Visual Attention Processes based on
Stochastic Models: Algorithms and Bounds

Tamar Avraham
Visual Attention Processes based on Stochastic Models: Algorithms and Bounds

Research Thesis submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Tamar Avraham

Submitted to the Senate of the Technion - Israel Institute of Technology

Iyar, 5767 Haifa May 2007
The Research Thesis Was Done Under The Supervision of Assoc. Prof. Michael Lindenbaum in the Department of Computer-Science. Part of the work was done in cooperation with Dr. Yaffa Yeshurun from the Psychology Department at the University of Haifa.

THE GENEROUS FINANCIAL HELP OF THE TECHNION IS GRATEFULLY ACKNOWLEDGED.
Contents

Abstract 1

1 Introduction 3
  1.1 Motivation and Goals . . . . . . . . . . . . . . . . . . . . . . . 3
  1.2 Main Contributions . . . . . . . . . . . . . . . . . . . . . . . 4
  1.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . 5
    1.3.1 Human Visual Attention - Theories and Models . . . . . 5
    1.3.2 Visual Attention in Computer Vision . . . . . . . . . . 14
    1.3.3 Analysis of Visual Search Difficulty . . . . . . . . . . 16
  1.4 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . 18

2 Framework 19
  2.1 Object-based/Region-based Visual Search . . . . . . . . . . . 19
  2.2 Sources of Information for Directing Visual Attention . . . . 20
  2.3 Models of Inner-Scene Visual Similarity . . . . . . . . . . . 21
  2.4 Probability Estimation . . . . . . . . . . . . . . . . . . . . . 22
  2.5 Local vs. Global Saliency . . . . . . . . . . . . . . . . . . . 22
  2.6 Dynamic vs. Static Visual Attention Processes . . . . . . . . 23
  2.7 Measures of Performance . . . . . . . . . . . . . . . . . . . . 24
3 Inherent Limitations of Visual Search Performance

3.1 Introduction ................................. 25

3.2 Notations ................................... 26

3.3 COVER: A Measure for Search Difficulty Combining Target Isolation and Candidate Scattering ................. 27

3.4 Lower Bounds on Search Algorithms ...................... 29

3.4.1 No Information Given ............................ 30

3.4.2 Given a Lower Bound \( d_0 \) on the Minimal Target-Nontarget Feature Space Distance .................. 30

3.4.3 Given \( d_0 \) and a Bounded Metric Space ................. 31

3.4.4 Given the COVER of the Input ..................... 31

3.5 FLNN (Farthest Labeled Nearest Neighbor): A Simple Search Algorithm and its Upper Bounds .................. 32

3.5.1 Given \( d_0 \) and a Bounded Metric Space ................. 33

3.5.2 Given the COVER of the Input ..................... 34

3.6 Illustration of the COVER Results ....................... 35

3.7 Experiments .................................... 36

3.8 Discussion ..................................... 38

4 VSLE (Visual Search using Linear Estimation): A Dynamic Stochastic Visual Search Algorithm ................. 43

4.1 Introduction .................................... 43

4.2 The Need to Extend FLNN ............................. 44

4.3 Candidates Labels as Dependent Random Variables ........... 45

4.4 Dynamic Search Framework ............................ 46

4.5 Minimum Mean Square Error Linear Estimation ............. 46

4.6 The VSLE Algorithm ................................ 47

4.7 Combining Prior Information ........................... 49
4.8 Experiments ................................................. 49
  4.8.1 Initial Demonstration ................................. 49
  4.8.2 VSLE Applied to the MIT+CMU Faces Database ... 53
  4.8.3 Combining VSLE and VJA (Viola & Jones Algorithm [90]) .................................................. 54
4.9 Discussion .................................................. 56

5 Esaliency (Extended Saliency): A Stochastic Attention Model
  Quantifying (Relaxed) Global Saliency. .................. 59
  5.1 Introduction .............................................. 59
  5.2 Framework and Underlying Assumptions ................. 62
    5.2.1 Modeling Object Identity by a Stochastic Approach . 63
    5.2.2 Underlying Observations ........................... 64
  5.3 The Esaliency Algorithm ................................ 70
    5.3.1 Specifying a Joint Label Distribution
      as a Bayesian tree ....................................... 71
    5.3.2 Finding the joint assignments with highest likelihood
      and deducting the (E)saliency of each candidate ........ 74
    5.3.3 Some Simple Variations on the Proposed Saliency Mech-
      anism ..................................................... 75
  5.4 Experimental Evaluation of Esaliency .................... 76
    5.4.1 Implementation ...................................... 77
    5.4.2 A Synthetic Illustration ............................ 78
    5.4.3 Testing Esaliency on Natural Scenes ................. 78
    5.4.4 Esaliency on Natural Scenes with Non-fixed Expected
      Value .................................................... 81
    5.4.5 Using Esaliency to Accelerate Pedestrian Detection .. 83
    5.4.6 Evaluation of Esaliency vs. Human Eye Fixations ... 84
6 A Psychophysical Study for Predicting Human Visual-Search Performance

6.1 Introduction .................................................. 107
  6.1.1 Related work ............................................. 108
  6.1.2 Extensions of the COVER and FLNN Models ........ 110
  6.1.3 Outline .................................................... 113

6.2 Experiments .................................................... 114
  6.2.1 Experiment 1: Orientation Unidirectional ............ 116
  6.2.2 Experiment 2: Orientation Bidirectional ............ 121
  6.2.3 Experiment 3: Color Unidirectional ................. 124
  6.2.4 Experiment 4: Color Bidirectional ................. 127
  6.2.5 Predicting Rosenholtz’s[72] Orientation Experiments 129

6.3 Discussion ..................................................... 131

7 General Discussion ............................................... 159

Bibliography ...................................................... 163

Appendices ........................................................ 177

A The EsaliencyVSLE Application ................................ 177
  A.1 General description ....................................... 177
  A.2 Parameters Description ..................................... 178
    A.2.1 Algorithms to apply .................................... 178
    A.2.2 Input Frames ............................................ 180
A.2.3 Segmentation Parameters .................................. 181
A.2.4 Initial Priors ................................................. 182
A.2.5 Features .................................................... 182
A.2.6 covariance function ......................................... 184
A.2.7 Esaliency parameters ...................................... 184
A.2.8 VSLE parameters .......................................... 185
A.2.9 Detection parameters ...................................... 185
A.2.10 Stop Criteria, Display and Save parameters ......... 186
A.3 Results Description .......................................... 187

B Combining VSLE and Viola&Jones faces detection: Implementation details 189
List of Tables

3.1 Experiment results for FLNN and COVER .......................... 40

4.1 Qualitative search performance dependencies on the target-
        target similarity and target–non-target similarity. .............. 52

4.2 VSLE applied to the MIT+CMU database. Demonstrates the
        percentage of targets found if VSLE is stopped at different
        times. ................................................................. 55

5.1 Results for Esaliency on 50 images from the Washington Ground
        Truth database[2]. ..................................................... 91

5.2 Comparing results of Esaliency and results reported in Itti et
        al. [44]. Searching for locally-salient objects (red can, emer-
        gency triangle and traffic signs). ................................ 93

5.3 Esaliency with further preferences. ................................. 95

5.4 Correlation coefficients between human saliency maps [59] and
        computational models saliency maps. ............................ 103

6.1 Psychophysical evaluation: The ability of the COVER, NOISY-
        COVER, Rosenholtz’s Saliency model and SDT-based models
        to predict Experiment’s 1 results. ............................... 137
6.2 Psychophysical evaluation: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCRef models for Experiment 1. . . . . . . . . . . . 139

6.3 Psychophysical evaluation: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 2 results. . . . . . . . . . . . . 140

6.4 Psychophysical evaluation: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCRef models for Experiment 2. . . . . . . . . . . 142

6.5 Psychophysical evaluation: The colors used in experiments 3 and 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144

6.6 Psychophysical evaluation: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 3 results. . . . . . . . . . . 146

6.7 Psychophysical evaluation: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCRef models for Experiment 3. . . . . . . . . 147

6.8 Psychophysical evaluation: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 4 results. . . . . . . . . . . 148

6.9 Psychophysical evaluation: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCRef models for Experiment 4. . . . . . . . . . 150

6.10 Psychophysical evaluation: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict ([72]) Experiment 1 results. . . . . . . . . 151
6.11 Psychophysical evaluation: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Rosenholtz’s Experiment 1. . . . 153

6.12 Psychophysical evaluation: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict ([72]) Experiment’s 2 results. . . . . . . . . . . . . . 154

6.13 Psychophysical evaluation: A Quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Rosenholtz’s Experiment 2. . . . 156

6.14 Psychophysical evaluation: Summary of the various models prediction abilities. . . . . . . . . . . . . . . . . . . . . . . . 157
List of Figures

1.1 Examples from Yarbus's [100] eye fixation study ............... 9
1.2 Feature-search (pop-out) vs. conjunction-search (sequential)
    according to Triesman et al. [85] ............................... 9
1.3 Itti et. al. [45] bottom-up visual attention model ............. 11

3.1 An illustration of the COVER upper bound ..................... 39
3.2 The COIL-100 database [56] images, car model images, and
    Gaussian derivative filters ................................. 39
3.3 Images from Berkeley hand segmented database [53] ............ 40
3.4 Graphic representation of FLNN and COVER experiment re-
    sults. .................................................. 40

4.1 The VSLE algorithm ........................................... 48
4.2 VSLE results for the COIL100 [56] images and Berkeley hand
    segmented [53] images. .................................. 51
4.3 Estimation of labels covariance vs. feature-space-distance, for
    COIL-100 images [56] and Berkeley hand segmented images
    [53]. .................................................... 52
4.4 VSLE applied on an automatic-color-segmented image to de-
    tect faces. ............................................. 55
6.7 Psychophysical evaluation: Experiment 4 results and predictions of the different models. . . . . . . . . . . . . . . . . . . . 149
6.8 Psychophysical evaluation: Rosenholtz’s Experiment 1 results and predictions of the different models. . . . . . . . . . . . . . . . . . . . 152
6.9 Psychophysical evaluation: Rosenholtz’s Experiment 2 results and predictions of the different models. . . . . . . . . . . . . . . . . . . . 155
A.1 The VSLE and Esaliency application User Interface . . . . . . . . 179
Abstract

Image analysis processes often scan the image exhaustively, looking for familiar objects of different location and size. An attention mechanism that suggests priorities to different image parts, and by that directs the analysis process to examine more interesting locations first, can make the whole process much more efficient. Motivated from studies of the human visual system, this study focuses mainly on inner-scene similarity as a source of information, testing its usefulness for directing computer visual attention from different perspectives. A study of the inherent limitations of visual search algorithms suggested the COVER measure. Taking a deterministic approach, we found that a measure similar to Kolmogorov’s epsilon-covering bounds the performance abilities of all visual search implementations, and can quantify the difficulty of a search task. It was analytically proven that a simple algorithm (denoted FLNN-Farthest Labeled Nearest Neighbor) meets this bound. Taking a stochastic approach, we model the identity of the candidate image parts as a set of correlated random variables and derive two attention/search algorithms based on it. The first algorithm, denoted VSLE (Visual Search using Linear Estimation) suggests a dynamic search procedure. Subsequent fixations are selected from combining inner-scene similarity information with the recognizer’s feedback on previously attended sub-images. We show that VSLE can accelerate even fast detection processes as the one suggested by Viola and Jones [90] for face detection. The second algorithm, denoted Esaliency (Extended Saliency) needs no recognition feedback and does not change the proposed priorities. It is therefore denoted a static algorithm and can compete with previous attention mechanisms that
also suggest a pre-calculated saliency map that is used to guide the fixations order. This algorithm incorporates inner-scene similarity information with the common expectation for a relatively small number of objects of interest in a scene and with the observation that the content of an image has a relatively clustered structure. Unlike other acceptable models of visual attention (e.g., Itti and Koch’s [45]) that associate saliency with local uniqueness, the Esaliency algorithm takes a global approach by considering a region as salient if there are only a few (or none) other similar regions in the whole scene. The algorithm uses a graphical model approximation that allows the hypotheses for target locations with the highest likelihood to be efficiently revealed. Its performance on natural scenes is extensively tested and its advantages for directing the recognition algorithm’s resources are demonstrated. While our main goal is attention mechanisms for computer vision, we have also tested the relevance of our measures and models for human performance prediction in a Cognitive Psychology study. We extended the COVER and FLNN models to account for internal-noise, and tested their predictive abilities for orientation-search and color-search tasks where distractors’ homogeneity and target-distractors similarity were systematically manipulated. In comparison to other prominent models of human visual search, the predictions of our models were the closest to the actual human performance.
Chapter 1

Introduction

1.1 Motivation and Goals

Image analysis processes often scan the image exhaustively, looking for familiar objects of different location and size. An attention mechanisms that suggests priorities to different image parts and by that directs the analysis process to examine more interesting locations first, can make the whole process much more efficient.

This study aims to develop a general framework for computer visual attention. Specifically, our objective is to find the target objects as fast as possible, in minimal expected time. We do not aim to model biology mechanisms but we do adopt ideas from relevant research in that field. Similar to Duncan and Humphreys’ model [28], we focus on inner-scene similarity and show that it has a major influence on a visual search task’s difficulty and that it can be an effective source of information when directing a search process.

We had two major goals in this research: The first was to provide computer vision algorithms that improve the trivial performance of an arbitrary
order based search, and that can compete with previous suggested mechanisms that use different sources of information for directing the attention fixations. The second goal was to study the inherent limitations of such visual search (or attention directing) algorithms, and to be able to quantify the difficulty of a given search task.

1.2 Main Contributions

The following main contributions were suggested:

1) The COVER measure: taking a deterministic approach, we suggest a measure similar to Kolmogorov’s epsilon-covering that sets inherent limitations on visual search and that quantifies the difficulty of a search task. We show that this measure bounds the performance of all search algorithms and suggest a simple algorithm (FLNN) that meets this bound.

2) The VSLE (Visual Search using Linear Estimation) algorithm: taking a stochastic approach, we model the identity of the candidate image parts as a set of correlated random variables and derive a dynamic search procedure. Subsequent fixations are selected from combining inner-scene similarity information with the recognizer’s feedback on previously attended sub-images. We show that VSLE can accelerate even fast detection processes as the one suggested by Viola and Jones for face detection.

3) The Esaliency (Extended saliency) algorithm: by incorporating inner-similarity information with the common expectation for a relatively small number of objects of interest in a scene and with the observation that the content of the image has a relatively clustered structure, we suggest a new method for quantifying the saliency of an image region. Unlike other acceptable models of visual attention (e.g., Itti and Koch’s) that associate saliency
with local uniqueness, Esaliency takes a global approach by considering a region as salient if there are only a few (or none) other similar regions in the whole scene. The algorithm uses a graphical model approximation that allows the hypotheses for target locations with the highest likelihood to be efficiently revealed. Its performance on natural scenes is extensively tested and its advantages for directing the recognition algorithm’s resources are demonstrated.

4) A Cognitive Psychology study for evaluating the abilities of COVER and FLNN models to predict human performance: We have extended the models to account for internal-noise, and tested their predictive abilities for orientation-search and color-search tasks where distractors’ homogeneity and target-distractors similarity were systematically manipulated. In comparison to other prominent models of biology visual search, our models’ predictions were the closest to human performance.

1.3 Related Work

1.3.1 Human Visual Attention - Theories and Models

The highly effective attention mechanisms in the human visual system were extensively studied from the psychophysical and physiological points of view. It is well known today that the resources used for visual processing are not divided equally over the whole viewed scene. Interestingly, it was found that effective attention is possible even in the absence of any information about the sought for entities.

Using the first eye tracking devices, Yarbus [100] found that the eyes rest much longer on the elements of an image that seem to carry essential information, while other elements may receive little or no attention. Yarbus’s
experiments show that the number of details contained in an element of a picture does not determine the degree of attention attracted to it. Human eyes voluntary and involuntary fixate on those elements of an object which may carry essential information (for instance - when looking at a human face, an observer usually pays most attention to the eyes, the lips, and the nose); see figure 1.1. In additional experiments Yarbus showed that the purpose of the observer effects the fixation pattern, and found that several types of eye movements are involved when observing an image.

There is much evidence that, in addition to the eye movements (referred to as the overt process), the attention mechanism focuses on different image parts without moving the eyes (a covert process). Posner et al. [66] suggested that this covert attentional mechanism acts as a spotlight moving about the scene. The spotlight metaphor studies suggested that attention covers only about 1 degree of visual angle [34]. However, further studies showed that attention can cover a much wider area of the visual field at some times and focus only on a tiny region at other times. This led to the zoom lens metaphor [35]. This theory suggests that the total attention capacity is limited, and therefore the smaller the attended region is, it can be processed in finer resolution. Such a highlighted circular region may contain one object but may also contain a part of an object or several objects. Methods that follow this approach are therefore referred to as space-based or feature-based.

An alternative approach suggests that human visual attention is object-based. That is, that the spatial region getting attention is not of fixed shape but rather adapted to a perceived object [27, 75]. This approach implies that the decision for attention follows some perceptual grouping processes. The term “object” should be specified with care in this context. It is unlikely, for example, that an almost perfect grouping process, dividing the
perceived scene to semantically meaningful objects such as cars or people, precedes the attention process. Such a complex grouping, seems to require information on the object identity or class and is likely to happen only after the region is attended. Several experimental psychology studies (e.g., [95]) provide evidence that the objects involved are specified by simple grouping processes related to basic Gestalt laws [96] such as proximity, similarity, and uniform-connectedness [63]. To avoid confusion between a model relying on semantical meaningful segmentation and one relying on relatively weak grouping cues we sometimes refer to the later as region-based.

Neisser [55] suggested that visual processing is divided into pre-attentive and attentive stages. The first consists of parallel processes that simultaneously operate on large portions of the visual field, and form the units to which attention may then be directed. The second stage consists of limited-capacity processes that focus on a smaller portion of the visual field.

Triesman and Gelade [85] suggested the feature integration theory (FIT). Consistently with physiology findings, they suggest that in the pre-attentive stage the visual input is represented by separate retinoscopic maps for each of the visual attributes such as color and shape. According to their hypothesis, the binding of the features requires focal attention. They have suggested that there is a dichotomy between visual search tasks in which the target is located immediately (pop-out) and search tasks that required scanning. According to their theory, the pop-out tasks are feature-search tasks in which binding is not necessary as the target is distinguished from the distractors by one feature. The second type are conjunction-search tasks that require binding of features in order to recognize the target; see figure 1.2. Later experiments showed that some tasks that fit the conjunction-search definition do not require serial scanning (e.g., [54]), which triggered some updates of
the FIT theory (e.g., [84]). Nevertheless, many computational models (e.g., [46, 99, 87, 45]) followed the FIT.

Koch and Ullman [46], using the Winner-Take-All neural network (term first used by Feldman [38]) and Inhibition-Of-Return principle, suggest to construct one saliency map that ‘grades’ each location in the scene by the amount of overall conspicuousness. The Winner-Take-All network promises that only one location is active at a time. Inhibiting the selected location causes an automatic shift towards the next most conspicuous location. This saliency map combines the information of the individual (pre-attentive) feature maps. Abstractly, the serial search will ‘visit’ objects in the scene in the order dictated by their (descending) values in the master saliency map.

Ullman and Koch’s model also includes the rules of proximity and similarity preferences. Each time the focus shifts from a certain location, the next attended location will preferably be close and similar to the one just attended. (In this work we claim, that in order to achieve good performance in a visual search task, it is preferable to do the opposite - prefer attention shifts to dissimilar locations until the first target is found). Koch and Ullman not only suggest how to implement a man-made such machine, but also suggest how such a network can exist in the biology system by proposing possible locations for such a saliency map in the visual pathway.

Wolfe in his Guided-search model [99], adds and suggests that attention is drawn to peaks in the activation map that represent areas in the image with the largest combination of bottom-up saliency and top-down saliency. An item will activate a bottom-up map if it varies from its close neighborhood. An item will activate a top-down map if it has joint properties with pre-specified targets. The weights given to each of the feature maps in order to combine them into one map is task dependent. Wolfe suggests, that even
Figure 1.1: Example results from Yarbus’s eye tracking [100]. The eyes focus more on some parts of the images.

Figure 1.2: An example for a feature-search (pop-out) vs. conjunction-search (sequential) according to Triesman et al. [85]
the same subject changes those weights over time, and by that can improve its efficiency to perform a task. Wolfe builds a computer simulation of this model, describing in details an algorithm for building an activation map for each feature, and for constructing the global activation map.

Tsotsos et al. [87], suggests the Selective Tuning model. This model is suggested both as a hypothesis for primate’s visual attention, and as a solution for robot vision implementation. It quite resembles Koch and Ullman’s model, though different in a few principles: The implementation of the winner-take-all principle is different and, according to the authors, fits better the knowledge on primate visual system. Proximity has no effect on choosing the next attended item. Their model includes a separate mechanism for detecting peripheral salient items, suggesting it models eye foveate saccade movements.

Itti et. al. [45], following Koch and Ullman [46], have suggested an updated bottom-up computational model; see figure 1.3 for a schematic description of their model. Given an input image, separate feature maps of color, intensity and orientation in different scales are extracted using linear filtering. Local spatial contrast is estimated for each feature at each location resulting conspicuity maps per feature. These are combined to form one saliency-map that guides the attention focus. Using the Winner-take-all and inhibition-of-return concepts, attention is drawn to the different locations in descending priority (saliency) order.

Duncan and Humphreys [28] rejected the dichotomy of parallel vs. serial search, central to FIT, and proposed an alternative hypothesis based on similarity. According to them, two types of stimulus similarity are involved in a visual search task and define the search difficulty. One, denoted the inter-alternative similarity, is the similarity between objects in the scene and prior
Figure 1.3: A schematic description of Itti et al’s visual-attention model (taken from [45]).
knowledge about the possible targets. The other similarity is between the objects in the scene. They suggested that a search procedure starts with a hierarchial segmentation, defining *structural units*, which are the candidates competing for the attention resource. Units that are similar to the potential target get higher weights, which are used as priorities. The search is facilitated if several structural units are similar, because these are linked and when the priority is changed in one unit, the change propagates to the linked units as well. In this way suppression may be spread among the units, without needing to treat each one individually. Thus, if all non-targets are homogeneous, they may be rejected together, resulting in a fast (pop-out-like) detection process; if they are more heterogeneous, the search is slower.

The SERR model proposed by Humphreys and Muller’s [43] relies on Duncan and Humphreys’s [28] theory. It uses an hierarchy neural network and a Bolzmann machine. Experiments (only on synthetic search tasks) show it matches reaction times measured by psychophysical experiments for character detection tasks, after the network’s weights and thresholds are set accordingly.

Wang [94] describes a neural network based on oscillatory correlation modeling attention shifts between objects. The oscillatory correlation technique adds the property of grouping pixels into objects. The network selects the largest object each time. An inhibitory mechanism causes the objects of a scene to be scanned by the descending order of their size.

While the behavior in a case of a pop-out target is clear and not so interesting, Yokosawa and Lindenbaum [101] studied the damaging effect of a pop-out distractor on visual search efficiency. Concluding their psychophysical experiments, they suggest that after the pop-out was examined and found not to be the target, it is ‘blocked’ for a while, but then the temporary ‘in-
hibition of return’ ends and the distractor attracts the attention again. This procedure is repeated until the target is found.

Other studies have highlighted the role of sensory factors in visual search, demonstrating that a substantial part of search efficiency is determined by low efficiency level factors such as target eccentricity and element density (e.g., [20, 21, 22, 40, 89, 60]).

The Signal-Detection-Theory (SDT) models (e.g., [29, 30, 42, 61, 74]), designed to predict human performance on synthetic search tasks, assume that the stimuli are observed with addition stochastic noise. According to this view, a false-detection may occur when one of the distractors, given its noisy-observation, is mistakenly perceived as a target (i.e., belonging to the target distribution), and a miss may occur when the target is mistakenly perceived as a distractor. Hence, the chances of such detection errors increase as the number of search items, and the similarity between the target and distractors increase.

Rosenholtz has suggested variations of the SDT based models and showed they improve prediction over the original SDT models for orientation search tasks [72]. While SDT models assume that the observer has a record of the exact distractors distribution, the Best-Normal model suggests that during visual search the observer uses a more simple approximated representation of the distractors’ distribution. The true distribution is represented only by its mean and variance, that is, by the normal distribution that best fits the true distractors distribution. Note that while both the Best-Normal and the original SDT models predict that search performance should get harder as target-distractors similarity increases, only the Best-Normal model can account for the increase in search difficulty with an increase in distractors heterogeneity. Another variation of SDT models is the RCref model (Relative-Coding With
Reference; [72]), which is a modification of the relative coding model [62]. Like the Best-Normal model, the RCref model suggests that the observer does not use the exact distribution of the search items. The recorded distribution does not correspond to the feature-values themselves, but to relative values. Specifically, the recorded distribution corresponds to the combination of the differences between the various items that are present in the display and the differences between display items and a reference target. Thus, part of the times an observed display item is compared to another display item and on other times it is compared to a reference-target.

1.3.2 Visual Attention in Computer Vision

The computer vision community is becoming more and more aware of the need to incorporate attention mechanisms in computer vision applications such as object recognition, object detection and scene analysis. Several successful search mechanisms relying (mostly) on top-down information were recently suggested as detection algorithms (e.g., [51, 90, 13, 80]). Local descriptors (e.g. SIFT by Lowe [51], Amit and Geman [3]), for example, can accurately characterize specific object parts and can serve as efficient indices for finding them in an image. Another option for accelerating the search is to use rejection based classifiers, which spend an adaptive amount of computational effort on different candidate sub-images, and can reject many non-target candidates very fast [13, 90]. In particular, the rejection cascade of AdaBoost classifiers combined with efficient Haar based components, lead to an extremely fast procedure for finding objects [90]. It seems, however, that the computational advantage is possible only when the visual appearance of the sought for object is not too complex or variable. Otherwise, a simple classifier, enabling fast rejection, is not accurate. Moreover, when there is
interest in several different objects, much of the procedure has to be repeated for each target type, slowing down the search.

