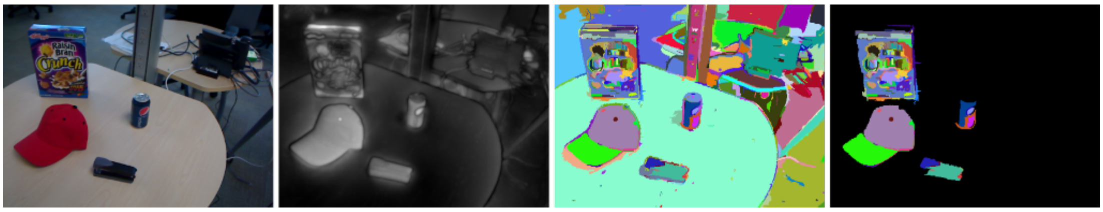


Figure 4.1.: From left to right: original image, saliency map, image segmentation and some of the generated object candidates.

method and compare it to other approaches to object discovery in Section 4.6.

Relevant publications The publications relevant for this Chapter are: [Martín García et al., 2015b] and [Horbert et al., 2015]. In [Martín García et al., 2015b] we propose an object discovery method using a single saliency map, integrate several segmentation methods, and finally compare three different ranking strategies. Our contribution in [Horbert et al., 2015] is an alternative way for computing saliency called multi-scale saliency, that is used in the general object discovery method.

4.1. Locating Candidate Objects: Salient Region Extraction

A key step of our method is the extraction of salient regions from a saliency map. In this section, we first introduce the VOCUS2 approach for computing saliency and briefly evaluate and compare the saliency maps to other saliency systems. Then we explain our approach for extracting salient regions from saliency maps.

4.1.1. Saliency Computation: VOCUS2

Visual saliency is the "distinct subjective perceptual quality which makes some items in the world stand out from their neighbours and immediately grab our attention" [Itti, 2007]. Since the first model of visual saliency from [Koch and Ullman, 1987] computational attention systems that compute saliency have been proposed in great abundance (see [Frintrop et al., 2010] for an extensive survey on the subject). However, as it is said in [Frintrop et al., 2015] computing visual saliency mainly reduces to a measure of contrast between visual regions.

In this section, we describe the VOCUS2 approach to computing saliency maps [Frintrop et al., 2015]. The saliency computation is not a contribution of this thesis, but rather a method we make use of.

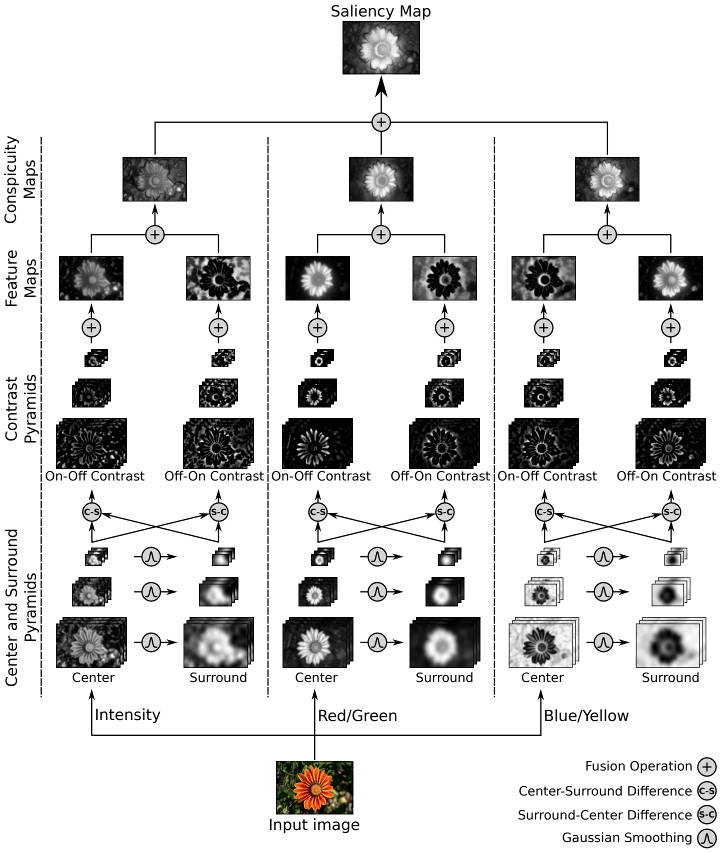


Figure 4.2.: General structure of the VOCUS2 saliency system. Figure from [Frintrop et al., 2015]

Fig. 4.2 gives an overview of the VOCUS2 saliency system. It follows the typical architecture of saliency systems: it contains several feature channels where contrast is computed at different scales of observation, and such contrasts are fused across scales and features into a saliency map. The approach has, however, several peculiarities that make it different from the classical model proposed by Itti and colleagues [Itti et al., 1998]. We outline their main differences in Table 4.1 and will explain them in more detail in the following paragraphs.

Feature Channels

VOCUS2 has one intensity channel (I), and two colour channels (RG and BY). The orientation channel is not used since it tends to highlight borders of the objects. Here, we are interested in having “valleys” at the border of objects, *i.e.*, we want to have black borders around the objects to ease the extraction of salient regions. The intensity

	iNVT	VOCUS2
Features	intensity (I), color (C), orientation (O)	intensity (I), color (C)
Pyramid structure	one pyramid	**twin pyramids** (main difference)
	one scale per layer	multiple scales per layer
Feature fusion	down-sampling	up-sampling
	weighting by uniqueness	arithmetic mean
	fuse color channels first, then intensity	treat all 3 channels equally

Table 4.1.: Main differences between iNVT [Itti et al., 1998] and the VOCUS2 system [Frintrop et al., 2015]. Table from [Frintrop et al., 2015].

channel I is obtained as

$$I = {}^{(R+G+B)}\!/_{3}. \tag{4.1}$$

The two colour channels are built as the two dimensions of an opponent colour space (as in [Klein and Frintrop, 2012]), which is based on the opponent theory of human perception [Hurvich and Jameson, 1957]. Thus, we have one channel for Red/Green:

$$RG = R - G, \tag{4.2}$$

and the other for Blue/Yellow:

$$BY = B - {}^{(R+G)}\!/_{2}. \tag{4.3}$$

Each of these colour maps can be visualised with grayscale values. For example, in the RG colour map, white pixels correspond to red colours and black pixels to green.

Contrast Computation

The core of a saliency system is the way it computes contrast. The method proposed in the Itti model was to use a Difference-of-Gaussians filter (DoG). For pixel coordinates x and y, and standard deviations σ_1 and σ_2 (more details in Chapter 2), the DoG is defined as

$$DoG(x,y) = \frac{1}{2\pi\sigma_1^2} \cdot e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} \cdot e^{-\frac{x^2+y^2}{2\sigma_2^2}}. \tag{4.4}$$

The DoG is known to emulate the computations performed by the retinal ganglion cells of the human visual system [Rodieck, 1965]. Itti's implementation subtracted non-consecutive layers of the Gaussian pyramid to achieve this. However, this restricts the possible center-surround ratios to the powers of 2 that are involved in the construction of the pyramid. How is this done in VOCUS2? To answer this question let us describe the scale-space structure used in VOCUS2.

Scale-space Structure VOCUS2 computes a twin pyramid structure for each of the feature channels: a center and a surround pyramid. Gaussian pyramids are often computed by iteratively performing two operations: Gaussian smoothing of the input image at the current level of the pyramid (using a Gaussian of $\sigma = 2$), plus a subsampling operation where the resolution of the image is halved. In this representation, the effective smoothing factor, σ_e, that is achieved at each level of the pyramid, also called octave (o), is $\sigma_e = 2^o$: e.g., at octave 1 ($o = 1$), the effective smoothing factor achieved is $\sigma = 2$. A more sophisticated approach to computing such a scale-space representation is the one used in [Lowe, 2004], where each octave contains a number of intermediate scales (S). In this representation, the effective smoothing factor that is achieved at scale s and octave o is $\sigma_e = 2^{(o+\frac{s}{S})}$.

This second approach is the one used in VOCUS2 for constructing the center and surround pyramids and is visualised in Fig. 4.3: the input to the process of constructing the scale-space representation is the original image pre-convolved with the desired smoothing factor (σ_c for the center pyramid or σ_s for the surround pyramid). The effective smoothing factor required at each scale and octave is $\sigma_e[s, o] = 2^{(o+\frac{s}{S})}$; these smoothing factors can be achieved by incrementally smoothing from one scale to the next by using the fact that $\sigma_e[s+1, o+1] = \sqrt{\sigma_e[s, o]^2 + \sigma_i^2}$. That means, that at each step we can find the appropriate σ_i with which we have to smooth the scale s in order to obtain the next one, $s + 1$. After S of such steps, the down-sampling operation is performed.

The same principle of incremental smoothing is used when building the center and surround pyramids: if the required center-surround ratio is, for example, 3-11, the center pyramid is built with a Gaussian of $\sigma_c = 3$ pixels and the surround pyramid is constructed with a Gaussian of $\sigma_s = 11$ pixels. The surround pyramid is actually computed from the center pyramid by convolving each level of the center pyramid with a Gaussian of standard deviation $\sigma_i = \sqrt{\sigma_s^2 - \sigma_c^2}$ (see Fig. 4.2).

Using this twin pyramid structure, the contrast computation is done by simply subtracting the surround pyramid from the center one for on-off contrast, and vice versa for off-on contrast. The result of the contrast computation step is two **contrast pyramids** for each of the three feature channels: intensity, red/green and blue/yellow —see Fig. 4.2. The contrast maps are denoted $X^f_{i,j} = C^f_{i,j} - S^f_{i,j}$ for on-off contrasts and $Y^f_{i,j} = S^f_{i,j} - C^f_{i,j}$ for off-on, with $f \in \{I, RG, BY\}$ and i and j the octave and scale indexes respectively. Negative values in the contrast maps are set to zero.

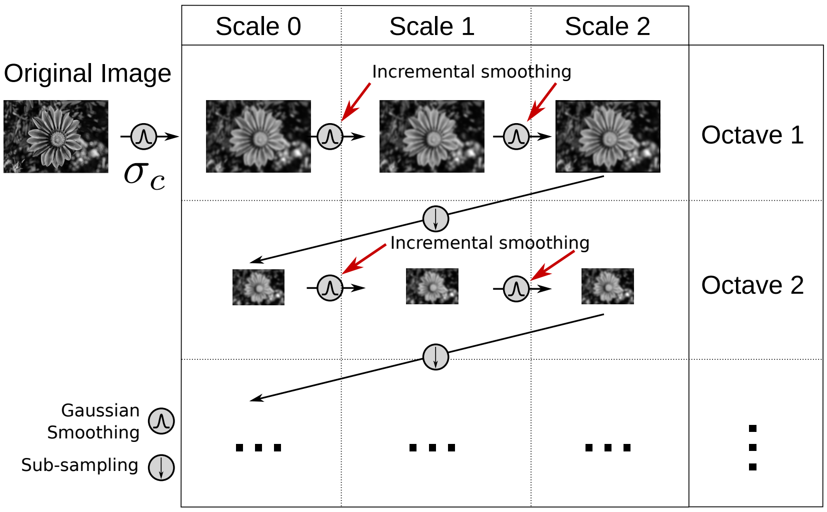


Figure 4.3.: Computation of a Gaussian center pyramid for 2 scales.

Feature Fusion

The next step is to obtain feature maps by doing across scale addition of the contrast maps. Thus, for each contrast pyramid, each layer is up-sampled to the size of the lowest layer of the pyramid and all the contrast maps are summed up into one single feature map. This results in one on-off, and one off-on feature map for each of the feature channels, thus, six feature maps in total.

Conspicuity and Saliency Maps

To obtain the conspicuity maps, the feature maps are fused into a single map. This can be done by different arithmetic operations such as summing, simple averaging, geometric averaging, etc. In VOCUS2, this is done by default by simply averaging the on-off and off-on maps into one conspicuity map per feature channel. The final saliency map is computed by averaging the three conspicuity maps into a single saliency map.

4.1.2. Multi-scale Saliency Computation

In addition to the VOCUS2 method for computing saliency, we propose a method that considers the different octaves of the pyramid independently, and therefore, computes octave-specific saliency maps. That is, instead of having a single saliency map as output, we have, in this approach, one saliency map per octave in the scale space. This is motivated by two ideas: first, since objects of different sizes will achieve the strongest response at different octaves, this allows us detecting nested candidates. For example, if a fruit lies on a bowl, one octave will highlight the fruit and another one the bowl,

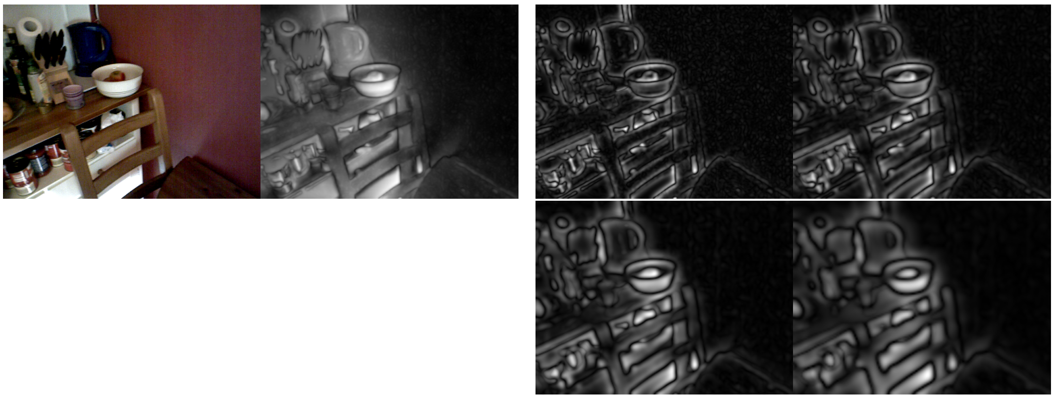


Figure 4.4.: Left: the original image and the saliency map. Right: the octave-specific saliency maps; from left to right and top to bottom, octaves 1, 2, 3 and 4 respectively.

resulting in candidates for both objects. Second, the response of each visual structure will be more precise at the saliency level that best fits with its size: large structures will show up clearer at higher levels.

In this multi-scale approach to compute saliency, the twin pyramids and the contrast maps are computed as before: $X^f_{i,j} = C^f_{i,j} - S^f_{i,j}$ for on-off contrasts and $Y^f_{i,j} = S^f_{i,j} - C^f_{i,j}$ for off-on, where negative values are set to zero, $f \in \{I, RG, BY\}$ and i and j the octave and scale indexes respectively. The difference is now, each octave i is kept separately: We first sum up the contrasts obtained at each octave into the feature maps $F^f_{i,on-off} = \bigoplus_j X^f_{i,j}$ and $F^f_{i,off-on} = \bigoplus_j Y^f_{i,j}$, where $\bigoplus$ denotes across scale addition, which in this case implies summing up the maps for a given octave i and upscaling to the original resolution. Now we can obtain each octave specific conspicuity map as $\mathcal{C}^f_i = g(F^f_{i,on-off}, F^f_{i,off-on})$, where g computes the average between the two feature maps.

Finally, the octave-specific saliency map Sal_i is obtained as $Sal_i = h(\mathcal{C}^I, \mathcal{C}^{RG}, \mathcal{C}^{BY})$, where h is simply the arithmetic mean. We keep g and h as different functions because, in principle, the fusion operations can be performed in different ways: using a max operator, or using a non-linear function that favours maps with few peaks as in [Frintrop, 2006].

A visual example is shown in Fig. 4.4: the four saliency maps on the right side are the octave-specific saliency maps for each of the octaves (1, 2, 3 and 4). It can be seen that the saliency map of octave 3 is where the apple has the highest response, as well as where its boundaries are most clearly separated. On the single saliency map (left side of the figure), the apple does not achieve such a clear separation.

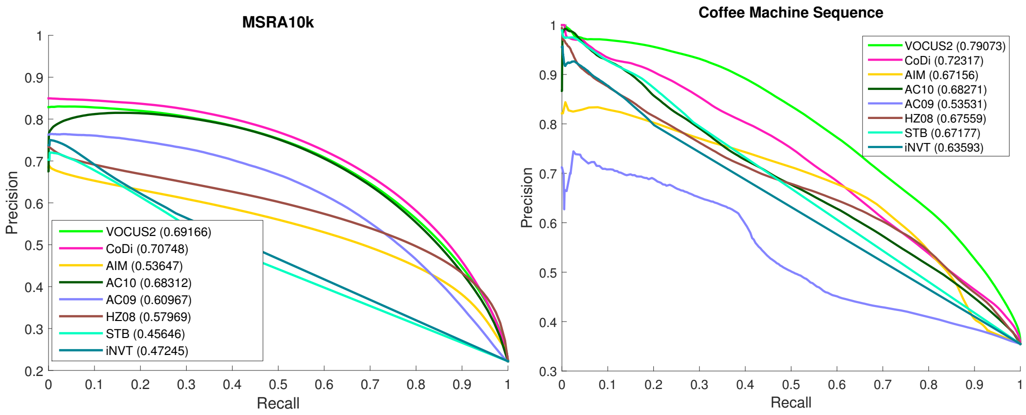


Figure 4.5.: Precision-recall curves for the MSRA-10k dataset (left) and the Coffee Machine Sequence (right). Figure adapted from [Frintrop et al., 2015].

4.1.3. Saliency System Evaluation

Before we continue with the description of the object discovery method, we want to justify the choice of the saliency system based on the results obtained in the task of salient object segmentation. Salient object segmentation is the task of segmenting the most salient object[s] in an image, and performing such task is a typical way of evaluating saliency systems. A popular benchmark is the MSRA 10k [Cheng et al., 2015]. It contains 10.000 images where the most salient object has been manually annotated by users.

We compare VOCUS2 with the following saliency systems: Itti's iNVT [Itti et al., 1998], the SaliencyToolbox (STB) [Walther and Koch, 2006], HZ08 [Hou and Zhang, 2008], AIM [Bruce and Tsotsos, 2009], AC09 [Achanta et al., 2009], AC10 [Achanta and Süsstrunk, 2010] and CoDi [Klein and Frintrop, 2012].

To evaluate the saliency maps we followed the method of [Achanta et al., 2009]: saliency maps are thresholded with an increasing $k \in [0, 255]$. This results in binarised maps which are intersected with the ground truth. The pixels in the intersection can be used to obtain precision and recall values: precision is the number of correct pixels divided by the number of pixels in the thresholded saliency map. On the other hand, recall is the number of correct pixels divided by the number of pixels in the ground truth.

The results are shown in the left plot of Fig. 4.5. The precision-recall curve shows that VOCUS2 obtains a similar curve as CoDi and outperforms the rest of the methods. Note, however, that in this evaluation we omitted the comparison with saliency methods that make use of segmentation processes. In Chapter 6, we propose a method for improving the saliency maps that makes use of segmentation, and show an extensive

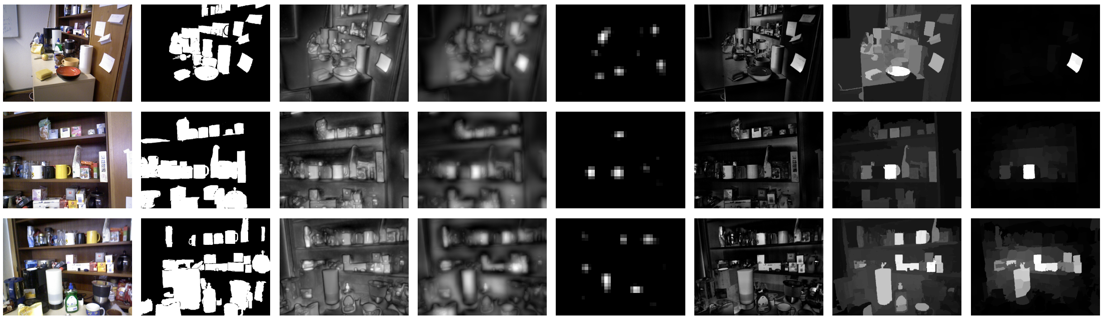


Figure 4.6.: Example results obtained on the CMS dataset. From left to right: original image, ground truth, saliency map for VOCUS2, CoDi, iNVT, AC10, HSaliency, Yan13. Figure from [Frintrop et al., 2015].

evaluation on the most popular benchmarks and methods on salient object segmentation.

Parameter Tuning for Object Discovery

We want to obtain the right set of parameters for the task at hand, which in our case is object discovery. For this, we tuned the saliency system parameters using the Coffee Machine Sequence (CMS), which appeared first in [Martín García and Frintrop, 2013]. The CMS is a challenging RGB-D sequence for object discovery which lasts for 436 frames and has manually annotated ground truth for every 30th frame. It has much clutter, a total of 80 distinct objects appearing throughout the sequence and up to 48 objects per frame.

We evaluated the performance of VOCUS2 on this sequence for different parameter sets, and compared the results to other saliency systems. The optimal parameters we obtained where the following: octaves from 0 to 4, 2 scales, a center σ_c of 2 pixels, and a surround σ_s of 6 pixels, thus, a center-surround ratio of 2:6. This parameter set will be used in the following evaluations of the object discovery system.

The saliency maps obtained for some frames of the CMS for all the saliency methods evaluated are shown in Fig. 4.6. The corresponding precision-recall curves are shown on the right plot of Fig. 4.5: it can be seen that VOCUS2 performs clearly better than the other saliency systems.

4.1.4. Salient Region Extraction

Once we have the saliency map, our next step is to obtain salient regions from it. Obtaining such regions by means of thresholding is difficult. In the case of a binary threshold one has to determine at which level to do it. However, it is rather the relative differences between pixel values what determines the different regions. If one decides then for adaptive thresholding, which considers the relative differences between the pixels' saliencies and not their absolute values, one still has to determine the size of the kernel that defines the neighbourhood of each pixel [Frintrop et al., 2014].

Instead, we propose a two step approach for obtaining salient regions. The first step is to find the local maxima $L = \{l_1, ..., l_n\}$ in the saliency map $Sal(x, y)$. In the ideal case, the peaks in the saliency map correspond to the centre of objects, where the highest saliency is reached.

The second step is to determine the salient regions by doing region growing [Adams and Bischof, 1994] seeded on the local maxima. Seeded region growing starts at the pixel seed and recursively investigates all the neighbour pixels. For every candidate pixel $p = (x_p, y_p)$, it computes whether $Sal(x_l, y_l) \geq Sal(x_p, y_p) \geq Sal(x_l, y_l) \times t$, with $0 < t \leq 1$. By running this procedure for two different values of t (we chose 0.3 and 0.4), we obtain two salient regions for each local maximum. Finally, the complete set of salient regions $R = \{r_1, ..., r_{2n}\}$ is returned for the next step in the pipeline.

Since this procedure can be applied to any saliency map, we propose two strategies based on the saliency methods described in Sections 4.1.1 and 4.1.2:

Single Saliency Map Region Extraction (S1)

The first strategy for extracting salient blobs (*S1*) is to compute a single saliency map using VOCUS2, as described in Section 4.1.1: compute contrast on several feature channels at different layers of the pyramid, and fuse the conspicuities into a single saliency map. There, one can apply the salient region extraction described before. We summarise the steps in Alg. 1:

Algorithm 1 S1 - Extract Salient Regions

```
procedure S1-ExtractSalientRegions
Input: Image I
Output: A set of salient regions $R = \{r_1, ..., r_{2n}\}$
    Compute saliency map $Sal$ on the input image $I$
    $L = \{l_1, ..., l_n\}$ := find_local_maxima($Sal$)
    $R = \{r_1, ..., r_{2n}\}$ := seeded_region_growing($Sal$,$L$)
```

Multi-Scale Saliency Region Extraction (S2)

In this method, we use the multi-scale saliency computation of Section 4.1.2. That means, instead of having a single saliency map, we have one per level of the pyramid.

Smaller structures will tend to show up in the regions extracted for the lower levels of the pyramid, and bigger ones in the higher levels. The salient region extraction procedure is therefore applied to the octave-specific saliency maps, resulting in a set of salient regions for each octave (in our experiments, we used octaves 1 to 4). We summarise the steps in Alg. 2:

Algorithm 2 S2 - Extract Salient Regions

```
procedure S2-ExtractSalientRegions
Input: Image I
Output: A set of salient regions $R = \{r_1, ..., r_{2n}\}$
    Compute octave-specific saliency maps $O = \{Sal_{o_1}, ..., Sal_{o_p}\}$ on the input image $I$
    for $Sal_{o_i} \in O$ do
        $L_i = \{l_1, ..., l_{n_i}\}$ := find_local_maxima($Sal_{o_i}$)
        $R_i = \{r_1, ..., r_{2n_i}\}$ := seeded_region_growing($Sal_{o_i}$,$L$)
        $R := R \cup R_i$
```

In the following algorithmic descriptions, we will refer to a general procedure called ExtractSalientRegions which maps to the desired mode for extracting salient regions (Alg. 1 for S1 or Alg. 2 for S2).

4.2. Integrating Image and Depth Segmentation

The salient regions extracted from the saliency map give us a good estimate of the location and dimensions of the objects present in the scene. However, in order to obtain their precise boundaries, we make use of a segmentation of the image as well as the depth map. We will evaluate four different methods. The first one (M1) relies purely on a colour segmentation; the second one (M2) is based on a segmentation of the depth map; the third one (M3) is a segmentation method that combines colour and depth —what we call the early fusion of colour and depth; the fourth one (M4) consists in putting together the candidates obtained independently with the M1 and M2 methods—what we call the late fusion approach.

4.2.1. Method 1 (M1): Felzenszwalb Segmentation

We chose the Felzenszwalb and Huttenlocher algorithm [Felzenszwalb and Huttenlocher, 2004] for segmenting colour images into perceptually coherent segments. The authors proposed a method that constructs a graph based on the pixel neighbourhoods, and iteratively merges groups of pixels into regions, keeping a trade-off between the internal variability of the regions and the difference between neighbouring components.

The algorithm constructs a graph on the image $G = (V, E)$ where each node $v \in V$ is a pixel, and each edge $e \in E$ connects two neighbouring pixels. Furthermore, a weight function is defined on the edges as the absolute difference between the intensity values of the pixels it connects:

$$w(e) = |I(p_i) - I(p_j)|, \tag{4.5}$$

where $e = (v_i, v_j)$ is the edge connecting vertices v_i and v_j, and $I(p_i)$ and $I(p_j)$ are their respective pixel intensities. This weight function can be easily adapted to compute the difference in colour values —e.g., in our case by computing their Euclidean distance.

A difference predicate between two components (subsets of neighbouring pixels $C \subseteq V$) $D(C_1, C_2)$ is defined in order to decide whether there is evidence for a boundary between them —and so, they should not be merged into one component. It is defined as

$$D(C_1, C_2) = \begin{cases} true & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & \text{otherwise} \end{cases}.$$

The predicate $D(C_1, C_2)$ evaluates whether the difference between two components, $Dif(C_1, C_2)$, is large enough when compared to a function of the internal difference of the two components:

$$MInt(C_1, C_2) = \min(Int(C1) + \frac{k}{|C1|}, Int(C2) + \frac{k}{|C2|}). \tag{4.6}$$

The function $MInt(C_1, C_2)$ has one adjustable parameter, k, that regulates the amount of variability that is tolerated inside the components and so, indirectly, the size of the segments. The internal difference of a component, $Int(C)$, is defined as the largest weight in the minimum spanning tree of the component $MST(C, E)$:

$$Int(C) = \max_{w \in MST(C,E)} w(e). \tag{4.7}$$

Finally, the difference between two components, $Dif(C_1, C_2)$, is the minimum edge weight connecting them:

$$Dif(C_1, C_2) = \min_{v_i \in C1, v_j \in C2, (v_i, v_j) \in E} w(v_i, v_j). \tag{4.8}$$

The algorithm iterates over the edges $e \in E$, sorted in ascending order according to their weight $w(e)$, picking the vertices joined by e: v_i and v_j. The regions C_i and C_j, in which v_i and v_j are contained, are merged if the predicate $D(C_i, C_j)$ does not hold

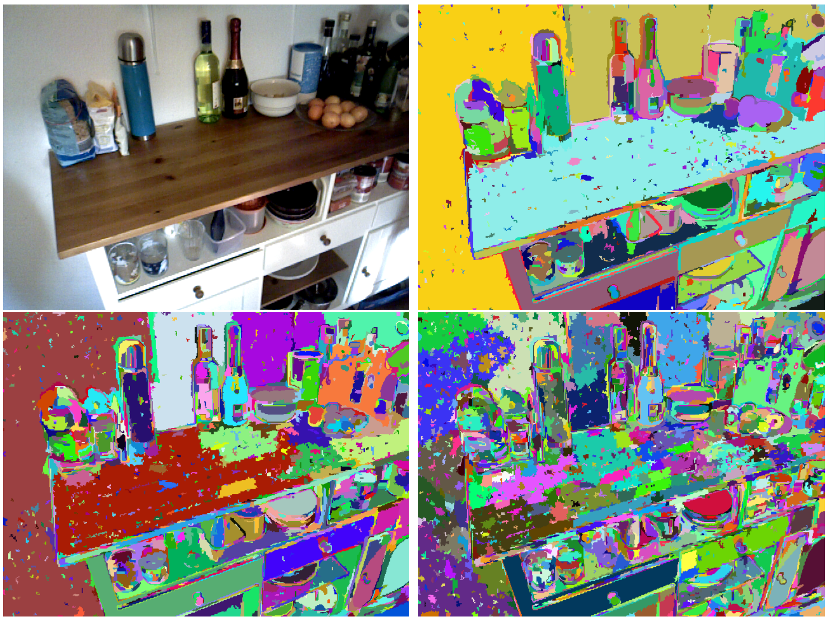


Figure 4.7.: Image from the Kitchen Dataset (top left) and its corresponding M1 segmentations for different values of the k parameter: $k = 200$ (top right), $k = 100$ (bottom left) and $k = 50$ (bottom right).

and are kept separate otherwise. If they are merged, the new internal difference of the resulting component is updated according to Eq. 4.7.

By tuning the parameter k one can adjust the level of over-segmentation required. We set $k = 200$ in our experiments. We show examples of the segmentations obtained for $k = 200$, $k = 100$ and $k = 50$ in Fig. 4.7.

4.2.2. Method 2 (M2): Surface Normals Clustering

The second method relies solely on depth information to produce image segments. It takes as input a 3D point cloud C — a set of 3D points $\{\mathbf{q}\}$ and their respective surface normals $\{\mathbf{n}\}$— obtained from a single depth map. Each point has three spatial coordinates $\mathbf{q} = (q_x, q_y, q_z)$ and a vector that is normal to the surface $\mathbf{n}$.

