



Federal University of Bahia
Programme of Post-graduation in Mechatronics

**Multiscale Spectral Residue for Faster
Image Object Detection**

José Grimaldo da Silva Filho

2012

Multiscale Spectral Residue for Faster Image Object Detection

José Grimaldo da Silva Filho

*Submitted in partial fulfillment of
the requirements for the degree of
Master in Mechatronics*

Programme of Post-graduation in Mechatronics
Federal University of Bahia

under supervision of
Prof. Dr. Luciano Oliveira (advisor)
Prof. Dr. Leizer Schnitman (co-advisor)

S586 Silva, José Grimaldo
Multiscale spectral residue for faster image object detection
/ José Grimaldo da Silva Filho. – Salvador, 2012.
60 f. : il. color.

Orientador: Prof. Dr. Luciano Rebouças de Oliveira

Dissertação (mestrado) – Universidade Federal da Bahia.
Escola Politécnica, 2012.

1. Visão por computador. 2. Processamento de imagens. 3.
Visão de robô. I. Oliveira, Luciano Rebouças. II. Universidade
Federal da Bahia. III. Título.

CDD: 004

TERMO DE APROVAÇÃO

JOSÉ GRIMALDO DA SILVA FILHO

MULTISCALE SPECTRAL RESIDUE FOR FASTER IMAGE OBJECT DETECTION

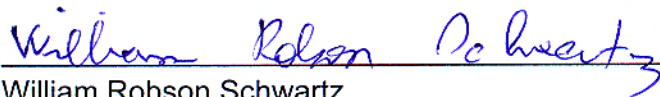
Dissertação aprovada como requisito parcial para obtenção do grau de Mestre
em Mecatrônica pela Universidade Federal da Bahia – UFBA

Aprovada em 18 de janeiro de 2013



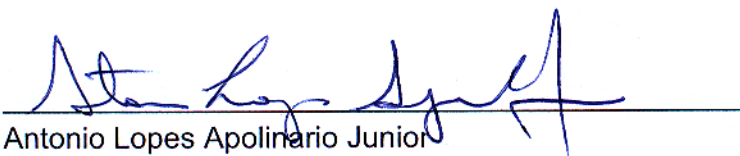
Luciano Rebouças de Oliveira – Orientador

Doutor em Engenharia Elétrica e de Computadores na Universidade de Coimbra
Universidade Federal da Bahia



William Robson Schwartz

Doutor em Ciência da Computação na Universidade de Maryland, College Park, EUA.
Universidade Federal de Minas Gerais



Antonio Lopes Apolinário Junior

Doutor em Engenharia de Sistemas e Computação na Universidade Federal do Rio de Janeiro
Universidade Federal da Bahia



Vinicius Moreira Mello

Doutor em Matemática no Instituto Nacional de Matemática Pura e Aplicada
Universidade Federal da Bahia

To those who shared the dreams

Acknowledgments

There were many who stood beside me during this process. I can't find the words to properly thank so much help and consideration. Even so, at least I am willing to try.

What I can say is that I will always be thankful to those who walked beside me, who shared both dreams and hardships. These friends and family are the most precious gifts I could ever dream of, and for that I am thankful. There will certainly be other hardships along this road. Even so I can still thread safely along it, for I have such precious companions, always willing to help me become a stronger man, even stronger than I would have believed capable.

I won't forget this, for what I am today I owe to all of you.

Abstract

Accuracy in image object detection has been usually achieved at the expense of much computational load. Therefore a trade-off between detection performance and fast execution commonly represents the ultimate goal of an object detector in real life applications.

Most images are composed of non-trivial amounts of background information, such as sky, ground and water. In this sense, using an object detector against a recurring background pattern can require a significant amount of the total processing time. To alleviate this problem, search space reduction methods can help focusing the detection procedure on more distinctive image regions.

Among the several approaches for search space reduction, we explored saliency information to organize regions based on their probability of containing objects. Saliency detectors are capable of pinpointing regions which generate stronger visual stimuli based solely on information extracted from the image. The fact that saliency methods do not require prior training is an important benefit, which allows application of these techniques in a broad range of machine vision domains. We propose a novel method toward the goal of faster object detectors. The proposed method was grounded on a multi-scale spectral residue (MSR) analysis using saliency detection. For better search space reduction, our method enables fine control of search scale, more robustness to variations on saliency intensity along an object length and also a direct way to control the balance between search space reduction and false negatives caused by region selection. Compared to a regular sliding window search over the images, in our experiments, MSR was able to reduce by 75% (in average) the number of windows to be evaluated by an object detector while improving or at least maintaining detector ROC performance. The proposed method was thoroughly evaluated over a subset of LabelMe dataset (person images), improving detection performance in most cases. This evaluation was done comparing object detection performance against different object detectors, with and without MSR. Additionally, we also provide evaluation of how different object classes interact with MSR, which was done using Pascal VOC 2007 dataset. Finally, tests made showed that window selection performance of MSR has a good scalability with regard to image size. From the obtained data, our conclusion is that MSR can provide substantial benefits to existing sliding window detectors.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 2 |
| 1.2 | Goals | 4 |
| 1.3 | Key contributions | 4 |
| 1.4 | Chapter map | 5 |
| 2 | Background | 7 |
| 2.1 | Saliency detection methods | 7 |
| 2.1.1 | Local contrast | 8 |
| 2.1.2 | Global contrast | 9 |
| 2.1.3 | Frequency domain analysis | 12 |
| 2.1.4 | Learning-based methods | 13 |
| 2.2 | Object search | 15 |
| 2.2.1 | Contextual methods | 17 |
| 2.2.2 | Branch-and-bound methods | 18 |
| 2.2.3 | Saliency-based methods | 20 |
| 2.3 | Object detectors | 21 |
| 2.3.1 | Feature extraction | 22 |
| 2.3.2 | Classification | 24 |
| 2.4 | Datasets | 26 |
| 2.5 | Relation to our work | 28 |
| 3 | Saliency analysis | 31 |
| 3.1 | On the use of saliency detectors | 33 |
| 3.1.1 | Runtime speed | 34 |
| 3.1.2 | Saliency map | 34 |
| 3.1.3 | Selection of search scale | 36 |
| 3.1.4 | Saliency for faster detection | 37 |
| 3.2 | Spectral residue | 38 |
| 3.2.1 | Statistical properties of natural images | 38 |
| 3.2.2 | Exploring $1/f$ law | 40 |

| | | |
|----------|---|-----------|
| 3.2.3 | Known issues | 42 |
| 3.3 | Closure | 44 |
| 4 | Multi-scale spectral residual object search | 47 |
| 4.1 | Selecting regions | 49 |
| 4.2 | Saliency over multiple scales | 50 |
| 4.3 | Quality function threshold | 52 |
| 4.4 | β and k values | 53 |
| 4.5 | Image pre-processing | 55 |
| 4.6 | Closure | 59 |
| 5 | Experimental evaluation | 61 |
| 5.1 | Experiments | 63 |
| 5.1.1 | Comparison of saliency methods in a multi-scale structure | 64 |
| 5.1.2 | Scalability | 66 |
| 5.1.3 | Detection performance | 67 |
| 5.1.4 | Runtime performance | 69 |
| 5.1.5 | Per-class MSR performance | 70 |
| 5.2 | Analysis and closure | 72 |
| 6 | Conclusion | 75 |
| A | Contrast normalization | 77 |
| A.1 | Histogram equalization | 77 |
| A.2 | Adaptive histogram equalization | 78 |
| | References | 81 |

List of Tables

| | | |
|-----|---|----|
| 3.1 | Comparison of runtime speeds for saliency detection over Achanta et al. dataset [Achanta <i>et al.</i> 2009]. Results taken from [Cheng <i>et al.</i> 2011a], which used a dataset where most images have size around 400 by 300 pixels | 34 |
| 3.2 | Resolution of generated saliency map when compared to the original image <i>S</i> . Part of the data is from [Achanta <i>et al.</i> 2009]. | 35 |
| 4.1 | Window selection performance over different parameters for AHE algorithm and compared to HE. Best results for each section are marked in bold . . . | 56 |
| 4.2 | Window selection performance over different parameters over a person dataset for CLAHE algorithm compared to HE and AHE. Best results for each section are marked in bold. | 57 |
| 4.3 | Results of window selection performance from a combination of both CLAHE and HE methods. | 58 |
| 4.4 | Comparison of runtime speed differences for saliency detection (MSR with β scaling) using different contrast normalization approaches | 59 |
| 5.1 | SFNR for each method at 20% of WSR | 65 |
| 5.2 | SFNR for each method at 30% of WSR | 65 |
| 5.3 | Runtime speed proportion for each method | 70 |
| 5.4 | Comparison of MSR performance using a hypothetical flawless detector in the Pascal VOC 2007 dataset. | 71 |

List of Figures

| | | |
|------|--|----|
| 1.1 | An abstract process which, given an input image (leftmost image) generates an output (rightmost image) with the regions with highest probability of containing objects | 2 |
| 2.1 | Comparison of several saliency methods. Image taken from [Achanta <i>et al.</i> 2009] | 10 |
| 2.2 | Examples of saliency detection and segmentation using [Cheng <i>et al.</i> 2011a]. In the second row the saliency maps generated from the input images from the first row. The third row are the object masks generated from the saliency map. Image taken from [Cheng <i>et al.</i> 2011a]. | 11 |
| 2.3 | Examples of saliency map generated by quaternion fourier transform. Image adapted from [Guo <i>et al.</i> 2008]. | 13 |
| 2.4 | The original image (a), participants gaze is then inspected for the image and forms the groundtruth (b); the image (c) represents the saliency map generated by using a gaussian filter over the fixation of the participants. Finally, the (d) image represents its 20% most salient parts. Image from [Judd <i>et al.</i> 2009]. | 14 |
| 2.5 | For each column, from left to right: the input image, multi scale contrast, center-surround, spatial color distribution and the final (binary) saliency map generated by the CRF. Image extracted from [Liu <i>et al.</i> 2011]. | 15 |
| 2.6 | Example of feature extraction by [Perko & Leonardis 2007]. From left to right: input image with a yellow star over the object of interest, image cropped over object, features extracted over five radii and twelve orientations, resulting feature vectors. Image from [Perko & Leonardis 2007]. | 18 |
| 2.7 | Matching a reference image against a dataset. Image taken from [Keysers <i>et al.</i> 2007] | 19 |
| 2.8 | Detection a region of interest using Difference of Gaussians. Image from [Yiu & Varshney 2011] | 20 |
| 2.9 | Some types of Haar-like features | 23 |
| 2.10 | A histogram of oriented gradients gradient values are accumulated over blocks, cells and bins. Image taken from [Oliveira 2010]. | 24 |
| 2.11 | Images from MIT+CMU dataset [Schneiderman & Kanade 1998, Schneiderman & Kanade 2000] | 26 |

| | | |
|------|---|----|
| 2.12 | Images from INRIA dataset [Dalal & Triggs 2005] | 26 |
| 2.13 | Images and the annotated objects from Pascal Voc Website [Everingham <i>et al.</i> 2012] | 27 |
| 2.14 | Example images of the PETS2006 dataset extracted from [PETS 2006] . . . | 27 |
| 3.1 | Example of thumbnails generated for images. Image taken from [Hou & Zhang 2008]. | 33 |
| 3.2 | Some limitations of saliency methods analysed in [Achanta <i>et al.</i> 2009]. Limitations range from non-uniform object highlighting and highlighting only salient regions smaller than a certain filter size. | 35 |
| 3.3 | General architecture of IT. The input images are separated into color, ori- entation and intensity maps over several scales. The maps are combined using a normalization operator generating the final saliency map. Image taken from [Itti <i>et al.</i> 1998]. | 36 |
| 3.4 | Example of scale control through image resizing in SR. In the larger im- ages smaller objects are more salient. Conversely, as the image is down- sampled bigger, objects start attracting more attention. Image taken from [Hou & Zhang 2007]. | 37 |
| 3.5 | Power spectrum averaged over orientations (thick lines) and also their linear fits (thin lines). In these images p denotes log frequency and $S(p)$ the log amplitude over frequency. Images taken from [Hsiao & Millane 2005]. . . . | 39 |
| 3.6 | First row shows samples of the image ensemble. Second row shows differ- ences in generated log spectrum representation generated with ensembles of different sizes. Image taken from [Hou & Zhang 2007]. | 40 |
| 3.7 | Comparison between log spectrum and log-log spectrum, where the first image is an average of 2277 natural scenes. Image taken from [Hou & Zhang 2007]. | 41 |
| 3.8 | Calculation of spectral residue: (a) input image, (b) calculated log spec- trum, (c) convoluted log spectrum, (d) spectral residue. The residue is obtained by subtracting the log spectrum from the residue. Image taken from [Hou & Zhang 2007]. | 42 |
| 3.9 | Comparison of PFT with SR. Image taken from [Guo <i>et al.</i> 2008]. | 44 |
| 4.1 | Overview of MSR structure for window selection. | 48 |
| 4.2 | Differences between original SR pixel selection and MSR window selection. From left to right: input image, SR per-pixel selection, MSR per-window selection | 49 |

| | | |
|-----|---|----|
| 4.3 | Difference of runtime speed required to select windows for an entire image octave using MSR with and without integral images for window saliency mean calculation. | 50 |
| 4.4 | Differences in saliency at multiple scales. In the left image, SR was calculated in 15% of the original image size, generating strong reactions on mostly small objects; in the right, using 7% of the original image size, bigger objects were also selected. The image reduction examples demonstrate how the image size influences on the scale of saliency detection, which will be tuned to best select objects in a given octave. | 51 |
| 4.5 | Trade-off curve for person detection using different β values. When the curve is closer to the origin it is better. | 53 |
| 4.6 | Comparison between multi-scale analysis and using the same saliency map for all scales. Methods presented are MSR and LC [Zhai & Shah 2006]. When the curve is closer to the origin it is better. | 54 |
| 4.7 | Different regions selected depending on threshold value. | 55 |
| 4.8 | Comparison of best results in terms of window selection performance versus false negatives generated over a person dataset. CLAHE is presented with 0.3 as contrast limit. | 58 |
| 4.9 | MSR window selection procedure (parameters for person detection) | 59 |
| 5.1 | Sample images of the dataset created from the LabelMe repository. | 62 |
| 5.2 | Sample images from the Pascal VOC 2007 dataset. | 62 |
| 5.3 | Comparison of best results from different saliency methods applied to guide multi-scale detectors, the methods are MSR (using CLAHE + HE), MSR HE [Silva <i>et al.</i> 2012], FT [Achanta <i>et al.</i> 2009], GB [Harel <i>et al.</i> 2007], IT [Itti <i>et al.</i> 1998], LC [Zhai & Shah 2006] and a baseline using random window scoring. The false negative rate represents only objects that the detector would have matched if a regular sliding window approach had been used. When the curve is closer to the origin it is better. | 64 |
| 5.4 | Comparison of window selection results using MSR (with HE + CLAHE) against a cluttered and the complete LabelMe dataset. | 66 |
| 5.5 | Trade-off between WSR and SFNR at different starting resolutions. Aspect ratio is kept by approximating the image resolution to the closest image size. When the curve is closer to the origin it is better. | 67 |
| 5.6 | Relation between number of windows at each image size and number of windows selected for the detector. An operating point was selected at 20% of WSR (see Fig. 5.3 for reference). | 68 |

| | | |
|------|--|----|
| 5.7 | ROC curve showing differences between person detection performance using a regular sliding window and MSR. | 69 |
| 5.8 | Positive results from using a HOG/SVM detector with MSR, positive results at 30% of WSR after non-max suppression. Blue rectangles indicate avoided false positives (improving performance); TP are marked with green. | 69 |
| 5.9 | Negative results from using a HOG/SVM detector with MSR, positive results at 30% of WSR after non-max suppression. Yellow rectangles indicate FN caused by MSR (affecting performance); blue rectangles indicate avoided false positives (improving performance); TP are marked with green, while red rectangles are FP. | 70 |
| 5.10 | Positive results from using a Viola Jones detector with MSR, positive results at 30% of WSR after non-max suppression. Blue rectangles indicate avoided false positives (improving performance); TP are marked with green and FP with red. | 70 |
| A.1 | Differences between the image before and after histogram equalization; the equalization was applied image-wide and the images presented are cropped around the object of interest. Original images from LabelMe [Russell <i>et al.</i> 2008] | 77 |
| A.2 | Demonstration of how a pixel is contrast normalized based on its immediate neighborhood. Image from public domain. | 78 |
| A.3 | Contrast mapping functions and each generated clipped histogram. Image adapted from [Pizer <i>et al.</i> 1987]. | 79 |

List of Acronyms

| | |
|-----------------|---|
| AC | Achanta's method. |
| Adaboost | adaptive boosting. |
| BB | branch-and-bound. |
| CA | context aware. |
| CLAHE | contrast limited adaptive histogram equalization. |
| CMU | Carnegie Mellon University. |
| CRF | conditional random fields. |
| ESS | Efficient Subwindow Search. |
| FT | frequency-tuned. |
| GB | graph-based. |
| GLCM | grey-level co-occurrence matrix. |
| GPU | graphics processing unit. |
| HC | histogram contrast. |
| HE | histogram equalization. |
| HOG | histogram of oriented gradients. |
| HSV | hue, saturation and value. |
| IT | Itti's Method. |
| k-NN | k-Nearest Neighbor. |
| LC | luminance contrast. |
| MIT | Massachusetts Institute of Technology. |
| MP | mega-pixels. |
| MSR | multi-scale spectral residue. |

| | |
|--------------|--|
| MZ | saliency method developed by Ma and Zhang. |
| NICTA | National ICT Australia Limited. |
| NMS | non-max suppression. |
| PETS | Performance Evaluation of Tracking and Surveillance. |
| PFT | phase-only Fourier transform. |
| RC | region contrast. |
| RGB | red green blue. |
| ROC | receiver operating characteristic. |
| SFNR | saliency false negative rate. |
| SIFT | scale-invariant feature transform. |
| SR | spectral residual. |
| SRM | structural risk minimization. |
| SVM | support vector machine. |
| VC | Vapnik-Chervonenski. |
| VOC | Visual Object Classes. |
| WSR | window selection rate. |

Glossary

| | |
|--------------|---|
| AC | Calculate saliency based a contrast maps over multiple scales. |
| AHE | Adaptive Histogram Equalization method that normalizes each pixel based on its local neighborhood pixel distribution. |
| CA | Mixed saliency method based on top-down and bottom-up information and also low-level, mid-level and high-level features. |
| CLAHE | Adaptive histogram equalization methods that relies on limiting noise amplification (common in adaptive histogram equalization methods) to achieve better contrast normalization. |
| FT | Saliency of a pixel is defined as from the average pixel distance from other pixels in LAB space. |
| GB | Saliency method using a graph-based approach for saliency detection. Saliency of a region is calculated by finding its dissimilarity in relation to its neighbors. |
| HC | Saliency detection method based on the rarity principle, using spatial contrast and spatial coherence. |
| HE | Histogram Equalization provides image-wise histogram equalization by using the image histogram to spread out the most frequent intensity values. |

| | |
|----------------|---|
| IT | Saliency detection method based on local contrast. It calculates orientation, intensity and color maps over multiple scales and uses a normalization operator to combine maps while giving higher priority to the ones with a distinctive peak. |
| LabelMe | An online and user-driven annotation and upload tool for images. |
| LC | Calculate saliency of a region based on its global contrast. |
| MSR | Our approach for window selection based on saliency information. |
| NMS | The non-max suppression combine detection rectangles which are similar in position and size into a single rectangle, thus avoiding or at least reducing repeated detections.. |
| PFT | Saliency detection method based on phase information of the frequency domain. |
| SFNR | The Saliency False Negative Rate indicates how many false negatives were caused by saliency. This is achieved by measuring what percentage of true positives were lost by using MSR. |
| SR | Frequency-based saliency calculation which detect objects of interest by analyzing properties of an ensemble of natural images. |
| WSR | The rate of windows selected from the total for actual object detection. |

List of Symbols

| | |
|----------------------|---|
| β | Resizing factor designed to change the search scale of the saliency detection to match that of the object detector. |
| $H(I)$ | Histogram of image I . |
| $H_i(I)$ | i th bin of histogram $H(I)$. |
| $\ \cdot\ _2$ | L_2 -norm distance. |
| N_R | Number of quality function evaluations of rectangular region in a sliding window approach.. |
| $\mathcal{N}(\cdot)$ | Normalization operator that combines several saliency maps into one. Gives more importance to maps that have a small number of strong responses.. |
| $a \triangleq b$ | a is defined as b . |
| R | denotes a rectangle within an image. |
| i, j, p, q | Image pixel position. |
| I_k | k -th image I pixel. |
| $I(i, j)$ | Value of pixel at position (i, j) of image I . |
| I^l | the l th image scale. |

Introduction

“For every complex problem there is an answer that is clear, simple, and wrong.”
H. L. Mencken”

Interest in machine vision algorithms has increased in recent years. The widespread use of computer vision intensive applications require the use of robust techniques that demand non-trivial amounts of computing power. Given these recent trends, techniques to reduce object detection times, and provide even faster responses, have attracted attention of the research community, for so many years [Viola & Jones 2001, Zhu *et al.* 2006, Prisacariu & Reid 2009].

In the scope of faster detection, when processing an image in search of a certain object, it is possible to assume that, in most cases, the object of interest will be found in only a fraction of the search space. This is so because images include background information, which gives objects contained within it both location and cultural context, but do not actually define the object of interest. Therefore we can assume that object search operations dedicate a significant amount of the total time processing background patterns, such as: sky, earth, water, walls and roads.

Given the aforementioned problem, by making an object detector focus only on more distinctive image regions one can alleviate total processing time. To achieve such thing, it is necessary to provide a function that, given an input image, is able to pinpoint which regions are worth further verification. An overview of this general procedure is depicted in Figure 1.1.

Another potential benefit of search space reduction is to allow for extra protection against false positives, that is, regions that a detector would falsely assume to be an object. In this case, a non-object region could be discarded before actual object detection, avoiding a potential false positive.

Determining which regions to remove is a challenging task. Incorrectly removing an object region will negatively affect detector performance and diminish the utility of such



Figure 1.1: An abstract process which, given an input image (leftmost image) generates an output (rightmost image) with the regions with highest probability of containing objects

method. Conversely, in case too few regions are removed, the algorithm may not improve runtime speed in a noticeable manner.

This work presents a solution aimed at harnessing the benefits of search space reduction and also to avoid most of its shortcomings. Our solution explores saliency information to sort image regions based on their importance. The regions which are within a previously defined importance threshold are selected. This process is then repeated over multiple image scales to capture important objects of different sizes.

In the remainder of this chapter we describe motivation, goals, contributions and description of the remaining of this work.

1.1 Motivation

Many widely used techniques for object detection and localization have achieved remarkable results in real life situations, such as [Dalal & Triggs 2005, Viola & Jones 2001]. Those positive results, however, demand for extra computational cost. As such, for many applications, reaching a good balance between detection results and computational cost is a challenging task. In particular, applications which have strict time-constrained requirements may have to settle for worse performing algorithms to achieve its runtime requirements. Common examples of such applications are perception for driver assistance (see a survey in [Enzweiler & Gavrila 2009]), video traffic analysis (see a survey in [Kastrinaki *et al.* 2003]) and surveillance systems (see a survey in [Hu *et al.* 2004]).

Computational cost of object detection becomes an even greater issue when taking into account availability of high resolution images which demand additional processing time. Recent advances in camera technology have produced a regular increase in megapixel resolution of mainstream cameras [Yiu & Varshney 2011]. Compared to the year 2000,

mainstream cameras average resolution on 2010 has increased from over 3 MP to 18 MP. Larger resolutions force image processing systems to either downsample the image and potentially discard important information or to find ways to deal with additional overhead.

Considering the aforementioned issues, by lessening the impact of growing processing requirements on existing computer vision systems, one can enable the use of more robust algorithms or can have a higher image processing throughput. To such end, a broad range of solutions have been developed. For instance, Zhu et al. [Zhu *et al.* 2006] and Viola and Jones [Viola & Jones 2001] have developed rejection cascades, reducing the time required to reject non-objects. These works were based on the so called dense sliding window search. This search technique relies on a fixed-size window that is moved over the image, and at each distinct position the detector quality function is evaluated. In contrast, some alternate approaches rely on a branch-and-bound technique [Lampert *et al.* 2008, Keysers *et al.* 2007], which is capable of discarding several regions simultaneously based on their bounding function quality score.

We chose a different approach to speed up an object detector, which relies on search space reduction using saliency information, instead. The question posed by our work is: *What particular characteristics of image objects can be used to reduce the search space of an object detector?*

To capture information particular to objects, we rely on saliency information. As saliency detectors are able to locate regions which stand out more in an image, they can be used in a broad spectrum of applications – from thumbnail generation [Hou & Zhang 2008] to semantic colorization [Chia *et al.* 2011]. Examples of such saliency methods are found in [Hou & Zhang 2007], which uses statistical properties of natural scenes to select regions of interest, and also in [Itti *et al.* 1998] based on the computation of saliency inspired on the pre-attentive phase of human visual system, responsible for drawing attention to specific parts of the visual stimuli.

To summarize, the aforementioned problems motivate our choice for search space reduction. To provide such reduction, saliency information is used. This choice for saliency is a consequence of its demonstrated usefulness in a large range of related applications. As such, saliency, in this work, is used to provide additional information about an image, helping to select likely object regions.

1.2 Goals

Sliding window detectors are the most common technique for high-performance object detection systems [Divvala *et al.* 2009]. Using a dense search with a fixed-size window over the image allows the sliding window to detect and localize objects. However, such dense search on the image space imposes a high processing overhead. Therefore, reducing the number of windows can make object detection faster, at the cost of negatively impacting detection performance and localization accuracy. Thus, we explore how to enable saliency detection to be capable of selecting which windows to discard, before actual object detection is applied. Given this direction, the goals are:

1. **To speed up object detection by means of window pre-selection.** To achieve effective detector speed up, the decision to select or discard an image before object detection has to be much faster than the cost of evaluating a detector quality function.
2. **To increase or to maintain detector performance.** This goal is necessary, as without it, the first goal could be achieved through random selection of windows to discard, with negative impact on detection effectiveness. Thus, our aim is to provide accurate window selection, with little to none mistakenly discarded objects. Additionally, performance can also be improved by avoiding potential false positives, that is, discarding regions that would have been incorrectly considered objects by a detector.

1.3 Key contributions

Our method, called multi-scale spectral residue (MSR), aims to achieve a better trade-off between the number of windows selected to be evaluated by a detector and the number of miss-detections. To do so, we developed a solution to select image regions based on their saliency information over multiple scales. Proper use of saliency information for the task of window selection poses many challenges, as the number of different saliency detection techniques, each based on distinct concepts, requires detailed understanding of their differences. An earlier iteration of our technique was published at [Silva *et al.* 2012].

MSR has demonstrated an average reduction of 75% of windows to be evaluated, while keeping or improving detection performance. Such results are important in several domains, such as: mechatronics, surveillance and satellite image analysis. These search

space reduction solutions can help alleviate the burden of image processing for time-constrained systems.

1.4 Chapter map

The rest of this document is structured as follows:

- **Chapter 2** presents the background information about the main topics related to our work. Included in this chapter are previous approaches, related concepts and general considerations.
- **Chapter 3** describes the range of applications in which saliency methods can be applied. Furthermore, details are also provided about practical differences between saliency methods. This information is used to select the best aligned saliency method for the task of search space reduction.
- **Chapter 4** presents our approach for search space reduction and also the different configurations that can be made on it. In this section, a methodology for measuring window selection performance is also presented, which will be useful for comparison of results.
- **Chapter 5** evaluates several characteristics of our method, including runtime speed, effect on detector performance, comparison against other saliency methods and also performance achieved with several different object classes. A discussion and an analysis of the obtained results are also presented to summarize the gathered information.
- **Chapter 6** concludes our work, with discussions and future work.

Background

Contents

| | | |
|-------|--------------------------------------|-----------|
| 2.1 | Saliency detection methods | 7 |
| 2.1.1 | Local contrast | 8 |
| 2.1.2 | Global contrast | 9 |
| 2.1.3 | Frequency domain analysis | 12 |
| 2.1.4 | Learning-based methods | 13 |
| 2.2 | Object search | 15 |
| 2.2.1 | Contextual methods | 17 |
| 2.2.2 | Branch-and-bound methods | 18 |
| 2.2.3 | Saliency-based methods | 20 |
| 2.3 | Object detectors | 21 |
| 2.3.1 | Feature extraction | 22 |
| 2.3.2 | Classification | 24 |
| 2.4 | Datasets | 26 |
| 2.5 | Relation to our work | 28 |

2.1 Saliency detection methods

An object “stands out” in a scene if it has strong contrast in relation to its neighborhood. Man-made objects such as stoplight and traffic signs are created to explore this property in order to be perceived faster than its surroundings. Image saliency detection methods, in this regard, are able to detect regions that draw more visual attention.