The indirect search approach [98] suggests to analyze the contribution of spatial relationship knowledge to the efficiency of the search. It is suggested, for instance, to look first for large intermediate objects which can be recognized in low resolution, and to then limit the search to image regions inferred from available spatial relations knowledge. Rimey and Brown [69] direct attention based on prior knowledge on relations like ‘part-of’ and ‘next to’ together with the information they gather during the process. A bayes-net is preprogrammed for the domain (‘domain’s are dinner tables, airport, road traffic etc.) to hold the prior information.

Torralba et al. [82] use the scene context in order to direct the attention to locations and scales in which a target object is most likely to appear. Using a training stage, they learn the conditional probability density function for the object to appear in each pose, scale and location, given context measures.

Itti et al.’s [45] attention mechanism was suggested also for computer vision. It was investigated, for instance, which approach of features maps combination best enables incorporating top-down information given a training set of target objects [44]. Frintrop et al. [39] used a variation of this method for directing attention in a system combining information from 3D laser scanner. Rutishauser et al. [93] have tested and showed that this selective visual attention mechanism enables learning and recognition of multiple objects in cluttered scenes by combining it with Lowe’s [51] object recognition algorithm.

Itti et. al.’s approach, and most other visual attention models, identify saliency with local exception. That is, the saliency value at each location is essentially the local contrast in one feature or more. They do not check
uniqueness in the context of the whole scene. An object with many appearances can be considered salient if each appearance is contrasting with the background, while another object which has only one or a few appearances can be considered as less salient if it is less contrasting with the background. Therefore, while this approach may be biologically plausible, it is suboptimal for computer vision; see more discussion in chapter 5.

A region-based approach relying on local-saliency was suggested in [50]. After calculating the local contrast of each pixel, the saliency of a region (resulting from a mean-shift based segmentation) is specified as the average contrast of its pixels.

Walker et. al. [92] have suggested that salient feature-points are “those which have a low probability of being miss-classified with any other features”, and by that associated the term saliency with global exception. Using a similar definition for saliency, the image and video analysis approach of Boiman and Irani [17] considers a region non-interesting if it is similar to another image region. Their approach seem to provide very good results as an analysis tool, however, as it relies on a relatively costly, part-based similarity evaluation, it does not seem useful as a quick pre-recognition attention mechanism.

A computer vision implementation for Duncan’s [27] object-based model was suggested in [77]. However, as it requires a high quality hierarchial segmentation that was not yet suggested by any computer implementation, this system uses man-made segmentation as input.

1.3.3 Analysis of Visual Search Difficulty

While several attention mechanisms and visual search algorithms were proposed, relatively little was done to analyze the limitations of visual search

16
algorithms or to quantify the difficulty of visual search tasks. Tsotsos [86] suggested an analysis of the efficiency of visual search. He distinguishes between ‘bounded’ and ‘unbounded’ search tasks. A search task is bounded when the model of the target is known. Then, each candidate is a set of connected pixels that can be compared to the model target. In an unbounded search the model is unknown. Each subset of pixels may possibly form a target. He proved that the upper-bound for unbounded search is exponential in the image/s size, and that bounded search’s complexity is linear in the image/s size. He also discussed the conditions in which a sequence of images facilitates the search.

Some factors that effect the efficiency of indirect search are suggested in [98]. Assuming the search task parameters’ probability density functions and the expected cost is known, they suggest a way to choose whether to perform a direct or an indirect search, and to decide which intermediate object to choose to make the indirect search most efficient.

Rosenholtz [71] has developed a simple measure for target’s saliency that comes to reflects biology visual search efficiency. Given the feature-space relevant to the search task (e.g.-velocity space in motion-search, color-space in color-search), and the points in that space describing the various search items, the target saliency measure is the number of standard deviations between the target point and the mean of the distractors points (Mahalanobis distance). This work is related to our COVER measure result as it is also inspired from Duncan and Humphrey’s [28] theory, and as it involves the factors of target-nontarget similarity and nontarget-nontarget similarities. However, while our measure’s validity is phrased mathematically and is connected to the ability of search algorithms, the target saliency measure is validated by psychophysical experiments. Chapter 6 provides a comparison between the
two measures in that context.

1.4 Thesis Outline

We start by setting a general framework for visual-search algorithms in Chapter 2, where a further clarification on the context of our approach relative to previous studies is provided. Chapter 3 takes a deterministic approach to the quantitative analysis of search tasks, and presents bounds and algorithms that achieve them. Chapters 4 and 5 demonstrate our stochastic approach describing and testing the VSLE and Esaliency algorithms. Finally Chapter 6 describes the Psychophysical study that tested our models in the context of human performance. The appendices provide implementation details. Specifically, appendix A describes an application that implements the VSLE and Esaliency algorithms.
Chapter 2

Framework

2.1 Object-based/Region-based Visual Search

We follow a common approach, dividing a visual search task into two sub-tasks. One is to select sub-images, which should be considered as possible candidates. The other, the object recognition task, is to decide whether or not a candidate is a sought for object. The input to the candidate selection task is the whole-scene image, and its output is a set of candidate sub-images. The candidate selection task can be performed by a segmentation process or even by a simple division of the image into small rectangles. It can be based on top-down or bottom-up processes, meaning that it uses prior knowledge about the target’s properties or just image-based knowledge. The candidates may be of different size, bounded or unbounded [86], and can also overlap. As we shall see, our algorithms will not require the candidate selection process to accurately divide the scene into semantically meaningful objects, and will sometimes even prefer an over-segmentation.

The input to the object recognition task is a sub-image (a candidate). The recognizer decides whether the given candidate is the required object. The
decision can be based on statistical modeling, part decomposition, functional description or any other method. This stage is usually considered a high level vision task and is commonly computationally expensive, as the possible appearances of the object may vary due to changes in shape, color, pose, illumination and imaging conditions. The object recognizer may need to recognize a category of objects (and not a specific model), which usually makes it an even more complex task.

The object recognition process gets the candidates, one by one, after some ordering. This ordering is the attentional mechanism on which we focus here. Many orderings are possible but some are better than others. A major goal of this work is to provide methods for specifying good ordering so that the number of calls to the recognizer is minimized.

Given this framework, this approach to attention belongs to the object-based stream of visual attention.

2.2 Sources of Information for Directing Visual Attention

The knowledge for directing visual attention may be different in different contexts:

1. **Bottom-up local saliency.** The human attention mechanism is attracted to locations that differ from their close surround in some property, such as dominant orientation, size, and color (e.g., [85][46][45]). Nevertheless, this criterion can sometimes be misleading and is not applicable when, say, the target’s uniqueness is defined by a combination of features, or when the local contrast does not account for global uniqueness.

2. **Prior top-down knowledge.** Both the candidate selection stage and
the attention stage may obviously benefit from available prior knowledge on
the targets (e.g., [99]). Indeed, the rejection based approach and efficient
implementation yield an extremely fast search mechanism (e.g., [90]). If we
look for a collection of objects or, say, for an object with highly various ap-
pearance, then either accuracy or speed are expected to decrease.

3. Scene Context. The location and scale of expected targets in a scene
can be estimated if the semantic global context of the scene is available, or
pre-estimated (e.g., [82]).

4. Inner-scene similarities. While other applications suggest to measure
visual similarity between different images (e.g., object recognition, content
based image retrieval – CBIR) and use it as the major information that
leads to the algorithms output, the usefulness of comparing visual appear-
ance of different parts of the same image was never tested. While it was
shown that inner-scene similarities are effective for directing biology visual
attention [28], it has not been considered before for computer vision. This
work focuses on this property and studies its effectiveness. We show that
this information can derive a dynamic framework for visual search (chapters
3 and 4) or be the basis for a new bottom-up saliency measure that identifies
global uniqueness (chapter 5).

2.3 Models of Inner-Scene Visual Similarity

Our basic assumption is that more similar candidates tend to have more
similar identities. This observation is actually an obvious assumption which
stand in the basis of all object recognition algorithms, which try to find an
object of certain identity using its visual similarity to other images of the
same object or to images of objects from the same class. However, here we
challenge its usefulness for visual attention, measuring similarity between
different parts (or objects) inside one scene.

We formalize the basic belief of mutual similarity in two alternative ways:

**Deterministic:** We assume that a measure of dissimilarity between two
candidates $d(c_1, c_2)$ is higher than some threshold $d_0$ if the candidates do not
share the same identity. (Chapter 3.)

**Stochastic:** We assume that dissimilarity vs. identity correlation is a mono-
tone descending function. (Chapters 4 and 5.)

These assumptions are supported by validation experiments examining
the behavior of dissimilarity of image parts vs. the correlations between the
identities of the objects they are associated with.

### 2.4 Probability Estimation

Our VSLE and Esaliency algorithms quantify the contribution of the inner
similarities information and other observations using stochastic modeling.
Then, deciding on the attentional fixation order becomes a matter of proba-
bility estimation.

### 2.5 Local vs. Global Saliency

After selecting the candidates for attention, we treat them as a set, without
considering the locations in the image they were extracted from. Thus, with
our approach a salient candidate is one that has relative *global* uniqueness,
unlike most attention modeling systems that identify saliency with *local* con-
trast. Taking this global approach, our Esaliency algorithm does not consider
items with high local contrast salient if they have many appearances in the scene. On the other hand, it will consider a few close by similar items as salient if the total number of their appearances in the scene is small. See details in chapter 5.

2.6 Dynamic vs. Static Visual Attention Processes

Usually, systems based on bottom-up or top-down approaches calculate a saliency map before the search starts, pre-specifying the preference and the implied scan order. In this case, we call the attentional process static. In addition to a new such static mechanism (Esaliency), we propose dynamic attention mechanisms that allow the priorities to be changed based on the results of the object recognition process. Iteratively, the candidate with the highest priority receives the attention. The relevant sub-image is investigated by a high-level object recognition module that we refer to as the recognition oracle. Based on the oracle’s response and the previous priority map, a new priority map is calculated, updating the priorities of the remaining candidates.

The knowledge that enables such algorithms to make such decisions is the availability of a similarity measure between each two candidates. After the identity of one (or few) candidates is already known, it can effect the likelihood of the remaining candidates having the same/different identity. For the implementation, the distance between extracted feature vectors provides the dissimilarity measure. The derived algorithms (FLNN, VSLE) differ in the way the priorities are estimated in each iteration. See Chapters 3 and 4.
2.7 Measures of Performance

A natural cost of a search process is either the time or the number of calls to the recognition oracle required before finding either the first target, or say, half of the targets, or even all of them. Predicting this cost can help stop the search process before all candidates are examined. In the next chapter we provide predictions on the number of calls required before finding the first target using a mathematical analysis. When using our proposed stochastic algorithms (chapters 4 and 5) we experimentally measure the number of calls required to find targets (one, 50%, 75%, all) and provide statistical results over databases of images.

Evaluating the number of queries is justified when the costs associated with calling the oracle are dominant relative to the costs associated with the ordering process. This assumption is justified for computationally expensive recognition oracles. However, to be effective also when faster (less accurate) recognition processes are involved, we keep in mind that we are looking for fast initial processes that set priorities (but do not filter out candidates) and are therefore allowed to give inaccurate results. Thus, we choose relatively simple segmentation processes and measures of similarity, and use approximations that facilitate the process for estimating the priorities of the candidates.

Choosing the number of recognizer calls as a cost also implicitly assumes that all calls to the recognizer ‘waste’ a constant processing time. This is a uniform, oracle independent, approach and a good approximation for many recognition mechanism, but is not true for rejection based approaches, for example. Therefore, in our experiments with such a recognizer we compared running time (in milliseconds).
Chapter 3

Inherent Limitations of Visual Search Performance

3.1 Introduction

A search task is easier if the mutual similarities are more informative. The simplest search situation is when all targets are very similar and all non-target candidates are mutually-similar but dramatically different from the target (one book among some pens and pencils). The hardest task is when all mutual similarities of pairs of candidates are alike and thus uninformative. Such situations happen when all candidates (both targets and non-targets) are very similar (some books all the same size and color) or when all candidates are different from each other (a bunch of office equipment). A task where the candidates are divided into a few groups, where members of the group are similar to each other and dissimilar to all candidates of other groups, and where one group contains all (and only) the targets, is of intermediate difficulty.

We consider the situation where some information on the search task is
provided before it is executed (from, say, experience with search tasks in a similar context). This information is used to develop a meaningful quantifier of the search task difficulty. As we will see, the more we are pre-informed, the more accurately we can characterize the task’s difficulty. The proposed maximal knowledge is not always available, however. Therefore, we consider other characteristics, which rely on weaker knowledge.

We define the cost of a search algorithm to be the number of recognition attempts required until a target is located. For each suggested characterization, we show that any search algorithm’s cost, in the worst case, is not below a certain lower-bound (see Section 3.4), and suggest a specific algorithm with cost that is never above a certain upper-bound (see Section 3.5). We show that these lower and upper bounds coincide, implying that in the worst case sense, the bounds are tight and the algorithm is optimal.

3.2 Notations

We consider an abstract description of a search task as a pair \((X, l)\), where \(X = \{x_1, x_2, \ldots, x_n\}\) is a set of partial descriptions associated with the set of candidates, and \(l = (l_1, l_2, \ldots, l_n)\) are the binary identity labels of the candidates. \(l_i = 1\) if candidate \(i\) is a target, and \(l_i = 0\) if it is a non-target. The partial description can be, for example, feature vectors in Euclidian space. A search algorithm is provided with the set \(X\), but not with the labels \(l\). We define the cost of a search algorithm \(A\), \(\text{cost}(A, X, l)\), as the number of queries to the recognizer oracle until a target is located.

We refer to the set of partial descriptions \(x_1, x_2, \ldots, x_n\) as points in a metric space \((S, d)\). \(d : S \times S \rightarrow \mathbb{R}^+\) being the metric distance function satisfies the conditions of reflexivity (\(\forall s \in S, d(s, s) = 0\)), positivity (\(\forall s_i \neq s_j, d(s_i, s_j) \neq 0\)), triangle inequality (\(\forall s_i, s_j, s_k \in S, d(s_i, s_k) \leq d(s_i, s_j) + d(s_j, s_k)\)).
$s_j \in S, d(s_i, s_j) > 0$), symmetry ($\forall s_i, s_j \in S, d(s_i, s_j) = d(s_j, s_i)$) and triangle inequality ($\forall s_i, s_j, s_k \in S, d(s_i, s_j) + d(s_j, s_k) \geq d(s_i, s_k)$).

### 3.3 COVER: A Measure for Search Difficulty

#### Combining Target Isolation and Candidate Scattering

The proposed measures for task difficulty combine two main factors:

1. The distance between target and non-target candidates.

2. The distribution of the candidates in the feature space.

Intuitively, the more the targets are different from non-targets, the easier the search is. However, if the non-targets are also different from each other, the search becomes hard again [28].

A useful quantification for expressing a distribution of points in a metric space (and thus expressing how scattered the candidates are) uses the notation of metric cover.

**Definition 1** Let $X \subseteq S$ be a set of points in a metric space $(S, d)$. Let $2^S$ be a set of all possible subsets of $S$. $\mathcal{C} \subseteq 2^S$ is ‘a cover’ of $X$ if $\forall x \in X \exists C \in \mathcal{C}$ s.t. $x \in C$.

**Definition 2** $\mathcal{C} \subseteq 2^S$ is a ‘$d_0$-cover’ of a set $X$ if $\mathcal{C}$ is a cover of $X$ and if $\forall C \in \mathcal{C}$ diameter$(C) \leq d_0$, where diameter$(C)$ is $\max_{c_1, c_2 \in C} d(c_1, c_2)$.

**Definition 3** A ‘minimum-$d_0$-cover’ is a $d_0$-cover with a minimal number of elements. We denote some particular minimum-$d_0$-cover by $C_{d_0}(X)$ and its size by $c_{d_0}(X)$. 

27
For simplicity we sometimes refer to $c_{d_0}(X)$ as the *COVER measure*. When $X$ is a set of feature vectors in an $m$-dimensional Euclidian space, for example, $c_{d_0}(X)$ will be the minimum number of $m$-dimensional spheres with diameter $d_0$ required to cover all candidates in $X$. The above definitions follow Kolmogorov’s $\epsilon$-covering [47].

**Definition 4** Given a search task $(X, l)$, let the ‘max-min-target-distance’, denoted $d_T$, be the greatest distance of a target to its nearest non-target neighbor.

When a search task contains one target, $d_T$ is the distance of this target to the closest non-target. When there are several targets in the scene, $T_1, \ldots, T_m$, $d_T = \max_{i=1}^m (d_{T_i})$.

The following result quantifies the difficulty of the search task using the max-min-target-distance and the COVER concept, and is the main result of this chapter.

**Theorem 1** Given a search task $(X, l)$, so that $d_T$ is its max-min-target-distance, the difficulty of finding a target depends on the minimum-$d_T$-cover size, $c_{d_T}(X)$, in the sense that:

1. Any search algorithm may need at least $c_{d_T}(X)$ queries before finding a target, for the worst case.

2. There is an algorithm that always finds a target after at most $c_{d_T}(X)$ queries.

The proof to the two parts of Theorem 1 is given in Sections 3.4 and 3.5.

The minimum-$d_T$-cover size, or the COVER measure, quantifies the intuition described in Section 3.1. In the case when one target differs dramatically
from all non-targets, which are similar, the corresponding COVER is 2. (All non-targets can be covered by one covering element, and the target alone is covered by a second element.) In a case where all \( n \) candidates are scattered and all mutual distances are \( \geq d_T \), COVER is \( n \). When there are \( k \) groups of candidates so that all inner-group distances \( < d_T \) and all outer-group distances \( \geq d_T \), COVER is \( k \).

The above examples are trivial extreme cases in which the task difficulty is easily revealed. The COVER measure also provides a measure for intermediate situations that are harder to analyze. For instance, the candidates may be scattered, but a target can be significantly far away from any of them, providing a relatively small COVER. On the other hand, the candidates may be divided into a small number of groups, but since the targets and non-targets are mixed together in one of these groups (or more), the COVER measure will have a relatively high value, characterizing the search difficulty correctly.

### 3.4 Lower Bounds on Search Algorithms

In this section, we prove Theorem 1.1 but first provide some other less tight lower bounds on all search algorithms. Consider a set of search tasks associated with some limitation on the partial descriptors \( X \in \mathcal{X} \) and some limitation on the possible assignments \( l \in \mathcal{L} \). Then, LB is a lower bound on the performance of algorithm \( A \), if there is an input \( X \in \mathcal{X} \) and an assignment \( l \in \mathcal{L} \) so that \( \text{cost}(A, X, l) \geq \text{LB} \).

\( \mathcal{X} \) and \( \mathcal{L} \) describe the information available about the search task. In this section we gradually reduce the uncertainty, and consequently, tighten the bound.
3.4.1 No Information Given

It is easy to show that if there is no limitation on the labeling, then only a trivial lower bound may be imposed. Note that knowledge about $X$ does not matter here: for any algorithm and any set of candidates, there is a labeling for which the target is the last candidate examined.

Claim 1 $\forall A, \forall X, \exists l, \text{cost}(A, X, l) \geq n$.

Proof: Until a search algorithm $A$ finds the first target, it receives only no answers from the oracle $O$. Therefore, given an input candidate set $X$, and a specific algorithm $A$, the sequence of attended candidates may be simulated under the assumption that the oracle returns only no answers. Let $\pi = \pi_1, \pi_2, \ldots, \pi_n$ be this resulting ordering of the candidates. Choose the assignment $l$ to be $l_i = 1$ for $i = \pi_n$ and $l_i = 0$, otherwise.

3.4.2 Given a Lower Bound $d_0$ on the Minimal Target-Nontarget Feature Space Distance

Suppose we know that at least one target is dissimilar from the non-targets. Quantitatively, suppose that $d_T$ is bounded from below: $d_T > d_0$. Let $L_{d_0}$ denote the possible labeling satisfying this condition. Furthermore, let $d_i$ denote the feature-space-distance of the $i$-th candidate to its nearest neighbor. We define an isolated point as a feature-vector $x_i$ for which $d_i > d_0$. Finally, let $X_{d_0, N_{ip}}$ denote the set of candidate sets for which the number of isolated points is $N_{ip}$. The claim below suggests that $N_{ip}$ is a lower-bound in this case.

Claim 2 $\forall A \exists X \in X_{d_0, N_{ip}} \exists l \in L_{d_0} \text{cost}(A, X, l) \geq N_{ip}$. 
Proof: Under the limitation of $L_{d_0}$, all candidates for which $d_i > d_0$ can be possible targets. Choose $l$ to assign the label 1 only to the last candidate in $\pi$ that satisfies $d_i > d_0$.

Note that the target for which the distance to a distractor is maximal ($d_T$) can be associated with $d_i < d_0$, if its closest neighbor is a target as well. Therefore, the last claim suggest a bound which is not tight.

3.4.3 Given $d_0$ and a Bounded Metric Space

If the feature vectors $x_1, x_2, \ldots, x_n$ belong to a bounded metric space $[0, 1]^m$, then the amount of feature vectors $x_i$ that can satisfy $d_i > d_0$ may be limited, leading to a smaller bound.

Claim 3 Let $X_m = \{ X \mid x \in X \Rightarrow x \in [0, 1]^m \}$, and specify $L_{d_0}$ according to the $L_\infty$ metric. Then, $\forall A \exists X \in X_m, l \in L_{d_0} \text{ cost}(A, X, l) \geq LB_m = \min(\lceil \frac{1}{d_0} \rceil^m, n)$

Proof: Consider a rectangular grid in the $[0, 1]^m$ space such that the distance between grid points is more than $d_0$. There are $\lceil \frac{1}{d_0} \rceil^m$ such points. Choose the $n$ candidates such that their description $X$ are points divided equally between those $\lceil \frac{1}{d_0} \rceil^m$ locations'. Given an algorithm $A$ defining the ordering $\pi$, choose the assignment $l$ that assigns 1 to the group of candidates located in the grid point whose first appearance in $\pi$ is last. Similar results for other metrics are available.

3.4.4 Given the COVER of the Input

Now, in addition to being informed that $d_T > d_0$, we also know that the possible input $X$ is limited to having some known minimal-$d_0$-cover of size
In this case, we show that $c$ is a lower bound on all search algorithms’ cost. The following claim is equivalent to Theorem 1.1.

**Claim 4** \[ \forall A \exists X \in X_{d_0,c}, l \in L_{d_0} \text{ cost}(A, X, l) \geq LB_{d_0,c} = c \]

**Proof:** Choose $c$ points in the metric space, so that the entire inter-point distance is more than $d_0$. Choose the $n$ candidates to be divided equally among these locations. Choose $l$ that assigns 1 to the group of candidates located in the point whose first appearance in $\pi$ (the ordering dictated by Algorithm $A$) is last.

The proof clearly resembles that of Claim 3. In fact, the grid built in Claim 3 is a set of points associated with a cover of size $\lceil \frac{1}{d_0} \rceil^m$. Note that no specific metric is considered in the more general theorem, and of course, the metric cover is a much tighter bound, for most input.

### 3.5 FLNN (Farthest Labeled Nearest Neighbor): A Simple Search Algorithm and its Upper Bounds

In this section we present a simple algorithm and show that its cost is bounded from above by the bounds presented in Section 3.4. $UB$ is an upper bound on the performance of an algorithm $A$ if for all input $X \in \mathcal{X}$ and assignment $l \in \mathcal{L}$, $\text{cost}(A, X, l) \leq UB$.

**Simple search algorithms**

The following two simple algorithms aim to minimize the cost of finding the first target.
**FNN**- Farthest Nearest Neighbor: Given the set of candidates’ partial description \( X = \{x_1, \ldots, x_n\} \), compute the distance \( d_i \) of each candidate \( i \) to its nearest neighbor. Order the candidates by descending \( d_i \) and query the oracle according to this order until finding a target.

**FLNN**- Farthest Labeled Nearest Neighbor: Given the set of candidates’ partial description \( X = \{x_1, \ldots, x_n\} \), randomly choose the first candidate, query the oracle and label this candidate. Repeat iteratively until a target is detected: for each unlabeled candidate \( i \), compute the distance \( d_{Li} \) to the nearest labeled neighbor. Choose the candidate \( i \) for which \( d_{Li} \) is maximum. Query the oracle to get its label.

FNN is optimal for target assignments \( l \in L_{d_0} \) when there is only one target in the scene. When two targets are present in the scene, they may be similar to each other and hence may be detected last by FNN, even if each of them is far from the nearest non-target. Therefore, we prefer FLNN, which can handle several targets. We now show that, in the worst case, it does not need more queries than those specified in Claims 3 and 4, implying that the latter bounds are tight.

### 3.5.1 Given \( d_0 \) and a Bounded Metric Space

As in Section 3.4.3, we assume a lower bound \( d_0 \) on \( d_T \) and consider candidates that belong to the bounded metric space \([0, 1]^m\). We show that FLNN always finds a target after at most \( LB_m \) queries.

**Claim 5** \( \forall X \in X_m, l \in L_{d_0} \) cost(FLNN, \( X, l \)) \( \leq UB_m = min([\frac{1}{d_0}]^m, n) = LB_m \).

**Proof:** Let the \( i \)-th candidate be a target for which the distance to its nearest distractor is \( d_T > d_0 \). Consider the hypercube \( H \) with sides of length
2d₀ so that xᵢ is its center. H includes only targets. The rest of the metric space [0, 1]ᵐ may be covered with at most ⌈d₀⁻¹⌉⁻¹ hypercubes of side d₀. By construction, FLNN does not query two candidates in one hypercube (which are at most a distance of d₀ from one another) before it queries at least one candidate in H. Therefore, a target (not necessarily i) will be found after at most min(n, ⌈d₀⁻¹⌉⁻¹) queries.

**Claim 6** Consider the partition of the [0, 1]ᵐ into [d₀⁻¹]ᵐ hypercubes, and let populated(X, d₀) be the number of hypercubes containing at least one candidate. Then,

\[ \forall X \in X, l \in L, cost(FLNN, X, l) \leq UB_{populated} = populated(X, d₀). \]

The proof is similar to that of Claim 5. While this result is simple, it provides a crude version of our main result, which uses the more complex cover concept, it gives a much tighter bound than UBₘ and is much easier to compute than the cover size.