As in the first stage of the method of Potapova *et al.* [Potapova et al., 2014], we cluster points into planar patches based on their surface normals. The surface normal $\mathbf{n}$ of each point $\mathbf{q}$ is used as the initial model of the plane at that point. The algorithm will now try to add neighbouring points to S, the segment seeded by point $\mathbf{q}$: for a candidate point $\mathbf{q}'$ with surface normal $\mathbf{n}'$, the point is added to the segment if two conditions are satisfied: 1) the scalar product of the two normal vectors is below a

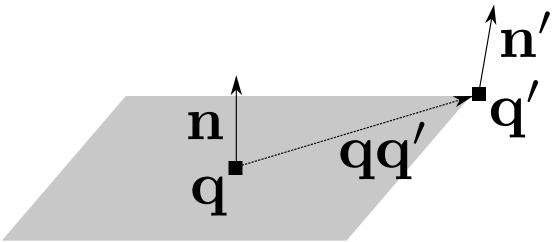


Figure 4.8.: In gray the plane model with point average $\mathbf{q}$ and normal average $\mathbf{n}$. A candidate point $\mathbf{q}'$ with surface normal $\mathbf{n}'$.

given threshold $t1$:

$$\mathbf{n} \cdot \mathbf{n}' < t1 \tag{4.9}$$

and 2), that

$$\mathbf{qq}' \cdot \mathbf{n}' > t2, \tag{4.10}$$

where $\mathbf{qq}'$ is the vector that goes from $\mathbf{q}$ to $\mathbf{q}'$. We illustrate this in Fig. 4.8: the first condition, Eq. 4.9, makes sure that the normal of the plane, $\mathbf{n}$, and the one of the candidate point, $\mathbf{n}'$, are as parallel as possible; the second condition, Eq. 4.10, makes sure that both are on the same plane, i.e., that the vector $\mathbf{qq}'$ is as perpendicular as possible to $\mathbf{n}'$.

If both conditions hold, the candidate point $\mathbf{q}'$ is added to the component S: $S := S \cup \{\mathbf{q}'\}$. Then, the component's normal $\mathbf{n}$ and average point $\mathbf{q}$ are updated as

$$\mathbf{n} := \frac{\mathbf{n} \cdot |S| + \mathbf{n}'}{|S| + 1}, \quad \mathbf{q} := \frac{\mathbf{q} \cdot |S| + \mathbf{q}'}{|S| + 1}. \tag{4.11}$$

Planar patches are iteratively created until all points belong to some patch —segment— or are labelled as noise.

An example of the segments obtained with this method is shown in Fig. 4.9. The limitations of the method have to do with the sensor used to obtain the depth maps. In the ASUS Xtion pro sensor —as in Kinect—, since the focal lengths of the depth and the colour sensors are not the same, not every pixel in the colour image gets a depth value in the end: see the black frame around the segmentation of Fig. 4.9 which corresponds to missing depth measurements. The second issue is that measurements farther away than 4 meters contain increasingly more noise and therefore result in noisy surface patches. Finally, because Kinect is an active sensor projecting an infra red pattern to the environment, black or glass made objects usually cause missing depth measurements to arise: see the bottles in the middle of the image for example. Despite

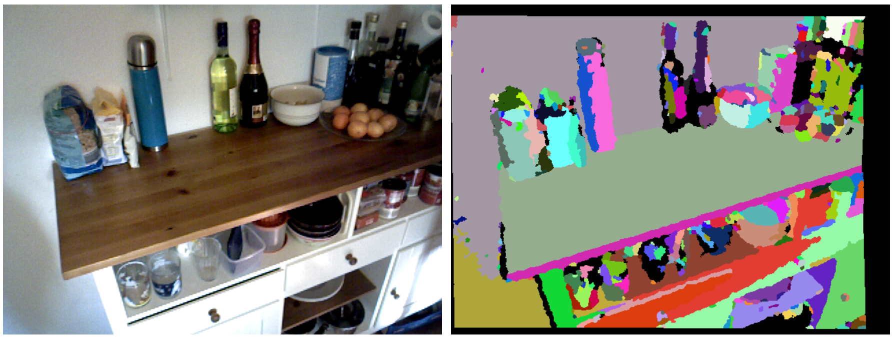


Figure 4.9.: Frame from the Kitchen Dataset and its corresponding M2 segmentation.

all these drawbacks, when the camera is close enough to the scene, the surface patches are reliably obtained.

4.2.3. Method 3 (M3): RGB-D Supervoxel Segmentation

Method *M3* is the RGB-D segmentation approach of [Papon et al., 2013], which generates volumetric segmentations of colour point clouds.[1] A colour point cloud is defined as a set of points $\{\mathbf{q}\}$ containing 3D position as well as colour information: $\mathbf{q} = (q_x, q_y, q_z, R, G, B)$. The use of this segmentation method is what we call the early fusion of colour and depth: both are used in the segmentation method before the object candidates are generated.

The algorithm starts by discretising the 3D space into a voxel grid V_G of resolution R_{voxel} (the size of the voxels). Seed voxels are evenly distributed over the 3D space by first creating a voxel grid S_G of resolution R_{seed} (with R_{seed} much greater than R_{voxel}), and then by choosing the closest occupied voxel in the voxel grid V_G to each of the voxel centers in S_G.

The connectivity between voxels is 26-adjacency (9 neighbours in the upper row, or horizontal section, 9 in the lower row, and 8 neighbours in the same row), which is used to compute an adjacency graph in 3D. The supervoxel segments are now computed by performing an iterative clustering algorithm in the following space:

$$\vec{F} = [x, y, z, L, a, b, FPFH_{1..33}] \tag{4.12}$$

where x, y, z are the 3D coordinates of the voxels, L, a, b are the colour values in the CIELAB space [Hunt, 1991], and $FPFH_{1..33}$ (for Fast Point Feature Histograms) are 33 local surface features which are pose invariant [Rusu et al., 2009]. The next

[1] We used the implementation from the PCL library http://pointclouds.org/documentation/tutorials/supervoxel_clustering.php

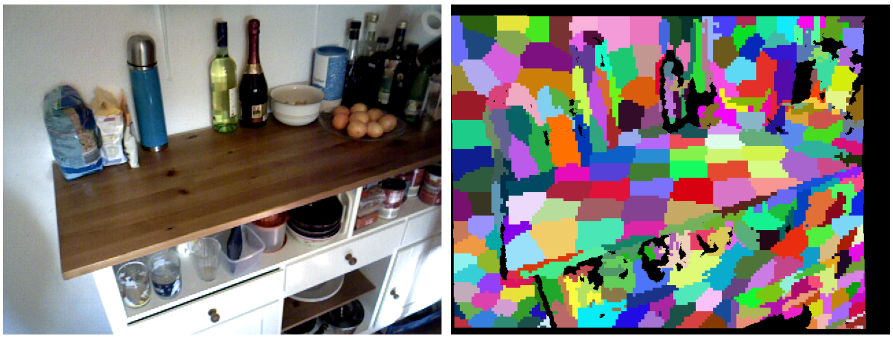


Figure 4.10.: Frame from the Kitchen dataset and its corresponding RGB-D supervoxel segmentation.

element we need to define is a distance measure in this space. The distance measure combines the individual distances in colour, space and local surface features, each of them properly normalised:

$$D = \sqrt{\frac{\lambda D_c^2}{m^2} + \frac{\mu D_s^2}{3R_{seed}^2} + \epsilon D_f^2} \tag{4.13}$$

where the spatial distance D_s is normalised by the largest possible distance between seed centres $\sqrt{3} \cdot R_{voxel}$ (the space diagonal of each seed voxel[2]); the colour distance D_c is the euclidean distance in the CIELAB colour space, and D_f measures the distance between local surface features using the Histogram Intersection Kernel [Barla et al., 2003]. Variables λ, μ and ϵ weight the relative importance of the distance measures.

The evenly distributed seeds are going to start a region growing process across the 3D adjacency graph where the neighbours are going to be recursively visited as long as their distance to the cluster center (seed) is the shortest of all. Therefore, for each cluster center —or segment—, if the distance of the visited voxel is the shortest of all, the voxel is assigned to that cluster. Exploring neighbours based on the adjacency graph prevents from having clusters that are not contiguous in space, and thus, eventually traversing object boundaries. After each iteration, the cluster centers are recomputed as the current mean of all its members.

This method is an alternative to M1 and M2 that integrates colour and depth. An example of a segmentation is shown in Fig. 4.10. This method has, a priori, one undesirable property: it over-segments the image into regions where we would expect one consistent segment. See for example the white wall in Fig. 4.10: it has a homogeneous texture and is a flat surface; however, the method breaks it in evenly distributed

[2] The space diagonal "of a polyhedron is a line connecting two vertices that are not on the same face" `https://en.wikipedia.org/wiki/Space_diagonal`

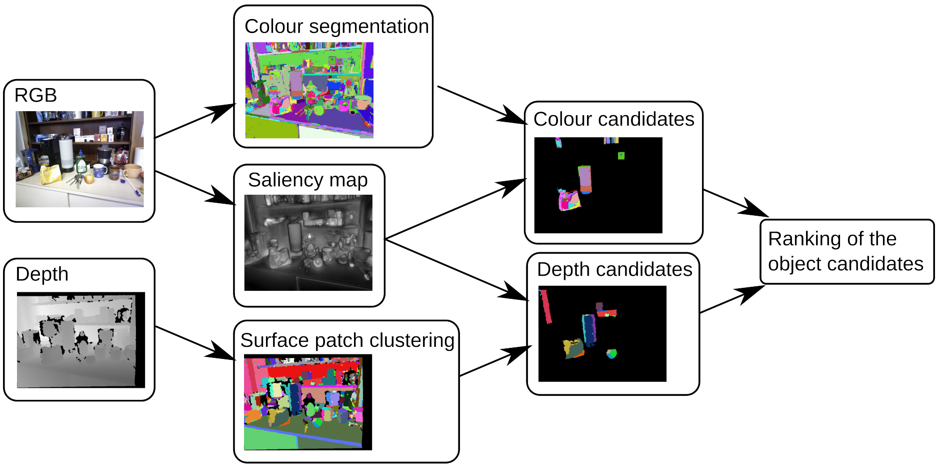


Figure 4.11.: Overview of the M4 Method: the image and the depth map are segmented independently. Saliency is used to generate colour and depth candidates based on their respective segmentations; both sets of candidates are finally merged together and ranked. Fig. from [Martín García et al., 2015b].

segments.

4.2.4. Method 4 (M4): Late Fusion of Color and Depth Candidates

As an alternative to the early fusion approach of method M3, we propose a late fusion approach of colour and depth. That is, instead of producing a segmentation that already considers depth and colour distances between pixels, we propose here to produce object candidates based on the colour and depth segmentations (M1 and M2) independently, and put them together afterwards.

As we will show in the evaluation, color and depth candidates are complementary: we obtain better results with this late fusion method than with the early fusion in M3. The late fusion approach of M4 is illustrated in Fig. 4.11.

4.3. From Segments to Object Proposals

At this point, we have salient regions and segments and we have to determine how segments form object candidates. The selection of segments works in the same way for each of the segmentation methods: for each salient region r, we pick the segments $s_i \in \{s_1, ..., s_m\}$ which overlap at least a fraction γ of the area of s_i. We set this overlap to $\gamma = 0.30$ with respect to the segment.

We show the general steps for generating object proposals for the M1, M2 and M3 segmentation methods in Alg. 3. The pseudo-code for generating M4 object candidates

Algorithm 3 M1-M2-M3 Generate Salient Object Proposals

```
procedure M1-M2-M3-GenerateProposals
Input: Image $I$, Depth map $D$
Input: A mode for extracting salient regions $ext \in \{"S1", "S2"\}$
Input: Segmentation mode $mode \in \{"M1", "M2", "M3"\}$
Output: A set of object proposals $\{C_1, ..., C_{2n}\}$
    $R :=$ ExtractSalientRegions($I$,$ext$)
    $S = \{s_1, ..., s_m\}$ :=over_segment($I$,$D$,$mode$)
    for $r_i \in R$ do
        for $s_j \in S$ do
            if $|r_i \cap s_j| > \gamma \cdot |s_j|$ then
                $C_i := C_i \cup \{s_j\}$
```

is shown in Alg. 4. In both algorithms we call a general procedure called ExtractSalientRegions which executes either S1 or S2 to extract salient regions.

To summarize the steps: first, saliency is computed on the input image; second, salient regions are extracted from the saliency map; finally, the salient regions are used to glue together the segments obtained from one of the four different segmentation algorithms.

4.4. Post-processing

As a last step before ranking the proposals in a suitable order, we perform two post-processing steps. First, non-maxima suppression is performed and duplicate candidates are removed. We check for the intersection of each pair of candidates and measure the two overlap ratios: each ratio is the number of pixels in the intersection divided by the number of pixels in one of the candidates. If both ratios are higher than a fixed threshold of 0.8 then the candidate with a lower ranking score (see next section) is removed. Second, object candidates larger than 2/3 of the image width/height in one of their dimensions are removed for being too large.

4.5. Proposal Ranking

An important step is to sort object proposals according to some objectness measure, i.e., to rank the proposals according to their quality so that the most promising candidates are picked first. We investigated three different approaches for ranking the object candidates.

Algorithm 4 M4 Generate Salient Object Proposals

```
procedure M4-GenerateProposals
Input: Image $I$, Depth map $D$
Input: A mode for extracting salient regions $ext \in \{"S1", "S2"\}$
Output: A set of object proposals $\{C_1, ..., C_{2n}\}$
    $R$ := ExtractSalientRegions(I)
    $S_1 = \{s^1_1, ..., s^1_{m_1}\}$ := over_segment(I, "M1")
    $S_2 = \{s^2_1, ..., s^2_{m_2}\}$ := over_segment(D, "M2")
    for $r_i \in R$ do
        for $s^1_j \in S_1$ do
            if $|r_i \cap s^1_j| > \gamma \cdot |s^1_j|$ then
                $C_i := C_i \cup \{s^1_j\}$
        for $s^2_k \in S_2$ do
            if $|r_i \cap s^2_k| > \gamma \cdot |s^2_k|$ then
                $C_{n+i} := C_{n+i} \cup \{s^2_k\}$
```

4.5.1. Saliency-Area Ranking (R1)

As a first approach, we used the average saliency of a candidate $c = \{p_i\}_{i=0}^{N}$ for ranking, where a candidate c contains N pixels p_i. However, small proposals have a higher saliency: the region growing process (Sec. 4.1.4) starts at a local maximum and the larger it grows, the more pixels enter the region that have lower saliency values. Thus, to avoid this size bias towards small objects we incorporated the size of the candidate into the ranking score:

$$\text{sal\_area\_score}(p) = \text{avg\_saliency}(p) * \sqrt{N}, \tag{4.14}$$

where the average saliency is simply avg_saliency(p) $= \frac{1}{N}\sum_{i=0}^{N} sal(p_i)$ and the area of the candidate is the number of pixels it contains, N.

4.5.2. 3D Convexity Ranking (R2)

Our second ranking approach (*R2*) sorts candidates according to their 3D convexity. 3D convexity is a feature that is commonly used in robotics algorithms for object discovery [Karpathy et al., 2013, Potapova et al., 2014]. It is also known to be among the Gestalt cues that influence the figure-ground segregation processes in human vision [Kanizsa and Gerbino, 1976].

We compute this feature following the method of [Potapova et al., 2014]. Given a candidate object's 3D point cloud $\{\mathbf{q_i}\}_{i=1}^{N}$, we compute their convex hull V, consisting of a set of visible faces $\{v_j\}$. The convexity measure κ is computed as the mean of the shortest distances from the object points to the visible surfaces of the object's convex hull:

$$\kappa = \frac{1}{N} \sum_{\mathbf{q_i}} d_{min}(\mathbf{q_i}, V), \tag{4.15}$$

where N is the number of object points and $d_{min}(\mathbf{q_i}, V)$ is the shortest distance from the point to the visible faces

$$d_{min}(\mathbf{q_i}, V) = \min_j d(\mathbf{q_i}, v_j). \tag{4.16}$$

Since this feature averages distances from points to planes, the lower the value the more convex the object is. In the limit, imagine a sphere, where every point has one tangent plane and therefore the distances are zero for each of the points.

4.5.3. SVM Ranking (R3)

Our last approach for ranking the object candidates *R3* uses Support Vector Machines (SVM), a well known machine learning algorithm, to learn the objectness of object candidate masks as a function of different features. The features we considered are —in parenthesis their index in the feature vector:

- (1-7) Hu's image moments [Hu, 1962]. These are well known descriptors of shape which are invariant to rotation and scale.
- (8) The 3D convexity measure of Section 4.5.2.
- (9) The area of the object candidate mask normalised to the area of the complete image.
- (10) The average saliency of the proposal as in ranking *R1* (Section 4.5.1).
- (11) The perimeter of the object candidate mask normalized to the image area.
- (12) The normalized average depth of the proposal.

Hu's moments We will try to synthesise the important ideas about Hu's moments following Chapter 2 of Liao's thesis [Liao, 1993]. Hu's moments were developed as a way to characterise a function —in this case an image— in a unique way. The moment of order $(p+q)$ of a function $f(x,y)$ is defined as

$$M_{pq} = \int \int x^p \; y^q \; f(x,y) \, dx \, dy \,. \tag{4.17}$$

The moment of order 0 of a function would measure the total mass of the function:

$$M_{00} = \int\int f(x,y)\,dx\,dy\,. \tag{4.18}$$

There are two first order moments:

$$M_{10} = \int\int x\ f(x,y)\,dx\,dy, \quad M_{01} = \int\int y\ f(x,y)\,dx\,dy, \tag{4.19}$$

which can be interpreted as the center of mass of f. The image coordinates of the center of mass of the function can be computed as

$$\bar{x} = \frac{M_{01}}{M_{00}}, \quad \bar{y} = \frac{M_{10}}{M_{00}}. \tag{4.20}$$

The coordinates of the center of mass can now be used as a reference point to describe function f. Let us now define the central moments of $f(x,y)$ as

$$\mu_{pq} = \int\int (x-\bar{x})^p\ (y-\bar{y})^q\ f(x,y)\,dx\,dy\,. \tag{4.21}$$

Based on this definition of central moments, Hu defined a set of seven functions that are invariant under object scale, translation and rotation, which we use as part of our set of features:

$$\begin{aligned}
\phi_1 &= \mu_{20} + \mu_{02} && (4.22)\\
\phi_2 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 && (4.23)\\
\phi_3 &= (\mu_{30} - 3\mu_{12})^2 + (\mu_{21} - \mu_{03})^2 && (4.24)\\
\phi_4 &= (\mu_{30} - \mu_{12})^2 + (\mu_{21} - \mu_{03})^2 && (4.25)\\
\phi_5 &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] &&\\
&\quad +(3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] && (4.26)\\
\phi_6 &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] &&\\
&\quad +4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}) && (4.27)\\
\phi_7 &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] &&\\
&\quad -(\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})] && (4.28)
\end{aligned}$$

With this set of features we trained a Support Vector Machine (SVM) [Chang and Lin, 2011] with a radial basis function (RBF) to learn a classification function between objects/non-objects. The output of the SVM, which is the distance to the separating hyperplane between the two classes, can be used as a score to rank the object candidates according to their objectness.

4.6. Evaluation

In this section, we evaluate our family of methods for object discovery on several publicly available datasets. As we have explained throughout this chapter, there are two possible strategies for extracting salient regions, S1 and S2; four possible segmentation methods, M1, M2, M3 and M4; and three different ranking strategies, R1, R2 and R3. All together, this means 36 possible combinations. To lower this number, we will first evaluate the ranking strategies separately, and fix the ranking method to the one that proved most successful.

This section is structured as follows: first, we introduce the datasets and metrics for the evaluation. Then, we evaluate the ranking methods independently. Based on the results obtained, in the following evaluations we will fix the ranking strategy to the one that performs best, and then evaluate our other 8 object discovery variants and compare them to other state-of-the-art methods.

4.6.1. Object Discovery Datasets

We use three publicly available datasets: the Washington Dataset for Object Recognition [Lai et al., 2011], our own Coffee Machine Sequence (CMS), which appeared first in [Martín García and Frintrop, 2013], and the Kitchen Object Discovery Dataset (KOD) [Horbert et al., 2015]. The three of them contain RGB-D data.

Washington Dataset The Washington Dataset[3] appeared in [Lai et al., 2011] as a benchmark for object recognition.[4] It contains 8 sequences of indoor scenes recorded with a Kinect camera (see some examples of the sequences in Fig. 4.12). The ground truth was manually annotated in the form of bounding boxes for some of the objects that show up in the sequences. It can be argued that the sequences are relatively easy, showing objects placed on table top surfaces.

Coffee Machine Sequence (CMS) The CMS is a challenging highly cluttered scene for object discovery, with a total of 80 distinct objects appearing throughout the sequence and up to 48 objects per frame. It lasts for 436 frames, and has manually annotated ground truth for every 30th frame with consistent labels throughout the frames. Some frames with their corresponding annotated ground truth are shown in Fig. 4.13.

[3] Available online at `http://rgbd-dataset.cs.washington.edu/dataset/`

[4] There is also the recent dataset of [Lai et al., 2014], but the sequences do not add difficulty to the task of object discovery.

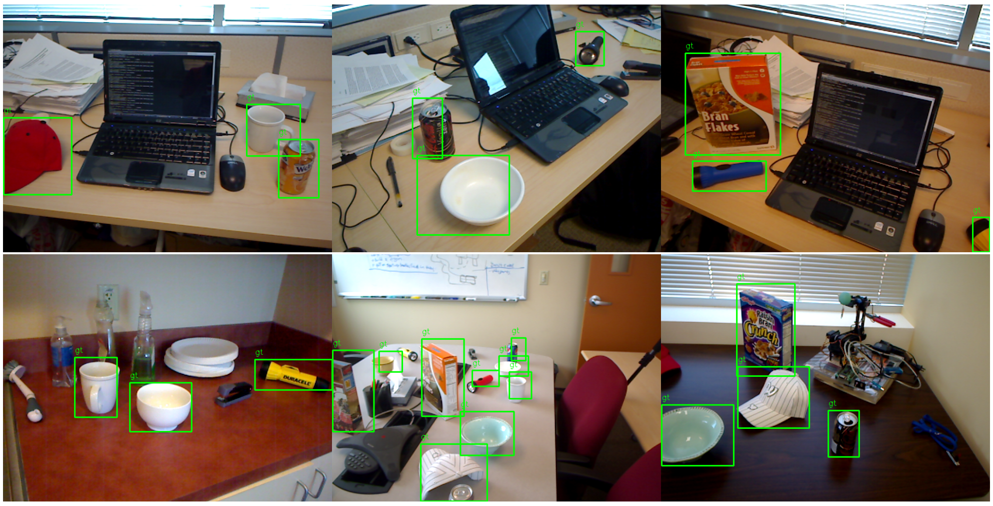


Figure 4.12.: Colour frames from six of the eight sequences of the Washington Dataset and the annotated ground truth as bounding boxes.

Kitchen Object Discovery Dataset (KOD) The last dataset consists of four challenging video sequences recorded in real-world kitchen environments containing a high degree of clutter. The sequences have on average about 600 frames and contain up to 80 objects. It was introduced in [Horbert et al., 2015][5] as an object discovery benchmark, with ground truth manually annotated on every 30th frame. The object identities are kept consistent in the labels throughout the sequences. Some frames together with the annotated ground truth are shown in Fig. 4.14.

4.6.2. Metrics

We consider object candidates to be correct if they satisfy the Pascal criterion, *i.e.*, if the intersection-over-union (IoU) ratio is greater than 0.5 [Everingham et al., 2007]. Keeping [Everingham et al., 2007] notation, for a candidate bounding box B_p and some ground truth bounding box B_{gt}, the candidate is considered correct if the ratio

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{4.29}$$

is greater than 0.5, where $B_p \cap B_{gt}$ is the intersection of both bounding boxes and $B_p \cup B_{gt}$ is their union.

Such a ratio can be computed at the pixel level when both the object mask and the annotated ground truth are pixel precise. Since some of the methods we compare to only provide bounding boxes, we compute this ratio for bounding boxes for all of the

[5] Available online at `http://www.vision.rwth-aachen.de/projects/kod/`

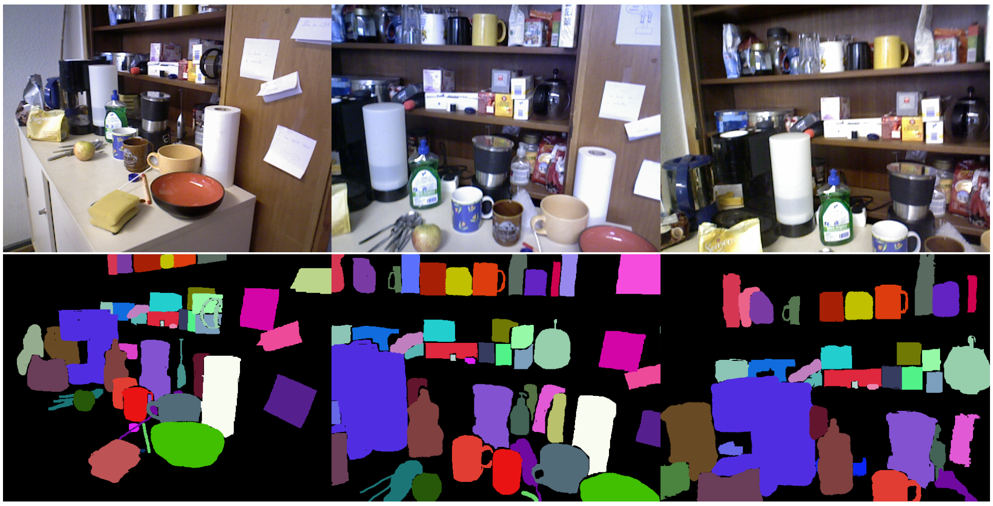


Figure 4.13.: Colour frames from Coffe Machine Sequence and its corresponding ground truth.

methods to have a fair comparison.

In the following we will measure the performance of the methods in terms of the candidates that successfully find objects, and in terms of the quality of those candidates. Therefore, we will evaluate the following metrics:

Precision It is the ratio of correct candidates that are returned by the algorithm for a given frame:

$$precision = \frac{|\ \{\texttt{ground truth objects}\}\ \cap\ \{\texttt{generated candidates}\}|}{|\texttt{generated candidates}|} \tag{4.30}$$

Recall It is the ratio of retrieved objects over the total number of ground truth objects in a given frame:

$$recall = \frac{|\ \{\texttt{ground truth objects}\}\ \cap\ \{\texttt{generated candidates}\}|}{|\texttt{ground truth objects}|} \tag{4.31}$$

Based on these two metrics we produce the following plots:

Precision / Number of Proposals (Frame-based) This plot shows how the precision evolves as the number of generated proposals grows. It is measured for each frame independently.

Recall / Number of Proposals (Frame-based) Equivalently, this plot shows how the recall evolves as the number of generated proposals grows. That means, of the objects present in a given frame, it measures the ratio of objects found by the generated candidates. It is measured for each frame independently.

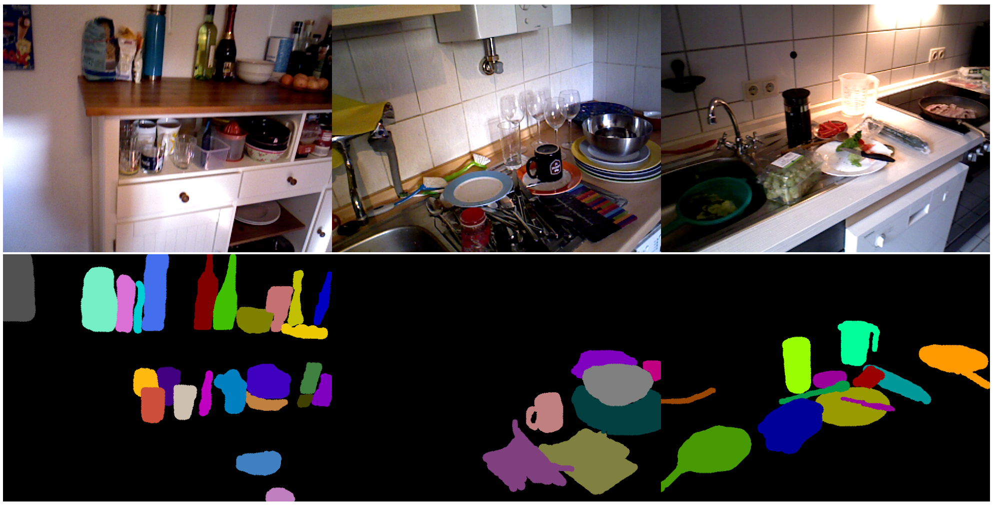


Figure 4.14.: Colour frames from three sequences of the Kitchen Dataset and their corresponding ground truth.

Recall / IoU (Frame-based) This plot shows the recall as a function of the IoU. It lets us evaluate the quality of the generated object candidates.

Global Recall / Time (sequence-based) This is a measure of recall considering the identity of the objects consistent throughout the sequence. That means, if an object appears in several frames of the sequence, for the global recall it is enough that a proposal finds it in one frame and not in the others. Therefore, it is measured globally on the whole sequence. In this type of plot we show how the global recall measure evolves over time.

Global Recall / Number of Proposals (Sequence-based) As before, this plot shows global recall as a global measure of the ratio of objects found throughout the whole sequence. In this plot, it is displayed as a function of the number of proposals. This plot gives an idea of how many proposals are needed to achieve a desired global recall value.

4.6.3. Evaluation of the Ranking Methods

In the first part of the evaluation, we compare the three ranking methods explained in Section 4.5. Namely, the saliency-area score (*R1*), the convexity score (*R2*), and the SVM score (*R3*). Because method *R3* involves learning on training data, we used the Washington Dataset as training and test set and divided it in two parts. The first part was used first for training and the second one for testing, and then the second

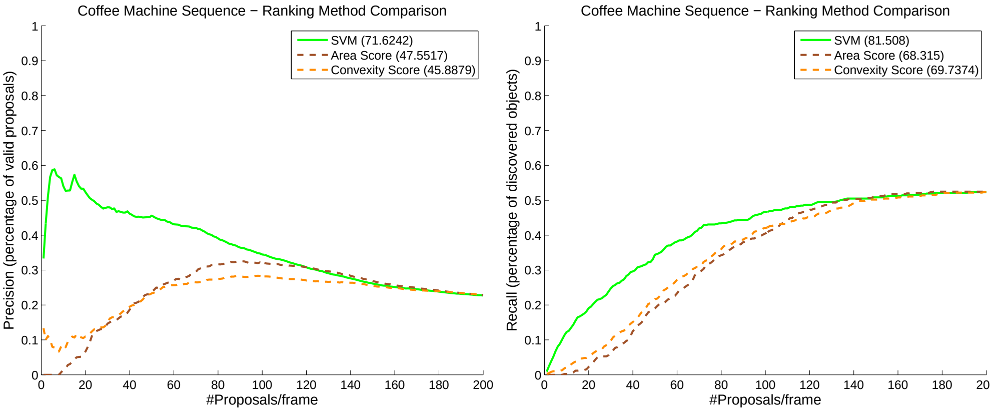


Figure 4.15.: Precision and recall for S1-M4 using the three ranking methods on the Coffee Machine Sequence. In parentheses the AUC values.

part was used for training and the first for testing. The model learned in the first part of the dataset was used to test on the Coffee Machine Sequence as well as the Kitchen Dataset in the rest of the experiments. As we mention in Section 4.5, we use a radial basis function (RBF) in our SVM: $K(x_i, x_j) = \exp(-\gamma \ ||x_i - x_j||^2)$, where x_i and x_j are points in the feature space, and the meta-parameter γ was determined by cross-validation on the Washington Dataset.