Approaches to salient region detection can be broadly categorized in four types: local contrast, global contrast, frequency domain analysis and learning based. Next, each one of them will be described along with the main works for each category.

2.1.1 Local contrast

Work on object analysis indicates that objects can be described solely through local information [Kadir & Brady 2001]. In this sense, algorithms for salient region detection using local information decide how “unique” a given pixel is in relation to its immediate surroundings, where the concept of uniqueness varies for each solution.

One of the first approaches to salient region detection, [Itti *et al.* 1998] is based on the behavior of the early primate visual system. Detection of salient regions uses center-surround features, in which the center of a region generates strong feature response while its (weaker) surrounding regions inhibit it. This center-surround feature is calculated as the difference between high frequency (fine scale) and low frequency components (coarse scale) of a region. The across-scale difference between each region is determined from interpolation of the coarse scale to the finer scale followed by point-by-point subtraction.

The center-surround features are calculated over color, intensity and orientation spaces over multiple image scales to detect salient points with a wide range of characteristics. The saliency maps generated from multiple scales and spaces are then combined to generate a final saliency map using

$$S_i = \frac{1}{3}(\mathcal{N}(\bar{\mathcal{I}}) + \mathcal{N}(\bar{\mathcal{C}}) + \mathcal{N}(\bar{\mathcal{O}})), \quad (2.1)$$

where $\bar{\mathcal{I}}$ represents the multiple intensity maps, $\bar{\mathcal{C}}$ and $\bar{\mathcal{O}}$ denote color maps and orientation maps respectively. The multiple scales of a given space are combined through a normalization function $\mathcal{N}(\cdot)$, that gives more importance to maps that have only a small number of strong responses. The three resulting normalized saliency maps are summed and averaged to generate a single saliency map.

A similar approach to [Itti *et al.* 1998] was presented in [Harel *et al.* 2007], which uses a graph-based approach for saliency detection. In this approach, saliency of a region is calculated from a dissimilarity measure from its neighbors. Given two points $I(i, j)$ and $I(p, q)$, on an image I , this dissimilarity is defined as

$$d((i, j) || (p, q)) \triangleq \left| \log \frac{I(i, j)}{I(p, q)} \right|, \quad (2.2)$$

where \triangleq represents “defined as”, while $d(\cdot||\cdot)$ the dissimilarity function itself. From this dissimilarity, the image can be represented as a fully connected directed graph, where each node is connected to every other node in the image. For each connection a weight is assigned using

$$w((i, j), (p, q)) \triangleq d((i, j)|| (p, q)) \cdot F(i - p, j - p), \quad (2.3)$$

where F is defined as

$$F(a, b) \triangleq \exp\left(-\frac{a^2 + b^2}{2\sigma^2}\right), \quad (2.4)$$

where σ denotes a free parameter defined empirically. The formulation for $w(\cdot)$ assigns more weight to nodes which are similar and spatially close to one another. From this concept, a Markov Chain is applied over the graph so that regions that are more locally dissimilar will generate a stronger saliency.

2.1.2 Global contrast

Unlike local contrast saliency detection approaches, global contrast methods rely on information of the entire scene to decide how salient a region is. This concept agrees with human intuition where less frequent features are more likely to stand out in a scene [Feng *et al.* 2011].

Using the concept of rarity, and based on the psychological studies about human perception sensitivity to contrast, [Zhai & Shah 2006] present a method for detection of prominent actions in video sequences. In this model, saliency of a pixel can be defined as

$$S_z(I_k) = \sum_{\forall I_i \in I} ||I_k - I_i||, \quad (2.5)$$

where I_i and I_k are pixels intensity value at i th (and k th) position in an image and $||\cdot||$ represents the Euclidean distance. This way, this model assumes that pixels with greater contrast in relation to other pixels are more salient. Equation 2.5 can also be represented as

$$S_{z'}(I_k) = \sum_{n=1}^{255} f_n ||I_k - I_n||, \quad (2.6)$$

where I_n denote an image pixel intensity value, f_n is the frequency of a particular pixel intensity, while n is a pixel intensity. In equation 2.6, the sum operator calculates the

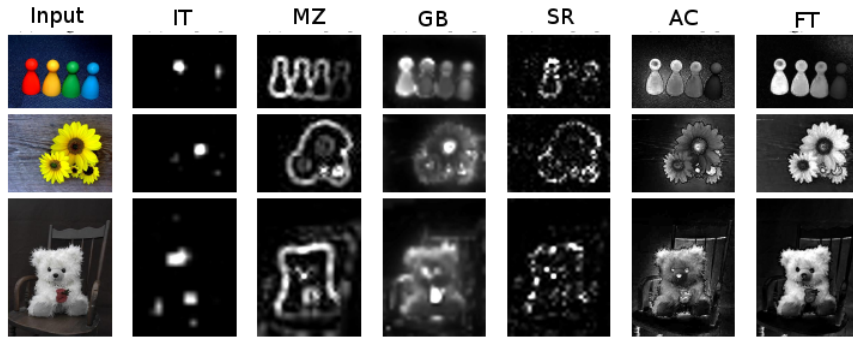


Figure 2.1: Comparison of several saliency methods. Image taken from [Achanta *et al.* 2009]

weighted difference of a_m from each distinct intensity value, all these values are within the range of $[0, 255]$. The algorithm complexity is dependent on the number of colors n . To avoid an overhead in runtime speed only the luminance channel of the LAB space is used for contrast calculation.

An earlier iteration of saliency detection by Achanta *et al.* (2008) finds saliency by generating contrast maps at multiple image scales. These maps are then combined to generate the final saliency map [Achanta *et al.* 2008]. More recently, Achanta *et al.* (2009) model the saliency of a pixel as its average distance from others in LAB space, using

$$S_a(x, y) = \|\mathbf{I}_\pi - \mathbf{I}(x, y)\|_2, \quad (2.7)$$

where \mathbf{I}_π is the mean image feature vector, $\mathbf{I}(x, y)$ is the original pixel value, $\|\cdot\|_2$ represents an L_2 -norm where each pixel is a feature vector of type $[L, a, b]$. In order to avoid high frequency noise and fine texture the input image is blurred using a 5×5 gaussian filter. This method preserves most of the high frequency content, thus generating high-resolution saliency maps unlike [Itti *et al.* 1998, Harel *et al.* 2007, Hou & Zhang 2007].

A comparison of Achanta *et al.* (2009) with other saliency methods is presented in Fig. 2.1. In this example, the high resolution of [Achanta *et al.* 2009] results is a distinctive feature.

One limitation of the work presented in Achanta *et al.* (2009) is that it only considers first-order average color information, this may affect its performance on more intricate patterns that are common on natural scenes [Cheng *et al.* 2011a]. Thus, in the work of Cheng *et al.* (2011), an approach was made based on:

- **spatial contrast**, defined by the rarity principle, that is, less frequent regions are



Figure 2.2: Examples of saliency detection and segmentation using [Cheng *et al.* 2011a]. In the second row the saliency maps generated from the input images from the first row. The third row are the object masks generated from the saliency map. Image taken from [Cheng *et al.* 2011a].

more salient. This definition is somewhat similar to the concept defined in Equation 2.6 from Zhai and Shah (2006). However, instead of using only the luminance channel and discarding potentially useful color information as [Zhai & Shah 2006], each color channel is quantized to 12 distinct values. The number of colors is further reduced from 12^3 to roughly 85 through elimination of colors that are too rare to be relevant for saliency detection. Furthermore, to avoid quantization artifacts, a color space smoothing is applied that replaces the saliency value of each color by an average of similar color saliency values;

- **spatial coherence**, representing spatial distances between image regions and how the distance affects their saliency intensity. That is, regions with high contrast with spatially close regions are more likely to be truly salient than when compared to regions with high contrast only to distant image regions.

The results of this approach can be seen in Fig. 2.2, where object segmentation is obtained from solely the saliency information.

2.1.3 Frequency domain analysis

The Fourier transform allows an image to be represented in the frequency domain. This transform gives spectral information that can be used to search for characteristics that are recurrent in salient regions.

As properties common to salient objects are hard to conceptualize, the Spectral Residual [Hou & Zhang 2007] explores properties of the background. For that, the $1/f$ law states that over an ensemble of natural images, the fourier spectrum will obey the distribution

$$E\{A(f)\} \propto 1/f, \quad (2.8)$$

where $E\{A(f)\}$ is the average amplitude over frequency f in the ensemble of natural images. As the $1/f$ law is likely not to hold true in individual images, parts of the spectrum that generate larger differences between the expected and the actual distribution are likely to contain novel information [Hou & Zhang 2007]. These regions with larger differences were interpreted by Hou and Zhang (2007) as composing possible objects in an image. To calculate such deviations the following formulation was used:

$$B = L(A) - h_n * L(A), \quad (2.9)$$

where $L(A)$ is the log amplitude of the fourier domain, and h_n is a 2D convolution filter of size $n \times n$. This formulation intends to capture regions containing statistical singularities, which jump out of the expected $1/f$ distribution. The residue B is then re-combined with the phase information and inverse fourier transformed to generate the final saliency map. This method is further detailed in Section 3.2.

Also using spectral information, [Guo *et al.* 2008] add spatio-temporal information to a salient detector based on phase information from the frequency domain. This method relies on the fact that phase information has information about salient regions of an image.

In this method, each image frame is represented by: one motion, one intensity and two color channels. The intensity channel is an average of the three channels from the image in RGB, while the motion information is calculated by subtracting the current frame from its preceding one. These individual channels are combined into a quaternion image in which a quaternion fourier transform [Ell & Sangwine 2007] is applied. To extract phase information from the frequency domain, its entire magnitude is set a constant value. In this case the value one was chosen but any non zero value would suffice. The image in frequency domain is then inverse transformed and the saliency map generated. An

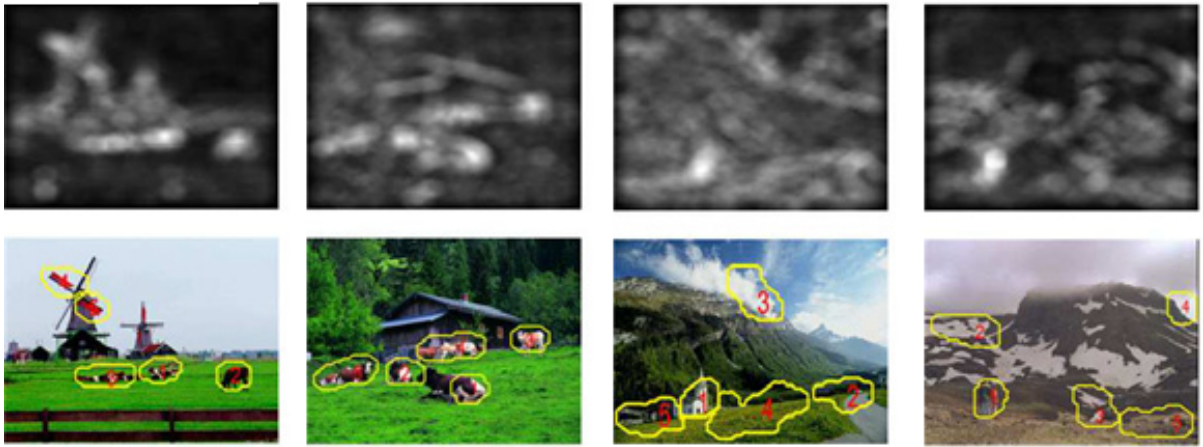


Figure 2.3: Examples of saliency map generated by quaternion fourier transform. Image adapted from [Guo *et al.* 2008].

example of the generated saliency map can be seen in Figure 2.3

The main advantage of using a quaternion transform is being able to compute all four channel in a parallel manner. Conversely, a common approach would be to calculate four distinct fourier transforms, one for each channel, substantially decreasing runtime speed.

2.1.4 Learning-based methods

Saliency detection can be generated using two distinct approaches: bottom-up and top-down. The former represents on information obtained from an image, while the latter require previous training and calibration for proper detection of objects. Learning-based methods are, by definition, a top-down approach.

Using a dataset of eye-tracking data, Judd *et al.* (2009) extracts several features in order to predict where humans will look [Judd *et al.* 2009], this prediction is compared to the groundtruth generated as shown in Fig. 2.4. To properly model human focus of attention in images, a broad range of features was selected. These are:

1. **low level features using local energy of steerable pyramid filters** [Simoncelli & Freeman 1995], which were shown to correlate with visual attention;
2. **mid-level gist features [Oliva & Torralba 2001] capable of detecting the line of the horizon**, this is useful because objects are normally found on earth's surface, therefore the horizon is a place where humans normally look for objects;

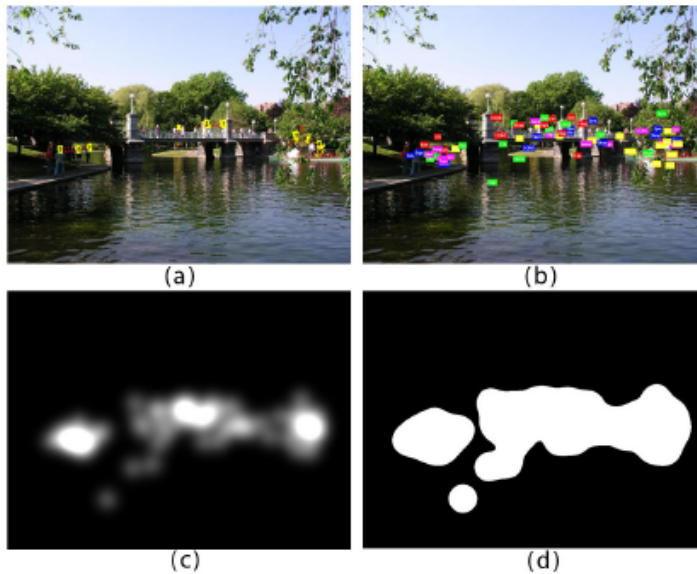


Figure 2.4: The original image (a), participants gaze is then inspected for the image and forms the groundtruth (b); the image (c) represents the saliency map generated by using a gaussian filter over the fixation of the participants. Finally, the (d) image represents its 20% most salient parts. Image from [Judd *et al.* 2009].

3. **high-level features using a face and a full body person detector**, based on the fact that human attention is easily fixated to other persons and faces
4. **Center bias**, photographs are normally taken with the object of interest in the center, as such, salient regions closer to the center will generate higher saliency.

Similarly, Liu et al. (2011) combines several features using a Conditional Random Field (CRF). One of the features is the center-surround, which is similar in purpose to the one described in Section 2.1.1, although implemented differently. This center-surround feature compares a region and its surroundings using χ^2 between RGB color histograms with

$$\chi^2(R, R_s) = \frac{1}{2} \sum_i \frac{(H_i(R) - H_i(R_s))^2}{H_i(R) + H_i(R_s)}, \quad (2.10)$$

where R represents the center region rectangle, R_s the surrounding region rectangle, and $H_i(R)$ the i th bin of the histogram of $H(R)$. This comparison of regions attempt to capture the natural local contrast of an object in relation to their surroundings.

Another feature is based on multiple scale contrast. Focusing on detection of strong pixel-level contrast in relation to a 9x9 neighborhood:

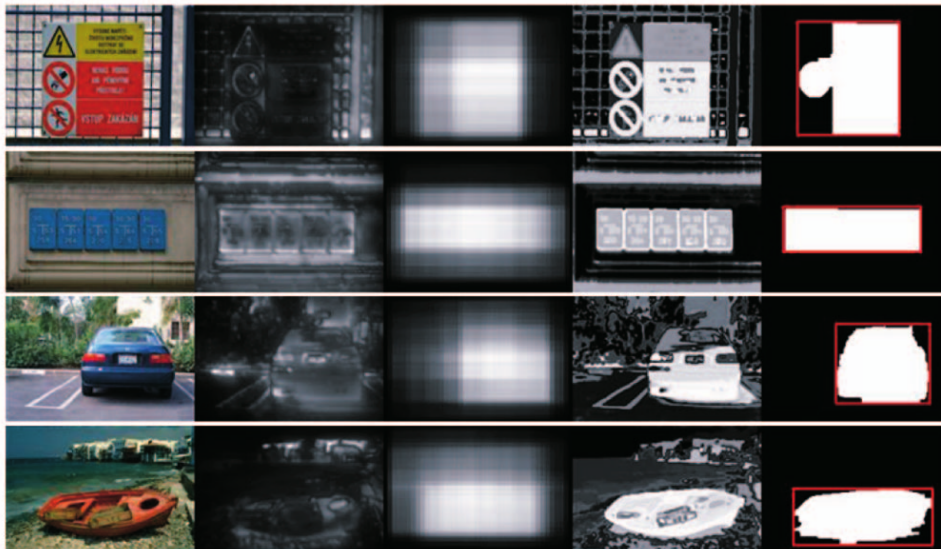


Figure 2.5: For each column, from left to right: the input image, multi scale contrast, center-surround, spatial color distribution and the final (binary) saliency map generated by the CRF. Image extracted from [Liu *et al.* 2011].

$$f_c(x, I) = \sum_{l=1}^L \sum_{x' \in N(x)} \|I^l(x) - I^l(x')\|^2, \quad (2.11)$$

where I^l is the l th image scale, L was empirically defined as six, and $N(x)$ is defined as a 9×9 window. This contrast feature gives strong responses on object borders and penalizes homogeneous regions.

The last feature uses the color rarity principle, similar to [Cheng *et al.* 2011a] (described in Section 2.1.2), where less frequent colors are more likely to contain salient regions [Liu *et al.* 2011]. An overview of each feature extraction is presented in Fig. 2.5.

2.2 Object search

Localization of an object within an image requires the use of search methods to define which regions an object detector can be applied. The output of a search procedure varies according to the method used, and ranges from a set of pixels, rectangles, contours or object centroids.

Among existing search methods, sliding window approaches are the most used on high-performance recognition systems [Divvala *et al.* 2009]. The method works by sliding a fixed-size window over the image, evaluating a quality function $g(\cdot)$ at each unique window

position. This quality function is commonly chosen to be one of the many available classifiers, some of which are described in 2.3.2. Given a quality value, $g(R)$, for each rectangle R , the actual object location is found, as shown in [Lampert *et al.* 2008], using

$$R_o = \operatorname{argmax}_{R \subset I} g(R), \quad (2.12)$$

where R is the subset of fixed size rectangles within an image. The use of a fixed-size window restricts the size of the object and the aspect-ratio, limiting the search scope. To avoid limiting object size, the image can be repeatedly downscaled by a constant ratio and the search done on each different image size. Therefore, as the image becomes smaller, the fixed-size window is able to wrap bigger objects.

A limitation of Eq. 2.12 is the implicit assumption that there is only one object of interest. In case the presence of multiple objects is possible, non-max suppression (NMS) can group several rectangles that are similar in size and position into a single one. Thus, after NMS, the remaining rectangles are considered actual object detections.

Although the sliding is a simple and effective method for finding objects, it has some important drawbacks. One of the most important disadvantages is the high number of rectangles evaluated by the quality function, this number can be calculated using

$$N_R = \left(1 + \frac{w_i - w_w}{s_h}\right) \cdot \left(1 + \frac{h_i - h_w}{s_v}\right), \quad (2.13)$$

where w_i and h_i are the image width and height, respectively, w_w and h_w are the window width and height, s_h is the horizontal stride and s_v is the vertical stride. The stride represents how many pixels the window will be displaced after to evaluation of $g(R)$. Thus, if evaluation of $g(R)$ is costly, processing a large image at multiple sizes may become prohibitive. An option to reduce total time is to increase the stride s_h and s_v reducing the number of windows to be evaluated. However, this change in stride may sacrifice localization accuracy.

Several attempts have been made reduce either the image search space or time required to evaluate the quality function. On the remaining of this section we will concentrate on describing contextual methods, branch and bound approaches and saliency-based approaches that aim to tackle that issue.

2.2.1 Contextual methods

Context is broadly defined as any information in the scene that influences the way objects are perceived [Strat 1993]. Context methods use information pertaining objects, their relations to other objects and the scene.

One might use context to discriminate between two similar objects according to characteristics of the background, for example, to help a detector with the choice between a computer monitor or a television as detection result of an object. Another use is to limit search to regions that are more likely to contain the object of interest.

A context detector, capable of finding regions likely to contain bicycles, cars and pedestrian, was presented in [Wolf & Bileschi 2006]. The architecture is similar to a rejection cascade, where the context detector is applied to a region and tested against a threshold value. In case a region score is above the threshold an appearance detector is applied.

The context features of [Wolf & Bileschi 2006] are based on several layers of contextual information, and can be divided in three broad categories: color, texture and position types. Color is represented in CIE LAB space, which is based on a non-linear compression of CIE XYZ and intended to model the way human perceive color variation. The texture layers capture local information about brightness gradient. While both texture and color layers are based on previous work of Belongie et al. (1998), the position layer was added to calculate the distance of a given pixel to a set of fixed points, allowing even linear classifiers to detect whether a feature is far or close from the center.

Photographers commonly put an object of interest in the center of the photograph. This is a type of cultural context [Divvala *et al.* 2009]. Algorithms that depend on such context, as the aforementioned [Wolf & Bileschi 2006], are sensitive to tilted, rotated or non-biased images. To avoid such restriction, [Perko & Leonardis 2007] enumerate several context sources and do not rely on object's position in a scene. That implementation is based on geometric features that segment an image into three groups: sky, vertical objects (buildings, etc.) and ground. Texture features are based in an image representation called blobworld, created in [Belongie *et al.* 1998]. A novel context feature is also included, based on a horizon estimate.

The algorithm from Perko et al. (2007) learns context features of objects by analyzing labeled images. Each positive and negative training sample context features are extracted over five radii and twelve orientations around the object, as shown in Fig. 2.6. These context features are used to feed a Support Vector Machine (SVM) to detect regions that are more correlated with the presence of the object of interest.

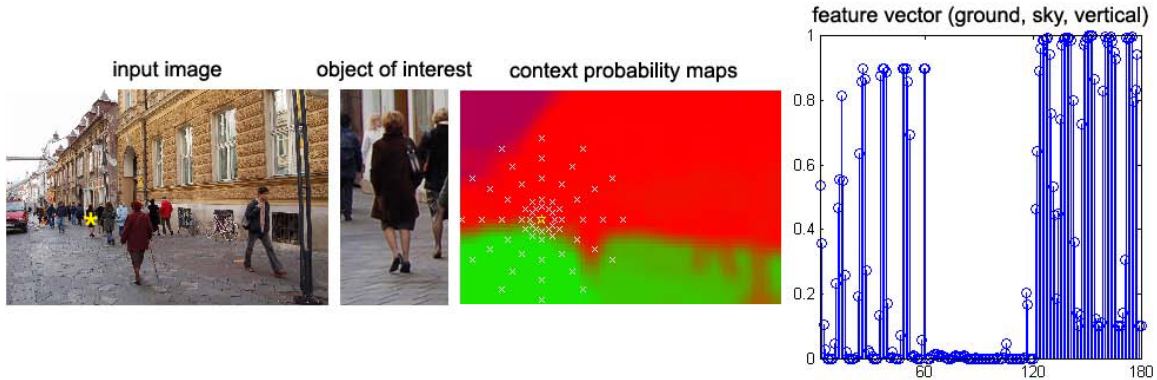


Figure 2.6: Example of feature extraction by [Perko & Leonardis 2007]. From left to right: input image with a yellow star over the object of interest, image cropped over object, features extracted over five radii and twelve orientations, resulting feature vectors. Image from [Perko & Leonardis 2007].

2.2.2 Branch-and-bound methods

In most object detection applications a large part of the input space is irrelevant. As such, discarding multiple regions, without evaluating the quality function, can yield better runtime performance. The branch-and-bound (BB) technique attempts to maximize a function $q(x)$, where x denotes the elements of the search space X .

A branching function divides X into n subsets X_1, X_2, \dots, X_n where $X_1 \cup X_2 \cup \dots \cup X_n = X$. Then, each subset has its maximum and minimum values calculated using a bounding function $\hat{q}(\cdot)$. When a subset has an upper bound smaller than the lower bound of any other subset, it can be safely removed (pruned) from the search space. While the maximum for $q(x)$ is not found, the branching procedure is repeated recursively and new bounds evaluated.

Recent approaches have adapted BB techniques to object search in machine vision applications. One such algorithm is the Efficient Sub-window Search (ESS), described in [Lampert *et al.* 2008]. In ESS the search space is the entire set of possible rectangles \mathfrak{R} in an image. In this model, the branching function splits the rectangle space and the bounding function $\hat{q}(\cdot)$ is required to meet two conditions:

$$\text{i. } \hat{q}(\mathfrak{R}) \geq \max_{\mathcal{R} \in \mathfrak{R}} q(\mathcal{R}) \quad (2.14)$$

$$\text{ii. } \hat{q}(\mathfrak{R}) = q(\mathcal{R}), \text{ if } \mathcal{R} \text{ is the only element in } \mathfrak{R} \quad (2.15)$$

where \mathcal{R} is the rectangles of a subset. Equation 2.14 guarantees that \hat{q} acts as a bounding function and Eq. 2.15 guarantees that the solution found is equivalent to an exhaustive

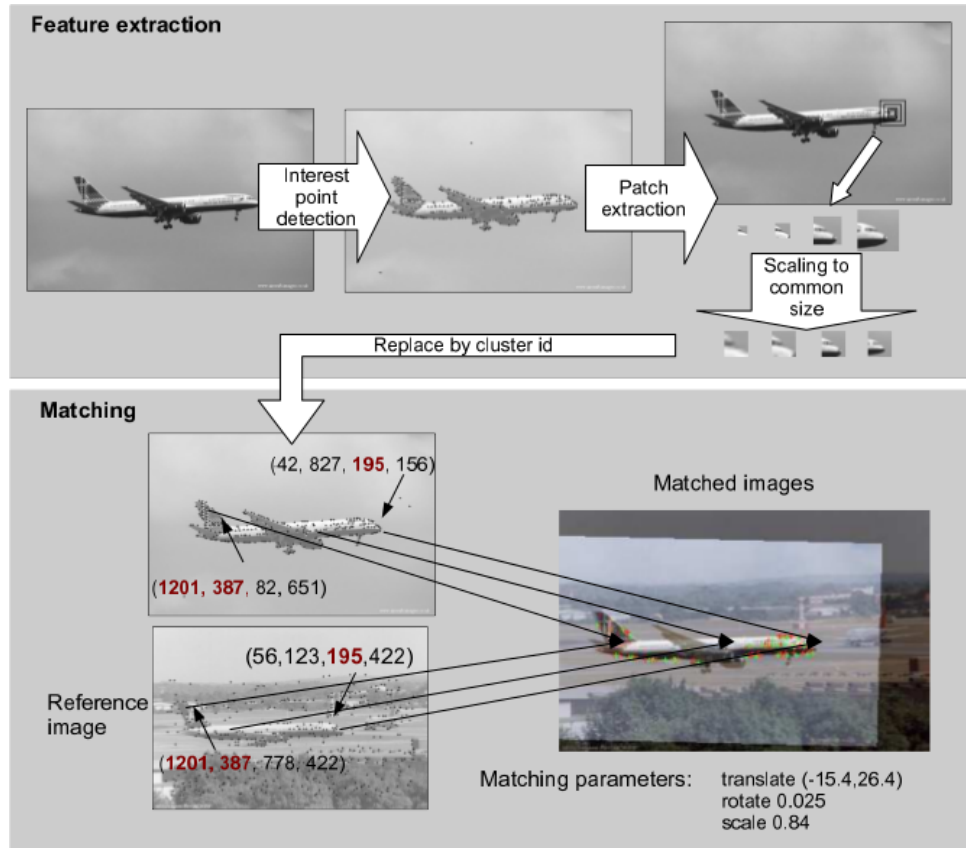


Figure 2.7: Matching a reference image against a dataset. Image taken from [Keysers *et al.* 2007]

search (evaluating the entire search space). Using this formulation, ESS average algorithm complexity is $O(n^2)$.

Another approach for faster object recognition in images uses optimal geometric matching [Keysers *et al.* 2007]. This method compares and matches two patch-based object representations even after rotation, translation or scaling. The matching between patches of the input image against a training dataset is done using BB optimization. This process is outlined in Fig. 2.7.