### 3.5.2 Given the COVER of the Input

The tightest upper bound, proposed in Theorem 1, is now proved:

**Claim 7** \( \forall X \in X_{d₀,c}, l \in L_{d₀} \text{ cost}(FLNN, X, l) \leq UB_{d₀,c} = c = LB_{d₀,c}. \)

**Proof:** Take an arbitrary minimum-\(d₀\)-cover of \(X, C_{d₀}(X)\). Let the i’s candidate be a target so that \(d(xᵢ, xⱼ) > d₀\) for every distractor \(j\). Let \(C\) be a covering element \((C \in C_{d₀}(X))\) so that \(xᵢ \in C\). Note that all candidates in \(C\) are targets. Excluding \(C\), there are \((c - 1)\) other covering elements in \(C_{d₀}(X)\) with diameter \(\leq d₀\). Since \(C\) contains a candidate whose distance from all distractors > \(d₀\), FLNN does not query two distractor-candidates in one covering element (whose distance \(\leq d₀\)), before it queries at least one
candidate in $C$. Therefore, a target will always be located after at most $c$ queries. (It is possible that a target, which is not in $C$, will be found before. The algorithm then stops even earlier.)

The minimum cover size, $c_{d_0}(X)$, can be up to $2^m$ times smaller than $\text{populated}(X, d_0)$, and always not above it. Therefore, Claim 7 is an improvement over Claim 6. Nevertheless, while $\text{populated}(X, d_0)$ can be calculated easily, computing $c_{d_0}(X)$ can be hard for a large set of candidates. The problem of finding the minimum cover is NP-hard. Gonzalez [41] proposed a 2-approximation algorithm for the problem of clustering a data set minimizing the maximum inner-cluster distance, and proved that it is the best approximation possible if $P \neq NP$.

### 3.6 Illustration of the COVER Results

We demonstrate the latest result using a toy example illustrated in Figure 3.1. (More realistic examples are described in Section 3.7.) The image contains seven different insects, which were roughly segmented using morphologic operations. For simplicity of demonstration, the search uses only the objects’ area as a partial description (feature); see the distribution of the objects in one-dimensional feature space in Figure 3.1c.

Taking the ladybug as the target, the value of the COVER measure is 3; see Figure 3.1d. This means that using the FLNN algorithm, the ladybug is detected after examining at most two other objects.

Note that a. the COVER and the implied search difficulty depend on the

---

1Consider a corner of one of the covering hypercubes discussed in the proof to Claim 5. This corner is also a corner of $2^m - 1$ other neighboring hypercubes. Imagine there is one candidate in each of these $2^m$ cubes, very close to that corner. In such a case $\text{populated}(X, d_0)$ is $2^m$ while $c_{d_0}(X)$ is 1.
partial description (features), and b. neither the COVER value nor the search time changes if more distractors (butterflies, ants) or targets (ladybugs), similar to the existing ones, are added.

### 3.7 Experiments

The first set of experiments considers several search tasks and focuses on their characterization using the proposed metric cover. Because calculating the COVER measure is computationally hard, we suggest several ways to bound it from above and below, and show that combining these methods yields a very good approximation. In this context we also test the FLNN algorithm and demonstrate its guaranteed performance. Finally, we provide the intuition explaining why indeed harder search tasks are characterized by larger covers.

The first three search tasks are built around the 100 images corresponding to the 100 objects in the COIL-100 database [56] in a single pose; see Figure 3.2a. We consider these images as candidates extracted from some larger image. The extracted features are first, second, and third Gaussian derivatives in five scales [68] resulting in feature vectors of length 45; see Figure 3.2c. A Euclidian metric is used as the feature space distance. The tasks differ in the choices of the target that were cups (10 targets), toy cars (10 targets) and toy animals (7 targets) in the three search tasks. The COVER value for every task is bounded as follows: First the minimal target-distractor distance, $d_T$, is calculated. We developed a greedy heuristic algorithm that prefers sparse regions and provides a possibly non-tight but always valid $d_T$-cover; For all tasks, this algorithm provided smaller (tighter) covers than those obtained with a 2-approximation algorithm [41]. Both algorithms pro-
vide upper bounds on the size of the minimal cover; see Table 3.1 for cover sizes. Being a rigorous 2-approximation, half of the latter upper bound value (42/2 = 21 for the cups) is also a rigorous lower bound on the COVER measure. Another lower bound may be found by running the FLNN algorithm itself, which, by Theorem 1, needs no more than COVER queries to the oracle. By running the algorithm 100 times, starting from a different candidate each run and taking the largest number of queries required, we get the tightest lower bound; see Table 3.1 where the average number of queries required by the FLNN is given as well. Note that the search for cars was the hardest. While the car targets are very similar to each other (which should ease the search), finding the first car is hard due to the presence of similar distractors ($d_T$ is small). The cups are also similar to each other, but are dissimilar to the distractors, implying an easier search. On the other hand, the different toy animals are dissimilar, but as one of them is very dissimilar from all candidates, the task is easier as well. Note that the COVER measure captures the variety of reasons characterizing search difficulty by a single scalar.

We also experimented with images from the Berkeley hand segmented database [53] and used the segments as candidates; see Figure 3.3. Small segments are ignored, leaving us with 24 candidates in the elephants image and 30 candidates in the parasols image. The targets are the segments containing elephants and parasols, respectively. For these color images we use color histograms as feature vectors. In each segment (candidate), we extract the values of $\frac{b}{r+g+b}$ and $\frac{r}{r+g+b}$ from each pixel, where $r$, $g$, and $b$ are values from the RGB representation. Each of these two dimensions is divided into 8 bins, resulting in a feature vector of length 64. Again, we use a Euclidean metric for distance measure. (Using other histogram comparison methods, such as the ones suggested in [79], the results were similar.) See the results in
Table 3.1 and Figure 3.4. Although the mean results are usually not better than the mean results of a random search, the worst results are much better.

### 3.8 Discussion

In this chapter we have suggested inherent limitations of visual search. The COVER measure which quantifies the search difficulty combines both the targets/non-targets similarity and the non-targets/non-targets similarity, suggested in [28] as the qualitative characteristics determining human visual search performance. The quantitative measures provided here allow not only to measure difficulty but also to predict the search time. Not surprisingly, our results easily recognize the two extreme situations of ‘pop-out’ and ‘sequential’ searches, while locating each search task in a point on a continuous axis between these two poles.

While the study of feature selection is usually in the context of object recognition, here we have shown that it is very important for visual search as well.

Rosenholtz [71] has suggested the target saliency measure in the context of human visual search. Her measure is very related to this work as it is also inspired from Duncan and Humphreys’s [28] work, and involves the factors of target-nontarget similarity and nontarget-nontarget similarity. However, while we set here mathematical proofs for the validity of the COVER measure and its connection to the ability of search algorithms, the target saliency measure is validated by psychophysical experiments. Chapter 6 provides a comparison between the two measures in that context.

We see the results suggested here as a first step for quantifying the performance of visual attention directing algorithms. In this chapter we have
Figure 3.1: An illustration of the COVER upper bound. (a) Input image. (b) Segmentation results. (c) The representation of candidates in 1-dimensional feature space, with the objects’ area serving as the feature. (d) A $d_T$-cover associated with the ladybug being a target.

![Figure 3.1](image)

Figure 3.2: (a) The objects from the COIL-100 database [56] used as candidates. (b) The three model images used for demonstrating the extension of VSLE using top-down information. (c) The Gaussian derivatives filters used for extracting a partial description of the candidates. First two rows: first derivatives in $0^\circ$ and $90^\circ$ in different scales; three successive rows: second derivatives in $0^\circ$, $60^\circ$ and $120^\circ$; last four rows: third derivatives in $0^\circ$, $45^\circ$, $90^\circ$, and $135^\circ$.

![Figure 3.2](image)
Figure 3.3: The *elephants* and *parasols* images taken from the Berkeley hand segmented database [53] and the segmentations used in our experiments (color images).

<table>
<thead>
<tr>
<th>Search task</th>
<th># of cand.</th>
<th># of targets</th>
<th>FLNN worst</th>
<th>FLNN mean</th>
<th>Heuristic cover size</th>
<th>2-Approx. cover size</th>
<th>Real cover size</th>
<th>VSLE worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>cups</td>
<td>100</td>
<td>10</td>
<td>18</td>
<td>8.97</td>
<td>24</td>
<td>42</td>
<td>21-24</td>
<td>15</td>
</tr>
<tr>
<td>cars</td>
<td>100</td>
<td>10</td>
<td>73</td>
<td>33.02</td>
<td>79</td>
<td>88</td>
<td>73-79</td>
<td>39</td>
</tr>
<tr>
<td>toy animals</td>
<td>100</td>
<td>7</td>
<td>22</td>
<td>9.06</td>
<td>25</td>
<td>42</td>
<td>22-25</td>
<td>13</td>
</tr>
<tr>
<td>elephants</td>
<td>24</td>
<td>4</td>
<td>9</td>
<td>5.67</td>
<td>9</td>
<td>11</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>parasols</td>
<td>30</td>
<td>6</td>
<td>6</td>
<td>3.17</td>
<td>8</td>
<td>13</td>
<td>7-8</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3.1: Experiment results for FLNN and COVER. The real value of the COVER measure is bounded from below by ‘FLNN worst’ and half of the ‘2-Approx. cover size’, and bounded from above by the ‘Heuristic cover size’ and the ‘2-Approx. cover size’. The rightmost column shows that VSLE improves the results of FLNN for finding the first target.

Figure 3.4: Graphic representation of FLNN and COVER experiment results. The results in Table 3.1 are normalized by the number of candidates to enable comparison between the difficulty of different search tasks. The vertical lines describe the results of all FLNN runs, the dots being the best, mean and worst results. The x describes the heuristic calculated COVER.
provided predictions on the number of calls to the recognition oracle required before finding the first target, and provide a worst case analysis. Other costs associated with finding more targets, considering nonuniform computational costs, or suggesting a statistical analysis, are left for future work.
Chapter 4

VSLE (Visual Search using Linear Estimation): A Dynamic Stochastic Visual Search Algorithm

4.1 Introduction

In this chapter we suggest a somewhat different and stochastic framework for dynamic visual search, as well as a specific algorithm (VSLE). The identity labels of the candidate sub-images are modeled as correlated random variables. Interactions with a recognition oracle and using its responses on previously attended candidates enables to estimate the values of the labels of the non-yet attended candidates, and to (greedily) choose the best candidate given the knowledge provided so far.

The VSLE can be considered as an extension of the FLNN algorithm suggested in the previous chapter. We start by suggesting why there is place for
such an extension (Section 4.2). We describe the statistic modeling and the
dynamic framework in sections 4.3 and 4.4. The Linear Estimation method
and the VSLE algorithm are described in sections 4.5 and 4.6. An extension
to VLSE that suggests to combine also top-down information given by mod-
els identity is described in section 4.7. In the experiments section (Section
4.8) we first show the relation between the algorithms’ performance and the
tasks’ difficulty, and continue by demonstrating the stochastic algorithm’s ef-
fectiveness, both by itself and in combination with Viola and Jones algorithm
(denoted VJA) [90].

### 4.2 The Need to Extend FLNN

Our search strategy is based on the assumption that a target is likely to be
different from all distractors. In chapter 3 we took a deterministic approach
and proposed the FLNN algorithm that indeed prefers the candidate that
is maximally different from the nearest recognized distractor. FLNN stops
after finding the first target. Extending it so it will continue and find more
targets after the first is located, introduces some problems. According to
our assumptions, the consequent targets should be close (in feature-space
distance) to the already found targets, and far from the attended distractors.
It is not self-evident how to quantify these distances and how to combine
them into one value indicating the likelihood of a candidate to be a target.
A second weakness of the FLNN algorithm is that it lacks robustness; a single
“error” in the form of an attended distractor close to an undetected target
will reduce the priority of this target and slow down the search.
4.3 Candidates Labels as Dependent Random Variables

Taking a stochastic approach, we model the object identities as binary random variables with possible values 0 (for non-target) or 1 (for target). These values are initially unknown, and are revealed one by one by the recognition oracle. Estimates of these variables may be available and will help in directing the search.

Intuitively, we know that objects associated with a similar identity (or category) tend to be visually more similar than objects that have different identities. Quantifying this intuition in a probabilistic way, we suggest that the random variables associated with two candidates are more dependent if the visually similarity between these candidates is higher. Specifically, we characterize the dependency behavior as follows:

**Basic probabilistic assumption: Similarity dependent stochastic dependency**

The covariance between two labels is a monotonic ascending function of their visual similarity and a descending function of the feature-space-distance between them,

\[ \text{cov}(l_i, l_j) = \gamma(d(x_i, x_j)) \]

where \( l_i \) and \( l_j \) are the labels of candidates \( i \) and \( j \), \( d(x_i, x_j) \) is the feature space distance between the two candidates, and \( \gamma \) is a monotone descending function.

In Section 4.8.1 we experimentally challenge this assumption and find that, indeed, such a monotonic dependency is found in most cases. From the
experiments we found that an exponentially descending $\gamma$ is a good approximation for the actual dependency in many cases.

This assumption is one way to quantify the intuition presented above. The advantage in knowing the second order statistics is that it allows us to infer unknown labels from known ones, using common least squares estimation techniques.

### 4.4 Dynamic Search Framework

We propose the following greedy approach to dynamic search. At each iteration, estimate the probability of each unlabeled candidate to be a target using all the knowledge available. Choose the candidate for which the estimated probability is the highest and apply the object recognition oracle to the corresponding sub-image.

Formally, after the $m$–th iteration, $m$ candidates with partial descriptions $x_1, x_2, \ldots, x_m$, have already been attended and $m$ labels, $l_1, l_2, \ldots, l_m$ are known. We use them to estimate the conditional probability of the label $l_k$ of each unlabeled candidate $k$ to be 1:

$$p_k = p(l_k = 1 \mid l_1, \ldots, l_m).$$

### 4.5 Minimum Mean Square Error Linear Estimation

Now, note that the random variable $l_k$ is binary and, therefore, its expected value is equal to its probability of taking the value 1. Estimating the expected value, conditioned on the known data, is generally a complex problem and requires knowledge about the labels’ joint distribution. We chose to use a
linear estimator minimizing the mean square error criterion, which needs only second order statistics. Given the measured random variables \( l_1, l_2, \ldots, l_m \), we seek a linear estimate \( \hat{l}_k \) of the unknown random variable \( l_k \),

\[
\hat{l}_k = a_0 + \sum_{i=1}^{m} a_i l_i,
\]

which minimizes the minimum mean square error \( e = E((l_k - \hat{l}_k)^2) \). This is a standard task with a known solution [64]:

\[
\hat{l}_k = E[l_k] + \vec{a}^T (\vec{l} - E[\vec{l}]),
\]

where \( \vec{l} = (l_1, l_2, \ldots, l_m) \) and \( \vec{a} = R^{-1} \cdot \vec{r} \), where \( r_i, i = 1, \ldots, m \), \( R_{ij}, i, j = 1, \ldots, m \) are given by \( R_{ij} = \text{cov}(l_i, l_j) \) and \( r_i = \text{cov}(l_k, l_i) \).

The estimated label \( \hat{l}_k \) is the conditional mean of a label \( l_k \) of an unclassified candidate \( k \), and, hence, may be interpreted as the probability of \( l_k \) to be 1:

\[
p_k = p(l_k = T \mid l_1, \ldots, l_m) \sim \hat{l}_k.
\]

A similar linear estimation framework was suggested before by [32] for image sampling and by [49] for selectively choosing the training sample for a classification task.

### 4.6 The VSLE Algorithm

The VSLE algorithm is described in figure 4.1. Note that: 1) The extraction of the candidates, and the choice of the feature vector and the feature space distance may depend on the application (see Section 5.4). 2) The first candidate may be chosen based on different considerations. Choosing, for example, the first candidate from a large subset of similar candidates has the potential to reduce the number of potential candidates significantly, and
The VSLE Algorithm

- Extract the set of feature vectors $X = \{x_1, x_2, \ldots, x_n\}$.
- Calculate pairwise feature space distances and the implied covariance for each pair.
- Select the first candidate/s randomly (or based on some prior knowledge).
- In iteration $m + 1$:
  - For each candidate $k$ out of the $n - m$ remaining candidates, estimate $\hat{l}_k \in [0, 1]$ based on the known labels $l_1, \ldots, l_m$.
  - Query the oracle about the candidate $k$ for which $\hat{l}_k$ is maximum.
  - If enough targets are found - abort.

Figure 4.1: The VSLE algorithm

---

therefore, may be the best choice, if the prior probability of all candidates to be a target is uniform. This, however, seems to be an unjustified assumption as targets usually do not appear in large groups. (If they did, then saliency would never work.) 3) The estimation of the label associated with any unknown candidate is accelerated only if its nearest (classified) neighbors are used. As the covariance decreases with distance, ignoring far neighbors may be justified. (For example, in the experiments described in Sections 4.8.2 and 4.8.3 we use only the $k = 3$ nearest classified neighbors for each estimation.)

4) If the non-targets are similar, and the target differs significantly from them, we get a pop-out like behavior, and the target is found after one or two iterations. 5) If the non-targets consist of $k$ clusters in the feature space, then at most $k$ candidates are tested before the target is attended. If the clustering is not strong, performance smoothly degrades and more attempts are required before the target is found. 6) If the targets themselves are one cluster, then after the first is detected, the rest follow immediately.
4.7 Combining Prior Information

Bottom-up and top-down information may be naturally integrated by specifying the prior probabilities (or the prior means) according to either the saliency or the similarity to known models. Moreover, if the top-down information is available as $k$ model images (one or more), we can simply add them as virtual candidates that were already examined by the oracle and got a ‘yes’ answer. Continuing the search from this point is likely to be faster. As this is not the focus of this work, we do not elaborate. See, however, a demonstration in Section 4.8.1.

4.8 Experiments

4.8.1 Initial Demonstration

The VSLE algorithm was implemented and applied to the same five visual search tasks described in Section 3.7. We model the dependencies of the covariance on the feature-space-distance using $\text{cov}(l_1, l_2) = \gamma(d(x_1, x_2)) = \mu(1 - \mu)e^{-d(x_1, x_2)}$, where $l_i$ is the label of candidate $i$, $d(x_i, x_j)$ is $x_i$ and $x_j$ feature-space distance, and $\mu = E(l_i)$. The results are described in Figures 4.2a to 4.2e. The solid lines in the graphs describe one typical run. Other runs, starting each time from a different candidate, are described by the size of the gray spots as a distribution in the (time, number of target found) space.

Unlike FLNN, which deals only with finding the first target, VSLE continues and aims also to find the other targets. Moreover, in almost all the experiments we performed, VSLE was faster in finding the first target (both in the worst and the mean results). See the rightmost column in Table 3.1.
In all above experiments the candidates of each task can be divided into a few weakly homogeneous groups (or clusters), which seems to be the situation in many natural scenes as well. To compare between the results of the experiments qualitatively, we focus on the two following difficulty factors: \(d(T,NT)\)-relative feature-space-distance between targets and non-targets, and \(d(T,T)\)-relative feature-space-distance between targets and targets. This qualitative description is summarized in Table 4.1.

VSLE relies on the covariance between candidates’ labels. We use a covariance function that depends only on feature-space-distance, and argue that for many search tasks this function is monotonically descending in this distance. To verify this assumption we estimate the covariance of labels vs. feature-space-distance of search tasks. Given a search task, i.e., the set of candidates and their labels, we compute the distances between each pair of candidates. The distances are sorted and divided into \(h\) intervals. (Non-uniform intervals containing an equal number of target-target pairs resulting less erratic estimates.) The labels covariances \(\gamma_i\) \(i = 1, \ldots, h\) are estimated separately for every interval \(I_i\) for the five search tasks described in Section 3.7 and plotted as a function of the feature space distance (Figure 4.3). Except for the task of finding toy animals (in which there is almost no resemblance among targets), all other results show a monotonically descending behavior.

Using the method suggested in Section 4.7, we incorporate top-down information and demonstrate it on the toy cars case: three toy cars that do not belong to the COIL-100 database are used as model targets; see figure 3.2b. The search time was significantly reduced as expected; see Figure 4.2f.

We experimented with a preliminary version of integrated segmentation and search. An input image (see Fig.4.4) was segmented using k means clustering in the RGB color space (using 6 clusters). All (146) connected
Figure 4.2: VSLE results for the COIL100 [56] images and Berkeley hand segmented [53] images. The solid lines describe one typical run. Other runs, starting each time from a different candidate, are described by the size of the gray spots as a distribution in the (time, number of targets found) space. Figures (a) to (c) are results for the candidates from the COIL-100 database. In (a) the cars are the targets; the first car is quite hard to find. Once it is found, the rest are detected quickly. In (b) cups are the targets. It is easy to find the first cup since most of the cups are different from the non-targets. Most of the cups resemble each other and follow pretty fast, but there are two cups (one without a handle and one with a special pattern) that are different from the rest of the cups, and are found rather late. In (c) the toy animals are the targets. It is easy to find the first animal since one animal differs significantly from all non-targets. There is no special resemblance among the different toy animals, therefore, the rest of the search proceeds similarly to a random search procedure. Figures (d) to (e) are results for the Berkeley hand segmented color images. In (d) finding the first elephant is of medium difficulty since the color of the elephants is not too salient. Once the first is found, the rest of the elephants, having a similar color, follow. In (e) all the parasols are detected very fast, since their color is similar and differs from that of all other candidates. In (f) we demonstrate how VSLE can be improved by top-down information for the toy cars search task.
Table 4.1: Qualitative search performance dependencies on the target-target similarity and target–non-target similarity.

<table>
<thead>
<tr>
<th></th>
<th>d(T,T)</th>
<th>d(T,NT)</th>
<th>Finding first target</th>
<th>Finding successive targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>cars</td>
<td>small</td>
<td>small</td>
<td>hard</td>
<td>easy</td>
</tr>
<tr>
<td>cups</td>
<td>some small, some big</td>
<td>big</td>
<td>easy</td>
<td>easy to find some, hard to find all</td>
</tr>
<tr>
<td>animals</td>
<td>big</td>
<td>big</td>
<td>easy</td>
<td>hard</td>
</tr>
<tr>
<td>elephants</td>
<td>small</td>
<td>intermediate</td>
<td>easy</td>
<td>intermediate</td>
</tr>
<tr>
<td>parasols</td>
<td>small</td>
<td>big</td>
<td>easy</td>
<td>easy</td>
</tr>
</tbody>
</table>

Figure 4.3: Estimation of labels covariance vs. feature-space-distance, for COIL-100 images [56] and Berkeley hand segmented images [53].
components larger than 100 pixels served as candidates. The VSLE algorithm searched for the (7) faces in the image, using a feature vector of length 4: each segment is represented by the mean values of red green and blue and the segment size.

No prior information on size, shape color or location was used. Note that this search task is hard due to the presence of similarly colored objects in the background, and due to the presence of hands which share the same color but are not classified as targets. Note that in most runs six of the seven faces are detected after about one-sixth of the segments are examined. We deliberately chose a very crude segmentation, demonstrating that very good segmentation is not required for the proposed search mechanism.

4.8.2 VSLE Applied to the MIT+CMU Faces Database

As a more intensive validation of the VSLE algorithm, we tested it using the MIT+CMU database [73][78], which includes 130 images with 511 labeled frontal faces.

For these images, candidates were dilated bounding-boxes of image segments resulting from a simple image-pyramids based segmentation [19]. Each candidate is assigned with a range of scales, depending on its size, in which recognition attempts are performed. (i.e., when a recognition oracle is applied to a candidate sub-image, it searches only for faces that fit the size of the candidate) The similarities between candidates are estimated by distance between gray level histograms (see implementation details in Appendix A). Using a perfect oracle, we compared the detections times when serially passing through the candidates to the detection times when using VSLE to set the order of queries; see Figure 4.5a. As expected, the curve of the serial search is close to a linear curve, while VSLE detects targets faster. For instance,
if for each image we stop VSLE after checking 30% of the candidates, 72% of the targets are found. If we stop VSLE after checking 55% of the candidates, 90% of the targets are found; see Table 4.2a and Figure 4.6. Note that this improvement is achieved using very simple features, and no model-based information. Also note, that using this simple candidate selection process, we are able to recognize 487 out of the 511 faces in the database. This can be improved, of course, if the segmentation process was model-based rather than a process that connects blobs of similar gray level.

In order to check whether VSLE helps in early rejection of non-targets and does not only help in detecting successive targets after other targets are found, we separately checked its behavior on images that include a single face; see Figure 4.5b. Out of the 130 images, there are 58 such images. On average, the face is detected after checking 35% of the candidates.

4.8.3 Combining VSLE and VJA (Viola & Jones Algorithm [90])

Aiming to test the proposed attention mechanism with a real oracle, we combined the VSLE algorithm with the popular face detector algorithm proposed by Viola and Jones [90]. The latter algorithm is based on a cascade of AdaBoost classifiers, implemented efficiently, resulting in an extremely fast model based search algorithm.

We used the same candidates selection and similarity measures described in Section 4.8.2, and sent the resulting candidates, one by one according to the order specified dynamically by VSLE to the VJA algorithm used as the recognition oracle. We used Intel’s OpenCV library [48] (see implementation details in Appendix B). The number of detected targets (faces) and the number of false detections are plotted in Figure 4.7 (solid lines) as a function
Figure 4.4: VSLE applied on an automatic-color-segmented image to detect faces. (a) The input image (colored image) (b) Results of an automatic crude color-based segmentation (c) VSLE results (see caption of figure 4.2 for what is shown in this graph).

Figure 4.5: VSLE vs. serial search applied to the MIT+CMU database. Using perfect oracle. (a) Applied to all the database. (b) Applied only to single face images.

Table 4.2: VSLE applied to the MIT+CMU database. Demonstrates the percentage of targets found if VSLE is stopped at different times.
of time and compared to the results of applying the VJA over the full images.