In Fig. 4.15, we show the results obtained in the Coffee Machine Sequence for our method S1-M4 (single saliency region extraction, late fusion of colour and depth candidates) using the three available ranking strategies. The curves show a much better performance of the SVM ranking method (R3) in both the precision and recall curves with respect to the other two. As can be seen in the plots, neither the area score nor the convexity score alone are able to give a good set of proposals for the first 50 candidates. The SVM ranking, however, achieves high precision values for the lowest number of proposals and decays progressively until it converges with the other methods.

The results justify that in the following, we chose the *R3* ranking strategy for the evaluation of the segmentation and salient region extraction methods.

4.6.4. Evaluation of the Object Proposals

Our method for object discovery consists of three main elements: salient region extraction (methods S1 and S2), image/depth map segmentation (methods M1, M2, M3, and M4), and candidate ranking strategy (methods R1, R2, and R3). In order to simplify the evaluation of all the possible combinations for each of the stages, we will use in the following the R3 ranking strategy, since it showed in the previous section (Sec. 4.6.3) a clear superiority with respect to the other two methods.

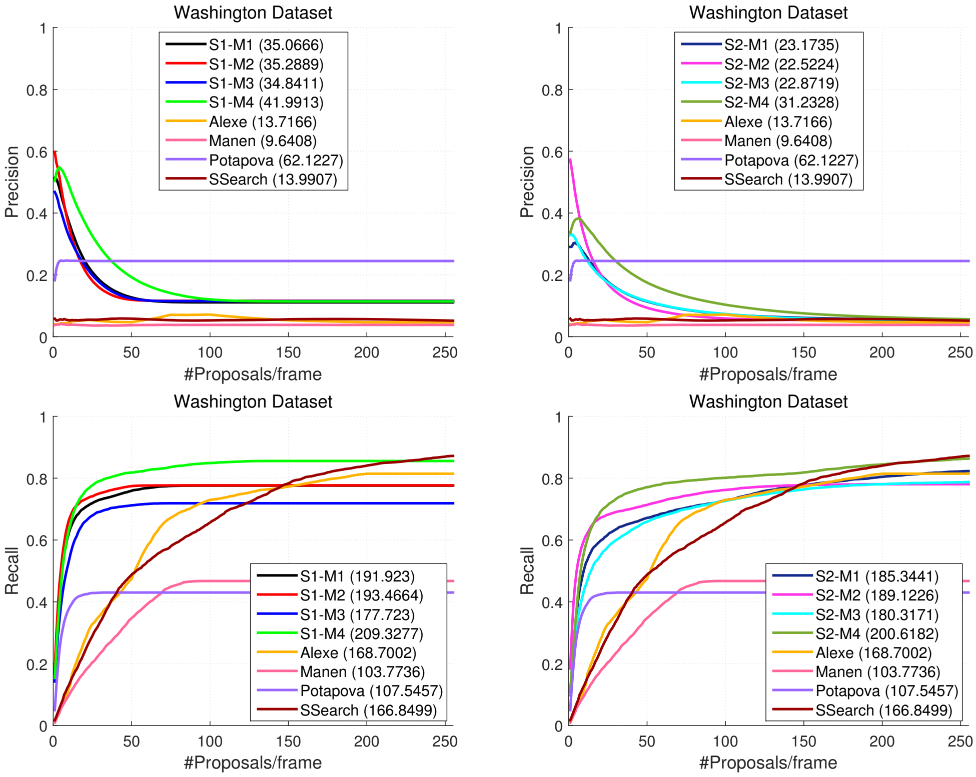


Figure 4.16.: Avg. precision and recall over proposals for the Washington Dataset (AUCs in parenthesis). Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search.

Thus, in the following, we evaluate the eight possible variants of our method internally (fixing R3, we have combinations such as S1-M1, S2-M1, etc.) and also compare them to four other state of the art methods in object discovery. Namely, the method of [Potapova et al., 2014], the Objectness measure of [Alexe et al., 2012], the Randomized Prim approach of [Manén et al., 2013] and the Selective Search method of [Uijlings et al., 2013].

Washington Dataset

To have an overview of the results, average precision and recall plots are shown in Fig. 4.16, the global recall over proposals is shown in Fig. 4.17 and recall over IoU average plots are shown in Fig. 4.18. The full set of plots for the eight individual sequences are shown in Appendix A. We compare our methods to the Selective Search of [Uijlings et al., 2013], the method of [Manén et al., 2013], the method of [Alexe et al., 2012] and the one of [Potapova et al., 2014].

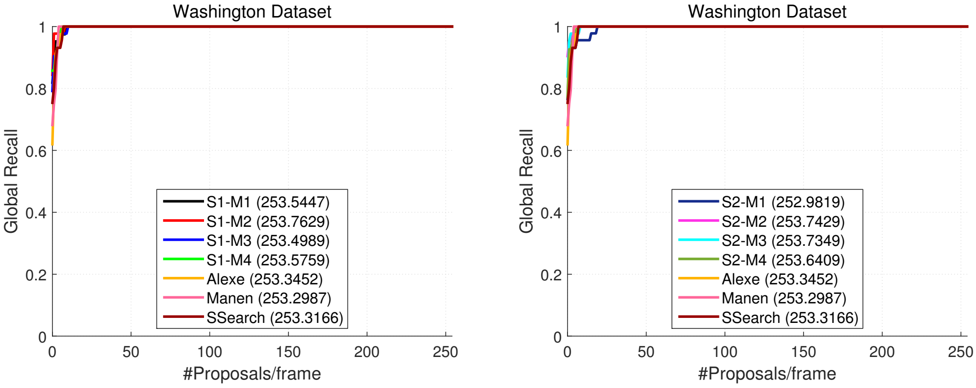


Figure 4.17.: Global recall over number of proposals. Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search. Average values for the Washington Dataset (AUCs in parenthesis).

Recall The Washington Dataset contains few objects and not all the objects that are present are labelled in the ground truth. Therefore, it can be argued that the dataset is relatively simple, and this is reflected in the recall plots (see the bottom row of Fig. 4.16): for a small number of proposals (below 50 per frame) all our methods achieve high recall values (above 0.8 for S1-M4, for example). Among the competitors, the methods of Alexe and the Selective Search reach comparable results towards the end of the curve (after 150 proposals are considered).

Precision In terms of precision (see the top row of Fig. 4.16), it can be seen that when generating below 50 proposals per frame, our methods achieve the highest precision values, together with the method of Potapova. It has to be noted, that the method of Potapova does not produce 255 proposals per frame, but rather something between 20 and 30 proposals, and we decided to prolong the curve in the plots at the end value it reached. Looking at the plots, it can be observed that the precision values reached by our methods are high at the beginning, meaning that the ranking method works well. The end values, however, are relatively low (our methods converge to a precision value below 0.2), which can be explained by the low number of objects that are present in this dataset, and that additionally not all of them are labelled. However, they are much higher than the other competitors: Selective Search, the method of Alexe and the one of Manén converge at a value below 0.1.

Global Recall The global recall is plotted as a function of the number of proposals in Fig. 4.17. The plots reflect the low difficulty of this dataset: all methods reach a

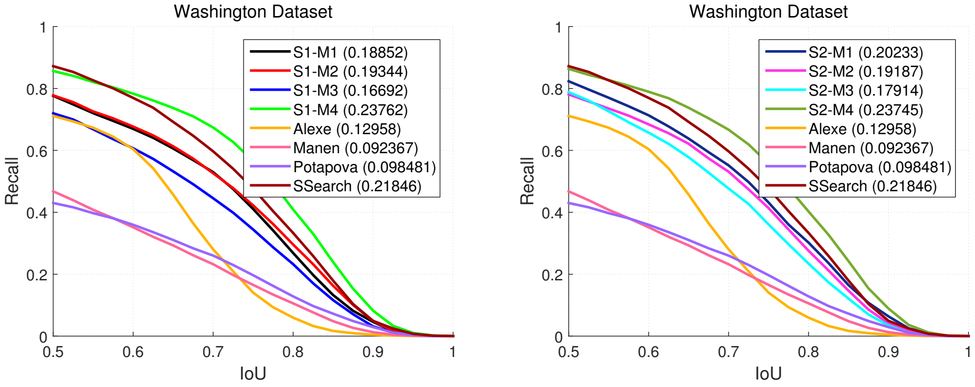


Figure 4.18.: Recall over IoU. Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search. Average values for the Washington Dataset (AUCs in parenthesis).

global recall of 1 with a few proposals per frame (below 50). In Appendix A, we show additionally the global recall over time plots for the individual sequences. There, it can be seen that after a few frames, most of the objects that show up in the scene are recalled by all methods.

Quality of the Proposals We show the recall over IoU in Fig. 4.18. These plots show what recall would be achieved if a given IoU is desired: the higher the IoU, the higher the quality of the object candidates. It can be seen that our methods are consistently better than the competitors, especially as the required IoU gets higher values. The best results were obtained by S1-M4 and S2-M4 with AUCs of 0.23762 and 0.23745 respectively.

To sum up the results, in terms of recall, the colour and depth modalities were complementary. That means that some objects are best retrieved using colour (M1) and some using depth (M2): regardless of the saliency method that was used (S1 or S2), the recall obtained by M1 and M2 independently is similar and it is boosted when both sets of candidates are combined (M4). We illustrate these results in Fig. 4.19: we show from top to bottom the results obtained with S2-M1, S2-M2, S2-M3 and S2-M4 for two frames from this dataset.

Coffee Machine Sequence

We now show the results obtained on the more challenging Coffee Machine Sequence. We compare our methods to the Selective Search of [Uijlings et al., 2013], the method of [Manén et al., 2013], [Alexe et al., 2012] and the one of [Potapova et al., 2014].

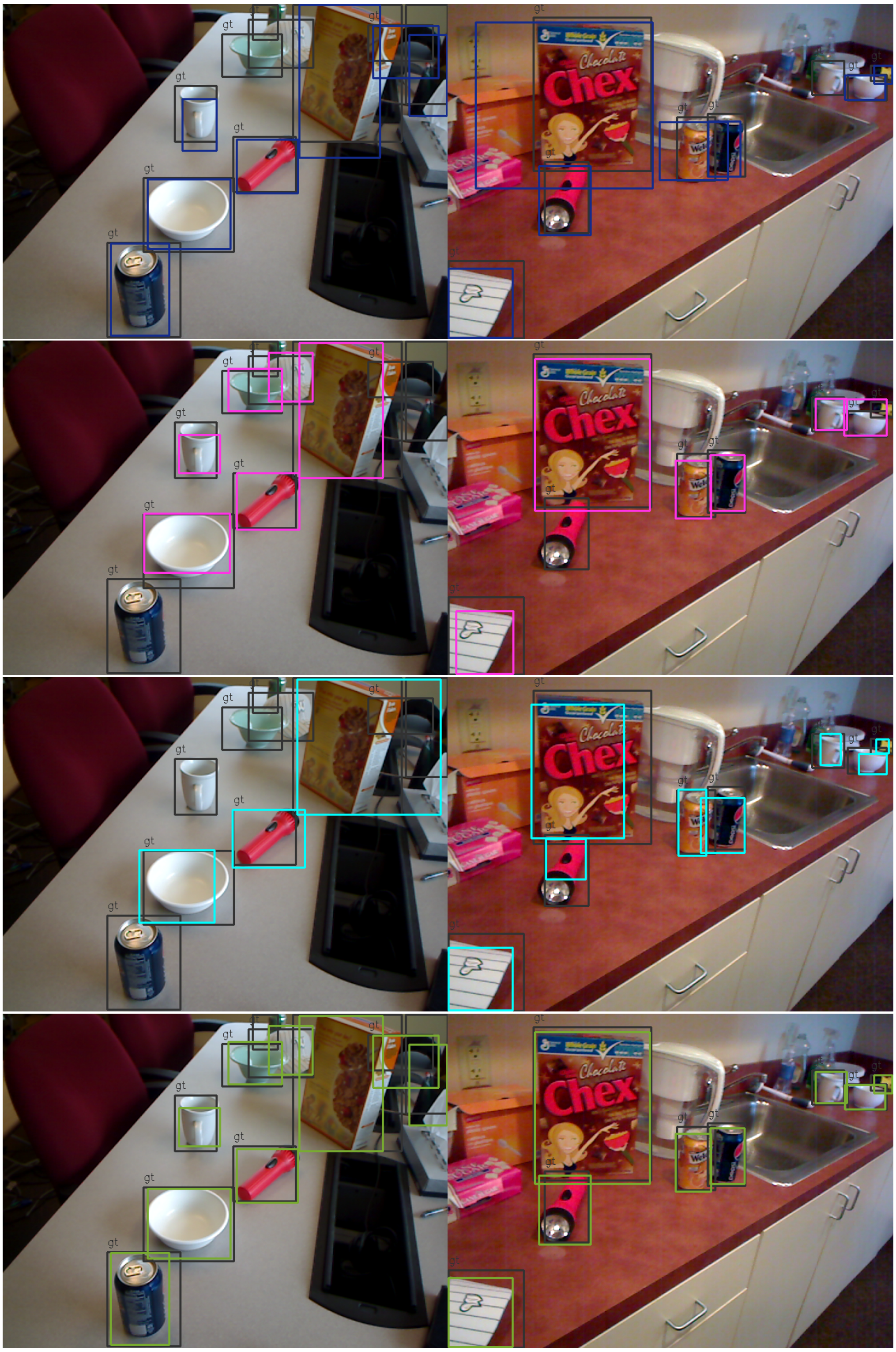


Figure 4.19.: Example results obtained in two sequences of the Washington Dataset. From top to bottom the successful candidates for S2-M1 (dark blue), S2-M2 (pink), S2-M3 (cyan) and S2-M4 (green) with the bounding boxes coloured according to the colour scheme shown in the plots. The ground truth is shown in gray.

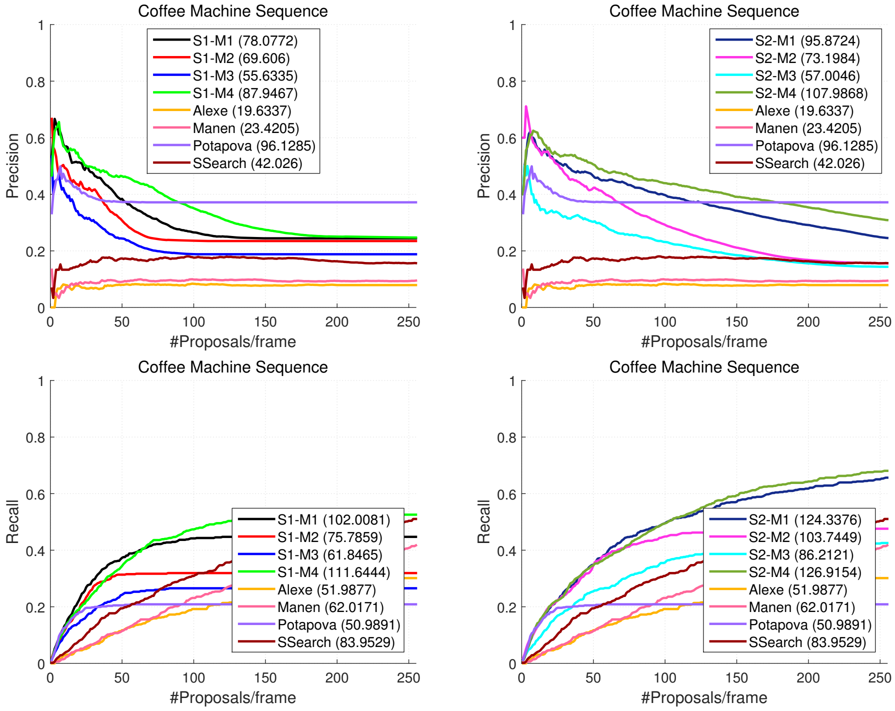


Figure 4.20.: Precision (top) and recall (bottom) for the Coffee Machine Sequence (AUCs in parenthesis). Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search.

Precision The precision plots in Fig. 4.20 (top row) show, as in the previous dataset, that for all our methods the precision values are high for a few proposals and slowly decrease; meaning that the ranking method is successful in choosing the good candidates first.

Recall In terms of frame-based recall —bottom row of Fig. 4.20—, the best performing method is S2-M4 (multi-scale saliency, late fusion of colour and depth), reaching almost 0.7 at the end of the curve, and followed closely by S2-M1 (multi-scale saliency and colour segmentation). The small offset between both curves in the multi-scale plot (0.68 vs 0.66 at the end of the curve for S2-M4 and S2-M1 respectively) suggests that colour is the most important modality in this sequence in order to find the objects, and depth does not add much to what colour alone can achieve. One possible explanation is that the distances of many of the objects to the camera are higher than in the Washington Dataset, making the depth clustering noisier and, therefore, the M2 method less precise. In any case, the late fusion approach (M4) is far superior than the early fusion approach (M3) in both saliency modes.

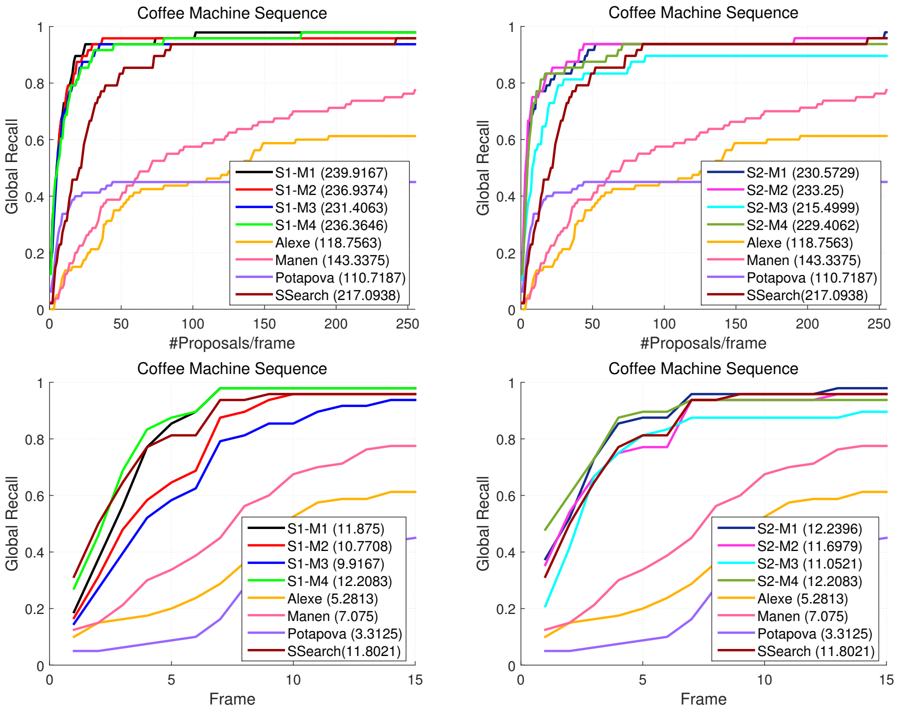


Figure 4.21.: Global recall over proposals (top) and global recall over time (bottom) for the Coffee Machine Sequence (AUCs in parenthesis). Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search.

Global Recall Global recall as a function of the number of proposals is shown on the top row of Fig. 4.21. It can be seen that generating 50 proposals per frame is enough to reach a very high global recall for our methods (e.g., 0.92 for S2-M1 or 0.94 for S2-M2). From the competitors, the most successful one is the Selective Search, reaching an equivalent recall (0.94) after 100 or more proposals are considered. The bottom row of Fig. 4.21 shows the evolution of the global recall over time. It can be seen that our proposed methods are close together on top (being S2-M1 the best, reaching a global recall of 0.98), with similarly good results by Selective Search (reaching 0.96).

Quality of the Proposals We plot the recall over IoU in Fig. 4.22. It can be seen that for the single saliency methods, our S2-M4 method performs best, closely followed by Selective Search, our S1-M1 as well as the method of Manén. The quality of our proposals gets a boost when we use our S2 method for extracting salient regions: with the split octaves method (S2) we are able to more precisely identify the boundaries of the objects compared to those obtained with a single saliency map (S1).

We show the results obtained in some frames of the CMS on the top row of Fig. 4.25.

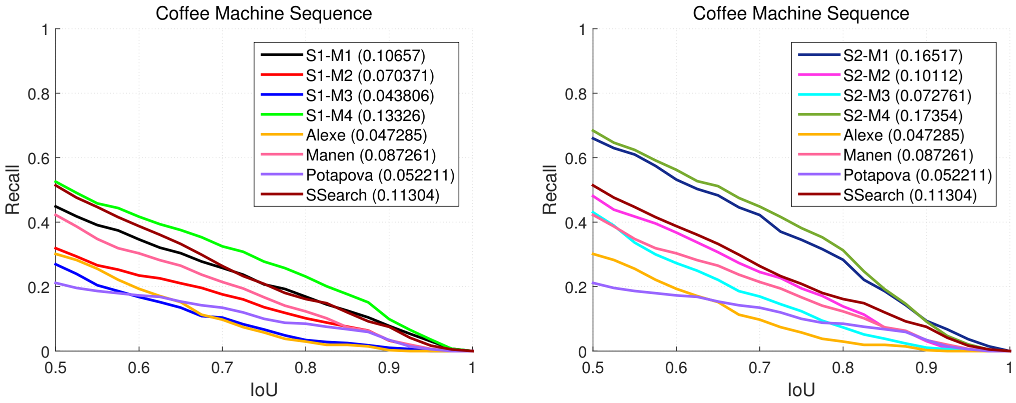


Figure 4.22.: Recall over IoU for the Coffee Machine Sequence (AUCs in parenthesis). Comparison of our S1 (left) and S2 (right) methods in their four segmentation variants with the methods of Alexe, Manén, Potapova and Selective Search.

Kitchen Dataset

We now show the results obtained in the Kitchen Dataset, which has comparable difficulty to the Coffee Machine Sequence. We compare all the variants of our approach to the Selective Search of [Uijlings et al., 2013], to the method of [Manén et al., 2013] and to the one of [Alexe et al., 2012].

Precision In terms of precision —top row of Fig. 4.23— our method S2-M4 obtains the higher average AUC (56.99) and is closely followed by our other variant S1-M4 (53.7). Here the differences are quite high with respect to our competitors: Selective Search (23.8), Alexe (6.5) and Manén (16.4).

Recall The highest recall —bottom row of Fig. 4.23— is achieved again by S2-M4: 128.42 AUC on average, followed as well by our S1-M4 method (single saliency, late fusion of colour and depth). In this dataset, we can see that the late fusion approach of colour and depth (M4) makes a difference in terms of frame based recall. For both the single saliency (S1) and the multi-scale saliency (S2) approaches the performance is boosted when using the M4 segmentation, as it happened in the Washington Dataset. The late fusion approach (M4) performs much better than the early fusion (M3) which is far behind in both precision and recall. Among the competitors, the Selective Search approach reaches good recall values at the end of its curve (it touches S1-M4 at the end of the plot).

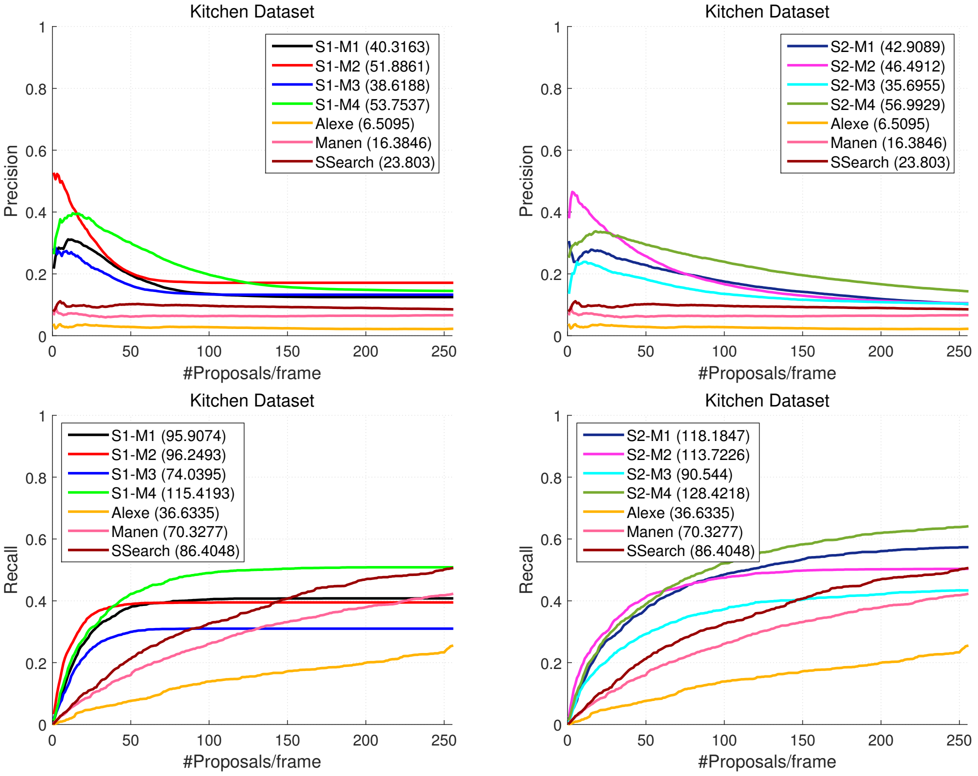


Figure 4.23.: Average precision and recall over number of proposals for the Kitchen Dataset (AUCs in parenthesis). Comparison of our S1 (top) and S2 (bottom) methods in their four segmentation variants with the methods of Alexe, Manén and Selective Search.

Quality of the Proposals We show the recall over IoU in the top row of Fig. 4.24. The results follow the same trend as in the other two datasets (Washington and CMS): the multi-scale approach for salient region extraction (S2) produces a boost in the quality of the proposals for all our methods. As before, M4 is the most successful in both saliency modes (S1 and S2). Among the competitors, the Selective Search was again the most successful (AUC of 0.106), with a quality slightly below the one of S1-M4 (AUC of 0.124). It is interesting to note that the performance of the Objectness measure of Alexe drops significantly with respect to the results obtained in the Washington Dataset (cf. Fig. 4.18).

Global Recall The global recall plots —bottom row of Fig. 4.24— reflect the difficulty of this benchmark: whereas in the Washington Dataset all the methods reach a global recall of 1 for a few proposals —see Fig. 4.17—, here most of the values are below 0.9. The highest global recall (0.91) is achieved by our S2-M4 method; following are

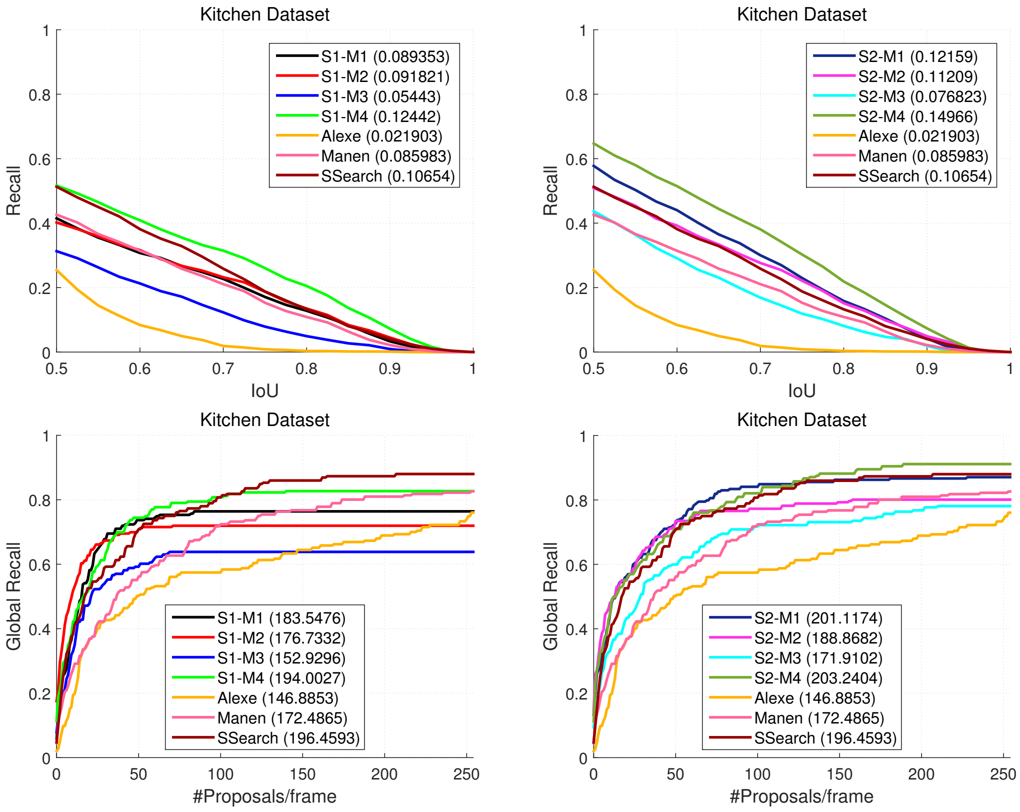


Figure 4.24.: Top: Recall over IoU. Bottom: Global recall over number of proposals. Comparison of our S1 (left col.) and S2 (right col.) methods in their four segmentation variants with the methods of Alexe, Manén and Selective Search. Average values for the Kitchen Dataset (AUCs in parenthesis). Average values for the Kitchen Dataset.

Selective Search (0.88) and our S2-M1 (0.87).

We show some exemplary results of the successful candidates on the four sequences of the dataset in Fig. 4.25 and Fig. 4.26. Additionally, the plots obtained in the individual sequences of this dataset are included in Appendix A.

4.6.5. Summary of the Evaluations

Ranking Strategies We have first compared the performance of the ranking methods R1 (average saliency times area), R2 (3D convexity measure) and R3 (ranking provided by an SVM trained on several features) on our own set of proposals. We showed that the R3 ranking method is superior to the other two, and it could be observed throughout the evaluation on all the datasets that the highest values on the precision plots are obtained at the beginning, meaning that the good proposals are picked first.