To match a target object in an image the first step is to perform an interest point detection, such as SIFT [Lowe 1999], followed by extraction of image patches from the interest points in multiple scales. The extracted patches are substituted by their closest patch clusters obtained from a training set, containing the object of interest. Finally, the algorithm searches for the optimal matching between the reference images and the training set, and decides which training image is the closest match to the target object using BB optimization to search for the patches in the input image.

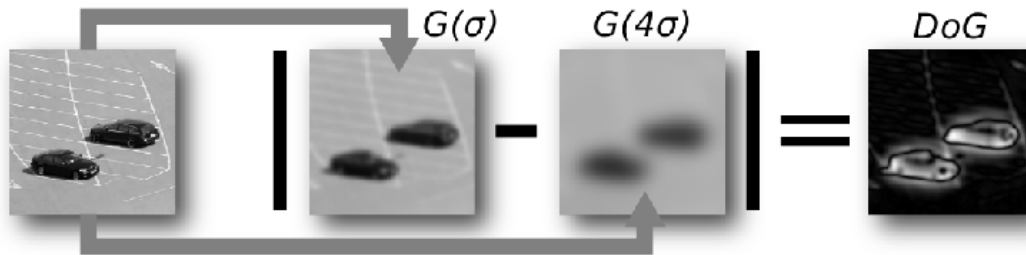


Figure 2.8: Detection a region of interest using Difference of Gaussians. Image from [Yiu & Varshney 2011]

2.2.3 Saliency-based methods

Focusing limited resources only on important regions of an image can reduce the time required to properly evaluate it. This way, saliency methods highlight regions that “stand out”. This information may allow one to reduce the actual search space during object detection by ignoring unlikely regions.

When navigating for huge giga-pixel images, the process of finding regions of interest can be tedious. This issue is tackled in [Yiu & Varshney 2011], where regions of interest are detect through analysis of saliency at multiple scales. To detect salient regions a Difference of Gaussians (DoG) is calculated (see Fig. 2.8) in a similar but simpler way than [Itti *et al.* 1998]. An important difference to [Itti *et al.* 1998] is that the saliency maps of each scale are not aggregated into a single saliency map, they are analyzed independently, which avoids mixing salient objects with different sizes: a parking lot with smaller objects within, like a person or car.

In a particular region of an image, even though a crack on the floor may be considered a salient region, the frequency of its appearance on the rest of the image may diminish its significance. Taking this into consideration, salient patches in [Yiu & Varshney 2011] are described using MPEG-7 color structure image descriptors, which counts colors frequencies using a 8×8 sliding block over each salient patch. Using this formulation, regions can be compared using Euclidean L_2 norm distance. From the description of each patch, a k -Nearest Neighbor (k -NN) is applied to detect unique regions and tease out locally salient regions that are too frequent in the scene. To achieve such effect, only the top 3% regions, from an average distance standpoint, are kept after k -NN.

In one of the test images, salient region detection selected over 18.000 regions, which were further reduced to 525 after k -NN. Manual intervention was required to change the

number of selected regions down to only 64. This selection of regions of interest enables a human observer to quickly focus on important regions, and it also allows automatic image navigation to important parts of the image.

Another approach to reduce the image space is based on finding the most unique regions on the image. This insight led [Feng *et al.* 2011] to develop a saliency method that tries to compose a region with others in the image. Thus, the harder it is to compose a region the more salient it becomes.

To find unique regions, pixels are first segmented into super-pixels using [Felzenszwalb & Huttenlocher 2004]. These regions are compared using a spatial and an appearance distance. Let p and q be any two segmented regions, the spatial distance $d_s(p, q)$ is calculated using a Hausdorff distance while appearance distance $d_a(p, q)$ uses histogram intersection of LAB space color histograms, given as

$$H(I) \cap H(I') = \sum_{j=1}^n \min(H_j(I), H_j(I')), \quad (2.16)$$

where I and I' are images, with $H(I)$ and $H(I')$ as their respective histograms. From these distances, the cost, $c(p, q)$, of composing a region with another is defined as

$$c(p, q) = [1 - d_s(p, q)] \cdot d_a(p, q) + d_s(p, q) \cdot d_a^{\max}, \quad (2.17)$$

where d_a^{\max} is the biggest value for appearance distance in the input image.

With the cost of composition defined for each region pair, a sliding window is applied over the image at multiple scales. For each window position, a greedy optimization is applied to select the best regions to compose the area within the window. Additionally, other cues are also applied, such as giving preference to objects closer to the center. Thus, in [Feng *et al.* 2011], the saliency value for a window is its composition score.

2.3 Object detectors

Object detectors are responsible for, given an input image, detection and localization of the object of interest. Object detectors are often composed of two main phases: feature extraction and classification.

2.3.1 Feature extraction

Feature extraction aims to reduce image space into more meaningful elements than, for example, color intensities. This is done through detection of distinctive properties of objects, such as: color, texture and shape.

Color information can normally be obtained directly from the image representation. Different images formats can be used to represent an image color distribution, such as RGB, HSV and LAB.

Filtering certain color types is important in case objects can be identified through a particular color. Another purpose is reducing the search space of an object detector that has a recurring color. An example of such case is face filtering, where [Peer *et al.* 2003] developed a set of rules to define the most common colors within a face:

$$R > 95 , G < 40 , B > 20 \quad (2.18)$$

$$\max\{R, G, B\} - \min\{R, G, B\} > 15 \quad (2.19)$$

$$|R - G| > 15 , R > G , R > B \quad (2.20)$$

According to [Vezhnevets *et al.* 2003], approaches relying on a trial and error procedure to define a specific set of filter rules is a disadvantage of methods such as of Peer *et al.* (2003). Modern approaches tend to rely on learning methods to automatically create color filters, such as in [Gomez & Morales 2002].

Although color is an important information of most objects, some objects may be better defined by recurring patterns within their area. A texture has repeating elements, where each individual element may be defined by a single pixel, a group of pixels, fractals and many other types [Ballard & Brown 1982].

A widely used texture extractor is the Gray Level Co-Occurrence Matrix or GLCM. It works by establishing a relation between every pixel in the image and its neighbor. Given a reference pixel, its neighbor is commonly defined as its right pixel, but other ways are also possible. Basically, a GLCM finds how well a certain reference and neighbor values correlate with each other [Hall-Beyer 2007]. This information can be used to calculate information that describe some properties of a texture, such as: contrast, correlation, energy and homogeneity.

In an attempt to capture shape information of an object, Haar-like features were initially proposed in [Papageorgiou *et al.* 1998] and extended in [Viola & Jones 2001]. These features attempt to infer the shape of an object through simple gradient calculation, as

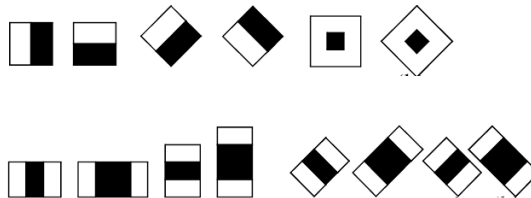


Figure 2.9: Some types of Haar-like features. These features are composed of two groups (black and white). The sum of the color intensities is calculated in each group and subtracted to calculate the gradient between the regions [Lienhart & Maydt 2002].

presented in Fig. 2.9.

Haar-like features can be calculated very fast using the integral image representation [Viola & Jones 2001]. This representation allows one to calculate the mean of a given rectangular region of interest in constant time. This is so as each point in an integral image is equivalent to the sum of all values from the left and above, as

$$I'(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y'). \quad (2.21)$$

Using the formulation presented above, a given rectangular sum can be calculated from an integral image using only four array references

$$s(x, y, w, h) = I'(x - 1, y - 1) - I'(x - 1, y + h) - I'(x + w, y - 1) + I'(x + w, y + h), \quad (2.22)$$

where w and h are the rectangle width and height, respectively. It is worth noting that the integral image has the same size as the original plus one at each dimension.

Although integral images [Viola & Jones 2001] allow fast Haar-like feature calculation, these features are not suitable for all situations. In particular, its ability to detect shapes is reduced when dealing with objects in complicated backgrounds and with strong illumination changes [Zhu *et al.* 2006]. Histogram of Oriented Gradients [Dalal & Triggs 2005] (HOG) achieve overall better performance. HOG is based on the assumption that much of an object appearance can be described by the distribution of local intensity gradients. This is achieved through small spatial cells which accumulate one-dimensional histogram of gradient directions. To achieve better invariance to illumination, the cells are contrast-normalized over a block, a larger spatial region, as illustrated in 2.10. For more detailed information on the very choice of each parameter please refer to [Dalal 2006].

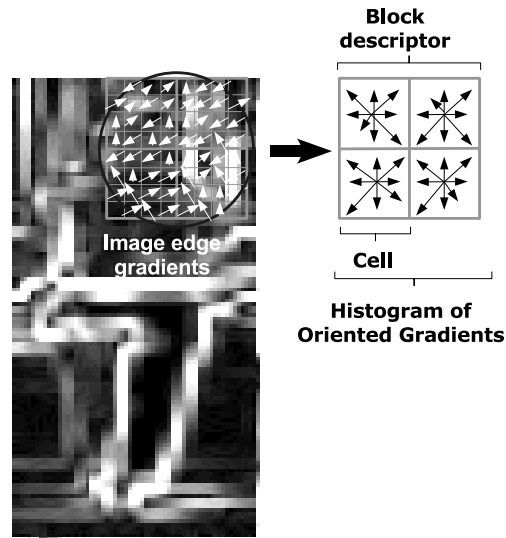


Figure 2.10: A histogram of oriented gradients gradient values are accumulated over blocks, cells and bins. Image taken from [Oliveira 2010].

2.3.2 Classification

Among the classifiers, decision trees, which are detailed on [Bradski & Kaehler 2008], are a common choice. In this method, the decision itself is represented by a tree-like data structure. Each node in that tree represents a decision, which is dependent on a feature value, for instance $f_i > 0.3$ where f_i is a specific feature i value. In case the condition is true, a branch is taken, otherwise, another one is chosen instead. Every branch changes the underlying probability that a given sample belongs to the class of interest or not, and this proceeds until the error margin is within the specified limits. The main advantages of decisions trees are its simplicity and natural interpretation of a trained tree. One of its limitations is the predisposition to overfit.

In contrast to decision trees, boosting techniques are based on combining several intermediate predictions, which are guaranteed to be only better than chance, to generate an accurate result [Freund *et al.* 1999]. Classifiers that are only required to be slightly better than chance are called weak classifiers.

An example of boosting is adaptive boosting (AdaBoost) [Freund & Schapire 1996], where each new classifier in the chain concentrate on samples that are being incorrectly classified. A common weak classifier for AdaBoost are Decision Trees, in this case however, each tree is made to only have few branches. From the result of each weak classifier, Adaboost combines them and decides whether a sample is positive (1) or negative (0) using

$$D(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}, \quad (2.23)$$

where α_t is $\log \frac{1}{\beta_t}$ and h_t is a weak classifier receiving only one feature, $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ where ε is the error, defined by

$$\varepsilon_t = \sum_i w_i |h_j(x_i) - y_i|, \quad (2.24)$$

where w_i is the weight of weak classifier i .

Adaboost was used with Haar-like features and integral images in [Viola & Jones 2001], to create a fast face detector. This fast classification was achieved through a rejection cascade, where the classification happens in several layers. The initial layers discard several samples using only a small number of features, achieving fast rejection. Each subsequent layer increases the number of features used, allowing detection of more negative samples at the cost of additional runtime. In this structure, only the last layers label an object as an actual positive sample.

The good classification results achieved with Adaboost, as seen in [Viola & Jones 2001, Lienhart & Maydt 2002], are related to its ability to find linear classifiers that separate high dimensional data [Freund *et al.* 1999]. Similarly, Support Vector Machines (SVM) also have this characteristic, however, it is achieved in a different way. To deal with high dimensional data, a SVM makes use of a Kernel, which allow for low dimensional calculations in a way equivalent to an inner product in a higher dimensional space [Freund *et al.* 1999].

The SVM was developed by [Vapnik 1999]. A short definition is given in [Araújo *et al.* 2008]: a binary classifier that embodies Structural Risk Minimization (SRM), Vapnik-Chervonenski (VC) dimension theory and Optimization Theory.

One of the most important concepts behind the SVM is the kernel mapping function. A kernel maps the input features to a higher dimensional space. However, a kernel function is required to satisfy Mercer's Condition, which imposes that the kernel matrix used to define the kernel mapping function has to be positive semidefinite [Araújo *et al.* 2008].

Among the many kernel types available, the SVMlight [Joachims 1999] implementation provides four kernel types: linear, polynomial, radial basis function and sigmoid. The specific choice of kernel type can made based on prior knowledge about the types of invariance on input features. In case no prior knowledge exists, the best match can be found by cross-validation.

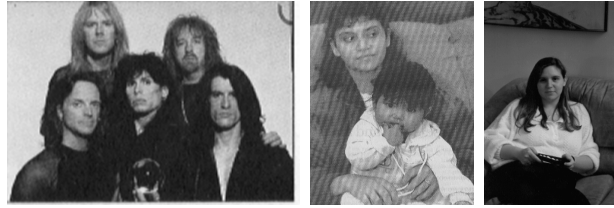


Figure 2.11: Images from MIT+CMU dataset [Schneiderman & Kanade 1998, Schneiderman & Kanade 2000]



Figure 2.12: Images from INRIA dataset [Dalal & Triggs 2005]

2.4 Datasets

Image datasets can be defined as a collection of related images organized in a common structure. Most datasets are built around an object or task, for example, person detection. There are several different ways to organize image datasets, but they are normally composed of three folders: training, validation and testing. The training folder is commonly created with fixed-size images and populated with positive and negative samples, and it can be directly used to train an object classifier. Similarly, the validation folder is composed with positive and negative samples of fixed-size images. However, these samples are used to reduce *overfitting* through methods such as cross-validation. In contrast, the testing folder can be composed either of entire scenes or be cropped around the object of interest (and is used to test the trained object detector performance in a distinct set).

With a common structure, existing datasets provide an easier alternative in comparison to manual collection of images from unstructured sources such as Flickr¹. Another advantage of using existing datasets is that comparison of results becomes easier, as not using the same images may bias the results towards one dataset or another. This is highlighted on Figs 2.11 and 2.12, where one dataset uses gray-level images have a mostly homogeneous background and the other has colored images and a very cluttered environment.

¹Available at www.flickr.com



Figure 2.13: Images and the annotated objects from Pascal Voc Website [Everingham *et al.* 2012]

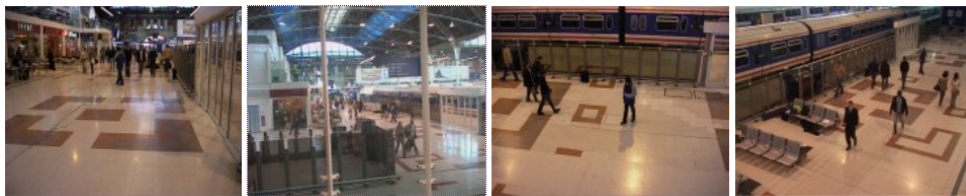


Figure 2.14: Example images of the PETS2006 dataset extracted from [PETS 2006]

Several competitions are based on public datasets. One of such is the Pascal Visual Object Classes (VOC) Challenge [Everingham *et al.* 2012]. The Pascal VOC is done every year and currently has 20 object classes, 11,530 images and 27,450 annotated objects. The images themselves have different levels of clutter and contain several environments. A sample of the images and their annotations are presented on Fig. 2.13.

Pascal VOC focuses mostly on general object detection and segmentation. Other datasets tackle different problems. One such example is [PETS 2006] that focus on detecting unattended luggage, as shown in Fig. 2.14. In this case, the scenario is mostly constant and several frames of surveillance cameras are annotated and made available for research purposes.

In order to make dataset creation easier, recent works have focused on building datasets dynamically. With the advent of image search tools (e.g., Google Image), automatic

generation of datasets has become easier. One such work searches a given human pose by keyword, and through incremental preprocessing and filtering selects only a subset of the returned images [Ikizler-Cinbis *et al.* 2010]. In order to detect recurring patterns in the extremely varied results, their method selects the twenty most relevant images returned, and uses them to train a logistic regression classifier. This classifier is used to measure what parts of the foreground they have in common. Another approach for web search was proposed by [Schroff *et al.* 2007], who not only uses the image visual properties but also textual information returned from the query.

Instead of searching through textual queries, [Ferecatu & Geman 2007] propose to capture the abstract knowledge of the user in order to iteratively find the object category of interest. The user is presented with a set of general categories, from those he chooses one with a closer match to the desired one. The proposed method, based on the chosen image, computes the next categories until the desired one is found. This solution is able to group images based on their general composition.

Dynamic generation of a dataset solves the problem of finding images of objects. However it cannot effectively provide object annotations without manual intervention. This issue was tackled on the LabelMe project [Russell *et al.* 2008], an online and dynamic database of annotated images. Image uploads and object annotations can be made on the fly and even crowdsourced [Sorokin & Forsyth 2008].

2.5 Relation to our work

Several techniques have been created to reduce the search space in an image. Some of these were described in Section 2.2. This can be useful, for example, to speed up object detection by reducing the number of times a detector has to be used. As a detector is comprised of image preprocessing, feature extraction and classification, such expensive operations should be done only on regions with greater probability of containing an object. Moreover, excluding regions may reduce risk of false positive detections.

Our work approaches the search space problem using a saliency-based approach, based on multi-scale spectral residue (MSR) analysis. This is achieved through an adaptation of the spectral residual method [Hou & Zhang 2007] in order to enable its use in multi-scale environments. From the use of saliency information in multiple scales, MSR attempts to discard unimportant regions before actual object detection. Moreover, MSR was intended to be fast from the ground up. This was meant to allow faster image processing for sliding-window based object detectors.

While similar in some aspects to MSR, other techniques have some significant differences. Although Feng et al. [Feng *et al.* 2011] assign a saliency score to each window, this score is calculated independently for each region, while in MSR the saliency itself is calculated only once for the entire image. Additionally, MSR relies on properties of objects on the frequency domain, while Feng et al. (2011) relies on spatial (location) and appearance properties of objects. Our approach is also distinct from Yiu et al. [Yiu & Varshney 2011], which relies on saliency detection based in local contrast and finding regions which are unlike others in the image.

In contrast to branch-and-bound techniques reviewed in Section 2.2.2, MSR does not require the use of linear classifiers or local image descriptors. [Keysers *et al.* 2007] directly rely on a bag of words representation, which greatly limits the scope of application, excluding sliding window detectors based on HOG or Haar-like features. This limitation is important since sliding window approaches are the most used search method in high performance object detectors [Divvala *et al.* 2009].

MSR uses a saliency method to detect possible regions of interest in an image without requiring previous learning from a training database, this is a bottom-up approach. While contextual methods, as [Wolf & Bileschi 2006], rely on a top-bottom approach, where each object class context is learned independently. Since MSR requires no training for each specific object class, it can be executed only once and its results may be used for each distinct object detector, saving computational time.

In summary, MSR is a novel method that was created to allow a trade-off between number of windows selected for detection, and the number miss detections. Our results show good selection performance within the range of 70-80% reduction on window evaluations, while improving or at least maintaining detector performance.

In the following chapter we provide the main differences between existing saliency methods. This will be used to justify our choice of saliency method used by MSR. Moreover, we also provide a more detailed analysis of our saliency method of choice, which is SR, including its main advantages and limitations.

Saliency analysis

Contents

| | | |
|-------|--|-----------|
| 3.1 | On the use of saliency detectors | 33 |
| 3.1.1 | Runtime speed | 34 |
| 3.1.2 | Saliency map | 34 |
| 3.1.3 | Selection of search scale | 36 |
| 3.1.4 | Saliency for faster detection | 37 |
| 3.2 | Spectral residue | 38 |
| 3.2.1 | Statistical properties of natural images | 38 |
| 3.2.2 | Exploring $1/f$ law | 40 |
| 3.2.3 | Known issues | 42 |
| 3.3 | Closure | 44 |

According to Rensink (2000), human visual system can be divided into two main phases:

- **The first phase** is the initial processing state, rapid and parallel over the input visual stimuli. This phase happens before focused attention starts. The result from this phase is low-level information about structures in the visual stimuli. These structures are called proto-objects, and describe regions that require additional visual processing.
- **In the second phase**, a set of detected proto-objects is observed in detail. This attention phase is task-oriented, serial and slow when compared to the first phase. Resulting from this attention phase, a detailed internal representation of these proto-objects with both temporal and spatial coherence [Rensink 2000].

Since the cognitive capabilities of the human brain are limited, focusing attention on a small number of possible salient regions (or proto-objects) can help reducing the visual information that has to be processed by the attention phase.

The concept of saliency can also be applied to digital images. In this case, saliency is normally associated to the concept of rarity, surprise, visual uniqueness or unpredictability [Cheng *et al.* 2011a]. There are many possible principles that can be used to find salient regions. These are described in at Section 2.1. To summarize, the ways to search for saliency are

1. Local contrast, as in [Itti *et al.* 1998, Harel *et al.* 2007].
2. Detection of regions with highest global contrast (rarity) [Feng *et al.* 2011, Zhai & Shah 2006].
3. Analysis of spectral information [Hou & Zhang 2007, Guo *et al.* 2008].
4. Learning from image examples [Judd *et al.* 2009, Liu *et al.* 2011].

Among the advantages of saliency detection in digital images, the ability to detect objects even when no prior information is available is one of the most important. Such bottom-up approach can find regions of interest in an image without relying in prior knowledge about specific objects. This flexibility enables the use of saliency information to solve a broad range problems.

In [Hou & Zhang 2008], thumbnails are generated by detecting the most salient region in images. Salient regions are found based on the global rarity of their color and texture information. Finally, an empirical trade-off is selected between information density and region size, generating the final cropped image (Fig. 3.1).

Detecting salient regions can also help bottom-up image segmentation, where saliency indicates regions in which objects are more likely to be found. In [Cheng *et al.* 2011a], global contrast is used to segment image objects. Likewise, [Goferman *et al.* 2011] also use saliency to segment image objects, but to achieve such result they rely instead in a combination of local contrast, global contrast and prior information about common salient objects, such as human faces.

Another application of saliency information is to reduce the object detector search space, which is the focus of our work. In this context, an actual object detector can be used solely at candidate regions, proto-objects, to find which correspond to the object of interest. This can help achieve faster image processing. Also, discarding regions can avoid some potential false positive detections. An example of work applying this concepts can



Figure 3.1: Example of thumbnails generated for images. Image taken from [Hou & Zhang 2008].

be found in [Feng *et al.* 2011], where only the most salient regions are selected for object detection.

The remaining of this chapter will evaluate different characteristics of saliency detection methods. These characteristics will outline practical differences between the methods and for which situations they are most suited.

3.1 On the use of saliency detectors

Methods used to detect important regions in an image extract information from local contrast, global rarity, spectral information and learned data. Some approaches explore a combination of low and high level information for better performance, such as [Goferman *et al.* 2011].

The concept used to detect salient objects can modify the generated saliency map characteristics. Some methods are capable of generating high resolution saliency maps while others have an excellent runtime speed. A correct choice of saliency method will depend on the application domain.

In the following sections, we explore the main characteristics of each method regarding runtime speed, resolution of generated saliency map and search scale selection. To facilitate comparison we define the following shorthands for each method:

1. Itti's Method (IT) [Itti *et al.* 1998], see Section 2.1.1.
2. Graph-based (GB) [Harel *et al.* 2007], see Section 2.1.1.

| Method | IT | GB | SR | FT | AC | CA | LC | HC |
|----------|--------|--------|--------|-------|-------|--------|-------|-------|
| Time (s) | 0.611 | 1.614 | 0.064 | 0.016 | 0.109 | 53.1 | 0.018 | 0.019 |
| Code | Matlab | Matlab | Matlab | C++ | C++ | Matlab | C++ | C++ |

Table 3.1: Comparison of runtime speeds for saliency detection over Achanta et al. dataset [Achanta *et al.* 2009]. Results taken from [Cheng *et al.* 2011a], which used a dataset where most images have size around 400 by 300 pixels

3. Spectral Residual (SR) [Hou & Zhang 2007], see Section 2.1.3
4. Achanta’s Method (AC) [Achanta *et al.* 2008], see Section 2.1.2
5. Frequency-tuned saliency detection (FT) [Achanta *et al.* 2009], see Section 2.1.2.
6. Context-Aware (CA) [Goferman *et al.* 2011], Section 2.1.4
7. Luminance Contrast (LC) [Zhai & Shah 2006], see Section 2.1.2.
8. Histogram Contrast (HC) [Cheng *et al.* 2011a], see Section 2.1.2.
9. Phase-only Fourier Transform (PFT) [Guo *et al.* 2008], see Section 2.1.3

3.1.1 Runtime speed

Saliency detection methods are normally built to be fast. One motivation for such design is related to their use in unsupervised object detection.

Recent evaluations on runtime speed by [Cheng *et al.* 2011a], presented in Table 3.1, have shown subsecond results for most methods when processing (mostly) 400 by 300 images, except for CA. Were the fastest methods are, in order: FT, LC, HC and SR. Moreover, as PFT executes faster than SR, as seen in [Guo *et al.* 2008], it also can be considered one of the fastest methods.

In part, speed variations between methods can be attributed to differences in environment, where C++ implementations are, in general, faster. However, manual inspection of saliency approaches indicates that methods with worse runtime indeed seem to depend on more intensive algorithms.

3.1.2 Saliency map

Saliency detectors output a “saliency map”, which is an intensity image where brighter values indicates stronger saliency. Some methods, such as IT, GB, SR and PFT generate

| Method | IT | GB | SR | FT | AC | CA | LC | HC |
|------------|---------|--------|-------|-----|--------------------------|-----|-----|-----|
| Resolution | $S/256$ | $S/64$ | 64x64 | S | $S \cdot \frac{15}{100}$ | S | S | S |

Table 3.2: Resolution of generated saliency map when compared to the original image S . Part of the data is from [Achanta *et al.* 2009].

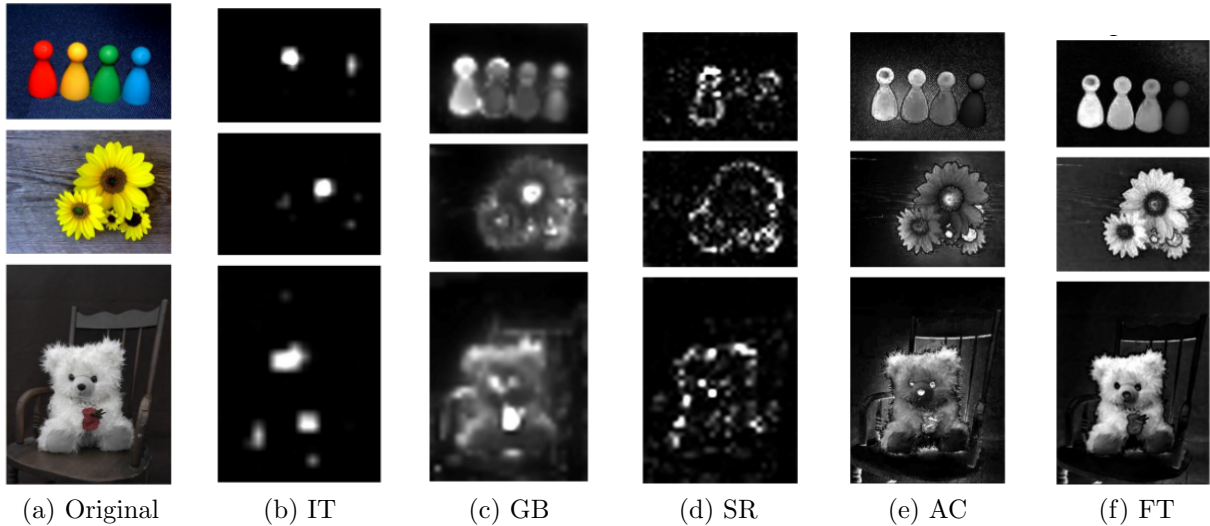


Figure 3.2: Some limitations of saliency methods analysed in [Achanta *et al.* 2009]. Limitations range from non-uniform object highlighting and highlighting only salient regions smaller than a certain filter size.

low resolution saliency maps, that is, intensity images smaller than the input image. Moreover, according to [Achanta *et al.* 2009], due to extreme downsizing some methods do not generate proper object boundaries on saliency maps, particularly GB and IT.

Methods that generate a high-resolution saliency map have, overall, obtained better performance in segmentation of images on tests found in [Cheng *et al.* 2011a] on the dataset created by Achanta *et al.*; examples of such good performing methods are HC, FT and AC. An overview of the generated saliency map resolution is summarized on table 3.2.

