TÉCNICO LISBOA

Active Perception: Scene Exploration using Foveal Vision

Luís Doutor Simões

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Alexandre José Malheiro Bernardino

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Alexandre José Malheiro Bernardino
Member of the Committee: Prof. Vicente Javier Traver Roig

September 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

Active perception and foveal vision are the foundations of our visual system. While foveal vision reduces the amount of information to process at any time instance, active perception will direct the eyes to promising parts of the visual field. Together, they allow a detailed perception of the objects on the environment with limited neuronal processing resources. We develop a method that combines both concepts to explore and identify all the objects on an image with the least number of gaze shifts. A foveal sensor will scan the image sequentially and create a semantic map of the scene, choosing at each step the location with higher information gain, regarding the identification of the objects. Our framework uses the foveated images as input to a state-of-the-art object detector, whose scores are modelled by a Dirichlet distribution that depends on the distance to the fovea, denoted Foveal Observation Model. After each new saccade, this Model is used to perform a Sequential Fusion of the detection scores in a global map. With the updated distributions at each map point, a decision based on information theoretic measures is made to find the next-best-viewpoint that maximizes our knowledge of the world. Despite the blur, we show that it is possible to combine foveated images with state-of-the-art object detectors using our proposed models. Furthermore, our models not only improve the identification of objects by 2-3%, but also reduce 3x (in average) the number of required gaze shifts to achieve similar performances against randomly choosing the next viewpoint.

Keywords

Active Perception; Foveal Vision; Object Detection; Active Object Search; Fusion of Classifiers.

Resumo

A perceção ativa e visão foveal são as bases do nosso sistema de visão. Enquanto a visão foveal reduz a quantidade de informação a processa, a perceção ativa irá direcionar os olhos para partes promissoras do campo de visão. Juntos, permitem uma perceção detalhada dos objetos com reduzida complexidade a nível neuronal. Desenvolvemos um método que combina ambos os conceitos para explorar e identificar todos os objetos numa imagem com o menor número de mudanças focais. Um sensor foveal percorre a imagem sequencialmente enquanto cria um mapa semântico, escolhendo em cada iteração o local com maior ganho de informação, no que diz respeito à identificação dos objetos. O nosso trabalho utiliza as imagens foveadas como entrada de um detetor de objetos estado-da-arte, cujas pontuações são modeladas por uma distribuição de Dirichlet que depende da distância para a fóvea, denotado Modelo de Observação Foveal. Após cada nova sacada, este Modelo é usado para executar uma Fusão Sequencial das pontuações de deteção num mapa global. Com as distribuições atualizadas em cada ponto de mapa, é tomada uma decisão baseada em medidas teoréticas de informação para encontrar o próximo melhor ponto que maximiza o nosso conhecimento do mundo. Apesar da "névoa" nas periferias, mostramos que é possível combinar imagens foveadas com detetores de objetos estado-da-arte usando os nossos modelos propostos. Além disso, não só melhoram a identificação de objetos em 2-3%, como também reduzem 3x (em média) o número de sacadas necessárias para obter desempenhos semelhantes à escolha aleatória do próximo ponto focal.

Palavras Chave

Perceção Ativa; Visão Foveal; Deteção de Objetos; Procura de Objetos Ativa; Fusão de Classificadores.

Contents

1 Introduction **1**

1.1 Motivation . . . 3

1.2 Problem Definition . . . 5

1.3 Organization of the Document . . . 8

2 Background & Related Work **9**

2.1 Foveal Vision & Object Detection . . . 11

2.1.1 Foveal Vision . . . 11

2.1.1.A Computational Retina Models . . . 11

2.1.1.B Classification and Detection on Foveated Images . . . 14

2.1.2 Detection Algorithms . . . 16

2.2 Probability Distributions . . . 18

2.2.1 The Categorical Distribution . . . 19

2.2.2 The Dirichlet Distribution . . . 20

2.2.2.A Estimating a Dirichlet Distribution . . . 22

2.3 Approaches on Fusing Classifiers . . . 25

2.3.1 Bayesian filtering - Naïve Bayes approach . . . 25

2.3.2 Sum Rule Fusion Approach . . . 27

2.3.3 Kaplan's Approach on Fusing Classifiers . . . 27

2.4 Active Perception . . . 29

2.4.1 Acquisition Functions . . . 30

2.4.2 Active Perception on Cartesian Domain . . . 31

2.4.3 Integration with Foveated images . . . 32

3 Approach **33**

3.1 Object Detection & Foveal Observation Model . . . 35

3.2 Fusion Model . . . 38

3.2.1 Background Class . . . 40

3.2.2 Extension to Integrate the Observation Model . . . 41

3.2.3 Update Normalization . . . 42
3.3 Active Perception - Gaze Selection . . . 44
4 Experiments & Results **49**
4.1 Foveal Observation Model Validity . . . 51
4.2 Map Update . . . 52
4.2.1 Overall Performance . . . 53
4.2.2 Fusion Example . . . 56
4.3 Next Best View Point . . . 56
4.3.1 Comparison of Update Normalization Methods . . . 58
4.3.2 Comparison of Acquisition Functions . . . 60
4.3.3 Active Gaze Selection Performance . . . 61
5 Conclusion **65**
5.1 Conclusions . . . 67
5.2 System Limitations and Future Work . . . 68
A List of Symbols **75**

List of Figures

1.1 Example of an image simulating human vision, with higher resolution on the center (fovea) and increasing blur over the peripheries. . . . 3

1.2 **Left:** Anatomy of the eye with a retina close-up, with focus on the non-uniform distribution of Cones and Rods. **Right:** Distribution of Cones and Rods in the Human retina [1]. . . . 4

1.3 Scene/image representation where the squares represent the objects O and the center of the dashed circles represents the focal point (x_t, y_t). The coordinates x, y correspond to the global coordinates of the image, while u_t, v_t represent the local (retinal) coordinates with reference to the center of the fovea (x_t, y_t). . . . 5

2.1 A summary of the steps in the Artificial Foveal Visual system proposed by Melicio [2], later updated by Figueiredo [3], in this case with four levels. The image G_0 corresponds to the original image and F_0 to the foveated one. . . . 12

2.2 Image obtained with the foveation system proposed by [2] for different sizes of the simulated fovea. . . . 13

a $f_0 = 30$. . . . 13

b $f_0 = 60$. . . . 13

c $f_0 = 90$. . . . 13

2.3 **Left:** Gaussian Receptive Fields on top of a retina tessellation. **Right:** The 4196 node tessellation used in [4] (where this figure was extracted from). . . . 14

2.4 **Left:** Cortex image. **Center:** Retina back-projected **Right:** Input image. (Figure extracted from [5]). . . . 14

2.5 Colour coded receptive field centres mapped onto the log-polar (left) and linear polar (right) spaces. Warmer colours indicate receptive fields closer to the peripheries,whereas colder colours indicate points closer to the fovea. [4] . . . 15

2.6 **Left:** Image with proposed bounding boxes from the You Only Look Once (YOLO) algorithm, with a ”toy” grid put on top of the image. **Right:** Example of pre-defined anchor boxes. . . . 18

2.7 Model architecture of the YOLOv3 implementation. 19
2.8 Beta distribution examples, for different α and β parameters 20
2.10 Hidden Markov Model (HMM) of the whole fusion process 26
2.11 Bayesian Network representing the joint Probability Density Function (pdf) of the measurement, where T represents the total number of time instants considered. 28

3.1 Diagram for this work. First we extract an image and convert it to a foveated image, then an Object Detection method is used to compute confidence scores and region proposals. The Foveal Observation Model models the uncertainty of the confidence scores and region proposals with respect to the location on the foveated image to then perform a Map Update where previous information is already stored. Using the information stored on the map, Gaze Selection acquisition functions choose the next best view point, the agent shifts its gaze to that point (Gaze Control) and then a new foveated image is produced, reiterating the process. 35
3.2 Object detection and Foveal Observation Model diagram. Dependencies between variables are represented by the arrows. 36
3.3 Representation of the map by a 10x10 grid of map cells on top of a foveated image. . . . 39
3.4 Intersection Over Union (IoU) formula with explanatory images. Adapted from [6] 43

4.1 Performance comparison between the scores outputted by the foveal observation model (blue) and the scores outputted directly from the object detection algorithm (without being modeled by the foveal observation model) (red). 51
a Accuracy comparison. 51
b Entropy comparison. 51
4.2 Scores comparison for the ground-truth class outputted by the foveal observation model and the object detection algorithm, as a function of the scores outputted directly from the object detector 53
4.3 Performance metrics of 4 different fusion algorithms, analysed time-wise as new bounding boxes are detected. 54
a Average Accuracy evolution. 54
b Evolution of the Average Expected Value for the ground-truth class. 54
c Average Kullback-Leibler (KL) divergence evolution. 54
4.4 Two examples of the evolution of the expected value, for two different images, upon fusing the resulting bounding boxes of each images when foveated in different points, using the 4 algorithms mentioned on this section. 55

4.5 Performance evolution of the Kaplan algorithm throughout 10 iterations. Each line represents **Left:** the same image with a different focal point and **Right:** an heat-map with the expected value of the ground-truth object(s) given by the Kaplan method at a certain iteration. 58
4.6 Performance metrics of 4 different fusion algorithms, analysed time-wise as new bounding boxes are detected. 59
a F1-Score comparison using the absolute gain of the difference between two peaks as the acquisition function. 59
b F1-Score comparison using the KL divergence gain as the acquisition function. . . 59
c F1-Score comparison using the classification entropy loss as the acquisition function. 59
4.7 Comparison between the F1-Score of the algorithm, using the three different acquisition functions combined with the best bounding box selection method for each one. 61
4.8 **Left:** F1-Score of the algorithm using the acquisition function ”KL Divergence Gain” in red, against the F1-Score of all fusion methods mentioned in section 3.2 when choosing the focal point randomly. **Right:** KL Divergence evolution with and without using Active Perception (acquisition function ”KL Divergence Gain”) to choose the next best view point. 62

List of Tables

2.1 Detection Systems on Pascal VOC2007. Comparing the performance and speed of state-of-the-art object detection methods. All timing information is on a Geforce GTX Titan X [7]. 17

A.1 List of symbols used on this document 75

Acronyms

AVS	Active Visual Search
CNN	Convolution Neural Network
DCNN	Deep Convolution Neural Network
DPM	Deformable Part Model
FPS	Frames per Second
HMM	Hidden Markov Model
IoU	Intersection Over Union
KL	Kullback-Leibler
mAP	Mean Average Precision
MLE	Maximum Likelihood Estimation
NMS	Non-Maximum Supression
pdf	Probability Density Function
RPN	Region Proposal Network
SS	Selective Search
SES	Sensory Ego-Sphere
SSD	Single Shot MultiBox Detector
YOLO	You Only Look Once

1

Introduction

Contents

1.1 Motivation 3

1.2 Problem Definition 5

1.3 Organization of the Document 8

Active Perception represents a broad spectrum of concepts, Bajcsy [8] defined it as:

> *An agent is an active perceiver if it knows why it wishes to sense, and then chooses what to perceive, and determines how, when and where to achieve that perception.*

Throughout the years, the concept has been considered in numerous studies and researches as a requirement for many artificially intelligent agents, improving the real-time performance of vision-based tasks. For a detailed review on the history of the computational perspective on the problem of active perception, the interested reader should refer to Bajcsy, Aloimonos & Tsotsos [8].

This work focuses on the optimization of scene understanding processes using the Active Perception concept, combining it with a biologically inspired foveal vision sensor. Foveal vision mimics the distribution of photoreceptors in the human eye, which allows a good perception of the objects centered in the image (high resolution) while allowing more efficient implementations and computational savings.

1.1 Motivation

Central vision (or foveal vision) is an indispensable feature of the human eye allowing to perform activities which require high-resolution visual details, in contrast with peripheral vision where the resolution is much lower (blurred image) (see fig. 1.1).

Figure 1.1: Example of an image simulating human vision, with higher resolution on the center (fovea) and increasing blur over the peripheries.

So, why are our eyes divided in these two regions? It would be reasonable to think that having a wider central vision could greatly improve our survival. But in fact, human eyes are built the other way around. The fovea comprises less than 1% of retinal size, but takes up over 50% of the cortex [9], thus one can imagine that our brain would have to be impractically large to handle the full visual field at high

resolution. This difference is explained by the non uniform concentration of nerve cells connecting the retina to the brain.

The distribution of cones and rods (as one can see in fig. 1.2 left) is what differentiates the fovea, where there is a high concentration of cones, from the remaining part of the retina, which is mainly composed of rods (fig. 1.2 right). Cones are able to perceive finer detail (high acuity) and colors, while rods have low acuity and are achromatic, although being more sensitive to external changes [10].

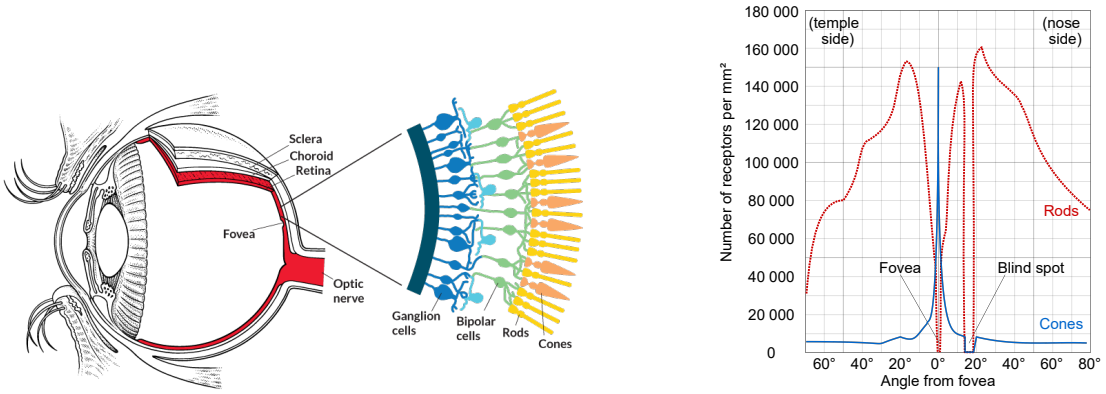


Figure 1.2: Left: Anatomy of the eye with a retina close-up, with focus on the non-uniform distribution of Cones and Rods. **Right:** Distribution of Cones and Rods in the Human retina [1].

However, since the amount of information is greatly reduced by the foveation mechanism, one could think that, to analyse a scene, it would just require a gaze shift to every location, and extract the information obtained by the fovea. Still, scanning the entire scene would require an unbearable amount of time. Nevertheless, the fovea does not need to cover the entire scene, the peripheral vision also extracts some useful information to guide the eyes to visit unexplored places where there is a high probability of existing objects, given all the acquired information.

Just like human vision, many computer vision applications are constrained by the involved computational effort, specially when implemented on artificial intelligent agents whose tasks depend on the analysis, in real-time, of their surroundings. Hence, urges the need to develop models capable of filtering and fusing information, ignoring what is not relevant for the task in hands. This is where the Active Perception models, combined with foveal vision, come to the picture. As defined before, active perception selectively chooses new targets for the acquisition of information based on the knowledge that the agent has about the current state of the world and what is promising or not to complete a certain task.

Although there have been a large amount of research and developments on attention and visual search models (as in [11], [12], [13] and [2]), there is still a long way to go, specially regarding the modeling of the mechanisms that help the decision of where to shift the gaze to. Besides, there is some work done on image processing using foveated images, but, at the extent of our knowledge, there are no attempts on combining state-of-the-art object detection mechanisms and active perception methods

to perform a scene exploration task using foveal vision.

1.2 Problem Definition

The main goal of this project is to implement a model to optimize the exploration of a scene, gathering as much information as possible about all the objects, in the least amount of gaze shifts, using foveal vision.