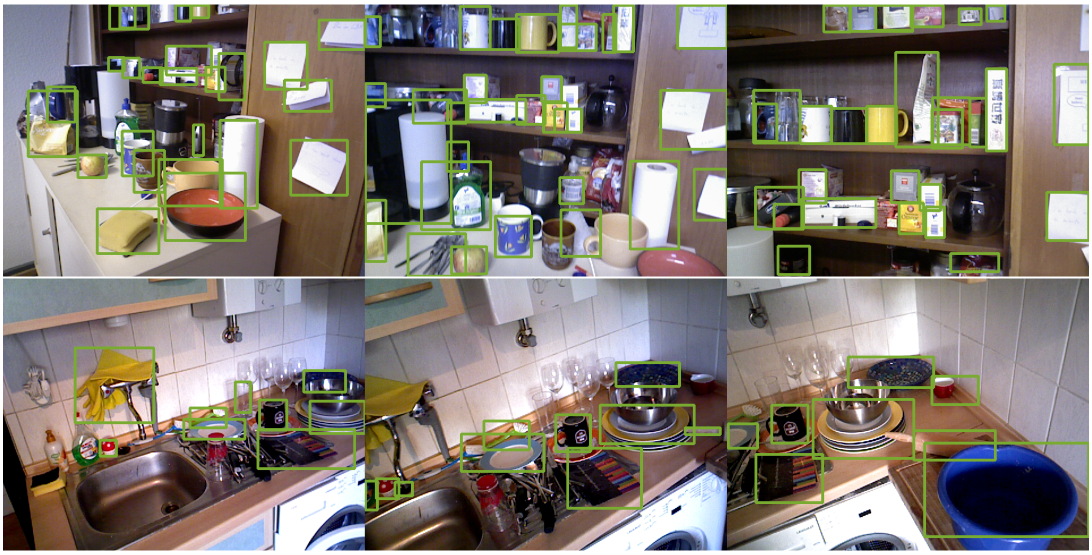


Figure 4.25.: Top row: successful candidates using the S2-M4 method in the Coffee Machine Sequence. Bottom row: successful candidates using the S2-M4 method in the Kitchen A Sequence of the Kitchen Dataset.

In the second part of the evaluation, we fixed the ranking method to R3 and evaluated 8 different variants of our proposal generation method: all the possible combinations of the two salient region extraction methods, S1 (single saliency), and S2 (multi-scale saliency), and the four segmentation methods: M1 (colour), M2 (surface clustering), M3 (early fusion of colour and depth), and M4 (late fusion of M1 and M2).

Segmentation The results showed that the late fusion approach (M4) is in general superior to the other segmentation methods in every aspect that we evaluated: quality of the proposals, frame recall, precision and global recall. We extract two conclusions from this: first, that colour and depth are complementary, and there are objects that are most easily found by relying on either colour or depth; and second, that the segmentation of colour and depth independent of each other (M4) gave better results than the segmentation integrating both modalities (M3). An explanation of this result is the way the M3 method produces segments that are homogeneously spread over the scene: a flat surface, or a homogeneous texture end up being partitioned although according to one of the two modalities they should not; and this results in parts of the objects missing in the generated proposals: M2 would produce one segment for each face of a cereals box, making it easy for saliency to grab those segments; however, in M3 each face of the box consists itself of several segments, some of which might be easy to miss (see the results in Fig.4.19). This results in lower recall values and lower quality of the

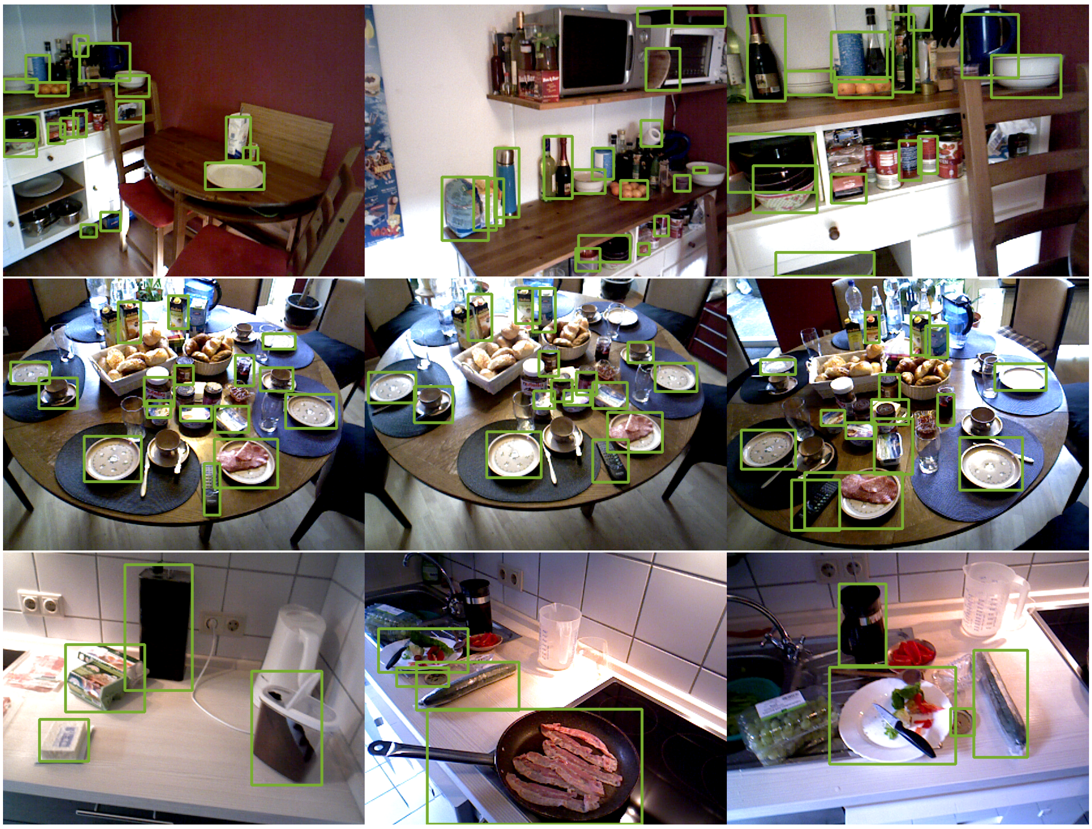


Figure 4.26.: Successful candidates using the S2-M4 method in the Kitchen B, C and D sequences of the Kitchen Dataset.

proposals as was reflected throughout the evaluation.

Salient Region Extraction Regarding the two approaches for extracting salient regions (S1 and S2), the results showed that in presence of clutter, using the multi-scale approach (S2) meant a significant boost in the recall and the quality of the proposals with respect to the single saliency one (S1). Also, the quality of the proposals improved when using S2 with respect to S1 (see the recall vs IoU plots throughout the evaluation).

Among the competitors, the method of [Potapova et al., 2014] proved to generate few but reliable object candidates (relatively high precision values in the plots), and the method of [Manén et al., 2013] showed to generate a rich variety of object proposals which let them reach high global recall values throughout the evaluation. The Selective Search [Uijlings et al., 2013] was the most successful method among the competitors, obtaining proposals of very good quality, being able to reach high recall on the frame and sequence level.

Our methods were successful in finding the objects in realistic scenes containing high clutter, and, especially S2-M4, proved to outperform other state-of-the-art methods in every aspect that we evaluated (particularly for a small number or proposals, e.g. 50): in terms of the objects that are recalled per frame (especially when the first few candidates are considered), in terms of the quality of the generated proposals, and in terms of the objects that are discovered throughout the sequence (global recall).

List of Own Publications for this Chapter

[a] Germán Martín García, Ekaterina Potapova, Thomas Werner, Michael Zillich, Markus Vincze and Simone Frintrop. *Saliency-based Object Discovery on RGB-D Data with a Late-Fusion Approach.* In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).* Seattle, USA, 2015.

[b] Esther Horbert, Germán Martín García, Simone Frintrop and Bastian Leibe. *Sequence-Level Object Candidates Based on Saliency for Generic Object Recognition on Mobile Systems.* In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).* Seattle, USA, 2015.

5. Scene Exploration: Inhibition of Return (IOR)

In Chapter 4, we have dealt with the problem of generating object candidates based on visual data, and we proposed a method for generating such candidates from single images. However, mobile systems such as a robot, that need to interact with the environment and make sense of it, are not exposed to single images but to a continuous stream of them.

As we described in the chapter about object candidate generation, our primary way of localising object candidates in images is by means of a bottom-up attention system (saliency). In this chapter, we will describe an extension of the attention system to inhibit object candidates that have already been the target of attention. This is a trivial question if we have a single image, but requires establishing correspondences between visual elements over time if we are dealing with a sequence of frames. This extension will let us visually explore the scene with a few candidates generated on every frame.

An overview of the architecture we propose is depicted in Fig. 5.1. A camera mounted on a mobile system captures a stream of RGB-D data. In the lower processing stream, the depth information is used to build a 3D map of the scene with the KinectFusion algorithm [Newcombe et al., 2011]. In the upper stream, an attention system computes a saliency map (1.) and a segmentation of the image is obtained (2.). Based on these two, object candidates are generated (3.). Information about already attended objects is stored in the 3D map, raycasted to the current camera pose (4.), and used to inhibit already attended objects (5.). Steps 1., 2. and 3. were explained in Chapter 4 (our method for object candidate generation). The focus of this chapter will be on the other steps: how to remember which objects were already the focus of attention, and how to inhibit them.

Relevant publications The publications relevant for this chapter are [Martín García and Frintrop, 2013] and [Martín García et al., 2013], where we presented our spatial IOR mechanism. In this chapter, we have improved the original IOR system to directly inhibit the object candidates that were generated with our

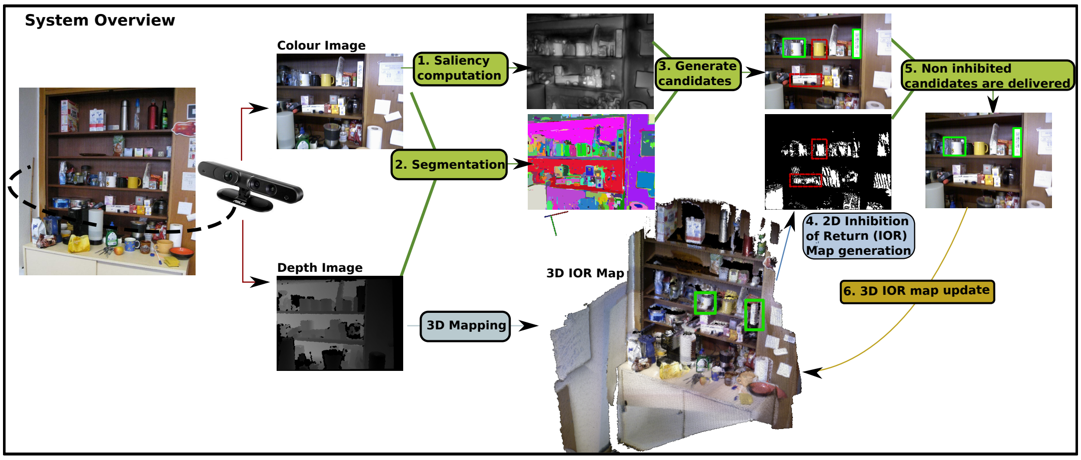


Figure 5.1.: Illustration of the whole scene exploration system.

newer object discovery method. The content of this chapter has been submitted to the Cognitive Processing Journal [Martín García et al., 2015a].

5.1. Inhibition of Return in Spatial Coordinates

How to shift the focus of attention is a classical problem in computational attention systems. Always choosing the most salient region as the focus of attention would make an attention system to always select the global maximum as the target of attention. As in human vision [Posner et al., 1985], computational IOR helps exploring a scene by inhibiting those regions that have already been attended. When working on single images it is often performed by simply zeroing the region of the saliency map that was already attended [Itti et al., 1998]. However, this is not enough when facing a sequence of frames from a given scene where correspondences between the visual elements should be established. Our approach roots the IOR mechanism in spatial coordinates in order to cope with camera motion; and for that, it first needs to build a 3D map of the scene.

5.1.1. KinectFusion

The KinectFusion algorithm [Newcombe et al., 2011] is a method for reconstructing a 3D scene by means of a sequence of depth measurements obtained from a moving camera. The algorithm works by iteratively tracking the pose of the camera with respect to the reconstructed model of the scene, and integrating the new depth measurements into the model. An open source implementation of the algorithm is available in the PCL library.[1] The method produces a very precise reconstruction of the environment

[1] http://pointclouds.org/

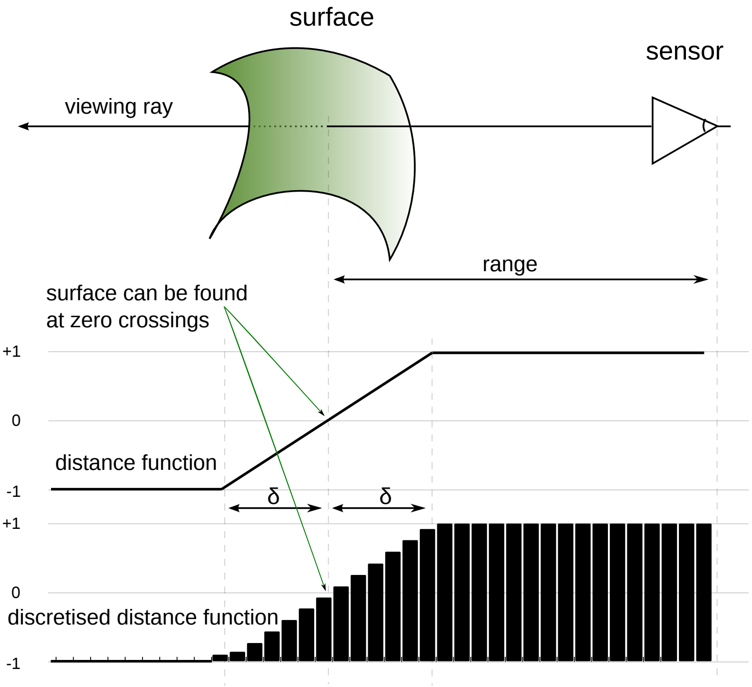


Figure 5.2.: Illustration of a truncated signed distance function. Inspired by the figure in [Pirovano, 2012].

and works in real time as long as a powerful enough GPU is available. In the following we describe some of the elements of KinectFusion that are of relevance for our purposes:

Scene Representation The scene is represented as a discretised version of a truncated signed distance function (TSDF) [Curless and Levoy, 1996]. In general, a truncated signed distance function is a function of the range that assigns 0 values to points in space where there is a surface, positive values to points before the surface, and negative values beyond the surface. See Fig. 5.2 for an illustration: as we come closer from the surface towards the sensor, the values of the TSDF increase up to a certain distance (δ) where they do not increase any more; the opposite happens in the other direction away from the surface.

The discretised TSDF (see the lower part of Fig.5.2) has the form of a voxel grid $G \subset \mathbb{N}^3$, where every voxel $\mathbf{c} \in G$ stores the actual distance to the closest surface F_k and a weight W_k, which is proportional to the surface measurement uncertainty, at each point in time k: $S_k[\mathbf{c}] \rightarrow \{F_k[\mathbf{c}], W_k[\mathbf{c}]\}$. Surface points can now be found by looking at zero crossings. The TSDF scene representation is kept in global coordinates. Integrating new measurements is easily done by a running weighted average method, and more importantly, it can be parallelised to make it run in real time on a GPU.

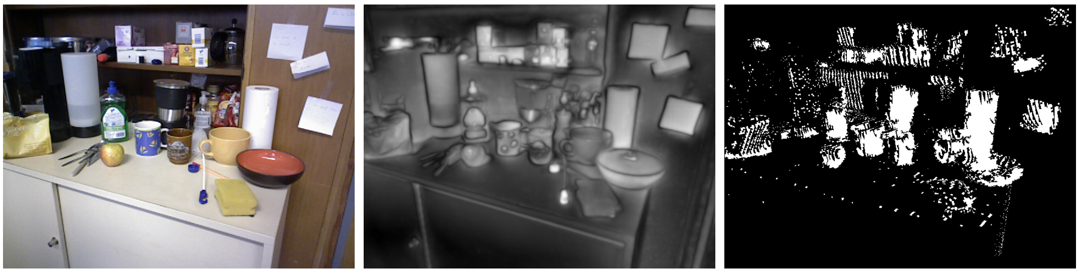


Figure 5.3.: Left: original frame from the Coffee Machine Sequence. Middle: the saliency map. Right: the raycasted 2D IOR Map.

Camera Pose Tracking The camera pose is tracked on every frame with respect to the previous one/existing model. For this, the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992] is used. ICP requires a coarse initialisation in order to converge to a correct solution. Here, the authors assume that the algorithm is operating at a high frequency (30 Hz) and so the camera displacements between frames are small.

Raycasting Depth Maps To track the pose of the camera, a raycast depth map from the scene model can be used instead of the raw depth map of the previous frame to be more robust against noise. The algorithm for raycasting a depth map given a TSDF and a camera pose proceeds by tracing rays from each pixel according to the given pose; when a ray goes through a zero crossing in the TSDF model, it means it has touched a surface. Therefore, the distance of that voxel to the camera is the range measurement that will be displayed in the raycast map.

5.1.2. KinectFusion 3D IOR Map Extension

We extend the KinectFusion voxel grid in order to store the IOR information. In particular, we want to store whether a point in space should be inhibited (an IOR flag, I_k), and for how long (an IOR weight, IW_k). Thus, each voxel $\mathbf{c}$ in the grid G now stores the following values:

$$S_k[\mathbf{c}] \rightarrow \{F_k[\mathbf{c}], W_k[\mathbf{c}], I_k[\mathbf{c}], IW_k[\mathbf{c}]\}. \tag{5.1}$$

This extended voxel grid is what we will refer to in the following as the 3D IOR map. For a given camera pose, the TSDF can be raycast to generate a depth map indicating the depth ranges at which surfaces are present. Since now we have additionally IOR information stored in the voxel grid, we can easily raycast the IOR flags I_k at the surfaces to produce a 2D IOR map (see an example of such a map on the right of Fig. 5.3).

5.1.3. 3D IOR Map Update

We now have a data structure where we can store when a particular region of the scene has been attended, whether it should already be inhibited, and for how long. Initially, the scene has not yet been explored and all its regions could potentially be the target of attention. Thus, the IOR weights and flags are set to zero for every voxel $\mathbf{c}$ in the grid G:

$$I_0[\mathbf{c}] = 0, \quad IW_0[\mathbf{c}] = 0, \quad \forall \mathbf{c} \in G. \tag{5.2}$$

Then, as the system is exposed to more frames of the sequence, object candidates are produced (Chapter 4). For each frame, the pixel-precise masks of the object candidates can be projected to the 3D IOR map in order to obtain the voxels that should be updated: let us call this set of voxels A. The IOR weights of the corresponding objects' voxels are increased by one:

$$IW_k[\mathbf{a}] := IW_k[\mathbf{a}] + 1, \quad \forall \mathbf{a} \in A. \tag{5.3}$$

When the IOR weight $IW_k[\mathbf{a}]$ eventually reaches a certain threshold *IOR_LIMIT*, the IOR flag is activated, $I_k[\mathbf{a}] = 1$, and the IOR weight is reset to a multiple mf of its value: $IW_k[\mathbf{a}] := IW_k[\mathbf{a}] \cdot mf$. This means that once the IOR activation threshold is reached, it will take more time for the inhibition to die out than it took to reach it. This is done to prevent the inhibition effect from quickly vanishing and the attention being allocated again on the same objects. Meanwhile, the IOR weights of the voxels that were not part of any object candidate are decreased by one:

$$IW_k[\mathbf{z}] := IW_k[\mathbf{z}] - 1, \quad \forall \mathbf{z} \in G - A. \tag{5.4}$$

When the weights reach zero, the IOR flag is again deactivated: $I_k[\mathbf{z}] = 0$. To sum up, regions in space that are the target of attention (i.e., those for which object candidates are generated) increase their IOR weight, and those that are not, decrease them. The IOR weight evolution is depicted in Fig. 5.4 and the 3D IOR map update procedure is illustrated in Fig. 5.5.

5.1.4. 2D IOR Map

In order to use the inhibition information within our attention system, we need to obtain a 2D map from the 3D data. Since our 3D IOR Map is embedded in the voxel grid of KinectFusion, it is possible to raycast a 2D IOR map $IOR(x, y)$ for any given camera pose. The result of such an operation is a binary map containing white pixels ($IOR(x, y) = 1$) for spatial locations $\mathbf{c} \in G$ that should be inhibited (visual regions

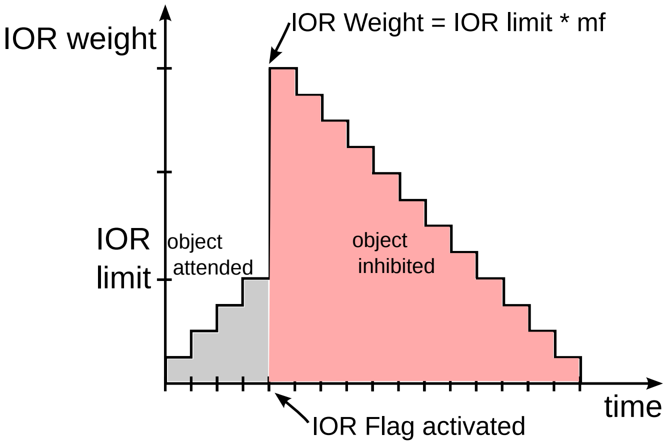


Figure 5.4.: Depiction of how the IOR weight evolves over time on a given voxel that has been attended enough frames to activate the IOR flag.

corresponding to spatial locations where $I_k[\mathbf{c}] = 1$), and black pixels ($IOR(x, y) = 0$) for those that should not. We show in Fig. 5.3 an example image (left), together with the saliency map (middle) computed from it, as well as the 2D IOR map that has been raycasted (right). The white pixels in the 2D IOR map indicate the locations where attention has been allocated up to that point in time. In principle, such a 2D IOR map can be used to inhibit points or regions in the saliency map. We show in Section 5.1.5 how we use it to directly inhibit the object candidates.

5.1.5. Inhibition of Object Candidates

At this point we can use the 2D IOR map to inhibit those candidates that correspond to regions that have already been attended. To decide which candidates to inhibit, we simply compute the intersection of the 2D IOR map $IOR(x, y)$ and the ith object candidate binary mask C_i as: $Z(x, y) = IOR(x, y) \cap C_i(x, y)$ —see Chapter 2 for more details about logical operations between binary images. We inhibit an object candidate if a certain percentage of its pixels are marked as inhibited in the 2D IOR map:

$$\frac{\sum_{x,y} Z(x, y)}{\sum_{x,y} C_i(x, y)} \geq \theta, \tag{5.5}$$

we set $\theta = 0.3$ in our experiments.

5.2. Evaluation

In this section, we evaluate the IOR mechanism on the Coffee Machine Sequence (see Section 4.6.1) in terms of how well it serves for visual scene exploration: our purpose

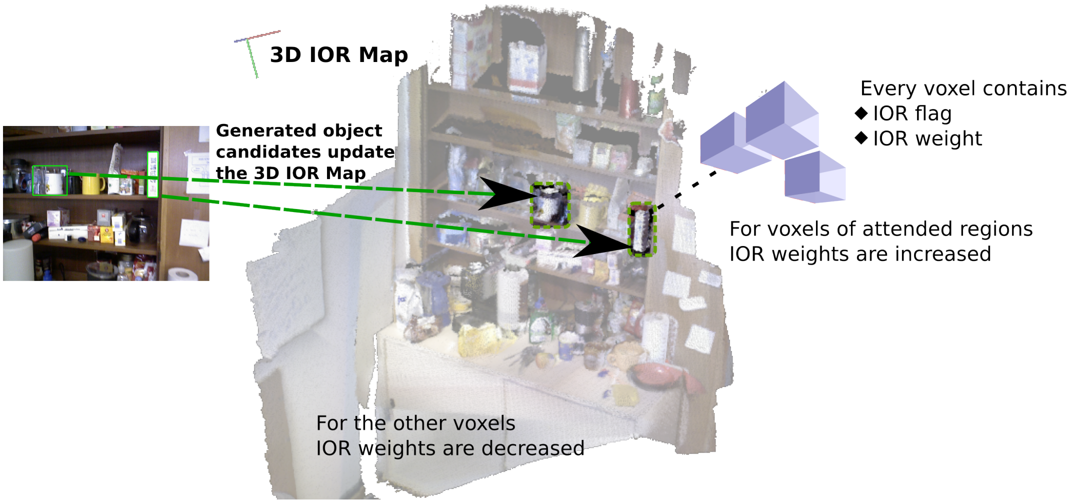

3D IOR Map
Generated object candidates update the 3D IOR Map
Every voxel contains
◆ IOR flag
◆ IOR weight
For voxels of attended regions IOR weights are increased
For the other voxels IOR weights are decreased

Figure 5.5.: 3D IOR Map update process: the object candidates generated on a particular frame are projected to the 3D map. At the voxels in the map corresponding to the object candidates, the IOR weights are increased. Everywhere else, the IOR weights are decreased.

is to show that with a few object candidates per frame we can still detect most of the objects in the scene by the end of the sequence. Thus, we constrain our object discovery method to produce a very small number of object candidates in the S1-M4 mode (single saliency map, late fusion of colour and depth), ranked according to the SVM score (Chapter 4, Section 4.5.3). An overview of the steps of the object discovery algorithm used in the evaluation is shown in Alg. 5. The generated object candidates will activate the IOR flags in the 3D map (Section 5.1.3). This will have the effect that in the following frames these regions will be inhibited and object hypotheses will be generated at other locations in the scene.

5.2.1. Ground Truth Annotation

We manually annotated our Coffee Machine Sequence on every 30th frame. That means, we created grayscale masks for every 30th frame, where every object kept a consistent grayscale level (or ID) throughout the sequence. This was already enough to evaluate our object candidates in Section 4.6, however, in order to test the IOR mechanism we require ground truth available in every frame: the IOR mechanism takes place from frame to frame, and so, its effect would be “lost” if we evaluated the results on every 30th frame.

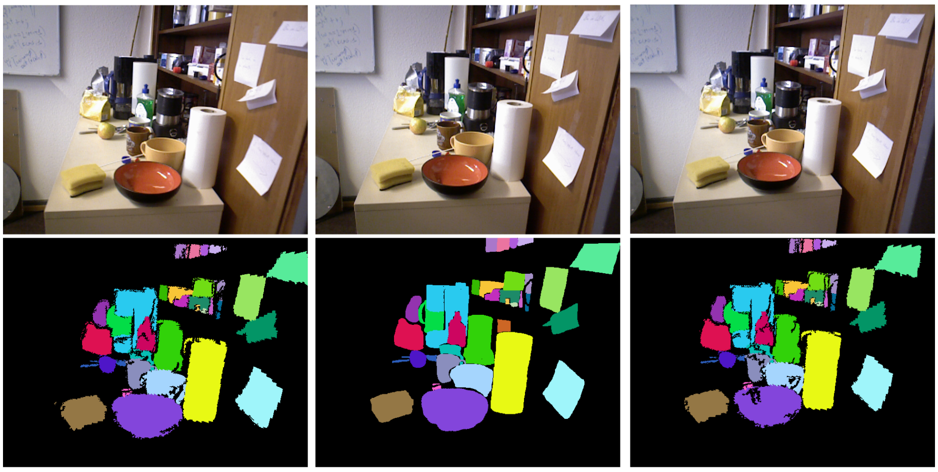
We developed a method to automatically propagate the sparse ground truth annota-

Algorithm 5 S1-M4 IOR Generate Salient Object Proposals

```
procedure S1-M4-IOR-GenerateProposals
Input: Image $I$, Depth map $D$
Input: A 2D IOR Map $IOR$
Input: A top number of object proposals $t$
Output: A sequence of sorted object proposals $(C'_1, ..., C'_t)$
    $R :=$ S1-ExtractSalientRegions(I)
    $S_1 = \{s^1_1, ..., s^1_{m_1}\} :=$ over_segment(I,“M1”)
    $S_2 = \{s^2_1, ..., s^2_{m_2}\} :=$ over_segment(D,“M2”)
    for $r_i \in R$ do
        for $s^1_j \in S_1$ do
            if $|r_i \cap s^1_j| > \gamma \cdot |s^1_j|$ then
                $C_i := C_i \cup \{s^1_j\}$
        if $\frac{|IOR \cap C_i|}{|C_i|} \leq \theta$ then
            $C := C \cup \{C_i\}$
        for $s^2_k \in S_2$ do
            if $|r_i \cap s^2_k| > \gamma \cdot |s^2_k|$ then
                $C_{n+i} := C_{n+i} \cup \{s^2_k\}$
        if $\frac{|IOR \cap C_{n+i}|}{|C_{n+i}|} \leq \theta$ then
            $C := C \cup \{C_{n+i}\}$
    $(C'_1, ..., C'_t)$ = SVM-Ranking(C)
```

tions to the unlabelled frames. The method proceeds as follows. We run the KinectFusion algorithm a first time in order to build the 3D map of the scene. The idea is that we want to use every annotated frame to generate interpolated ground truth for the closest frames before and after it. For example, manually annotated ground truth frame 90 will be used to automatically generate the ground truth of frames 75 to 105. Thus, we run KinectFusion another two times: once backwards, generating ground truth for the 15 frames before every annotated one; and once forwards, generating the ground truth of the 15 frames following every annotated one.

So, for every frame for which ground truth exists, we project the annotated ground truth masks to the 3D map, and store the object labels in the corresponding voxels. For every frame for which no ground truth exists, the object labels are raycasted according to the current camera pose to form a raycasted ground truth map. The results of this method can be seen in Figure 5.6 for some frames of the sequence.



Frame 20
(GT interpolated)

Frame 30
(GT manually annotated)

Frame 40
(GT interpolated)

Figure 5.6.: Manually annotated and interpolated ground truth for the Coffee Machine Sequence.

5.2.2. Results

The evolution of the IOR experiment is shown in Fig. 5.7. There, we show on the upper row the successful candidates generated by the object discovery system for some non-consecutive frames: we used our S1-M4 method constrained to the best 20 candidates per frame, and with the IOR mechanism active (Alg. 5). On the bottom row we show the projected 2D IOR map corresponding to those frames. The red arrows depict candidates that have activated the inhibition values enough so that the same candidates are not generated again in the following frames. See, for example, the yellow cup, which is a candidate in the first displayed frame, until the IOR map has enough activation to inhibit it (it is not a candidate in the second frame). Eventually, the IOR effect dies out and, after some frames, the yellow cup is generated again as a candidate (third column of Fig. 5.7).