Another type of flaw in saliency maps is that some detectors cannot uniformly highlight the entire salient object, being SR an example. This flaw is caused by the limited range of spatial frequency that remain from the original image when computing the saliency map [Achanta *et al.* 2009]. Some differences in the generated saliency can be perceived over Fig. 3.2, like variations on saliency map resolution and also how much of the object is highlighted.

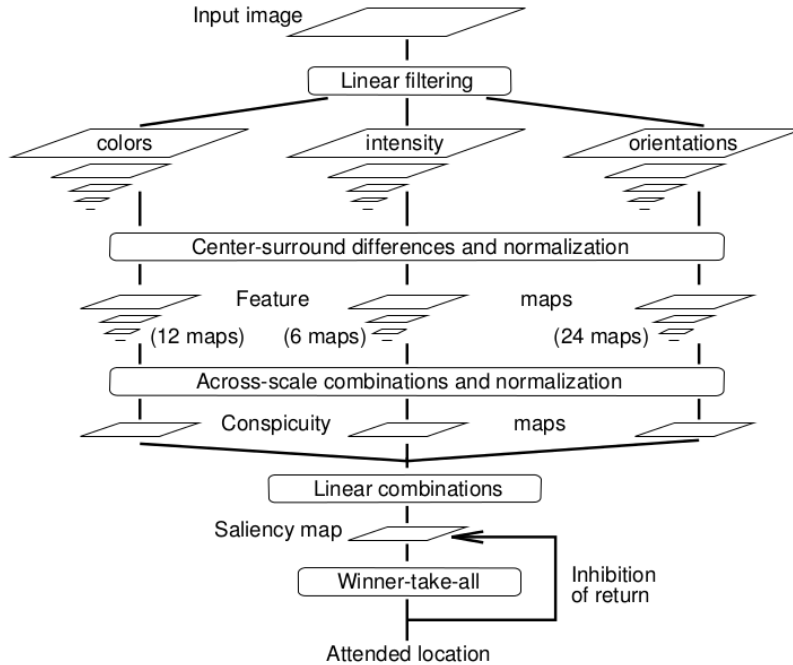


Figure 3.3: General architecture of IT. The input images are separated into color, orientation and intensity maps over several scales. The maps are combined using a normalization operator generating the final saliency map. Image taken from [Itti *et al.* 1998].

3.1.3 Selection of search scale

When searching for salient objects one might desire for only salient regions of a certain size. Fine-grained control over search scale can help attenuating regions with an undesired size from the saliency map.

Saliency detection using IT and GB combines information from multiple scales into a single saliency map using a normalization operator $N(\cdot)$. Particularly, the saliency map generation overall architecture for IT is shown in Fig. 3.3. However, some methods, such as [Rutishauser *et al.* 2004], calculate saliency with IT but avoid the normalization operator, processing each scale separately. This provides information about salient objects and their scale, avoiding potential loss of information.

In contrast to IT and GB, the SR method empirically defines a search scale based on common object sizes. For general-purpose object detection, this scale was found to be the best compromise between saliency region detection and error rates. The scale in SR is controlled through adjusting the input image size, as shown in Fig 3.4, where smaller images attributes more saliency to bigger objects. This characteristic is also shared with

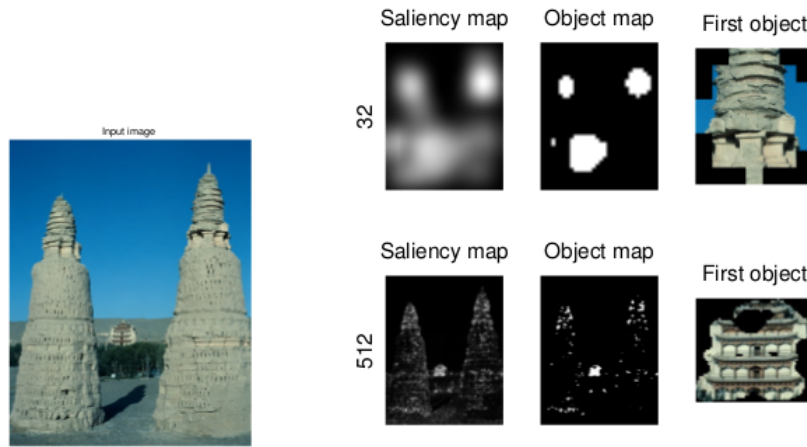


Figure 3.4: Example of scale control through image resizing in SR. In the larger images smaller objects are more salient. Conversely, as the image is downsampled bigger, objects start attracting more attention. Image taken from [Hou & Zhang 2007].

the PFT method.

Some other methods do not provide a direct way to control saliency search scale, such as FT, HC and LC. However, it is worth noting that, even in these cases, changing the image size affects the final saliency map, but how this correlates to changes in search scale is not clear. In contrast, CA combines information from multiple scales with scale-invariant information, one such information is the result from a face detector.

3.1.4 Saliency for faster detection

Our work aims at reducing a detector search space using saliency detection, in a certain way that some characteristics for saliency detectors are important:

1. Fast saliency detection, as calculating the saliency represents an additional processing overhead for object detection;
2. Scale selection is required as to search in the same scale as the object detector. This avoids polluting the saliency map with information from undesired scales;
3. Good saliency detection over cluttered images. Most saliency detection methods have been tested in datasets where there is a clear salient object in a non-cluttered background;
4. Acceptable resolution of saliency map. By generating a saliency map with too low resolution may affect search space reduction.

Runtime speed requirements are an important consideration. In case the saliency detector is too slow, it will not effectively reduce object detection time. This requirement greatly limits the use of IT, GB and AC method, where average detection times in a single image of 400 by 300 is greater than half a second, which is at least nine times worse than SR, FT, LC methods.

Concerning fine control over scale selection, both IT, GB, SR, CA can be modified to suit this purpose. Conversely, in methods like FT, LC and HC, scale selection seems to be harder to directly control. However, even in methods where scale selection is not explicitly controlled, changing input image sizes does change the generated saliency map in an unspecified way.

Performance metrics over non-cluttered images have to be analyzed in depth, as results may vary according to the dataset of choice and type of object. However, the HC method can be disregarded from further consideration in the scope of search space reduction, as it was described to perform better on images with a clear salient object and mostly uniform background.

The resolution of generated saliency map is less of a concern in our application. As high-resolution operations such as segmentation, which requires fine-grained edge information, are not used in our method.

Considering the requirements of speed, scale selection, robustness and resolution, the SR presents a good balance between the desired properties. In the next section, we provide an in-depth evaluation of SR method in order to understand its origin and characteristics.

3.2 Spectral residue

Approaches for salient region detection are commonly focused on properties common to objects. However, in spectral residual (SR) [Hou & Zhang 2007], properties of the background are explored instead. In contrast to most approaches for salient detection, SR is based on information extracted from the frequency domain.

In the following sections, we discuss some properties of natural images on frequency domain, the law used by SR and also its known limitations.

3.2.1 Statistical properties of natural images

Natural scenes are not random. They contain a particular structure and represent only a fraction of the actual image space [Ruderman & Bialek 1994]. Furthermore, in

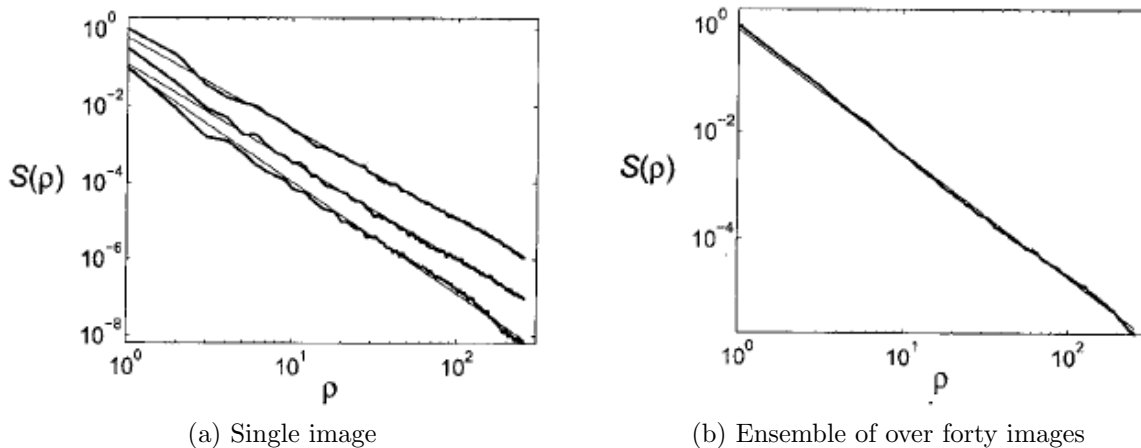


Figure 3.5: Power spectrum averaged over orientations (thick lines) and also their linear fits (thin lines). In these images p denotes log frequency and $S(p)$ the log amplitude over frequency. Images taken from [Hsiao & Millane 2005].

[Ruderman 1994] it is stated that natural images cannot be properly defined from any known elementary distribution commonly used in image models, as Gaussian, for example.

Understanding properties of natural images allows one to create compact image representations and to reduce effects of noise [Ruderman 1994]. One of the most widely known properties of ensemble of natural images is the power-law scaling. This property dictates that the power spectrum of a natural image ensemble (averaged over orientations), $E\{A(f)\}$, follows a distribution given as

$$E\{A(f)\} \propto 1/f, \quad (3.1)$$

where $E\{A(f)\}$ is the mean of amplitude over frequency f . Scale invariance properties such as that implies that image statistics do not change with different angular scales [Ruderman 1994]. Thus, independent of a camera focal length, the image properties will be retained as long as the image is multiplied by a proper constant value so that the modified image statistics are identical to the original image (self-affine). In practice, scale invariance indicates that objects in natural images are not more likely to appear in a particular size (angular scale), that is, there is no prevalent object size.

The formulation can be visually perceived when the amplitude spectrum from an ensemble of natural images is presented on a log-log scale. In this case, the amplitude spectrum, averaged over orientations, is approximately a straight line, as show in Fig. 3.5.

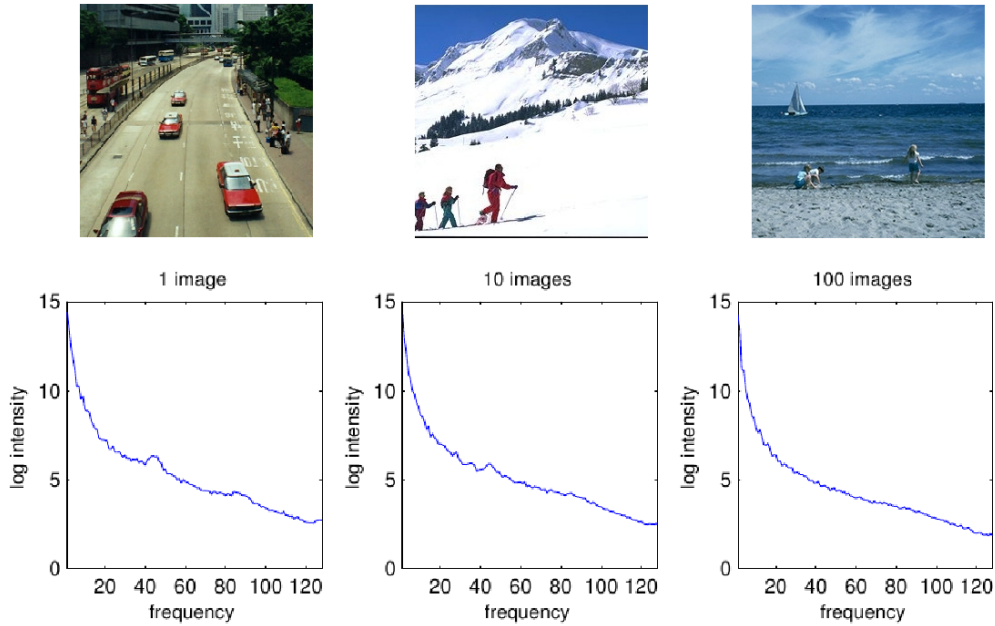


Figure 3.6: First row shows samples of the image ensemble. Second row shows differences in generated log spectrum representation generated with ensembles of different sizes. Image taken from [Hou & Zhang 2007].

3.2.2 Exploring $1/f$ law

The $1/f$ law applies only to an ensemble of natural images, demonstrated in Fig. 3.6, Hou and Zhang theorized that the statistical singularities found in the power spectra of individual images are linked to presence of proto-objects. In other words, regions that are more likely to contain objects are generally responsible for most of the perturbations over the smooth curve of the power spectra.

Commonly, an ensemble of image power spectrum is represented in log-log format, as described in Section 3.2.1. However, as SR works on individual images, the log spectrum format was used instead. The main reason for using log spectrum is that, for an individual image, the log-log spectrum is not well-proportioned along the frequency domain (too few samples on low frequencies), suffering from noise. The graphical differences between the log-log spectrum and the log spectrum are summarized in Fig. 3.7.

In order to detect the location of statistical singularities of an image, the concept of spectral residue was created. Spectral residue, depicted in Fig. 3.8, is calculated by subtracting a convoluted log spectrum from the original log spectrum, as

$$B = L(A) - h_n * L(A), \quad (3.2)$$

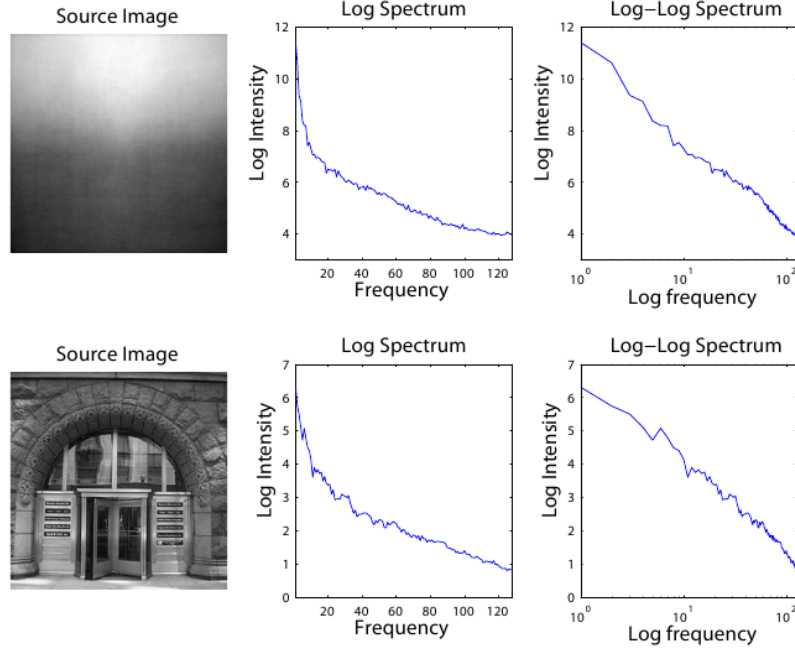


Figure 3.7: Comparison between log spectrum and log-log spectrum, where the first image is an average of 2277 natural scenes. Image taken from [Hou & Zhang 2007].

where $L(A)$ is the log amplitude of the Fourier domain and h_n is a 2D convolution filter of size $n \times n$. The convolution filter h_n is defined as

$$h_n = \frac{1}{n^2} \begin{Bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{Bmatrix} \quad (3.3)$$

This formulation intends to capture regions which strongly deviate from the expected $1/f$ distribution, or in other words, that jump out of the smooth log spectrum curves [Hou & Zhang 2007]. After calculation of the spectral residue, the saliency map, S_{SR} , is generated using

$$B = \log(A(f)) - h_n * \log(A(f)), \quad (3.4)$$

$$S_{SR}(x) = g(x) * \mathcal{F}^{-1}[\exp(i \cdot P(f) + B(f))]^2, \quad (3.5)$$

where \mathcal{F}^{-1} denotes inverse Fourier transform, $P(f)$ represents the phase information extracted from the frequency domain and $g(\cdot)$ is a 2D Gaussian filter.

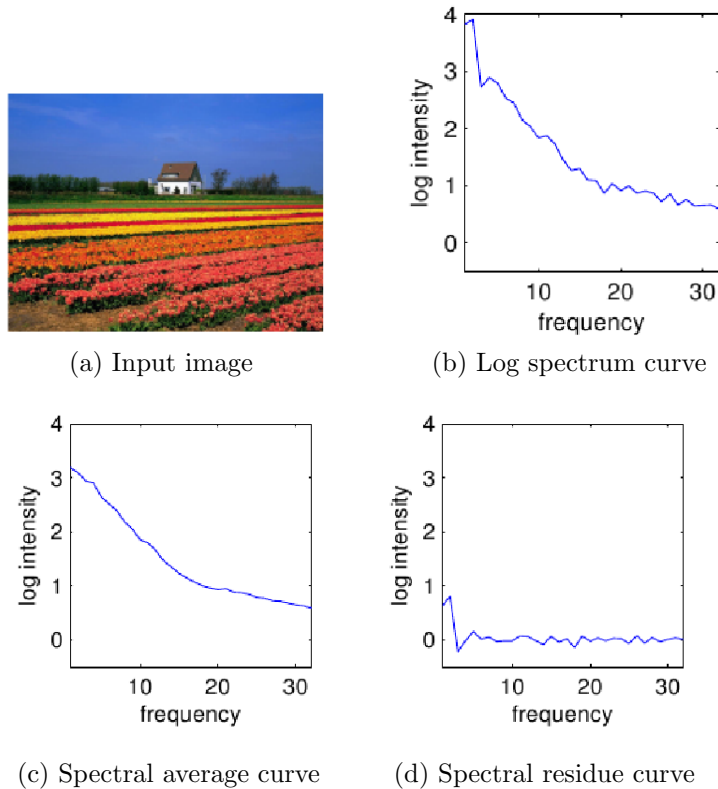


Figure 3.8: Calculation of spectral residue: (a) input image, (b) calculated log spectrum, (c) convoluted log spectrum, (d) spectral residue. The residue is obtained by subtracting the log spectrum from the residue. Image taken from [Hou & Zhang 2007].

From the generated saliency map, actual proto-objects are segmented using a per-pixel thresholding. In this case, a pixel is considered salient iff

$$O(x) = \begin{cases} 1 & \text{if } S_{\text{SR}}(x) > \text{threshold} \\ 0 & \text{if otherwise} \end{cases}, \quad (3.6)$$

where $S_{\text{SR}}(x)$ is a pixel of the saliency map. For SR, the threshold = $E(S(x)) * 3$, that is, three times the mean saliency intensity of the saliency map. Using these properties actual regions of interest are selected.

3.2.3 Known issues

Despite the advantages of the spectral residue, some works have demonstrated SR limitations. Among these limitations, the SR performance in general purpose saliency detection on images where there is a clear and distinctive salient object was among the worst when compared to other state-of-art methods. One such comparison was presented in

[Cheng *et al.* 2011a], where SR was the second worst. In [Achanta *et al.* 2009], the SR was also among the worst performing methods.

The most likely explanation for such low performance is scale selection. In SR the scale of search is controlled by the image size. As such, a constant image size is defined based on common salient object sizes over several images (as described in Section 3.1.3). This excludes objects that are bigger or smaller than the scale of search. This means that for general purpose saliency detection other methods are most likely a better choice.

Another issue is that, in contrast to what Hou and Zhang theorized, the most important information for SR saliency detection is the phase information extracted from the frequency domain, not the spectral residue. The importance of phase information was highlighted in [Oppenheim & Lim 1981], and in the context of saliency detection this was proven in [Guo *et al.* 2008], where a comparison was made between SR and PFT, where the latter is similar to phase-only filtering (described in [Horner & Gianino 1984, Chen *et al.* 1994]), obtaining similar results to the original SR. Phase information can be extracted by setting frequency domain amplitude information to a constant non-zero value [Guo *et al.* 2008]. In this way, saliency can be calculated with

$$\mathcal{D}(f) = \mathcal{F}(I(x)) \quad (3.7)$$

$$\mathcal{P}(f) = P(\mathcal{D}(f)) \quad (3.8)$$

$$S_{\text{PFT}}(x) = g(x) * \mathcal{F}^{-1}[\exp(i \cdot \mathcal{P}(f))]^2 \quad (3.9)$$

where \mathcal{F} denotes the Fourier transform, $g(\cdot)$ is a 2D Gaussian filter and $\mathcal{P}(f)$ represents the phase information of the frequency domain. In most cases, SR can be replaced with a PFT with similar results. This procedure does not significantly affect saliency detection performance in an observable manner, as shown in Fig. 3.9. Although PFT and SR are almost completely interchangeable, our work will build upon SR as it is a more widely used method. Furthermore, it is the method used for saliency method benchmark in many recent works, such as [Achanta *et al.* 2009, Cheng *et al.* 2011a].

Saliency calculation in SR and PFT is always done with gray-scale images, which discards color information that could be used to improve saliency detection. Furthermore, the lack of support for motion information can reduce performance in video frames, when saliency can be originated from an object motion and not its inherent characteristics (shape, color, edges). Some saliency methods have been built to capture motion

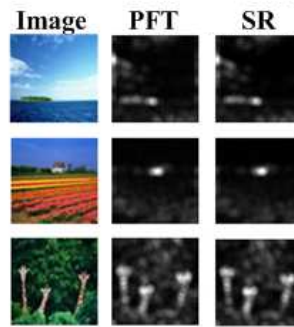


Figure 3.9: Comparison of PFT with SR. Image taken from [Guo *et al.* 2008].

information and combine it with bottom-up image information for detection of important regions, as [Zhai & Shah 2006] and [Guo *et al.* 2008].

3.3 Closure

Saliency methods relies on different techniques and have different application domains. The main differences between the methods can be divided into four categories: runtime speed, saliency map resolution, search scale selection and intended use.

Considering the task of faster object detection, we have chosen to use spectral residue analysis in order to speed up image object detection. This choice was made based on the possibility of adding fine control of scale selection, acceptable runtime speed and saliency map resolution. The original formulation of SR detects salient objects over a single search scale, which was chosen based on common object sizes. This default scale is not suitable for window selection because an object detector have to search for objects over multiple scales. To this end, in following chapters, SR will be modified to allow proper scale selection.

To show that SR was indeed the best choice for the task of window selection, comparison against other saliency methods is provided during result analysis of our solution. A throughout evaluation of SR will be presented in following chapters, in order to evaluate the detector behavior when associated with the use of this saliency method. Even though some SR limitations could negatively affect performance, such as the use of gray-scale images, we found that none of them represent a hard barrier for use of SR in the context of faster object detection.

Our structure, which we will present in Chapter 4, is called multi-scale spectral residue (MSR) and uses saliency saliency detection over multiple scales to select which windows will be processed by a full-fledged object detector. This modification required changing

SR from using a fixed image resize, to one that depends on the current detector search scale. This structure is described in details over the following chapter.

Multi-scale spectral residual object search

Contents

| | | |
|-----|---|----|
| 4.1 | Selecting regions | 49 |
| 4.2 | Saliency over multiple scales | 50 |
| 4.3 | Quality function threshold | 52 |
| 4.4 | β and k values | 53 |
| 4.5 | Image pre-processing | 55 |
| 4.6 | Closure | 59 |

When searching for objects in images using a sliding window approach, each unique window position is evaluated by the object detector. However, an object of interest is normally present in only a small fraction of the input space. Thus, an expensive object detection is applied disregarding the probability of a region containing the object of interest.

Saliency detection methods usually associate a measure of global or local importance for a image region. Information about a region importance can be used to choose whether a object detection will be applied in a given window. Discarding windows before detection can reduce the time required to process an image. Furthermore, discarding detection windows based on saliency information can avoid false positive detections. On the other hand, this approach may inadvertently discard windows that contain the object of interest, generating false negatives. MSR aims to harness the advantages of window selection, while avoiding most of its pitfalls. An overview of MSR structure can be seen in Fig. 4.1.

Among the possible salient methods available, two requirements are important for use in the context of region selection: speed and robustness. In this case, speed relates to how

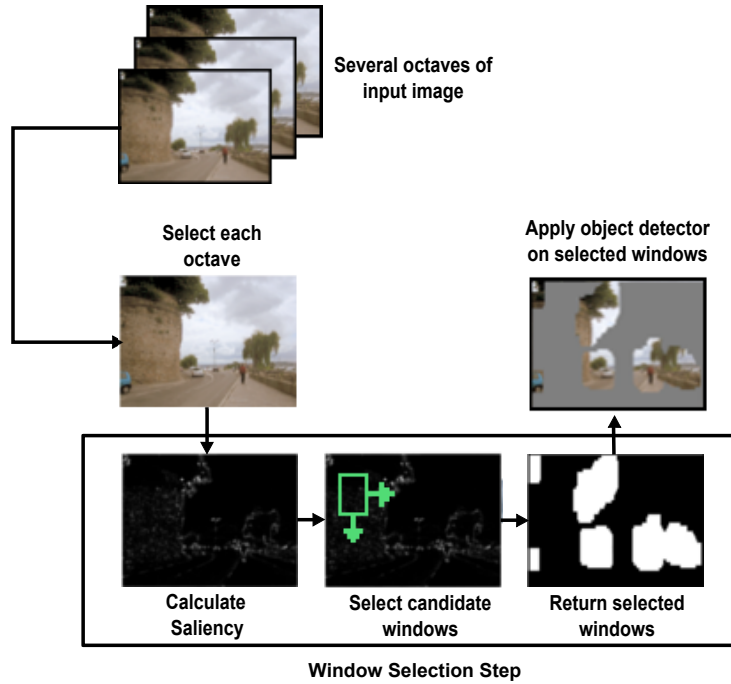


Figure 4.1: Overview of MSR structure for window selection.

fast saliency is calculated, and robustness to how well the saliency can discard regions without missing objects. To reach the best possible results for those requirements, we were motivated by the idea of spectral residue analysis for window selection, which we call Multi-scale Spectral Residue (MSR).

The SR is a fast and efficient method for computing salient regions. However, some limitations of its original formulation make its use inadequate for multi saliency detection on cluttered images. These are:

1. Objects are searched in a single scale. SR resizes images to a fixed size – 64 by 64 (or the closest resolution that preserves aspect ratio). Restricting the scale of search limits the size of objects that can be found;
2. The threshold to decide which pixels of the saliency map are actually salient is defined as $k_{\text{SR}} = 3 \cdot E(S(x))$, or three times the mean saliency map, $S(x)$, intensity. Such formulation may incorrectly regard objects in cluttered images (many objects) as non-salient. This happens because with too many salient objects the threshold k becomes too large, excluding most regions;
3. Regions are chosen based on per-pixel saliency value, which may partially exclude objects that have a strong intra-object saliency variation.



Figure 4.2: Differences between original SR pixel selection and MSR window selection. From left to right: input image, SR per-pixel selection, MSR per-window selection

The aforementioned points were overcome in MSR, and they are detailed in the following sections.

4.1 Selecting regions

After calculating the saliency of an image, SR chooses salient pixels based on whether each individual pixel x is greater than three times the mean image saliency intensity, that is, $x > 3 \cdot E(S(x))$. There are two problems with this formulation: (i) due to object saliency variations along its length, objects may be only partially salient, creating incomplete object selection; (ii) using the k_{SR} as threshold on scenes with many objects (cluttered) can set the threshold to a value too high to include all image objects.

To solve these issues, the first step is exploring a per-window region selection. When sliding over the image, the average saliency of the pixels inside a window will be calculated and compared against the threshold. In case a window mean saliency is greater than the threshold, it will be selected for object detection. Using an entire window compensates object saliency variation, either selecting the entire window or discarding it. The differences between per-pixel and per-window region selection are illustrated in Fig. 4.2.

Using a rectangular window for saliency calculation has many advantages. Among these is the possibility of using integral images [Viola & Jones 2001] to calculate mean saliency of a window in $O(1)$. This is beneficial, as calculation of an integral image requires only an initial operation with linear complexity, that is, $O(n)$. This runtime speed advantage when compared to a normal approach without integral image is demonstrated in Fig. 4.3.

Another advantage is that common detectors, such as HOG, Haar-like features, also depends on a rectangular window of fixed-size for detection, this allows a one-by-one relation between the salient window selection and the object detection.