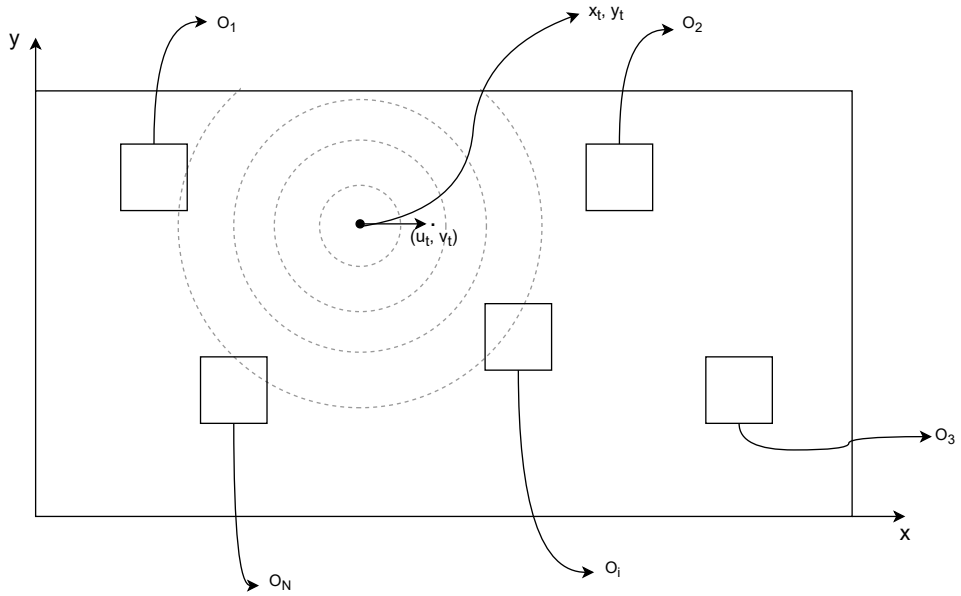


Figure 1.3: Scene/image representation where the squares represent the objects O and the center of the dashed circles represents the focal point (x_t, y_t). The coordinates x, y correspond to the global coordinates of the image, while u_t, v_t represent the local (retinal) coordinates with reference to the center of the fovea (x_t, y_t).

So, let's start by considering the set of N objects $\mathcal{O}$ represented on the scene (a complete list of symbols is provided on appendix A),

$$\mathcal{O} = \{m\}, \quad m = 1, ..., N \tag{1.1}$$

where

$$O_m \in \mathcal{C}, \quad \forall_m \tag{1.2}$$

being $\mathcal{C}$ the set of possible classes of objects

$$\mathcal{C} = \{c_o, c_1, ..., c_K\} \tag{1.3}$$

where K is the number of classes, and c_0 is the label of the background class.

The set of detected objects $\mathcal{I}_t$ that are actually seen by a detector algorithm, at instant t, is given by

$$\mathcal{I}_t = \{I_{t,l}\}, \quad l = 1, ..., L_t \tag{1.4}$$

where

$$I_{t,l} = (\boldsymbol{B}_{t,l}, \boldsymbol{S}_{t,l}) \tag{1.5}$$

is the l-th detection at instant t, L_t is the number of detected objects at instant t, which may differ from the actual number of objects represented on the scene N, and the pair ($\boldsymbol{B}_{t,l}$, $\boldsymbol{S}_{t,l}$) are the outputs of a single detection $I_{t,l}$.

The detector algorithm outputs a bounding box $\boldsymbol{B}_{t,l}$, which is an array containing the location and size of the object and an array of confidence scores $\boldsymbol{S}_{t,l}$. The position of the bounding box $\boldsymbol{B}_{t,l}$ can be given by the local coordinates $(u_{t,l}, v_{t,l})$ representing the relative position of the center of the bounding box to the focal point (x_t, y_t). On the other hand, the confidence scores $\boldsymbol{S}_{t,l}$ contain the probability of a given detection $I_{t,l}$ belonging to each of the K classes of objects $\mathcal{C}$ for which the detector was trained to detect:

$$\boldsymbol{S}_{t,l} = [s_{t,l,1}, s_{t,l,2}, ..., s_{t,l,K}]^T, \quad 0 \leq s_{t,l,j} \leq 1 \tag{1.6}$$

The confidence scores $\boldsymbol{S}_{t,l}$ are in the probability simplex after normalizing the probabilities to sum to one, $\sum_{k=1}^{K} s_{t,l,k} = 1$ and $s_{t,l,k} \geq 0$ for all $k \in \{1, ..., K\}$.

After having the output of the detections on the resulting foveated images of each saccade, we first need to build an Observation Model that models how the detections and their confidence scores vary depending on their relative position to the fovea. The Observation Model is then defined for each detection $I_{t,l}$ as the distribution of its confidence scores $\boldsymbol{S}_{t,l}$, given the distance to the fovea $d_{t,l} = \|(u_{t,l}, v_{t,l})\|$, for each possible object class label c_k, :

$$p(\boldsymbol{S}_{t,l}|c_k, d_{t,l}) \tag{1.7}$$

Secondly, a world map has to accumulate over time the knowledge that the observations provide, for our application we can consider a body centered 2D map of the surroundings. At each saccade, the map information has to be updated with the new observations, and, therefore, a Fusion Model is required to solve this part of the problem.

$$\boldsymbol{M}_t(x, y) = P(C_{x,y}|I_{0:t,f_{t,l}(x,y)}, x_{0:t}, y_{0:t}) \tag{1.8}$$

where $\boldsymbol{M}_t$ can be seen as the map information (state) at iteration t, containing at each pixel (x, y) a vector of parameters that encode the probability distribution of the fusion of all observations that overlap the pixel (x, y), which are given by the function $f_{t,l}(x, y)$, where (x, y) are the global coordinates of the

image (the referential can be seen in fig. 1.3), $(x_{0:t}, y_{0:t})$ are the location of the focal point at each instant of time, and $C_{x,y}$ is the class label that we want to estimate, where $C_{x,y} \in \mathcal{C}$.

Having now the updated map information, in order to make a full exploration in the least number of gazes, the focal points can not be randomly chosen, this is where Active Perception comes to the picture. An Active Perception method that consists in choosing the point to look next that maximizes the gain of information about the scene has to be implemented

$$x^*, y^* = \underset{x,y}{\operatorname{argmax}} \, F(x, y, \boldsymbol{M}) \tag{1.9}$$

where $F(x, y, \boldsymbol{M})$ corresponds to the gain of information or the loss of confusion (for example, maximizing the Kullback-Leibler (KL) Divergence or minimizing the classification entropy of the map distributions, or maximizing the difference between the two most probable classes) on moving the focal point to the coordinates (x, y) knowing the current state of the map $\boldsymbol{M}$. (x^*, y^*) are the coordinates of the best point where to look next.

Summarizing, the problem of providing an efficient scene exploration using foveal vision, will be tackled by the integration of the following components/contributions:

1. the development of an Observation Model, that combines a state-of-the-art object detector with foveal vision (eq. (1.7)) (object detectors were built and trained for Cartesian images).

    - The observation model depends on the particular detector used so we will learn it from a dataset. This requires building a data set composed of detections on foveated images to train it.

2. the development of a Fusion Model (eq. (1.8)) that can use the Observation Model (eq. (1.7)) to keep the world knowledge updated. Here we adopt a sequential Bayesian approach to allow the fusion of the measurements in a online fashion.

3. the development of Active Perception methods (eq. (1.9)) that optimize a scene exploration, minimizing the number of required gaze shifts and exploration time.

Therefore, besides studying state-of-the-art methods on active learning, object detection, and foveal vision, research on probability distributions parameter estimation methods and probability fusion algorithms will also be of major importance throughout this project, for modeling each of the components listed above.

1.3 Organization of the Document

The remainder of this thesis is organized as follows: in chapter 2, related work and state-of-the-art approaches on foveal vision, object detection, fusion methods and active perception are reviewed. In chapter 3 the proposed approach is described where each component of the framework is explained in detail. In chapter 4 we explain the experiments and the respective results, where the validity of the implemented observation model is evaluated, as well as the performance of the proposed active perception methods. Finally, Chapter 5 is where we present our conclusions, and some discussion on possible future contributions.

2

Background & Related Work

Contents

2.1 Foveal Vision & Object Detection 11
2.2 Probability Distributions 18
2.3 Approaches on Fusing Classifiers 25
2.4 Active Perception 29

The research for this work was divided into three main topics. Each of these topics corresponds to the models required to solve the problems defined in section 1.2. First, the Observation Model, where foveal vision, its implications upon detecting objects, and object detection algorithms are analysed, followed by some notes on the probability distributions used on this work. Then, the Fusion Model, where approaches on fusing classifiers (observations) are discussed. And, finally, the Active Perception Model, where related works on this topic are reviewed, with special focus on recent approaches that use Active Perception methods on foveated images.

2.1 Foveal Vision & Object Detection

Nature provides an immense number of different stimulus, which humans detect using their five senses (sight, sound, smell, taste and touch). Colativa [14] led an interesting study, where human subjects were presented with both visual and auditory stimulus. The subjects' responses seemed to indicate a domination by the visual stimulus over the auditory ones, showing the importance of humans' visual system.

So what is vision? It is commonly known as the perception of objects features (like color, size and form) through the light that enters the eye. More precisely, light rays are received and converted from visual stimuli into electrical signals on the retina, which are then transmitted to the visual cortex in the brain. However, the retina is not uniform neither is the concentration of nerve cells connecting the retina to the brain.

2.1.1 Foveal Vision

For many robotic applications, having the central part of an image with much higher resolution can be useful, even if it means that the remaining part of the image needs to have considerable low resolution, just like the images our eyes produce.

Consequently, the number of researches inspired on the human/mammalian vision system is increasing. Foveal vision will be studied throughout this work, meaning that a model to recreate the retina will be needed.

2.1.1.A Computational Retina Models

Two of the works which inspired this project were performed by Almeida [13] and Melicio [2] where they combine visual attention and foveal vision to detect and locate objects. Their work will be cited in different areas of this project, but for now we will focus on how they transform Cartesian images into foveated ones.

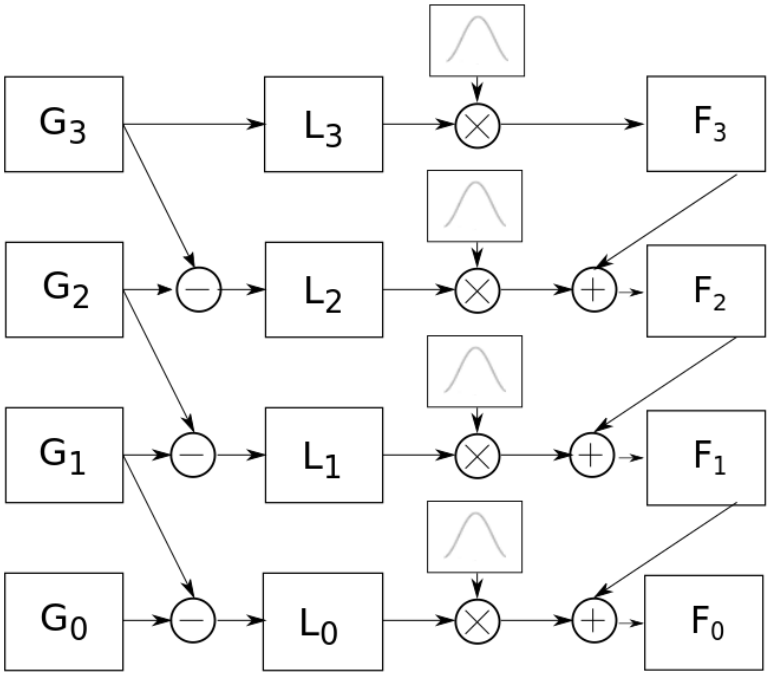


Figure 2.1: A summary of the steps in the Artificial Foveal Visual system proposed by Melicio [2], later updated by Figueiredo [3], in this case with four levels. The image G_0 corresponds to the original image and F_0 to the foveated one.

Three steps are comprised in their approach (can be seen graphically in fig. 2.1): The first step is to build a Gaussian scale-space where each level corresponds to a low-passed version of the previous level. Each level has an increasing level of blur, but similar resolution. The first level contains the original image G_0 which serves as input to a Gaussian (low-pass) filter g_1, resulting in the image G_1 at level 1.

The image G_k can, equivalently, be obtained from the image G_0, via Gaussian filter kernels of the form

$$g_k(u,v) = \frac{1}{2\pi\sigma_k^2} e^{-\frac{u^2+v^2}{2\sigma_k^2}}, \quad 0 \leq k \leq K \tag{2.1}$$

where u and v are the image coordinates, K is the total number of levels, and $\sigma_k = 2^{k-1}\sigma_1$ is the Gaussian standard deviation at the k-th level, for $k \geq 1$, being σ_0 a small value so that G_0 is almost identical to the original image (in fact, the first filtering operation may be skipped because it does not change the input image but is convenient to consider for the sake of the theoretical analysis). The Fourier transform of the Gaussian filter kernels is given by

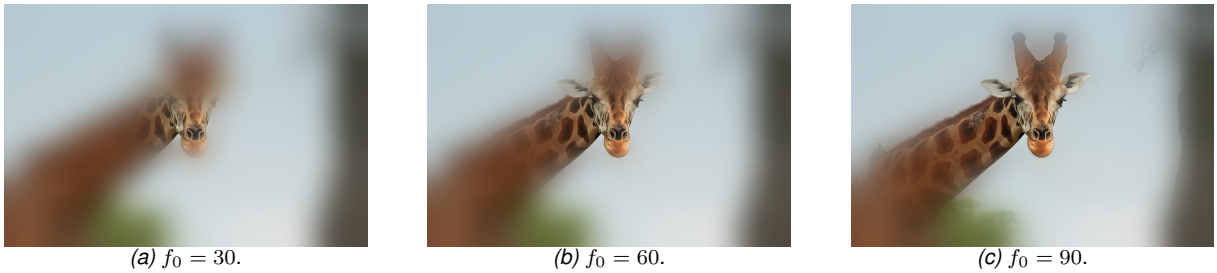
$$\tilde{g}_k(e^{jw_u}, e^{jw_v}) = e^{-\frac{\sigma_k^2}{2}(w_u^2+w_v^2)}, \quad 0 \leq k \leq K \tag{2.2}$$

where w_u and w_v are the horizontal and vertical spatial frequencies, respectively. Also, note that $\tilde{g_0} \approx 1, \forall w_u, w_v$.

Then, a Laplacian scale-space is built where the difference between adjacent Gaussian levels is computed (see fig. 2.1), resulting in a set of error images. Finally, each level is multiplied by exponential kernels to emulate a smooth fovea. The exponential kernels are of the form

$$H_k(u,v) = e^{-\frac{(u-u_0)^2+(v-v_0)^2}{2f_k^2}}, \quad 0 \leq k \leq K \tag{2.3}$$

where (u_0, v_0) corresponds to the foveation point (center of the fovea), f_0 to the size of the kernel in the level 0 of the scale-space, and $f_k = 2^k f_0$ denotes the standard deviation of the exponential kernel at the k-th level. An example of the resulting foveated image, for different sizes of the fovea, is represented in fig. 2.2.



(a) $f_0 = 30$. *(b)* $f_0 = 60$. *(c)* $f_0 = 90$.

Figure 2.2: Image obtained with the foveation system proposed by [2] for different sizes of the simulated fovea.

This approach creates an image which has a higher resolution around the foveation point, decreasing gradually over the periphery. This is equivalent to applying a non-uniform blurring filter over the original image instead of changing the pixel size and distribution along the foveated image. Consequently, although simulating foveal vision, this approach does not reduce the image size, meaning that it does not take advantage of the decrease of resolution over the periphery to reduce computational costs. Anyway, it is a simple and convenient process to analyse the consequences of foveal images in artificial vision and machine learning methods.

In order to take full advantage of the possible memory reduction when using foveal vision, a different approach was recently proposed by Ozimek [4] and Siebert [5]. They use a self-similar neural network to define retina sampling locations as described by Clippingdale & Wilson [15]. As a result, a network of N nodes jointly undergoing random translations produce a tessellation with a near-uniform dense foveal region that progressively transitions into a sparse periphery (Figure 2.3). Each of the nodes in the described tessellation corresponds to the location of a receptive field's center and they all have a Gaussian response profile where the standard deviation scales linearly as a function of the local node density (these parameters were manually chosen while checking whether the sub-sampling was sufficiently sharp and free from aliasing artefacts).

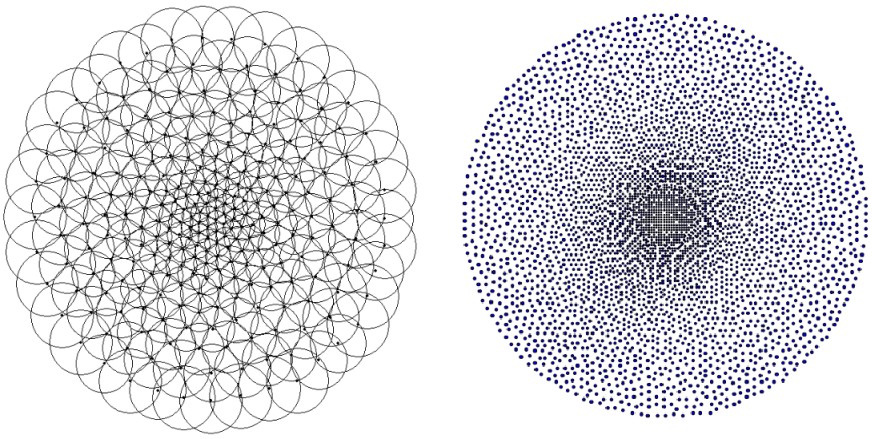


Figure 2.3: Left: Gaussian Receptive Fields on top of a retina tessellation. **Right:** The 4196 node tessellation used in [4] (where this figure was extracted from).