Multiple Factor Parameter Evaluation

We show the results of our experiments in terms of global recall over time in Fig. 5.8. Here, we are interested in seeing how many objects of the scene we are able to retrieve with as few object candidates as possible. First, we show the effects of altering the multiple factor parameter mf, *i.e.*, the parameter that controls how long the IOR effect lasts. Higher values for this factor have the effect that once the IOR activation value is reached, it takes longer to die out (see Section 5.1.3 for details). We show the results obtained when generating 20 candidates per frame for our S1-M4 object

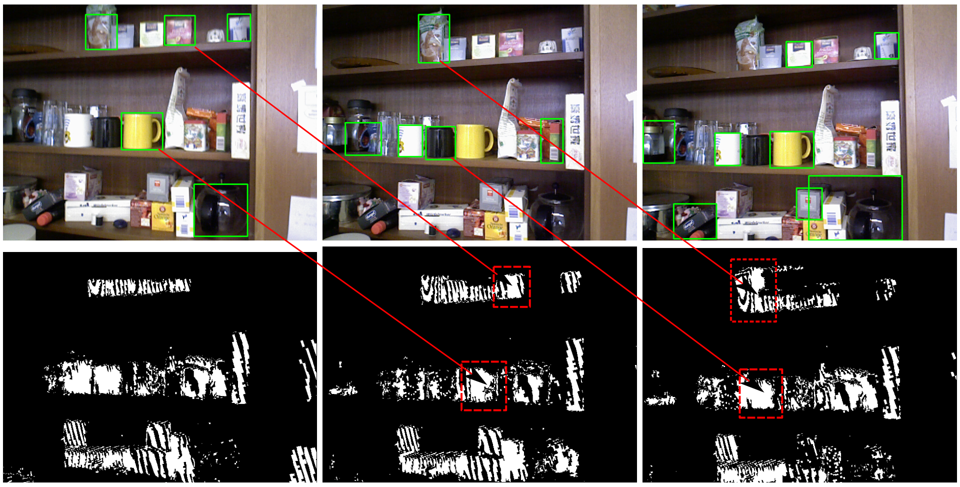


Figure 5.7.: Illustration of the IOR experiment: on top, some frames from the CMS; the green bounding boxes show the candidates that successfully matched an object out of 20 generated candidates per frame. The bottom row shows the 2D IOR map at those frames. The red arrows depict candidates that have been attended long enough to activate the inhibition flags.

discovery method, for three different values of mf: 2, 4 and 6 (green, black and violet curves respectively in Fig. 5.8). The global recall values achieved were 86% for $mf = 2$ and $mf = 4$, and 91% for $mf = 6$.

The frame based precision and recall plots are shown in Fig. 5.8. In terms of precision, it can be observed how increasing mf lowers precision. This has an obvious explanation: if the first ranked candidates correspond to actual objects and they are inhibited, the method will look further in the list to retrieve candidates with a lower ranking; the longer the inhibition effect lasts, the fewer "good" candidates will be retrieved on a frame basis. This, on the other hand, causes the global recall over time to increase (as we saw in Fig. 5.8), since a wider variety of objects is explored.

IOR vs. No-IOR

In the second part of this set of experiments, we compare the results of running our object discovery method with IOR and without it. We chose the method that performed best in terms of global recall in the previous experiment: 20 candidates per frame method with IOR and $mf = 6$. We compare it to two different configurations without IOR: generating 20 candidates (red curve) and 255 candidates per frame (blue curve). As the results show (top plot of Fig. 5.9), fixing the number of candidates on 20, using the IOR mechanism makes a big difference in terms of global recall: 91% with IOR

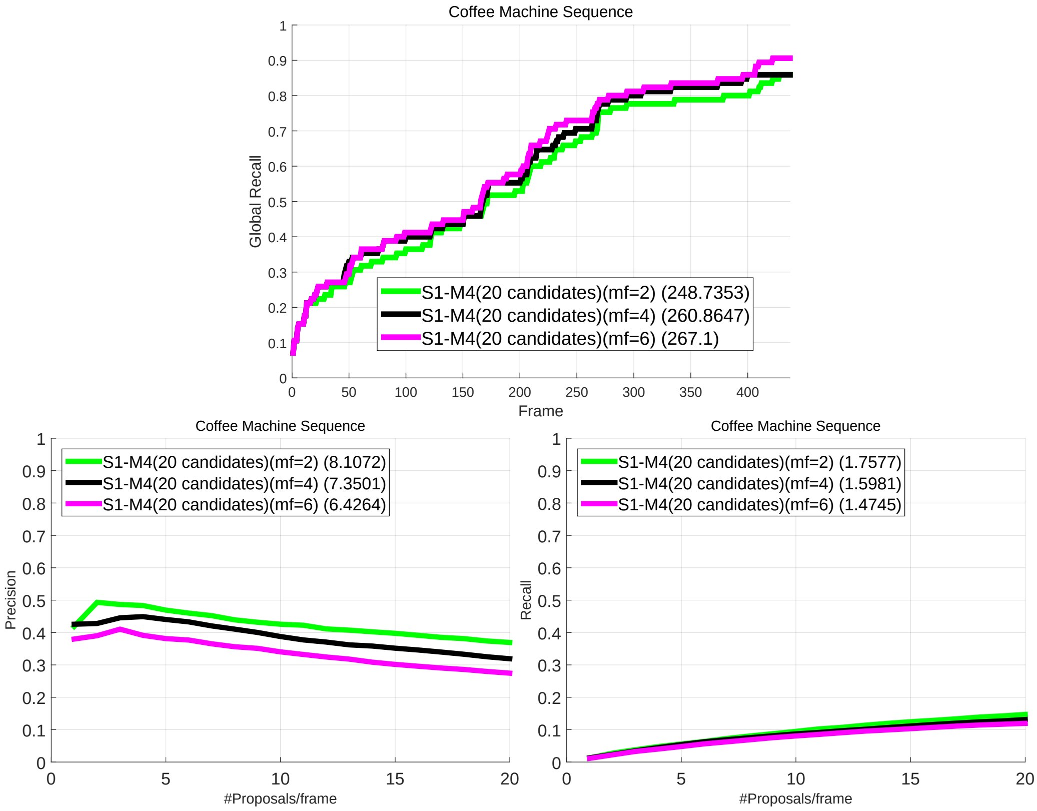


Figure 5.8.: Top: Global recall over time in the CMS. Bottom left: precision over number of proposals. Bottom right: recall over number of proposals. Comparison of different values for the IOR factor mf: green ($mf = 2$), black ($mf = 4$) and violet ($mf = 6$).

(value reached by the violet curve) vs 73% without IOR (value reached by the red curve). Furthermore, the results for 20 candidates with IOR were only 2% behind with respect to the method generating 255 candidates per frame and no IOR (global recall of 93%).

For completeness, we show in Fig. 5.9 the precision and recall plots over the number of proposals for our object discovery method with and without the IOR mechanism. As expected, because good candidates are inhibited for a certain time from being generated, the precision and recall curves (in violet) on a frame level are consistently below the curves without the IOR mechanism (blue and red).

Conclusion

The results show that, by using the IOR mechanism, we can rely on a much smaller number of candidates per frame (20 as opposed to 255) and yet retrieve most of the

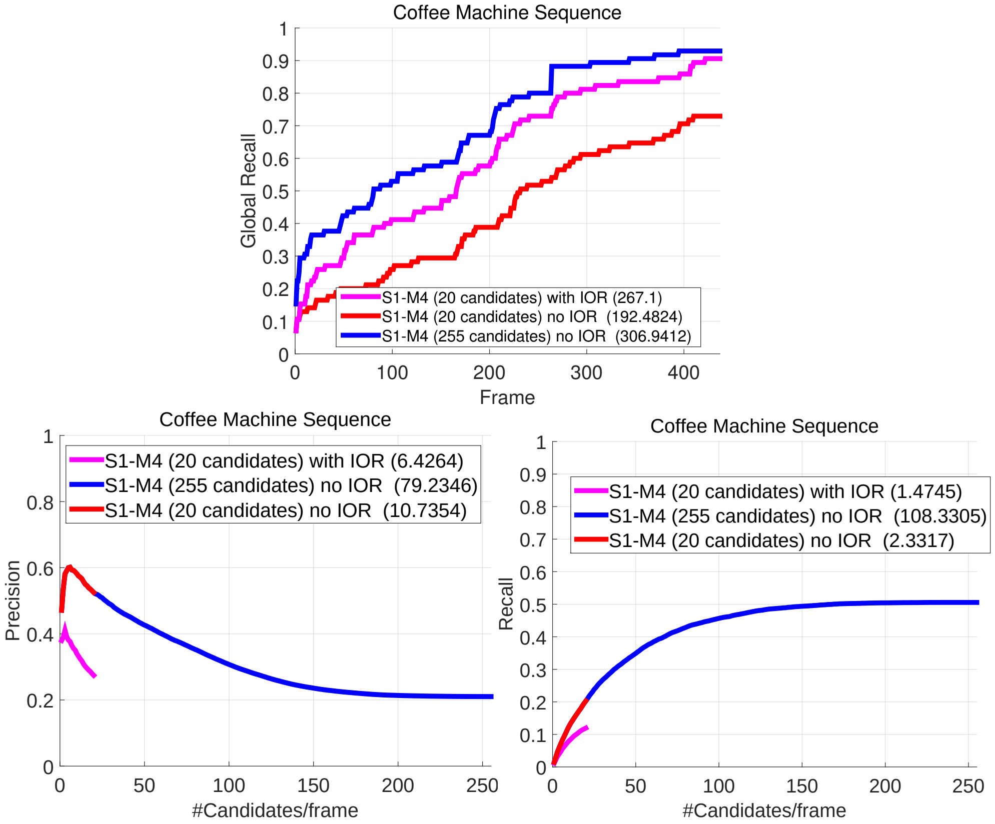


Figure 5.9.: Top: global recall over time. Bottom Left: precision over number of proposals. Bottom right: recall over number of proposals. Violet curve: 20 candidates per frame using IOR. Red curve: 20 candidates per frame not using the IOR mechanism. Blue curve: 255 candidates per frame without IOR.

objects in the scene (91%). A small number of candidates is beneficial because it means less queries for recognition, and particularly for robotic applications were interactions with the potential objects might be required.

List of Own Publications for this Chapter

[a] Germán Martín García, Mircea Pavel and Simone Frintrop. *A Computational Framework for Attentional Object Discovery in RGB-D Videos.* Submitted to: *Cognitive Processing Journal.*

[b] Germán Martín García, Simone Frintrop and Armin B. Cremers. *Attention-Based Detection of Unknown Objects in a Situated Vision Framework.* In: *KI - Künstliche Intelligenz, Springer, 27 (3), 2013.*

[c] Germán Martín García and Simone Frintrop. *A Computational Framework for Attentional 3D Object Detection.* In: *Proc. of the Annual Conference of the Cognitive Science Society.* Berlin, Germany, 2013.

6. Salient Object Segmentation: Proposal Maps

In this chapter, we apply our basic algorithm for generating object candidates (Chapter 4) to the task of salient object segmentation. This is the task of determining the most salient object[s] in a given image. See for example the left image in Fig. 6.1: the man performing martial arts has been labelled by some person as the most salient "object" in the scene. In fact, this is the way benchmarks are created in this task: by compiling a set of hundreds or thousands of pictures and showing them to several subjects in order to label what they consider the most salient object is.

Segmenting the most salient object is a task where saliency systems are typically evaluated. As we have seen in the description of the VOCUS2 saliency system —Section 4.1.1—, the computation of saliency is essentially a center-surround contrast computed in a scale-space representation. A popular and recent trend has been to incorporate segmentation into this process [Perazzi et al., 2012, Yan et al., 2013, Zhu et al., 2014]. Our goal in this chapter is to extend the VOCUS2 saliency system with segmentation; for that, we will transform our set of objects proposals (which carry segmentation information) into a single saliency map.

Relevant publications The content of this chapter is based on one publication: [Frintrop et al., 2015], where we contributed with a method that includes segmenta-

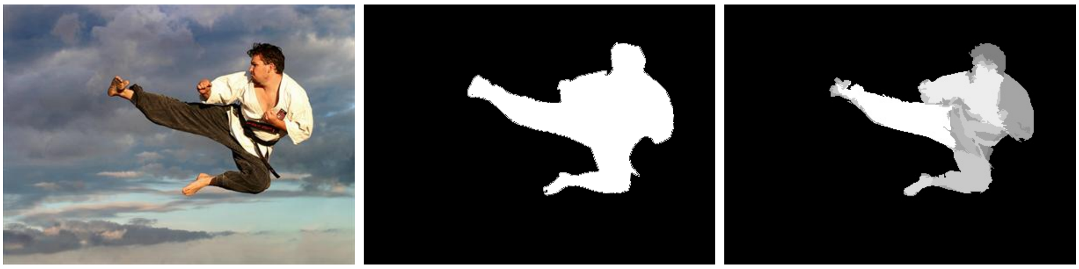


Figure 6.1.: Salient object segmentation example. Left: an example image. Middle: the corresponding ground truth where the most salient object has been manually labelled. Right: our salient segmentation.

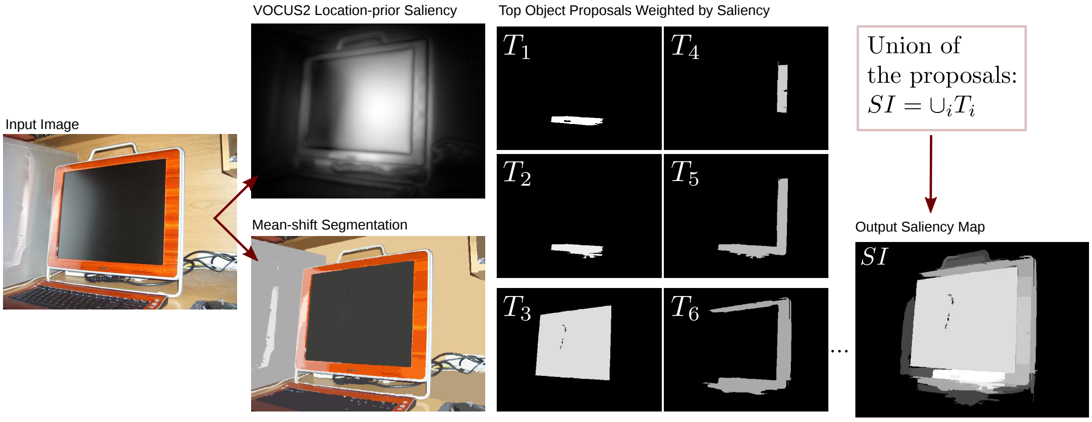


Figure 6.2.: The main algorithm for generating proposal maps

tion to improve the saliency maps.

6.1. Proposal Map Generation

We propose a method that combines the VOCUS2 saliency maps and a minor modification of our object proposal generation algorithm (described in Chapter 4) to produce saliency maps that incorporate segmentation information. We follow the approach of [Li et al., 2014] in order to combine several object proposals into one single saliency map.

Our approach is based on the S1-M1 object candidate generation method explained in Chapter 4 —S1 for extracting salient regions from a single saliency map (here, weighted with a location prior), and M1 for colour segmentation. We sketch the main steps in Fig. 6.2: For a given image, we compute the VOCUS2 saliency map, where we follow the single-saliency map salient region extraction strategy (S1) and compute a colour segmentation of the image (M1). The saliency map has been weighted with a location prior that strengthens central regions of the image by means of a Gaussian function. This is a common practise in saliency systems when they are evaluated in salient object segmentation benchmarks (e.g. [Jiang et al., 2011, Yan et al., 2013]), since the images typically have a bias to contain the most salient object in the centre region. Thus, for an initial saliency map $Sal(x, y)$ at pixel coordinates x and y, the weighted saliency map $Sal'(x, y)$ is computed as

$$Sal'(x, y) = Sal(x, y) \cdot \exp\{-\frac{||(x, y) - (x_c, y_c)||^2}{2\sigma^2}\}, \tag{6.1}$$

where (x_c, y_c) are the coordinates of the image center, and $\sigma = 79$ pixels is the standard deviation of the Gaussian. The weighted saliency map $Sal'(x, y)$ is finally normalised, and so the constant term of the Gaussian can be ignored.

The way of refining the salient regions by means of segmentation is the same as in the proposal generation method of Section 4.3, except that we use the Mean Shift algorithm [Comaniciu and Meer, 2002]. The main steps of the algorithm are sketched in the following pseudocode:

Algorithm 6 Generate Segment-based Saliency Map

```
procedure GenerateSegmentSaliency
Input: Image I
Input: Top number of proposals $t$
Output: A saliency map $SI$
    Compute location-prior saliency map $Sal'(x,y)$ on the image $I$
    $\{C_1,...,C_{2n}\}$ := GenerateProposals($I$,$Sal'$)
    Compute avg. saliency of each proposal: $\{\overline{sal}_1,...,\overline{sal}_{2n}\}$
    Rank the proposals according to their saliency: $(C'_1,...,C'_{2n})$ := RankSaliency($(C_1,...,C_{2n})$,$\{\overline{sal}_1,...,\overline{sal}_{2n}\}$)
    for $i = 1$ to $t$ do
        $T_i := C'_i \cdot \overline{sal}_i$
    for pixel coordinates $x, y$ do
        $SI(x,y) := max((T_1(x,y),...,T_t(x,y)))$
```

In the call to the GeneratePrOPOSALS procedure we pass the location-prior saliency map as an argument, $Sal'(x,y)$, to indicate that we use it also in the proposal generation process. The proposals are now ranked according to their average saliency: $(C'_1,...,C'_n)$. Since each proposal C'_i is a binary map, we can compute its corresponding proposal-specific saliency map T_i as

$$T_i = C'_i \cdot \overline{sal}_i. \tag{6.2}$$

Finally, to compute the output saliency map SI, the saliency at each pixel location, $SI(x,y)$, is defined as the maximum saliency value obtained at the corresponding location in each proposal-specific saliency map:

$$SI(x,y) := max((T_1(x,y),...,T_t(x,y))) \tag{6.3}$$

Two thresholds are applied in this process: first, we take the top $t = 25$ proposals, and second, we discard those whose average saliency is below one third of the highest proposal saliency.

6.2. Evaluation

In this section, we evaluate the performance of our proposal maps (denoted as V2-Prop in the following plots) in the task of salient object segmentation. We have compared our salient object detection method on the MSRA 10k [Cheng et al., 2015], ECSSD [Yan et al., 2013], PASCAL-S [Li et al., 2014], SED1 and SED2 datasets [Alpert et al., 2007] with the following saliency methods: Itti's iNVT [Itti et al., 1998], the SaliencyToolbox (STB) [Walther and Koch, 2006], HZ08 [Hou and Zhang, 2008], AIM [Bruce and Tsotsos, 2009], AC09 [Achanta et al., 2009], AC10 [Achanta and Süsstrunk, 2010], CoDi [Klein and Frintrop, 2012], HSaliency [Yan et al., 2013], Yang 2013 [Yang et al., 2013] and DRFI [Jiang et al., 2013]. We additionally show the results of the VOCUS2 method (V2) and VOCUS2 with location prior (V2-LP).

6.2.1. Metrics

We use two different metrics throughout the evaluation to compare the performance of the salient object detectors: the first one is the method of [Achanta et al., 2009], which has been the most used technique in the literature. The second one is a complementary measure called the Weighted F-measure, which as we will see, attempts to solve some flaws of the Achanta metric.

Achanta Method [Achanta et al., 2009] This method works by binary thresholding the saliency maps at all the possible levels: 0 to 255, and then calculating the intersection with the ground truth. From this intersection, four values are possible for each pixel: TP (true positive), if the pixel is 1 and the corresponding ground truth is also 1; TN (true negative), if the pixel is 0 and the ground truth is also 0; FP (false positive), if the pixel is 1 and the ground truth is 0; and FN (false negative), if the pixel is 0 and the ground truth is 1. From these four quantities, we can compute precision and recall as

$$precision = \frac{TP}{TP + FP}, \tag{6.4}$$

$$recall = \frac{TP}{TP + FN}. \tag{6.5}$$

To plot the actual curve, recall r is sampled at regular intervals and a precision value $p(r)$ is interpolated as $p(r) = \max_{\hat{r}:\hat{r}>r} p(\hat{r})$, i.e., the maximum precision obtained at higher recall levels. In this way, precision and recall values can be computed for each

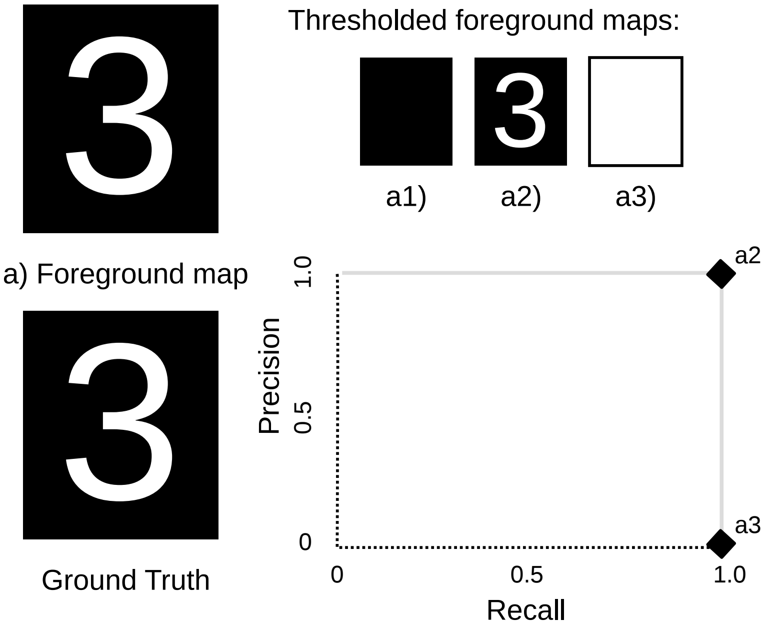


Figure 6.3.: Achanta method [Achanta et al., 2009] for evaluating foreground maps. The foreground map is thresholded at all possible grayscale values [0,255], obtaining three different thresholded maps: a1), a2) and a3). Precision and recall values are obtained for each thresholded map. Precision values for non-existing points in the plot are interpolated. In this case, since the foreground map is the same as the ground truth, the obtained precision-recall curve is the best possible. Figure inspired from [Margolin et al., 2014].

threshold and each image. The values can be averaged and finally plotted in a precision-recall curve that shows precision as a function of recall. An example of this process is depicted in Fig. 6.3.

Weighted F-measure In [Margolin et al., 2014], the authors identified three problems with the previous metric:

- Interpolation flaw: the method of interpolating precision values for every recall makes lower precision values not to affect the final result if a high precision value is obtained for that recall. See Fig. 6.4 for a visualisation: even though the foreground map a) is much better than map b), they both obtain the same precision-recall curve.

- Dependency flaw: false positives that are scattered among true positives should obtain better results than when being concentrated on certain parts: in the latter, complete parts of the foreground will be missing.

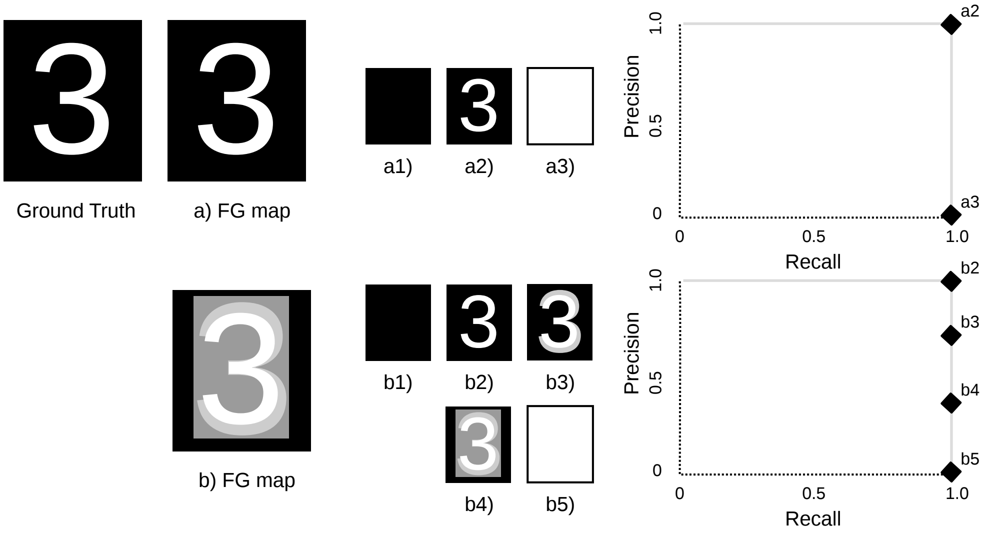


Figure 6.4.: Interpolation flaw: foreground map b) is worse than map a) but both obtain the same precision-recall curve. Figure inspired from [Margolin et al., 2014].

- Equal-importance flaw: false positives are preferable if they are next to true positives. The Achanta metric does not make a distinction between false positives. However, one sort of foreground map is preferable to the other.

We illustrate the three flaws in Fig. 6.5.

The new metric works in the following way. First, the interpolation flaw, which is the result of thresholding the foreground maps (D) into binary maps, is solved by redefining TP, FP, TN and FN in terms of the ground truth map (G) as

$$TP' = D \cdot G, \tag{6.6}$$

$$TN' = (1 - D) \cdot (1 - G), \tag{6.7}$$

$$FP' = D \cdot (1 - G), \tag{6.8}$$

$$FN' = (1 - D) \cdot G. \tag{6.9}$$

By using this redefinition no thresholding operation needs to be done in the maps, thus, solving the interpolation flaw. The dependency and the equal-importance flaws have to do with the location of the false positives and false negatives respectively. The metric redefines TP', FP', TN' and FN' in terms of the absolute error of detection $E = |G - D|$:

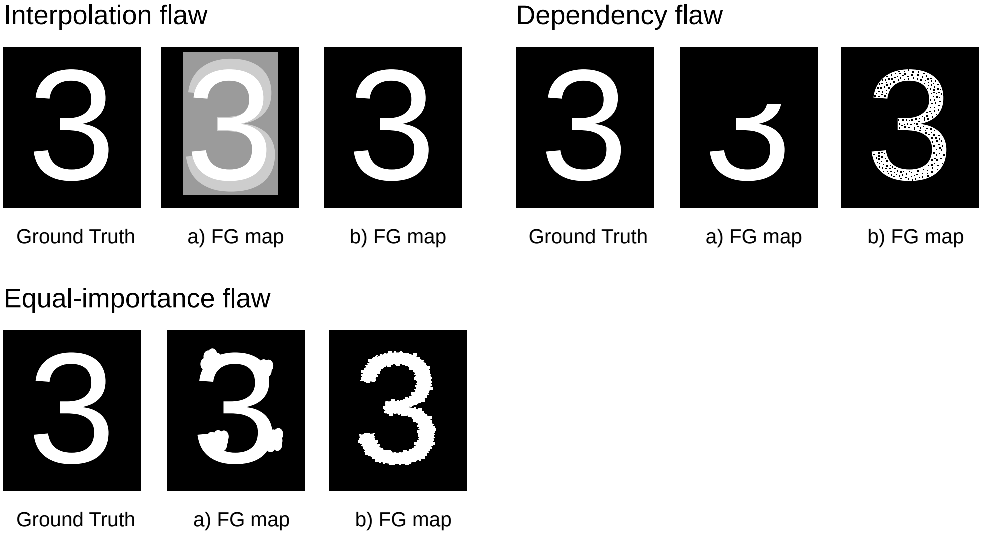


Figure 6.5.: Illustration of the three flaws of the Achanta method [Achanta et al., 2009] for evaluating foreground maps claimed by [Margolin et al., 2014]. In all three cases the b) maps look better than the a) maps, however they both obtain the same precision-recall curves using the Achanta method. Figure inspired from [Margolin et al., 2014].

$$TP' = (1 - E) \cdot G, \tag{6.10}$$

$$TN' = (1 - E) \cdot (1 - G), \tag{6.11}$$

$$FP' = E \cdot (1 - G), \tag{6.12}$$

$$FN' = E \cdot G. \tag{6.13}$$

Instead of this, we can compute a weighted error map E^w to take into account the pixel dependencies and the distances to the foreground, thus solving the dependency and equal-importance flaws. The weighted measures now become:

$$TP^w = (1 - E^w) \cdot G, \tag{6.14}$$

$$TN^w = (1 - E^w) \cdot (1 - G), \tag{6.15}$$

$$FP^w = E^w \cdot (1 - G), \tag{6.16}$$

$$FN^w = E^w \cdot G, \tag{6.17}$$

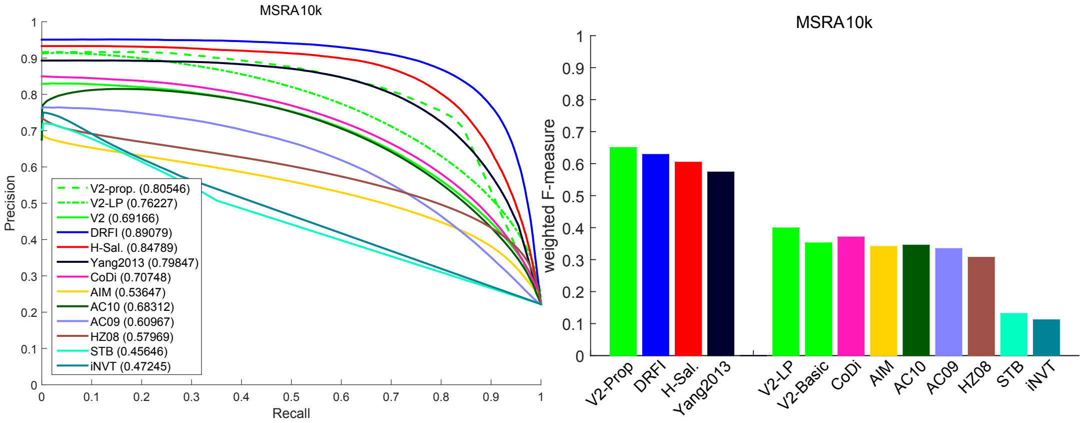


Figure 6.6.: Results for the MSRA-10k dataset. Left: Precision-recall curve. Right: average weighted F-measure. Our approach is denoted as V2-Prop.

which in turn can be used to compute the weighted precision and recall values as usual. Finally, a weighted F-measure can be obtained as

$$F_\beta^w = (1+\beta^2)\frac{Precision^w \cdot Recall^w}{\beta^2 + Precision^w + Recall^w} \tag{6.18}$$

In the following evaluations we will display the F_1^w values obtained in this way. For more details about this metric, we refer the readers to [Margolin et al., 2014].