The window size for mean saliency calculation is defined to have the same size as the

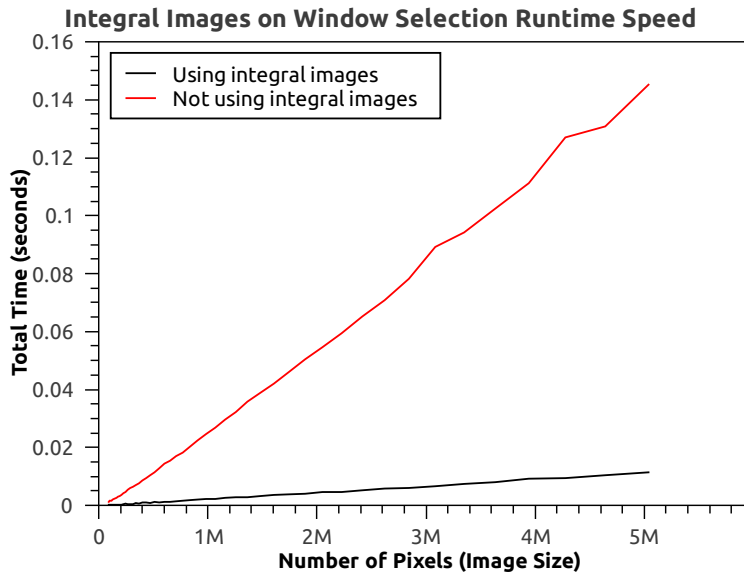


Figure 4.3: Difference of runtime speed required to select windows for an entire image octave using MSR with and without integral images for window saliency mean calculation.

object detector window. However, reducing the window size for mean saliency in some circumstances can improve saliency performance. One such case is when the classifier was trained with context from the object’s surrounding region, such as ground context for person in INRIA [Dalal & Triggs 2005] and NICTA [Overett *et al.* 2008]. As surrounding context is commonly non-salient, the saliency mean may be calculated using a window smaller than the detector window.

4.2 Saliency over multiple scales

A sliding window based object detector searches for objects with a fixed size window. To find objects of different sizes, the image is progressively downsampled using a function such as $I_{i+1} = R(I_i, s)$, where R represents the resize function, I_i denotes the i -th image octave and s the resizing factor. This downsampling allows a fixed-size window to encompass objects with different sizes. Our objective is to provide at each scale I_i a saliency map tuned to the search scale of the object detector.

Saliency detectors usually combine information from multiple scales to score regions based on their visual importance. Particularly, both [Itti *et al.* 1998] and [Harel *et al.* 2007] use a normalization operator to combine information from multiple visual scales into a single saliency map. This is not adequate for scale-specific saliency

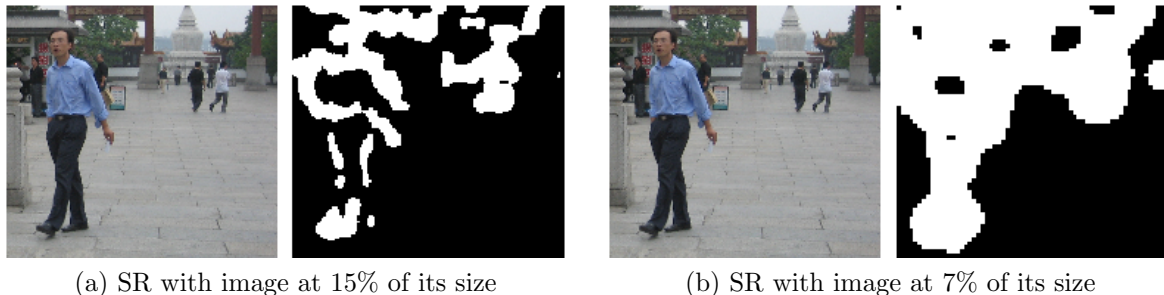


Figure 4.4: Differences in saliency at multiple scales. In the left image, SR was calculated in 15% of the original image size, generating strong reactions on mostly small objects; in the right, using 7% of the original image size, bigger objects were also selected. The image reduction examples demonstrate how the image size influences on the scale of saliency detection, which will be tuned to best select objects in a given octave.

detection as information from the final saliency map will be composed with information from multiple scales. In contrast, SR searches for interesting regions at a single scale. The scale of search in SR is defined by the input image size. As the image gets smaller, the search scale focuses more on larger objects. This can be seen in Fig. 4.4 where using a smaller image allows for bigger salient regions to be detected. A specific value for search scale was defined based on estimation of object size over common visual conditions – the closest possible resolution to 64 by 64.

Although the original SR formulation is an adequate choice for general purpose salient detection, it is not suited for integration with a sliding window object detector. This is a consequence of its fixed scale search, which does not select salient region that are aligned with the object detector search scale. Thus, we attempted to find, at a given image octave I_i , a resizing factor β that allows for SR to search for salient objects at the same scale as the object detector. That is, before a sliding window is applied at a particular image octave I_i , a resizing function $\mathcal{Z}(I_i, \beta)$ is applied and salient detection performed. The generated saliency map is then used to calculate a quality value $f(w)$ for each window w , and to decide whether the object detector should be used or not.

Resizing the image using β has an additional advantage of reducing the computational complexity of saliency calculation. Thus, a smaller β induces a smaller overhead for saliency calculation. However, the choice at a specific value for β depends on different factors, such as object class, saliency method and window size. In Section 4.3, we define performance metrics that will help with choosing a proper value for β on Section 4.4.

4.3 Quality function threshold

Proper evaluation of window selection impact on detector performance was done by means of an analysis of the window selection rate (WSR) and saliency false negative rate (SFNR). WSR denotes the number of windows selected for further processing, while SFNR represents how many objects the detector failed to recognize after MSR pruning.

Both WSR and SFNR depend on a threshold k which represents a minimum score for a window to be selected for actual object detection. Thus, given that W is the set of all windows generated from sliding on the entire collection of images at every scale and M the set of all objects of interest from this same collection of images, we can calculate the trade-off between WSR_k and $SFNR_k$ in a five-step process. First, we define the set of selected windows S_k as

$$S_k = \{w \in W \mid f(w) \geq k\}, \quad (4.1)$$

where $f(w)$ is the quality value of a window w and k is the threshold for window selection. Given S_k , it is possible to calculate the window selection rate with

$$WSR_k = \frac{n(S_k)}{n(W)}, \quad (4.2)$$

where $n(\cdot)$ denotes cardinality of a set. To calculate the $SFNR_k$ one should enumerate for each object $j \in M$ the number of windows in which the object was correctly matched, $C_{k,j}$, given by

$$C_{k,j} = \{w \in S_k \mid o(w) = j\}, \quad (4.3)$$

where $o(w)$ is a function that, in case an object exists at window w , and this is correctly classified by a detector, returns the matched object from set M ; otherwise $o(w)$ returns any element $\notin M$. From that, it is trivial to find the set of objects detected, F_k , defined as

$$F_k = \{j \in M \mid n(C_{k,j}) \geq 1\}. \quad (4.4)$$

Finally, in order to calculate how many miss detections were caused by the saliency method (SFNR), we used

$$SFNR_k = \frac{n(F_{k_{\min}}) - n(F_k)}{n(F_{k_{\min}})}, \quad (4.5)$$

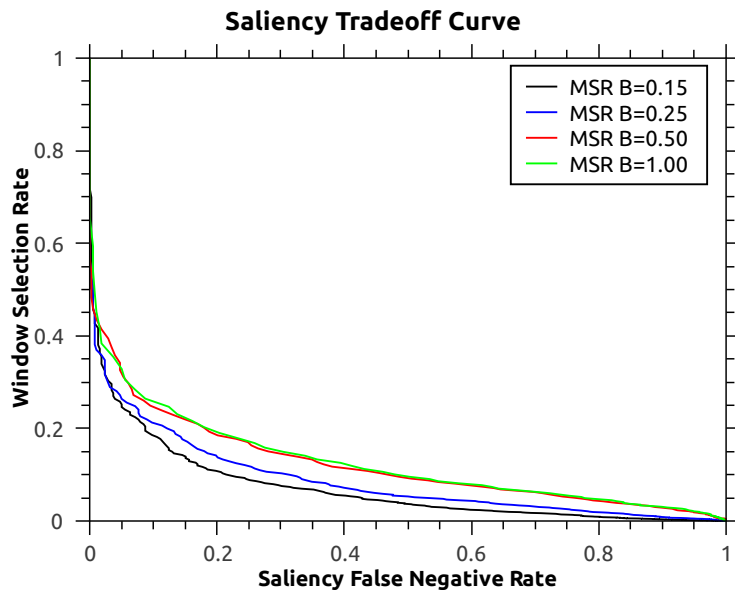


Figure 4.5: Trade-off curve for person detection using different β values. When the curve is closer to the origin it is better.

where k_{\min} is the minimum threshold value, which guarantees $S_{k_{\min}} = W$. Thus, to generate a full trade-off curve. This process is repeated for each $k \in K$, where K is the set of unique window scores.

4.4 β and k values

The value of β controls the search scale of the saliency detector. Thus, each object class may have a different optimal β value, varying according to the characteristics of the object. In particular, we found that person detection benefits more from salient region information at β value of 0.15 – or 15% of the current octave size. This size was found for person objects over the LabelMe [Russell *et al.* 2008] dataset, further details are shown in Chapter 5. Still in the context of person detection, Figure 4.5 shows the trade-off of WSR and SFNR for different values of β . This trade-off indicates that the value 0.15 for β allow the saliency search scale to better match the search scale of the detector for person objects. In practice, this allows more windows to be discarded without additional false negative detections.

Using only single scale, as in the original SR, would result on a smaller footprint for saliency detection, at the cost of much reduced window selection performance. However, as our multi-scale approach had such superior performance in comparison to single scales

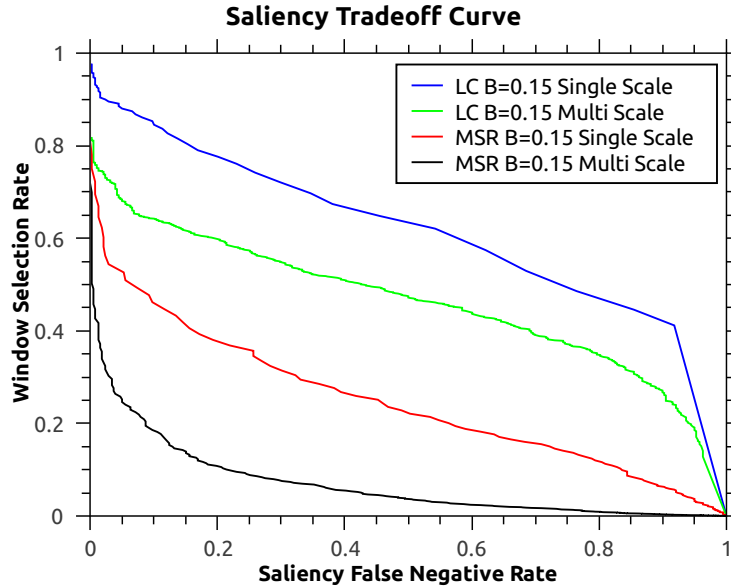


Figure 4.6: Comparison between multi-scale analysis and using the same saliency map for all scales. Methods presented are MSR and LC [Zhai & Shah 2006]. When the curve is closer to the origin it is better.

methods, the additional cost of recalculating saliency for each scale is offset by the greater number of windows discarded, as can be noted in Fig. 4.6. This figure shows that saliency analysis over multiple scales has, for all threshold values (and fixed β), a better performance when compared to single scale analysis.

On the following sections, to facilitate analysis of results, we have two values for threshold k which, in average, are equivalent to operating points 20% and 30% of WSR. In this manner, at 20% of WSR, the detection is expected to run five times faster, generating a high false negatives, while at 30% of WSR the algorithm executes over three times faster but generating very few false negatives. The actual runtime speed impact, taking into account the overhead of saliency calculation, is presented in Section 5.1.4.

The choice of threshold value is important, as choosing a threshold that is too high will most likely inadvertently discard windows that contain an object. Conversely, choosing a too low threshold will discard too few windows. These possibilities are depicted in Figure 4.7.

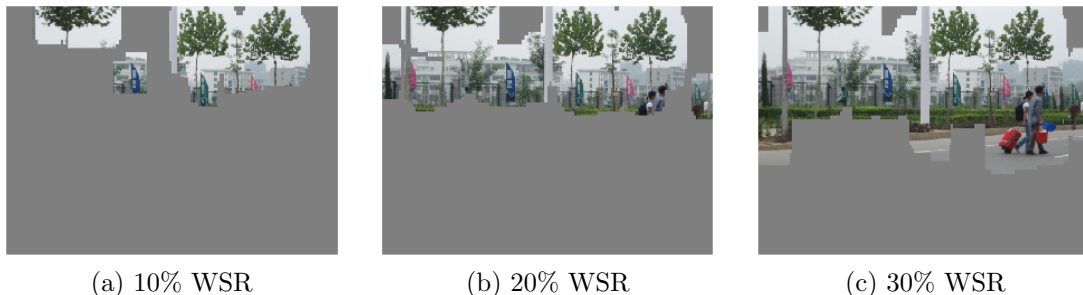


Figure 4.7: Different regions selected depending on threshold value.

4.5 Image pre-processing

Preliminary analysis of the saliency false negatives demonstrated that most were effects of bad illumination, clothing with darker tones and distance from the camera. These characteristics reduced the difference from an object to its background, negatively impacting its saliency.

The first attempt to tackle the aforementioned problem was based on contrast stretching, which normalizes the gray-level intensity distribution in the entire zero and one range. This is done through a linear normalization of an image I , using

$$I_N = (I - \min(I)) \frac{255 - 0}{\max(I) - \min(I)}. \quad (4.6)$$

Stretching the intensity distribution did not improve algorithm window selection performance (nor degraded it). Therefore, we supposed that most photographs, taken in natural conditions, already include an overall good gray-level intensity distribution.

Instead of normalizing gray-level intensity, our second attempt used contrast normalization by histogram equalization (HE), detailed in Appendix A. As such, this algorithm was applied in an attempt to correct low contrast caused by poor illumination.

The histogram equalization is applied after β scaling and image conversion to gray-level. This equalization requires calculation of the image pixel frequency histogram. This process was used to provide a better trade-off with minimal extra runtime cost, as it is calculated in the image generated after β scaling. Adding this equalization, in our tests, it provided 8.7% of SFNR at 20% of SWR and also 2.9% SFNR at 30% SWR. This solution was used in the first iteration of our method, presented in [Silva *et al.* 2012]. On the remaining of the current section we evaluate other normalization approaches and their impact on window selection performance.

Another approach to image pre-processing is based on Adaptive Histogram Equaliza-

| Method | Block Size | N. Bins | WSR 20% | WSR 30% |
|------------|-----------------|-----------|--------------------|--------------------|
| HE | - | - | 08.70% SFNR | 02.90% SFNR |
| AHE | 48 by 48 | 16 | 06.61% SFNR | 03.17% SFNR |
| AHE | 48 by 48 | 32 | 06.61% SFNR | 03.70% SFNR |
| AHE | 48 by 48 | 64 | 05.55% SFNR | 02.64% SFNR |
| AHE | 48 by 48 | 96 | 05.55% SFNR | 02.64% SFNR |
| AHE | 64 by 64 | 16 | 07.93% SFNR | 05.55% SFNR |
| AHE | 64 by 64 | 32 | 07.93% SFNR | 05.55% SFNR |
| AHE | 64 by 64 | 64 | 07.93% SFNR | 05.02% SFNR |
| AHE | 64 by 64 | 96 | 05.82% SFNR | 02.64% SFNR |

Table 4.1: Window selection performance over different parameters for AHE algorithm and compared to HE. Best results for each section are marked in bold

tion (AHE) algorithms [Pizer *et al.* 1987]. Specifically, both AHE and the contrast limited version (CLAHE) were tested. When certain regions are darker or more illuminated than the rest of an image, adaptive methods are capable of a better contrast enhancement in comparison to global methods, such as histogram equalization. Additional information about these methods can be found on Appendix A.

Some adjustments were made to MSR structure to allow use of AHE and CLAHE contrast normalization methods. These adaptive contrast normalization methods divide an image into several distinct regions (or blocks). However, to avoid partial blocks at image edges, the input image is resized to the nearest upper multiple of block size:

$$b(x, m) = x + m - (x \bmod m), \quad (4.7)$$

where x denotes the size of an image dimension (width or height) and m the block size. Our algorithm is tested with block size of 48 and of 64. Furthermore, another parameter required is the number of bins for each block histogram. In our tests, we evaluate as possible the values of 16, 32, 64, 96. Henceforth, the AHE was applied over several images in several configurations, and the results were summarized in Table 4.1. The same process was repeated with CLAHE, but limited to the best results of AHE (from Table 4.1), over different contrast limit values. In Table 4.2, the results for CLAHE are presented.

The CLAHE method achieved better results (see Table 4.2) in window selection performance, when configured to use a block size of 48 by 48 with 96 histogram bins and 0.3 contrast normalization. The contrast limit avoids over-amplification of noise, common in

| Method | Block Size | N. Bins | Contrast Limit | WSR 20% | WSR 30% |
|--------------|-----------------|-----------|----------------|--------------------|--------------------|
| HE | - | - | - | 08.70% SFNR | 02.90% SFNR |
| AHE | 48 by 48 | 96 | - | 05.55% SFNR | 02.64% SFNR |
| AHE | 64 by 64 | 96 | - | 05.82% SFNR | 02.64% SFNR |
| CLAHE | 48 by 48 | 96 | 0.3 | 04.23% SFNR | 01.58% SFNR |
| CLAHE | 48 by 48 | 96 | 0.6 | 04.23% SFNR | 02.38% SFNR |
| CLAHE | 48 by 48 | 96 | 0.9 | 05.82% SFNR | 03.17% SFNR |
| CLAHE | 64 by 64 | 96 | 0.3 | 05.29% SFNR | 02.11% SFNR |
| CLAHE | 64 by 64 | 96 | 0.6 | 04.76% SFNR | 02.91% SFNR |
| CLAHE | 64 by 64 | 96 | 0.9 | 06.87 % SFNR | 02.91% SFNR |

Table 4.2: Window selection performance over different parameters over a person dataset for CLAHE algorithm compared to HE and AHE. Best results for each section are marked in bold.

HE and AHE methods. The limit itself defines an upper bound to how much the contrast in a pixel neighborhood can be amplified. Thus, a limit value of one would indicate no contrast limiting, and as it approaches zero only smaller contrast amplifications are allowed. For further details about CLAHE parameters see Appendix A.

A performance overview of best configuration for each method is presented in Fig. 4.8. These results indicate that CLAHE methods are best suited to enhance MSR window selection performance within the operating points of 20% and 30% of SWR. However, other operating points or objects may require different approaches.

It is important to note that CLAHE operate over image blocks, while HE operate over the entire input image. Thus, combining both local and global approaches may yield superior window selection performance. To this end, we test how performance is affected when both CLAHE and HE are applied simultaneously over the image. The results, presented in Table 4.3, indicate that CLAHE and HE are better at window selection performance than each method in isolation, with 3.34% of SFNR at 20% SWR and 1.05% of SFNR at 30% SWR.

A proper choice for contrast normalization depends not only on window selection performance considerations but also on runtime speed requirements. Evaluation of each method impact on saliency detection speed is measured in Table 4.4. These experiments have indicated that for smaller resolutions the runtime overhead of contrast normalization is minimal. However, as image resolution grows, the overhead for HE, AHE and CLAHE methods also increases.

Images, that even after β scaling are very large, may impose a large runtime overhead.

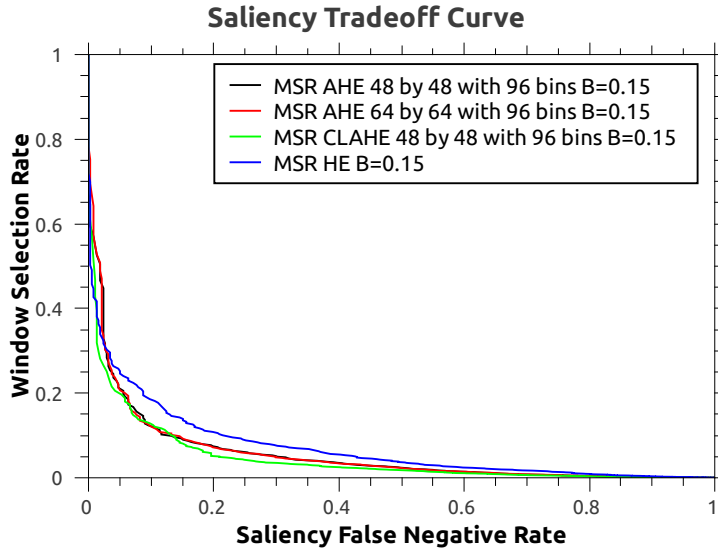


Figure 4.8: Comparison of best results in terms of window selection performance versus false negatives generated over a person dataset. CLAHE is presented with 0.3 as contrast limit.

| Method | Block Size | N. Bins | Contrast Limit | WSR 20% | WSR 30% |
|------------|------------|---------|----------------|-------------|-------------|
| HE | - | - | - | 08.70% SFNR | 02.90% SFNR |
| CLAHE | 48 by 48 | 96 | 0.3 | 05.82% SFNR | 03.17% SFNR |
| CLAHE | 64 by 64 | 96 | 0.3 | 05.29% SFNR | 02.11% SFNR |
| CLAHE + HE | 48 by 48 | 96 | 0.3 | 03.43% SFNR | 01.05% SFNR |
| CLAHE + HE | 64 by 64 | 96 | 0.3 | 03.17% SFNR | 01.05% SFNR |

Table 4.3: Results of window selection performance from a combination of both CLAHE and HE methods.

Thus, a possible solution to this effect is to split an image into several sub-parts, and each one can be normalized independently. This may, however, introduce fragments in the edges between regions.

Judging whether the overhead of contrast normalization is acceptable or not will depend on the choice of object detector. In this case, a necessary condition for MSR effectiveness is that $T_{MSR}(w) \ll T_d(w)$, where $T_{MSR}(w)$ is the time required by saliency detection (with normalization) divided by total number of windows, and $T_d(w)$ is the time required by a detector to process a window.

Our conclusion is that the enhanced image contrast improved the results at window selection. This implies that spectral residual relies on high contrast regions for some of its saliency detection. Thus, contrast equalization shows itself to be an important step

| Resolution | Saliency Runtime (s) | | | | |
|--------------|----------------------|---------------------|--------|--------|--------|
| | HE | Contrast Stretching | AHE | CLAHE | None |
| 1600 by 1200 | 0.0880 | 0.0870 | 0.0863 | 0.0863 | 0.0847 |
| 2048 by 1536 | 0.4618 | 0.1209 | 0.1449 | 0.1453 | 0.1203 |
| 2592 by 1944 | 0.6900 | 0.1400 | 0.7000 | 0.6800 | 0.1400 |

Table 4.4: Comparison of runtime speed differences for saliency detection (MSR with β scaling) using different contrast normalization approaches

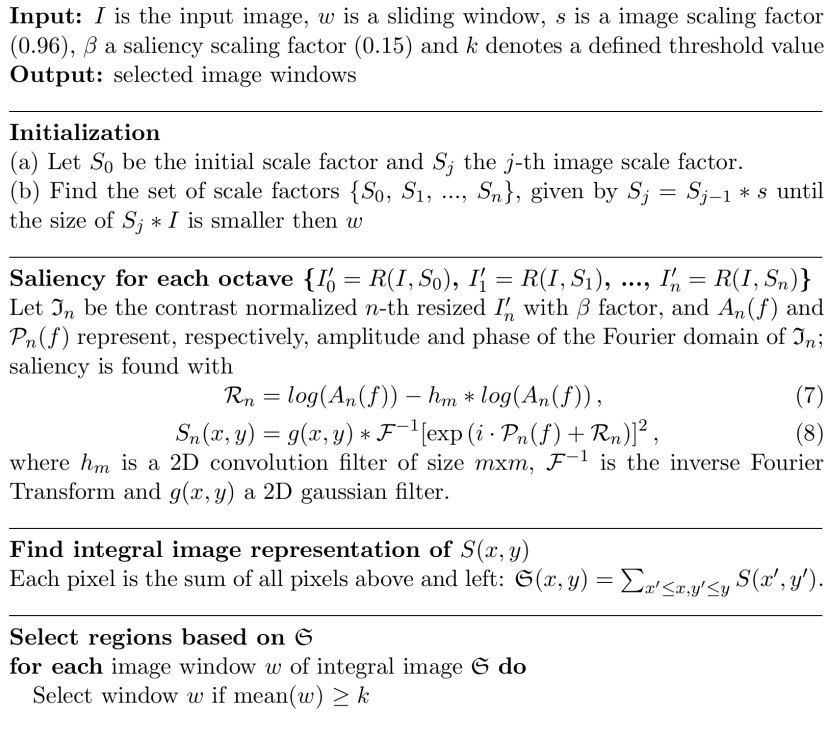


Figure 4.9: MSR window selection procedure (parameters for person detection)

for MSR. Figure 4.9 summarizes all the important steps in MSR.

4.6 Closure

This chapter described overall characteristics of MSR. This method was created to enable faster object detection, avoiding the use of computationally expensive object detectors in regions with low probability of containing objects. To achieve such result, our method combines techniques such as saliency detection, contrast normalization, integral images

and object detection. Moreover, concepts of WSR and SFNR are also presented to allow clear analysis of performance variations in Chapter 5.

Saliency detection was enhanced by combining contrast normalization techniques. Best results were achieved through combined CLAHE and HE methods. Additionally, runtime differences between different techniques were evaluated and compared. These results have indicated that SR saliency detection relies on edge information for proper saliency detection, information which can be exploited in future works.

Experimental evaluation

Contents

| | | |
|-------|---|-----------|
| 5.1 | Experiments | 63 |
| 5.1.1 | Comparison of saliency methods in a multi-scale structure | 64 |
| 5.1.2 | Scalability | 66 |
| 5.1.3 | Detection performance | 67 |
| 5.1.4 | Runtime performance | 69 |
| 5.1.5 | Per-class MSR performance | 70 |
| 5.2 | Analysis and closure | 72 |

Proper evaluation of how well search space reduction is performed in a sliding window-based detector is necessary, as this helps to gauge MSR usefulness. This way, several points must be evaluated:

- (i) Compare the detection performance considering several saliency detection methods using the same sliding window parametrization in a multi-scale analysis;
- (ii) Analyze MSR scalability with respect to detection, i.e., how it behaves on different image resolutions;
- (iii) Quantify how MSR affects a detector's receiver operating characteristic (ROC) curve with respect to a regular sliding window;
- (iv) Measure the impact on detection runtime speed with different parameters;
- (v) Evaluate the performance for different object classes.

To determine each of the the aforementioned items, two datasets have been chosen:



Figure 5.1: Sample images of the dataset created from the LabelMe repository.



Figure 5.2: Sample images from the Pascal VOC 2007 dataset.

- 330 images¹ extracted from LabelMe repository [Russell *et al.* 2008] containing persons in several different environments, and with image sizes ranging from 320 by 240 to 2592 by 1944. Additionally, the dataset encompasses several environments, including city, snow, forest and river, where each selected scene contains at least one person. Examples can be seen in Figure 5.1. Even though other datasets could be used, such as INRIA, the LabelMe provides scenes from several authors in a wide range of situations, which allows for a more randomized image sampling;
- Pascal VOC 2007 dataset, containing 4952 images with twenty object classes. Example images from that dataset can be seen in Figure 5.2.

Analyses of items (i), (ii), (iii), (iv) are done with the dataset extracted from LabelMe. In these cases, the persons are detected using MSR and a combination of histogram of oriented gradients (HOG) [Dalal & Triggs 2005] and Support Vector Machine (SVM). The reason of using HOG/SVM was not only because it is a state-of-the-art detector, but also to facilitate comparison with other future search reduction methods, since its source code is publicly available. Our HOG/SVM detector was trained using a person dataset distinct from the one created with images from LabelMe. Additionally, the detector was set up with window size of 64 by 128 pixels, a stride of 8 horizontal pixels, and 16 vertical pixels, and image resizing rate of 0.96, for each octave.

¹The images and annotations are available for download at goo.gl/pmuEw

Additionally, for (iii) we also include a comparison of detection performance against a standard Viola Jones object detector [Viola & Jones 2001]. This is done to help understand how MSR interacts with different detection approaches. This detector was configured with window size of 56 by 112, and the same stride as used by the HOG detector.