2.1.1.B Classification and Detection on Foveated Images

Classification and Detection algorithms usually assume uniformly distributed pixels locations on a rectangular shape matrix, thus are not prepared to perform detection on non-uniformly sampled images. One approach would be to invert the foveation process, in order to recreate the shape of a Cartesian image.

Almeida's work [13] used the method presented in fig. 2.1 that preserves the original image resolution. The approach is interesting since the resulting image is ready to be processed by a Convolution Neural Network (CNN) (see fig. 2.2); One just has to resize the image to fit the input requirements of the network.

On the other hand, Siebert [5] stored the values sampled by his non-uniform resolution retina on a one dimensional array of intensity values which is then transformed into a *cortical image* (fig. 2.4 left) by projecting its intensity values via Gaussian kernels centred on the polar-transformed retina sampling locations. This *cortical image* corresponds to a regular image matrix, thus it is compatible with current Deep Convolution Neural Network (DCNN) visual processing networks.

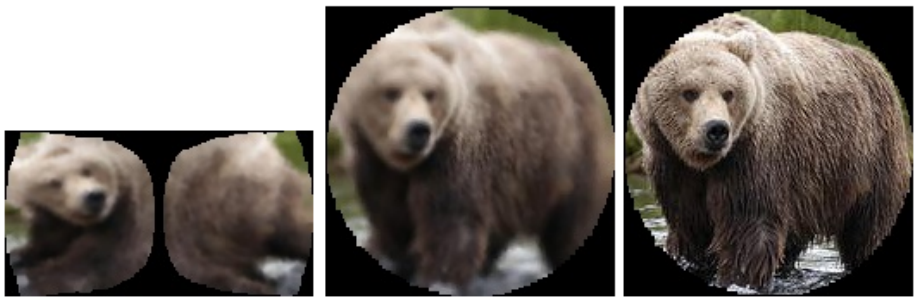


Figure 2.4: Left: Cortex image. **Center:** Retina back-projected **Right:** Input image. (Figure extracted from [5]).

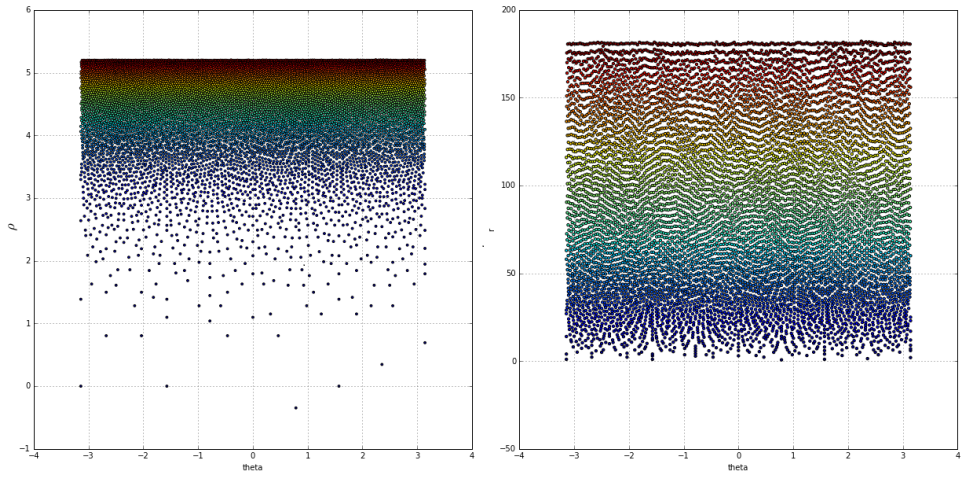


Figure 2.5: Colour coded receptive field centres mapped onto the log-polar (left) and linear polar (right) spaces. Warmer colours indicate receptive fields closer to the peripheries,whereas colder colours indicate points closer to the fovea. [4]

The cortical image is created by first mapping the receptive field centres onto a new space. Log-polar coordinates are the most commonly used ones, since they offer not only a *conformal mapping* (local angles are preserved with respect to the retinal image), but also rotation and scaling "*invariance*" (when the original image is rotated or scaled around its center, this corresponds to simple translations on the log-polar image) [16]. Log-polar coordinates consist of θ, the angle about the origin, and ρ, the logarithm of the euclidean distance from the origin:

$$\rho = \log\sqrt{x^2 + y^2}, \quad \theta = \text{atan2}(y, x) \tag{2.4}$$

where x and y are Cartesian coordinates relative to the origin.

Sibert and Ozimek considered that switching from log-polar coordinates to a linear polar space would provide an improvement on the severe sparsity of the cortical image in the fovea and extreme density at the periphery imposed by the log-polar coordinates, as one can see on fig. 2.5. This also mitigates the singularity issue when mapping the center of the fovea (log-polar coordinates do not map the Cartesian coordinate $x = 0$, $y = 0$). Although the cortical image will still be *conformal*, it will lose the scale invariance of the log-polar mapping [16].

Even with the improvement in node uniformity by switching to a linear polar space, the foveal region is still undesirably sparse and the extreme peripheries are packed in tight rows. As a consequence, a normalising parameter α was introduced, and the retina tessellation was vertically split into two halves (Figure 2.4 left), where each half is now mapped separately.

The resultant equations for the cortical mappings are as follows:

$$y_{right} = \sqrt{(x + \alpha)^2 + y^2}, \quad X_{right} = \text{atan2}(y, (x + \alpha)) \tag{2.5}$$

being y_{right} and x_{right} the new coordinates for the right side of the cortical mapping, and y_{left} and x_{left} the new coordinates for the left side of the cortical mapping:

$$y_{left} = -\sqrt{(x-\alpha)^2 + y^2}, \quad x_{left} = \text{atan2}(y, x-\alpha) - \text{sign}(\text{atan2}(y, x-\alpha))\pi \tag{2.6}$$

Siebert and Ozimek tested their approach on an image classification task. Their method reduced the visual data by approximately 7 times, the input data to the DCNN by 40% and the number of training epochs by 36%, at the expense of a reduction of 0.06 on the F1 score (0.80 in the foveated images instead of 0.86 in the original cartesian images), when compared to an identical network but trained and classified using full-resolution images (*e.g.,* fig. 2.4 right). Nevertheless, besides classifying images or objects, our project requires that we can localize the given objects on the world frame and assign to each point a classification score. The performance of Sibert and Ozimek's method to produce foveated images from Cartesian ones when detecting objects has still to be studied, since the object detection algorithms were built on the assumption of receiving as input Cartesian images. Thus, implementing their method would oblige a re-train of the chosen object detector, and the complexity upgrade from classifying images to classifying and locating objects (assigning a bounding box to an object) within an image would impose a problem when trying to detect on the resulting cortex image (fig. 2.4 left). This is something interesting for future work, but for our work we are more concerned on how to actively explore a scene and search for objects using foveated images.

So, even though Almeida's work [13] does not explore the computational advantages of a foveal image, it still provides a useful simulation to build our work upon. Using Almeida's foveated images, the object detection algorithms do not need to be re-trained, but their performance might need to be modeled to take into consideration the gradual increasing blur over the peripheries. An analysis on several state-of-the-art objection detection methods is made below.

2.1.2 Detection Algorithms

Classify and localize objects in an image is known as object detection. This problem can be approached by assigning one method to the classification of objects and another to the localization, or by taking advantage of their correlation, and build, for example, a single CNN capable of performing both classification and localization.

Current state-of-the-art object detection systems use a DCNN where, as a basis, they have in common a high-quality classifier and then vary on how to compute bounding boxes. Since our project is about active perception, both accuracy (Mean Average Precision (mAP)) and speed (Frames per Second (FPS)) are important. A comparison for different state-of-the-art approaches is presented on Table 2.1.

Table 2.1: Detection Systems on Pascal VOC2007. Comparing the performance and speed of state-of-the-art object detection methods. All timing information is on a Geforce GTX Titan X [7].

Method	mAP	FPS
Fastest DPM [17]	30.4	15
Faster R-CNN (VGG16) [18]	73.2	7
YOLO [19]	63.4	45
Fast YOLO [19]	52.7	**155**
SSD300 [20]	**74.3**	46

Before the advent of convolutional neural networks, Deformable Part Model (DPM) and Selective Search (SS) [21] were the state-of-the-art models for object recognition. The former was based on sliding windows and the latter on region proposal classification. After that, a considerable improvement was brought by R-CNN [22] where SS was combined with a convolutional network. This approach required the classification of thousands of image crops, which is expensive and time-consuming. From there on, several improvements have been made in a variety of ways.

Faster R-CNN [18] replaced SS proposals by ones learned from a Region Proposal Network (RPN), where the RPN was integrated with the R-CNN by sharing convolutional and prediction layers for these two networks. Faster R-CNN works by using a fixed set of anchor boxes proposed by the RPN to pool features and then evaluate them using the R-CNN. Although presenting an mAP above 70%, due to the complexity of the method it runs only at 7 FPS.

The state-of-the-art methods in terms of speed, skip the proposal step of the Faster R-CNN and predict bounding boxes and confidence scores for multiple categories directly. You Only Look Once (YOLO) [19] is an example of a method that unifies the separate components of object detection into a single neural network.

YOLO can already be considered a real-time detection method, but it pays the price of having a reduction in terms of performance when comparing with other state-of-the-art methods. It looks at the whole image, and by sharing the information between predicted confidence scores for multiple categories and bounding boxes, it outputs the final detections. The significant improvement in terms of speed is also due to the considerable decrease on the number of proposed bounding boxes, when compared to Faster R-CNN.

Single Shot MultiBox Detector (SSD) [20] followed the approach of using a single neural network to perform object detection. It combined both previously mentioned approaches by using default boxes (just like the anchor boxes on the Faster R-CNN) instead of using the whole image, but without having the proposal step. The results on Table 2.1 show that SSD can both combine real-time detection with state-of-the-art performance, although it is very sensitive to the bounding box size, which means it has much worse performance on smaller objects than bigger ones.

Since we are exploring the scene for objects in an iterative manner, the speed of the detector is of the utmost importance. Both YOLO and SSD achieve a reasonable performance in terms of FPS,

although YOLO lacks on performance when compared to the other method. For our purpose this slight drop in performance is not of concern, since it will be compensated by the active exploration. On the other hand, the much worse performance on smaller objects of the SSD could raise problems when searching for objects, since in a real environment there are objects of different sizes and they might be in a considerable distance to the camera, making them even smaller. This way,the detector chosen for this work was the YOLO object detector, more precisely a *TensorFlow* implementation of YOLOv3 [23].

Going a bit into more detail about the YOLO implementation used in this work, the YOLO first divides the image in a grid to then classify each grid cell using pre-defined anchor boxes (fig. 2.6).

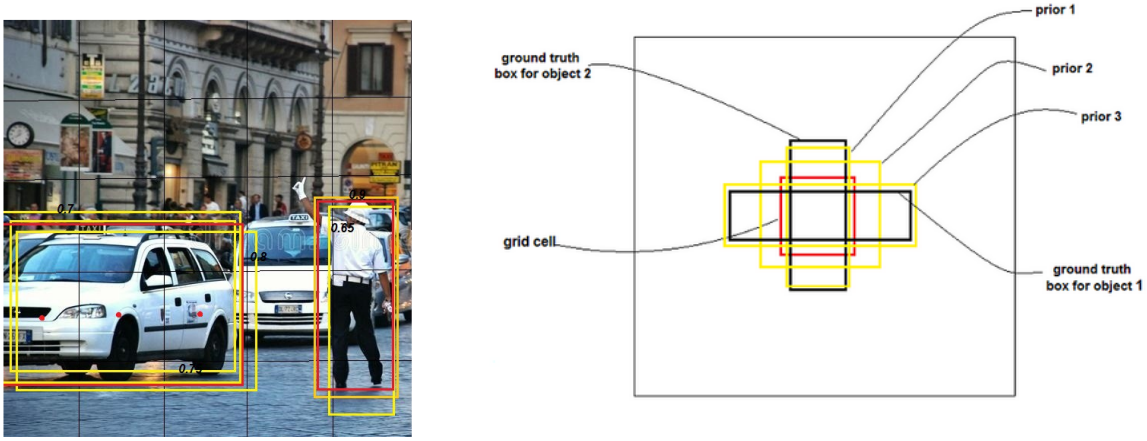


Figure 2.6: Left: Image with proposed bounding boxes from the YOLO algorithm, with a "toy" grid put on top of the image. **Right:** Example of pre-defined anchor boxes.

Since the implementation used corresponds to the YOLOv3 object detector, instead of using 1 fixed size grid, uses 3 different scales. On fig. 2.7 it is represented the model architecture for the implementation of the YOLOv3 used in this work [1]. On the last layers of the networ, one can see the three different feature maps that correspond to the three different grids put on top of the image.

2.2 Probability Distributions

This section is just an overview of the probability distributions considered for our framework. We will first go through the categorical distribution, since the classifiers outputs are often interpreted as the expected value of this distribution, and then, we will introduce the Dirichlet distribution, which will allow us to add more variables to the observation model, such as the distance to the fovea and the object class.

[1] Wizyoung, YOLOv3 TensorFlow implementation: https://github.com/wizyoung/YOLOv3_TensorFlow.

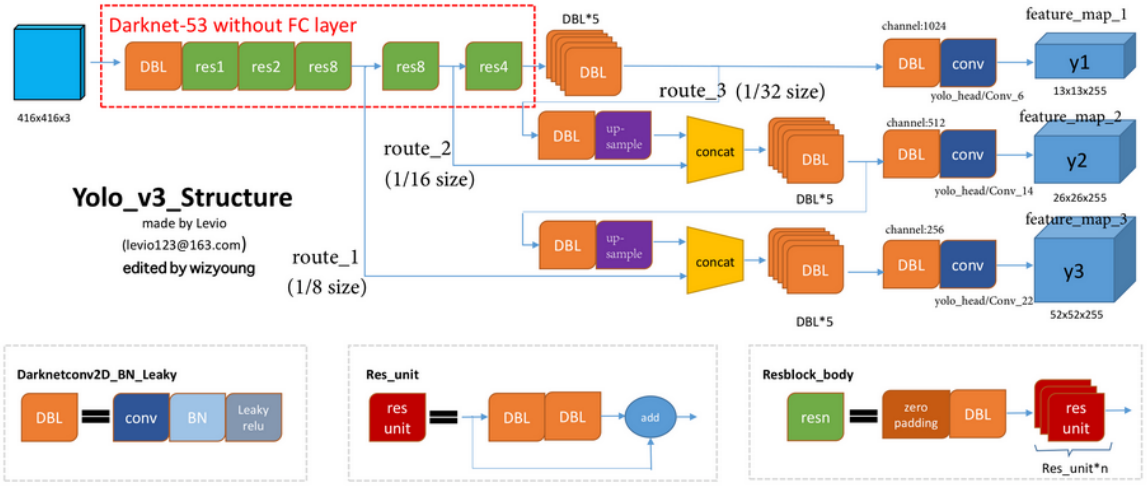


Figure 2.7: Model architecture of the YOLOv3 implementation.

2.2.1 The Categorical Distribution

The categorical distribution is a discrete probability distribution that describes the results of a random variable that can belong to one of K possible classes (is a special case of the multinomial distribution but for a single trial rather than multiple trials). The parameters of this distribution can be represented as

$$\boldsymbol{p} = [p_1, ..., p_K] \tag{2.7}$$

The outputs of the classifiers are often interpreted as the expected value of the categorical distribution that models the generation of objects:

$$p(c_k|\boldsymbol{p}) = p_k = s_{t,l,k} \tag{2.8}$$

considering a generic time instant t and index l, and $k = 1, ..., K$.