6.2.2. Results on MSRA 10k

The MSRA dataset [Liu et al., 2007] is one of the most popular datasets for salient object detection. Several subsets exist for which pixel precise ground truth is available, such as the recent MSRA 10k [Cheng et al., 2015]. It consists of 10.000 images where subjects had to label the most salient object and is available online.[1]

In Fig. 6.6 we show the results obtained by our method (V2-prop) compared to the others. On the left we display the precision-recall curves according to the Achanta metric: our method is the third in terms of AUC (area under curve), behind H-Saliency (second) and DRFI (first).

On the right of Fig. 6.6, we show the average weighted F-measure for each of the methods. There, our method outperforms all the others. We show some examples of the images and the corresponding proposal and saliency maps in Fig. 6.10.

[1] http://mmcheng.net/msra10k/

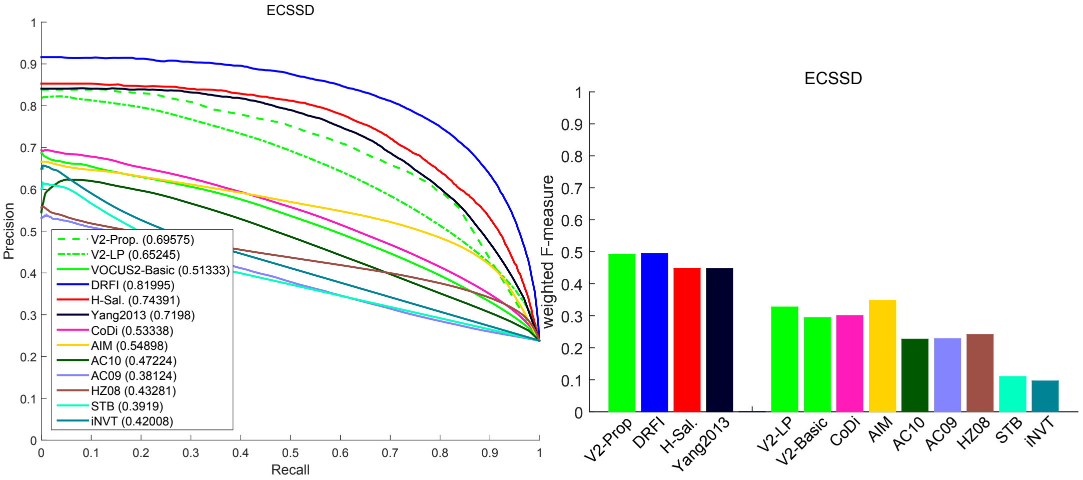


Figure 6.7.: Results for the ECSSD dataset. Left: Precision-recall curve. Right: average weighted F-measure. Our approach is denoted as V2-Prop.

6.2.3. Results on ECSSD

The Extended Complex Scene Saliency Dataset (ECSSD) [Yan et al., 2013] contains 1000 images where users labelled the most salient objects. The authors argue that existing datasets in salient object segmentation usually contain images where the background is homogeneous and not textured. Instead, in their set of images one can find textured patterns in both foreground and background. The dataset is available online.[2]

In Fig. 6.7 we show the results obtained by our method compared to the others. On the left we display the precision-recall curves according to the Achanta metric: we perform slightly worse than in the previous dataset, but our method is still among the top four. In terms of weighted F-measure, right of Fig. 6.7, we are still the second best, being only DRFI slightly better than us. We show some examples of this benchmark in Fig. 6.11.

6.2.4. Results on PASCAL-S

The PASCAL-S dataset [Li et al., 2014] contains 850 images labelled by 12 subjects. In contrast to the other datasets, each image was shown to several subjects and there was no limit in the number of objects to be labelled. This resulted in different ground truth labels for each of the objects that were labelled. The dataset is available online.[3]

In Fig. 6.8 we show the results obtained by our method compared to the others. On the left we display the precision-recall curves according to the Achanta metric: our

[2] http://www.cse.cuhk.edu.hk/leojia/projects/hsaliency/dataset.html

[3] `http://cbi.gatech.edu/salobj/`

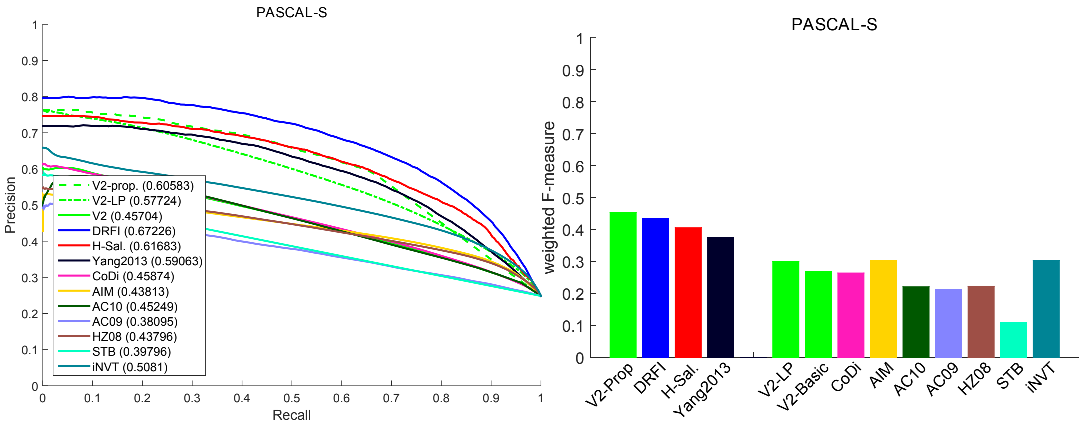


Figure 6.8.: Results for the PASCAL-S dataset. Left: Precision-recall curve. Right: average weighted F-measure. Our approach is denoted as V2-Prop.

method is third, and again DRFI is first and H-Saliency is second.

On the right of Fig. 6.8, the average F-weighted measure for each of the methods is shown: our method has the highest score, followed by DRFI and H-Saliency. In Fig. 6.12 we show some examples of the results obtained on images of the PASCAL-S dataset.

6.2.5. Results on SED1 and SED2

The Segmentation Evaluation Database (SED) [Alpert et al., 2007] consists of two datasets where either the most salient object is labelled (SED1), or the two most salient objects are labelled (SED2). The datasets can be found online.[4]

In Fig. 6.9 we show the results obtained by our method compared to the others. On the left we display the precision-recall curves according to the Achanta metric: our method is among the top four in SED1 and the third in SED2.

On the right of Fig. 6.9, we show the average F-weighted measure for each of the methods: our method is third in SED1 and first in SED2. Some examples of this dataset are displayed in Fig. 6.13.

6.2.6. Summary of the Evaluations

Overall, in terms of weighted F-measure, our method is the best on the MSRA 10k, Pascal-S and SED2 datasets; second best on ECSSD; and the 3rd on SED1. In terms of the Achanta metric (area under curve on the precision recall plots) our method is the 2nd on SED2; 3rd on MSRA 10k, ECSSD and Pascal-S. The results are consistently

[4] `http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/dl.html`

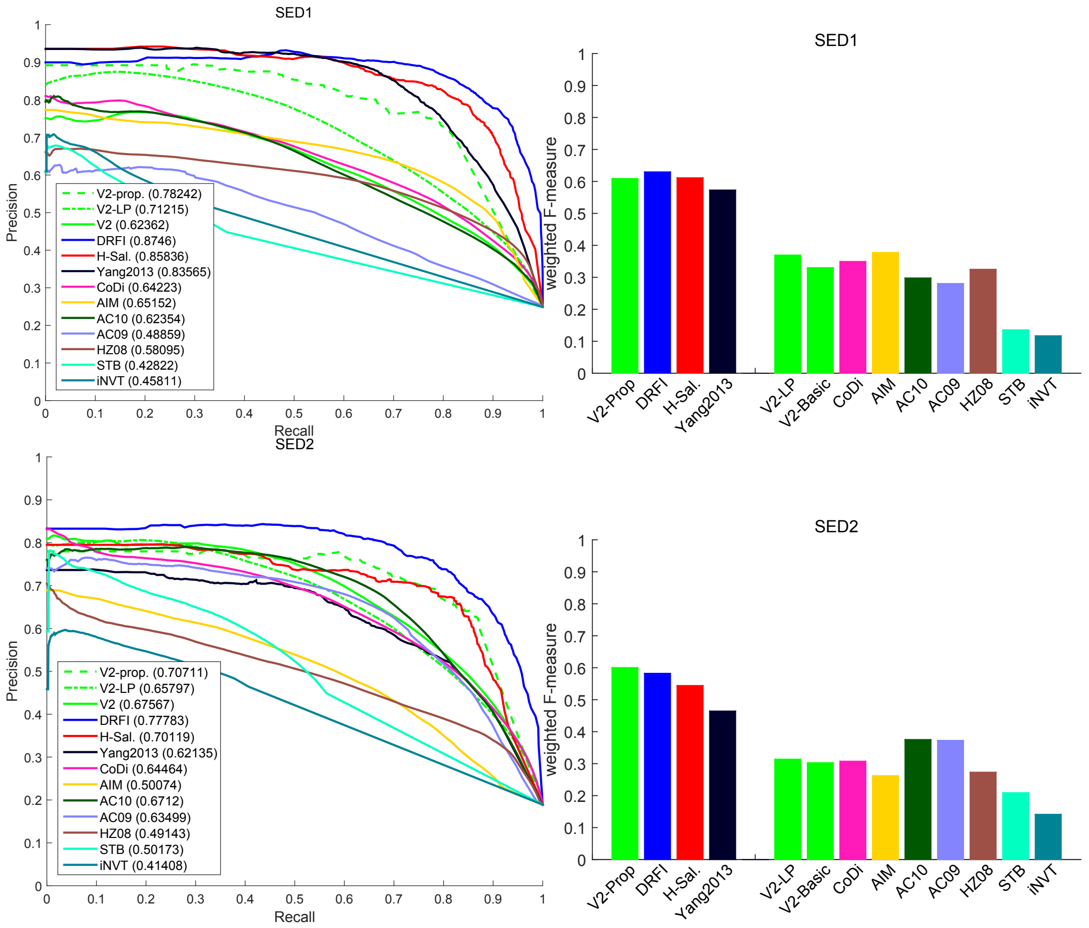


Figure 6.9.: Top row: results for the SED1 dataset. Bottom row: results for the SED2 dataset. Left column: Precision-recall curves. Right: average weighted F-measure. Our approach is denoted as V2-Prop.

better for the F-weighted measure, which as we saw in Section 6.2.1 tries to evaluate fairly those foreground maps that are in fact better. In summary, the results obtained show that our method is state of the art in salient object segmentation.

List of Own Publications for this Chapter

[a] Simone Frintrop, Thomas Werner and Germán Martín García. *Traditional Saliency Reloaded: A Good Old Model in New Shape. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA, 2015.*

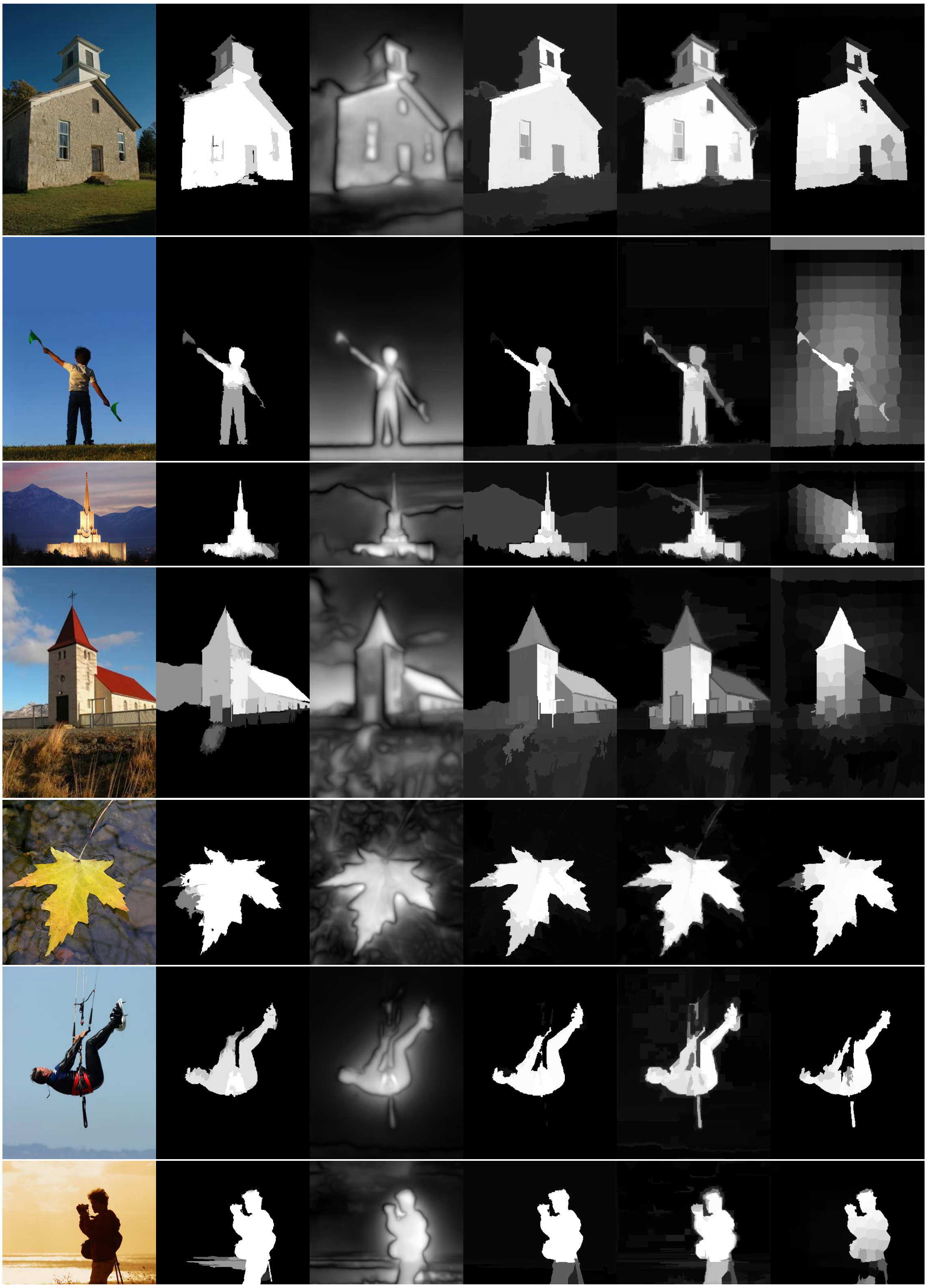


Figure 6.10.: Example results from the MSRA10k dataset. From left to right: original images, Proposal Maps (ours), VOCUS2, HSaliency, DRFI, Yang13.

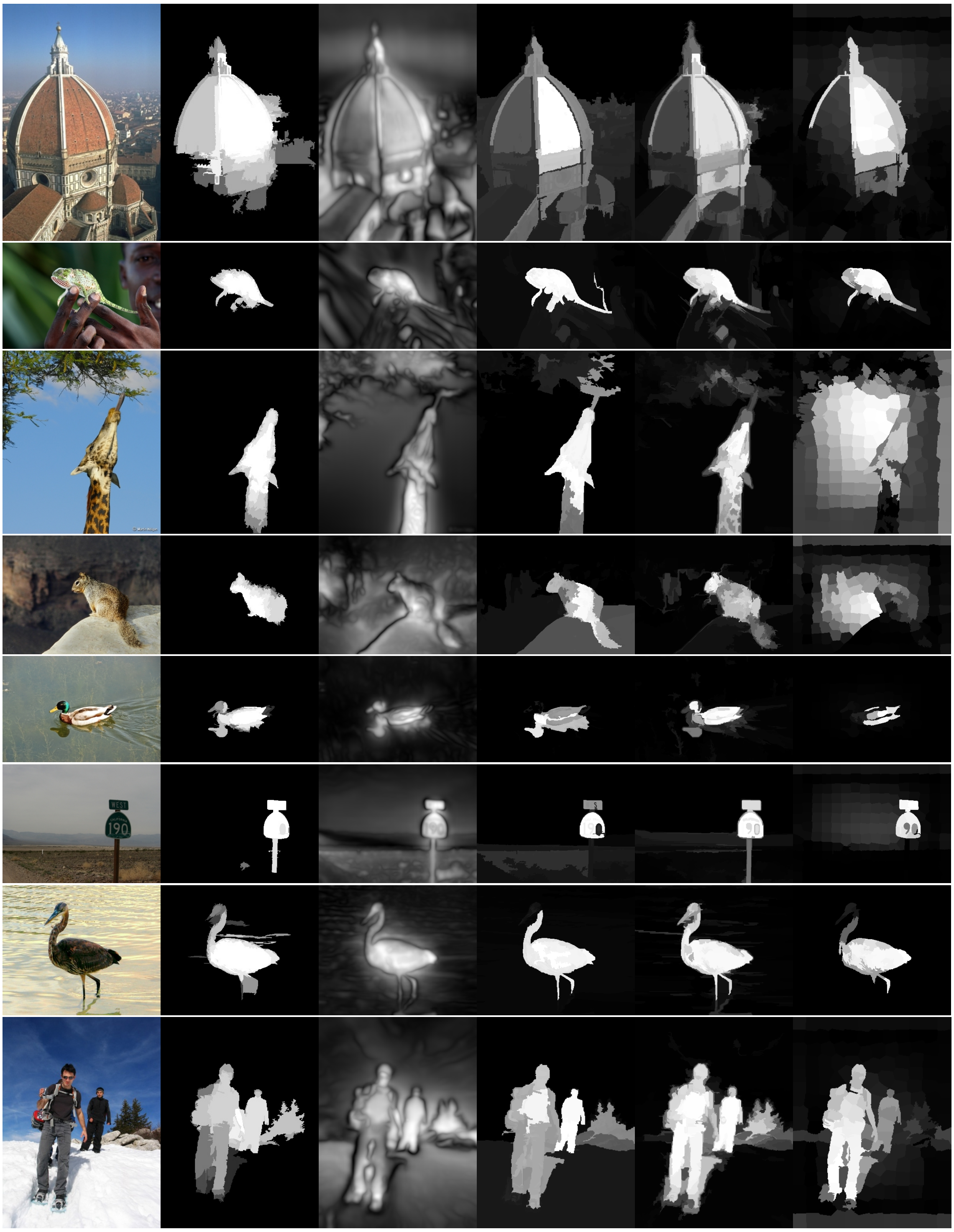


Figure 6.11.: Example results from the ECSSD dataset. From left to right: original images, Proposal Maps (ours), VOCUS2, HSaliency, DRFI, Yang13.

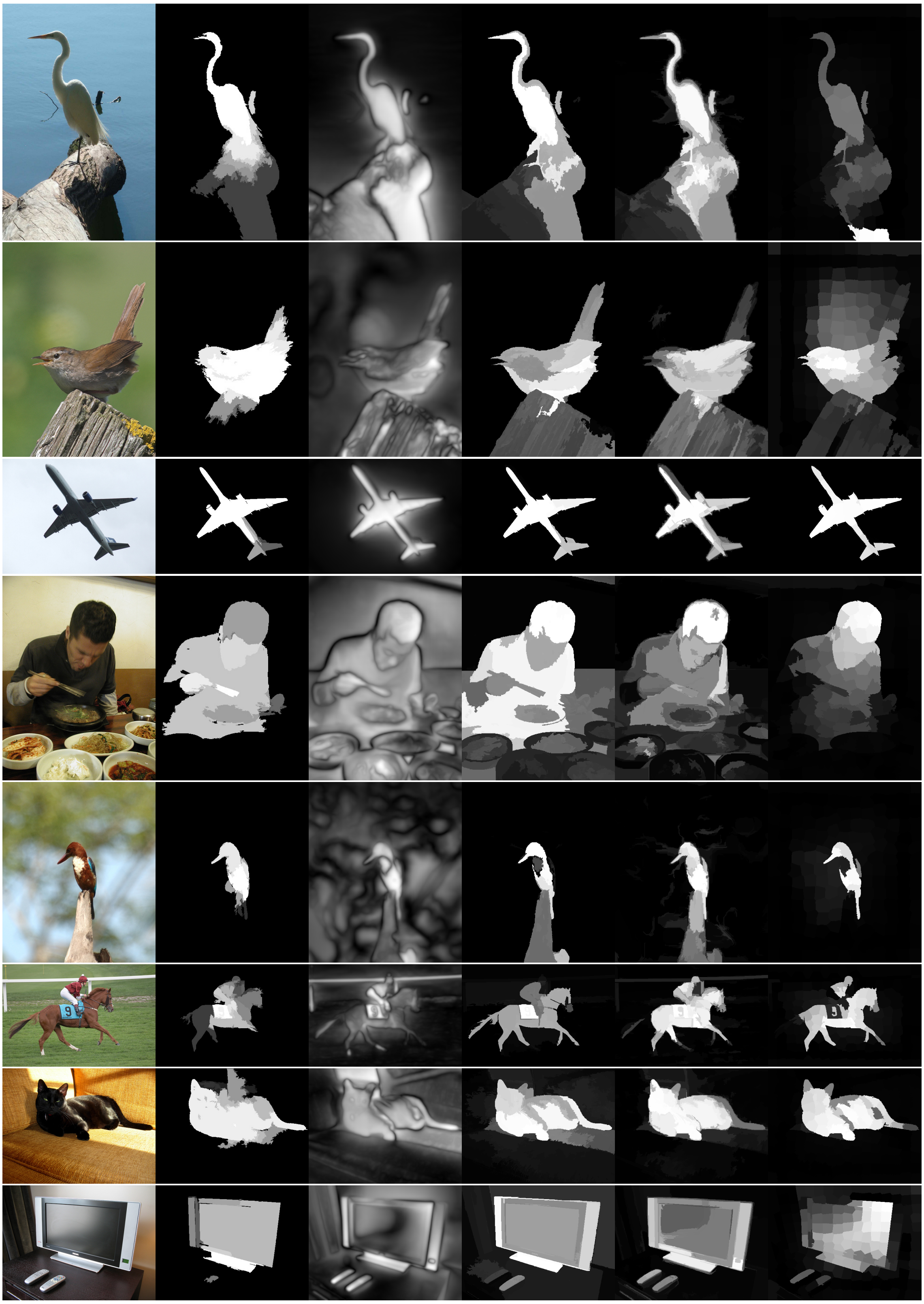


Figure 6.12.: Example results from the PASCAL-S dataset. From left to right: original images, Proposal Maps (ours), VOCUS2, HSaliency, DRFI, Yang13.

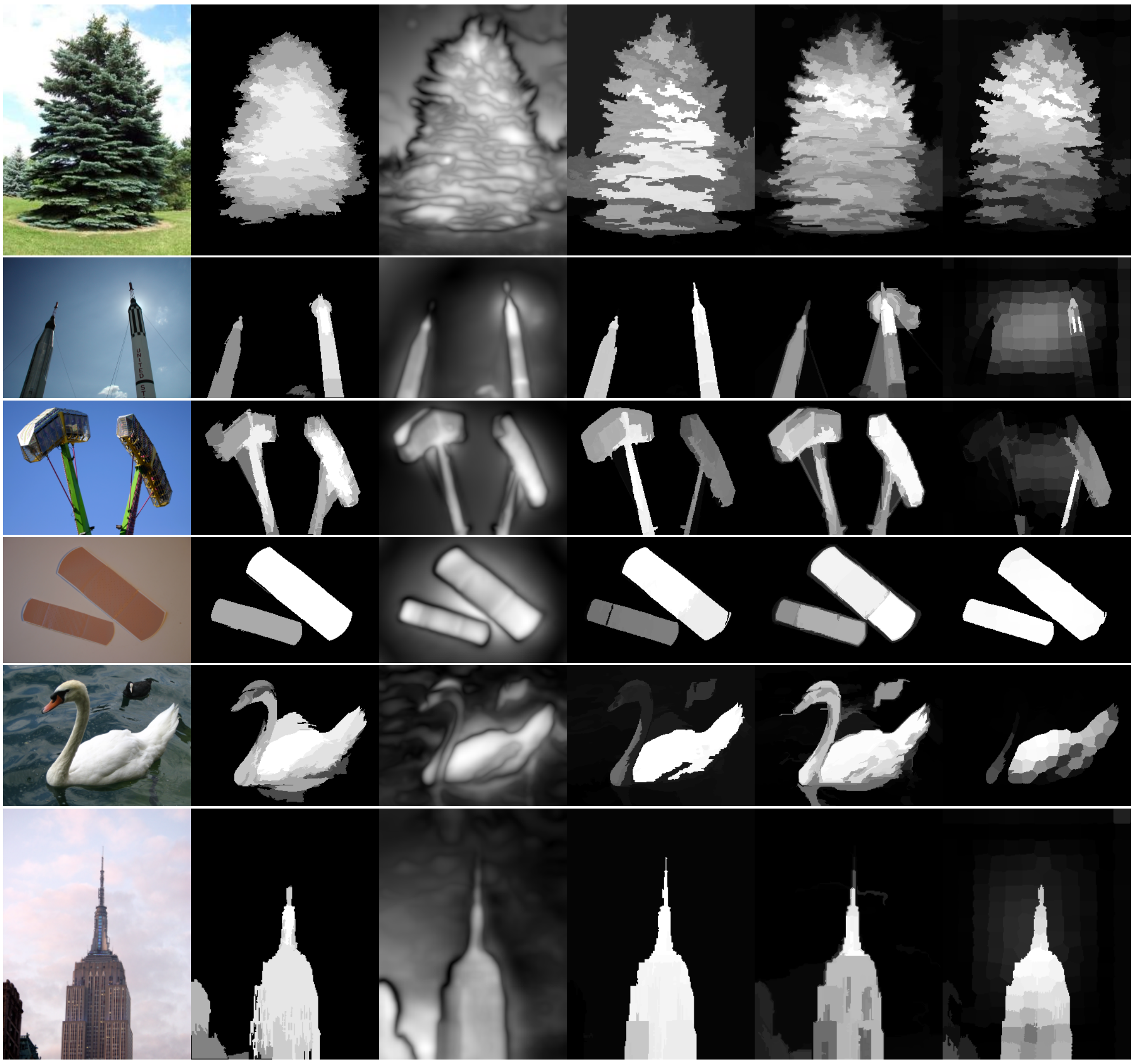


Figure 6.13.: Example results from the SED dataset. From left to right: original images, Proposal Maps (ours), VOCUS2, HSaliency, DRFI, Yang13.

7. Conclusion and Future Work

In the first part of this thesis, we have proposed a novel method for object discovery. The task in object discovery is to find the objects that are present in a scene without having prior knowledge about their appearance or categories. Our goal was to develop a method that can achieve higher recall values than current state-of-the-art methods with fewer generated object candidates, and thus, that is suitable for robotic applications.

Our approach has two main stages: 1) To generate a set of object candidates based on saliency and segmentation; here, we proposed two strategies for extracting salient regions (S1 and S2) based on a single saliency map and a multi-scale saliency approach; we also evaluated four different segmentation strategies that use colour (M1), depth (M2), an early fusion of colour and depth (M3) and a late fusion of the colour and depth candidates (M4). 2) The second stage consists of ranking the generated object candidates according to their objectness. We proposed and evaluated three different ranking strategies based on the saliency and area of the candidate (R1), the 3D convexity of the candidate (R2) and an SVM trained on a set of features (R3).

We evaluated the variants of our method and compared them internally as well as with respect to four other methods from the computer and robotic vision literature. We used three publicly available benchmarks to perform the evaluations. The internal evaluation showed that the R3 ranking strategy was superior to R1 and R2 and proved to generalise well across different datasets. Overall, R3 was successful in sorting the good candidates first. The segmentation and salient region extraction variants were jointly evaluated. The results consistently showed that the late fusion of colour and depth (M4) was superior to the early fusion (M3) in terms of the quality of the proposals, as well as the recall achieved. Furthermore, colour and depth were complementary modalities: some objects were most easily found by relying on depth (M2) and some by relying on colour (M1). The multi-scale saliency approach (S2) was able to recall more objects than the single saliency alternative (S1). Regarding the quality of the object proposals (recall vs IoU plots), having fixed the segmentation method, the S2 candidates were in general more precise than the S1 ones. Our approach was superior to other state-of-the-art methods in object discovery in terms of the quality the proposals and the recall of the objects, particularly in cluttered scenes.

In the second part of this thesis we turned our focus to RGB-D sequences rather than single images. Our goal was to sequentially explore the scene, so that with very few candidates per frame (20 instead of the about 200 that are produced per frame) we can still (globally) recall most of the objects by the end of the sequence. Therefore, we proposed a spatial inhibition of return mechanism (IOR) that prevents the system from producing object candidates that were already generated in the previous frames. The IOR mechanism relies on a 3D map to store the inhibition values in 3D coordinates to naturally cope with camera motions. The experiments showed that with the IOR mechanism we can recall a similar number of objects throughout the sequence with significantly fewer candidates per frame (20 instead of 255).

In the last part, we showed an application of our S1-M1 object discovery algorithm to the task of salient object segmentation. We compared our approach to several other state-of-the-art methods on five popular benchmarks on salient object segmentation. The results showed that our method is state-of-the-art in salient object segmentation.

7.1. Future Work

We propose two lines of research that would continue the work carried out in this thesis. The first one has to do with improving the object discovery method on single images, and the second one with the way temporal information is used. We sketch the two in the following sections.

The role of top-down information

The most popular methods for object discovery rely either on the physical properties of the image alone to produce object candidates [Uijlings et al., 2013], on machine learning methods that are trained on general aspects of objects to rank randomly sampled object candidate windows [Alexe et al., 2012], or on combinations of both [Manén et al., 2013].

An interesting line of research would be to integrate top-down information in the bottom-up process of generating object candidates: several evidences show that top-down information plays a crucial role in visual organisation [Behrmann and Kimchi, 2003], and past experience is also one of the well known Gestalt principles [Wagemans et al., 2012].

Starting with a segmentation of the image, and driven by cues such as saliency, a possible algorithm would be to start with a segment, and start grouping it to neighbouring segments until an object is recognised; then proceed to another segment that has not been categorised and repeat the process until the whole image is understood.

Image and Video Segmentation

In the work we presented in [Horbert et al., 2015] we introduce the idea of sequence-level proposals by tracking the generated object candidates over time. The tracking method, however, is a separate module using the object candidates as starting points for generating tracks.

Since the object candidates consist of segments, we would like to develop a method for video segmentation that reliably finds the perceptually coherent spatio-temporal segments that compose a video sequence. Video segmentation can be seen as an equivalent method of image segmentation for video sequences. Here, we define video segmentation as the task of identifying perceptually coherent visual regions as they appear throughout a sequence, in contrast to what is known as video object segmentation, where the task is to directly extract the objects.