It is noteworthy that, for analysis (i), two state-of-the-art saliency methods have not been included – [Cheng *et al.* 2011b] and [Goferman *et al.* 2011]. The former, because the saliency detection is concentrated mostly on images with a single and clear salient object; the latter, because of its very slow runtime speed. In (ii), we examine how MSR performance changes over different image resolutions and how each image octave contributes to its results. This experiment is important when recent increases in availability of high resolution images are considered. In (iii), we built a ROC curve to show the effects of MSR on a person detector at different WSR configurations in comparison to a normal sliding window. Moreover, in (iv), we evaluate if the number of windows discarded before detection is sufficient to compensate for the additional processing required by MSR. For (v), we use Pascal VOC 2007 dataset to see which object classes are best suited for MSR. The resulting MSR performance for each class shows which objects adapt better to MSR structure for window selection.

5.1 Experiments

In this section we describe experiments performed to evaluate MSR overall performance. Experiments (i-iv) which use LabelMe dataset are defined respectively in Sections 5.1.1, 5.1.2, 5.1.3 and 5.1.4. On experiment (v), presented on Section 5.1.5, results were collected from the application of MSR for each class type of Pascal VOC 2007 dataset.

For practical reasons, we used on experiments (i), (ii) and (iv) only a HOG/SVM detector while on (iii) a Viola Jones rejection cascade detector is also included.

Given the several possible configurations for our method, we define the following short-hands as being configured with:

MSR defines our method configured to use both CLAHE (with block size of 64 by 64 and 96 bins) and HE. This choice was based on the good results obtained in comparison to other configurations. This can be seen in Section 4.5.

MSR HE defines our method using only HE for histogram equalization, which was used in the first iteration of our algorithm, presented in [Silva *et al.* 2012].

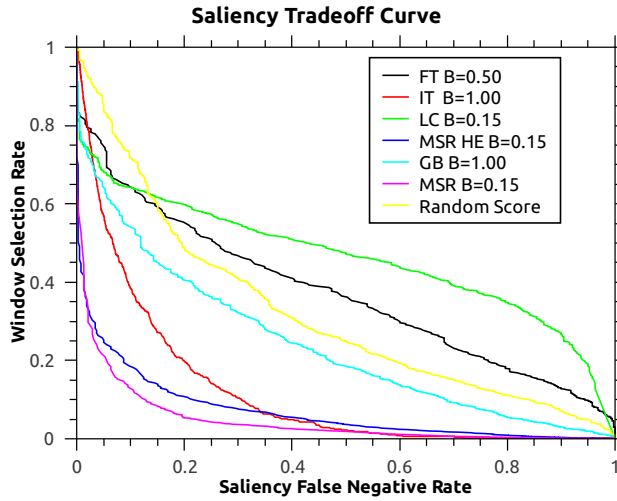


Figure 5.3: Comparison of best results from different saliency methods applied to guide multi-scale detectors, the methods are MSR (using CLAHE + HE), MSR HE [Silva *et al.* 2012], FT [Achanta *et al.* 2009], GB [Harel *et al.* 2007], IT [Itti *et al.* 1998], LC [Zhai & Shah 2006] and a baseline using random window scoring. The false negative rate represents only objects that the detector would have matched if a regular sliding window approach had been used. When the curve is closer to the origin it is better.

5.1.1 Comparison of saliency methods in a multi-scale structure

A comparison of MSR against other state-of-the-art methods in the same multi-scale structure is presented in Table 5.3. As the results represent only the best configuration of each method, a more detailed information is organized on Table 5.1 and 5.2.

In these experiments, both IT [Itti *et al.* 1998] and GB [Harel *et al.* 2007] were only evaluated using the original octave size, with no β scaling, as these methods already perform saliency detection at multiple scales internally. We also did not compare MSR with the original SR [Hou & Zhang 2007] since the latter one was designed to operate on a single image size.

The results indicated that MSR achieved superior performance on almost the entire trade-off curve. Both MSR and IT became nearly equal in window selection performance when operating under 2% WSR. However, even if it is necessary to operate with such small WSR, saliency calculation speed of SR (from which MSR was inspired) still outperforms IT in runtime speed, see Chapter 3 for runtime speed comparisons. In addition, false negative rates at 20% and 30% of WSR were at least ten times better when compared to other methods. Yet, at 50% of WSR, the SFNR is close to zero (less than 0.3%).

The MSR and IT methods had the best overall trade-off between WSR and SFNR. Both SR and IT were among the worst on recent evaluation of general purpose saliency

Table 5.1: SFNR for each method at 20% of WSR

| Method | β | | | |
|------------------|---------|--------|--------|--------|
| | 0.15 | 0.25 | 0.50 | 1.00 |
| MSR ^a | 03.17% | 5.29% | 9.25% | 22.48% |
| LC | 93.92% | 93.92% | 94.18% | 94.43% |
| FT | 87.83% | 82.54% | 76.46% | 77.25% |
| GB | - | - | - | 47.09% |
| IT | - | - | - | 38.89% |

^aThe original SR was not used since it is not adequate for searching in multiple scales, see Section 4.4.

Table 5.2: SFNR for each method at 30% of WSR

| Method | β | | | |
|------------------|---------|--------|--------|--------|
| | 0.15 | 0.25 | 0.50 | 1.00 |
| MSR ^a | 1.05% | 1.85% | 6.08% | 14.81% |
| LC | 86.24% | 86.24% | 85.98% | 87.04% |
| FT | 70.37% | 64.29% | 59.79% | 62.69% |
| GB | - | - | - | 33.86% |
| IT | - | - | - | 23.81% |

^aThe original SR was not used since it is not adequate for searching in multiple scales, see Section 4.4.

detection found in [Cheng *et al.* 2011a]. Some possible causes of this discrepancy are:

1. Saliency detection is done in a single scale in [Cheng *et al.* 2011a], while in our tests the saliency was recalculated at every octave. This provided a better performance for most methods;
2. Differences in scene selection. The dataset used in [Cheng *et al.* 2011a] was gathered by [Achanta *et al.* 2009] with images containing mostly uncluttered objects and natural background. In our tests, images were extracted from LabelMe [Russell *et al.* 2008], wherein the images contain a wide range of locations and varying degrees of clutter;
3. Insufficient background information on some images (large objects).

MSR was also compared against simple random window scoring. This guarantees that our approach for saliency detection is better than chance. The results indicate that MSR

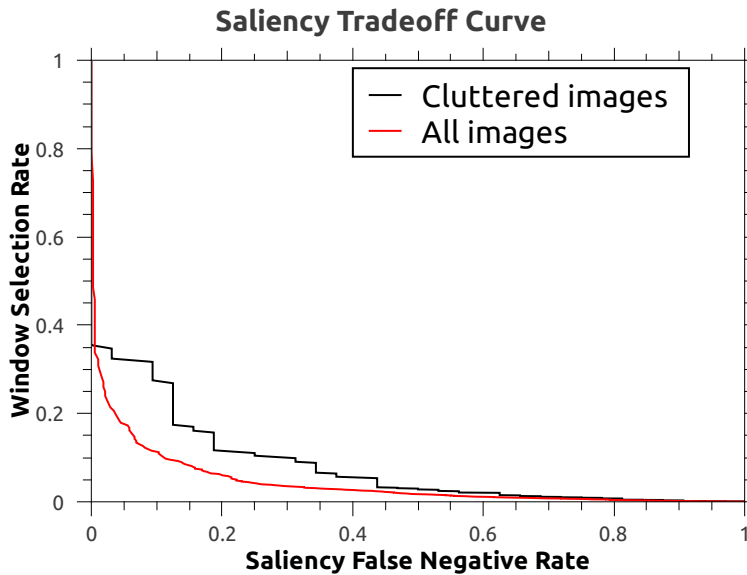


Figure 5.4: Comparison of window selection results using MSR (with HE + CLAHE) against a cluttered and the complete LabelMe dataset.

greatly outperforms random window selection, showing that our approach does indeed allow for better search space reduction.

Furthermore, a subset of 20 cluttered images from the LabelMe dataset were used to evaluate variations on expected window selection performance, these results are presented in Figure 5.4. The results showed that a variation in performance exists, showing that in very cluttered scenarios window selection may not be as useful.

5.1.2 Scalability

We compared how well MSR could select image windows at different starting image resolutions in Fig. 5.5. From this information, we can conclude that increasing image size allows for an even better trade-off between SFNR and WSR.

To further confirm the scalability of MSR on larger resolutions, we compared its ability to eliminate windows at a fixed threshold in several octaves in Fig. 5.6, showing that larger size images contributed for better WSR. In this test, the number of windows in each octave was calculated using Eq. 2.13.

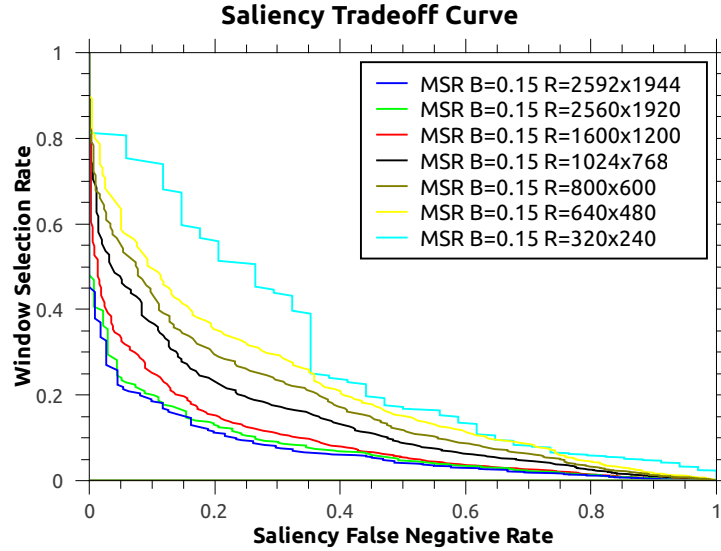


Figure 5.5: Trade-off between WSR and SFNR at different starting resolutions. Aspect ratio is kept by approximating the image resolution to the closest image size. When the curve is closer to the origin it is better.

5.1.3 Detection performance

To measure the impact on MSR impact on detector performance, a ROC curve for the detector with with and without MSR was compared. This curve is generated by sliding a fixed size window along the image in several scales, and then applying non-max suppression (NMS) on the detected windows. The remaining windows (after NMS) are then compared to the groundtruth using

$$A_1 \text{ is true positive} \iff \frac{A_1 \cap A_2}{A_1 \cup A_2} \geq 0.5 \quad (5.1)$$

where A_1 is a detected rectangle and A_2 a groundtruth rectangle.

Comparison between a HOG/SVM object detector with and without MSR is presented on Fig. 5.7a. In the tests, MSR at 20% of WSR provided greater TPR than regular sliding window within the range of 0 and 1.48 of FPPI. At 30% of WSR and within the range of 0 and 1.98 of FPPI, our method also obtained larger TPR than a regular sliding window approach. The maximum TPR of a regular sliding window was 0.71, while for MSR at 30% of WSR the maximum is 0.69. Even though the difference was small to match the actual maximum TPR of a regular sliding window, MSR operated at least on 50% of WSR, which still represents a twice as fast image processing with only a negligible performance loss (less than 0.3%).

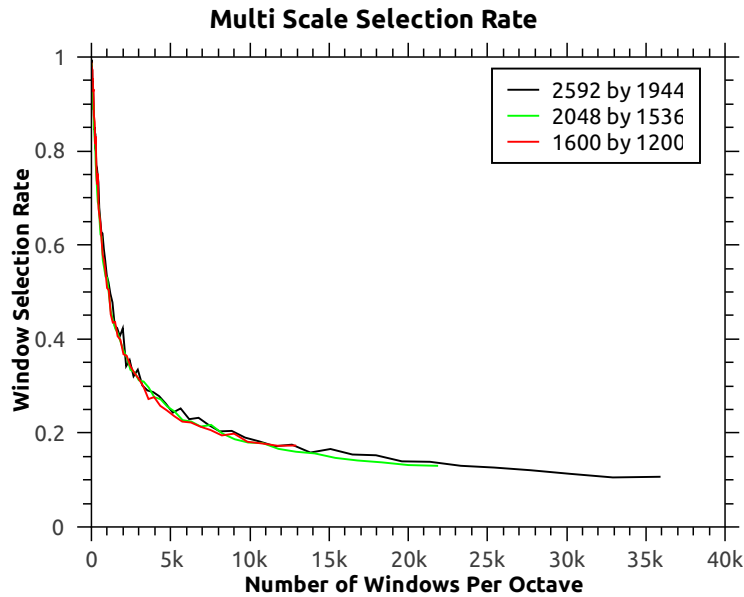


Figure 5.6: Relation between number of windows at each image size and number of windows selected for the detector. An operating point was selected at 20% of WSR (see Fig. 5.3 for reference).

Some examples of positive and negative results at 30% of WSR can be found, respectively, on Fig. 5.8 and Fig. 5.9.

Furthermore, to show that MSR can be used with different detectors, person detection performance using Viola and Jones' method with and without MSR was evaluated. In Figure 5.7b the comparison of detector performance is showed. The ROC curve shows that Viola and Jones' detector, in average, makes more mistakes per image than a HOG/SVM detector. Even so, MSR was able to reduce the number of false positives per image, improving or at least maintaining performance within the range of 0 to 5.45 FPPI. Compared to our method, that obtained 3.68 FPPI with 0.35 TPR, the regular approach was only able to achieve this performance with 4.59 FPPI.

Another point is that the FPPI for the maximum TPR of each method was: (1) FPPI 6.81 with 0.38 TPR for the standard Viola Jones and (2) 5.45 FPPI but also with 0.38 TPR for our version combining Viola Jones and MSR. The actual difference between the maximum TPR of both methods was of only 0.004033, showing that the number of lost detections was negligible. Thus, in Fig. 5.10 we present some examples of positive results of MSR, we omit the negative results since there are too few samples for a meaningful overview.

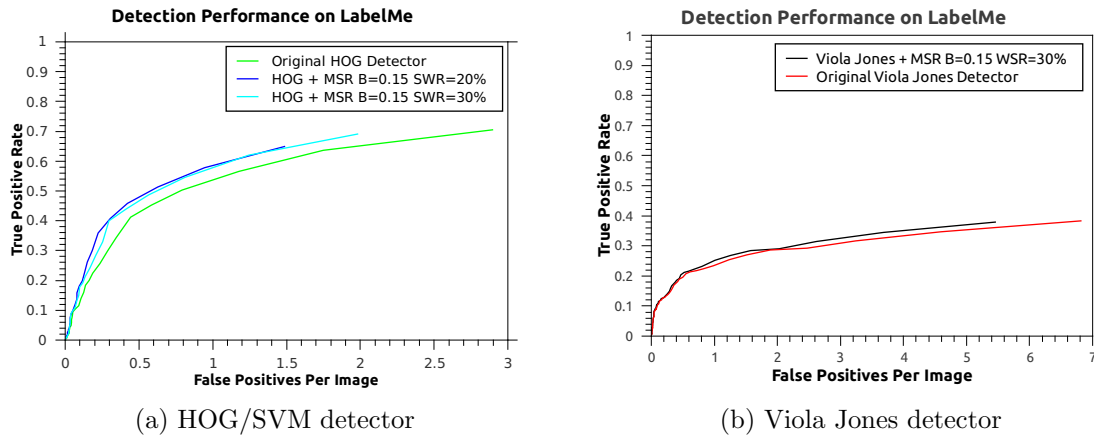


Figure 5.7: ROC curve showing differences between person detection performance using a regular sliding window and MSR.

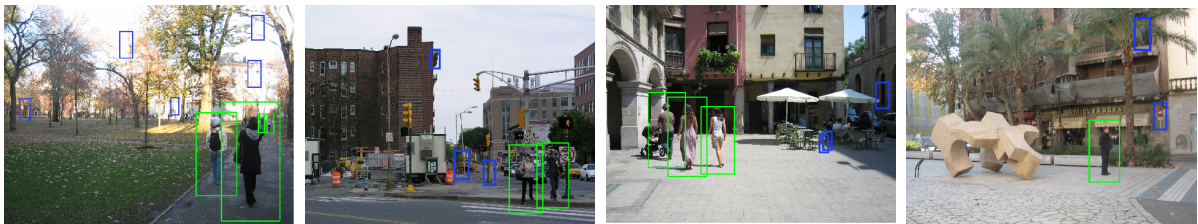


Figure 5.8: Positive results from using a HOG/SVM detector with MSR, positive results at 30% of WSR after non-max suppression. Blue rectangles indicate avoided false positives (improving performance); TP are marked with green.

5.1.4 Runtime performance

In order to evaluate MSR runtime speed, a comparison was performed with the traditional sliding window HOG detector. We summarized the results on Table 5.3. Time was calculated as the proportion of the total detection time for a specific WSR value of a regular sliding window execution.

Expected gain, considering elimination of 80% and 70% of windows to be classified, was 5x and 3.3x when compared to the same detector using no window selection (regular sliding window). However, the results demonstrated that for both 19.9% and 29.6% of WSR², the actual runtime speed gain was smaller than 4.8x and 3.2x. This indicates that MSR window selection mechanism imposed only a small processing overhead for each window, which was compensated by the large number of windows discarded.

²The closest thresholds to 20% and 30% of WSR, respectively. Equivalent to 80% and 70% of window elimination.



Figure 5.9: Negative results from using a HOG/SVM detector with MSR, positive results at 30% of WSR after non-max suppression. Yellow rectangles indicate FN caused by MSR (affecting performance); blue rectangles indicate avoided false positives (improving performance); TP are marked with green, while red rectangles are FP.



Figure 5.10: Positive results from using a Viola Jones detector with MSR, positive results at 30% of WSR after non-max suppression. Blue rectangles indicate avoided false positives (improving performance); TP are marked with green and FP with red.

5.1.5 Per-class MSR performance

Although MSR was capable of providing, in general, better performance for person detection this does not hold for all object classes. As such, providing clear information about what classes could generate better MSR results can shed light in which use cases are better suited for MSR.

To evaluate a broad range of classes we chose Pascal VOC 2007 dataset, which contains several different scenes containing many different objects. For this evaluation 19 object classes (but excluding persons) were selected, which were divided into a set of 4952 images.

As it is not practical to use an object detector for each distinct object class, we consider a hypothetical flawless classifier (groundtruth). That is, we evaluated how well the object

Table 5.3: Runtime speed proportion for each method

| Method | WSR | Total Time Proportion | Avg. Time Proportion Per Window |
|--------------------|-------|-----------------------|---------------------------------|
| Regular Slide | 100% | 1.0000 | 1.0000 |
| MSR $\beta = 0.15$ | 19.9% | 0.1932 | 0.1996 |
| MSR $\beta = 0.15$ | 29.6% | 0.2852 | 0.2994 |

| Class | with 20% WSR | with 30% WSR |
|--------------|--------------|--------------|
| Cow | 18.56% SFNR | 8.38% SFNR |
| Bicycle | 17.83% SFNR | 11.53% SFNR |
| Aeroplane | 14.35% SFNR | 9.90% SFNR |
| Potted Plant | 37.39% SFNR | 24.36% SFNR |
| Chair | 45.92% SFNR | 35.38% SFNR |
| Horse | 9.00% SFNR | 5.90% SFNR |
| Sofa | 26.15% SFNR | 17.53% SFNR |
| Dining Table | 33.15% SFNR | 21.19% SFNR |
| Bird | 24.14% SFNR | 16.19% SFNR |
| Bus | 17.70% SFNR | 10.41% SFNR |
| Boat | 24.30% SFNR | 18.23% SFNR |
| Train | 22.13% SFNR | 11.88% SFNR |
| Cat | 9.28% SFNR | 6.81% SFNR |
| Bottle | 31.56% SFNR | 21.98% SFNR |
| TV Monitor | 36.79% SFNR | 23.80% SFNR |
| Dog | 13.77% SFNR | 6.44% SFNR |
| Sheep | 22.56% SFNR | 14.63% SFNR |
| Motorbike | 18.54% SFNR | 10.18% SFNR |

Table 5.4: Comparison of MSR performance using a hypothetical flawless detector in the Pascal VOC 2007 dataset.

itself stands out for window selection and not how it affected detector performance.

It is important to highlight that, when using a flawless object detector, the SFNR will generally be higher than when using an actual object detector. This increase in SFNR is a result of scenes with partially cropped, not illuminated or hidden objects, which generate low saliency and are also hard for an object detector to find. In these cases, an object detector would have a higher probability to generate a false negative, therefore, it would not affect the SFNR as both object detector and window selection would have been mistaken.

When evaluating object classes, a higher SFNR indicate that, in general, these objects generate a smaller saliency. Thus, this small saliency makes window selection less cost effective. Based on this information, the results from the evaluation of different object classes test are presented in Table 5.4. These indicate that the worst performing object classes are chair, potted plant, TV monitor and dining table, with, respectively, 45.92%, 37.39%, 36.79% and 33.15% of SFNR at 20% WSR. In contrast, the best performing classes were horse with 9% SFNR, cat with 9.28% SFNR and dog with 13.77% SFNR; also at 20% SFNR. Curiously, the three best performing classes were all animals while

the worst performing were man made objects mostly found within a house.

There are several hypothesis that could explain the aforementioned results. One of such is that man made objects are included in an environment which was design to be aesthetically pleasing (no strong colors). Other is that many of such man made objects have transparent parts or regions that, when demonstrated in a fixed-size rectangular window, include too many non-salient parts from the surrounding environment, for instance, a chair.

5.2 Analysis and closure

The MSR was designed to speed up object detection while maintaining or increasing detection performance. This chapter evaluated several characteristics of our method in order to understand its effect on an existing detector.

Our results indicated that, for person detection, MSR has been able to increase detector performance and reduce runtime speed. Particularly, at 20% of WSR our method obtained 3.17% of SFNR in LabelMe dataset. Yet, at 30% of WSR a result of 01.05% of SFNR was achieved.

These results indicate that MSR could allow a detector to execute close to five times faster with only a slight reduction in number of true positives. Moreover, this speed up was shown to scale well to larger images, increasing its applicability.

As MSR can also discard regions which would generate false positives, in this way, it also positively complements an existing detector overall performance. To reach this conclusion, different object detectors where combined with MSR and thoroughly tested. In all cases, MSR was capable to maintain or increase person detection performance.

Based on the analysis of person detection performance of different detectors, the HOG/SVM has achieved the best performance among tested detectors. Thus, this combination of HOG, SVM and MSR can be used as an effective tool for faster person detection.

Throughout evaluation different object classes included in Pascal VOC 2007 shows that, in general, MSR works best with animals and vehicles. On the other hand, man made house objects (chair, table) did not perform as well. Further work is necessary to objectively understand what object characteristics most affect saliency results.

The aforementioned tests and results indicate not only MSR can increase runtime speed in some existing state-of-the-art detectors but also its performance. This performance gain is a consequence of correctly discarding potential false positive detections, avoiding errors; while minimizing the number of lost true position detections caused by

window selection.

Work is being done to further speed up MSR by doing part of the saliency detection within the GPU. This will allow for better integration of our method with GPU-based object detectors.

Conclusion

This work described our approach to speed up image object detection by region selection using saliency information. This way, regions with small visual importance, indicated by the saliency detection method, were discarded before a full-fledged object detector was used, saving processing time.

Region selection was achieved by our method, called multi-scale spectral residue (MSR), which was based on a saliency detector called Spectral Residual (SR). MSR modified the original SR to better fit the purpose of region selection. These modifications allow for saliency analysis over multiple scales, provide better region selection even when objects inside a region have strong saliency intensity variations along its length, and also provide an easy way to control the balance between the number of regions discarded and the number of false negatives caused by this selection. Furthermore, to obtain better results, contrast normalization was used on the image to enhance the edges of objects. This normalization was necessary because SR has a strong dependence in object edges for saliency detection.

Evaluation of our approach indicated that MSR was able to increase execution speed of a HOG/SVM person detector while also increasing its ROC performance. For the Viola and Jones' detector, performance was also improved, showing that MSR can be adapted to different detector types. Moreover, our choice for spectral residue was compared to other state-of-the-art saliency detectors, showing that it indeed provide the best results by a wide margin. To summarize, the results obtained shown that MSR allowed object detectors to achieve not only faster detection but also better detection performance. The detection performance gains are achieved when MSR discard regions which would generate false positives in case a detector was applied at that particular region.

Among the restrictions of the MSR is that some object classes generate results which are not as good as others, decreasing its usefulness for some classes. Particularly, man made house objects were shown to generate less saliency than was expected, increasing the number of false negatives (at a constant number of windows discarded).

To further enhance the results obtained by MSR, we plan the following enhancements:

For short term we plan to determine what object particular characteristics most affect saliency and also how to capture such information more reliably, creating a better saliency method;

For long term our aim is to combine top-down information in our method. That is, capture object-specific saliency information to help further reduce the number of windows selected for detection.

Contrast normalization

Contrast normalization techniques help to bring an image contrast to a range more familiar to our visual senses. Several techniques exist that achieve such normalization, which will be described in following sections.

A.1 Histogram equalization

Contrast adjustment in image can be done using histogram equalization, and can reduce effects caused by low illumination, as showed in Fig. A.1. This technique works by increasing an image global contrast by better spreading out the pixel intensity values along the entire intensity range.

To achieve such effect, the first step relies on finding the image intensity histogram, H_i , using

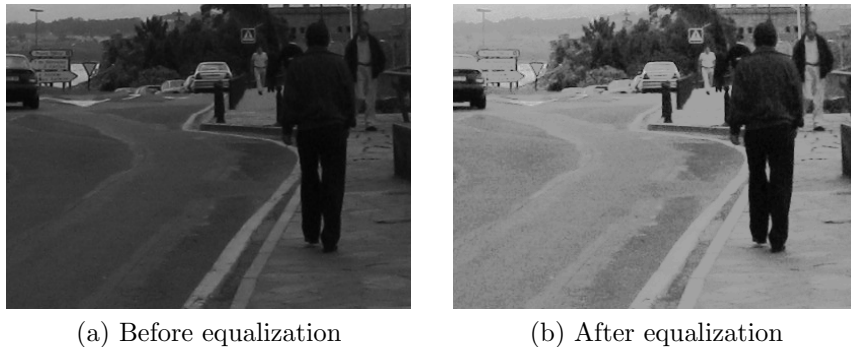


Figure A.1: Differences between the image before and after histogram equalization; the equalization was applied image-wide and the images presented are cropped around the object of interest. Original images from LabelMe [Russell *et al.* 2008]

$$\mathcal{H}_i = \sum_{y=1}^n \sum_{x=1}^m \left(\begin{cases} 1 & \text{if } i = I(x, y) \\ 0 & \text{otherwise} \end{cases} \right) \quad (\text{A.1})$$

where I is the input image. Then, the histogram is normalized to make the sum of all bins 255 with

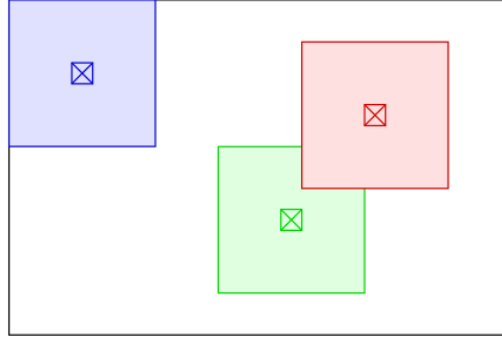


Figure A.2: Demonstration of how a pixel is contrast normalized based on its immediate neighborhood. Image from public domain.

$$\mathcal{H}'_i = \frac{H(i) * 255}{\sum_{n=1}^{255} \mathcal{H}(i)}$$

then, a cumulative sum of each bin is calculated with

$$\mathcal{H}''_i = \sum_{n=0}^i \mathcal{H}'(n)$$

to generate the contrast equalized image

$$I'_{x,y} = \mathcal{H}''(I(x,y)) \quad (\text{A.2})$$

Histogram equalization can also be applied to color images, however it cannot be directly applied in the RGB color channels.

A.2 Adaptive histogram equalization

Commonly, histogram equalization is applied over an entire image. However, this tends to not enhance contrast in regions which are significantly brighter or darker than the rest of the image. To solve this problem, the histogram equalization can be applied only in an image subset, over a local area surrounding a given pixel. As such, adaptive histogram equalization (AHE) allow for regions that are a statistically small portion of the image space to be properly contrast normalized [Ketcham *et al.* 1974].