Nevertheless, we propose a different observation model, where the parameters depend not only on the class k, but also on the distance to the fovea $d_{t,l}$:

$$p(\boldsymbol{S}_{t,l}|c_k, d_{t,l}) = Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{k,d_{t,l}}) \tag{2.9}$$

where $\boldsymbol{\alpha}_{k,d_{t,l}}$ are the pre-trained parameters of the Dirichlet distributions that depend on the class of objects c_k and on the distance $d_{t,l}$ of the detection $I_{t,l}$ to the center of the fovea. The Dirichlet distribution and its estimation process will be explained below.

2.2.2 The Dirichlet Distribution

Although the foveation method implemented by Almeida [13] transforms the image in a way that does not require a re-train of the detection algorithms, it does not model the output of the classifier as the distance to the focal point varies, to then use correctly this information on the data fusion process.

This way, it would be interesting to model the response of the detection scores outputted by the object detector, as the objects appear closer or further away to the center of the fovea. Since the detection scores can be interpreted as parameters of a categorical distribution, their distribution can be modeled by the Dirichlet distribution.

Let's imagine a two-dimensional case, where we wish to model the distribution of possible probabilities of an event resulting in a given outcome, over 2 possible ones ($x = 0, x = 1$), by observing all prior occurrences of such event. One way to model this event would be to observe the number of times that the outcome $x = 0$ occurs and the same for the outcome $x = 1$, and store them in α and β variables, respectively. The distribution is, thus, given by

$$f(x) \propto x^{\alpha-1}(1-x)^{\beta-1} \tag{2.10}$$

which corresponds to the Beta distribution. Examples of the Beta distribution can be seen on fig. 2.8, where the expected value is given by the occurrences ratio, and the variance (uncertainty) decreases as the number of occurrences increase.

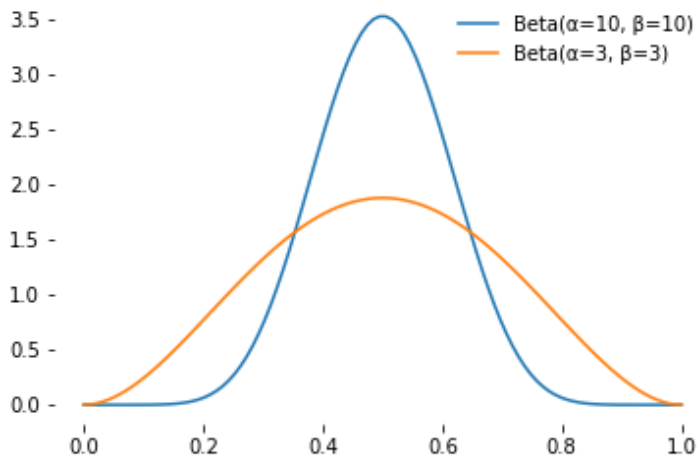


Figure 2.8: Beta distribution examples, for different α and β parameters

The Dirichlet distribution is simply a generalization of the beta distribution to higher dimensions. A K-dimensional Dirichlet will be defined as a distribution over a K-tuple $(p_1, ..., p_K)$ (which represents the

parameters of a multinomial distribution, where the categorical distribution is a special case of such), where $\sum_{k=1}^{K} p_k = 1$ and $p_k \geq 0$ for all $k \in \{1, ..., K\}$. The Dirichlet distribution is given by

$$Dir(\boldsymbol{p}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \tag{2.11}$$

where the α parameters are all positive and $B(\boldsymbol{\alpha})$ corresponds to the multivariate beta function, which serves as a normalizing constant and can be written in terms of the Gamma function $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$, which is a generalization of the factorial to real and complex numbers:

$$B(\boldsymbol{\alpha}) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)}, \quad \boldsymbol{\alpha} = [\alpha_1, ..., \alpha_K]. \tag{2.12}$$

Figure 2.9 plots several examples of the shape of the Dirichlet distribution for the three-dimensional case (K=3), considering different sets of $\boldsymbol{\alpha}$ parameters.

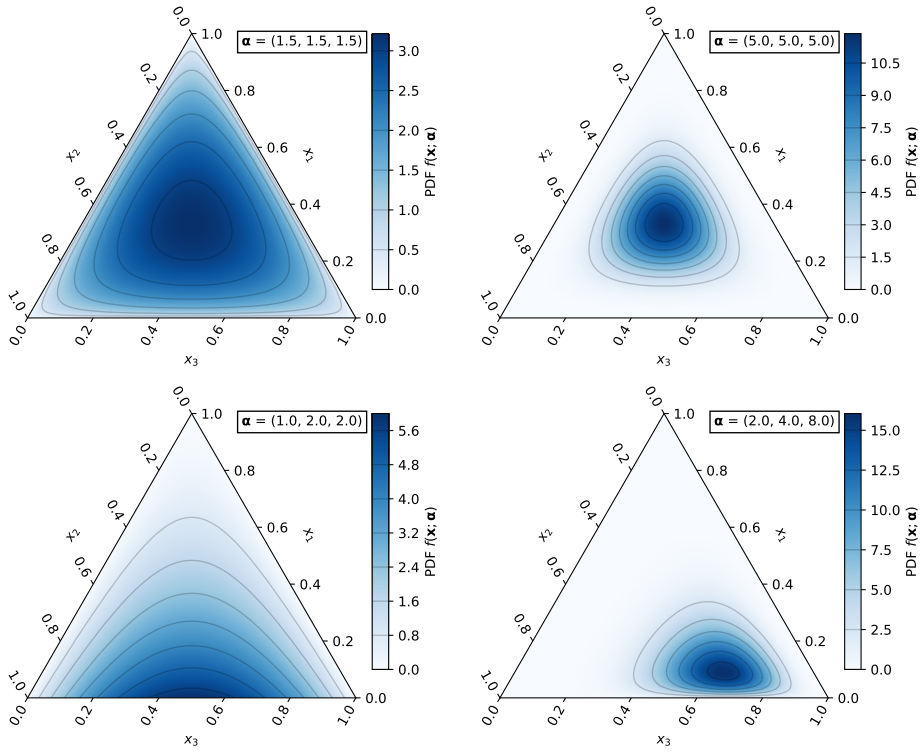


Figure 2.9: Several three-dimensional Probability Density Function (pdf) of the Dirichlet distribution, for different α parameters. (Image extracted from https://en.wikipedia.org/wiki/Dirichlet_distribution).

As said previously, the Dirichlet distribution would be an useful tool to model the confidence scores outputted by the object detector. This is possible since the scores can be interpreted as a random variable due to the natural diversity of the objects' appearance and form on a scene, as well as, in our

case, their distance to the center of the fovea (due to the blur imposed by the foveal sensor). Moreover, since the scores are normalized to 1, can be generated by a Dirichlet distribution sampling with a given α parameters. Therefore, eq. (2.11) can be written as follows

$$Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{j,d_{t,l}}) = \frac{1}{B(\boldsymbol{\alpha}_{j,d_{t,l}})} \prod_{k=1}^{K} s_{t,l,k}^{\alpha_{j,d_{t,l},k}-1} \tag{2.13}$$

where $\boldsymbol{S}_{t,l}$ is the l-th confidence score outputted by the object detector at instant t, $s_{t,l,k}$ is the score on $\boldsymbol{S}_{t,l}$ for the k-th class, as formulated on eq. (1.6), and $\alpha_{j,d_{t,l},k}$ is the parameter k of the vector $\boldsymbol{\alpha}_{j,d_{t,l}}$. The parameters $\boldsymbol{\alpha}_{j,d_{t,l}}$ determine the amount of variation of the confidence scores $\boldsymbol{S}_{t,l}$, and the correlation between them, depending on the object (given by the class label c_j) and their position with respect to the focal point $d_{t,l}$. Thus, the parameters of the corresponding Dirichlet distributions have to be estimated not only according to each particular object class but also its distance to the fovea. A possible way of estimating each Dirichlet distribution was proposed by Minka [24], which we will replicate in our work.

Minka uses a simple reparameterization of the Dirichlet, given by defining the precision ν and the mean $\boldsymbol{m}$ of the Dirichlet distribution, which are given by

$$\nu = \sum_{k=1}^{K} \alpha_k \tag{2.14}$$

$$\boldsymbol{m} = \left(\frac{\alpha_1}{\nu}, ..., \frac{\alpha_K}{\nu}\right) \tag{2.15}$$

thus, the parameters of the Dirichlet can be written on the form

$$\alpha_k = \nu m_k \tag{2.16}$$

Since $\boldsymbol{m}$ corresponds to the mean of the distribution, it sums to unity ($\sum_{k=1}^{K} m_k = 1$) and each parcel k of the mean vector actually corresponds to the expected value of the score vector for class k ($m_k = E[s_k]$). On the other hand, ν being the precision of the Dirichlet, controls how concentrated the distribution is around its mean, as one can see on the examples plotted on fig. 2.9, the greater the Dirichlet parameters α, the greater the precision ν, and, thus, a higher concentration around the mean of the distribution.

2.2.2.A Estimating a Dirichlet Distribution

Generally, the $\boldsymbol{\alpha}$ parameters of a Dirichlet distribution can be estimated from a training set of multinomial data, $\mathcal{D} = \{\boldsymbol{S}_1, ..., \boldsymbol{S}_N\}$, where N is the number of training score vectors, by maximising the log-likelihood

function of the data given by [24]

$$\begin{aligned}\log p(\mathcal{D}|\boldsymbol{\alpha}) &= \log \prod_{i=1}^{N} Dir(\boldsymbol{S}_i|\boldsymbol{\alpha}) \\ &= \log \prod_{i=1}^{N} \frac{\Gamma\left(\sum_k \alpha_k\right)}{\prod_k \Gamma\left(\alpha_k\right)} \prod_{k=1}^{K} s_{i,k}^{\alpha_k - 1} \\ &= N\left(\log \Gamma\left(\sum_k \alpha_k\right) - \sum_k \log \Gamma\left(\alpha_k\right) + \sum_k \left(\alpha_k - 1\right) \log \bar{s}_k\right)\end{aligned} \tag{2.17}$$

where $\log \bar{s}_k = \frac{1}{N} \sum_i \log s_{i,k}$. For the purpose of the Dirichlet estimation, for simplicity, we are ignoring the instant of time of the detections and we are also considering a generic $\boldsymbol{\alpha}$ parameter, ignoring the dependence of this parameter on the type of object and distance to the focal point.

As there is no closed form solution to maximize this objective function, iterative methods to find the maximum have to be adopted. One possible way to estimate the Dirichlet distribution, is one implemented by Minka [24], optimizing the mean and the precision alternately, fixing one parameter and only optimizing the other, obtaining simplifications and speedups that ease the training process.

After doing a reparameterization of the α parameters as in eq. (2.16) and substituting in eq. (2.17), one can extract the likelihood for the precision ν alone, and it has the form

$$p(\mathcal{D}|\nu) \propto \left(\frac{\Gamma(\nu)\exp(\nu \sum_k m_k \log \bar{s}_k)}{\prod_k \Gamma(\nu m_k)}\right)^N \tag{2.18}$$

Whose derivatives are:

$$\frac{d \log p(\mathcal{D}|\nu)}{d\nu} = N\left(\Psi(\nu) - \sum_k m_k(\Psi(\nu m_k) + \log \bar{s}_k)\right) \tag{2.19}$$

$$\frac{d^2 \log p(\mathcal{D}|\nu)}{d\nu^2} = N\left(\Psi'(\nu) - \sum_k m_k^2 \Psi'(\nu m_k)\right) \tag{2.20}$$

where $\Psi(\nu) = \frac{d \log \Gamma(\nu)}{d\nu}$, which is know as the digamma function and is similar to the natural logarithm, and $\Psi'(\nu) = \frac{d\Psi(\nu)}{d\nu}$.

Using Minka's [24] generalized Newton iteration, results the following update:

$$\frac{1}{\nu^{new}} = \frac{1}{\nu} + \frac{1}{\nu^2}\left(\frac{d^2 \log p(\mathcal{D}|\nu)}{d\nu^2}\right)^{-1}\left(\frac{d \log p(D|\nu)}{d\nu}\right) \tag{2.21}$$

To initialize the precision ν, one can use Stirling's approximation to Γ [25], resulting in

$$\frac{\Gamma(\nu)\exp(\nu\sum_k m_k \log \bar{s}_k)}{\prod_k \Gamma(\nu m_k)} \approx \left(\frac{\nu}{2\pi}\right)^{(K-1)/2} \prod_k m_k^{1/2} \exp\left(\nu \sum_k m_k \log \frac{\bar{s}_k}{m_k}\right) \tag{2.22}$$

and then extract the initialization of the precision $\hat{\nu}$, given by

$$\hat{\nu} \approx \frac{(K-1)/2}{-\sum_k m_k \log \frac{\bar{s}_k}{m_k}} \tag{2.23}$$

Now, fixing the precision ν to estimate the mean $\boldsymbol{m}$, and doing the same reparameterization and substitution as for the precision. The likelihood for $\boldsymbol{m}$ alone is:

$$p(\mathcal{D}|\boldsymbol{m}) \propto \left(\prod_{k=1}^{K} \frac{\exp(\nu m_k \log \bar{s}_k)}{\Gamma(\nu m_k)}\right)^N \tag{2.24}$$

Now reparametrizing the likelihood function with an unconstrained vector $\boldsymbol{z}$ to get the gradient:

$$m_k = \frac{z_k}{\sum_{j=1}^{K} z_j} \tag{2.25}$$

the log-likelihood for $\boldsymbol{m}$ alone can be written as

$$\log p(\mathcal{D}|\boldsymbol{m}) = N \sum_{k=1}^{K} \left[\frac{z_k}{\sum_j z_j} \log \hat{s}_k - \log \Gamma\left(\nu \frac{z_k}{\sum_j z_j}\right)\right] \tag{2.26}$$

In order to find the maximum, one can compute the gradient of the log-likelihood, which is given by

$$\frac{d \log p(\mathcal{D}|\boldsymbol{m})}{dz_k} = \frac{\nu N}{\sum_j z_j} \left(\log \bar{s}_k - \Psi(\nu m_k) - \sum_j m_j (\log \bar{s}_j - \Psi(\nu m_j))\right) \tag{2.27}$$

We are now in conditions to use the Maximum Likelihood Estimation (MLE) to find the new mean value $\boldsymbol{m}^{new}$. The MLE can be computed by the fixed-point iteration by solving the equation

$$\frac{d \log p(\mathcal{D}|\boldsymbol{m})}{dz_k} = 0 \tag{2.28}$$

which can be rewritten as

$$\Psi(\alpha_k) = \log \bar{s}_k - \sum_j m_j^{old} (\log \bar{s}_j - \Psi(\nu m_j^{old})) \tag{2.29}$$

and results on the new mean value

$$m_k^{new} = \frac{\alpha_k}{\sum_k \alpha_k} \tag{2.30}$$

This process of alternating between estimating the mean and precision converges very quickly. [26]

2.3 Approaches on Fusing Classifiers

The fusion problem consists in, given a set of classification scores for a single pattern (which can be from distinct classifiers that produce observations at the same time, to a single classifier that obtains consecutive measurements for the same pattern but in different time instants), how can one calculate a single global classification score $\boldsymbol{p}$ and/or estimate its distribution. For our specific case the objective here would be to update the world map M enunciated in eq. (1.8)

As an example, Montesano [27] developed an algorithm that learns local visual descriptors of good grasping points based on a set of trials performed by the robot. The parameters of the corresponding distribution (in this case Beta distribution) are updated as a simple function of the number of successes and failures.

Although we will not have ”successes” and ”failures” to use Montesano’s approach, we can use Figueiredo’s sequential Bayesian filtering. Figueiredo’s work [28] was about modeling depth uncertainty in stereo reconstruction, due to space-variant discretization in foveated images that decreases stereo matching accuracy in the image periphery. At each instant the modeled uncertainty is used to update a map that is then used to decide what is the next best view point. Bayesian filtering allows one to accumulate sensor inputs and update the likelihood of a map point being the desired object, at each time instant, assuming that we know the probability distribution of the data.

2.3.1 Bayesian filtering - Naïve Bayes approach

Bayesian filtering, or Naïve Bayes fusion, is a classical fusion approach where there are a number of incoming measurements, which in our case correspond to the confidence scores defined by eq. (1.6), that need to be fused in order to estimate their distribution.