Note that in the initial stage of VSLE, some time is spent on the segmentation process (about 12 sec) and in calculating the similarities (about 5 sec). Then it finds targets much faster than the serial VJA, until a certain point, when the remaining targets are dissimilar to those already found, slowing the search.

To compare the combined VSLE and VJA to the original VJA, the graphs in Figure 4.7 are plotted against time and not against the number of candidates (as was the case in previous experiments). Note that this makes the change in priorities, caused by the VSLE, less visible because the VJA (using its rejection cascade) spends more time in processing targets than on the processing of most non-targets; see Table 4.2b. Still, the curve is much above a linear curve, and at a certain range above the curve of the original VJA as well.

The results suggest that the proposed combination is preferable (over the VJA) if the search time is limited to 75% or less of the VJA run time. Note that the advantage of the combined algorithm is somewhat better: with the already trained VJA we used, the false detections rate is lower with the combined method. For a fair comparison, we could train the independently used VJA for similar false detection rates. This would increase the VJA misses as well, but the change would not be substantial, as apparent from the ROC curve in [90].

### 4.9 Discussion

In this chapter we have considered the usage of inner-scene similarity combined with recognition feedback to facilitate object recognition and detection.
Figure 4.6: Some examples of the times faces were detected when applying the VSLE algorithm on the MIT+CMU database using a perfect oracle. The percentages above the images indicate the percentage of candidates investigated before VSLE was stopped.

Figure 4.7: Detection times of the Viola and Jones algorithm (VJA [90]), and of the combined implementation of VJA and VSLE, applied to the MIT+CMU database.
Similarity is useful for search because, intuitively, similar objects are more likely to share the same identity. The intuitive assumption was quantified taking a stochastic approach, allowing similar non-targets to be ‘rejected together’ and for similar targets to be detected successively.

While many visual search mechanisms rely only on top-down or bottom-up knowledge, inner scene similarities always helps, and may even become the dominant source of knowledge. Consider, for example the parasols search task example (Figures 3.3 and 4.2e). Note that the targets encompass a significant fraction of the image, and therefore, cannot be salient. The parasols are similar to each other and different from the non-targets in their color, but if this color is unknown, they cannot be searched using top-down information.

As expected, using inner-scene similarity allows the majority of targets to be found earlier than when using a sequential search. This was experimentally verified to be the case even in comparison to a very efficient detection algorithm. While the performance of the combined algorithm (VSLE+VJA) and that of the VJA by itself seems comparable, we expect that harder search tasks involving, say, objects from several classes and objects associated with a more visually varying appearance would make the advantage of combined algorithms higher. Such tasks require a more complex recognition oracle (e.g., [37]), which is not likely to be implementable with the extreme efficiency associated with rejection based methods, and in particular, VJA. Then, investing more in similarity evaluation and in ordering would save more time.
Chapter 5

Esaliency (Extended Saliency): A Stochastic Attention Model
Quantifying (Relaxed) Global Saliency.

5.1 Introduction

In this chapter we present the Esaliency (Extended saliency) algorithm. By incorporating inner-similarity information with the common expectation for a relatively small number of targets in the scene and with the observation that the content of the image has a relatively clustered structure, we suggest a new method for quantifying the saliency of an image region. The algorithm uses a graphical model approximation that allows the hypotheses for target locations with the highest likelihood to be efficiently revealed. We show that using the fixation order dictated by this (descending) saliency makes the whole process of object recognition, or object detection, much more efficient.
Both the Esaliency and the VSLE algorithms share the *object-based* (or *region-based*) framework, by starting with a segmentation-based candidate selection. They also share the stochastic modeling of candidates similarity vs. identity of their labels. However, while VSLE implements a supervised search mechanism, optimizing the interaction between search and recognition, the *Esaliency* algorithm proposed here is purely bottom-up, and, like previous saliency algorithms (e.g. [45]), neither assumes the availability of object recognition oracle nor changes the saliencies after they are set.

As described in chapter 1, the vast majority of attention modeling identify saliency with local exception. That is, the saliency value at each location is essentially the local contrast in one feature or more. They do not check uniqueness in the context of the whole scene. An object with many appearances can be considered salient if each appearance is contrasting with the background, while another object which has only one or a few appearances can be considered as less salient if it is less contrasting with the background. Therefore, while this approach may be biologically plausible, it is suboptimal for computer vision. Esaliency takes a global approach by considering a region as salient if there are only a few (or none) other similar regions in the scene. Then if, for example, a locally-salient object appears many times in the image, its saliency is reduced. On the other hand, if a few similar objects appear in neighboring locations, they will be considered salient if the total number of appearances is small.

The proposed *extended saliency* (or *Esaliency*) algorithm differs significantly from previous methods also as the motivation, the methodology and the end result are completely different: rather than trying to build a model explaining human attention, the proposed process uses a validated stochastic model to estimate the probability that a candidate is of interest, in a
mathematically well-defined sense. Referring to this probability as saliency specifies a saliency with meaningful values, not only by comparing them to saliency values in other places, but also by themselves. Note also that with this saliency, the common practice of scanning the candidates with decreasing saliency order is not only a very reasonable heuristic but is also an optimal strategy for minimizing the expected scanning time until an object of interest is found.

We are aware of only a few related attention mechanisms as most attention mechanisms are feature-based. A computer vision implementation of Duncan’s [27] object-based attention model was suggested in [77]. To be effective it requires however a high quality hierarchal segmentation (and indeed uses, in the experiments, man-made segmentations), and therefore seems non-practical for attention of complex natural scenes.

Saliency as a measure of being global exception was already used in computer vision. In the context of feature point location, Walker et. al. [92] have suggested that salient feature-points are “those which have a low probability of being miss-classified with any other features”. Boiman and Irani’s [17] image (or video) analysis approach, considers a region non-interesting if it is similar to another image region. This principle is close to our approach, however, it differs in several important issues: First, it does not rely on an explicit stochastic model regarding the objects’ identities. Then, it does not allow for two (or a few) similar items that together are globally unique to be considered as salient. Moreover, it relies on a relatively costly, part-based similarity evaluation. Therefore, while it gives very good results as an analysis tool, it does not seem useful as a quick pre-recognition attention mechanism.

The basic assumptions and their quantification are presented in Section
5.2. The Esaliency algorithm is described in Section 5.3. Section 5.4 includes the description of experiments that test the algorithm on a few data sets of natural scenes, compare its results to those of a feature-based attention method [45], test a few possible extensions for the basic algorithm, evaluate Esaliency compared to human fixation maps, and test the benefits of combining Esaliency for the task of pedestrians detection. Section 5.5 concludes.

5.2 Framework and Underlying Assumptions

The bottom-up attention process proposed in this chapter is region-based. That is, it builds on a pre-attentive grouping process, dividing the image into segments, which are the candidates for attention. The attention process associate each candidate with a quantitatively meaningful saliency value, which is an estimate of its likelihood to be a target.

The segmentation process need not be accurate and can actually be very rough, resulting in fragmented objects. In fact, even with the best available segmentation algorithms, setting the parameters so that an oversegmentation results is the only way to ensure, with high probability, that most objects are not split between segments. A (segment) candidate is denoted “target” if it corresponds to an object of interest or to a part of it. We consider realistic scenes where several objects of interest may be present. This, and the inevitable object fragmentation, imply that several candidates may be targets.

Unlike the VLSE algorithm where targets were associated with objects of the same category, in the Esaliency context objects of interest of different types can be “targets” allowing to set a high saliency to both.
The dominant approaches to saliency (largely based on the feature integration theory and on the implied center-surround mechanisms) essentially look for a local exception. This approach may fail in two ways: First, the presence of several targets in nearby locations may reduce the response to each and cause the attention mechanism to miss them. Moreover, a large set of similar objects (or even just regions), which are, by definition, non-exceptions, may be falsely detected as salient if they are associated with a high local contrast. Then the true exceptions, associated with a lower contrast, are missed. For instance, consider the simple synthetic case of ten red disks and one yellow disk scattered over a white background. Although it is obvious that the yellow disk is the exception, locally-salient seeking models may suggest each of the red disks as more salient as they contrast more with the background.

Our Extended saliency approach intuitively seeks for relaxed global exceptions. That is, Esaliency prefers objects that belong to small groups of similar objects that are relatively dissimilar to the rest of the image. For the colored disks example, described above, it would recognize the yellow exception. Moreover, if, for instance, there were three yellow disks, each of them would still be recognized as the most ‘visit-worth’ item in that display.

5.2.1 Modeling Object Identity by a Stochastic Approach

Our approach to the design of the saliency algorithm, is to quantify the object identity in a probabilistic model, which would eventually identify the saliency with a probability to be a target.

Formally, let \( (c_1, ..., c_n) \) denote the candidates for attention. Taking a stochastic approach, we consider the labels (or identities) of the candidates
$l_1, \ldots, l_n$ as binary random variables, which take value 1 if the candidate is a target and the value 0 if it is a non-target. Estimating the probability $P(l_i) = p(l_i = 1)$, that the candidate is a target, is the goal of this work. This probability is the proposed saliency.

To estimate this probability, we take an indirect approach and start with the corresponding joint distribution. Let $\bar{l} = (l_1, \ldots, l_n)$ denote a vector of candidate identities, and $\mathcal{L} = \{\bar{l} = (l_1, \ldots, l_n); l_1, \ldots, l_n \in \{0, 1\}\}$ be the set of all $2^n$ identity vectors. Let $p(\bar{l})$ be a probability distribution function on $\mathcal{L}$.

We shall now make some observations which constrain the distribution $p(\bar{l})$. Then, in section 5.3, we propose a specific distribution satisfying these constraints, and a computational efficient way to estimate it. Finally, the probability for every particular candidates to be a target is estimated by finding some joint assignments with highest likelihood.

5.2.2 Underlying Observations

To construct the Esaliency process we combine the following observations:

**Observation 1: The number of target candidates is usually small.**

The number of interesting objects in a scene, and the total area that they cover, is usually relatively small.

**Observation 2: Visual similarity and identity are correlated.** This observation was the basis to the VSLE algorithm. In the Esaliency context, two visually similar candidates are likely to be both of interest or both of no interest. If the candidates are dissimilar then independently, every one of them may be of interest or of no interest, allowing two dissimilar candidates to both be considered as targets.
Observation 3: Natural Scenes are often composed of clustered structural units. We argue that often, natural images may be partitioned into small parts which are clustered in some feature space. That is, the feature vectors characterizing the parts are not uniformly distributed in the feature space but are rather concentrated in some places.

These properties do not hold for every scene, but they do for many of them. We found that using them as the basis for the proposed stochastic mechanism enabled it to direct the attention focus on interesting objects and reject most of the image background, even when it is highly textured.

Note that combining the two last properties implies that every cluster is associated with candidates of the same target/nontarget identity. Note also that adding the first one implies that the targets are in small cluster(s).

A direct usage of these observations would lead to a simple attention algorithm: start by clustering and then choose the smallest resulting clusters. This approach, however, has some drawbacks: First, clustering requires some knowledge for setting the number of clusters or the maximal cluster diameter. Secondly, clustering is a hard decision, and a wrong clustering can change the saliency of certain candidates significantly. Lastly, it is not clear how to assign continuous saliency values or even just priorities, to the members of the selected small clusters, and in particular how the distances within the clusters and between them influence this assignment. Our approach avoids this difficulties by constructing a distribution on the possible target/nontarget joint assignment, which may be viewed as a “soft” clustering.

In the rest of this section, we provide some evidence to these observations and quantify in the context of the proposed stochastic model.
The number of target is usually small

Spatial attention is needed mostly in the context where a small part of the scene matter more than its other, larger, part. Then a sequential (or random) search would not be efficient for finding the important part. This is the case for many natural scenes: the number of interesting objects as well as the fraction of the image they occupy is small.

Typical images of natural scenes taken from the University of Washington Ground Truth Database [2], were, in our experiments, over-segmented on the average to 306 regions. The number of segments associated with objects such as people, cars, or animals were on the average 12.3 or about 4%. Therefore, property 1, suggesting that the number of targets is likely to be relatively small, is justified.

In the stochastic context this observation is simply expressed as a relatively low expected value $\mu_i$ of every random variable, $l_i$. When no knowledge about the size, location, and the properties of targets, and no general knowledge about the scene, is available, the probability of a candidate to be a target is uniformly set as $\mu_i = \mu$. In our basic implementation of Esaliency we set $\mu = 0.05$. We show in the experiments section that this setting is a reasonable description of natural scenes. Moreover, we found that the Esaliency algorithm is not sensitive to the exact value of $\mu$, as long as it is relatively small.

The uniform setting of priors may be modified when some candidates are preferred, due to say high local saliency. Another preference, to objects closer to the image center, is found in biological attention mechanisms [20] and is present also in images produced by a human photographer. A location based preference may be learned for a specific set of images. See Section 5.3.3 and 5.4.4 for related experiments.
Visual similarity and identity are correlated

This observation was already tested and quantified in the context of the VSLE algorithm (chapter 4). For the self-sufficiency of this chapter, the quantification of the observation is repeated here with the appropriate context, and a further and more intensive test of its validity is presented, now allowing ‘targets’ in one image to belong to different categories.

Let $d_{ij}$ denote the feature-space distance between the two vectors associated with the $i$-th and the $j$-th candidates. The dependency between the corresponding two labels $l_i, l_j$ is modeled by the correlation coefficients as a descending function $\gamma$ of $d_{ij}$,

$$\rho(l_i, l_j) = \frac{\text{cov}(l_i, l_j)}{\sqrt{\text{var}(l_i)\text{var}(l_j)}} = \gamma(d_{ij}),$$

(5.1)

We took a set of images (156 images from the Washington ground truth database [2]), and asked naive observers to mark the significant objects in the scene (people, cars, etc.). We then measured the correlation coefficients between the target-nontarget identity of such pairs as a function of the feature-space-distance (using segmentation, features and similarity measure described in section 5.4.1. See figure 5.1a. Note that, as expected, the
correlation is higher for similar candidates (low feature space distance) and decreases for decreasing similarity. Experimenting with several natural scene images, we found that a piecewise linearly descending function, shown in Figure 5.1b, is a good approximation of the measured correlation coefficient behavior for several choices of feature space and metrics. This dependency model is used in all the experiments.

Natural Scenes are often composed of clustered structural units

We argue that segments-candidates from one image are clustered. That is, their feature vectors may be divided into a small number of subsets so that all vectors in the same subset are close. This observation is actually intuitive: typical scenes are associated with a limited palette of colors, related to the type of the scene (e.g. urban or landscape), the season, the particular type of trees, flowers or animals in it, etc. (e.g., [88, 91]). A scene is usually associated with a small number of textures, as evident from the success of segmentation methods that rely on texton clustering (e.g., [52]).

To demonstrate the clustering property we carried out a simple experiment comparing the clustering within specific images to clustering within a set of different natural images. For each image, we applied the same segmentation and feature extraction process as in the Esaliency algorithm implementation. This gave a 7-dimension feature vector for each candidate of each image. (see section 5.4.1 for details.)

For a given set of feature vectors, we estimated their inclination to cluster as follows: The data was clustered to a mixture of multi-variate Gaussians using the EM algorithm [26] following the method suggested in [23]. The number of clusters $k$ was specified in the range 1–10. The EM algorithm was repeated ten times for each $k$ value, with different initializations, and for each
Figure 5.2: Clustering Analysis: candidates from one image are more clustered than candidates randomly chosen from various images. Results on 156 images from the Washington University natural scenes database [2]. See section 5.2.2 for details.
For this experiment, we again used the same image set from the Washington database [2]. The clusterings were done first for sets of feature vectors coming from a specific image and then, for similarly sized sets of feature vectors randomly sampled from the full image set. Figure 5.2 describes the histograms of best MDL for the two cases, as well as the histograms of the associated number \( k \) and the likelihoods corresponding to the selected best clusterings. Clearly, when the data is taken from specific images we get a larger number of clusters, associated with much higher likelihood and a much lower MDL, corresponding to much narrower distributions, i.e. clusters. Data drawn from many diverse images is described best, on the other hand, by a few wide Gaussian and is much less clustered. (The distribution is a continuous density, making some of the description length negative, but this is acceptable as only relative length matter [70].)

This observation is not quantified and is directly used, in the next section for choosing the joint target/nontarget distribution.

5.3 The Esaliency Algorithm

The proposed saliency algorithm is based on the stochastic model and on the observations, presented in the previous section. As discussed above, the algorithm is essentially a method for estimating the probability that a candidate is a target. The Esaliency algorithm is summarized in Figure 5.3. The segmentation and the feature selection (Steps 1 and 3) are parts of the
The Esaliency Algorithm

1) Select candidates using some segmentation process.
2) Use the preference for a small number of expected targets (and possibly other preferences) to set the initial (prior) probability for each candidate to be a target.
3) Measure visual similarity between every two candidates and infer the correlations between the corresponding labels.
4) Represent the label dependencies by a Bayesian network.
5) Find the N most likely joint assignments.
6) Deduct the saliency of each candidate by marginalization.
7) Scan the image by the descending order of saliency.

Figure 5.3: The Esaliency algorithm

algorithm, but need not be the specific procedures we used in our experiments. The features chosen should be informative in the sense that they provide some rough but not necessarily robust distinction between the different types of objects in the scene. The distance (not necessarily Euclidean) between every two vectors $d_{ij}, \ i, j = 1, 2, \ldots, n$ is calculated and used to obtain $\rho(l_i, l_j)$ by Eq. 5.1.

In addition to the segmentation and to the similarity calculations, it contains two additional crucial stages: First, a distribution on the joint candidate labels is specified using a Bayesian model. Then, the individual probability of every candidate to be a target is deducted from the probabilities of some most likely joint assignments. These two components are described in detail below.

5.3.1 Specifying a Joint Label Distribution as a Bayesian tree

The pairwise correlations, calculated from the image similarities, together with the priors $p_0(l_i)$’s suggested in the previous section, can now be used to
specify a joint probability distribution, \( p(\bar{l}) \), on a binary hypothesis vector \( \bar{l} \).

There are, in principle, many distributions satisfying any given set of correlations (provided the covariance matrix is positive definite). We use a simple, tree-based, Bayesian network, which takes only the strongest correlations into account. This choice follows the third observation, that the feature vectors associated with the candidates are clustered, rather than distributed uniformly. With the second observation this means that for strongly clustered data, all the labels within the cluster are strongly correlated and that labels in different clusters are independent. Therefore, the dependence of a label of some candidate on all the other labels may be replaced by its dependence on another label in the same cluster. A second consideration for choosing the tree based distribution is that estimating more complex distribution necessarily requires higher order joint statistics between the candidates labels (e.g., \([24, 76, 31]\) ), which we do not have. Finally, as we shall see, efficient estimation of each candidate to be a target is possible with the tree based Bayesian network.

Other distribution choices seem to have severe disadvantages. The general joint Gaussian distribution seems attractive as it is the maximum entropy distribution for the 2nd order statistics. It ignores however the clustering (3rd)observation as well as the knowledge that the random variables are binary.

Let \( G \) be a graph with \( n \) nodes representing the random candidate labels and edges representing the pairwise dependency between the random variables. All the labels pairs associated with non-zero correlation are thus connected. The joint distribution of all labels may be written as a function of all joint distributions associated with cliques in this graph (see e.g., \([31]\) ). For general graphs this distribution is difficult to estimate or even to describe.
Deleting edges so that the resulting graph becomes a tree, leads to a simplified distribution, depending only on second order statistics. To get the best approximation, the edges of $G$ are weighted by the mutual information between the corresponding nodes, and the maximum weighted spanning tree of $G$ is selected. Then the resulting tree describes a distribution that is closest (by the Kullbak-Leibler divergence) to the original distribution (relative to all approximations by trees)[25, 65].

Calculating the mutual information

$$I(l_i, l_j) = \sum_{l_i=0,1} \sum_{l_j=0,1} p(l_i, l_j) \log \frac{p(l_i, l_j)}{p(l_i)p(l_j)},$$

for each pair of candidates, requires the probabilities of the joint events. Recall that every label $l_i$ is a binary r.v. with expected value $\mu_i$ and variance $\mu_i(1 - \mu_i)$. Then, a straightforward calculation, relying on Eq. (5.1), leads to the following joint probabilities:

$$p(l_i = 1, l_p = 1) = \gamma(d_{ij}) \sqrt{\mu_i(1 - \mu_i) \mu_j(1 - \mu_j) + \mu_i \mu_j}$$

$$p(l_i = 1, l_j = 0) = p(l_i = 1) - p(l_i = 1, l_j = 1) = \mu_i - p(l_i = 1, l_j = 1)$$

$$p(l_i = 0, l_j = 1) = \mu_j - p(l_i = 1, l_j = 1)$$

$$p(l_i = 0, l_j = 0) = 1 - \mu_i - \mu_j + p(l_i = 1, l_j = 1).$$

Given $I(l_i, l_j)$ as the weights, the maximum weighted spanning tree is found by the PRIM algorithm [67]. Choosing some node of this tree as a root $r$ makes it a directed tree (with no effect on the resulting distribution). The directed tree is converted into a Bayesian network as follows. For the root, $p(l_r = 1) = E[l_r] = \mu_r$. For each of the other nodes in the tree, two conditional probabilities should be set: $p(l_i = 1|l_p = 0)$ and $p(l_i = 1|l_p = 1)$, where $i$ is the index of the node and $p$ is the index of its parent:

$$p(l_i = 1|l_p = 0) = \frac{p(l_i = 1, l_p = 0)}{p(l_p = 0)} = \frac{p(l_i = 1, l_p = 0)}{1 - \mu_p}$$
\[ p(l_i = 1|l_p = 1) = \frac{p(l_i = 1, l_p = 1)}{p(l_p = 1)} = \frac{p(l_i = 1, l_p = 1)}{\mu_p}. \]

Finally, given a vector of labels \( \bar{l} = (l_1, \ldots, l_n) \), we may calculate its probability by

\[ p(\bar{l}) = p(l_r) \prod_{i=1,\ldots,n; i \neq r} p(l_i | l_{\text{par}(i)}), \tag{5.2} \]

where \( \text{par}(i) \) is the parent node of node \( i \) in the tree [65].

While the tree is indeed only an approximation to the true distribution, we found that it works well for the relatively clustered feature vectors in one image. In particular, the joint assignments that are associated with “1” value given to the members of small tight clusters are those with the highest probabilities.

### 5.3.2 Finding the joint assignments with highest likelihood and deducting the (E)saliency of each candidate

Intuitively, assignments giving a target value (1) to the members of small clusters and nontarget values (0) to other candidates, get, both by the qualitative model and by the Bayesian network, higher probabilities. With the tree-based graphical model, the \( N \) assignments associated with the highest likelihood \( L_{\text{best}} = \{\bar{l}^1, \bar{l}^2, \ldots, \bar{l}^N\} \) are found using Nilsson’s algorithm [57]. This algorithm uses exact inference to find the top \( N \) configurations and their likelihoods by a sequence of maximum propagations. For a general Bayesian network, this algorithm’s efficiency depends on the number of clicks in the network, multiplied by an exponent of the clicks’ size. However, for a tree based Bayesian network the complexity is \( N n \log(Nn) \) (where \( n \) is the number of nodes in the tree, or in our case, the number of candidates).
The saliencies are calculated by marginalization over the $N$ most likely assignments:

$$\hat{p}_T(c_i) = \sum_{\{\bar{l} \mid \bar{l} \in \mathcal{L}_{\text{best}}, l_i = 1\}} \hat{p}((\bar{l})),$$

(5.3)

We found that for scenes containing $100 - 500$ candidates, finding the 100 first most probable assignments was informative enough for directing attention to salient locations.

The order of attention is defined by the descending order of the calculated candidates’ saliencies $\hat{p}_T(c_i)$.

### 5.3.3 Some Simple Variations on the Proposed Saliency Mechanism

The proposed probabilistic model may be used to create several simple variations on the basic attention algorithm, described above.

**Global, non-extended, saliency**

One possible approach to saliency may be to look for global exceptions. That is, a single candidates that is globally unique. Note that with this approach, we need to consider only $n$ different label vectors for which exactly one candidate is a target (“1”) and the rest are non-targets (“0”). Thus the algorithm does not look for the N most likely hypotheses, but just simply evaluate the probability for the N one-target vectors using the Bayesian network and eq. (5.2). While this algorithm seems reasonable, it turns out that it is not as good as the Esaliency approach. (See section 5.4.3 for experiments).
Esaliency using learned expected value

A natural extension would be to use non-fixed expected value parameter $\mu$. We considered two versions. In the first version, $\mu$ is uniform but is set adaptively to a specific context. In the second $\mu$ is space varying, and is either estimated from training data or is just set higher in the image center according to some heuristic. See more details in section 5.4.4. We found that using the value of the uniform $\mu$ learned from a training set has almost no effect on the results, suggesting that the Esaliency algorithm is not sensitive to the value of $\mu$ as long as it is small. The learned preference and the preference to the center, however, improve the algorithms’ performance in many cases.

5.4 Experimental Evaluation of Esaliency

This section describes a comprehensive set of experiments, which illustrate the Esaliency algorithm and test it, quantitatively, on several data sets, and with respect to competing algorithms and to human attention.

We start by discussing some implementation issues and illustrating the idea behind Esaliency with a simple synthetic example. Then, we test the algorithm on 4 image sets of natural outdoor scenes. We compare the Esaliency’s performance to that of the feature based approach described in [45] using the iLab implementation [1]. Some variations of the algorithm are tested as well. The computational savings available from Esaliency in a complex detection task (pedestrian detection) are estimated in a separate experiment. Finally we make a preliminary attempt to relate the Esaliency algorithm to the human vision attention mechanism.
5.4.1 Implementation

All the experiments were carried out with the same attention implementation and with the same default parameters, unless otherwise stated. The candidates are specified by a simple, fast, multi-scale segmentation process [19]. We used its OpenCV implementation. Segments with bounding boxes’ widths and heights being between 2% and 20% of the image height are specified as candidates. Better segmentation algorithms may reduce search time, but on the other hand, could be computationally expensive themselves.