Adaptive histogram equalization is very similar to a common histogram equalization. However, each pixel is contrast normalized based on a surrounding rectangular region. This calculation based on a pixel neighborhood is shown in Figure A.2. The size of the neighborhood controls the scale of normalization, smaller blocks increases contrast

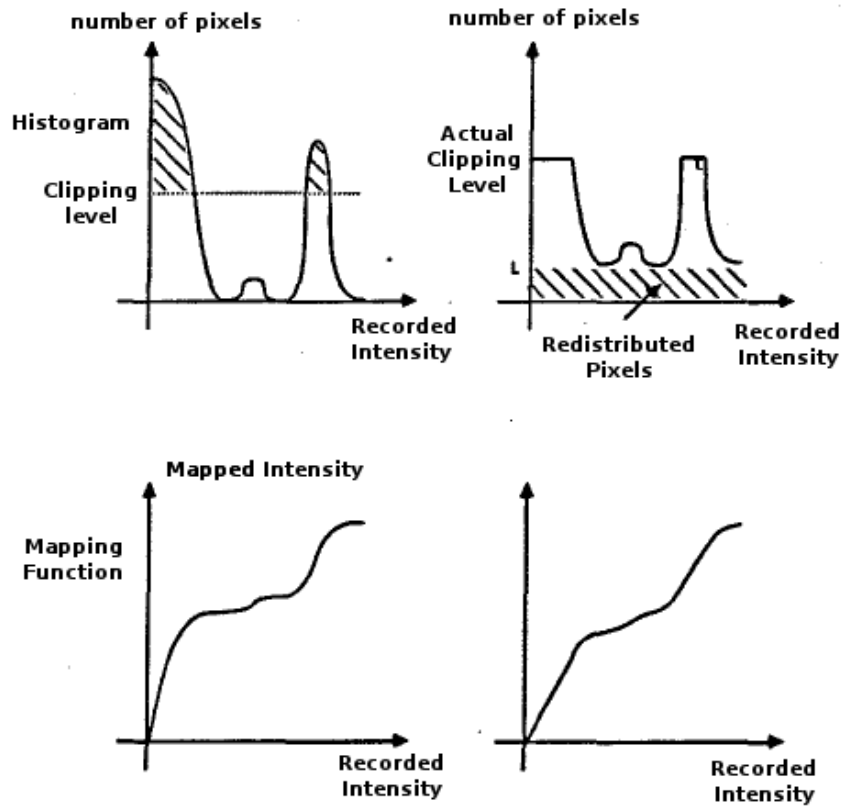


Figure A.3: Contrast mapping functions and each generated clipped histogram. Image adapted from [Pizer *et al.* 1987].

amplification of finer image details.

Calculating the neighborhood for each pixel in an image can be time consuming. To avoid such overhead a common optimization is to divide the image into a grid, then a given pixel contrast normalization can be calculated from interpolation between its four closest grid points. This allow for speed gains of over an order of magnitude when compared to a naive implementation [Pizer *et al.* 1987].

The main disadvantage of AHE is that, in case a pixel neighborhood is mostly homogeneous, it will over-amplify existing noise. To tackle this shortcoming, Contrast Limited Adaptive Histogram Equalization [Pizer *et al.* 1987] (CLAHE) was developed. The contrast limit is done by clipping the maximum height of the image intensity histogram.

The information which is clipped from histogram peaks is then redistributed equally along the entire histogram range. This is done to allow the entire input range to be mapped to the entire output range. After the redistribution, if any bin is over the contrast limit the process is repeated until no bin is over the limit. The impact of contrast limiting can

be seen in Figure A.3.

The contrast limit can be seen as controlling the slope between the mapping function of input intensity to output intensity [Pizer *et al.* 1987].

References

- [Achanta *et al.* 2008] R. Achanta, F. Estrada, P. Wils and S. Ssstrunk. *Salient region detection and segmentation*. Computer Vision Systems, pages 66–75, 2008.
- [Achanta *et al.* 2009] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada and Sabine Ssstrunk. *Frequency-tuned Salient Region Detection*. In IEEE International Conference on Computer Vision and Pattern Recognition, 2009.
- [Arajo *et al.* 2008] R. Arajo, U. Nunes, L. Oliveira, P. Sousa and P. Peixoto. *Support Vector Machines and Features for Environment Perception in Mobile Robotics*. Computational Intelligence Paradigms, pages 219–250, 2008.
- [Ballard & Brown 1982] Dana Ballard and Christopher Brown. Computer vision. Prentice Hall, first edition dition, 1982.
- [Belongie *et al.* 1998] S. Belongie, C. Carson, H. Greenspan and J. Malik. *Color-and texture-based image segmentation using EM and its application to content-based image retrieval*. In IEEE International Conference on Computer Vision, pages 675–682, 1998.
- [Bradski & Kaehler 2008] G. Bradski and A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. O’Reilly Media, 2008.
- [Chen *et al.* 1994] Q. Chen, M. Defrise and F. Deconinck. *Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 16, no. 12, pages 1156–1168, 1994.
- [Cheng *et al.* 2011a] M.M. Cheng, G.X. Zhang, N.J. Mitra, X. Huang and S.M. Hu. *Global contrast based salient region detection*. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 409–416. IEEE, 2011.
- [Cheng *et al.* 2011b] M.M. Cheng, G.X. Zhang, N.J. Mitra, X. Huang and S.M. Hu. *Global contrast based salient region detection*. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 409–416. IEEE, 2011.
- [Chia *et al.* 2011] A Y S Chia, S Zhuo, R K Gupta, Y W Tai, S Y Cho, P Tan and S Lin. *Semantic colorization with internet images*. ACM Transactions on Graphics (TOG), vol. 30, no. 6, page 156, 2011.
- [Dalal & Triggs 2005] N. Dalal and B. Triggs. *Histograms of oriented gradients for human detection*. In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 886–893. IEEE, 2005.
- [Dalal 2006] N. Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006.

- [Divvala *et al.* 2009] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros and M. Hebert. *An empirical study of context in object detection*. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 1271–1278. IEEE, 2009.
- [Ell & Sangwine 2007] T.A. Ell and S.J. Sangwine. *Hypercomplex Fourier transforms of color images*. IEEE Transactions on Image Processing, vol. 16, no. 1, pages 22–35, 2007.
- [Enzweiler & Gavrila 2009] M. Enzweiler and D.M. Gavrila. *Monocular Pedestrian Detection: Survey and Experiments*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 12, pages 2179–2195, 2009.
- [Everingham *et al.* 2012] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, 2012.
- [Felzenszwalb & Huttenlocher 2004] P.F. Felzenszwalb and D.P. Huttenlocher. *Efficient graph-based image segmentation*. International Journal of Computer Vision, vol. 59, no. 2, pages 167–181, 2004.
- [Feng *et al.* 2011] Jie Feng, Yichen Wei, Litian Tao, Chao Zhang and Jian Sun. *Salient object detection by composition*. In ICCV, pages 1028–1035, 2011.
- [Ferecatu & Geman 2007] M. Ferecatu and D. Geman. *Interactive search for image categories by mental matching*. In IEEE International Conference on Computer Vision, pages 1–8, 2007.
- [Freund & Schapire 1996] Y. Freund and R.E. Schapire. *Experiments with a new boosting algorithm*. In Machine Learning International Workshop, pages 148–156. Citeseer, 1996.
- [Freund *et al.* 1999] Y. Freund, R. Schapire and N. Abe. *A short introduction to boosting*. Japanese Society For Artificial Intelligence, vol. 14, no. 771-780, page 1612, 1999.
- [Goferman *et al.* 2011] Stas Goferman, Lihi Zelnik-Manor and Ayellet Tal. *Context-Aware Saliency Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, 2011.
- [Gomez & Morales 2002] G. Gomez and E. Morales. *Automatic feature construction and a simple rule induction algorithm for skin detection*. In ICML Workshop on Machine Learning in Computer Vision, pages 31–38. Citeseer, 2002.
- [Guo *et al.* 2008] Chenlei Guo, Qi Ma and Liming Zhang. *Spatio-temporal Saliency Detection Using Phase Spectrum of Quaternion Fourier Transform*. IEEE International Conference on Computer Vision and Pattern Recognition, no. 220, 2008.

- [Hall-Beyer 2007] Mryka Hall-Beyer. *GLCM Tutorial*, 2007. Available in <http://www.fp.ucalgary.ca/mhallbey/tutorial.htm>. Accessed in January 08, 2011.
- [Harel *et al.* 2007] Jonathan Harel, Christof Koch and Pietro Perona. *Graph-based visual saliency*. Advances in neural information processing systems, vol. 19, page 545, 2007.
- [Horner & Gianino 1984] J.L. Horner and P.D. Gianino. *Phase-only matched filtering*. Applied optics, vol. 23, no. 6, pages 812–816, 1984.
- [Hou & Zhang 2007] Xiaodi Hou and Liqing Zhang. *Saliency Detection: A Spectral Residual Approach*. In IEEE International Conference on Computer Vision and Pattern Recognition, pages 1–8, 2007.
- [Hou & Zhang 2008] X. Hou and L. Zhang. *Thumbnail generation based on global saliency*. Advances in Cognitive Neurodynamics, pages 999–1003, 2008.
- [Hsiao & Millane 2005] W.H. Hsiao and R.P. Millane. *Effects of occlusion, edges, and scaling on the power spectra of natural images*. Journal of the Optical Society of America: Optics, image science, and vision, vol. 22, no. 9, pages 1789–1797, 2005.
- [Hu *et al.* 2004] W. Hu, T. Tan, L. Wang and S. Maybank. *A survey on visual surveillance of object motion and behaviors*. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 34, no. 3, pages 334–352, 2004.
- [Ikizler-Cinbis *et al.* 2010] N. Ikizler-Cinbis, R.G. Cinbis and Sclaroff S. *Learning actions from the web*. In IEEE International Conference on Computer Vision, pages 995–1002, 2010.
- [Itti *et al.* 1998] L. Itti, C. Koch and E. Niebur. *A model of saliency-based visual attention for rapid scene analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, pages 1254–1259, 1998.
- [Joachims 1999] T. Joachims. *Making large scale SVM learning practical*. Advances in Kernel Methods - Support Vector Learning, 1999.
- [Judd *et al.* 2009] T. Judd, K. Ehinger, F. Durand and A. Torralba. *Learning to predict where humans look*. In IEEE International Conference on Computer Vision, pages 2106–2113. IEEE, 2009.
- [Kadir & Brady 2001] T. Kadir and M. Brady. *Saliency, scale and image description*. International Journal of Computer Vision, vol. 45, no. 2, pages 83–105, 2001.
- [Kastrinaki *et al.* 2003] V. Kastrinaki, M. Zervakis and K. Kalaitzakis. *A survey of video processing techniques for traffic applications*. Image and Vision Computing, vol. 21, no. 4, pages 359–381, 2003.

- [Ketcham *et al.* 1974] D.J. Ketcham, R.W. Lowe and J.W. Weber. *Image enhancement techniques for cockpit displays*. Rapport technique, DTIC Document, 1974.
- [Keysers *et al.* 2007] D. Keysers, T. Deselaers and T.M. Breuel. *Optimal geometric matching for patch-based object detection*. ELCVIA, vol. 6, no. 1, pages 44–54, 2007.
- [Lampert *et al.* 2008] Christoph H. Lampert, Matthew B. Blaschko and Thomas Hofmann. *Beyond sliding windows: Object localization by efficient subwindow search*. IEEE International Conference on Computer Vision and Pattern Recognition, pages 1–8, June 2008.
- [Lienhart & Maydt 2002] R. Lienhart and J. Maydt. *An extended set of Haar-like features for rapid object detection*. IEEE International Conference on Image Processing, pages 900–903, 2002.
- [Liu *et al.* 2011] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaou Tang and Heung-Yeung Shum. *Learning to detect a salient object*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 2, pages 353–67, 2011.
- [Lowe 1999] D.G. Lowe. *Object recognition from local scale-invariant features*. In IEEE International Conference on Computer Vision, volume 2, pages 1150–1157, 1999.
- [Oliva & Torralba 2001] A. Oliva and A. Torralba. *Modeling the shape of the scene: A holistic representation of the spatial envelope*. International Journal of Computer Vision, vol. 42, no. 3, pages 145–175, 2001.
- [Oliveira 2010] Luciano Oliveira. *Semantically Integrating Laser and Vision in Pedestrian Detection*. PhD thesis, University of Coimbra, Department of Electrical and Computer Engineering, 2010.
- [Oppenheim & Lim 1981] Alan V Oppenheim and Jae S Lim. *The importance of phase in signals*. Proceedings of the IEEE, vol. 69, no. 5, pages 529–541, 1981.
- [Overett *et al.* 2008] Gary Overett, Lars Petersson, Nathan Brewer, Lars Andersson and Niklas Pettersson. *A new pedestrian dataset for supervised learning*. IEEE Intelligent Vehicles Symposium, pages 373–378, June 2008.
- [Papageorgiou *et al.* 1998] C.P. Papageorgiou, M. Oren and T. Poggio. *A general framework for object detection*. In IEEE International Conference on Computer Vision, pages 555–562, 1998.
- [Peer *et al.* 2003] P. Peer, J. Kovac and F. Solina. *Human skin colour clustering for face detection*. EUROCON International Conference on Computer as a Tool, 2003.

- [Perko & Leonardis 2007] R. Perko and A. Leonardis. *Context driven focus of attention for object detection*. Attention in Cognitive Systems. Theories and Systems from an Interdisciplinary Viewpoint, pages 216–233, 2007.
- [PETS 2006] PETS. *Performance Evaluation of Tracking and Surveillance*. <http://www.cvg.rdg.ac.uk/PETS2006/data.html>, 2006.
- [Pizer *et al.* 1987] S.M. Pizer, E.P. Amburn, J.D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J.B. Zimmerman and K. Zuiderveld. *Adaptive histogram equalization and its variations*. Computer vision, graphics, and image processing, vol. 39, no. 3, pages 355–368, 1987.
- [Prisacariu & Reid 2009] Victor Prisacariu and Ian Reid. *fastHOG - a real-time GPU implementation of HOG*. Rapport technique 2310/09, Department of Engineering Science, Oxford University, 2009.
- [Rensink 2000] R.A. Rensink. *Seeing, sensing, and scrutinizing*. Vision research, vol. 40, no. 10-12, pages 1469–1487, 2000.
- [Ruderman & Bialek 1994] D.L. Ruderman and W. Bialek. *Statistics of natural images: Scaling in the woods*. Physical Review Letters, vol. 73, no. 6, pages 814–817, 1994.
- [Ruderman 1994] D.L. Ruderman. *The statistics of natural images*. Network: computation in neural systems, vol. 5, no. 4, pages 517–548, 1994.
- [Russell *et al.* 2008] B.C. Russell, A. Torralba, K.P. Murphy and W.T. Freeman. *LabelMe: a database and web-based tool for image annotation*. International Journal of Computer Vision, vol. 77, no. 1, pages 157–173, 2008.
- [Rutishauser *et al.* 2004] U. Rutishauser, D. Walther, C. Koch and P. Perona. *Is bottom-up attention useful for object recognition?* In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–37. IEEE, 2004.
- [Schneiderman & Kanade 1998] Henry Schneiderman and Takeo Kanade. *Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98), pages 45–51, July 1998.
- [Schneiderman & Kanade 2000] Henry Schneiderman and Takeo Kanade. *A Statistical Model for 3D Object Detection Applied to Faces and Cars*. In IEEE Conference on Computer Vision and Pattern Recognition. IEEE, June 2000.
- [Schroff *et al.* 2007] F. Schroff, A. Criminisi and A. Zisserman. *Harvesting image databases from the web*. In IEEE International Conference on Computer Vision, pages 1–8, 2007.

- [Silva *et al.* 2012] Grimaldo Silva, Leizer Schnitman and Luciano Oliveira. *Multi-Scale Spectral Residual Analysis to Speed up Image Object Detection*. In Conference on Graphics, Patterns and Images, 2012.
- [Simoncelli & Freeman 1995] E.P. Simoncelli and W.T. Freeman. *The steerable pyramid: A flexible architecture for multi-scale derivative computation*. In International Conference on Image Processing, volume 3, pages 444–447. IEEE, 1995.
- [Sorokin & Forsyth 2008] A. Sorokin and D. A. Forsyth. *Utility data annotation with Amazon Mechanical Turk*. In Internet Vision, pages 1–8, 2008.
- [Strat 1993] T.M. Strat. *Employing contextual information in computer vision*. DARPA93, pages 217–229, 1993.
- [Vapnik 1999] V. Vapnik. *The nature of statistical learning theory*. springer, 1999.
- [Vezhnevets *et al.* 2003] V. Vezhnevets, Sazonov. V. and Andreeva. A. *A survey on pixel-based skin color detection techniques*. In Graphicon, volume 3. Citeseer, 2003.
- [Viola & Jones 2001] P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. IEEE International Conference on Computer Vision and Pattern Recognition, pages 511–518, 2001.
- [Wolf & Bileschi 2006] L. Wolf and S. Bileschi. *A critical view of context*. International Journal of Computer Vision, vol. 69, no. 2, pages 251–261, 2006.
- [Yiu & Varshney 2011] Cheuk Ip Yiu and Amitabh Varshney. *Saliency-Assisted Navigation of Very Large Landscape Images*. IEEE Transactions on Visualization and Computer Graphics, vol. 17, pages 1737–1746, 2011.
- [Zhai & Shah 2006] Yun Zhai and M. Shah. *Visual Attention Detection in Video Sequences Using Spatiotemporal Cues Categories and Subject Descriptors*. In Proceedings of the 14th annual ACM international conference on Multimedia, volume 32816, pages 815–824, 2006.
- [Zhu *et al.* 2006] Q. Zhu, M.C. Yeh, K.T. Cheng and S. Avidan. *Fast human detection using a cascade of histograms of oriented gradients*. In IEEE International Conference on Computer Vision and Pattern Recognition, volume 2, pages 1491–1498, 2006.

Multi-Scale Spectral Residual Analysis to Speed up Image Object Detection

Grimaldo Silva, Leizer Schnitman, Luciano Oliveira
Programme of Post-graduation in Mechatronics
Intelligent Vision Research Laboratory, UFBA
{jgrimaldo, leizer, Irebouca}@ufba.br

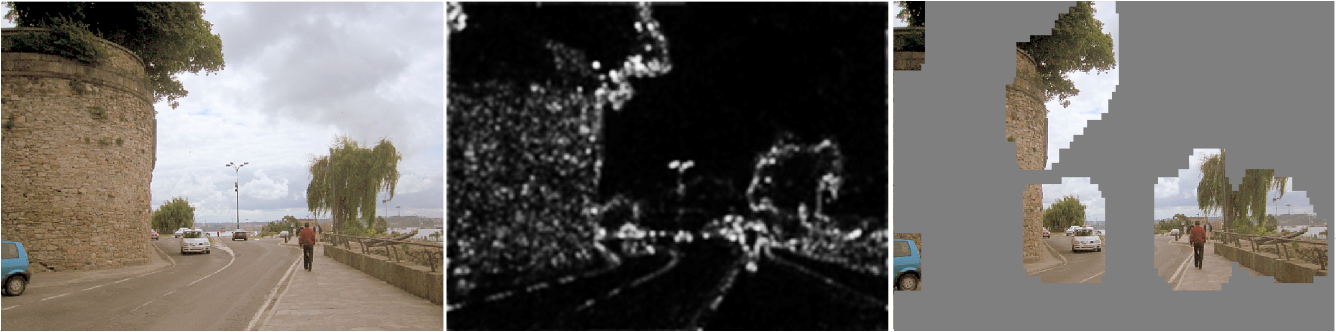


Fig. 1. From left to right: original image, saliency map, candidate regions in the saliency map. A very usual approach to search for an image object is sliding window, which performs a dense search in image space. By using a multi-scale saliency map, we are able to tease out image regions which are likely unnecessary for object search when sliding image windows. After that, a detector can be attached to only selected regions, allowing faster object detectors.

Abstract—Accuracy in image object detection has been usually achieved at the expense of much computational load. Therefore a trade-off between detection performance and fast execution commonly represents the ultimate goal of an object detector in real life applications. In this present work, we propose a novel method toward that goal. The proposed method was grounded on a multi-scale spectral residual (MSR) analysis for saliency detection. Compared to a regular sliding window search over the images, in our experiments, MSR was able to reduce by 75% (in average) the number of windows to be evaluated by an object detector. The proposed method was thoroughly evaluated over a subset of LabelMe dataset (person images), improving detection performance in most cases.

Keywords—multi-scale spectral residue, saliency, person detection

I. INTRODUCTION

Image object localization has been reaching remarkable results in real life applications. However, the more accurate is the method, the heavier it is with respect to computational cost. Achieving the best trade-off between detection performance and computational cost usually represents a challenging task. Indeed, in many practical situations, object detection requires on-the-fly execution in order to be feasible in practice. Among these time-critical tasks, there are: perception for driver assistance [1], video traffic analysis [2] and surveillance systems [3]. If we still consider the current availability of high resolution images, which demands additional processing time, the mentioned trade-off presents an even bigger challenge.

To cope with the aforementioned trade-off problem, many methods have been proposed. Zhu et al. [4] and Viola and Jones [5] have developed rejection cascades, reducing the time required to detect non-objects. These works were based on the so called sliding window search. Toward methods to avoid or to reduce the overhead of a dense search, saliency detectors have demonstrated promising results. As saliency detectors are able to locate regions of interest in images, they can be used in a broad spectrum of applications – from thumbnail generation [6] to semantic colorization [7]. Examples of such saliency methods are found in [8], which uses statistical properties of natural scenes to select regions of interest, and also in [9] based on the computation of saliency inspired on the pre-attentive phase of human visual system, responsible for drawing attention to specific parts of the visual stimuli.

The positive traits of saliency methods on search space reduction allowed Ip et al. [10] to make a saliency analysis in very large images in order to assist human visualization by means of possible regions of interest (ROI). ROI are found through a difference of Gaussians at multiple image scales¹. Likewise, Rutishauser et al. [11] proposed an object recognition (among grocery items) based on the saliency method found in [9] and a scale invariant feature transform (SIFT) keypoint detector [12]. First, the saliency method is applied to determine the most likely areas to have an object; instead of thresholding the saliency map generated in the first

¹Throughout the text, the words ‘octave’ and ‘scale’ are used interchangeably.

step, a region growing segmentation defines the best object hypothesis; at the end, image object silhouette is delineated by means of the keypoints detected over the salient areas. Feng et al. [13] address the problem of object detection using a sliding window over an image, specifying each window saliency as the cost of composing it with remaining parts of the image; therefore the image is segmented into regions based on similarity; the difference between regions is calculated over LAB histograms and spatial distances; these features are then used to select the most differentiated windows which hopefully present the most salient objects.

On the reduction of image search space, Lampert et al. [14] propose the use of a branch-and-bound optimization applied on the score of the classifier, which is used to separate input space. The method was called Efficient Subwindow Search (ESS). The target function is subjected to maximize the classification score whereas minimizing the number of windows evaluated by a detector. In its original form, that method only detects one object per image, but it can be modified to search for multiple objects. ESS effectively reduces the number of evaluated windows over the image in contrast to regular sliding window based detectors [5], [15], [16].

Following all these ideas, the multi-scale spectral residue (MSR) analysis aims to speed up sliding window-based object detection by spectral residual analysis on multiple scales. Our method relies on a sliding window approach based on the image saliency with the goal of assigning a score to each window before object detection stage (see Fig.). Although Feng et al. [13] also assign a saliency score to each window, our approach has some important differences. MSR computes an image-wise saliency following the rationale in [8], in a more flexible way, allowing saliency detection in the original image aspect ratio. Additionally, we explore properties of the frequency domain to extract interesting regions in contrast to the use of spatial properties such as composability of segments. MSR differs from ESS in the requirements and methodology. ESS avoids a dense detector search by using an optimization method that requires a linear classifier and local image descriptors such as [12]. MSR does not impose such constraints, and can be used on most sliding window based detectors by relying solely on an object saliency. Our approach also avoids assumptions about an object shape to reduce the search space, as such, it does not attempt to segment an object based on salient locations, as in Rutishauser et al. [11]; instead, MSR indicates regions of interest and relies on a classifier for actual object detection and localization. Recent solutions of rejection cascades [4] [5] in a sliding window search can easily be integrated to MSR. This latter can be combined with MSR in order to achieve faster processing time.

This work is structured as follows: an overview of saliency detection methods is given in Section II. Section III describes MSR and a methodology to evaluate the impact of window selection on detector performance. In Section IV, the MSR is compared to other saliency methods, and its runtime and detection performance are measured over a person dataset. Finally, overall conclusions are drawn in Section V.

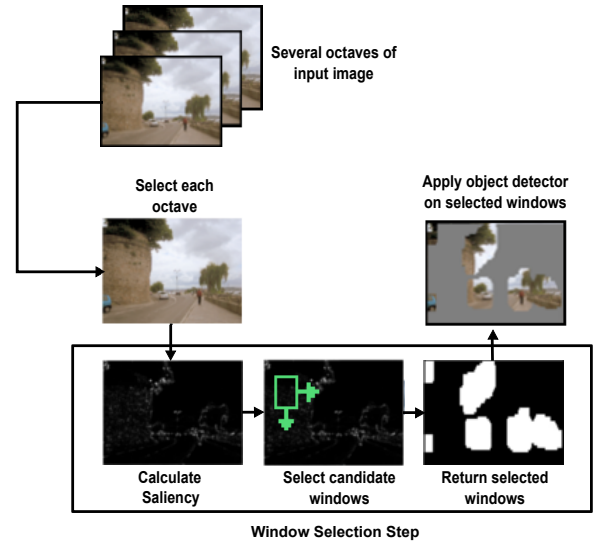


Fig. 2. Overview of MSR. For each octave of the original image, the saliency map is computed, and a sliding window is applied on the saliency map. Candidate windows are selected according to their scores given by a quality function. Finally, an object detector is applied only in the candidate windows.

Contributions: Our contribution resides in a novel method, called MSR, with the aim of achieving a better trade-off between the number of windows selected to be evaluated by a detector, and the number of miss detections. MSR has demonstrated an average reduction of 75% of windows to be evaluated, while keeping or improving detection performance.

A. Proposed method at a glance

When performing a dense search for an object, only a small subset of the image might contain objects. However, sliding window based detectors are only able to provide image object localization after running a classification function over each window on multiple orientations and octaves (scales). For that, the use of a full-fledged object detector implies an expensive operation, requiring preprocessing, feature extraction and classification. In order to reduce the number of windows which will be evaluated by a detector, we propose a bottom-up saliency approach to select windows of interest before running the detector in each window. Although MSR has been motivated by [8], it was conceived to overcome some limitations of that method when used on uncontrolled scenes. These improvements are listed below:

- 1) resizing each image octave by a constant resizing factor – 15% of its size, instead of making assumptions about object scale by using a fixed image size for saliency detection. This change allows search of salient objects at multiple scales;
- 2) choice of threshold k for region selection is not dependent on each image saliency map, but on a constant global value based on a trade-off between selected regions and false negatives (FN) in the classification. In [8], the threshold is calculated as $k = 3 \cdot E(S(x))$, or three times the mean saliency map $S(x)$ intensity. How-

ever, this latter formulation incorrectly regards objects in cluttered images (many objects) as non-salient.

- 3) saliency quality in a region is calculated from a window-wise saliency mean, instead of using pixel values individually as in [8], allowing detection of entire objects even when their saliency is non-uniform along its length.

Instead of relying on the object detector to choose the most likely image region to contain an object (just after obtaining the saliency map), windows are slid over an integral saliency space. This latter step corresponds to computing the integral image of the pixels in the saliency space in the same way as Viola and Jones [5]. After that, a quality function $f(\cdot)$ is applied at each window w , providing a score. The score of a given window is calculated using the mean of its saliency intensity, and a window is selected if its score is greater than or equal to a threshold k . A higher k selects smaller number of windows, while potentially missing more true positive (TP) detections in the further steps of the method. Conversely, as value of k gets lower, MSR approaches to a method based on regular sliding window search. An overview of MSR mechanism for window selection is summarized in Fig. 2.

II. OVERVIEW OF SALIENCY DETECTION APPROACHES

An object draws more attention when it has a strong contrast in relation to its neighbourhood, objects such as traffic signs or a stop light were created to explore this property in order to be perceived faster than surrounding objects. While an attention mechanism can help a person focus on specific objects in a scene, in a similar way, an algorithm capable of detecting salient objects in images must search for characteristics such as visual uniqueness, rarity and unpredictability [17]. This is so in order to correctly highlight image regions which demand extra attention. Following these ideas, we briefly summarize some of saliency detectors:

Itti's method (IT): Among the first salient methods, a biologically inspired approach was developed by Itti et al. [9]. In that approach, saliency of a given pixel is calculated based on its uniqueness in relation to local surroundings. Uniqueness is defined on the analysis of color, intensity and orientation over multiple scales. After that, these features are then normalized and combined in a way where channels with larger contrasts are preferred.

Graph based (GB) visual saliency: Similarly to Itti, Harel et al. [18] form activation maps from particular feature channels, and normalize them to better highlight salient regions.