For the purpose of this section let’s assume that we have T sequential measurements for the same image location, one at each instant of time t and, therefore, we can write the set of observations as follows

$$\mathcal{L} = \{\boldsymbol{L}_1, ..., \boldsymbol{L}_{t-1}, \boldsymbol{L}_t, \boldsymbol{L}_{t+1}, ..., \boldsymbol{L}_T\} \tag{2.31}$$

where $\boldsymbol{L}_t$ is the likelihood vector containing the likelihoods $l_{t,j}$ of a given detection outputted at instant t belonging to each object class c_j, for $j = 1, ..., K$. On our case, this likelihoods correspond to the

confidence scores defined on eq. (1.6)

$$l_{t,j} = s_{t,1,j} \tag{2.32}$$

being $s_{t,1,j}$ the first confidence score for the object class c_j outputted at the instant t.

We are now in conditions to rewrite the posterior distribution after fusing T classifiers, following eq. (1.8) but ignoring the image location, since we are considering observation for the same image location, as follows

$$\boldsymbol{M}_T = P(C|\mathcal{L}) = \frac{P(C)P(\mathcal{L}|C)}{P(\mathcal{L})} \tag{2.33}$$

which is our goal to predict. The last term is given by applying the Bayes' Theorem.

Starting by the numerator of the result of the Bayes' theorem on eq. (2.33), it is equivalent to the joint probability

$$P(C)P(\mathcal{L}|C) = P(C, \boldsymbol{L}_1, ..., \boldsymbol{L}_T) \tag{2.34}$$

where, by applying the chain rule for repeated applications of the definition of conditional probability:

$$P(C, \boldsymbol{L}_1, ..., \boldsymbol{L}_T) = P(\boldsymbol{L}_1|\boldsymbol{L}_2, ..., \boldsymbol{L}_T, C)P(\boldsymbol{L}_2|\boldsymbol{L}_3, ..., \boldsymbol{L}_T, C) \cdots P(\boldsymbol{L}_{T-1}|\boldsymbol{L}_T, C)P(\boldsymbol{L}_T|C)P(C) \tag{2.35}$$

Under the "naïve" assumption that the observations are mutually independent, conditional on the category C, i.e., the fusion process is a first-order Markov chain model (as can be seen graphically by the Hidden Markov Model (HMM) represented on fig. 2.10), the joint probability can be written as follows

$$P(\boldsymbol{L}_t|\boldsymbol{L}_{t+1}, ..., \boldsymbol{L}_T, C) = P(\boldsymbol{L}_t|C) \tag{2.36}$$

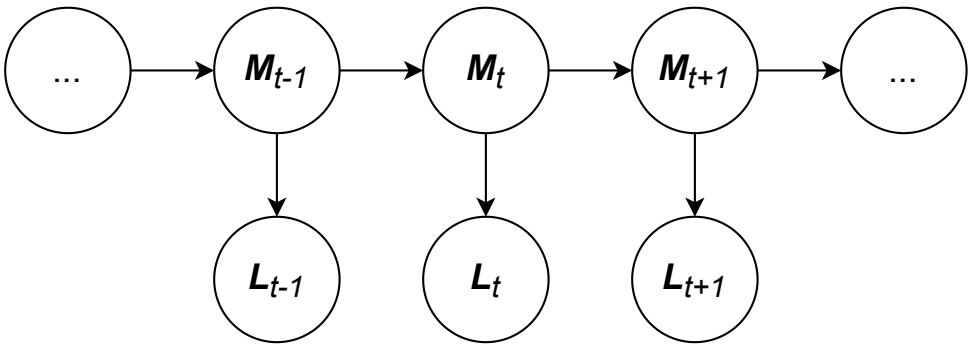


Figure 2.10: HMM of the whole fusion process

Regarding the denominator on eq. (2.33), it can be seen as a constant since it does not depend on C. Thus, the whole fusion model can be expressed as

$$\boldsymbol{M}_T = P(C|\boldsymbol{L}_1, ..., \boldsymbol{L}_T) = \frac{1}{Z}P(C)\prod_{t=1}^{T} P(\boldsymbol{L}_t|C) = \frac{1}{A}P(\boldsymbol{L}_T|C)\boldsymbol{M}_{T-1} \tag{2.37}$$

and, individualizing for each class of objects c_j

$$p(c_j|\mathcal{L}) = \frac{1}{Z} p(c_j) \prod_{t=1}^{T} l_{t,j} \tag{2.38}$$

where A and Z are normalizing constants that do not depend on the object class, and $p(c_j)$ is the prior class probability for c_j. Equation (2.38) corresponds to the classical product rule, or Naïve Bayes fusion approach.

2.3.2 Sum Rule Fusion Approach

There is also another classical approach that tries to approximate the posterior class probabilities, but instead of multiplying the observations, it sums them. This is called the sum rule, which is given by

$$p(c_j|\mathcal{L}) = \frac{1}{T} p(c_j) \sum_{t=1}^{T} \frac{l_{t,j}}{\sum_{k=1}^{K} p(c_k) l_{t,k}} \tag{2.39}$$

The sum rule is interesting since it is more robust to outliers (as the number of classifications increase) than the Naïve Bayes approach, for example, likelihood values close to zero would automatically lead the result of the Naïve Bayes to low probability values. This can be important since our observations might have a large uncertainty specially when the object are on the peripheries of the fovea. An interesting work on testing the performances of the product and sum rules was performed by Kaplan [29], where they were compared to a new approach to fuse classifiers.

2.3.3 Kaplan's Approach on Fusing Classifiers

The fusion method proposed by Kaplan maps the "opinions" of the classifiers into a Dirichlet distribution, being able to take into consideration the uncertainty associated to each classifier when fusing two "opinions". This is an interesting topic to our project, since depending on the location of the bounding box, in relation to the center of the fovea, the uncertainties associated to the classification should be different.

Let us define a Dirichlet distribution with parameters $\boldsymbol{\beta}$ for the categorical distribution $\boldsymbol{p}$,

$$h(\boldsymbol{p}|\boldsymbol{\beta}) = \begin{cases} Dir(\boldsymbol{p}|\boldsymbol{\beta}), & \text{for } \boldsymbol{p} \in \mathcal{P}, \\ 0, & \text{otherwise} \end{cases} \tag{2.40}$$

Kaplan's method implies that opinions are formed by observations that increment the Dirichlet parameters of the current multinomial opinion. Given that the current multinomial opinion corresponds to Dirichlet parameters $\boldsymbol{\beta}$, then the prior distribution for $\boldsymbol{p}$ is $h(\boldsymbol{p}|\boldsymbol{\beta})$. Thus, when the target class is observed, the probability of observing the class as the j-th singleton, given $\boldsymbol{p}$ is $p(c_j|\boldsymbol{p})$, which is simply p_j. This way, the posterior for $\boldsymbol{p}$ given that the class label C of an arbitrary observation corresponds to the

class c_j is given by

$$P(\boldsymbol{p}|C = c_j) = \frac{P(c_j|\boldsymbol{p})P(\boldsymbol{p})}{P(C = c_j)} = \frac{p_j h(\boldsymbol{p}|\boldsymbol{\beta})}{\int_{\mathcal{P}} p_j h(\boldsymbol{p}|\boldsymbol{\beta})} = h(\boldsymbol{p}|\{\boldsymbol{\beta} + \boldsymbol{e}_j\}) \tag{2.41}$$

being the second term of the equation the direct application of Bayes' theorem, and $\boldsymbol{e}_j$ a vector where the j-th element is one and all the others are zero. Thus, on this situation, the updated Dirichlet parameters are simply

$$\beta_{t+1,k} = \beta_{t,k} + \delta_{kj} \tag{2.42}$$

where $\boldsymbol{\beta}_{t+1}$ are the updated Dirichlet parameters, $\boldsymbol{\beta}_t$ the current Dirichlet parameters, and δ is the Kronecker delta, for k=1,...,K. This is actually what Montesano [27] uses in his work, where occurrences of each singleton k can be counted N_k, resulting on a simple update,

$$\beta_{t+1,k} = \beta_{t,k} + N_k \tag{2.43}$$

Nevertheless, as stated before, the singletons can not be directly observed. Thus, this update method can not be directly applied.

What we do have, is a statistical measure related with the occurrence of the singleton. By Kaplan, a naïve approach for the update of the Dirichlet parameters is to spread the mass of the Dirichlet update in eq. (2.42) via the normalized likelihood

$$\beta_{t+1,k} = \beta_{t,k} + \frac{l_{t,k}}{\sum_{j=1}^{K} l_{t,j}} \tag{2.44}$$

This approach is actually equivalent to the sum rule in eq. (2.39), considering the uniform prior. Nevertheless, Kaplan states that this naïve approach does not yield a posterior Dirichlet distribution that fits well the actual posterior distribution of $\boldsymbol{p}$.

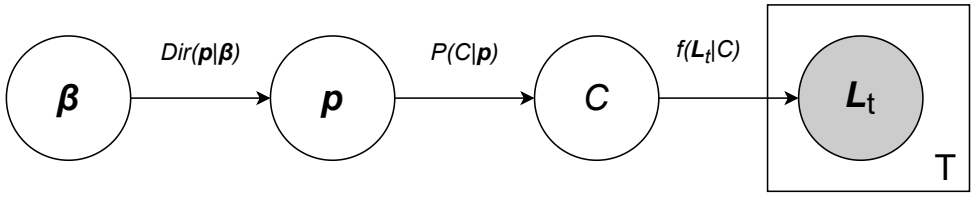


Figure 2.11: Bayesian Network representing the joint pdf of the measurement, where T represents the total number of time instants considered.

Kaplan then tries to find the actual posterior after a measurement update. Kaplan starts the derivation with the joint pdf of the measurement, the hidden observation, and the observation probabilities

conditioned on the current multinomial opinion (the graphical model is represented on fig. 2.11):

$$f(\boldsymbol{L}_t, C = c_j, \boldsymbol{p}|\boldsymbol{\beta}) = f(\boldsymbol{L}_t|C = c_j)p(C = c_j|\boldsymbol{p})h(\boldsymbol{p}|\boldsymbol{\beta}) = l_{t,j}p_j f_\beta(p|\boldsymbol{\beta}). \tag{2.45}$$

Then, removing the hidden variable c_j by marginalization leads to

$$f(\boldsymbol{L}_t, \boldsymbol{p}|\boldsymbol{\beta}) = \left(\sum_{k=1}^{K} l_{t,k}p_k\right) h(\boldsymbol{p}|\boldsymbol{\beta}) \tag{2.46}$$

so that the posterior for the observation probabilities after the measurement update is

$$f(\boldsymbol{p}|\boldsymbol{\beta}, \boldsymbol{L}_t) = \left(\sum_{k=1}^{K} \beta_k\right) \frac{\left(\sum_{k=1}^{K} l_{t,k}p_k\right) h(\boldsymbol{p}|\boldsymbol{\beta})}{\left(\sum_{k=1}^{K} l_{t,k}\beta_k\right)} \tag{2.47}$$

It is possible to note that, when the likelihood is zero for all classes except for one, eq. (2.47) simplifies to eq. (2.42), which means that there is no uncertainty on the target class (it is completely visible). For the case where all classes have equal likelihoods eq. (2.47) simplifies to $h(\boldsymbol{p}|\boldsymbol{\beta})$, meaning that a vacuous observation (where all classes have equal likelihoods) does not update the posterior probability, which clearly does not happen when using the naïve approach given by eq. (2.44).

The next step is to approximate the posterior by the Dirichlet distribution. Kaplan considers a moment matching approach to do the approximation. The moment matching approach determines the Dirichlet distribution that exhibits the same mean as the posterior, and attempts to approximate the variance of the posterior. By the moment matching approach and adding the constraint that the update can not be negative, i.e, $\beta_{t+1,k} > \beta_{t,k}$, for all $k = 1, ..., K$, Kaplan states that the updated Dirichlet parameters are

$$\beta_{t+1,k} = \frac{\beta_{t,k}\left(1 + \frac{l_{t,k}}{\sum_{j=1}^{K}\beta_{t,j}l_{t,j}}\right)}{1 + \frac{\min_j l_{t,j}}{\sum_{j=1}^{K}\beta_{t,j}l_{t,j}}} \tag{2.48}$$

This update proposed by Kaplan (Kaplan's update) represents a new approach on fusing classifiers, thus, it will be interesting to test how this different updates work, as we will be dealing with the uncertainty imposed by the foveal sensor that is not constant: depends on the object location. We will, therefore, expand Kaplan's work on comparing different fusion methods, to our specific problem.

2.4 Active Perception

Machine Learning techniques often rely on huge amounts of labeled data. The data is then processed by a training algorithm, which optimizes the parameters to perform the task for what it was designed. One constrain of these machine learning techniques, and perhaps the biggest one, is the insufficient

amount of available data to train the algorithm, and the time it would take to process it.

To overcome this issue, active learning began to emerge as a hot scientific topic. Active Learning is built upon the principle that the learning algorithm has the ability to choose the data from which it learns, and, this way, if the the data is well chosen, the algorithm can perform better with less training [30].

Active perception is a particular subset of active learning. The agent acquires information directly from the sensors, which is combined with prior knowledge of the world and the current state, to then select the next information to gather [31]. Active perception can be performed with different kinds of sensors and stimulus. The focus of this work is on solving a search problem using visual sensory information, and, therefore, the active perception specialization that will be studied here is given the name of active vision [32]. This problem can be seen as a planning problem, denominated "next best view point".

2.4.1 Acquisition Functions

The choice of the next best view point is made through the use of acquisition functions, where the objective is to choose the point that maximizes a function related to our objective. In our case this function would be the information gained about the scene which needs to be estimated for each possible view point. This function is known as a reward function and we will represented it as $r^t_{x,y}$, where (x, y) are the world coordinates of the view point (refer back to fig. 1.3) and t represents the instant of time to which we are referring.

Figueiredo [28] on his work used three common acquisition functions that can be seen as a reference. These acquisition functions are defined as follows:

- Upper Confidence Bound - where at each instant of time, the alternative with maximal upper confidence bound on the expected reward is chosen, given the past observations, according to the following expression

$$x^*, y^* = \underset{x,y}{\operatorname{argmax}} \, \mathrm{E}\left[r^t_{x,y}\right] + \sigma \mathrm{Var}\left[r^t_{x,y}\right] \tag{2.49}$$

  where $\mathrm{E}[\cdot]$ and $\mathrm{Var}[\cdot]$ denote the expectation and variance operators, (x^*, y^*) correspond to the world coordinates of the estimated next best view point, and σ is a user-defined parameter that will balance the exploratory behaviour of the agent (higher σ means more exploration over exploitation).

- Probability of Improvement - where at each instant of time, selects the action with highest probability of leading to an improvement upon the current best r^*, as follows

$$x^*, y^* = \underset{x,y}{\operatorname{argmax}} \, P\left(r^t_{x,y} > r^*\right) \tag{2.50}$$

- Expected Improvement - tries to maximize the expected magnitude of the improvement upon the so far best,

$$x^*, y^* = \underset{x,y}{\operatorname{argmax}} \, \mathrm{E}\left[r^t_{x,y} > r^*\right] \tag{2.51}$$

The application of the acquisition functions depends obviously on the type of information acquired by the sensors and others can be derived from these reference ones. On section 3.3 the acquisition functions developed to our specific case will be explained in detail.

2.4.2 Active Perception on Cartesian Domain

Since our work is built upon using foveated images, approaches on applying active perception methods on this type of images will be explored in detail. Nevertheless, it is important to state that there have been attempts on visually searching for objects on the Cartesian domain.

Ayedemir [11] focused his research on actively searching for objects by first searching for the most plausible locations. In other words, Ayedemir proposed an Active Visual Search (AVS) strategy considering topological relations between objects. The approach had a major drawback, the amount of prior information needed, which the user had to input whenever a new search was to be performed, as the results showed that the prior probabilities had a great influence on the outcome of the search. The same goes for his next work where he added the uncertain semantic of the environment [12]. Anyway, Aydemir latest work already provided promising results when compared against humans on performing an object search task on unknown map.

On another approach, a new mechanism combining stereo vision and active perception was proposed by Grotz, where a more task-related gaze selection was explored, based on multiple saliency maps [33]. His objective was to reduce the uncertainty associated to the desired object pose to then be able to grab it more efficiently. For that purpose, Grotz used two saliency maps containing cues that should attract the robot's attention. Since multiple features can be of interest when choosing the next best view point, multiple saliency maps can be accumulated using different weights. The region with the highest resulting saliency defines the next best view direction. One of the saliency maps proposed by Grotz maps the pose uncertainty of the desired object. The detection of the object was made based on the extraction of local features and the uncertainty associated to the object pose was modelled as a Gaussian distribution and updated using a Kalman filter. The other saliency map tries to draw the attention of the robot to single color blobs on the scene. Nevertheless, the use of local feature detectors greatly reduces the possible complexity of the objects present on the scene. One solution would be to take advantage of state-of-the-art object detectors, but that would greatly increase the computational effort. Thus, even tho Cartesian images are easier to understand and integrate with the world state knowledge and the gaze selection, the gain on computational time and performance provided by the

foveal mechanism is of great interest for active perception purposes.