Each candidate (segment) is characterized by a short feature vector describing some simple properties: average color (R, G and B), dimensions of its bounding box, and its area relative to the area of its bounding box. Every feature is separately normalized to the [0, 1] range, and the color features are weighted by a factor of 10. This of course makes them dominant. The distance $d_{ij}$ between two candidates was calculated as the weighted Euclidean distance between the feature vectors, normalized so that the mean distance between pairs of feature vectors (coming from the candidates of the processed image) is 0.5, and clipped to the [0, 1] range. The correlations were calculated using these distances and the $\gamma$ function described in Figure 5.1b (with $D = 0.3$). The expected values $\mu_i$ were uniformly set to $\mu_i = 0.05$ in all experiments excluding those that specifically test their influence (section 5.4.4). $N$ was set to 100 in all experiments.

With the current implementation, calculating Esaliency for a $512 \times 384$ image takes on average about 250ms (on a Pentium 4, with a 3 GHz processor and 1GB memory). This is fast enough for most complex applications and is now being further improved.
5.4.2 A Synthetic Illustration

The first example is nonrealistic and is brought here as an illustration of the proposed attention algorithm. In this example, we consider the Esaliency assigned to the objects in an image containing 21 painted disks: 10 of them red, 5 blue, 3 green, 2 yellow and 1 pink (Figure 5.4a). The dependencies’ maximum spanning tree, which is built out of the correlations, is shown in Figure 5.4b. The saliency map and the attention fixation path are shown in Figures 5.4c and 5.4d. The results are as expected – the item that appears once gets the maximal saliency and is attended to first. The saliency is a bit lower for the items that appear twice, followed by the saliency of those repeated three times and so on.

The red and the blue candidates appear several times, and therefore, are definitely non-exceptions. Yet, their higher contrast with the white background makes them locally more salient. Indeed, running iLab’s toolkit (with default parameters), we see that the yellow and pink candidates are not attended to early in the search; see Figure 5.4e. In fact, due to the local approach and to the inhibition of return time constants, the pink target, which is the only global exception in the image, is never attended.

5.4.3 Testing Esaliency on Natural Scenes

The first set of natural scenes was selected from the University of Washington Ground Truth Database, available on the Web [2]. This database includes many outdoor scene images of size 512×768, and an annotation file describing the content of each image. 206 images that contain objects such as people, cars, houses, animals, bags, signs, and boats were selected. Images containing only “background items” such as sky, grass, trees, clouds, street and rocks,
were not used.

The attention experiments were carried using 50 of these images. (The other 156 images were used to validate our basic observations (section 5.2), and as a training set for setting nonuniform expected values (section 5.4.4)).

Two subjects, unaware of the research goal, were asked “to mark the interesting objects in each scene”. These markings served as ground truth for the targets in this experiment. The average number of marked objects (targets) in an image in the test set was about 3.5. Some of the images (13) contained only one target while the rest of them contained between 2 and 12 targets. Some of the multiple target images included targets with the same identity and some with different identities; see the two leftmost columns in Figure 5.5 for a sample of the images and the corresponding marked targets.

We ran Esaliency with the default parameters. The candidates (segments) were scanned in descending order of saliency. Let \( m_i \) be the number of candidates scanned until all targets are detected. A target was considered as detected when an attended candidate segment intersects with the corresponding marked region. Figure 5.5 (third column) shows the scan path associated with the first \( \min(m_i, 20) \) scanned candidates.

Some statistics of the search task results are summarized in the left column of Table 5.1. Note that the search mechanism is very efficient: only a few candidates were falsely visited on the way to detecting the true targets. With our implementation, every image contained an average of 330 segments. This means that only a small fraction of the image was scanned.

We then applied the local saliency model to the same 50 images using iLab’s toolkit (with its default parameters). Some of the resulting scan paths are demonstrated in Figure 5.5 (rightmost column). Many targets are efficiently detected but some problems are apparent. In the third image, for
example, the sky patches between the trees are indeed locally salient and are selected, by the local saliency process, long before the pedestrians. The proposed algorithm, on the other hand, is able to take advantage of the larger number of sky patches, reducing their Esaliency and focusing attention on the pedestrians earlier.

We observed that about 30% of the targets were not detected by the local saliency algorithm even after a very large number of fixations. Therefore, comparing the results by average detection time over all targets (table 5.1) was meaningless. We compared the algorithms by plotting the number of fixations needed for each target to be found (Figure 5.7)

Note the advantage of the proposed algorithm both in finding all the targets and in finding more targets for the same number of fixations.

We also experimented with the global, non-extended, saliency algorithm (section 5.3.3). The results are summarized in (the right column of) Table 5.1. Clearly, it is not as good as the Esaliency algorithm, probably due to the sharp demand for uniqueness which is not consistent with many images. Although the results of global saliency is also impressive for many images, when there are a few salient regions with similar appearance (associated with different targets or with the same target), it fails; See figure 5.6 for some examples.

The proposed Esaliency algorithm is not necessarily better than local saliency for all tasks. We considered the tasks reported in [44], where (variations of the) iLab algorithm were tested in detecting red cans, traffic signs and emergency triangles. Table 5.2 compares the results obtained for the three tasks using the default normalized iLab algorithm and the proposed Esaliency algorithm. The images in the red cans and triangles datasets are of size 640x480, while the traffic signs images are of size 512X384. In [44]
a fixation is considered successful if some part of the target object is inside the circle centered at the chosen fixation point with radios 80 for the two first datasets and radios 64 for the third dataset. To enable comparison, we considered (in these experiments) a fixation of Esaliency to be successful if the circle of radios 80/64 centered at the center of the selected segment intersects with a target. (Note that in other experiments a fixation was considered successful only if the selected segment intersect with a marked target.)

The Esaliency algorithm performed somewhat better for the emergency triangle task, and somewhat worse for the red cans and traffic signs. Note that these objects are designed to be locally salient on common background, either for safety or for commercial reasons. In the latter data set Esaliency fails mostly on the roadside light-reflectors which are discriminative by their dominant oblique orientation. We found that the fixations of Esaliency that come before the targets in these cases are always on other interesting objects in the scene.

5.4.4 Esaliency on Natural Scenes with Non-fixed Expected Value

We further tested how other preferences coming from human behavior or soft learning effects Esaliency. We try three extensions: 1) getting the portion-of-target preference from relevant training sets; 2) setting simple location preference, preferring the center over the periphery; and 3) setting target portion and location preference from relevant training sets.

We tested whether setting the expected value parameter $\mu$ according to the context makes a difference. To specify $\mu$ we used training sets of natural images and corresponding (manually obtained) binary maps of target locations. $\mu$ was uniformly set as the average fraction of target pixels. The $\mu$
values were 0.024, 0.015, 0.013 and 0.037 for the Washington database, the red cans, the triangles and the traffic signs, respectively. See table 5.3 (first two rows) for the results of this extended versions vs. the default performance. Clearly, the trained $\mu$ has almost no effect for all 4 data sets, suggesting that the Esaliency algorithm is not sensitive to the value of $\mu$ as long as it is small.

We then tested the effect of nonuniform expected value. Based on known eccentricity effect in human vision [20, 97] we specified an exponentially decreasing function which is maximal in the image center and is lower by a factor of $e^{-1}$ in all corners. This function was normalized so that its average value is the default value $\mu = 0.05$; see figure 5.8a. The prior $\mu_i$ for the $i-$th candidate is the maximum value of the map in it. The performance was improved for all datasets; see table 5.3. We verified that the improvement cannot be explained simply by ordering the candidates according to the variable expected values; see bottom row of table 5.3.

We also experimented with a space varying, context dependent, expected values. A priority map was created by averaging the binary maps of target locations, separately for each training set, and then smoothing them. See figure 5.8b-e. All the priority maps show a preference to the center reflecting the tendency of people to center their photography objects. Here, as well, $\mu_i$ are set to the maximum value of the map inside the corresponding candidate region. Note that while the median number of fixations never increases, their average number may be higher in some cases; see table 5.3. This is probably due to candidates being in untypical locations, not represented in the training set.
5.4.5 Using Esaliency to Accelerate Pedestrian Detection

We now turned to test the Esaliency algorithm in the context of pedestrian detection. We used the MIT Street Scene database [16] containing 3,547 images of urban scenes. The locations of objects from nine categories, cars, pedestrians, bicycles, buildings, trees, roads, sky, sidewalks and stores, are annotated in the images. We focus on the task of detecting pedestrians. Esaliency (using default parameters) was applied to all (852) images that contained marked pedestrians. (The images were down-sampled from 1280 × 960 to 640 × 480). The mean number of false-fixations before locating first pedestrian was 27. The median was 12. The mean and the median of the number of false fixations before all pedestrian were located were 41.55 and 21, respectively; see figures 5.9 to 5.14.

To estimate the computational savings, we consider a simple model of the search process. Without attention, a common detection mechanism (e.g, [90]) evaluate sub-images. Suppose that the detection process starts from the upper left corner of an \( h \times w \) image, and a window of size 20X30 scans the image in raster scan, jumping in steps of two pixels. This is done for, say, 6 scales (to detect pedestrians in different size), each time for an image 1.5 times smaller in both dimensions. The total number of windows checked is \( T = \sum_{i=0}^{5} \frac{(h/1.5^i-30)(w/1.5^i-20)}{4} \). For \( w \times h = 1280 \times 960 \), \( T = 5.1019 \times 10^5 \).

Consider an alternative scenario, where Esaliency set the order of fixations. For each fixation we let the detection algorithm check all the windows that include the center fixation point in all 6 scales. For \( k \) fixations, the number of checked windows are \( T' = k \frac{20 \times 30}{4} \times 6 = 900k \). Hence, \( \frac{T'}{T} = 0.0018K \).

That is, with Esaliency, the median number of windows tested before all pedestrian are detected is just \( \frac{T'}{T} = 0.0378 \), or less than 4% of the total.
number of windows used in the sequential scan. (Note that the estimate is conservative: we could reduce the number of tested windows, for example, by taking only those that contain all or most of the segment.)

Note that the results for StreetScenes are not as good as those obtained for the Washington images. One obvious reason is that in the Washington images, all ‘interesting objects’ were marked and considered to be targets. Here, only pedestrians are considered targets, and other salient objects, such as cars, signs, etc., which attract attention, are considered as nontargets. Besides, most images in the Washington data set are of natural scenes, and are less crowded with nontarget salient objects. Actually, some of the pedestrians in the StreetScenes images were not marked, and therefore sometimes pedestrians are found earlier, without counting it as a success (see, e.g., image 2786).

Nevertheless, as shown above, the usage of Esaliency definitely increase the detection efficiency. We emphasize that the proposed bottom up Esaliency algorithm does not use any knowledge about the pedestrians. It would have been possible to improve by adding such knowledge, by say locating sidewalks or zebra-crossing first, or by optimizing the segmentation or the feature selection for the pedestrians context.

### 5.4.6 Evaluation of Esaliency vs. Human Eye Fixations

The main goal of this work was to develop an efficient and quantitatively meaningful attention process for computer vision. Yet, it is tempting to relate the proposed approach to the human vision attention mechanism. We describe here a preliminary study of this relation.

We follow the approach proposed in [59], which evaluates an attention
model by comparing the saliency map proposed by the model to an empirical human saliency map. The latter is constructed by recording human eye fixations over a (limited time displayed) image, and convolving each fixation point in the image with a Gaussian. Averaging the human saliency maps obtained from several (7 or more) subjects yields a mean human saliency map for each image. The map are intensity normalized and down sampled (from $512 \times 384$ to $32 \times 24$) so that they can be easily compared to the saliency maps of the iLab algorithm [45], available in [1]. See Figure 5.15. The correlation coefficient between two saliency maps serves as a quantitative scalar measure of similarity. See the upper rows of table 5.4 for the correlation between Itti’s model to the empirical mean human saliency map, as reported in [59].

To evaluate Esaliency, we ran it on the same images and created saliency-maps in a similar way. The sequence of fixations were specified as the centers of the attend candidate regions. See Figure 5.15. See also the correlation coefficients between the resulting Esaliency-maps and the human saliency maps in table 5.4.

All correlations between the mean human saliency maps and those of the computational models are significant ($p$-value $\ll 0.05$). The correlations between the mean human maps and the Esaliency maps is higher than the correlations with iLab saliency in 5 out of 6 images, implying a somewhat better agreement.

5.4.7 Extending the Regions for Recognition

One difficulty of using common saliency methods for object recognition or detection is that they provide only a rough indication where the candidate object is, and not, say, the segment or even a bounding box containing it.
One way to deal with this difficulty, suggested in the context of the iLab model [45], is described in [93]. Their approach is to identify the feature map and the scale that are contributing to the maximal saliency, and to use them for locally segmenting a region around the fixation point.

Esaliency does a step forward by directing attention to segments. However, these segments are often just parts of the objects of interest. Some postprocessing stage is needed to specify the region of the objects. We experimented with a simple aggregation process which recursively adds neighboring segments associated with high saliency, to the segment suggested by Esaliency; see figure 5.16.

Unlike [93], the aggregated object may contain dissimilar parts. In the swan example, for instance, the beak is the most salient segment, and the dissimilar swan’s body that is moderately salient is added to the selected region. We now consider several methods for improving this aggregation process.

5.4.8 Esaliency using Movement as a Distinguishing Feature

As described above, the Esaliency algorithm was suggested and tested for detecting saliency in still images. When dealing with a video stream, saliency of a candidate may be defined not only from unique color and shape, but also from unique movement. The Esaliency (and VSLE) application described in appendix A support also video input. Then, the optical flow between two successive video frames is computed (the current implementation uses a block matching based method), providing the movement vector at each location. The average movement vector of the pixels of a segment-candidate is added to the candidate’s feature-vector, and participate in the mutual-similarities
measure. Then, a unique movement direction or speed (of one or a small
group of candidates) contributes to the probability of the candidates to be
considered as salient. Note that since what counts is the saliency in the
movement, the algorithm should not be bothered by camera movement that
approximately moves all the flow to the same direction.

See some examples in figure 5.17.

5.5 Discussion

In this chapter we proposed a new, region-based bottom-up saliency mea-
sure. This measure is the (approximate) probability of an image region to
be salient, estimated from preferences on the number of objects of interest in
the scene and from validated stochastic modeling of the likely target assign-
ments. This quantitative approach differs from the traditional feature-based
(or space-based) methods. Moreover, the resulting Esaliency estimates are
based on global considerations, which are more justified. We found that the
proposed method is fast, reliable and performs better in complex, cluttered
scenes.

We believe that the approach proposed here may serve as solid foundation
for other search and detection tasks. Here we focused on a pure bottom-
up approach. For specific applications, top down information is available
and may be integrated naturally in the framework described here, either
by adapting the segmentation, and the similarity measures to the specific
context, or even by setting the expected values $\mu_i$ depending on the candidate
properties and on the context (see, e.g., [82]).

Sometimes, local contrast is a valuable indicator of importance. Then, we
may want to add it as an additional criterion for saliency. One way would be
to make the expected values $\mu_i$ dependent on the local contrast. An easier alternative would be to add spatial coordinates as a feature. This will lead to independent labeling of distant candidates even if they are visually similar and will degrade the priority of closely located similar candidates.

We used $N = 100$ best assignments in all the experiments as it gave satisfactory results. However, there is place to further understand the influence of this parameter and to suggest a method to optimally set its value.

Improving regions for recognition is very important as it can reduce the amount of computations of the recognizer per attended candidate. This can be achieved by using better segmentation processes, by combining top-down information, or by further developing the direction demonstrated in section 5.4.7 for unifying nearby regions that are both salient.

Finally, to our best knowledge the mechanism described here is the first quantitative and practical model for object-based bottom-up attention. It should be interesting to check whether it may indeed explain some aspects of human perception. The preliminary saliency comparisons, described in Section 5.4.6, are encouraging.
Figure 5.4: Demonstrating the Esaliency algorithm on a synthetic image: (a) The input image; (b) the dependencies’ spanning tree calculated by the algorithm (each node is colored according to the corresponding candidates); (c) The saliency map suggested by Esaliency; (d) The 20 first fixations suggested by Esaliency; (e) The 20 first fixations suggested by local saliency (iLab’s toolkit). (It is important that this figure be viewed in color.)
Figure 5.5: Example of results of Esaliency vs. iLab’s toolkit for images from the Washington Ground-Truth Database. Left to right: The input images, the objects marked as interesting by human subjects, and the resulting fixation order of Esaliency marked on the segmented image. Candidates intersecting with marked targets are marked in yellow, others in red. The rightmost column show iLab’s toolkit results. For both algorithms either the first 20 fixations or a smaller number of all targets are detected, are plotted. Best viewed on a colored computer screen.
<table>
<thead>
<tr>
<th>False detections before</th>
<th>Esaliency</th>
<th>Non extended global saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st target detected</td>
<td>0.70 ± 1.54; 0</td>
<td>1.84 ± 3.68; 0</td>
</tr>
<tr>
<td>before 50% of targets detected</td>
<td>0.88 ± 1.53; 0</td>
<td>2.29 ± 3.88; 0.55</td>
</tr>
<tr>
<td>before 75% of targets detected</td>
<td>1.73 ± 2.27; 0.67</td>
<td>3.09 ± 3.63; 2</td>
</tr>
<tr>
<td>before all targets detected</td>
<td>2.50 ± 3.02; 1</td>
<td>3.66 ± 3.67; 3</td>
</tr>
<tr>
<td>50% of targets detected</td>
<td>1.60 ± 2.75; 0</td>
<td>3.44 ± 5.62; 1.5</td>
</tr>
<tr>
<td>before 75% of targets detected</td>
<td>5.48 ± 8.89; 2</td>
<td>8.36 ± 10.12; 4.50</td>
</tr>
<tr>
<td>before all targets detected</td>
<td>12.20 ± 24.43; 3</td>
<td>14.06 ± 18.84; 7.5</td>
</tr>
</tbody>
</table>

Table 5.1: Results for Esaliency non-extended global saliency (section 5.3.3) on 50 images from the Washington Ground Truth database[2]. Mean, standard-deviations and median are reported.
Figure 5.6: Some additional examples for Esaliency, compared with non-extended global saliency. Left to right: input images, the objects marked as interesting by human subjects, the fixation set by non-extended global saliency, and the fixation order specified by Esaliency. Candidates intersecting with marked targets are marked in yellow, others in red. Best viewed on colored computer screen.
Figure 5.7: Comparing Esaliency and iLab’s results on the Washington Database. The average fraction of targets found is plotted as a function of the number of fixations per target.

<table>
<thead>
<tr>
<th></th>
<th>False detection before targets found by Itti et al.</th>
<th>False detection before targets found by Esaliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red can</td>
<td>1.67 ± 2.01</td>
<td>2.95 ± 3.36; 2</td>
</tr>
<tr>
<td>Emergency triangle</td>
<td>1.69 ± 2.28</td>
<td>1.25 ± 1.60; 1</td>
</tr>
<tr>
<td>Traffic signs first</td>
<td>0.49 ± 1.06</td>
<td>0.69 ± 1.28; 0</td>
</tr>
<tr>
<td>Traffic signs all</td>
<td>1.27 ± 2.12</td>
<td>2.09 ± 2.12; 1</td>
</tr>
</tbody>
</table>

Table 5.2: Searching for locally-salient objects (red can, emergency triangle and traffic signs). Comparing results of Esaliency and results reported in Itti et al. [44]. The mean, standard deviations and median of the number of false detections before the targets are detected are reported. Some images from the traffic sign database include more than one sign (while the other databases always include one target per image).
Figure 5.8: Target locations priority maps. The map (a) specifies a preference to objects that are closer to the center. The other 4 maps are learned from training sets associated with the 4 data sets described in section 5.4.4.
Table 5.3: The performance associated with several versions of Esaliency, described in section 5.4.4, as well as some reference figures (two bottom lines). The mean, standard deviations and median of the number of false detections before the targets are detected are reported.
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency’s fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0072</td>
<td><img src="image1" alt="Marked pedestrians" /></td>
<td><img src="image2" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0143</td>
<td><img src="image3" alt="Marked pedestrians" /></td>
<td><img src="image4" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0286</td>
<td><img src="image5" alt="Marked pedestrians" /></td>
<td><img src="image6" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0290</td>
<td><img src="image7" alt="Marked pedestrians" /></td>
<td><img src="image8" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0425</td>
<td><img src="image9" alt="Marked pedestrians" /></td>
<td><img src="image10" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0447</td>
<td><img src="image11" alt="Marked pedestrians" /></td>
<td><img src="image12" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0453</td>
<td><img src="image13" alt="Marked pedestrians" /></td>
<td><img src="image14" alt="Esaliency’s fixation path" /></td>
</tr>
<tr>
<td>0575</td>
<td><img src="image15" alt="Marked pedestrians" /></td>
<td><img src="image16" alt="Esaliency’s fixation path" /></td>
</tr>
</tbody>
</table>

Figure 5.9: Examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency's fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0683</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>0687</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>0858</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>1179</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>1301</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>1305</td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>1319</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
</tr>
<tr>
<td>1321</td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 5.10: More examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency’s fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1361</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>1387</td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td>1640</td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>1641</td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td>1653</td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>1656</td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
<tr>
<td>1687</td>
<td><img src="image13.png" alt="Image" /></td>
<td><img src="image14.png" alt="Image" /></td>
</tr>
<tr>
<td>1764</td>
<td><img src="image15.png" alt="Image" /></td>
<td><img src="image16.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 5.11: More examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency's fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1779</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1794</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1795</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1797</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1805</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1817</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>1881</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
<tr>
<td>2042</td>
<td><img src="Image" alt="Marked Pedestrians" /></td>
<td><img src="Image" alt="Fixation Path" /></td>
</tr>
</tbody>
</table>

Figure 5.12: More examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency’s fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>2076</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
</tr>
<tr>
<td>2113</td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
</tr>
<tr>
<td>2217</td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
</tr>
<tr>
<td>2288</td>
<td><img src="image7" alt="Image" /></td>
<td><img src="image8" alt="Image" /></td>
</tr>
<tr>
<td>2412</td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
</tr>
<tr>
<td>2603</td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
</tr>
<tr>
<td>2652</td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
</tr>
<tr>
<td>2755</td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 5.13: More examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
<table>
<thead>
<tr>
<th>Image</th>
<th>Marked pedestrians</th>
<th>Esaliency’s fixation path</th>
</tr>
</thead>
<tbody>
<tr>
<td>2786</td>
<td><img src="image1" alt="Marked pedestrians" /></td>
<td><img src="image2" alt="Fixation path" /></td>
</tr>
<tr>
<td>2794</td>
<td><img src="image3" alt="Marked pedestrians" /></td>
<td><img src="image4" alt="Fixation path" /></td>
</tr>
<tr>
<td>2861</td>
<td><img src="image5" alt="Marked pedestrians" /></td>
<td><img src="image6" alt="Fixation path" /></td>
</tr>
<tr>
<td>2985</td>
<td><img src="image7" alt="Marked pedestrians" /></td>
<td><img src="image8" alt="Fixation path" /></td>
</tr>
<tr>
<td>3382</td>
<td><img src="image9" alt="Marked pedestrians" /></td>
<td><img src="image10" alt="Fixation path" /></td>
</tr>
<tr>
<td>3435</td>
<td><img src="image11" alt="Marked pedestrians" /></td>
<td><img src="image12" alt="Fixation path" /></td>
</tr>
</tbody>
</table>

Figure 5.14: More examples for Esaliency’s performance for locating pedestrians in the StreetScenes images
Figure 5.15: Comparing computer saliency models and human saliency maps. From left to right: The input images; The mean human saliency maps created from recording human eye fixations [59]; iLab’s model [45] resulting saliency maps. Esaliency’s resulting saliency maps.
<table>
<thead>
<tr>
<th></th>
<th>road1</th>
<th>road 2</th>
<th>road 3</th>
<th>coke</th>
<th>swissalps</th>
<th>forest</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>iLab vs. human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All subjects</td>
<td>0.232</td>
<td>0.3624</td>
<td>0.482*</td>
<td>0.4</td>
<td>0.523</td>
<td>0.436</td>
</tr>
<tr>
<td>Best subject</td>
<td>0.321</td>
<td>0.348</td>
<td>0.462</td>
<td>0.45</td>
<td>0.608</td>
<td>0.477</td>
</tr>
<tr>
<td>Worst subject</td>
<td>0.079</td>
<td>0.194</td>
<td>0.082</td>
<td>0.154</td>
<td>0.134</td>
<td>-0.078</td>
</tr>
<tr>
<td><strong>Esaliency vs. human</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All subjects</td>
<td>0.5795</td>
<td>0.7093</td>
<td>0.6469</td>
<td>0.6709</td>
<td>0.3712</td>
<td>0.6114</td>
</tr>
<tr>
<td>Best subjects</td>
<td>0.6155</td>
<td>0.7411</td>
<td>0.7150</td>
<td>0.6805</td>
<td>0.5142</td>
<td>0.4759</td>
</tr>
<tr>
<td>Worst subjects</td>
<td>0.3051</td>
<td>0.3813</td>
<td>0.2171</td>
<td>0.3222</td>
<td>0.0590</td>
<td>-0.0294</td>
</tr>
<tr>
<td>subject 1</td>
<td>0.5018</td>
<td>0.7411</td>
<td>0.5730</td>
<td>0.5749</td>
<td>0.5142</td>
<td>0.2963</td>
</tr>
<tr>
<td>subject 2</td>
<td>0.3228</td>
<td>0.5486</td>
<td>0.6483</td>
<td>0.6805</td>
<td>0.3005</td>
<td>0.1952</td>
</tr>
<tr>
<td>subject 3</td>
<td>0.5884</td>
<td>0.5079</td>
<td>0.7087</td>
<td>0.3222</td>
<td>0.3435</td>
<td>0.4759</td>
</tr>
<tr>
<td>subject 4</td>
<td>0.6155</td>
<td>0.6360</td>
<td>0.3670</td>
<td>0.5272</td>
<td>0.2143</td>
<td>0.2506</td>
</tr>
<tr>
<td>subject 5</td>
<td>0.3051</td>
<td>0.3813</td>
<td>0.7150</td>
<td>0.6258</td>
<td>0.1846</td>
<td>0.2126</td>
</tr>
<tr>
<td>subject 6</td>
<td>0.3961</td>
<td>0.6713</td>
<td>0.6198</td>
<td>0.3243</td>
<td>0.0590</td>
<td>-0.0294</td>
</tr>
<tr>
<td>subject 7</td>
<td>0.4395</td>
<td>0.4543</td>
<td>0.2171</td>
<td>0.4589</td>
<td>0.3064</td>
<td>0.2655</td>
</tr>
<tr>
<td>subject 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.3246</td>
</tr>
<tr>
<td>subject 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.0144</td>
</tr>
<tr>
<td>subject 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.2219</td>
</tr>
<tr>
<td>subject 11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.4321</td>
</tr>
<tr>
<td>subject 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.4296</td>
</tr>
<tr>
<td>subject 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.4141</td>
</tr>
<tr>
<td>subject 14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1935</td>
</tr>
<tr>
<td><strong>Esaliency vs. Itti</strong></td>
<td>0.5972</td>
<td>0.4628</td>
<td>0.4881</td>
<td>0.6050</td>
<td>0.5137</td>
<td>0.6770</td>
</tr>
</tbody>
</table>

Table 5.4: Correlation coefficients between human saliency maps [59] and computational models saliency maps. Upper rows: correlation coefficients between human saliency maps and iLab’s model saliency maps; middle rows: correlation coefficients between human saliency maps and Esaliency’s saliency maps; bottom: correlation coefficients between the saliency maps of the two computational models. * I measured 0.357
Figure 5.16: Extending the candidate for recognition. Left: input images; middle: results of applying Esaliency; right: candidates extended by joining neighboring candidates that are also salient (see section 5.4.7)
Figure 5.17: Demonstrating use of movement as a distinguishing feature. In the first example we used the synthetic circles image and created another frame in which two circles were moved to the right, and two were moved down. In the second (real movie) example all flowers and leaves move due to wind and camera movement, however, the butterfly’s wings movement is salient. The combination of color and movement features gives the best results for detecting the butterfly. In the third example the car proceeds towards the viewer. Using only color information or only the (not so good) optic flow results in this case, Esaliency does not detect the car in its 5 first fixations. However, when color and movement features are combined, the non-relevant salient optic flow in background is reduced, and the first fixation is on the car.
Chapter 6

A Psychophysical Study for Predicting Human Visual-Search Performance

1

6.1 Introduction

Although our main goal is attention mechanisms for the use of computer vision, we have also tested the relevance of the COVER measure and the FLNN model in for human performance prediction.