Frequency tuned (FT) saliency region detection: Instead of using local information to define the saliency, Achanta et al. [19] define saliency of a pixel as its distance from the image pixel mean on LAB space, formally represented as

$$S_a(x, y) = \|\mathbf{I}_\pi - \mathbf{I}(x, y)\|_2, \quad (1)$$

where \mathbf{I}_π is the mean image feature vector, $\mathbf{I}(x, y)$ is the original pixel value, $\|\cdot\|_2$ represents an L_2 norm where each pixel is a feature vector of type $[L, a, b]$.

Luminance contrast (LC): Also using global contrast, Zhai and Shah [20] developed a method for pixel-level saliency detection using the contrast of a pixel with respect to the others in a scene. It is given by

$$S_z(I_k) = \sum_{\forall I_i \in I} \|I_k - I_i\|, \quad (2)$$

where I_i and I_k are pixels in the image and $\|\cdot\|$ represents the Euclidean distance.

Spectral residual (SR): Similar to global methods, frequency based approaches also explore properties of the entire image. Hou and Zhang [8] used these properties based on $1/f$'s law, which states that an ensemble of images on the Fourier Spectrum obeys the distribution

$$E\{A(f)\} \propto 1/f, \quad (3)$$

where $A(f)$ is the amplitude averaged over orientations, and f is a given spectrum in the frequency domain. Whilst objects do not follow properties of natural scenes, detection of potential salient points is based on finding statistical singularities on the spectrum of an image. These singularities are called spectral residues.

III. PRUNING WINDOWS BY MULTI-SCALE SPECTRAL RESIDUE

Saliency detectors are able to associate a degree of local or global uniqueness for each image pixel (or group of pixels). This information is useful to help pruning undesired windows. In this regard, during a search for objects via sliding windows, the capability to choose whether a detector will evaluate a particular window or ignore it (based on its object likelihood) can bring benefits to speed up the classification task in further steps.

Saliency detectors face additional complexities when dealing with uncontrolled scenes, such as variations in object (color, size, illumination and noise). Particularly, it is noteworthy that spectral residual (SR) analysis [8] is susceptible to those factors when selecting image ROI, since an object may have intense intra-variability. To avoid that, in MSR, saliency is measured in a per-window basis, and the saliency of a window is defined as the mean intensity of its salient pixels, enabling higher resilience to variability of salient pixels.

Another limitation of SR in the context of aiding object detectors is its threshold for region selection, defined as $k = 3 * E(S(x))$ where $E(S(x))$ denotes the mean value of the saliency map and k the threshold. Such scheme expects that images have but a small number of salient regions. If it is not the case, that method potentially excludes important objects because of the high lower bound. Given that situation, we define the threshold k as a constant value throughout the entire collection of images, representing an average trade-off between the number of selected windows and false negatives (FN) caused by window selection.

From the aforementioned improvements, summarized on Fig. 3, the underlying concepts required for multi-scale analysis have been conceived in Section III-A.

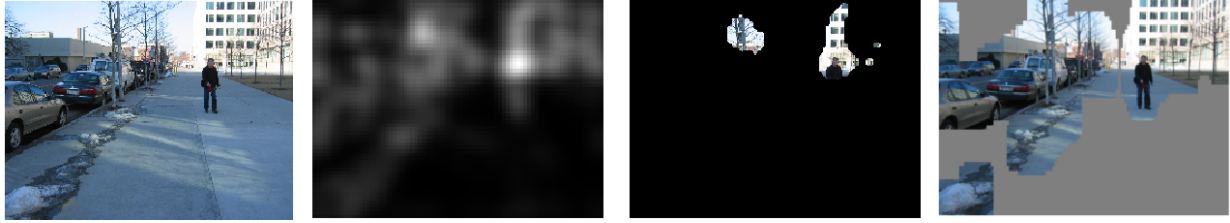


Fig. 3. Comparison between SR and MSR. From left to right: the original image, SR saliency map, region selection using SR formulation in original image, and MSR window selection at a particular octave.

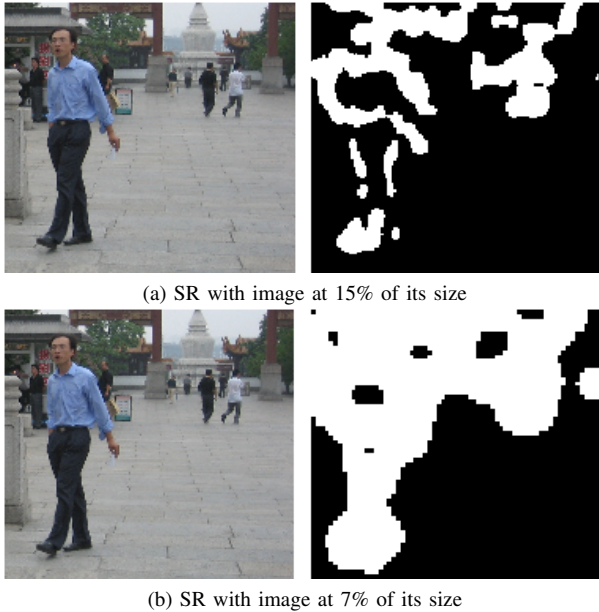


Fig. 4. Differences in saliency at multiple scales. In 4a, SR was calculated in 15% of the original image size, generating strong reactions on mostly small objects; in 4b, using 7% of the original image size, bigger objects were also selected. The image reduction examples demonstrate how the image size influences on the scale of saliency detection, which will be tuned to best select objects in a given octave.

A. Multi-scaling the spectral residue

Most saliency methods are able to detect objects of different sizes. Methods such as [9] and [18] make direct use of feature analysis at multiple image scales to achieve that result. In contrast, SR searches objects at a single scale, which is specified based on an estimation of common object sizes over normal visual conditions [8]. For that, SR cannot be used in an uncontrolled multi-scale environment, as the saliency detector will not search for objects at the same scale as the object detector. Because of that, it was necessary to establish a connection between the search scale of the object detector and the saliency detector.

The scale of salient objects in SR is implicitly defined by the image size. Therefore smaller objects are more salient on bigger images, because the smaller an image gets, the bigger are the objects that become salient, as depicted on Fig. 4. In this case, searching for salient objects with various sizes has a strong relation to how a sliding-window based object detector

searches for bigger objects in an image using a fixed size window. This search is accomplished by resizing an image at a fixed compound rate, such as $I_{i+1} = R(I_i, s)$, where R represents the resize function, I_i denotes the i -th image octave and s the resizing factor; the detector thus slides the detection window over each octave i .

As we focus on detection of saliency and objects within the same search scale, using a fixed-size window, we may conclude that from a particular octave I_i , there is a constant resizing factor β capable of adjusting the two detectors to the same scale. Given a value of β , saliency detection will be executed on each octave i over a reduced image, $R(I_i, \beta)$, with its color histogram normalized. This histogram normalization is applied to increase object contrast, enhancing the overall saliency of the object against the scene. Another practical use of further resizing the image using β is to reduce the computational load of saliency calculation. Defining a specific value for β will depend on factors such as: object of interest, scale of search and saliency detector. A β value of 0.15 was chosen based on experimental data. The choice of this value is discussed in detail in Section III-C.

After obtaining the image octaves, and consequently the generated saliency maps for each octave, a quality value $f(w)$ for each window w was calculated from its mean saliency intensity. To speed up mean computation, the quality value $f(w)$ is calculated after computing the integral image of the saliency map (having then mean calculation with constant time complexity).

B. Determining the quality function threshold

Proper evaluation of window selection impact on performance was done by means of an analysis of the window selection rate (WSR) and saliency false negative rate (SFNR). WSR denotes the number of windows selected for further processing, while SFNR represents how many objects the detector failed to recognize after MSR pruning.

Both WSR and SFNR depend on a threshold k which represents a minimum score for a window to be selected for actual object detection. Thus, given that W is the set of all windows generated from sliding on the entire collection of images at every scale and M the set of all objects of interest from this same collection of images, we can calculate the trade-off between WSR_k and $SFNR_k$ in a five-step process. First, we define the set of selected windows S_k as

$$S_k = \{w \in W \mid f(w) \geq k\}, \quad (4)$$

where $f(w)$ is the quality value of a window w and k is the threshold for window selection. Given S_k , it is possible to calculate the window selection rate with

$$\text{WSR}_k = \frac{n(S_k)}{n(W)}, \quad (5)$$

where $n(\cdot)$ denotes cardinality of a set. To calculate the SFNR_k one should enumerate for each object $j \in M$ the number of windows in which the object was correctly matched, given by

$$C_{k,j} = \{w \in S_k \mid o(w) = j\}, \quad (6)$$

where $o(w)$ is a function that, in case an object exists at window w , and this is correctly classified by a detector, returns the matched object from set M ; otherwise $o(w)$ returns any element $\notin M$. From that, it's trivial to find the set of objects detected, F_k , defined as

$$F_k = \{j \in M \mid n(C_{k,j}) \geq 1\}. \quad (7)$$

Finally, in order to calculate how many miss detections were caused by the saliency (SFNR), we use

$$\text{SFNR}_k = \frac{n(F_{k_{\min}}) - n(F_k)}{n(F_{k_{\min}})}, \quad (8)$$

where k_{\min} is the minimum threshold value, which guarantees $S_{k_{\min}} = W$. Thus, to generate a full trade-off curve, this process is repeated for each $k \in K$ where K is the set of unique window scores.

C. Parameter choice

The proper choice of value for β will change according to the scale and characteristics of a given object. For person detection, the best value for β was found to be 0.15. This was achieved over the LabelMe [21] dataset for persons (see Section IV-A for more detail). Figure 5 shows the trade-off of WSR and SFNR for different values of β .

A possible consideration is to use the parameter β only in the original image (full resolution). It would save processing time dedicated for calculation of the saliency at each scale. However, multi-scale methods had dominant superior performance in our tests, as can be noted in Fig. 6.

Henceforth, to facilitate result analysis, we focus on the operating points of 20% and 30% of WSR. The choice of these operating points intends to evaluate a preferable runtime performance (20% of WSR) in spite of detection performance, or to keep detection performance (30% of WSR) with acceptable speed gains.

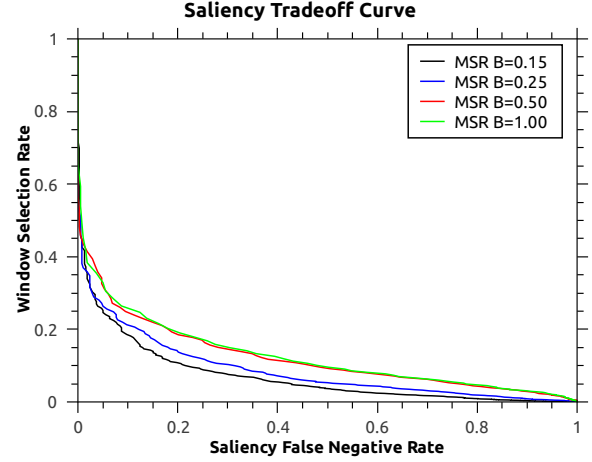


Fig. 5. Trade-off curve for person detection using different β values. When the curve is closer to the origin it is better.

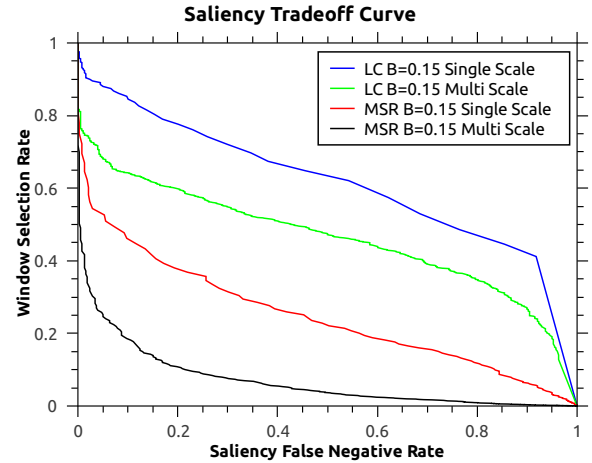


Fig. 6. Comparison between multi-scale analysis and using the same saliency map for all scales. Methods presented are MSR and LC [20]. When the curve is closer to the origin it is better.

IV. EXPERIMENTAL EVALUATION

A. Methodology

Evaluation of MSR was accomplished by a four-step analysis: (i) comparing the detection performance considering several saliency detection methods using the same sliding window parametrization in a multi-scale analysis; (ii) analysing MSR scalability with respect to detection, i.e., how it behaves on different image resolutions; (iii) how MSR affects a detector receiver operating characteristic (ROC) curve with respect to a regular sliding window, and, finally, (iv) impact on detection runtime speed with different parameters.

To standardize comparisons, a set of 330 images was extracted from the LabelMe [21] dataset. Image sizes range from 320 by 240 to 2592 by 1944. Additionally, the dataset encompasses several environments, including city, snow, forest and river, where each scene contains at least one person.

For all analyses using the aforementioned dataset, a com-

TABLE I
SFNR FOR EACH METHOD AT 20% OF WSR

| Method | β | | | |
|--------|---------|--------|--------|--------|
| | 0.15 | 0.25 | 0.50 | 1.00 |
| MSR | 08.73% | 11.38% | 17.99% | 17.99% |
| LC | 93.92% | 93.92% | 94.18% | 94.43% |
| FT | 87.83% | 82.54% | 76.46% | 77.25% |
| GB | - | - | - | 47.09% |
| IT | - | - | - | 38.89% |

bination of histogram of oriented gradients (HOG) [16] and Support Vector Machine (SVM) was used as the method to classify persons. The rationale of using HOG/SVM was not only because it is a state-of-the-art detector, but also to facilitate comparison with other future search reduction methods, since its source code is publicly available. Our HOG/SVM detector was trained using a person dataset distinct from the one created with images from LabelMe. Additionally, for the sliding window, the detector was set up with window size of 64 by 128 pixels, a stride of 8 pixels horizontal-wise, and 16 pixels vertical-wise, and image resizing rate of 0.96, for each octave.

It is noteworthy that, for analysis (i), two state-of-the-art saliency methods have not been included – [22] and [23]. The former, because the saliency detection is concentrated mostly on images with a single and clear salient object; the latter, because of its very slow runtime speed. In (ii), we examine how MSR performance changes over different image resolutions and how each image octave contributes to its results. This experiment is important to considering recent increases in availability of high resolution images. In (iii), we built a ROC curve to show the effects of MSR on a person detector at different WSR configurations in comparison to a normal sliding window. Finally, in (iv), we evaluate if the number of windows discarded before detection is sufficient to compensate for the additional processing required by MSR.

B. Comparison of saliency methods in a multi-scale structure

A comparison of MSR against other state-of-the-art methods in the same multi-scale structure is presented in Fig. 7. As the results represent only the best configuration of each method, a more detailed information is organized on Table I and II.

In these experiments, both IT [9] and GB [18] were only evaluated using the original octave size, with no β scaling, as these methods already perform analysis at multiple scales internally. We also did not compare MSR with the original SR [8] since the latter one was designed to operate on a single image size.

The results indicate that MSR achieved superior performance on almost the entire trade-off curve. In addition, SFNR at 20% and 30% of WSR were at least ten times better when compared to other methods. Yet, at 50% of WSR, the SFNR is close to zero (less than 0.3%), which indicates that a detector could process images twice as fast with a negligible loss in TP.

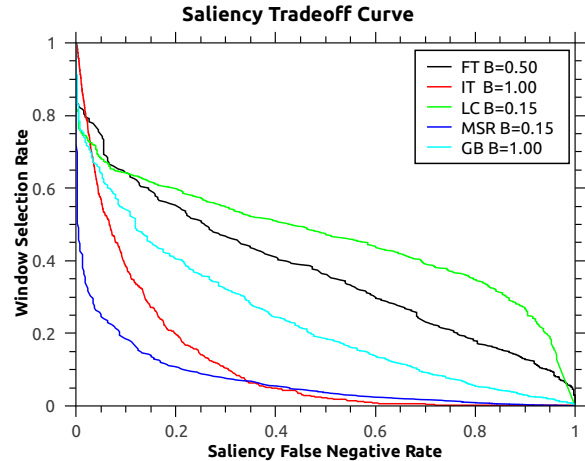


Fig. 7. Comparison of best results from different saliency methods applied to guide multi-scale detectors, the methods are MSR, FT [19], GB [18], IT [9], LC [20]. The false negative rate represents only objects that the detector would have matched if a regular sliding window approach had been used. When the curve is closer to the origin it is better.

TABLE II
SFNR FOR EACH METHOD AT 30% OF WSR

| Method | β | | | |
|--------|---------|--------|--------|--------|
| | 0.15 | 0.25 | 0.50 | 1.00 |
| MSR | 02.91% | 02.64% | 06.08% | 05.55% |
| LC | 86.24% | 86.24% | 85.98% | 87.04% |
| FT | 70.37% | 64.29% | 59.79% | 62.69% |
| GB | - | - | - | 33.86% |
| IT | - | - | - | 23.81% |

The MSR and IT methods had the best overall trade-off between WSR and SFNR. Both SR and IT were among the worst on recent evaluation of general purpose saliency detection found in [17]. Some possible causes of this discrepancy are:

- 1) saliency detection is done in a single scale in [17], while in our tests the saliency was recalculated at every octave. This provided a better performance for most methods;
- 2) differences in scene selection. The dataset used in [17] was gathered by [19] with images containing mostly uncluttered objects and natural background. In our tests, images were extracted from LabelMe [21], wherein the images contain a wide range of locations and varying degrees of clutter;
- 3) little background information on some images (large objects).

C. Scalability

We compare how well MSR can select image windows at different starting image resolutions in Fig. 8. From this information, we can conclude that increasing image size allows for an even better trade-off between SFNR and WSR.

To further confirm the scalability of MSR on larger resolutions, we compare its ability to eliminate windows at a fixed threshold in several octaves in Fig. 9, showing that

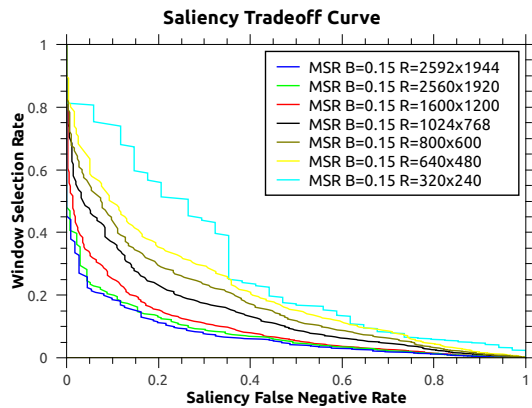


Fig. 8. Trade-off between WSR and SFNR at different starting resolutions. Aspect ratio is kept by approximating the image resolution to the closest image size. When the curve is closer to the origin it is better.

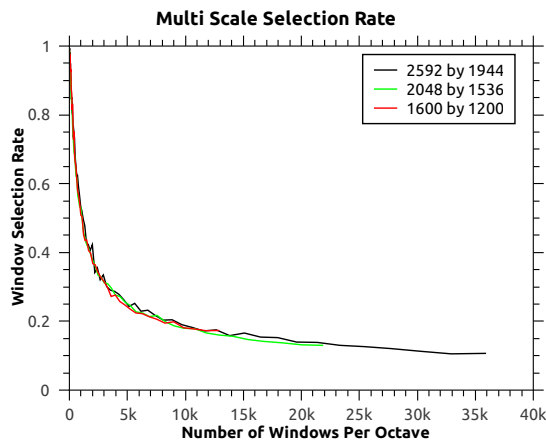


Fig. 9. Relation between number of windows at each image size and number of windows selected for the detector. An operating point was selected at 20% of WSR (see Fig. 7 for reference).

larger size images contribute for better WSR. In this test, the number of windows in each octave is calculated with $1 + [(w_i - w_w)/s_h] * [(h_i - h_w)/s_v]$, where w_i and h_i are the image width and height, respectively, w_w and h_w are the window width and height, s_h is the horizontal stride and s_v is the vertical stride.

D. Detection performance

Comparison between an object detector with and without MSR is presented on Fig. 10. In the tests, MSR at 20% of WSR provided greater TPR than regular sliding window within the range of 0 and 1.48 of FPPI. At 30% of WSR and within its range of 0 and 1.98 of FPPI, our method also obtained larger TPR than a regular sliding window approach. The maximum TPR of a regular sliding window is 0.71, while for MSR at 30% of WSR the maximum is 0.69. Even though the difference was small to match the actual maximum TPR of a regular sliding window, MSR operated at least on 50% of WSR, which still represents a twice as fast image processing with only a negligible performance loss (less than 0.3%).

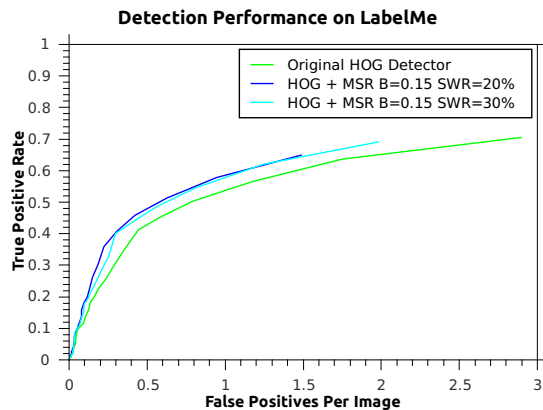


Fig. 10. ROC curve showing differences between person detection performance using a regular sliding window and MSR. Both methods use the same HOG+SVM detector.

Some examples of positive and negative results at 30% of WSR can be found, respectively, on Fig. 11 and Fig. 12.

E. Runtime performance

In order to evaluate MSR runtime speed, a comparison was performed with the traditional sliding window HOG detector. We summarized the results on Table III. Time was calculated as the proportion of the total detection time for a specific WSR value of a regular sliding window execution.

Expected gain, considering elimination of 80% and 70% of windows to be classified, was 5x and 3.3x. However, the results demonstrated that for both 19.9% and 29.6% of SWR², the actual runtime speed gain was smaller than 4.8x and 3.2x. This indicates that MSR window selection mechanism imposed only a small processing overhead for each window, which was compensated by the large number of windows discarded.

TABLE III
RUNTIME SPEED PROPORTION FOR EACH METHOD

| Method | WSR | Total Time Proportion | Avg. Time Proportion Per Window |
|--------------------|-------|-----------------------|---------------------------------|
| Regular Slide | 100% | 1.0000 | 1.0000 |
| MSR $\beta = 0.15$ | 19.9% | 0.1932 | 0.1996 |
| MSR $\beta = 0.15$ | 29.6% | 0.2852 | 0.2994 |

V. CONCLUSION

This work presented a method to speed up sliding window-based object detectors by multi-scale spectral residual analysis, named MSR. This way, MSR avoids using a full-fledged object detector on windows unlikely to contain objects, speeding up detection. In our experiments, MSR was able to provide better or similar detection performance, and faster detection with scalability to increasing image resolutions. Furthermore, our

²The closest thresholds to 20% and 30% of SWR, respectively. Equivalent to 80% and 70% of window elimination.

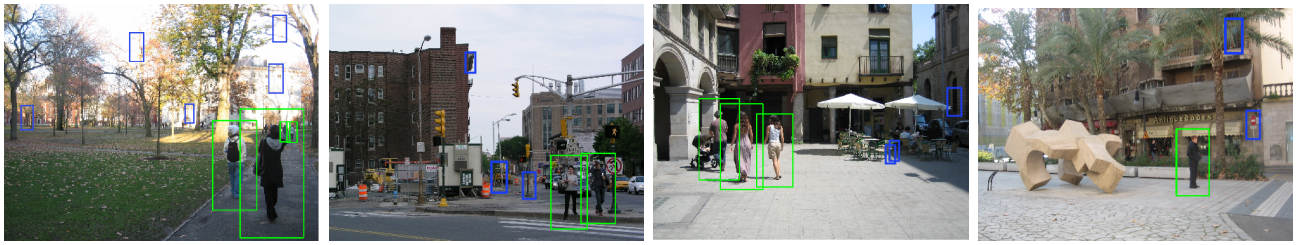


Fig. 11. MSR positive results at 30% of WSR after non-max suppression. Blue rectangles indicate avoided false positives (improving performance); TP are marked with green.



Fig. 12. MSR negative results at 30% of WSR after non-max suppression. Yellow rectangles indicate FN caused by MSR (affecting performance); blue rectangles indicate avoided false positives (improving performance); TP are marked with green, while red rectangles are FP.

choice for spectral residual analysis has demonstrated comparatively better results on the task of faster object detection than other state-of-the-art saliency methods. Although the initial goal was of faster execution, we plan to modify MSR to take object-specific spectral information into account in order to improve even more detection performance.

ACKNOWLEDGMENT

Grimaldo was supported with a scholarship by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)

REFERENCES

- [1] M. Enzweiler and D. Gavrilu, "Monocular pedestrian detection: Survey and experiments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 12, pp. 2179–2195, 2009.
- [2] V. Kastiraki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.
- [3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.
- [4] Q. Zhu, S. Avidan, M. Yeh, and K. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1491–1498, 2006.
- [5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 511–518, 2001.
- [6] X. Hou and L. Zhang, "Thumbnail generation based on global saliency," *Advances in Cognitive Neurodynamics*, pp. 999–1003, 2008.
- [7] A. Y. S. Chia, S. Zhuo, R. K. Gupta, Y. W. Tai, S. Y. Cho, P. Tan, and S. Lin, "Semantic colorization with internet images," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 156, 2011.
- [8] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [9] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1254–1259, 1998.
- [10] C. Y. Ip and A. Varshney, "Saliency-assisted navigation of very large landscape images," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1737–1746, 2011.
- [11] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 37–44.
- [12] D. Lowe, "Object recognition from local scale-invariant features," *IEEE International Conference on Computer Vision*, pp. 1150–1157, 1999.
- [13] J. Feng, Y. Wei, L. Tao, C. Zhang, and J. Sun, "Salient object detection by composition," in *IEEE International Conference on Computer Vision*, 2011, pp. 1028–1035.
- [14] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: object localization by efficient subwindow search," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
- [15] P. F. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.
- [17] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu, "Global contrast based salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2011, pp. 409–416.
- [18] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," *Advances in neural information processing systems*, vol. 19, p. 545, 2007.
- [19] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.
- [20] Z. Zhai and M. Shah, "Visual Attention Detection in Video Sequences Using Spatiotemporal Cues Categories and Subject Descriptors," in *ACM International Conference on Multimedia*, 2006, pp. 815–824.
- [21] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1, pp. 157–173, 2008.
- [22] M. M. Cheng, G. X. Zhang, N. J. Mitra, X. Huang, and S. M. Hu, "Global contrast based salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2011, pp. 409–416.
- [23] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2376–2383, 2011.



Jose Grimaldo <jose.jgrimaldo@gmail.com>

Invitation - Pattern Recognition Letters Special Issue devoted to best papers of SIBGRAPI 2012

Luciano Silva <luciano.ufpr.br@gmail.com>
Para: luciano.reboucas@gmail.com
Cc: jose.jgrimaldo@gmail.com, leizer@ufba.br

17 de diciembre de 2012 09:59

Dear authors,

We are pleased to invite you to submit an extended version of your paper #101446 to Pattern Recognition Letters – Special Issue on "**Advances in Pattern Recognition and Computer Vision**", devoted to the best papers of SIBGRAPI 2012 conference.

We sincerely hope you accept the invitation. Once you have decided, please notify us about your decision on submitting or not the extended version. This should be done until December 28th, 2012.

Author submission guidelines:

(1) You should read carefully our recommendations bellow and follow the guidelines for authors in the Pattern Recognition Letters website:

<http://www.journals.elsevier.com/pattern-recognition-letters>

(2) The paper should present some new aspects that were not included in the conference paper, some new Figures or Tables and, possibly, more experimental results. We encourage you to take into account any suggestions from previous reviews that are relevant for this version.

(3) Even if most of the material used for the contribution to the SI is the same as the material used for the conference proceedings, it is necessary at least some rewording to avoid that the two papers include parts that are verbatim the same. (THE GUEST EDITORS WILL ENFORCE THIS.)

(4) Maximal length of each contribution: 7500 words and around 10 Figures/Tables.

(5) The conference paper should be included in the list of reference.

Submission deadline: **February 15, 2013**

First Review feedback to authors: **May 30, 2013 (tentative)**

Second review and final decision: **July 30, 2013 (tentative)**

Best Regards,

Luciano Silva, Sudeep Sarkar, Roberto Scopigno and Carla Freitas

Guest Editors of the special issue **Advances in Pattern Recognition and Computer Vision for Pattern Recognition Letters**