2.4.3 Integration with Foveated images

Earlier work on the integration of foveal vision mechanisms and active vision was performed by Rivlin and Rotstein where they formulate a setup where the combination of foveal vision and a tracking scheme can be evaluated in a systematic manner [34]. For this purpose, they had to take into consideration the camera control and the image processing (next best view point).

Our emphasis will be on the next best view point problem and the corresponding integration with foveated images. We were inspired on a recent work developed by Figueiredo [28] where depth information was combined with the uncertainty in stereo matching to perform an active gaze selection method. The objective was to extract the maximum amount of information of the closest object to the camera while updating the world map using foveal mechanisms.

Figueiredo's results showed that, with the right parameters, foveal vision would outperform Cartesian, regarding the amount of information extracted. Nevertheless, besides the promising results, the optimization criteria was to choose the closest object, with disregard for the type of object itself. This is something that we want to explore further, by trying to differentiate the objects that compose the scene.

Following that line of thought, and following Figueiredo's work [28], an iterative approach combining saliency maps (inspired on Grotz approach [33]) with active perception to improve the detection of objects was proposed by Almeida [13]. Almeida proposes a biological inspired object classification and localization framework combining DCNN with foveal vision. First, a DCNN operates over the foveated image to predict the class labels. Then, a color-based saliency map is used to obtain the object location proposal. At the next iteration, the center of the location proposal is used as the new foveation point, and the process is repeated, in order to try to improve the classification and localization of the object. As in Grotz work, the use of this kind of saliency maps reduces the quality of the localization of the objects as the image gets more complex. Besides that, Almeida's framework considered images with just one object. We wish to remove these constraints and explore a complex scene, gathering information about all the objects, increasing the complexity of choosing the next best view point.

Other biological inspired work was performed by Melicio [2], where attention mechanisms were combined with foveal vision to perform image classification. Melicio dropped the model based saliency maps by using a CNN to both detect salient regions and classify the foveated image in just one step. The salient regions outputted by the CNN were then used to shift the foveation point to locations that would potentially improve the classification. Melicio showed that after the gaze shift, the performance improves. In her work the localization of the object is not required, since, as in Almeida's work, the objective was to classify an image containing one central object. Thus, the uncertainty in the detection imposed by the foveal vision did not need to be modelled.

3

Approach

Contents

3.1 Object Detection & Foveal Observation Model 35

3.2 Fusion Model 38

3.3 Active Perception - Gaze Selection 44

We wish to perform a scene exploration on an image, so, first of all, it is assumed that the agent can center the fovea at every position on the image, simulating a steady person that can rotate his head to inspect the environment. Secondly, we are assuming that the objects remain static, i.e., their position does not change between saccades.

The proposed project involves the integration of several components (see fig. 3.1). These components will be described and explored throughout this section.

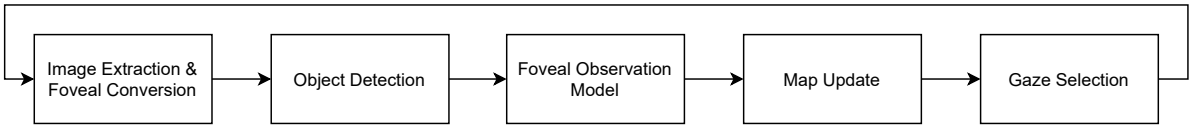


Figure 3.1: Diagram for this work. First we extract an image and convert it to a foveated image, then an Object Detection method is used to compute confidence scores and region proposals. The Foveal Observation Model models the uncertainty of the confidence scores and region proposals with respect to the location on the foveated image to then perform a Map Update where previous information is already stored. Using the information stored on the map, Gaze Selection acquisition functions choose the next best view point, the agent shifts its gaze to that point (Gaze Control) and then a new foveated image is produced, reiterating the process.

The Foveal Conversion will collect the image that we wish to explore and use Almeida [13] and Melício [2] model defined on section 2.1 to foveate the collected Cartesian image with the center of the fovea being the one returned by the Gaze Selection block at each iteration.

3.1 Object Detection & Foveal Observation Model

The foveated image serves as input to the object detection method, which outputs are modeled by the foveal observation model. The whole process is represented graphically in fig. 3.2 and explained in detail on the remaining of this section.

For each image location, given by the global coordinates (x, y), there is a probability of appearing a given object, represented by a probability vector

$$\boldsymbol{p} = [p_1, p_2, ..., p_K]^T \tag{3.1}$$

sampled from a Dirichlet prior with parameters $\boldsymbol{\beta}$ that depend on the environment (on our case we are assuming a uniform $\boldsymbol{\beta}$ generates a uniform $\boldsymbol{p}$, i.e. there is no preference for any class of objects). K is the number of possible object classes.

Given $\boldsymbol{p}$, an object represented by the random variable C is sampled, which is then associated to a bounding box.

Given C and the position on the foveated image, our YOLO detector generates, for each instant t and each object detected l (where $l = 1, ..., L_t$), a multinomial score vector $\boldsymbol{S}_{t,l}$. The score vector contains

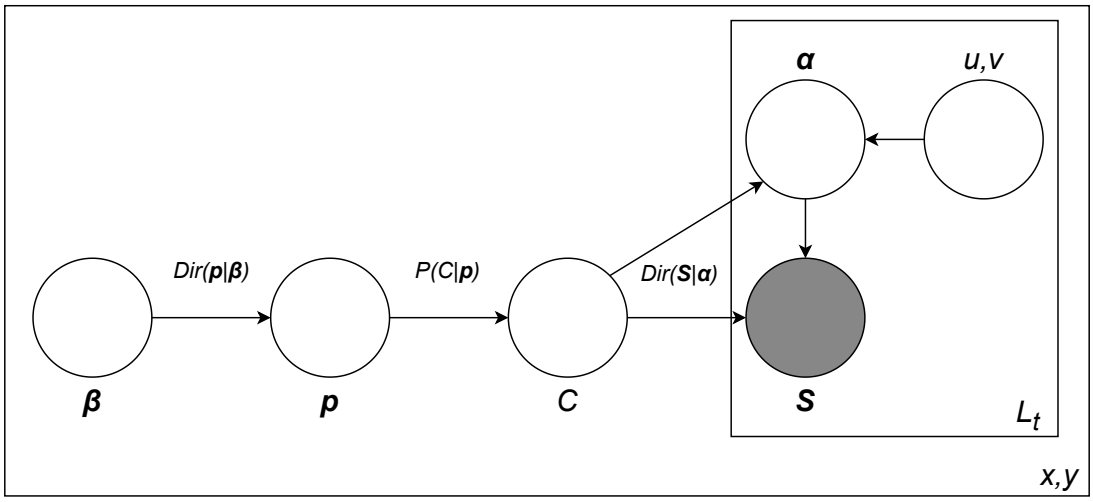


Figure 3.2: Object detection and Foveal Observation Model diagram. Dependencies between variables are represented by the arrows.

the confidence scores of the detection algorithm for each class of object, as enunciated on eq. (1.6).

It's important to note that object detection algorithms, on our case YOLOv3, were built upon the assumption that the input image is Cartesian, meaning that they were not trained to detect the blur imposed by the foveal sensor on the objects. Therefore, the confidence scores might not represent the correct uncertainty that the algorithm has on a given detection.

Moreover, object detection algorithms do not know if the input image is foveated, and consequently, do not know the location of the focal point. That information could be useful to better classify an object affected by the blur on the peripheries, since it would be already expected that the uncertainty and the confusion between object classes would be higher there.

Therefore, the multinomial score vector $\boldsymbol{S}_{t,l}$ has a Dirichlet prior that depends on the location of the image (which is expected to have less entropy near the center of the fovea, and higher entropy on the peripheries).

The parameters of the Dirichlet prior that characterizes the uncertainty on the output of the score vector $\boldsymbol{S}_{t,l}$, can be written as:

$$\boldsymbol{\alpha}_{k,d_{t,l}} = [\alpha_{k,d_{t,l},1}, \alpha_{k,d_{t,l},2}, ..., \alpha_{k,d_{t,l},K}]^T \tag{3.2}$$

that depends on the distance $d_{t,l}$ between the outputted bounding box, and the center of the fovea (x_t, y_t), i.e., depends on the detection local coordinates $(u_{t,l}, v_{t,l})$. Since the fovea may not be evenly distributed on both dimensions (horizontal and vertical), the Mahalanobis distance was used to generalize the approach. The Mahalanobis distance is characterized by a matrix that assigns weights to the

different directions, which can be written, in general, as follows:

$$d_{t,l}(u_{t,l}, v_{t,l}) = \sqrt{[u_{t,l}, v_{t,l}]\Sigma^{-1}[u_{t,l}, v_{t,l}]^T} \tag{3.3}$$

where Σ is the weight matrix (covariance matrix). We are only considering the two-dimensional case, and a fovea that can only expand horizontally or vertically, therefore, the Σ matrix can be expressed as a diagonal matrix

$$\Sigma = \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \tag{3.4}$$

where σ_x and σ_y correspond to the size of the fovea on the horizontal and vertical direction, respectively.

The parameters $\boldsymbol{\alpha}$ will be learned from a supervised training set, with the outputs of the classifier obtained on different observation conditions for each of the classes.

Thus, this training set contains not only the confidence scores of each detection and the associated ground-truth object, as well as the distance between the bounding box and the center of the fovea.

The full training process of the Foveal Observation Model is explained as follows:

1. Every image on the COCO training-set was foveated and served as input to the YOLOv3 object detector.

2. Each detection was then associated to an object whenever the Intersection Over Union (IoU) of the detected bounding box with a ground truth object was greater than 30%. A typical IoU threshold for Cartesian images is 50%, nevertheless, due to the blur imposed by the foveal sensor, the bounding boxes sizes were much less accurate, imposing a reduction on these threshold.

3. A Dirichlet distribution was estimated, by the approach described on section 2.2.2.A, for each object and level of distance to the focal point. 7 different levels of distance were considered and, therefore, each object class is represented by 7 different Dirichlet distributions, depending on its location in relation to the central point of the fovea.

The Foveal Observation Model is then structured in a total of 80x7 different Dirichlet distributions, 80 different object classes k and 7 distance levels $d_{k,l}$ for each class. Thus, following eq. (2.9), whenever a detection $I_{t,l}$ appears, depending on the distance to the focal point, a Dirichlet distribution is chosen for each class of object $k = 1, ..., K$

$$\boldsymbol{S}'_{t,l} = [s'_{t,l,1}, ..., s'_{t,l,K}] = \frac{1}{D}[Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{1,d_{t,l}}), ..., Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{K,d_{t,l}})]^T \tag{3.5}$$

Where D is a normalization factor given by $D = \sum_{k=1}^{K} Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{k,d_{t,l}})$, so that $\sum_{k=1}^{K} s'_{t,l,k} = 1$, and $\boldsymbol{\alpha}_{k,d_{t,l}}$ are the parameters of the pre-trained Dirichlet distribution for the object class k and level of distance from the object to the center of the fovea $d_{t,l}$. These new score vector $\boldsymbol{S}'_{t,l}$ is expected to have

less confusion than the ones outputted by the detector.

Amplifying the confidence of a detection on locations with higher uncertainty is something relevant when searching for objects using this kind of vision, since it alerts the algorithm that there might be something of interest on those areas. On the other hand, amplifying the confidence of a detection might result in an overestimation of the new confidence scores outputted by the Foveal Observation Model. Nevertheless, since we are exploring the scene iteratively (changing the focal point), the constant updates on the information the algorithm has about the world will occlude this overestimation because it will be treated as an outlier. As we will see on the end of this chapter, the Foveal Observation Model also makes it possible to predict the evolution of the map, depending on the next focal point.

3.2 Fusion Model

In order to simplify the exploration, our world corresponds to a single image, where we wish to correctly detect and classify every object on the least number of gaze shifts. Therefore, the information obtained every time the algorithm moves the eyes has to be stored in a map and fused with the information already obtained on previous iterations.

The map itself is a grid put on top of the image (the scene/image representation is displayed on fig. 1.3), with equally sized and uniformly distributed cells (fig. 3.3). For future work, as the scene becomes more complex and we wish to detect objects with a moving camera, other types of structures might be useful to represent the map. The Sensory Ego-Sphere (SES) described in Figueiredo's work [28] is a good example of one of those structures, where the distribution of the map cells is optimized to perform the task in-hands.

Each map cell (x_m, y_m), where each (x_m, y_m) contains the set of pixels (x, y) that are inside the boundaries of the map cell, stores the current belief $\boldsymbol{M}_t(x_m, y_m)$ about what exists on that particular area of the image at a given instant of time t (refer back to eq. (1.8), but instead of $\boldsymbol{M}$ being computed pixel-wise, it is computed cell-wise). Whenever new detections $\mathcal{I}_{t+1}$ appear, the new information is used to update the current knowledge of the corresponding map cells as in eq. (1.8). For our case, this knowledge is represented by the state of the fusion methods (described in section 2.3) on each map cell.

So that the algorithm not only knows if it is probable to exist an object and, if so, which class has a higher probability of being that object, but also the uncertainty associated to that particular map cell, the stored state outputted by the fusion methods require some attention. Let's look at the fusion methods that will be tested on our framework. As proposed by Kaplan [29], the fusion results of both the sum rule and "Kaplan's approach" can be mapped onto a Dirichlet distribution with parameters $\boldsymbol{\beta}$. Therefore, storing on the map these parameters $\boldsymbol{\beta}_t^{x_m, y_m}$, for the current instant t, for each map cell (x_m, y_m) will

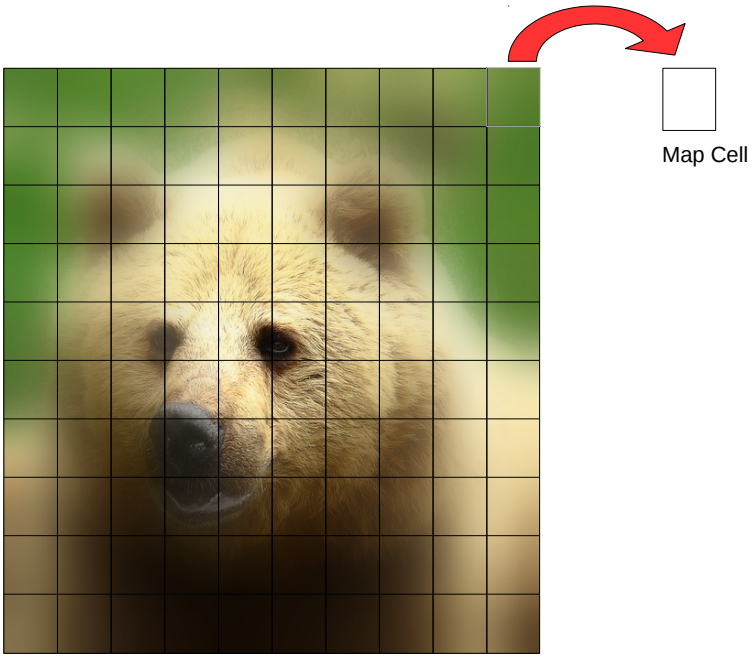


Figure 3.3: Representation of the map by a 10x10 grid of map cells on top of a foveated image.

allow one to not only extract the expected probability of each class of objects k (for $k = 1, ..., K$) on that cell

$$p_{t,k}^{x_m,y_m} = \frac{\beta_{t,k}^{x_m,y_m}}{\sum_{j=1}^{K} \beta_{t,j}^{x_m,y_m}} \tag{3.6}$$

where

$$\boldsymbol{\beta}_t^{x_m,y_m} = [\beta_{t,1}^{x_m,y_m}, ..., \beta_{t,K}^{x_m,y_m}]^T \tag{3.7}$$

and

$$\boldsymbol{p}_t^{x_m,y_m} = [p_{t,1}^{x_m,y_m}, ..., p_{t,K}^{x_m,y_m}]^T \tag{3.8}$$

but also to have more information about the uncertainty of these expected values, since the parameters $\boldsymbol{\beta}$ of the Dirichlet distribution contain more information than the categorical distribution $\boldsymbol{p}$.