In four experiments, observers searched for a horizontal target presented among distractors of different orientations (orientation-search; Experiments 1 and 2) or a gray target appearing among distractors of different colors (color-search; Experiments 3 and 4). Distractors’ homogeneity and target-

1Joint work with Yaffa Yeshurun from the Psychology Department at the University of Haifa
distractors similarity were systematically manipulated. We extended the Cover and FLNN models to account for internal-noise, and compared the prediction abilities of our models with that of other prominent models of visual search.

All the experiments were performed in a two-interval forced-choice (2IFC) manner. At each trial the observer views two successive displays, each one for a limited time. One is a target-absent display and the other is a target present display. A success is encountered if the observer recognized correctly (by keyboard response) the target-present display. Usually, the measure for performance, the accuracy, is the percent of trials in which the observer succeeded.

### 6.1.1 Related work

As described in chapter 1 many computational models were suggested to account for biology visual search performance. We mention here the models which we found most relevant for comparing their prediction abilities with these of our models.

The Temporal-Serial model suggests that the observer can process only $k$ out of the $n$ stimuli of each display at the limited exposure time. If the target was one of the $k$ selected stimuli in the target-present display, a correct decision is taken. Otherwise a guess would yield 50% success [15].

The Signal Detection Theory (SDT) based models (e.g., [29, 30, 42, 61, 74]) assume that the stimuli is observed with addition stochastic noise. Then, a false-detection may occur when one of the distractors, given its noisy-observation, is mistakenly perceived as coming from the target distribution, and vice versa for a miss.

Rosenholtz, inspired by Duncan and Humphreys [28], similarity theory, has
recently suggested the target saliency measure [71] and the Best-Normal and RCref (Relative Coding with reference) models ([72]). Given the feature-space relevant to the search task (e.g.-velocity space in motion search, color-space in color-search), and the points in that space describing the stimuli, Rosenholtz’s target’s saliency measure is the number of standard deviations between the target point and the mean of the distractors points (Mahalanobis distance). It suggests that a search-task is more difficult as the distance between the mean-distractor-value and target-value decreases, and the distractors variance increases. We denote the Mahalanobis distance, measured for non-noisy input, the SALIENCY model, and the Mahalanobis distance measured for input with additive, normally-distributed, random internal noise the NOISY-SALIENCY model. For a single dimension feature-space, \[ \text{SALIENCY} = \frac{(x - \mu)^2}{\sigma_D^2} \] where \( x \) is the target value, \( \mu \) is the distractors’ mean-value and \( \sigma_D^2 \) is the variance of the distractors values, and \[ \text{NOISY-SALIENCY} = \frac{(x - \mu)^2}{\left(\sigma_D^2 + \sigma_N^2\right)} \] where \( \sigma_N^2 \) is the variance of the internal-noise. While the target saliency measure is a qualitative abstractive mathematical phrase for search-tasks difficulty, the Best-Normal and RCref models are more quantitative and suggest methods for predicting performance in 2IFC experiments. The Best-Normal and RCref models are variations on a SDT model. While SDT models assume that the observer has a record of the exact distracters distribution, the Best-Normal model suggests that during visual search the observer uses a more simple approximated representation of the distracters’ distribution. The true distribution is represented only by its mean and variance, that is, by the normal distribution that best fits the true distracters distribution. Note that whereas both the Best-Normal and the classical SDT models predict that search performance should get harder as target-distracters similarity increases, only the
Best-Normal model can account for the increase in search difficulty with an increase in distracters heterogeneity. The RCref model is a modification of the relative coding model [62]. Like the Best-Normal model, the RCref model suggests that the observer does not use the exact distribution of the search items. The recorded distribution does not correspond to the feature-values themselves, but to relative values. Specifically, the recorded distribution corresponds to the combination of the differences between the various items that are present in the display and the differences between display items and a reference target. Thus, part of the times an observed display item is compared to another display item and on other times it is compared to a reference-target.

### 6.1.2 Extensions of the COVER and FLNN Models

To adapt our models for human vision we extend them to account for internal-noise and suggest the NOISY-COVER and NOISY-FLNN models. Both the original and the extended versions abilities to predict human performance are evaluated in this study.

**The NOISY-COVER Model**

The COVER measure gives some indication of the difficulty of search tasks, however, a) it is deterministic and discrete, while human responses are not, b) it does not quantify the growing difficulty of homogeneous tasks as the similarity between the target and the distractors increases, and c) it does not quantify the growing difficulty as the number of distractors increase. Fortunately, these issues are simply addressed by assuming, like in the signal-detection-theory (SDT), that a noisy representation of each stimulus is observed rather than the feature-value itself. This noise is assumed to be nor-
 normally distributed with mean 0 and observer-dependent standard-deviation $\sigma$. Considering this internal noise, the resulting cover associated with the observation is random. We propose to use the expected cover as the search difficulty indicator and denote it NOISY-COVER. Unlike the COVER model, NOISY-COVER provides a non-integer continuous value that allows more distinctive predictions (and provide answers to the three items above). Given the feature-values associated with the stimuli of a specific experiment’s condition, and the internal noise variance $\sigma^2$, the NOISY-COVER measure is estimated as follows:

- An observed target-present display is randomly generated by picking the feature-value of each element from the normal distribution it belongs to (with mean being its true displayed value and $\sigma$ being a parameter describing the amount of noise).

- The COVER value is calculated for the specific generated case.

- This is repeated $N$ (1000) times, resulting in $N$ COVER values.

- The NOISY-COVER is the average of those calculated values.

The COVER model explicitly suggests that the difficulty of a search-task is connected to some grouping of the stimuli. It suggests that the variability inside one group depends on the difference between the target and the distractor most similar to it. If the target and this distractor are very similar, more groups will be generated. If the target and closest distractor are less similar, there will be less groups, and even non-similar distractors could be members of the same group given they are more similar to each other than any distractor is to the target. The difficulty measure suggested is proportional to the number of those groups. Considering the observer’s internal
noise as suggested here actually suggests that more groups are generated as the noise level grows. Stimuli that were originally similar (and therefore in one group) can now be in separate groups as they are observed differently, making the task more difficult.

The COVER and NOISY-COVER models suggest measures for comparing the difficulty of different conditions, which should influence the reaction time and accuracy of human observers performance. The relation between COVER and the accuracy is not explicit, though. A link between the COVER and human performance may be achieved by modeling the search using the FLNN mechanism.

The NOISY-FLNN Model

Like the COVER model, FLNN originally considers non-noisy observation. We suggest a model that assumes noisy observations. Under this model the observed items are random and so is the chance that the observer succeeds in some test. We propose to take the probability of success as the accuracy prediction. The accuracy prediction model is denoted NOISY-FLNN. For a 2IFC experiment the prediction is done in the following way:

- An observed target-present display is randomly generated by picking the feature-value of each element from the normal distribution it belongs to (with mean being its true displayed value and $\sigma$ being a parameter describing the amount of noise).

- In a similar way, an observed target-absence display is generated.

- It is assumed that in the limited presentation time of the display only $k$ elements can be processed. Therefore, the FLNN algorithm is simulated
for $k$ steps on each of the two generated ‘displays’, and we get $k$ selected elements for each display.

- The algorithm points to the display that contains the stimulus that is most similar (feature-value-wise) to the true target-value (without noise), out of the $2k$ selected stimuli, as the target-present display.

- If the algorithm pointed to the target-present generated display this is counted as a success, otherwise this is a failure.

- All the above is repeated many times (10,000 in our case) and the fraction of successes is the accuracy prediction suggested.

Note that his model requires two parameters, $\sigma$ and $k$, for characterizing each individual observer. Similarly to the Temporal-Serial model mentioned above, when the display time is limited FLNN considers only $k$ out of the $n$ stimuli. However, while the $k$ items are selected randomly for the Temporal-Serial model, in FLNN the $k$ chosen items depend on the feature-values of the stimuli.

### 6.1.3 Outline

In the following section four experiments are reported. In experiments 1 and 2 the search is for a horizontal line segment among tilted distractors (orientation-search). In experiments 3 and 4 the search is for a gray-colored disk among distractors of different colors (color-search). The prediction abilities of the COVER and NOISY-COVER are compared to that of the Saliency model [71]. The NOISY-FLNN prediction abilities are compared to those of a SDT-based model [61], the Best-Normal and RCref models [72], and the Temporal-Serial model [15, 29]. The prediction attempts were performed for
the four experiments mentioned above, and for two additional experiments reported in [72]. The overall results show that the models suggested here predict the participants performance better than the compared to models. Some possible further improvements of the models are suggested and discussed (Section 6.3).

6.2 Experiments

Our experiments include orientation-search tasks and color-search tasks. In the orientation experiments (experiments 1 and 2) the target is always a horizontal line segment and the distractors are oblique line segments. In the color-search experiments (experiments 3 and 4) the target is always a gray disk, while the distractors are disks in different reds and greens. See figure 6.1 for a schematic description of the stimuli in each condition of each experiment, and the amount of elements in each condition, both in target-absent and target-present displays. Our four experiments are denoted: 1. Orientation Unidirectional, 2. Orientation Bidirectional, 3. Color Unidirectional, and 4. Color Bidirectional. In the ‘Unidirectional’ experiments all the distractors’ feature-values lie in one side compared to the target’s feature-value, while in the ‘Bidirectional’ experiments the distractors’ feature-values are in both sides of the target’s feature value (always in a symmetric way). Both the unidirectional and the bidirectional experiments were designed to test whether the COVER, NOISY-COVER, FLNN and NOISY-FLNN models have advantages over previous models. The goals of the unidirectional experiments were: a) To get more evidence for the fact that tasks with heterogeneous distractors may be harder than homogeneous ones, even when the distractors in the heterogeneous case are less similar to the target than
in the homogeneous case. This claim is modeled by the COVER and the SALIENCY models but is not predicted by SDT-based models. b) To show that the difficulty is effected by the target-distractors feature-space distances relative to the the distractors-distractors feature-space distances, and not just by the absolute values of either. c) Qualitatively, both ours and Rosenholtz’s models predict arguments a and b. Therefore, we wanted to compare quantitatively the abilities of the models to predict the participants’ performance.

As to the bidirectional experiments: a) Rosenholtz’s models cannot predict differences in difficulty for cases in which the distractors’ feature-values are symmetric around the target feature-value. In such cases Rosenholtz’s SALIENCY and NOISY-SALIENCY measure is 0 for all conditions. However, the models suggested here can predict differences between different symmetric conditions. The bidirectional experiments were designed to test whether human performance will demonstrate such difficulty differences and whether these differences will follow our models predictions. b) In each bidirectional experiment conditions 2 and 3 use the same distractors, but the number of each type is different. (In condition 3 the number of distractors that are more similar to the target is larger.) The same holds for conditions 4 and 5. (See figure 6.1.) The idea behind such pairs of experiments was to test the prediction ability of the FLNN relative to the COVER’s. While the COVER model does not change its prediction between conditions 2&3 and between conditions 4&5, FLNN suggests no difference between conditions 2&3, but suggests a difference in difficulty between conditions 4&5. Condition 5, in which there are more distractors that are more similar to the target, should be harder according to FLNN. See more details in section 6.2.2.

Below we present the results of our four experiments and test the ability of the suggested models to predict the results of these four experiments and
two experiments reported in [72]. The prediction abilities of the COVER and
NOISY-COVER are compared to that of the Saliency model ([71]). Specif-
ically we check whether each model can predict the relative difficulty of the
experiments’ conditions implied from the participants performance, and com-
pare the correlation-coefficients of the results and the models predictions. See
details in 6.2.1. To compare the prediction ability of the NOISY-FLNN model
to that of a SDT-based model, Rosenholtz’s Best-Normal and RCref models
and the Temporal-Serial model we use the reduced-chi-square ($\chi^2/df$) value
and the chi-square-test ($\chi^2$-tests). See details in 6.2.1. The implementation
of the SDT-based model, the Best-Normal model and the RCref model are
according to [72] and personal communication. The implementation of the
Temporal-Serial model is according to [29]. For predicting experiment 1 &
2 by the SDT, Best-Normal, and NOISY-FLNN models we considered the
distribution of orientation to be a wrapped distribution as suggested in [72]
– a line segment of $\alpha^\circ$ can be considered both as of $\alpha^\circ$ and of $(180 – \alpha)^\circ$.
In prediction of the color-search experiments (3 & 4) the distributions are
considered non-wrapped.

6.2.1 Experiment 1: Orientation Unidirectional

Method

Observers: Five students (A.P., Y.B., D.A., V.S., A.P.Z.) from the University Of Haifa with normal or corrected to normal vision, who were naive to
the purpose of the study.

Stimuli and Apparatus: The stimuli were present on a 21-in. monitor of
a PowerMac G4 computer (Resolution: 1280x1024 85Hz), using Vscope™
([33]). The search display was consisted of 36 black line segments, each
subtending 0.5° height x 0.1° width of visual angle, presented on a white background. The lines were randomly scattered within a non-visible circle with a radius of 4° (Figure 6.2a). The target was always a horizontal (0°) line segment, and it was present on the first or second interval equally often. The orientation of the distractors lines presented in each of the four conditions of this experiment was as follows (see also Figure 6.1a): In condition 1, the orientation of all the distractors was 15°. In condition 2, half of the distractors had an orientation of 15° and the other half of 40°. In condition 3, all the distractors had an orientation of 40°. Finally, in condition 4, half of the distractors had an orientation of 40° and the other half of 65°. The fixation mark was a plus sign (0.5° x 0.5°) presented in the center of the screen, and a plus (0.33° x 0.33°) or a minus (0.33° x 0.1° height) sign served as the feedback.

**Procedure:** An experimental trial included two temporal intervals. Each interval began with 750-ms of the fixation mark followed by 500-ms of the search display. The observers were required to indicate whether the target appeared on the first or second interval. Immediately after observers responded the appropriate feedback sign was presented for 1-s. Each observer participated in 3 experimental sessions. A single session was consisted of four blocks of 100 trials, each corresponding to one of the four experimental conditions. The order of the blocks within a session and the order of trials within a block were randomized. Overall, observers participated in 300 trials per experimental condition and a total of 1200 trials.

**Results**

As can be seen in Figure 6.3, all five participants performed significantly worse (z-test, p < 0.05) in the second condition (distractors at 15° and
40°). For participants A.P. and V.S. the second most difficult condition was condition 1 (all distractors at 15°), while condition 3 (all distractors at 40°) and condition 4 (distractors at 40° and 65°) were the easiest for both of them with non-significant performance difference between them (z-test, $p > 0.05$). For participants Y.B., D.A. and A.P.Z. there was no significant performance difference between conditions 1, 3, and 4.

**Predictions and Discussion**

One of the motivations of the this experiment was to get more evidence for the fact that tasks with heterogeneous distractors may be harder than homogeneous ones, even when the distractors are less similar to the target in the heterogeneous case. This claim is modeled by the COVER and the SALIENCY models but is not predicted by SDT-based models. This behavior is clearly demonstrated in the experiment’s results (Comparing for instance the performance in condition 1 and 2). Another goal of the experiment was to show that the difficulty is effected by the target-distractors feature-space distances relative to the distractors-distractors feature-space distances, and not just by the absolute values of either. If this assumption was correct, the difference in difficulty between condition 1 and 2 should be greater than the difficulty difference between conditions 3 and 4. The experiment verified this hypothesis for all five participants.

Qualitatively, both ours and Rosenholtz’s models predict the above arguments. Therefore, we wanted to compare quantitatively the abilities of the models to predict the participants’ performance. First we check the ability of the COVER, NOISY-COVER, SALIENCY and NOISY-SALIENCY models to predict the difficulty ordering of the conditions. See Table 6.1. The COVER for conditions 1 (all distractors at 15) and 3 (all distractors at 40) is
2 as all the distractors are homogeneous in both conditions. For condition 2 (distractors at 15 and 40) the COVER is 3, as $d_T = 15$ and the difference between distractors at $15^\circ$ and at $40^\circ$ exceeds $d_T$. For condition 4 (distractors at 40 and 65) the COVER is 2 as $d_T = 40$ and the difference between the distractors at $40^\circ$ and at $65^\circ$ is less than $d_T$. Therefore, the COVER model suggests that condition 2 is the most difficult, and that the other three conditions are equally difficult. This fits the results of three out of five participants. The NOISY-COVER model with the noise level being a parameter whose value is evaluated for each participant, can predict also the difficulty order of results for all five participants. Rosenholtz's SALIENCY model (without noise) suggests that condition 2 is the most difficult (SALIENCY = \( \frac{27.5^2}{12.5^2} = 4.84 \)), followed by condition 4 (SALIENCY = \( \frac{52.5^2}{12.5^2} = 17.64 \)). Conditions 2 ($15^\circ$, $40^\circ$) and 4 ($40^\circ$, $65^\circ$) have the same variance among distractors, however the target is more similar to the distractors in condition 2, which makes the target's saliency smaller. For conditions 1 and 3 the SALIENCY is infinite, as there is no variance between the distractors (SALIENCY = \( \frac{15^2}{0} = \infty \) and SALIENCY = \( \frac{40^2}{0} = \infty \)). This ordering does not predict any of the difficulty ordering of the participants results. Adding noise to the Saliency model can predict the order of four out of five participants. SDT-based models cannot predict any of these difficulty ordering because no-matter what amount of noise is added, condition 1 will be considered the hardest, as there the distractors are most similar to the target.

To get a more quantitative comparison between the SALIENCY, NOISY-SALIENCY, COVER and NOISY-COVER models, we measured the correlation-coefficients ($r$) between the predictions of each model and the participants accuracy [64]. We report the correlation-coefficient $r$, its significance and the resulting noise parameter ($\sigma$, if applicable) for each participant in the
right-side of Table 6.1. In the correlation-coefficient test we tried to check
whether there exists some linear transform that connects the measures sug-
gested by these models and the participants’ accuracy. For the SALIENCY
model \( r \) cannot be calculated in this experiment as the SALIENCY of some
conditions is \( \infty \). Here too we find that the COVER predicts the performance
of four out of the five participants, NOISY-COVER predicts 5/5, and SDT
none.

We also compare the experiment’s results to the predictions of the NOISY-
FLNN model, a SDT-based model, Rosenholtz’s Best-Normal and RCref
models and the Temporal-Serial model. In Figure 6.3 and Table 6.2 we report
the predictions that gave the lowest chi-square (\( \chi^2 \)) for each combination
of model and participant. Specifically, we report for each case the values of the
parameters that gave that fit (\( \sigma, k, \) or both), the reduced-chi-square (\( \chi^2/df \)),
and whether the chi-square-test (\( \chi^2 \)-test) rejected the model. We employed
the reduced-chi-square measure because it enables the comparison between
models that use a different number of parameters ([81]). A model with more
parameters get a lower fitting grade relative to a model with less parame-
ter suggesting the same prediction. 

\[
\chi^2/df = \frac{1}{c-p} \sum_{i=1}^{c} \frac{(Acc_i-Prediction_i)^2}{SE_i^2},
\]

where \( c \) is the number of conditions, \( p \) is the number of the model param-
eters, \( Acc_i \) is the accuracy of the participant on condition \( i \), \( Prediction_i \) is
the prediction of the model for condition \( i \), and \( SE \) is the standard-error of
the participant’s accuracy for the \( i \)-th condition.) The lower the \( \chi^2/df \) value
is, the better the model can predict the results. The \( \chi^2 \)-test reports whether
the probability of obtaining a \( \chi^2 \) larger than the one measured given that
the data followed the model, and taking into account the degrees of freedom
\( df \), is higher than 0.05. If it is not, the model is rejected.

For four out of the five participants, NOISY-FLNN’s predictions get the
lowest (best) $\chi^2/df$ score. It is rejected by only one out of five $\chi^2$-tests, while the Best-Normal model is rejected by three $\chi^2$-tests, the RCref by four tests, and the SDT and Temporal-Serial models are rejected by all five $\chi^2$-tests.

6.2.2 Experiment 2: Orientation Bidirectional

Method

Observers: Four students (A.D., A.A., M.D., L.F.) from the University Of Haifa with normal or corrected to normal vision, who were naive to the purpose of the study, and did not participate in the other experiments.

Stimuli and Apparatus: The stimuli and apparatus were identical to Experiment 1 except for the following: The search display was consisted of 18 line segments (Figure 6.2b). The orientation of the distractors lines presented in each of the 5 conditions of this experiment was as follows: In condition 1, half of the distractors had an orientation of 20° and the other half of −20°. In condition 2, the orientation of the distractors was approximately equally divided between −35°, −20°, 20°, and 35°. Condition 3, employed the same type of distractors as condition 2, but with more of −20°, and 20° and less of −35° and 35° (see details in Figure 6.1b). In condition 4 the orientation of the distractors was approximately equally divided between −50°, −20°, 20° and 50°. Finally, condition 5 employed the same type of distractors as condition 4, but with more of −20°, and 20° and less of −50° and 50°.

Procedure: The procedure was identical to Experiment 1 except for the following: A single session was consisted of five blocks of 80 trials, each corresponding to one of the five experimental conditions. Each observer participated in 3 such sessions, for a total of 240 trials per condition and 1200 trials for the entire experiment.
Results

The mean accuracy and standard-error for each participant and each condition are reported in Figure 6.4. A.D.’s performance was with no significant difference between all 5 conditions. For all other participants there were some significant performance differences between conditions. There is a significant difference between the performance of A.A. in condition 3 (distractors of $-35^\circ$, $-20^\circ$, $20^\circ$, and $35^\circ$; more of $\pm 20$ than $\pm 35$) and his performance in conditions 4 and 5 (distractors at $\pm 20$ and $\pm 50$). For M.D. condition 1 (all distractors at $\pm 20$) was significantly harder than the rest of the conditions. For L.F., conditions 1 and 5 (more at $\pm 20$, less at $\pm 50$) were significantly harder than conditions 4 (equal at $\pm 20$ and $\pm 50$) and 2 (equal at $\pm 20$ and $\pm 35$), conditions 1,3, and 5 were significantly harder than condition 2.

Predictions and Discussion

For three out of the four participants there was a significant difference in the accuracy achieved between some of the conditions. While Rosenholtz’s models predict no difficulty differences for cases in which the distractors’ feature-values are symmetric around the target feature-value (In such cases Rosenholtz’s SALIENCY and NOISY-SALIENCY measure is 0 for all conditions), the models suggested here can predict difficulty differences between different symmetric conditions.

In conditions 2 and 3 the same distractors are displayed, with different amount of each type. (In condition 3 the distractors that are more similar to the target are more frequent.) The same is true for conditions 4 and 5. The idea behind designing such pairs of experiments was to test FLNN’s prediction ability relative to the COVER prediction ability. The COVER model has similar predictions for conditions 2&3 and conditions 4&5. FLNN
also predicts no difference between conditions 2 & 3, but predicts a difference in difficulty between conditions 4 & 5. Condition 5, in which the distractors that are more similar to the target are more frequent, should be harder according to FLNN. Only with one out of four participants performance in condition 5 was harder than condition 4.