Top-down information would be used to resolve the identities or categories of the object hypotheses being generated. Once an object candidate is classified in one frame, its corresponding segments in the following and preceding frames are automatically classified.

A. Appendix: Object Proposal Evaluation

We include here the complete set of plots obtained in the evaluations of the object proposal generation methods of Chapter 4. In Section A.1, we show the results obtained in the Washington Dataset for Object Recognition. We include the precision and recall plots over proposals: Figs. A.1 and A.2 for the S1 (single saliency) salient extraction mode, and Figs. A.3 and A.4 for the S2 (multi-scale saliency) salient extraction mode We also show the global recall over time and over proposals plots: Figs. A.5 and A.6 for the S1 mode, and Figs. A.7 and A.8 for the S2 mode. We show the results obtained in all the individual sequences of the dataset: Desk 1, Desk 2, Desk 3, Kitchen Small, Meeting Small, Table, Table Small 1 and Table Small 2.

We show in Section A.2 the results obtained in the Kitchen Dataset. We include the precision and recall plots over proposals: Fig. A.9 for the S1 (single saliency) salient extraction mode, and Fig. A.10 for the S2 (multi-scale saliency) salient extraction mode. We also show the global recall over time and over proposals plots: Fig. A.11 for the S1 mode, and Fig. A.12 for the S2 mode. We show the results obtained in all the individual sequences of the dataset: Kitchen A, B, C and D.

A.1. Washington Dataset

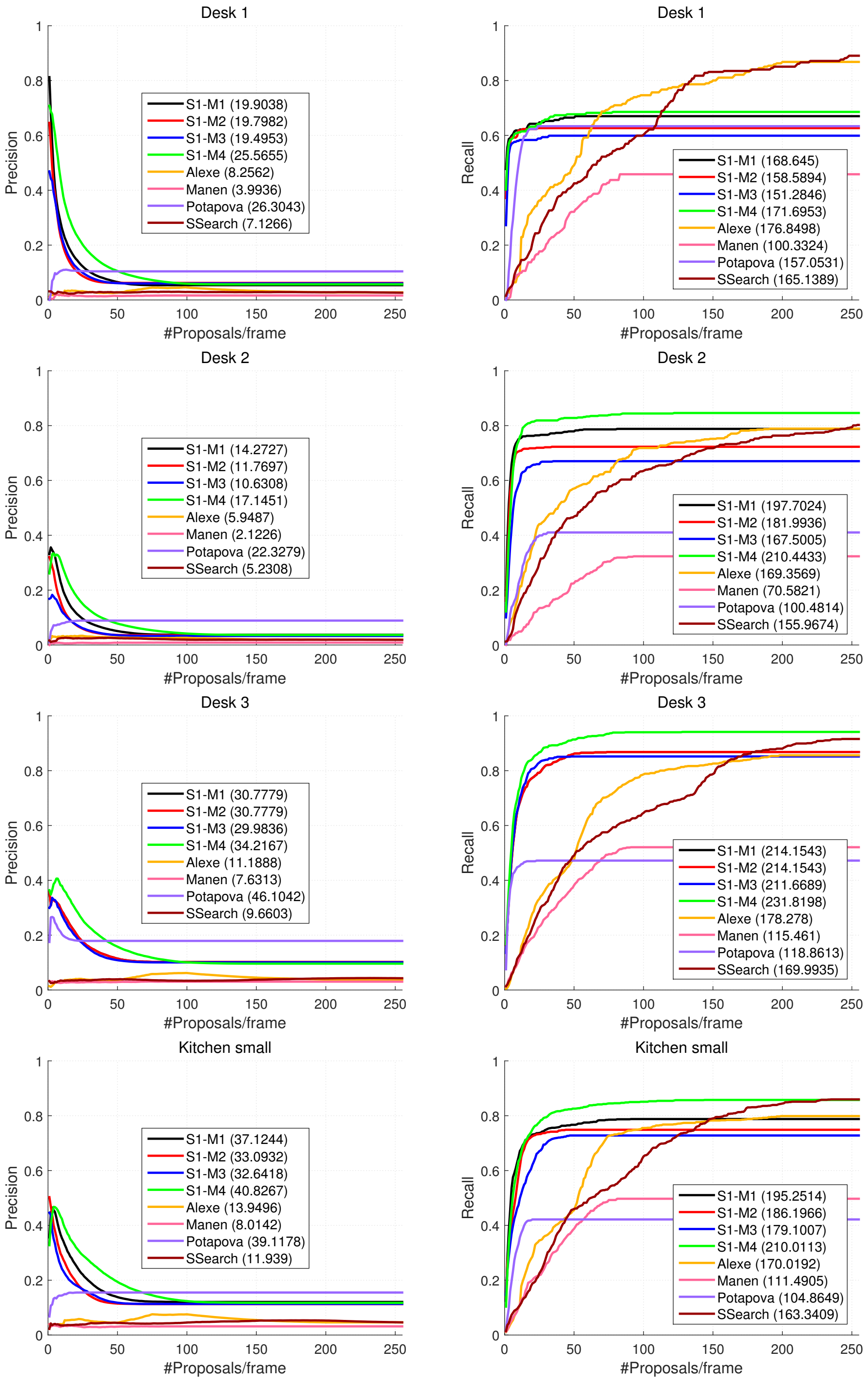


Figure A.1.: Precision and recall plots for the Washington Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

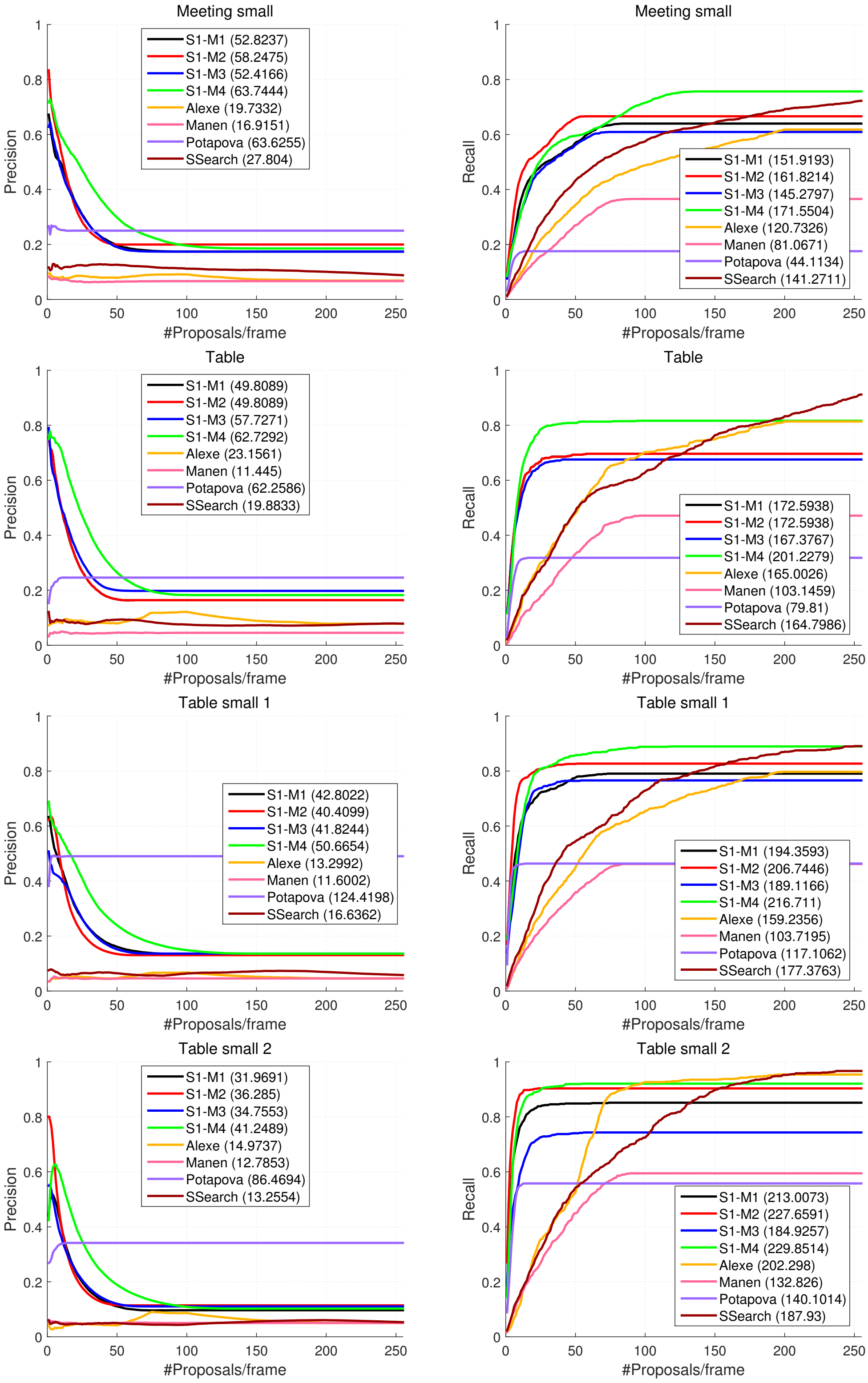


Figure A.2.: Precision and recall plots for the Washington Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

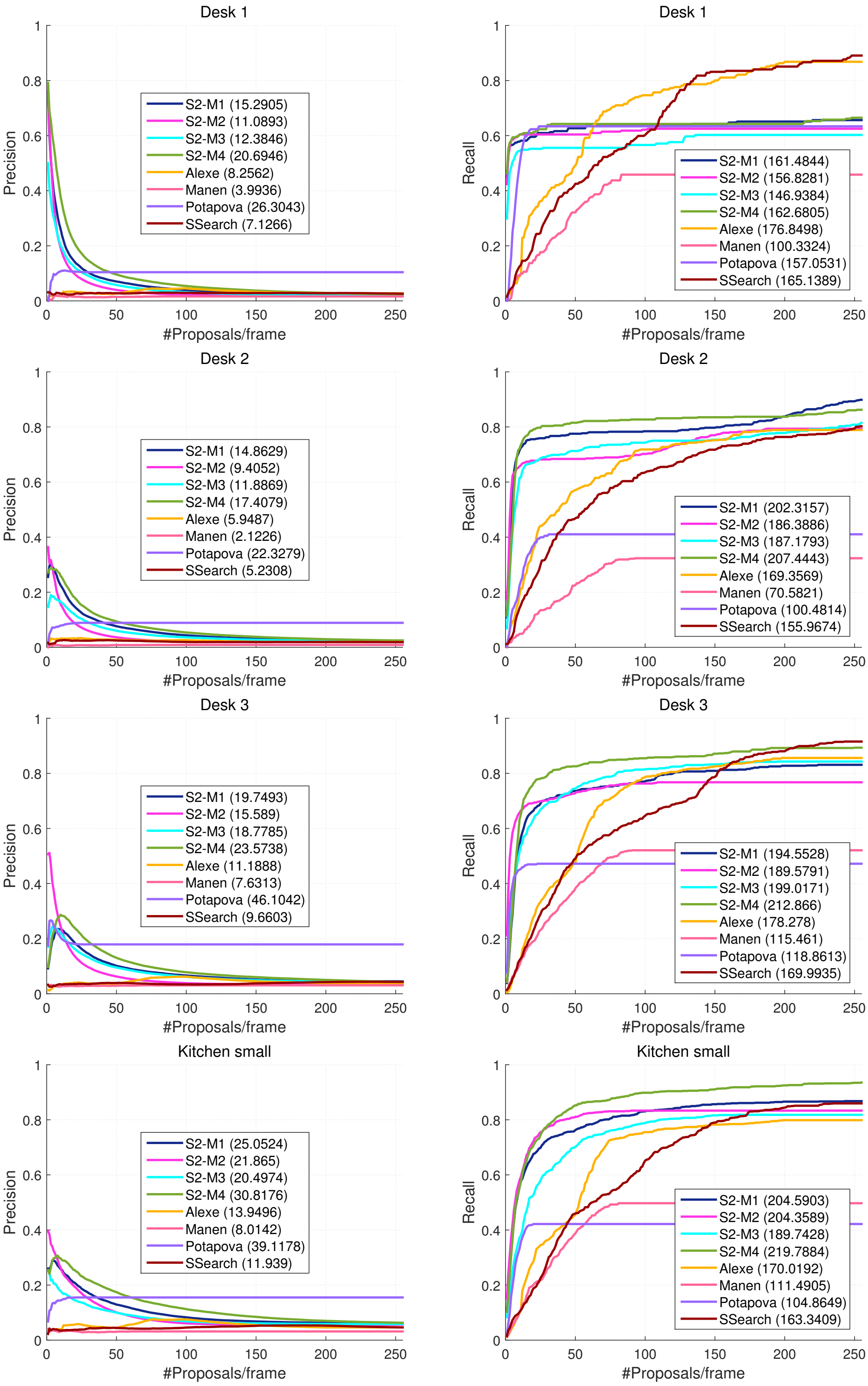


Figure A.3.: Precision and recall plots for the Washington Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

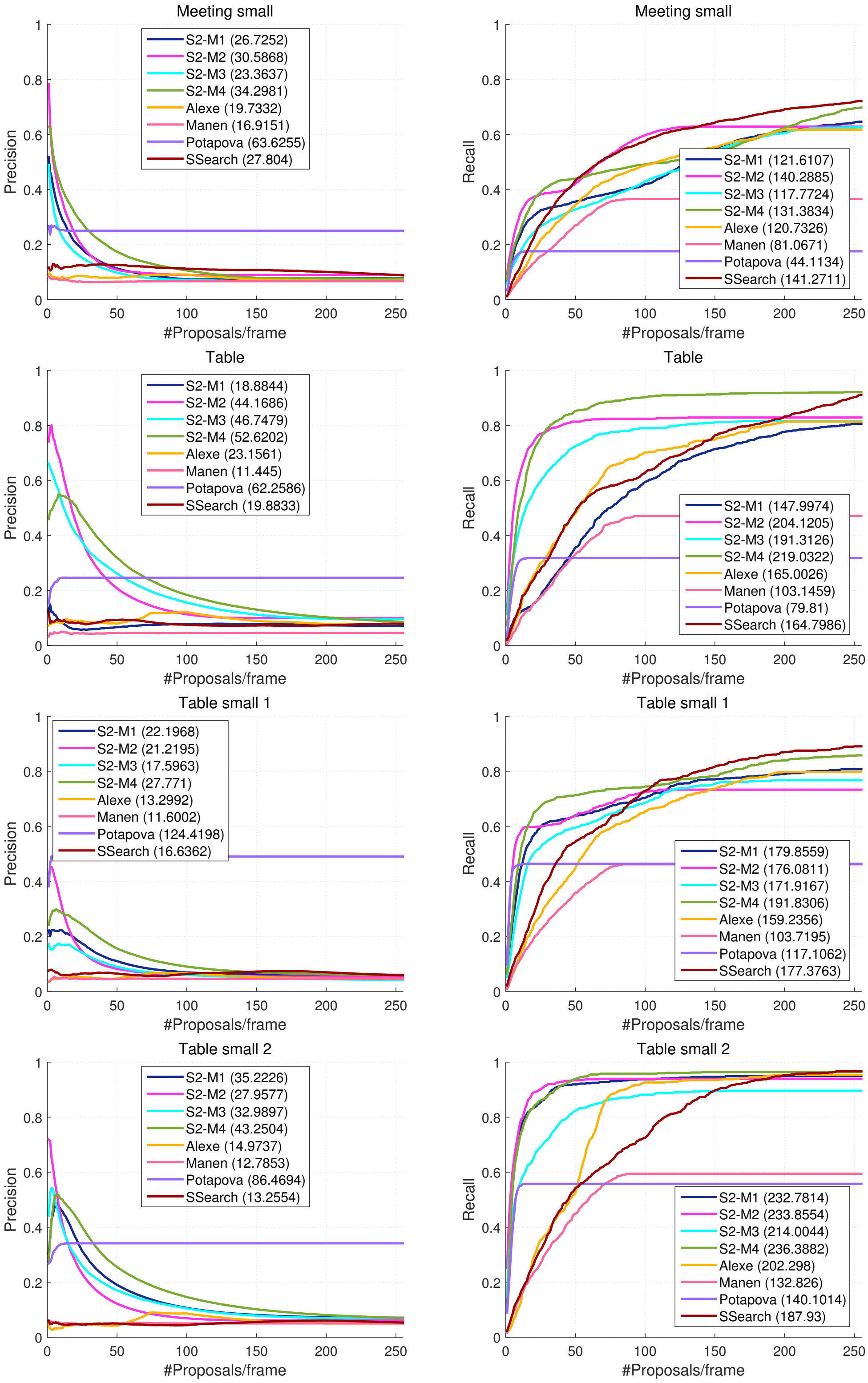


Figure A.4.: Precision and recall plots for the Washington Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

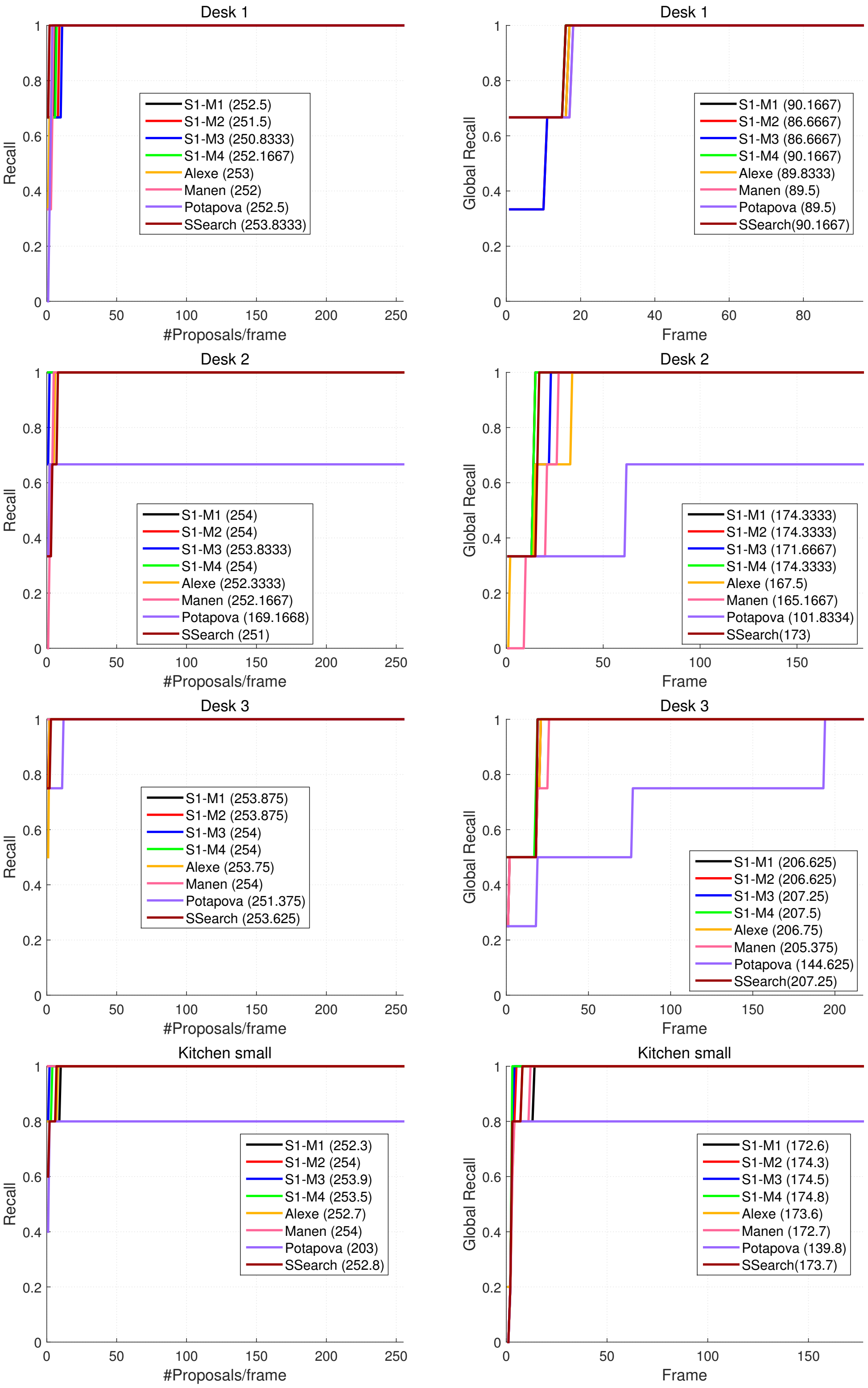


Figure A.5.: Global recall over proposals and over time for the Washington Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

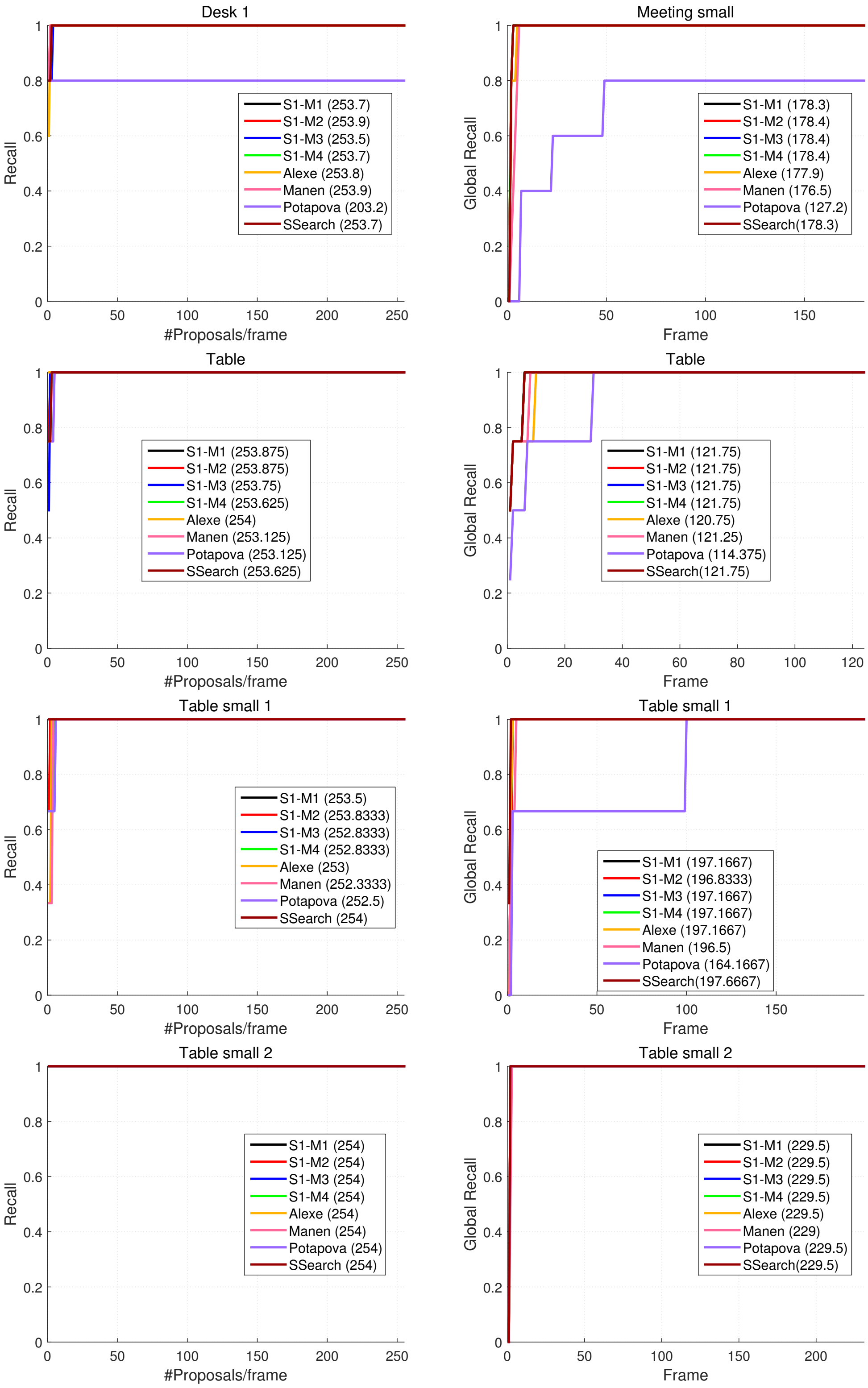


Figure A.6.: Global recall over proposals and global recall over time for the Washington Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

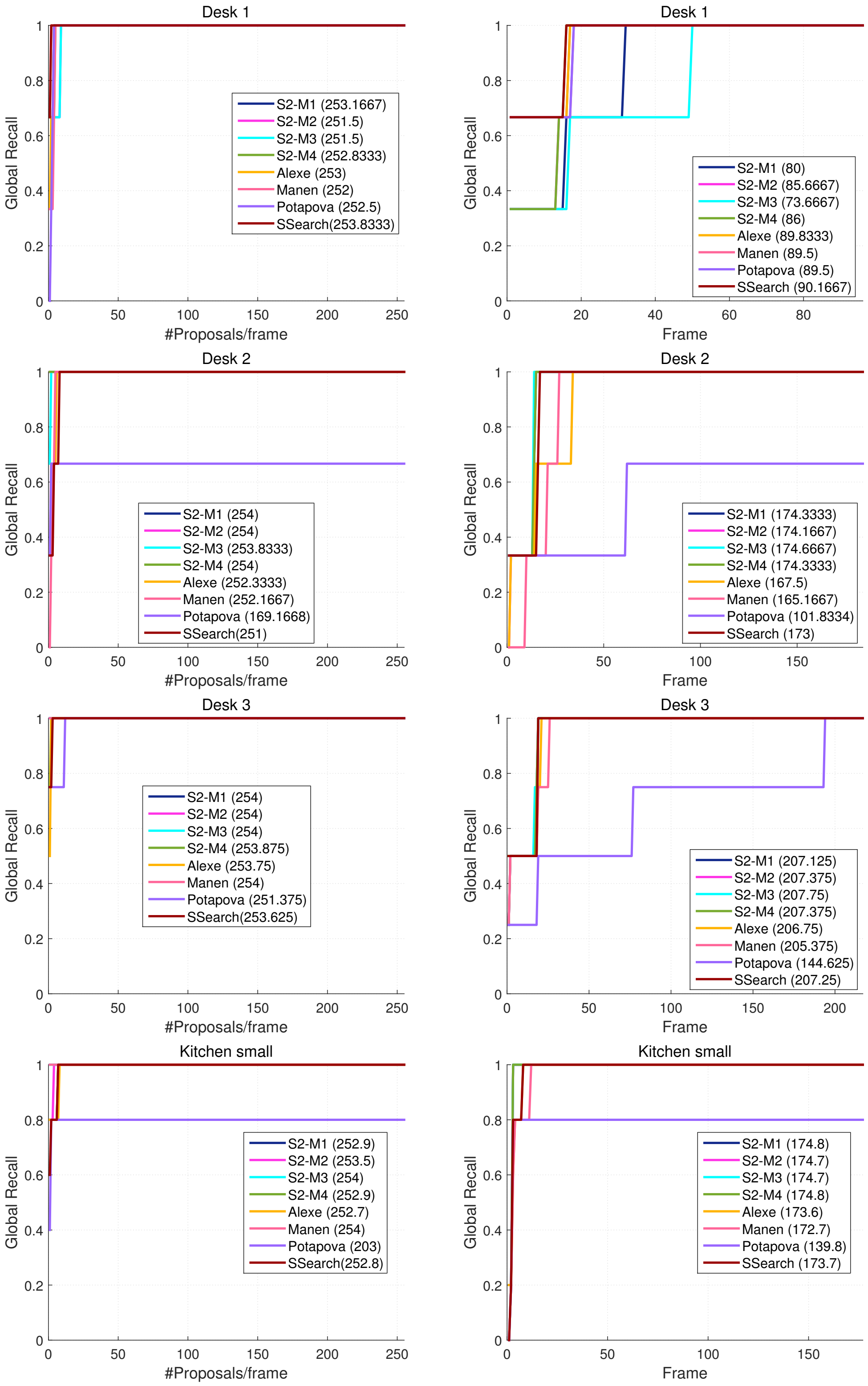


Figure A.7.: Global recall over proposals and over time for the Washington Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

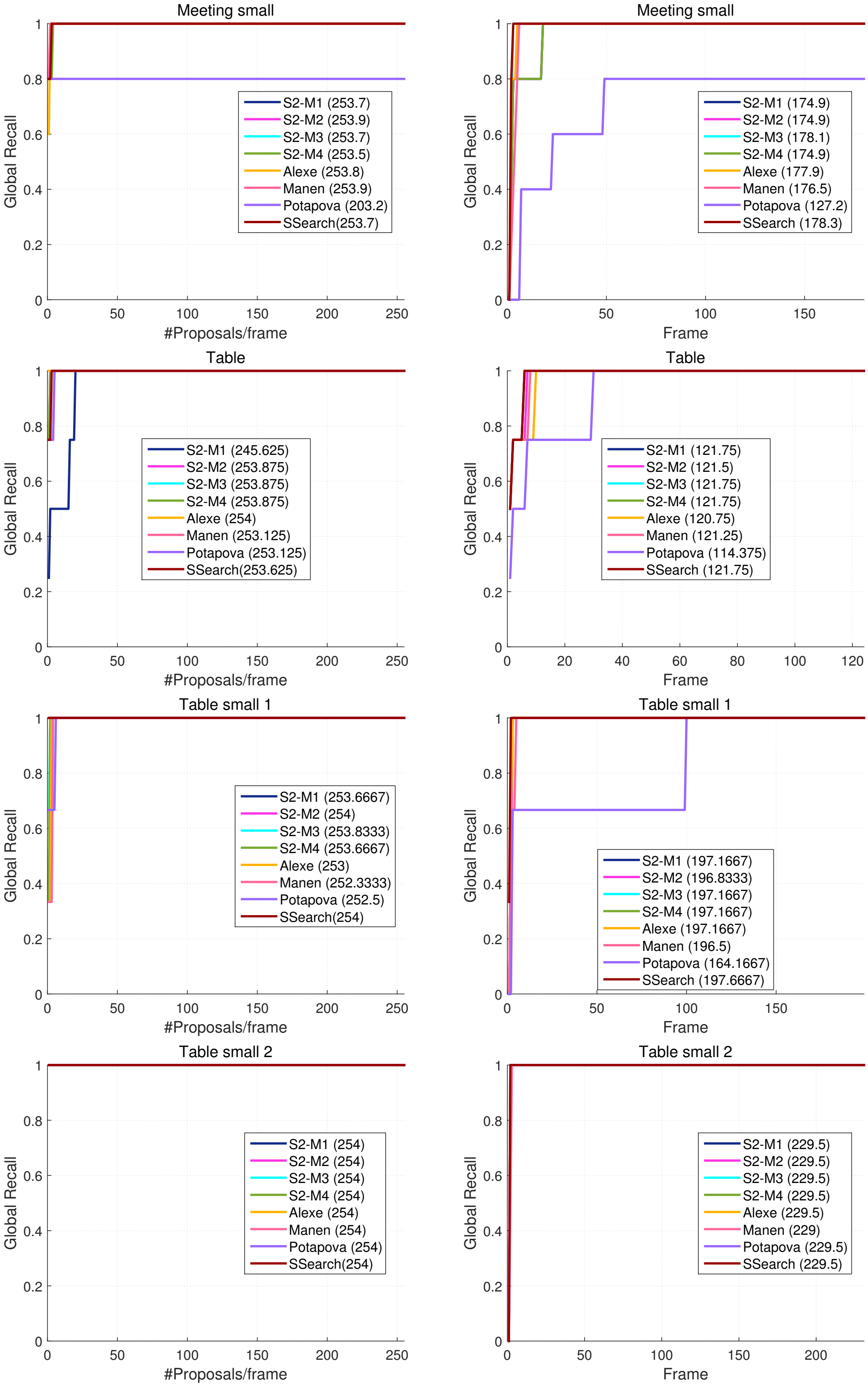


Figure A.8.: Global recall over proposals and over time for the Washington Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén, Potapova and Selective Search.