Updating the map information with the sum rule can then be done with

$$\beta_{t+1,k}^{x_m,y_m} = \beta_{t,k}^{x_m,y_m} + \frac{l_{t,k}^{x_m,y_m}}{\sum_{j=1}^{K} l_{t,j}^{x_m,y_m}} \tag{3.9}$$

which refers back to eq. (2.44), where $l_{t,k}^{x_m,y_m}$ is the likelihood of the k-th class outputted from the classifier at instant t, for the map cell (x_m, y_m). Using Kaplan's approach, updating the map information can

be done as follows

$$\beta_{t+1,k}^{x_m,y_m} = \frac{\beta_{t,k}^{x_m,y_m}\left(1+\frac{l_{t,k}^{x_m,y_m}}{\sum_{j=1}^{K}\beta_{t,j}^{x_m,y_m}l_{t,j}^{x_m,y_m}}\right)}{1+\frac{\min_j l_{t,j}^{x_m,y_m}}{\sum_{j=1}^{K}\beta_{t,j}^{x_m,y_m}l_{t,j}^{x_m,y_m}}} \tag{3.10}$$

which refers back to eq. (2.48). Since we are updating iteratively the parameters, an initial state has to be defined. This initial state has of course to represent a uniform distribution (all Dirichlet parameter equal), so that there are maximum uncertainty on each map cell. Following Kaplan's [29] work we defined the initial parameters for each fusion algorithm and map cell as $\beta_k^0 = 0.5$ for $k = 1, .., K$.

The other fusion approach that will be tested on our framework, the Naïve Bayes (see section 2.3.1), can not be modelled by a Dirichlet (this is shown later on eq. (3.23)) and therefore each map cell will store the categorical distribution $\boldsymbol{p}$ outputted with the Naïve Bayes approach, instead of the $\boldsymbol{\beta}$ parameters as in the other approaches. The Naïve Bayes approach will be explained later on this chapter.

For our specific case, the confidence scores outputted by the YOLO algorithm $\boldsymbol{S}_{t,l}$ would be used as the likelihood. Nevertheless, for each map cell (x_m, y_m) there might be 0, 1 or more detections at a given instant of time t, meaning that for that instant of time each fusion process for the map cell (x_m, y_m) is repeated for every detection belonging to the set $\mathcal{I}_t^{x_m,y_m}$, where $\mathcal{I}_t^{x_m,y_m}$ is the set of detections at instant t which bounding boxes intersect with the map cell (x_m, y_m),

$$\mathcal{I}_t^{x_m,y_m} = \{(\boldsymbol{B}_{f_{t,l}(x_m,y_m)}, \boldsymbol{S}_{f_{t,l}(x_m,y_m)})\} \tag{3.11}$$

Ignoring the time separation, one can generalize to $i = t + l$, being $S_i^{x_m,y_m}$ the i-th confidence score outputted by the YOLOv3 detector, which bounding box intersects with (x_m, y_m). Thus, starting on the sum rule, eq. (3.9) can be re-written as

$$\beta_{i+1,k}^{x_m,y_m} = \beta_{i,k}^{x_m,y_m} + \frac{l_{i,k}^{x_m,y_m}}{\sum_{j=1}^{K} l_{i,j}^{x_m,y_m}} \tag{3.12}$$

where $l_{i,k}^{x_m,y_m} = s_{i,k}^{x_m,y_m}$ and, for Kaplan's approach, eq. (3.10) can be re-written as

$$\beta_{i+1,k}^{x_m,y_m} = \frac{\beta_{i,k}^{x_m,y_m}\left(1+\frac{l_{i,k}^{x_m,y_m}}{\sum_{j=1}^{K}\beta_{i,j}^{x_m,y_m}l_{i,j}^{x_m,y_m}}\right)}{1+\frac{\min_j l_{i,j}^{x_m,y_m}}{\sum_{j=1}^{K}\beta_{i,j}^{x_m,y_m}l_{i,j}^{x_m,y_m}}} \tag{3.13}$$

3.2.1 Background Class

Since a map cell may or may not contain an object (or part of it), the likelihood vector should also contain the probability of a given cell not representing any object. Nevertheless, the YOLO algorithm only assigns confidence scores to objects, ignoring the background. In order to solve this issue, a "background" class was appended to each score vector, just as it was a confidence score outputted by the object detector.

The confidence score of the background was chosen to be the value that the detector would output on every class in the highest uncertainty case, the uniform distribution case. Therefore, the new likelihood vector is given by

$$l_{i,k}^{x_m,y_m} = \begin{cases} \frac{1}{Q} l_{i,k}^{x_m,y_m}, & k = 1, ..., K \\ \frac{1}{K+1}, & k = K+1 \end{cases} \tag{3.14}$$

with

$$l_{i,k}^{x_m,y_m} \geq 0, \forall_k \quad \wedge \quad \sum_{j=1}^{K+1} l_{i,j}^{x_m,y_m} = 1 \tag{3.15}$$

where $Q = \frac{K+1}{K}$ is the normalization factor.

3.2.2 Extension to Integrate the Observation Model

Using Kaplan fusion method (eq. (2.48)) with simply the output of the detector algorithm would ignore the knowledge that we have about the location of the objects in relation to the center of the fovea, ignoring, therefore, the confusion that might be imposed by the gradually increasing blur over the peripheries. Thus, in order to analyse if this knowledge can improve the performance of a scene exploration, the full observation model has to be considered on a modified version of Kaplan's method.

Considering the full observation model, following eq. (2.9), the likelihoods for the i-th classification given its output confidence scores $\boldsymbol{S}_i^{x_m,y_m}$, are given by

$$l_{i,k}^{x_m,y_m} = \begin{cases} \frac{1}{Q.D} Dir(\boldsymbol{S}_i^{x_m,y_m}, \alpha^{k,d_i}), & k = 1, ..., K \\ \frac{1}{K+1}, & k = K+1 \end{cases} \tag{3.16}$$

with

$$l_{i,k}^{x_m,y_m} \geq 0, \forall_j \quad \wedge \quad \sum_{j=1}^{K+1} l_{i,j}^{x_m,y_m} = 1 \tag{3.17}$$

where the likelihood of the background class $l_{i,k+1}^{x_m,y_m}$ follows the same logic as before.

The Naïve Bayes approach requires a different attention, when using the foveal observation model. Let's apply it to our model.

Being $\boldsymbol{p}_0^{x_m,y_m}$ an initial estimation of $\boldsymbol{p}^{x_m,y_m}$ and $\boldsymbol{p}_i^{x_m,y_m}$ an estimation of $\boldsymbol{p}^{x_m,y_m}$ after observing the output of i classifiers, for the map cell (x_m, y_m)

$$\boldsymbol{p}_i^{x_m,y_m} = [p_{i,1}^{x_m,y_m}, p_{i,2}^{x_m,y_m}, ..., p_{i,K}^{x_m,y_m}]^T \tag{3.18}$$

with

$$\boldsymbol{p}_{i,k}^{x_m,y_m} = p(C_{x_m,y_m} = c_k | \boldsymbol{S}_{1:i}^{x_m,y_m}) \tag{3.19}$$

by Bayes Law:

$$p(C_{x_m,y_m} = c_k|\boldsymbol{S}_{1:i}^{x_m,y_m}) \propto p(\boldsymbol{S}_i^{x_m,y_m}|C_{x_m,y_m} = c_k, \boldsymbol{S}_{1:i-1}^{x_m,y_m})p(C_{x_m,y_m} = c_k|\boldsymbol{S}_{1:i-1}^{x_m,y_m}) \tag{3.20}$$

and assuming independence of the observations:

$$p_{i,k}^{x_m,y_m} \propto p(\boldsymbol{S}_i^{x_m,y_m}|C_{x_m,y_m} = c_k)p_{i-1,k}^{x_m,y_m} \tag{3.21}$$

where

$$p(\boldsymbol{S}_i^{x_m,y_m}|C_{x_m,y_m} = c_k) = Dir(\boldsymbol{S}_i^{x_m,y_m}, \boldsymbol{\alpha}^{k,d_i}) \tag{3.22}$$

and $Dir(., \boldsymbol{\alpha}^{k,d_i})$ was learned in a training phase. So

$$p_{i,k}^{x_m,y_m} = \frac{Dir(\boldsymbol{S}_i^{x_m,y_m}, \boldsymbol{\alpha}^{k,d_i})p_{i-1,k}^{x_m,y_m}}{\sum_{j=1}^{K} Dir(\boldsymbol{S}_i^{x_m,y_m}, \boldsymbol{\alpha}^{j,d_i})p_{i-1,j}^{x_m,y_m}} \tag{3.23}$$

Note that the distribution of $p_{i,k}^{x_m,y_m}$ is not Dirichlet (or any other closed form distribution). So, instead of storing on each cell the resulting distribution, one can store directly the parameters $\boldsymbol{p}$, and use that as the state of the map for the Naïve Bayes fusion algorithm. The results of the Naïve Bayes will be compared with the sum rule (eq. (3.12)), and Kaplan fusion approach (eq. (3.13)), where the parameters of a Dirichlet distribution that tries to fit the posterior $\boldsymbol{p}$ are updated at each iteration, and a "modified Kaplan approach", where the Kaplan update of eq. (3.13) is modified to use the observation model instead of the direct observations (eq. (3.16)). These four fusion methods will be tested under the same conditions to then compare classification performances.

3.2.3 Update Normalization

Except for the Naïve Bayes, all other fusion methods can be characterized by a Dirichlet distribution. Whenever an object is detected, the parameters of the multinomial opinion are incremented, meaning that the more bounding boxes are detected, the higher the parameters of the Dirichlet will be.

Usually, higher parameters correspond to a stronger confidence on the current multinomial opinion. So, an object with more bounding boxes associated, can have a lower uncertainty than an object with less detections, even if the scores are lower. This may be an issue since YOLOv3 creates multiple overlapping boxes, which are then removed by a Non-Maximum Supression (NMS) criteria.

The YOLOv3 does NMS by imposing a threshold on the IoU of bounding boxes where the higher score corresponds to the same object. As one can see on fig. 3.4, the IoU is a coefficient between areas, making it more sensible for smaller objects, since, for a small area of union, small differences on the area of overlap impose large changes on the IoU value.

This observation, in practice, makes it easier for the detection algorithm to assign more boxes to

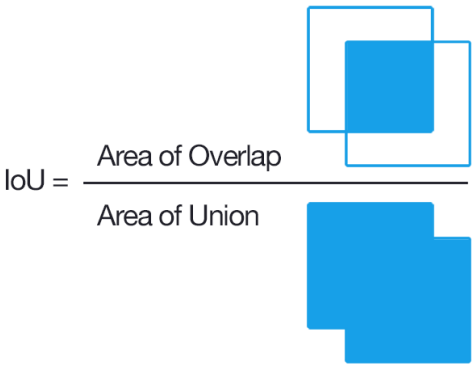


Figure 3.4: IoU formula with explanatory images. Adapted from [6]

smaller objects. The number of boxes assigned to a given object is even bigger since we are dealing with foveated images, due to the increased uncertainty imposed by the image distortion on the peripheries. Therefore, it might be necessary to deal with this problem, by normalizing the number of updates on each map cell.

As defined before, $\mathcal{I}_t^{x_m,y_m}$ is a subset of all detections outputs which bounding boxes overlap with the map cell (x_m, y_m) at iteration t (eq. (1.4)), and $\boldsymbol{S}_{t,j}^{x_m,y_m}$ the output score vector of the j-th classifier contained on that subset.

$$\boldsymbol{S}_{t,j}^{x_m,y_m} = [s_{t,j,1}^{x_m,y_m}, ..., s_{t,j,K}^{x_m,y_m}]^T \tag{3.24}$$

As said before, the subset $\mathcal{I}_t^{x_m,y_m}$ may be empty, may contain one detection ($j = 1$), or be composed of $n_{x_m,y_m,t}$ detections ($j = 1, ..., n_{x_m,y_m,t}$).

The following three methods for normalizing the update increments will be tested and compared against each others:

- **Average of Detections** - for each iteration t, each map cell (x_m, y_m) is updated with the average of the resulting outputs of all bounding boxes detected on that cell, on that iteration. Each dimension $k = 1, ..., K$ of this new "average box" $\overline{\boldsymbol{b}}^{x_m,y_m,t}$ is computed as follows:

$$\overline{b}_k^{x_m,y_m,t} = \frac{1}{n^{x_m,y_m,t}} \sum_{j=1}^{n^{x_m,y_m,t}} s_{t,j,k}^{x_m,y_m} \tag{3.25}$$

- **Most Probable Box Selection** - Each map cell (x_m, y_m), at each iteration t, is only updated with the most probable detection $\boldsymbol{b}_{\boldsymbol{max}}^{xyt}$ (bounding box with maximum higher score).

$$\boldsymbol{b}_{\boldsymbol{max}}^{x_m,y_m,t} = \boldsymbol{S}_{t,j^*}^{x_m,y_m}, \qquad \text{where} \quad j^* = \underset{j}{\operatorname{argmax}} \left\{ \max_k s_{t,j,k}^{x_m,y_m} \right\} \tag{3.26}$$

- **Artificial Increments** - At each iteration t, for each map cell (x_m, y_m) there are fabricated $m^{x_m,y_m,t}$

artificial updates in order to have the same number of updates on every map cell. The distribution of these artificial updates correspond to the average value of the updates for a given cell (x_m, y_m) at the corresponding iteration t.

The number of artificial updates at iteration t at the map cell (x_m, y_m) is given by:

$$m^{x_m,y_m,t} = \max_{a,b} n^{a,b,t} - n^{x_m,y_m,t} \tag{3.27}$$

3.3 Active Perception - Gaze Selection

The Gaze selection is the ”block” responsible for dealing with the problem of ”where to look next”. There are several active perception methods to choose the next best view point based on the values associated to each point on the map and their uncertainty. These decision methods are typically based on the maximization of a function that depends on the uncertainty, denominated *acquisition functions*. Figueiredo tested some methods on his work on active perception [28] (see section 2.4.1), nevertheless his methods assume that it is possible to measure or predict the reward of each action, and how close is the algorithm from the objective.

For the purpose of this work, finding and correctly classifying the most objects on one image in the least number of gaze shifts, it is not possible to measure or predict the reward of each action because we can not measure how close we are from the objective. First of all, the exploratory algorithm never knows the ground truth information, therefore it does not know if it is correctly classifying the objects in the scene or not, neither does it know how many objects are left to find. Secondly, the uncertainty associated to each categorical distribution does not correspond to the uncertainty on the classification of a given object, but rather the amount of confusion on that particular map point, since the algorithm never knows if it has found an object, and if that object is correctly classified, the algorithm just knows the confidence that it has on a given object of a certain class being on a certain map cell.

Therefore, the best we can do is to predict what is the next focal point that might maximize the reduction of the confusion on the map. The confusion on the map can be represented by different metrics, three of which will be considered in this work:

- **KL Divergence** (D_{KL}) - Measures how different one probability distribution Q is from another reference distribution R. [35]

$$D_{KL}(Q||R) = \int_{-\infty}^{+\infty} q(x) \log\left(\frac{q(x)}{r(x)}\right) dx. \tag{3.28}$$

  We will take advantage of the output of the fusion algorithms being Dirichlets' distributions, to measure the KL Divergence between the output of the fusion methods and a Dirichlet distribution

with uniform parameters, in this case the initial state of each fusion algorithm ($\beta_k^0 = 0.5$ for all $k = 1, ..., K$). This way, before the first detections, the parameters will be equal, and, therefore, the KL Divergence will be 0, meaning maximum uncertainty. As the map cells are updated, the KL Divergence gradually increases.