Similar to Experiment 1, we compare the difficulty ordering prediction abilities of COVER, NOISY-COVER, Rosenholtz’s SALIENCY, NOISY-SALIENCY and SDT (Table 6.3). The SDT model prediction depends on how similar the distractors are to the target. Therefore, condition 1 (all at ±20) is considered the hardest, then condition 3 (more at ±20, less at ±35), followed by condition 5 (more at ±20, less at ±50), 2 (equal at ±20 and ±35), and finally 4 (equal at ±20 and ±50). This predictions follow to a certain degree the performance of M.D., and with some differences the ordering of the other participants. The COVER model predicts here that conditions 4 and 5 with COVER=5 are harder than conditions 1, 2, and 3 with COVER=3. It therefore can only predict, to a certain degree, A.D.’s order of difficulty. Even after adding noise, the NOISY-COVER cannot predict condition 1 to be significantly the hardest, and can only predict A.D.’s order of difficulty. The SALIENCY and NOISY-SALIENCY models predict a 0 saliency for all conditions as the target value and the mean of the distractors meet. $r$ cannot be calculated for these models as it is not-defined for constant vectors (as no linear transform can transform the constant values to the participants’ accuracies). Therefore, they cannot explain the significant differences in the difficulty of these symmetric bidirectional conditions. The NOISY-COVER model succeeds in providing significant correlation-coefficients for three out of four participants.

The quantitative predictions comparison between NOISY-FLNN model,
a SDT-based model, Rosenholtz’s Best-Normal and RCref models and the Temporal-Serial model are reported in Table 6.4 and Figure 6.4. Both the SDT model and NOISY-FLNN models can predict performance of three out of four participants (i.e., not rejected by $\chi^2$-test). The Best-Normal, RCref, and Temporal-Serial models are rejected by the $\chi^2$-test for two out of four participants.

6.2.3 Experiment 3: Color Unidirectional

To see whether these results are replicated with a different feature, we performed similar experiments but with the feature of color rather than orientation.

Method

Observers: Four students (D.A., S.M., E.D., G.S.) from the University of Haifa with normal or corrected to normal vision, who were naive to the purpose of the study, and did not participate in the other experiments.

Stimuli, Apparatus, and Procedure: The stimuli, apparatus and procedure were identical to Experiment 1 except for the following: The search elements consisted of 18 colored-disk, with a diameter of 1.2°, presented on a black background (Figure 6.2c). The disks were randomly placed within a non-visible circle with a radius of 4.5°. A detailed description of the various colors employed in this experiment is given in Figure 6.1c and Table 6.5. These colors were selected after we measured the appropriate $u'$, $v'$, cd/m² by a Tektronix J18 LumaColor™ II Photometer. We used the $u'v'L^*$ color-space because it was designed so that the distances in the color-space are linear with differences in color perception (C.I.E. 1978; for equations see e.g.,[83]). Figure 6.5a depicts the exact distances in $u'v'$ space (when $L^*$ is
constant). Approximately, we tried to keep $v'$ also constant and change only $u'$. Hence, the distance in feature space between the search items (i.e., their feature value) is computed based on the $u'$-value differences. Additionally, we chose the black color as the background because than it is approximately in equal perceptual-distance from all stimuli and, therefore, it could be ignored by the model’s considerations (Figure 6.5c). The color of the target-disk was always gray (0 feature-value; Table 6.5), and it was present on the first or second interval equally often. The color of the distractor-disks presented in each of the four conditions of this experiment was as follows (see also Figure 6.1c): In condition 1, all the distractors had the same greenish color (-10 feature-value). In condition 2, half of the distractors had one greenish color (-10 feature-value) and the other half had another greenish color (-30 feature-value). In condition 3, all the distractors had the same greenish color (-30 feature-value). Finally, in condition 4, half of the distractors had one greenish color (-30 feature-value) and the other half had another greenish color (-50 feature-value).

Results

The mean accuracy and standard-error for each participant and each condition are reported in Figure 6.6. For participants D.A. and S.M. condition 1 (all distractors at $-10$) was the hardest, then significantly easier were conditions 2 (distractors at $-10$ and $-30$) and 4 (distractors at $-30$ and $-50$), without significant difference between them. Condition 3 (all distractors at $-30$) was significantly the easiest. For participants E.D. and G.S. conditions 2 (distractors at $-10$ and $-30$) and 1 (all distractors at $-10$) where significantly harder than conditions 3 (all distractors at $-30$) and 4 (distractors
at −30 and −50). For GS there was also a significant performance difference between condition 2 (harder) and 1 (easier).

Predictions and Discussion

Similar to experiment 1, experiment 3 was designed to: a) get more evidence for the fact that tasks with heterogeneous distractors may be harder than homogeneous ones, even when the distractors are less similar to the target in the heterogeneous case. This claim is modeled by the COVER and the SALIENCY models but is not predicted by SDT-based models. This behavior, that was clearly demonstrated in experiment 1 for orientation-search, is not consistent for the color-search experiment reported above. b) Show that the difficulty is effected by the target-distractors feature-space distances relative to the the distractors-distractors feature-space distances, and not just by the absolute values of either. If this assumption was correct, the difference in difficulty between condition 1 and 2 should be greater than the difference in difficulty between conditions 3 and 4. The experiment verified this hypothesis for two out of four participants.

Considering the results reported in Table 6.6, the original COVER model cannot predict this participants’ orders of difficulty: Similarly to experiment 1, COVER predicts condition 2 where COVER=3 to be more difficult than the other conditions where COVER=2. The NOISY-COVER is able to predict the ordering for all four participants and passes the correlation-coefficient test for two out of four participants. SDT cannot predict the difficulty ordering of all four participants. It can predict that condition 1 (all distractors at −10) is the hardest as the distractors have the highest probability to be confused with the target, but cannot predict condition 4 (distractors at −30 and −50) being harder than condition 3 (all distractors at −30) as
with the latter the distractors are more similar to the target. Rosenholtz’s SALIENCY model without noise cannot predict these orders of difficulty and since it suggests an infinite value, it is impossible to calculate correlation-coefficients. The NOISY-SALIENCY model is able to predict the ordering for one out of the four participants and passes the correlation-coefficient test for that one participant.

As can be seen in Figure 6.6 and Table 6.7, the SDT and Temporal-Serial models do not pass any $\chi^2$-test, the Best-Normal and RCref models pass one out of four, and NOISY-FLNN two out of four cases.

6.2.4 Experiment 4: Color Bidirectional

Method

Observers: Four students (R.A., O.R., R.I., A.O.) from the University Of Haifa with normal or corrected to normal vision, who were naive to the purpose of the study, and did not participate in the other experiments.

Stimuli and Apparatus: The stimuli and apparatus were identical to Experiment 3 except for the following: The color of the distractor-disks presented in each of the 5 conditions of this experiment was as follows (see also Figure 6.1d, Table 6.5, and a sample display in Figure 6.2d): In condition 1, half of the distractors had a greenish color (-15 feature-value) and the other half had a pinkish color (15 feature-value). In condition 2, the color of the distractors was approximately equally divided between 2 greenish colors and 2 pinkish colors (feature-values of -25, -15, 15, 25, respectively). Condition 3, employed the same type of distractors as condition 2, but with more of -15, and 15 and less of -25 and 25 (see details in Figure 6.1d). In condition 4, the color of the distractors was approximately equally divided between 2 greenish colors
and 2 pinkish colors (feature-values of -35, -15, 15, 35, respectively). Finally, condition 5 employed the same type of distractors as condition 4, but with more of -15, and 15 and less of -35 and 35.

Procedure: The procedure was identical to Experiment 2.

Results

The mean accuracy and standard-error for each participant and each condition are presented in Figure 6.7. For O.R. and R.I. there was no significant difference in performance between all 5 conditions. For R.A. conditions 2 (equal number of distractors at -25, -15, 15, and 25) and 5 (more at ±15, less at ±35) were significantly harder than conditions 1 (all at ±15), 3 (more at ±15, less at ±25), and 4 (equal at ±15 and ±35). For A.O. conditions 4 and 5 were significantly harder than conditions 1, 2, and 3.

Predictions and Discussion

As in experiment 2, the fact that there are sometimes significant differences in performance between conditions for the same participant, does not agree with the SALIENCY and NOISY-SALIENCY models that predict equal difficulty for all the conditions. Similarly, these results do not agree with the linear-separable-model ([14]) suggesting that all tasks are equally hard whenever there are distractor colors from the two sides of the target color in the color-space.

The original COVER model predicts that conditions 4 and 5 should be more difficult (COVER=5) than conditions 1, 2, and 3 (COVER=3). Therefore, it can predict only the difficulty order of A.O. (Table 6.8). The NOISY-COVER is able to predict the ordering for three out of four participants and passes the correlation-coefficient test for all participants. The SDT cannot
predict the ordering of all four participants, as it considers condition 1 as hardest, followed by 3, 5, 2, and finally 4. Rosenholtz’s Saliency and Noisy-Saliency models can predict the difficulty ordering of R.A. and R.I., as they performed similarly for all conditions. Same as for experiment 2, the correlation-coefficients cannot be calculated for these models for this experiment.

Considering Table 6.9, RCref is rejected by three out of four $\chi^2$-tests, SDT is rejected by two out of four tests, NOISY-FLNN is rejected by one out of four, and Temporal-Serial and Best-Normal are not rejected by all four tests.

6.2.5 Predicting Rosenholtz’s[72] Orientation Experiments

[72] compares and reports the ability of the Best-Normal model, the RCref model, and the SDT-based model to predict performance in orientation-search experiments. To further test the models suggested here, we tried also to use them to predict some of those experiments (denoted there experiment 1 and 2).

In all conditions of experiment 1 the target was a $0^\circ$ horizontal line segment while the distractors changed between the four different conditions. In condition 1 all distractors had orientation of $30^\circ$. In condition 2, 1/3 of the distractors had orientation of $30^\circ$ and 2/3 of the distractors had orientation of $50^\circ$. In condition 3, 1/3 of the distractors had orientation of $30^\circ$, 1/3 of the distractors had orientation of $50^\circ$ and 1/3 of the distractors had orientation of $70^\circ$. In condition 4, 1/3 of the distractors had orientation of $30^\circ$ and 2/3 of the distractors had orientation of $70^\circ$. The task was a 2IFC detection task with 36 items in each display. In the conditions of experiment 2 the
target and the distribution of the distractors is exactly the same. The only
difference is that 8 elements appear in each display (instead of 36). For more
details about the method used see [72].

Models prediction for [72] experiment 1

For one out of two participants conditions 3 (1/3 30°, 1/3 50°, 1/3 70°) and 4
(1/3 30°, 2/3 70°) were significantly harder than conditions 1 (30°) and 2 (1/3
30°, 2/3 50°). For the other participant condition 2 was significantly easier
than the other three conditions, and condition 4 was significantly harder than
condition 1.

As Rosenholtz suggests, the SDT-based method fails to predict such dif-
ficulty ordering, as it considers condition 1 the hardest. The COVER model
predicts here that conditions 3 and 4 are harder (COVER=3) than conditions
1 and 2 (COVER=2), and therefore qualitatively can predict the behavior of
the two participants.

Comparing the NOISY-COVER and NOISY-SALIENCY models for this
experiment (Table 6.10) reveals that the NOISY-COVER model succeeds in
predicting the difficulty ordering for both participants, while the NOISY-
SALIENCY model predicts the ordering of one participant. Both do not
pass the correlation-coefficient test for any participant.

As can be seen in Table 6.11, all models are rejected by the \( \chi^2 \)-test trying
to predict B.L.B.’s results. Only Best-Normal and NOISY-FLNN pass the
\( \chi^2 \)-test for R.E.R.. For both participants NOISY-FLNN’s \( \chi^2 /df \) values are
the lowest (best)\(^2\); see also Figure 6.8.

\(^2\)Note that since we looked for fits that minimize \( \chi^2 \), and Rosenholtz minimized the
sum of square differences, there are minor differences between the predictions of the SDT,
Best-Normal and RCref reported here and those reported in [72].
Models prediction for [72] experiment 2

Two other participants participated in experiment 2 of [72]. For both participants there were no significant differences in performance between all conditions.

According to SDT condition 1 (30°) should be the most difficult followed by conditions 2 (1/3 30°, 2/3 50°), then 3 (1/3 30°, 1/3 50°, 1/3 70°) and finally condition 4 (1/3 30°, 2/3 70°). Both the COVER model and the Saliency model predict conditions 3 and 4 should be harder than conditions 1 and 2. The Saliency model also predicts condition 2 to be harder than condition 1. Both models could not predict the participants’ order of difficulty (Table 6.12). The NOISY-COVER model can predict the ordering for both participants, and passes both correlation-coefficient tests. The NOISY-SALIENCY could not suggest the exact difficulty ordering of the participants but also passes both correlation-coefficient tests.

As can be seen in Table 6.13 and Figure 6.9 the NOISY-FLNN model provides the lowest (best) $\chi^2/df$ values for both participants. The SDT model is rejected by one out of two $\chi^2$-tests, while all other models pass both $\chi^2$-tests.

6.3 Discussion

The main goal of this work was to test the ability of the COVER and FLNN models suggested previously for artificial intelligence in [7, 8], to predict human visual-search performance. These models have the ability to capture the effect of the visual similarity between the target and the distractors and between the different distractors on visual search difficulty. As suggested by [28], these effects can account for the fact that search tasks with het-
heterogeneous distractors can be more difficult than those with homogeneous
distractors, a phenomena that is not accounted for by other computational
models, such as SDT-based models. The originally proposed COVER and
FLNN models, however, suggest only a measure that takes the relative differ-
ences in stimuli similarities and not the absolute ones. This does not allow,
for instance, to predict the difference in difficulty between two homogeneous
condition where in one the distractors are quit similar to the target and in
the second the target is much more distinguishable. To solve these draw-
backs, here we have suggested extensions of those models that consider the
observed input as noisy (as suggested by SDT-based models).

We have tested the prediction abilities of the original and the extended
models for both orientation-search and color-search tasks, and compared
them to predictions of a few other models, using both qualitative and quan-
titative measures. Especially, we judged these models against models sug-
gested by Rosenholtz [71, 72], which were also designed to account for the
performance differences suggested by [28]. As can be clearly seen in Ta-
ble 6.14, these comparison revealed that our models predicted human per-
formance better than the other models. The NOISY-COVER succeeded in
predicting the search-task difficulty order of 17 out of 21 cases, while NOISY-
SALIENCY succeeded in only 9. The correlation-coefficients of the NOISY-
COVER predictions passed the significance test 16 times, while the NOISY-
SALIENCY only 3 times. In the comparison of the NOISY-FLNN, SDT,
Best-Normal, Temporal-Serial and RCref predictions: the NOISY-FLNN
passed the \( \chi^2 \)-test 15 times, followed by Best-Normal (12), Temporal-Serial
(8), RCref (7) and SDT (6). NOISY-FLNN achieved the lowest reduced-chi-
square (\( \chi^2/df \)) in 10 out of the 21 cases, followed by RCref (5/21).

However, our models are still rejected in some cases implying that these
models do not account for all the factors mediating human search performance. Some of the possible reasons might be:

1. The models do not take into consideration the spatial position of the stimuli. It was shown, for instance, that targets appearing at peripheral locations are detected more slowly and less accurately than those appearing near the central fixation point (The eccentricity effect, [20, 21]).

2. The COVER model suggested that computer-vision recognition tasks might benefit from grouping stimuli by visual similarity. This grouping considers all the display at once and groups items without location considerations. However, it is known from Gestalt rules [96] that both similarity and proximity effect grouping in human visual processing. The tested models do not consider the effect of proximity.

3. The predictions of the models assume that the feature-spaces used are linear. For the orientation-search prediction it assumes that the observed difference between different orientations is proportional to the difference in the degrees measured. However, it was suggested, for instance, that orientation sensitivity is better around principle meridians than around oblique ones. (The oblique effect, [4, 18]). Moreover, by attempting to find the differential threshold (JND) for 15 different orientations (rather than only the 4 main ones), [58] showed that the JND increases as a function of obliquity from the principal orientation up to 20° obliquity and then levels off. For the color-search prediction it assumes that the differences in the L*u’v’ color-space is proportional to the differences recognized by human observers. This assumption is commonly used, however, recently there have been some criticism about the accuracy of this color system (see e.g., [36]).
Following this work, it should be interesting to modify the models tested here so that they consider the above findings, and to then test their prediction abilities. Also, we are trying to phrase the predictions of NOISY-COVER and NOISY-FLNN in an analytic deterministic way, so it can be more accurately predicted for the distribution of input rather than using simulation with many repeats as it is done now.
Figure 6.1: Feature values of the different items in each of the four experiments. Experiments 1 and 2 are orientation-search tasks (a, b). Experiments 3 and 4 are color-search tasks (c, d). In each experiment the numbers below the horizontal line describe the feature-value (orientation or color) of the stimulus property. Above each point it is indicated whether it is a target (T) or a distractor (D), and the number of stimuli with this value, both for the target-present and target-absence displays. (e.g., D-17/18 means there are 17 such distractors in a target-present display and 18 such distractors in a target-absent display.) For the corresponding colors used in experiments 3 and 4 in the L*u*v’ and RGB color spaces see table 6.5.
Figure 6.2: Examples of target-present displays from each experiment: a) experiment 1, b) experiment 2, c) experiment 3, and d) experiment 4.
Table 6.1: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 1 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a – sign indicates that in the model prediction at least two conditions are in reverse order, and a ~ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients (r) between the participants’ accuracy and the models predictions. \( p^+ (p^-) \) indicates significance(no significance) of \( r \). \( \sigma \) is the noise level corresponding to the best \( r \). For the Saliency model \( r \) cannot be calculated as the Saliency of some conditions is \( \infty \).
Figure 6.3: Experiment 1 results and predictions of the different models. The data points are the participant’s accuracy for the four conditions, together with the standard error. The lines connect the model’s accuracy predictions for the corresponding conditions.
Table 6.2: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Experiment 1. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (– if the model is rejected and + otherwise), and the parameters that gave the best fit.
Table 6.3: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 2 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a – sign indicates that in the model prediction at least two conditions are in reverse order, and a ∼ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients (r) between the participants’ accuracy and the models predictions. \( p^+ \) (\( p^- \)) indicates significance (no significance) of \( r \). \( \sigma \) is the noise level corresponding to the best \( r \). For the SALIENCY and NOISY-SALIENCY models \( r \) cannot be calculated as the SALIENCY is constant for all conditions.
Figure 6.4: Experiment 2 results and predictions of the different models. The data points are the participant’s accuracy for the five conditions, together with the standard error. The lines connect the model’s accuracy predictions for the corresponding conditions.
Table 6.4: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Experiment 2. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (− if the model is rejected and + otherwise), and the parameters that gave the best fit
Figure 6.5: A graphic demonstration of the colors used in Experiments 3 and 4 (respectively) in the L*u’v’ color-space. Panels a and b depict the colors used in Experiments 3 and 4 (respectively) in the u’v’ space, and demonstrate the feature-space distances. In panels c and d the color black ((0,0) point) is also plotted to demonstrate the approximately equal distances of all stimuli from the background.
<table>
<thead>
<tr>
<th>description</th>
<th>u'</th>
<th>v'</th>
<th>cd/m² = Y</th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>black</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-50</td>
<td>0.138</td>
<td>0.422</td>
<td>22.1</td>
<td>0</td>
<td>185</td>
<td>158</td>
</tr>
<tr>
<td>-35</td>
<td>0.151</td>
<td>0.425</td>
<td>23.0</td>
<td>49</td>
<td>180</td>
<td>159</td>
</tr>
<tr>
<td>-30</td>
<td>0.150/60</td>
<td>0.427</td>
<td>23.6</td>
<td>73</td>
<td>178</td>
<td>159</td>
</tr>
<tr>
<td>-25</td>
<td>0.162</td>
<td>0.426</td>
<td>23.2/3</td>
<td>81</td>
<td>177</td>
<td>159</td>
</tr>
<tr>
<td>-15</td>
<td>0.173</td>
<td>0.426/7</td>
<td>24.1</td>
<td>115</td>
<td>172</td>
<td>160</td>
</tr>
<tr>
<td>-10</td>
<td>0.179</td>
<td>0.427</td>
<td>24.3/4</td>
<td>130</td>
<td>168</td>
<td>160</td>
</tr>
<tr>
<td>- 0 (target)</td>
<td>0.189</td>
<td>0.427</td>
<td>24.6</td>
<td>160</td>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>15</td>
<td>0.205/6</td>
<td>0.424</td>
<td>24.2/3</td>
<td>204</td>
<td>143</td>
<td>161</td>
</tr>
<tr>
<td>25</td>
<td>0.211</td>
<td>0.421</td>
<td>23.8/9</td>
<td>218</td>
<td>135</td>
<td>162</td>
</tr>
<tr>
<td>35</td>
<td>0.223</td>
<td>0.416/7</td>
<td>23.1</td>
<td>240</td>
<td>121</td>
<td>163</td>
</tr>
</tbody>
</table>

Table 6.5: Color values used in experiments 3 and 4
Figure 6.6: Experiment 3 results and predictions of the different models. The data points are the participant’s accuracy for the four conditions, together with the standard-error. The lines connect the model’s accuracy predictions for the corresponding conditions.
Table 6.6: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 3 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a – sign indicates that in the model prediction at least two conditions are in reverse order, and a ~ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients ($r$) between the participants’ accuracy and the models predictions. $p^+$ ($p^-$) indicates significance (no significance) of $r$. $\sigma$ is the noise level corresponding to the best $r$. For the SALIENCY model $r$ cannot be calculated as the SALIENCY of some conditions is $\infty$.

<table>
<thead>
<tr>
<th>Participant</th>
<th>SDT</th>
<th>COVER</th>
<th>NOISY COVER</th>
<th>Saliency</th>
<th>NOISY Saliency</th>
<th>Cover</th>
<th>NOisy Cover</th>
<th>Saliency</th>
<th>NOisy Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.A.</td>
<td>–</td>
<td>–</td>
<td>+</td>
<td>–</td>
<td>~</td>
<td>0.076 $p^-$</td>
<td>0.862 $p^+$ $\sigma = 11.0$</td>
<td>not - applicable</td>
<td>0.900 $p^- \sigma = 8.0$</td>
</tr>
<tr>
<td>S.M.</td>
<td>–</td>
<td>–</td>
<td>+</td>
<td>–</td>
<td>~</td>
<td>0.192 $p^-$</td>
<td>0.883 $p^- \sigma = 11.0$</td>
<td>not - applicable</td>
<td>0.945 $p^- \sigma = 7.4$</td>
</tr>
<tr>
<td>E.D.</td>
<td>~</td>
<td>~</td>
<td>+</td>
<td>–</td>
<td>+</td>
<td>0.576 $p^-$</td>
<td>0.993 $p^+ \sigma = 4.3$</td>
<td>not - applicable</td>
<td>0.996 $p^+ \sigma = 8.7$</td>
</tr>
<tr>
<td>G.S.</td>
<td>–</td>
<td>~</td>
<td>+</td>
<td>–</td>
<td>~</td>
<td>0.851 $p^-$</td>
<td>0.997 $p^+ \sigma = 2.8$</td>
<td>not - applicable</td>
<td>0.880 $p^- \sigma = 8.6$</td>
</tr>
</tbody>
</table>
Table 6.7: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Experiment 3. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (– if the model is rejected and + otherwise), and the parameters that gave the best fit
Table 6.8: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict Experiment’s 4 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a – sign indicates that in the model prediction at least two conditions are in reverse order, and a ∼ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients (r) between the participants’ accuracy and the models predictions. $p^+$($p^-$) indicates significance(no significance) of r. $\sigma$ is the noise level corresponding to the best r. For the SALIENCY and NOISY-SALIENCY models r cannot be calculated as the SALIENCY is constant for all conditions.
Figure 6.7: Experiment 4 results and predictions of the different models. The data points are the participant's accuracy for the five conditions, together with the standard-error. The lines connect the model's accuracy predictions for the corresponding conditions.
<table>
<thead>
<tr>
<th>participant</th>
<th>NOISY-FLNN</th>
<th>SDT</th>
<th>Best-Normal</th>
<th>Temporal-Serial</th>
<th>RCref</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\chi^2/df$</td>
<td>$\chi^2$-test</td>
<td>$\sigma$</td>
<td>$k$</td>
<td>$\chi^2/df$</td>
</tr>
<tr>
<td>R.A.</td>
<td>3.665</td>
<td>–</td>
<td>5.6</td>
<td>5.0</td>
<td>3.028</td>
</tr>
<tr>
<td>O.R.</td>
<td>1.067</td>
<td>+</td>
<td>6.0</td>
<td>5.0</td>
<td>0.252</td>
</tr>
<tr>
<td>R.I.</td>
<td>1.069</td>
<td>+</td>
<td>5.6</td>
<td>5.4</td>
<td>0.782</td>
</tr>
<tr>
<td>A.O.</td>
<td>1.759</td>
<td>+</td>
<td>5.5</td>
<td>1.6</td>
<td>3.131</td>
</tr>
</tbody>
</table>

Table 6.9: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Experiment 4. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (– if the model is rejected and + otherwise), and the parameters that gave the best fit.
Table 6.10: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict ([72]) Experiment 1 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a – sign indicates that in the model prediction at least two conditions are in reverse order, and a ~ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients (r) between the participants’ accuracy and the models predictions. $p^+$ ($p^-$) indicates significance (no significance) of r. $\sigma$ is the noise level corresponding to the best r. For the SALIENCY model r cannot be calculated as the SALIENCY of some conditions is $\infty$. 

<table>
<thead>
<tr>
<th>participant</th>
<th>SDT</th>
<th>COVER</th>
<th>NOISY COVER</th>
<th>SALIENCY</th>
<th>NOISY SALIENCY</th>
<th>COVER</th>
<th>NOISY COVER</th>
<th>SALIENCY</th>
<th>NOISY SALIENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.E.R.</td>
<td>~</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>~</td>
<td>0.904 p$^-$</td>
<td>0.910 p$^-$</td>
<td>$\sigma$ = 1.5</td>
<td></td>
</tr>
<tr>
<td>B.L.B.</td>
<td>~</td>
<td>~</td>
<td>+</td>
<td>–</td>
<td>~</td>
<td>0.842 p$^-$</td>
<td>0.918 p$^-$</td>
<td>$\sigma$ = 8.6</td>
<td>not-applicable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.894 p$^-$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.903 p$^-$</td>
</tr>
</tbody>
</table>
Figure 6.8: Rosenholtz’s Experiment 1 results and predictions of the different models. The data points are the participant’s accuracy for the four conditions, together with the standard-error. The lines connect the model’s accuracy predictions for the corresponding conditions.
Table 6.11: A quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Rosenholtz’s Experiment 1. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (– if the model is rejected and + otherwise), and the parameters that gave the best fit.