A.2. Kitchen Dataset

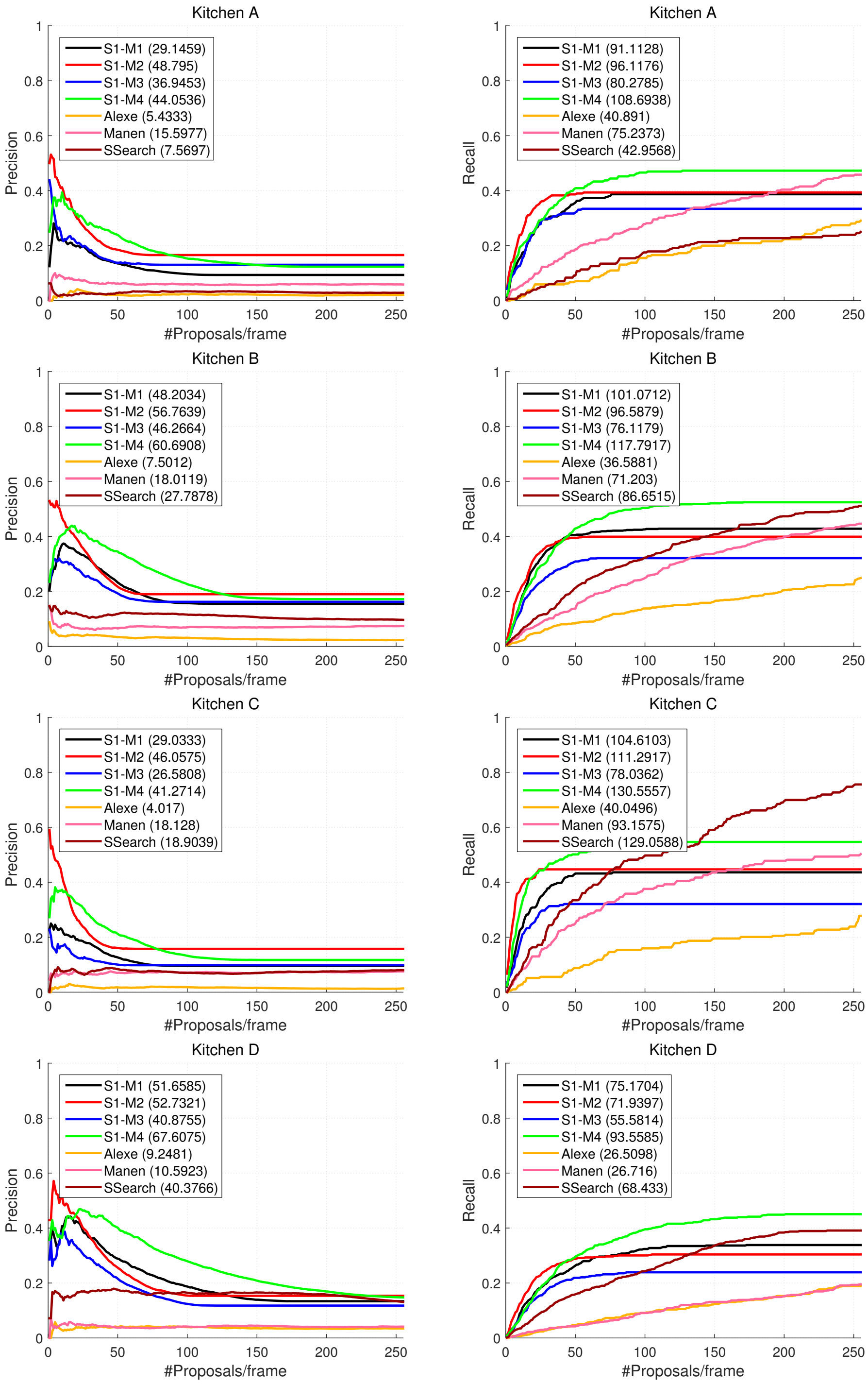


Figure A.9.: Precision and recall plots for the Kitchen Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén and Selective Search.

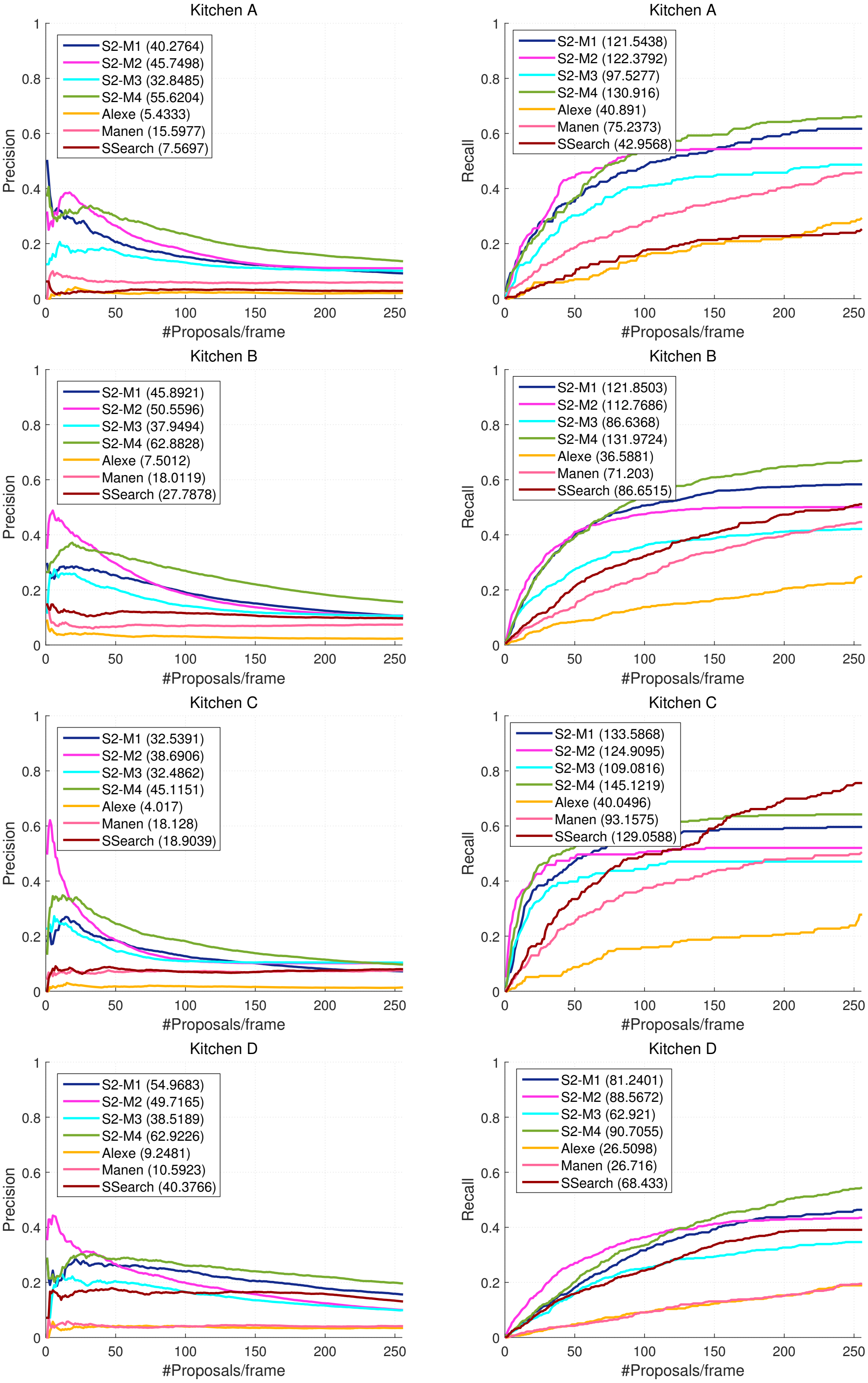


Figure A.10.: Precision and recall plots for the Kitchen Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén and Selective Search.

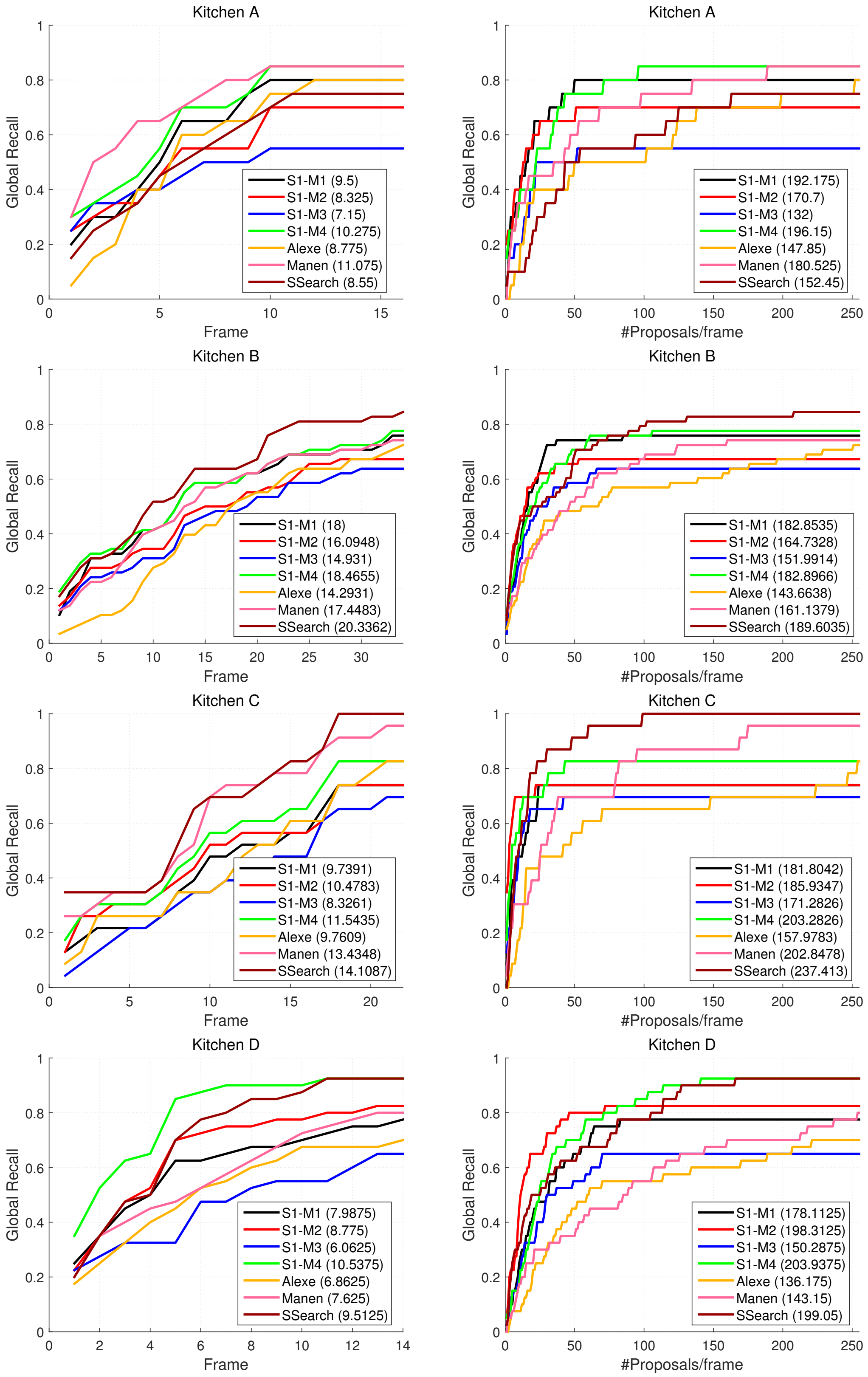


Figure A.11.: Global recall over time and over proposals for the Kitchen Dataset. Comparison of our S1 method (in its four segmentation variants) with the methods of Alexe, Manén and Selective Search.

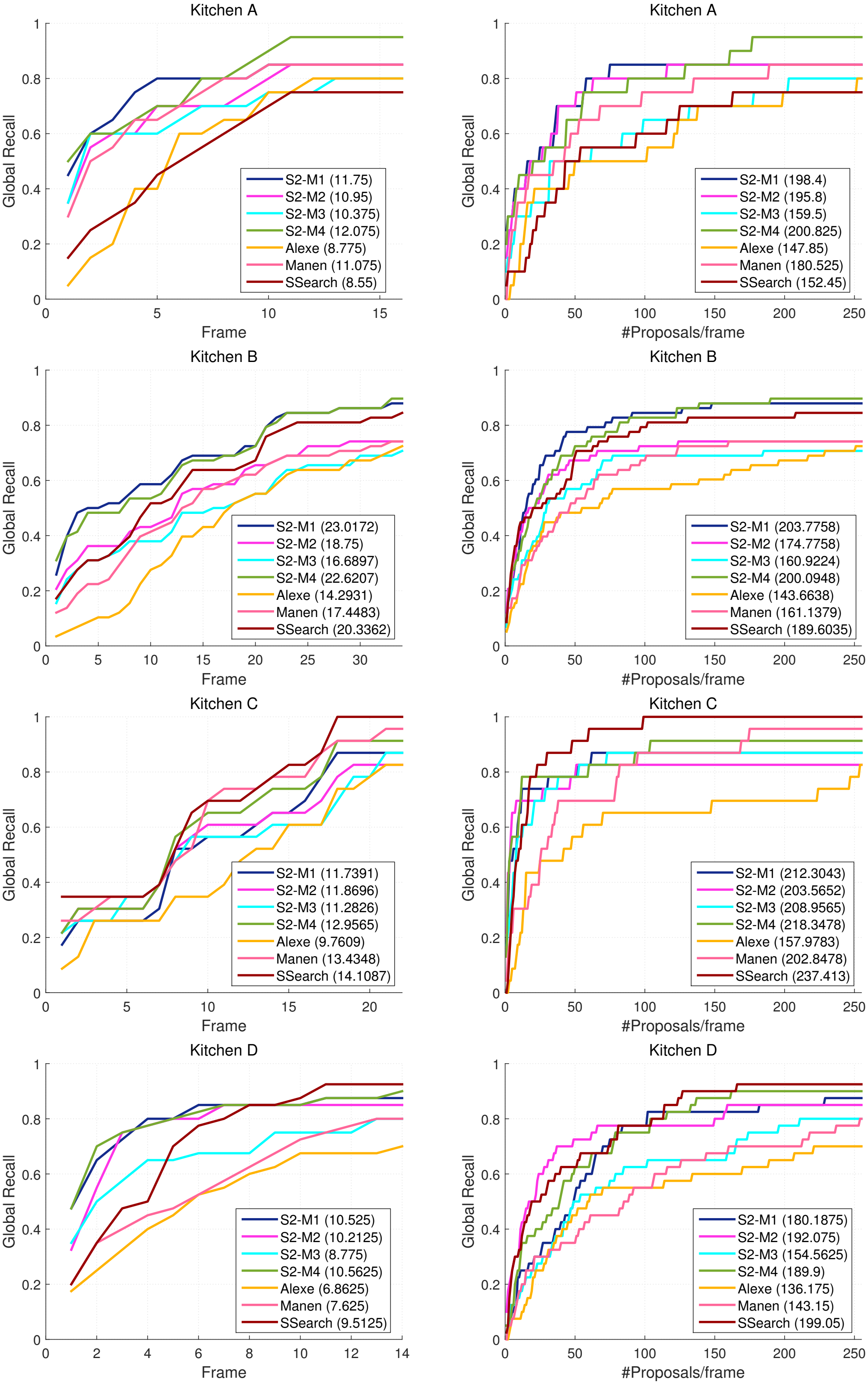


Figure A.12.: Global recall over time and over proposals for the Kitchen Dataset. Comparison of our S2 method (in its four segmentation variants) with the methods of Alexe, Manén and Selective Search.

List of Figures

List of Tables

Bibliography

[Achanta et al., 2009] Achanta, R., Hemami, S., Estrada, F., and Süsstrunk, S. (2009). Frequency-tuned salient region detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Achanta and Süsstrunk, 2010] Achanta, R. and Süsstrunk, S. (2010). Saliency Detection using Maximum Symmetric Surround. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*.

[Adams and Bischof, 1994] Adams, R. and Bischof, L. (1994). Seeded Region Growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 16(6):641 – 647.

[Alexe et al., 2012] Alexe, B., Deselaers, T., and Ferrari, V. (2012). Measuring the Objectness of Image Windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2189–2202.

[Alpert et al., 2007] Alpert, S., Galun, M., Basri, R., and Brandt, A. (2007). Image segmentation by probabilistic bottom-up aggregation and cue integration. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Arandjelović and Zisserman, 2011] Arandjelović, R. and Zisserman, A. (2011). Smooth object retrieval using a bag of boundaries. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

[Arbeláez et al., 2012] Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., and Malik, J. (2012). Semantic segmentation using regions and parts. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Backer et al., 2001] Backer, G., Mertsching, B., and Bollmann, M. (2001). Data- and model-driven gaze control for an active-vision system. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(12).

[Baker et al., 2011] Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., and Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31.

[Barla et al., 2003] Barla, A., Odone, R., and Verr, A. (2003). Histogram intersection kernel for image classification. In *Proc. of the IEEE International Conference on Image Processing (ICIP).*

[Behrmann and Kimchi, 2003] Behrmann, M. and Kimchi, R. (2003). What does visual agnosia tell us about perceptual organization and its relationship to object perception? *Journal of Experimental Psychology: Human Perception and Performance*, 29(1):19.

[Besl and McKay, 1992] Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):239–256.

[Borji and Itti, 2013] Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):185–207.

[Bruce and Tsotsos, 2009] Bruce, N. D. B. and Tsotsos, J. K. (2009). Saliency, Attention, and Visual Search: An Information Theoretic Approach. *Journal of Vision*, 9(3):1–24.

[Carreira and Sminchisescu, 2010] Carreira, J. and Sminchisescu, C. (2010). Constrained parametric min-cuts for automatic object segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27.

[Cheng et al., 2015] Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H. S., and Hu, S.-M. (2015). Global Contrast based Salient Region Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(3):569–582.

[Cinbis et al., 2013] Cinbis, R. G., Verbeek, J., and Schmid, C. (2013). Segmentation driven object detection with fisher vectors. In *Proc. of the IEEE International Conference on Computer Vision (ICCV).*

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

[Curless and Levoy, 1996] Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).*

[Endres and Hoiem, 2010] Endres, I. and Hoiem, D. (2010). Category independent object proposals. In *Proc. of the European Conference on Computer Vision (ECCV).*

[Everingham et al., 2007] Everingham, M., Van Gool, L., Williams, C., Winn, J., and Zisserman, A. (2007). The Pascal Visual Object Classes Challenge 2007 Results. `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/`. [Online; accessed 17-Feb-2015].

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

[Felzenszwalb et al., 2010] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(9):1627–1645.

[Felzenszwalb and Huttenlocher, 2004] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision (IJCV)*, 59(2).

[Forsyth et al., 1996] Forsyth, D. A., Malik, J., Fleck, M. M., Greenspan, H., Leung, T., Belongie, S., Carson, C., and Bregler, C. (1996). *Finding pictures of objects in large collections of images.* Springer.

[Forsyth and Ponce, 2003] Forsyth, D. A. and Ponce, J. (2003). *Computer Vision: A Modern Approach.* Prentice Hall.

[Frintrop, 2006] Frintrop, S. (2006). *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*, volume 3899 of *LNAI.* Springer.

[Frintrop, 2011] Frintrop, S. (2011). Computational visual attention. In *Computer Analysis of Human Behavior*, pages 69–101. Springer.

[Frintrop et al., 2014] Frintrop, S., Martín García, G., and Cremers, A. B. (2014). A cognitive approach for object discovery. In *Proc. of the International Conference in Pattern Recognition (ICPR).*

[Frintrop et al., 2010] Frintrop, S., Rome, E., and Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception*, 7(1).

[Frintrop et al., 2015] Frintrop, S., Werner, T., and Martín García, G. (2015). Traditional saliency reloaded: A good old model in new shape. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Gonzalez and Woods, 2006] Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[Guizzo, 2011] Guizzo, E. (2011). How google's self-driving car works. *IEEE Spectrum Online, October*, 18.

[Harzallah et al., 2009] Harzallah, H., Jurie, F., and Schmid, C. (2009). Combining efficient object localization and image classification. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

[He et al., 2014] He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. of the European Conference on Computer Vision (ECCV)*. Springer.

[Heinke and Humphreys, 2004] Heinke, D. and Humphreys, G. W. (2004). Computational models of visual selective attention. A review. In *Connectionist models in psychology*. Psychology Press.

[Herbst et al., 2011] Herbst, E., Henry, P., Ren, X., and Fox, D. (2011). Toward Object Discovery and Modeling via 3-D Scene Comparison. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Horbert et al., 2015] Horbert, E., Martín García, G., Frintrop, S., and Leibe, B. (2015). Sequence Level Object Candidates Based on Saliency for Generic Object Recognition on Mobile Systems. *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Hosang et al., 2015] Hosang, J., Benenson, R., Dollár, P., and Schiele, B. (2015). What makes for effective detection proposals? *arXiv:1502.05082*.

[Hosang et al., 2014] Hosang, J., Benenson, R., and Schiele, B. (2014). How good are detection proposals, really? In *Proc. of the British Machine Vision Conference (BMVC)*.

[Hou and Zhang, 2008] Hou, X. and Zhang, L. (2008). Dynamic visual attention: Searching for coding length increments. In *Advances in Neural Information Processing Systems*.

[Hu, 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187.

[Hunt, 1991] Hunt, R. W. G. (1991). *Measuring colour*. Ellis Horwood Limited, Chichester, West Sussex, England.

[Hurvich and Jameson, 1957] Hurvich, L. and Jameson, D. (1957). An opponent-process theory of color vision. *Psychological review*, 64(6).

[Itti, 2007] Itti, L. (2007). Visual salience. *Scholarpedia*, 2(9):3327. revision 72776.

[Itti et al., 1998] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

[Jiang et al., 2011] Jiang, H., Wang, J., Yuan, Z., Liu, T., Zheng, N., and Li, S. (2011). Automatic salient object segmentation based on context and shape prior. In *Proc. of the British Machine Vision Conference (BMVC)*, volume 6, page 9.

[Jiang et al., 2013] Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., and Li, S. (2013). Salient object detection: A discriminative regional feature integration approach. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Kanizsa and Gerbino, 1976] Kanizsa, W. and Gerbino, W. (1976). *Convexity and symmetry in figure-ground organization*. Springer.

[Karpathy et al., 2013] Karpathy, A., Miller, S., and Fei-Fei, L. (2013). Object Discovery in 3D Scenes via Shape Analysis. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Klein and Frintrop, 2011] Klein, D. A. and Frintrop, S. (2011). Center-surround divergence of feature statistics for salient object detection. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

[Klein and Frintrop, 2012] Klein, D. A. and Frintrop, S. (2012). Salient Pattern Detection using W_2 on Multivariate Normal Distributions. In *Proc. of (DAGM-OAGM)*.

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

[Kootstra and Kragic, 2011] Kootstra, G. and Kragic, D. (2011). Fast and bottom-up object detection, segmentation, and evaluation using gestalt principles. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).*

[Krähenbühl and Koltun, 2014] Krähenbühl, P. and Koltun, V. (2014). Geodesic object proposals. In *Proc. of the European Conference on Computer Vision (ECCV).* Springer.

[Lai et al., 2014] Lai, K., Bo, L., and Fox, D. (2014). Unsupervised Feature Learning for 3D Scene Labeling. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).*

[Lai et al., 2011] Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A Large-Scale Hierarchical Multi-view RBG-D Object Dataset. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).*

[Li et al., 2014] Li, Y., Hou, X., Koch, C., Rehg, J. M., and Yuille, A. L. (2014). The secrets of salient object segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 280–287. IEEE.

[Liao, 1993] Liao, S. X. (1993). *Image Analysis by Moments.* PhD thesis, The University of Manitoba.

[Liu et al., 2007] Liu, T., Sun, J., Zheng, N.-n., Tang, X., and Shum, H.-Y. (2007). Learning to Detect A Salient Object. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Liu et al., 2009] Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., and Shum, H.-Y. (2009). Learning to Detect a Salient Object. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).*

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int'l J. of Computer Vision (IJCV)*, 60(2):91–110.

[Malisiewicz and Efros, 2007] Malisiewicz, T. and Efros, A. A. (2007). Improving spatial support for objects via multiple segmentations. In *Proc. of the British Machine Vision Conference (BMVC).*

[Manén et al., 2013] Manén, S., Guillaumin, M., and Van Gool, L. (2013). Prime Object Proposals with Randomized Prim's Algorithm. In *Proc. of the IEEE International Conference on Computer Vision (ICCV).*

[Margolin et al., 2014] Margolin, R., Zelnik-Manor, L., and Tal, A. (2014). How to evaluate foreground maps? In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)).*

[Martín García and Frintrop, 2013] Martín García, G. and Frintrop, S. (2013). A Computational Framework for Attentional 3D Object Detection. In *Proc. of the Annual Conference of the Cognitive Science Society (CogSci).*

[Martín García et al., 2013] Martín García, G., Frintrop, S., and Cremers, A. B. (2013). Attention-based Detection of Unknown Objects in a Situated Vision Framework. *KI - Künstliche Intelligenz, Springer.*

[Martín García et al., 2015a] Martín García, G., Pavel, M., and Frintrop, S. (2015a). A Computational Framework for Attentional Object Discovery in RGB-D Videos. *Submitted to the Cognitive Processing Journal.*

[Martín García et al., 2015b] Martín García, G., Potapova, E., Werner, T., Zillich, M., Vincze, M., and Frintrop, S. (2015b). Saliency-based Object Discovery on RGB-D Data with a Late-Fusion Approach. *Proc. of the IEEE International Conference on Robotics and Automation (ICRA).*

[Mishra and Aloimonos, 2012] Mishra, A. K. and Aloimonos, Y. (2012). Visual segmentation of "simple" objects for robots. *Robotics: Science and Systems VII*, pages 1–8.

[Newcombe et al., 2011] Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR).*

[Palomino et al., 2011] Palomino, A. J., Marfil, R., Bandera, J. P., and Bandera, A. (2011). A novel biologically inspired attention mechanism for a social robot. *EURASIP Journal on Advances in Signal Processing*, 2011:4.

[Papon et al., 2013] Papon, J., Abramov, A., Schoeler, M., and Wörgötter, F. (2013). Voxel cloud connectivity segmentation - supervoxels for point clouds. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Perazzi et al., 2012] Perazzi, F., Krahenbuhl, P., Pritch, Y., and Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *CVPR*.

[Pirovano, 2012] Pirovano, M. (2012). Kinfu —an open source implementation of kinectfusion + case study: implementing a 3d scanner with pcl. Technical report, POLIMI (Politecnico di Milano).

[Posner et al., 1985] Posner, M. I., Rafal, R. D., Choate, L. S., and Vaughan, J. (1985). Inhibition of return: Neural basis and function. *Cognitive neuropsychology*, 2(3):211–228.

[Potapova et al., 2014] Potapova, E., Varadarajan, K. M., Richtsfeld, A., Zillich, M., and Vincze, M. (2014). Attention-driven Object Detection and Segmentation of Cluttered Table Scenes using 2.5 D Symmetry. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Pylyshyn, 2001] Pylyshyn, Z. W. (2001). Visual Indexes, Preconceptual Objects, and Situated Vision. *Cognition*, 80(1-2):127–158.

[Rensink, 2000] Rensink, R. (2000). The Dynamic Representation of Scenes. *Visual Cognition*, 7:17–42.

[Richtsfeld et al., 2012] Richtsfeld, A., Morwald, T., Prankl, J., Zillich, M., and Vincze, M. (2012). Segmentation of unknown objects in indoor environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[Rodieck, 1965] Rodieck, R. W. (1965). Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*.

[Rusu et al., 2009] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (FPFH) for 3D registration. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE.

[Schiebener et al., 2014] Schiebener, D., Ude, A., and Asfour, T. (2014). Physical Interaction for Segmentation of Unknown Textured and Non-textured Rigid Objects. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Szeliski, 2010] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

[Treisman and Gelade, 1980] Treisman, A. M. and Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, 12:97–136.

[Uijlings et al., 2013] Uijlings, J., van de Sande, K., Gevers, T., and Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*.

[Van de Sande et al., 2011] Van de Sande, K. E., Uijlings, J. R., Gevers, T., and Smeulders, A. W. (2011). Segmentation as selective search for object recognition. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*.

[Viola and Jones, 2004] Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57(2):137–154.

[von Hofsten and Spelke, 1985] von Hofsten, C. and Spelke, E. (1985). Object Perception and Object-directed Reaching in Infancy. *Journal of Experimental Psychology*, 144(2).

[Wagemans et al., 2012] Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., and von der Heydt, R. (2012). A Century of Gestalt Psychology in Visual Perception: I. perceptual Grouping and Figure-Ground Organization. *Psychological Bulletin*.

[Walther and Koch, 2006] Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*.

[Yan et al., 2013] Yan, Q., Xu, L., Shi, J., and Jia, J. (2013). Hierarchical Saliency Detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Yang et al., 2013] Yang, C., Zhang, L., Lu, H., Ruan, X., and Yang, M.-H. (2013). Saliency Detection via Graph-based Manifold Ranking. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhu et al., 2014] Zhu, L., Cao, Z., Klein, D. A., Frintrop, S., and Cremers, A. B. (2014). A multi-size superpixel approach for salient object detection based on multivariate normal distribution estimation. *TIP*, 23(12).