<table>
<thead>
<tr>
<th>participant</th>
<th>NOISY-FLNN</th>
<th>SDT</th>
<th>Best-Normal</th>
<th>Temporal-Serial</th>
<th>RCref</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\chi^2/df$</td>
<td>$\chi^2$-test</td>
<td>$\sigma$</td>
<td>$k$</td>
<td>$\chi^2/df$</td>
</tr>
<tr>
<td>R.E.R.</td>
<td>1.565</td>
<td>+</td>
<td>6.1</td>
<td>2.7</td>
<td>15.964</td>
</tr>
<tr>
<td>B.L.B.</td>
<td>5.725</td>
<td>–</td>
<td>0.8</td>
<td>1.6</td>
<td>15.025</td>
</tr>
</tbody>
</table>
Table 6.12: The ability of the COVER, NOISY-COVER, Rosenholtz’s Saliency model and SDT-based models to predict ([72]) Experiment’s 2 results. The left side of the table reports whether the model can qualitatively predict the difficulty ordering of the conditions for each participant. A + sign indicates that the exact difficulty ordering can be predicted by the model, a − sign indicates that in the model prediction at least two conditions are in reverse order, and a ~ sign indicates that there is no reverse ordering, but some present/absence difficulty differences are not predicted. The right side of the table reports the best correlation-coefficients (r) between the participants’ accuracy and the models predictions. $p^+$ ($p^-$) indicates significance (no significance) of $r$. $\sigma$ is the noise level corresponding to the best $r$. For the Saliency model $r$ cannot be calculated as the Saliency of some conditions is $\infty$.

<table>
<thead>
<tr>
<th>participant</th>
<th>SDT</th>
<th>COVER</th>
<th>NOISY COVER</th>
<th>Saliency</th>
<th>NOISY Saliency</th>
<th>COVER</th>
<th>NOISY COVER</th>
<th>Saliency</th>
<th>NOISY Saliency</th>
</tr>
</thead>
<tbody>
<tr>
<td>J.A.K.</td>
<td>~</td>
<td>~</td>
<td>+</td>
<td>~</td>
<td>~</td>
<td>0 p−</td>
<td>0.994 p+$\sigma = 14.6$</td>
<td>not applicable</td>
<td>0.998 p+$\sigma = 13.0$</td>
</tr>
<tr>
<td>J.O.E.</td>
<td>~</td>
<td>~</td>
<td>+</td>
<td>~</td>
<td>~</td>
<td>0.302 p−</td>
<td>0.986 p+$\sigma = 13.0$</td>
<td>not applicable</td>
<td>0.982 p+$\sigma = 11.9$</td>
</tr>
</tbody>
</table>
Figure 6.9: Rosenholtz’s Experiment 2 results and predictions of the different models. The data points are the participant’s accuracy for the four conditions, together with the standard-error. The lines connect the model’s accuracy predictions for the corresponding conditions.
Table 6.13: A Quantitative comparison of the predictions of NOISY-FLNN, SDT, Best-Normal, Temporal-Serial, and RCref models for Rosenholtz’s Experiment 2. Reports for each model and participant, the reduced-chi-square ($\chi^2/df$) values, the chi-square-test result (− if the model is rejected and + otherwise), and the parameters that gave the best fit.
### Table 6.14: Summary of the various models prediction abilities.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COV</td>
<td>N-COV</td>
<td>SAL</td>
<td>N-SAL</td>
</tr>
<tr>
<td>Exp.1</td>
<td>3/5</td>
<td>5/5</td>
<td>0/5</td>
<td>4/5</td>
</tr>
<tr>
<td>Exp.2</td>
<td>0/4</td>
<td>1/4</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>Exp.3</td>
<td>0/4</td>
<td>4/4</td>
<td>0/4</td>
<td>1/4</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>1/4</td>
<td>3/4</td>
<td>2/4</td>
<td>2/4</td>
</tr>
<tr>
<td>Ros.Exp.1</td>
<td>1/2</td>
<td>2/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>Ros.Exp.2</td>
<td>0/2</td>
<td>2/2</td>
<td>0/2</td>
<td>0/2</td>
</tr>
</tbody>
</table>

From left to right: 1. The number of participants out of the total number of participants for which the COVER (COV), NOISY-COVER (N-COV), SALIENCY (SAL), and NOISY-SALIENCY (N-SAL) were able to predict the difficulty ordering. 2. The number of participants out of the total number of participants for which the predictions of these models achieved significant correlation coefficients. 3. The number of participants out of the total number of participants for which each of the NOISY-FLNN (FLNN), the SDT, the Best-Normal (B-N), the Temporal-Serial (T-S) and the RCref models achieved the fit that gave the lowest reduced-chi-square($\chi^2/df$) (relative to the other models). 4. The number of participants out of the total number of participants for which each of these models achieved a fit that was not rejected by the $\chi^2$-test.
Chapter 7

General Discussion

This thesis is based on the papers by Tamar Avraham and the co-authors Michael Lindenbaum and Yaffa Yeshurun [5, 7, 6, 8, 9, 10, 12, 11].

This work suggests two novel directions for computer visual attention, each is specified in general and demonstrated by a practical algorithm. One suggests the dynamic visual search framework. The second suggests a new concept for saliency detection.

Most previous models of visual-attention are based on the feature integration theory and take a feature-based approach, associating a saliency value for each image point based on its local contrast. Our approach is very different. We take an object-based approach, as we propose to start by defining the set of candidate sub-images resulting from a relatively simple preceding segmentation process. We suggest algorithms that set priorities to the different candidates based on validated observations that are statistically modeled. In a mathematically well defined sense, the priorities are estimated probabilities for candidates to be associated with target objects. The resulting estimates are based on global considerations. In addition, we suggest methods for dynamically changing those priorities incorporating the feedback of
a recognizer after it is applied on previously selected candidates.

As expected, using inner-scene similarity allows the majority of targets to be found earlier than when using a sequential search. This was experimentally verified to be the case even in comparison to a very efficient detection algorithm.

Intuitive assumption are quantified either by a deterministic or a stochastic approach. Taking a quantitative approach allowed us not only to optimize the search but to also quantitatively predict its performance. Search performance naturally depends both on the task difficulty and the algorithm effectiveness. The COVER measure of the search task difficulty, proposed here, is an inherent, algorithm independent, measure and bounds the performance of every possible algorithm.

While we have tested the proposed algorithms using a variety of recognition and segmentation processes, it should be interesting and useful to extend the analysis so that it takes into account the common imperfections of these processes.

Feature selection has a strong effect on attention and visual search. There is place to extend this work by finding criterions for optimizing feature selection in this context. The feature weighting can, for instance, change dynamically as a result of a learning process that is informed from the recognition oracle’s responses.

We have presented and tested each of the Esaliency and VSLE algorithms separately. It is natural that they are combined: Esaliency will first recognize saliency setting the initial priorities. Then, those priorities will be used as initial estimates for VSLE that will proceed. Intuitively, Esaliency will set high priorities to a few types of objects and will ‘reject’ background regions, followed by VSLE that will direct the attention to one candidate of each
type of object until the relevant category is recognized. The EsaliencyVSLE application described in appendix A supports this combination, but we have not yet tested the results of this cooperation intensively and suggest that it is done in future work.

Both the VSLE and the Esaliency algorithms use probability estimation methods. One uses the Linear estimation method and the other builds a graphical model to represent a probability distribution of the joint assignments. Both methods use only second order statistics. Other methods using, for instance, higher order statistics may produce more accurate estimates. Then the tradeoff between accuracy and processing time should be investigated. Also, when combining the two approaches (extended global saliency and dynamic search) to one framework, it is possible that they share the probability estimation method. For instance, the graphical model used by Esaliency can be used not only for setting the initial priorities, but can be further used when the priorities are dynamically changed: Each recognition response will turn a node in the graph to an ‘evidence node’. This will change the probability distribution on the joint assignments and new labels probabilities can be inferred.

Interestingly, while originally we did not aim at modeling the HVS attention system, it turns out that the proposed models are very related to Duncan and Humphreys’s HVS attention model [28]. As such, our work can be considered as a quantification of their observations. We have performed initial tests for comparing our models performance with the performance of human observers (Chapter 6, Section 5.4.6) and got encouraging results, suggesting that there is place to extend this research direction.
Bibliography


[22] M. Carrasco, T.L. McLean, S.M. Katz, and K. S. Frieder. Feature asymmetries in visual search: Effects of display duration, target eccen-


Appendix A

The EsaliencyVSLE Application

A.1 General description

This appendix describes the EsaliencyVSLE application that was implemented as part of this work. A copy of the source code as well as a running application are available in the ISL (Intelligent systems Lab) in the Computer Science Department.

This application implements both the VSLE and the Esaliency algorithms, enabling to apply each of them separately or both algorithms together. The application may be applied on single images, on a directory of images, on video frames, or on frames from a live camera. The input images/frames may be gray levelled or colored. For segmentation, the application uses the implementation of Burt et. al.’s [19] algorithm in the openCV library. It also uses (optional) the Viola and Jones detection algorithm.

It has a graphical user interface (see figure A.1) that enables the user to choose the parameters for the segmentation process, to set initial priors, to
choose the features and weighting that will be used for the inner-similarity measures, to set parameters for the covariance/correlation vs. feature-space-distance function, to set some internal parameters for the Esaliency and the VSLE algorithms, to choose parameters for a detection process, and to choose the number of fixations to compute and to display. The default recommended parameters are chosen automatically when the application is invoked.

After the user sets his preferences and presses the GO button, the input image with the resulting fixations are displayed as demonstrated for instance in figure 5.9. The results are also saved on disk together with statistics on the last run’s performance.

A.2 Parameters Description

A.2.1 Algorithms to apply

The ‘Apply Esaliency’ and the ‘Apply VSLE’ check boxes enable the user to choose which of those algorithms are applied. It is possible to choose neither, one or both.

If neither are checked, the segmentation process is applied and the candidate-segments are scanned in raster order.

The Esaliency algorithm does not depend on recognizer feedback and therefore does not need the target mask files, or a recognition process to be involved. The target mask is only needed, in this case, for evaluating its performance (see A.2.9 for a description of the target mask file format).

The VSLE algorithm does need a recognizer’s feedback and it should get this information from a target mask file or by incorporating the Viola & Jones detection algorithm. If such information is not available, the VSLE assumes that all its fixations where on non-targets and its estimations will
Figure A.1: The VSLE and Esaliency application User Interface
rely on this (sometimes) mistaken conclusion.

A.2.2 Input Frames

Input From Camera

When this check box is set, the application grabs frames from a live camera. It is possible that not all frames are processed. The application grabs a frame after completing to process the previous grabbed frame.

Number of Camera Frames To Capture

If the user selects ‘Input From Camera’ and presses the ‘GO’ button, the application will apply the algorithms to ‘Number of Camera Frames To Capture’ frames and stop.

Image File/DirectoryName/Video File

This field can be a name of an image file, of a video file, or of a directory. It is ignored if the ‘Input From Camera’ check box is set. When the input is a directory name, the algorithms are applied to each file is the directory that starts with a ‘c’. (This enables to consider all files starting with a ‘m’ as target mask files. See A.2.9.)

Frames Jump

This field is relevant only when the input is a video file. When set to x, the algorithm is applied on the frames 1+ix, i=0,1,2,... When it is set to 1 all frames are processed.
A.2.3 Segmentation Parameters

The application uses the implementation of Burt et. al.’s [19] segmentation by openCV. The three first parameters are those required for this segmentation function. The last two further controls the candidate selection for attention.

Threshold 1, Threshold 2, Pyramid Level

The function cvPyrSegmentation OpenCV function implements image segmentation by pyramids. The pyramid builds up to the level ‘Pyramid Level’. The links between any pixel a on level i and its candidate father pixel b on the adjacent level are established if the Euclidean RGB color distance is below ‘Threshold 1’. (Note that for gray-level images, the distance is measured in the same way where all 3 color channels hold the same value.) After the connected components are defined, they are joined into several clusters. Any two segments A and B belong to the same cluster, if the Euclidean color distance between the average of the connected components is below ‘Threshold 2’. If Threshold2 is set to 0 the last stage is not performed. If it is not 0 and the clustering procedure is performed, we follow it with a connected-components procedure to get separate continuous segments. See more details in [19] and in the OpenCV manual.

Leave Segments of size between s1 and s2...

These two parameters enable to filter out segments that are irrelevant as candidates for attention as they are two small or two large. The candidates for attention will be those that the width and height of their bounding box is at least s1, and are no more than s2.
A.2.4 Initial Priors

By default a uniform prior is set to all candidates. However, it is possible to set different priors based on location, as demonstrated in Section 5.4.4.

Use Initial saliency map

Indicates whether the initial priors are defined by an initial map of locations.

Uniform $\mu$

Relevant only when ‘Use Initial saliency map’ is not set. Defines the uniform prior for all candidates. Should be low when E-saliency is applied to express the belief that there is a relatively small number of interesting objects in the scene.

Initial saliency map file

Relevant only when ‘Use Initial saliency map’ is set. The name of the file that specifies the initial priorities of each location. The file should be a gray level 8 bit image file. The initial priority of a candidate-segment is set to the maximal value of its corresponding pixels in this map divided to 255.

A.2.5 Features

These parameters set the description-features that will be used for the inner-similarity measures and the weight that is given to each type of features.

Color weight

The weight provided to the color features when computing the feature space distance between two candidate segments. If not set to zero, the average R,G
and B of the segment are used as descriptors.

**Shape weight**

The weight provided to the shape features when computing the feature space distance between two candidate segments. The shape features include the relative width and height of the segment’s bounding box, the bounding box’s width relative to its (height+width), and its segment’s area relative to its bounding box area.

**Movement weight**

This parameter indicates whether to compute the optic flow of the input image, and the weight that is given to the movement descriptors. When this parameter is not set to zero, the application requires the frame successive to the processed frame, in order to compute the optic flow. If the input is a single image named `.*`, the successive frame should be placed in the same directory, and named `.*_FollowingFrame.*`. When the input is a Directory, the successive frame of each image in that directory should start with ‘t’ (when the image it is related to is with the same name and extension, with ‘c’ before the name instead of ‘t’). For Video or Camera input, the successive frame is grabbed automatically.

The optic flow is computed based on block matching. (Each 16x16 block’s movement is estimated allowing a maximal movement of 12 pixels between successive frames. Then, the movement of each pixel is estimated using a bilinear interpolation.) After the optic flow provides the movement at each pixel, the average movement \( \vec{u} = (u_x, u_y) \) of each segment-candidate is computed and used for describing the candidates movement.
Gray-level histogram weight

When not set to zero, a gray-level histogram with 10 bins is added to the candidate feature vector. If the image is colored it is converted to gray scale for this computation. To accelerate the computation, a Gaussian pyramid is computed for the whole image. When computing the histogram of a candidate, the histogram is computed for the bounding box of the candidate-segment, in the pyramid level in which it is of size 24-47 pixels.

A.2.6 covariance function

Defines the covariance vs. feature-space distance function $\gamma$ used in the Esaliency and the VSLE algorithms. The user can choose whether to use an exponential descending function or a linear descending function. In case of a linear descending function the user should set the threshold $D$. After the application computes the (weighted) feature vectors, it computes the feature-space distance and then normalizes the distances so that the average distance between a pair of candidates will be 0.5. The distances are then clipped to the range $[0, 1]$. The exponential descending function implements the function $\text{var}_i \cdot \text{var}_j \cdot \exp(-d)$ where $\text{var}_i = \sqrt{\mu_i(1 - \mu_i)}$ and $d$ is the feature-space distance corresponding to candidates $i$ and $j$. The linear descending function descends linearly from $\text{var}_i \cdot \text{var}_j$ to 0 when $d$ increases from 0 to $D$. For $D < d < 1$, $\gamma(d) = 0$. (see figure 5.1b).

A.2.7 Esaliency parameters

N best

The number of assignments with highest likelihood to be computed by Nilsson’s algorithm. See details in Section 5.3.
Calculate Non-extended Global saliency

When this option is set, a non-extended version of the Esaliency algorithm is applied. (See Section 5.3.3.) Instead of preferring assignments with high likelihood that have a few target assignments, only assignments with one target are considered.

A.2.8 VSLE parameters

k nearest-neighbors to use for estimation

The number of known labels that are used for estimating unknown labels by the VSLE algorithm.

A.2.9 Detection parameters

Once a segment-candidate is selected for attention, the recognition can be simulated by using a pre-marked target mask file, or by applying the Viola & Jones detection algorithm.

When the input file starts with a ‘c’, the application looks for the target mask file starting with a ‘m’ with the same name. The extension of the target mask should be ‘pgm’ if the input image’s extension is ‘ppm’. Otherwise it should be with extension ‘bmp’. The ground truth file is a black and white image file. The mask of the target objects is colored in white while the rest of the image is in black.

Use Viola and Jones

If set, the Viola and Jones algorithm is used for recognition. Each time a segment-candidate is selected the detection algorithm is applied to the sub-image containing the bounding box of the segment. See details in Appendix
B.

xml file

The file resulting from the Viola & Jones learning process. Such files are provided with the openCV library for face detection, full body detection, lower body detection, upper body detection and profile face detection.

FOA (Focus Of Attention) radius

Relevant when the target mask files are used for deciding whether a fixation is on a target or on a non-target.

When this parameter is set to -1, the selected segment-candidate is considered as associated with a target object if it intersects with a marked target in the targets mask file. If the parameter is set to a positive integer, r, the selected segment-candidate is considered as associated with a target object if the circle with radius r that its center is the center of the segment intersects with a marked target in the targets mask file. (This option was used for comparing performance with the iLab [1] application.)

A.2.10 Stop Criteria, Display and Save parameters

Number of fixations to evaluate

How many fixations to evaluate. If set to -1 all candidates are visited. This can be used for getting statistics on the algorithms’ performance

Number of fixations to show

How many fixations to display. If set to -1, all fixations according to ‘Number of fixations to evaluate’ will be displayed.
Set ‘Number of fixations to evaluate’ to -1 and a small ‘Number of fixations to show’ to get statistics while getting results displayed nicely. Use a small ‘Number of fixations to evaluate’ to get a faster run when applying on video and camera input.

**Save Results**

If checked, the results are saved in files as described in the next section.

### A.3 Results Description

As mentioned above, after the algorithm is applied on an image/video frame the image/frame with the resulting fixation path are displayed on the computer screen. Each fixation is marked by a rectangle that is the bounding box of the candidate-segment. When a target-mask file is available or if the Viola & Jones algorithm was used for detection, the fixations that were recognized as targets are marked in yellow, while other fixations are marked in red. Otherwise, all fixations are marked in red.

When the option to save the results is chosen, two new image files are created. One displays the results of the segmentation, and the second displays the fixation path on the displayed image. When the input is a directory of images, a video file, or when the input is a video stream from a camera, the above files are saved for each frame.

The statistics about the algorithms performance, such as the number of fixations required to locate the targets are reported in a text file named ‘report.txt’.
Appendix B

Combining VSLE and Viola&Jones faces detection: Implementation details

This appendix discusses, in detail, the experiments described in Sections 4.8.2 and 4.8.3.

The segmentation process is an implementation of [19] in the openCV library. We use the function cvPyrSegmentation with parameters levels = 4, threshold1 = 255, threshold2 = 17.

To create the set of candidates, each segment is first bounded by a rectangle denoted $r$. Let $s_1(s_2)$ denote the short (long) side of $r$. Rectangles for which $s_1 < 12$ (too small) or for which $\frac{s_2}{s_1} > 3$ (too narrow) are ignored. Then, $r$ is dilated in each direction by $\frac{1}{2}s_2$. The dilated candidate is denoted $R$. For each candidate we set a minimal ($s_1 \times s_1$) and maximal ($1.5s_2 \times 1.5s_2$) scale for faces to be detected inside this rectangle, depending on its size.

VSLE measures similarity between the candidates simply by comparing gray level histograms with 10 bins using $L_2$ norm. To get the effective
histograms of the candidates, we first compute a Gaussian pyramid of the original image. Then, for each candidate, we compute its histogram from the pixels in the pyramid level in which the candidate’s side $s_1$ is between 24 and 47.

The linear estimates are performed using only the 3-nearest-neighbors version of VSLE (Section 4.6). That is, in each iteration each unlabelled candidate is estimated based on the three labeled candidates that are most similar to it. VSLE starts (arbitrarily) from the first candidate (which corresponds to the image’s most upper-left segment).

In Section 4.8.3 we used the extended implementation of VJA [48] provided by the OpenCV library. We did not train the classifier ourself, but used the default training results supplied by the OpenCV library (haarcascade_frontalface_default.xml). When applying the OpenCV detection function (cvHaarDetectObjects) on an image, it tries to detect faces of size $24 \times 24$ to faces of the image’s size minus 10 pixels, jumping in scale factors of 1.1. In each scale the detection window is shifted by $s\Delta$ where $\Delta = 2$ or 4, depending on previous windows’ results, and $s$ is the scale factor.

When VSLE is incorporated with VJA, the same candidates and similarity measures described above are used. Each time VSLE decides to apply VJA on a candidate, it is applied on the sub-image corresponding to the candidate. The detection function was extended to be applied from the given minimal and maximal scales for detection, which are limited to fit the candidate’s size as described above (from $s_1 \times s_1$ to $1.5s_2 \times 1.5s_2$). If VJA recognized at least one target inside this sub-image, VSLE treats the answer as ‘yes’.

Before reporting detections and false detections at a given time, the following processing takes place: all the squares recognized as faces by the
classification cascade are collected. Overlapping groups of squares turn into one (average) square. Individual squares that were not recognized as part of a group are thrown out (to lower the false detection rate). This processing is part of the cvHaarDetectObjects function. When we use the VSLE+VJA version, we do not perform this processing on each candidate, but only on the whole image before reporting detection and false alarm rates.
The measure was constructed by combining the information from the images with the expectation of a small number of salient clusters. The ESaliency algorithm takes a global approach by considering a region as salient in contrast to other models. Those similar to this region in the whole image (or not at all) are only considered if there are few attended locations. The saliency algorithm is applied at a local level (graphic model) to evaluate the locations in the image that can be used by the effective algorithms on natural images. High locations are connected with interesting objects, and the effectiveness and demonstrated results were compared with other algorithms intensively. The integration of this algorithm in the identification process also examined the relevance, which was determined in studies conducted with FLNN (internal noise) and other biological systems that can predict the visual performance of human observers in search tasks based on color orientation. The similarities of the models in our studies were closest. Different in the different conditions, the reasons for the difference in the effective performance of the algorithms compared to the conventional models in visual search and eye movements can be used to improve the performance of the algorithms. The presented results encourage us to try and develop new ideas and methods that can be used to test other methods. The same ideas can be used to test other methods.
The frameworks that we shall elaborate on are the approaches of Duncan [27] and Humphreys [95].

Duncan and Humphreys advocate that similarity and proximity should be used as a basis for matching features. Their approach is based on the idea of matching features between different parts of an image and showing that the number of matched features can be a very useful measure in the context of visual saliency.

Robust feature-based models like [10] that are used for directing interest are useful in the context of visual saliency. These models do not require the matching of biological relevant features. Instead, they focus on matching between different parts of the image and show that the number of matched features can be a very useful measure in the context of visual saliency.

Aiming at visual search, we propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms. We propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms.

We propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms.

We propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms.

We propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms.

We propose two main goals: achieving high performance in search algorithms and developing a method that can be used to evaluate the performance of visual search algorithms.

II
תקציר

The analysis of an image is often a complex process involving attention mechanisms and various attention mechanisms. These mechanisms are crucial in searching for familiar objects in different locations.

In the visual attention mechanism (Yarbus [100]), the eyes tend to focus on areas that are considered important, while other areas are given less attention. This is supported by physiological and psychological studies that indicate that the human visual system is not divided in a clear-cut manner for processing.

For example, Yarbus's experiments showed that the eyes tend to focus on elements that contain critical information, while other areas receive less attention. This is evident from the spot light metaphor (Posner [66]) and zoom lens metaphor (Eriksen & James [35]), which suggest that the visual field is divided between overt attention, which involves the eyes, and covert attention, which is not related to eye movements.

The human visual attention system is not limited to eye movements. It is also influenced by other factors such as task complexity and the nature of the stimuli. Feature integration theory (Triesman & Gelade), which was proposed based on Nessier's work [55], suggests that the search process is divided into two stages: pre-attentive and top-down. The pre-attentive stage involves the extraction of basic features of each image element, while the top-down stage involves the selection of objects with similar properties to the desired goals.

The similarity theory (Humphreys & Duncan), which was proposed by Wolfe [99], Itti and his colleagues [45], and Koch [46], suggests that the selection of objects is based on the saliency map, which indicates the uniqueness of each point on the image in relation to its surroundings. This map reflects the nature of the objects and their relationship to the desired goals.

As a result, the task of searching for objects in an image is not a straightforward process. It involves the interaction of various attention mechanisms and the selection of objects based on their properties and their relationship to the desired goals. The use of computer vision techniques can help in this process by supporting the selection of objects and their relationship to the desired goals.
המחקר נועש בהנחיית פרופ' מיכאל לינדנבוואם בפקולטה لمמדיهج המחשב.

חלק המחקר נועש בשיתוף עם החוג לפסיכולוגיה ולפסיכולוגיה שלו ד"ר ימי ישורון מהאוניברסיטה העברית בירושלים.

אני מודה לטכניון על התמיכה הכספית הנדיבת בהשתלמותי.

אני מודה לטכניון על התמיכה הכספית הנדיבת בהשתלמותי.
תהליך קשב ויזואלי
הمبוססים על מודלים הסתברותיים:
אלגוריתמים והסמים

תמר אברהם

מכון טכנולוגי לישראל-ש לסנט הטכניון הוג
מאי 2007

הוגש לפנמי התכנית - מכון טכנולוגי לישראל
תיונה
מאי 2007
החלקית קשב ויזואלי

המשויכים על מודלי הסתברותיים: אלגוריתמים והם

תמר אברים