So, if we consider that the state of a map cell (x_m, y_m) at a given iteration t is represented by the Dirichlet parameters $\boldsymbol{\beta}_t^{x_m,y_m}$, and the reference distribution is given by $\boldsymbol{\beta}^0$, for $k = 1, ..., K$, then, as derived on Kurt's article [36], the KL divergence can be computed as follows

$$D_{KL}^{x_m,y_m,t} = \log\Gamma\left(\beta_{t,0}^{x_m,y_m}\right) - \sum_{k=1}^{K}\log\Gamma\left(\beta_{t,k}^{x_m,y_m}\right) - \log\Gamma\left(\sum_{k=1}^{K}\beta_k^0\right) + \sum_{k=1}^{K}\log\Gamma\left(\beta_k^0\right) + \\ + \sum_{k=1}^{K}\left(\beta_{t,k}^{x_m,y_m} - \beta_k^0\right)\left(\Psi\left(\beta_{t,k}^{x_m,y_m}\right) - \Psi\left(\beta_{t,0}^{x_m,y_m}\right)\right) \quad (3.29)$$

where $\beta_{t,0}^{x_m,y_m} = \sum_{k=1}^{K}\beta_{t,k}^{x_m,y_m}$. [36]

- **Classification Entropy** - is related to the amount of uncertainty or confusion on a classification (on an array of scores). Considering the vector of parameters $\boldsymbol{p}_t^{x_m,y_m}$ that is given by the expected values of the state of a map cell with coordinates (x_m, y_m) at iteration t, the classification entropy is given by

$$Entropy^{x_m,y_m,t} = -\sum_{k=1}^{K} p_{t,k}^{x_m,y_m}\log p_{t,k}^{x_m,y_m} \quad (3.30)$$

- **Difference between Two Peaks** (D_{2Peaks}) - represents the uncertainty on the classification of a given object by checking the difference between the two highest confidence scores. The reasoning behind this metric is, if the two top scores are close to each other, the confusion between the corresponding classes is high and therefore the uncertainty on that classification is also predicted to be high. We wish to measure this confusion on a given map cell (x_m, y_m), at a given iteration t, which state is represented by the vector of parameters $\boldsymbol{p}_t^{x_m,y_m}$

$$D_{2Peaks}^{x_m,y_m,t} = \max_k p_{t,k}^{x_m,y_m} - \max_{k\setminus\underset{j}{\operatorname{argmax}}\left\{p_{t,j}^{x_m,y_m}\right\}} p_{t,k}^{x_m,y_m} \quad (3.31)$$

Having metrics that can measure the amount of uncertainty/confusion on each map cell, the *acquisition functions* have then to predict the global (average) uncertainty of the map if the focal point changed to another pixel of the image. The *acquisition functions* considered in this work aim to find the cell that minimizes the average map uncertainty with each of the the metrics above.

For the KL Divergence, one aims to maximize the expected improvement on the average KL Divergence of the map. Let's consider a $X \times Y$ map, the map cell for the expected next best view point (x_m^*, y_m^*), at iteration t, using this acquisition function, is given by:

$$(x_m^*, y_m^*) = \underset{i,j}{\operatorname{argmax}} \left\{ \sum_{x_m=1}^{X} \sum_{y_m=1}^{Y} \left(E^{ij} \left\{ D_{KL}^{x_m, y_m, (t+1)} \right\} \right) \right\} \tag{3.32}$$

where $E^{ij}\{.\}$ corresponds to the expected value for a fovea centered on (i, j).

Using the Classification Entropy metric, the aim is the same as the KL divergence, but instead of maximizing the KL divergence, one wishes to minimize the entropy. Thus, following the same notation as above, using this acquisition function:

$$(x_m^*, y_m^*) = \underset{i,j}{\operatorname{argmax}} \left\{ - \sum_{x_m=1}^{X} \sum_{y_m=1}^{Y} E^{ij} \left\{ Entropy^{x_m, y_m, (t+1)} \right\} \right\} \tag{3.33}$$

For the Difference between Two Peaks, one wishes to maximize the absolute gain of this metric. This measure will tell us how much information is predicted to be gained at each iteration, a positive gain means a reduction on the confusion whilst a negative gain means that the algorithm is changing the opinion of a given object, or that there are overlapping objects in one cell.

Following the same notation as before:

$$(x_m^*, y_m^*) = \underset{i,j}{\operatorname{argmax}} \left\{ \underset{x_m, y_m}{max} \left\{ \left| D_{2Peaks}^{x_m, y_m, t} - E^{ij} \left\{ D_{2Peaks}^{x_m, y_m, (t+1)} \right\} \right| \right\} \right\} \tag{3.34}$$

Predicting the resulting detections and updates of the map, if the algorithm shifts the gaze to a certain position, is possible due to knowing that the fovea will have a higher resolution than the peripheries. Therefore, there will be more detections and with less uncertainty as close as the objects are from the focal point. Taking advantage of the distance to the center of the fovea to try to predict which objects are where is exactly what the Foveal Observation Model does, and that's why we hope to contribute with a useful tool to implement this type of active perception process.

Let's start by remembering that state of the map is represented on the form of the $\boldsymbol{\beta}_t^{x_m, y_m}$ parameters for the map cell (x_m, y_m) at instant t. The expected value of each of the K classes is given by the categorical distribution $\boldsymbol{p}_t^{x_m, y_m}$ through eq. (3.6). These expected values are the current belief about the world, and, therefore, can be used as an estimation to what the object detector would output on the next time instant $(t + 1)$, after the gaze shift. Now, this expected values do not depend on the position of the fovea, only on the state of each map cell, therefore, are not useful to choose the next point where to shift the gaze as they are. Nevertheless, modeling the expected values with the foveal observation model would allow us to estimate the evolution of the map depending on the next focal point. The estimated likelihood of the k-th class on the cell map (x_m, y_m) for the possible next focal point (x'_m, y'_m) can then

be predicted as follows

$$E^{x'_m,y'_m}\left\{\hat{l}^{x_m,y_m}_{t+1,k}\right\} = \begin{cases} \frac{1}{Q.D} Dir\left(\boldsymbol{p}_t^{x_m,y_m}, \boldsymbol{\alpha}^{k,d_{f_{t,l}(x'_m,y'_m)}}\right), & k = 1,...,K \\ \frac{1}{K+1}, & k = K+1 \end{cases} \tag{3.35}$$

The expected value of the metrics defined on eq. (3.32), eq. (3.33) and eq. (3.34) are then computed by simulating map updates (e.g., using eq. (3.13)). The center that presents better expected updates will be the one chosen to shift our gaze to, depending on the chosen acquisition function and fusion method.

After shifting the gaze, new information is extracted and the whole process is repeated.

4

Experiments & Results

Contents

4.1 Foveal Observation Model Validity 51

4.2 Map Update 52

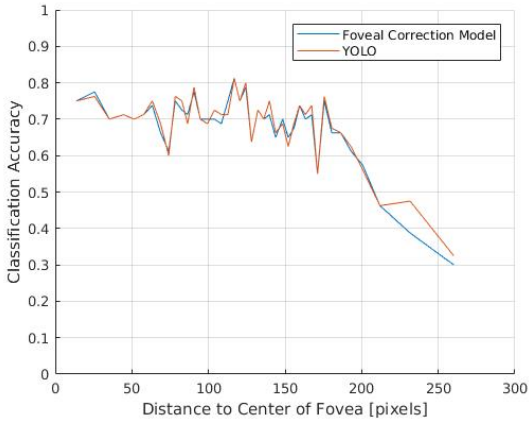
4.3 Next Best View Point 56

4.1 Foveal Observation Model Validity

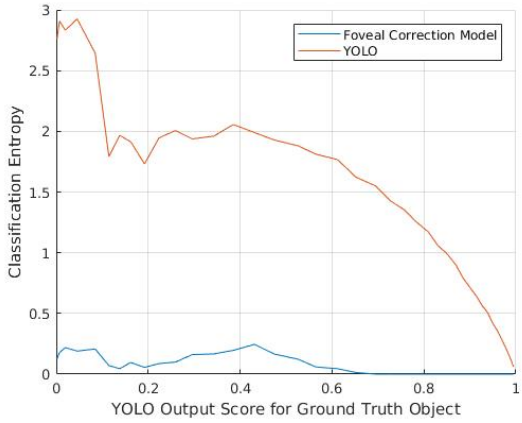
The foveal observation step is supposed to take advantage of the relative position of the objects to the center of the fovea, in order to model the uncertainty imposed by the blur on the peripheries on the output scores of the detected objects. This way, in order to check the validity of this model, it is important to compare the performance of the classification task with and without the foveal observation model.

For this comparison, several random images were taken (from the COCO dataset) and foveated using randomly chosen focal points. The process is explained as follows:

1. Every foveated images served as input to the YOLOv3 algorithm.
2. The classification outputs which had a bounding box with an IoU greater than 30% with the ground-truth information for that image were considered as corresponding to a certain ground-truth object.
3. The classification outputs were then modelled by the foveal observation model using the pre-trained distributions (taking into consideration the distance of each detection to the center of the fovea, eq. (2.9)).
4. And the results were grouped using different metrics, such as the distance to the fovea and the scores outputted by the detector. An average was computed in order to better represent the performances.



(a) Accuracy comparison.



(b) Entropy comparison.

Figure 4.1: Performance comparison between the scores outputted by the foveal observation model (blue) and the scores outputted directly from the object detection algorithm (without being modeled by the foveal observation model) (red).

On fig. 4.1 one has the comparison of the classification performance on foveated images when using the information of the distance between the detection and the fovea, with the performance of the object detector algorithm without using the foveal observation model.

The accuracy lines on fig. 4.1a are very similar, meaning that modelling the detections with the observation model does not impose a drop on performance, but it reduces the uncertainty that the algorithm has on the detection, as one can see on fig. 4.1b. In this case, the classification entropy was used to measure the uncertainty, and, considering a generic score vector $\boldsymbol{S}$, the classification entropy is given as follows

$$entropy = -\sum_{k=1}^{K} s_k \log s_k \tag{4.1}$$

where

$$\boldsymbol{S} = [s_1, ..., s_K] \tag{4.2}$$

and

$$\sum_{k=1}^{K} s_k = 1, \quad \wedge \quad s_k \geq 0 \; \forall_k \tag{4.3}$$

If one class has a much higher confidence than all the other object classes, the entropy will be almost zero, which is exactly what happens for most cases when using the foveal observation model (fig. 4.1b). Basically, the model is trying to combine the information of the scores outputted by the object detector with the distance of the detected bounding boxes to the center of the fovea, to provide a score vector with a higher degree of confidence.

In more detail, fig. 4.1b shows that low confidence scores directly outputted by the detector (*e.g.* detections affected by the blur on the peripheries) have a high degree of entropy, but when these scores are modelled by the observation model, the entropy of the confidence score vector is much lower. The model tries to find the object, for that distance, that better fits the distribution of the scores, even if there is a big confusion among some of the classes, to present a more certain classification. This can also be analysed on fig. 4.2, where the confidence of the classification is amplified by the foveal observation model.

In conclusion, the results of this experiment prove the validity of the foveal observation model, since it does not reduce the performance of the object detection algorithm, but it does amplifies the confidence on the classification. This amplification of the confidence on the classification does not prove to be an improvement on a 1-step classification approach (as seen on fig. 4.1a), but will be useful for the multi-step classification approach that we are taking.

4.2 Map Update

Having analysed the performance and validity of the foveal observation model, it's now time to check the performance when using this model to update the information on the map, at each iteration.

Four different fusion methods were implemented to update, at each iteration, the current state of each

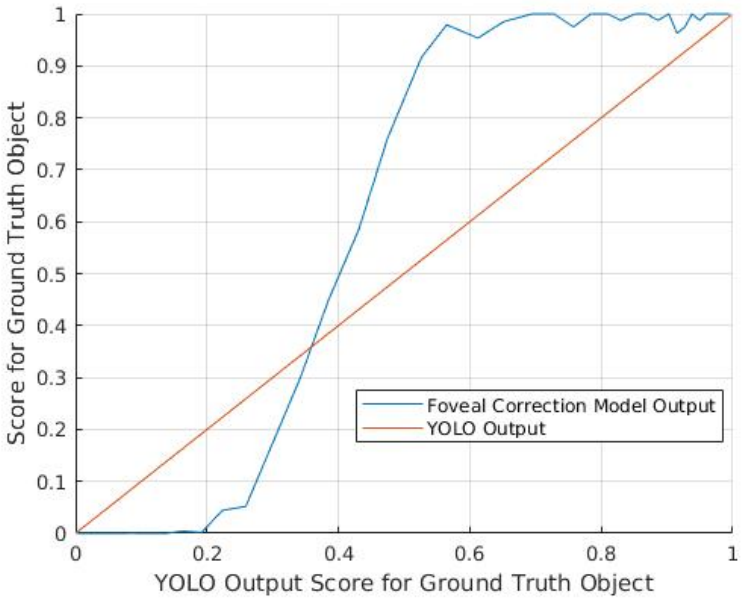


Figure 4.2: Scores comparison for the ground-truth class outputted by the foveal observation model and the object detection algorithm, as a function of the scores outputted directly from the object detector

map cell - Naïve Bayes (eq. (3.23)), Kaplan Update (eq. (3.13)), Modified Kaplan Update (eq. (3.16)), and Sum Rule (eq. (3.12)). On this section, the performance of each method will be tested when fusing detections on foveated images.

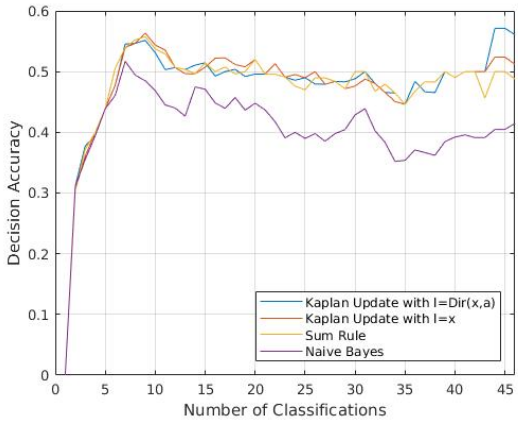
4.2.1 Overall Performance

To compare the overall performance of each method, an experiment was conducted to analyse the evolution of the accuracy, expected value of the ground-truth class and uncertainty of the classification by the KL divergence (eq. (3.28)) as new bounding boxes arrive.

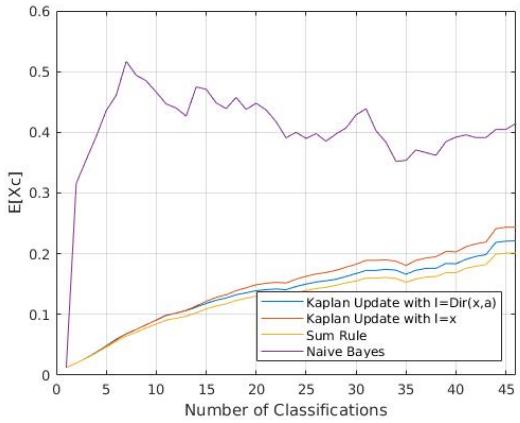
On this experiment, a random exploration approach will be used instead of an active one, allowing to isolate the performance of the fusion methods without considering the gaze selection step.

The following steps were taken:

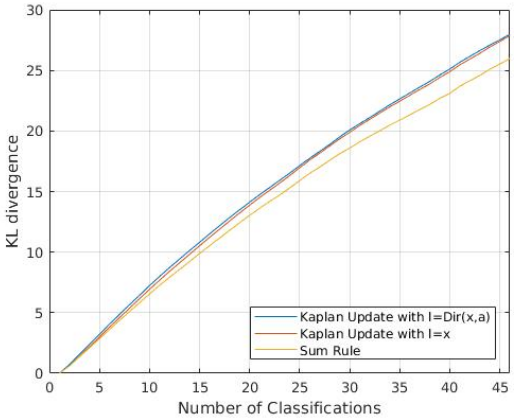
1. A set of 50 images was randomly chosen from the COCO database.
2. Each image was foveated 10 times sequentially, with different focal points (also randomly chosen).
3. Object detection (using YOLOv3) was performed on every foveated image. Only bounding boxes with an IoU with a certain ground-truth object greater than 30% were considered.
4. For each image on the set, every considered detection was fused iteratively using all 4 fusion methods (Naïve Bayes, Kaplan Update, Modified Kaplan Update, and Sum Rule).



(a) Average Accuracy evolution.



(b) Evolution of the Average Expected Value for the ground-truth class.



(c) Average KL divergence evolution.

Figure 4.3: Performance metrics of 4 different fusion algorithms, analysed time-wise as new bounding boxes are detected.

The values used in the plots to evaluate this experiment correspond to the average of the accuracy, expected value, and uncertainty metrics over all 50 images, on an attempt to reduce the impact of outliers, such as wrongly matching a detection with a ground-truth bounding box.

On fig. 4.3a one can see how the average accuracy evolves as new bounding boxes are detected (each bounding box corresponds to one classification). The accuracy is considered 1 if the ground-truth class corresponds to the most probable class (higher expected value) and 0 otherwise. It is possible to note that every algorithm achieves a similar performance on the accuracy, except for the Naïve Bayes, where the performance is lower. Once we analyse the Expected value, the meaning of this difference will become clearer.

The average expected value (fig. 4.3b) that the Naïve Bayes fusion method achieves has, in average, a much higher value than the other methods. If we couple the results of the accuracy and the expected value for the Naïve Bayes, the curves are almost the same. The explanation becomes clear if instead

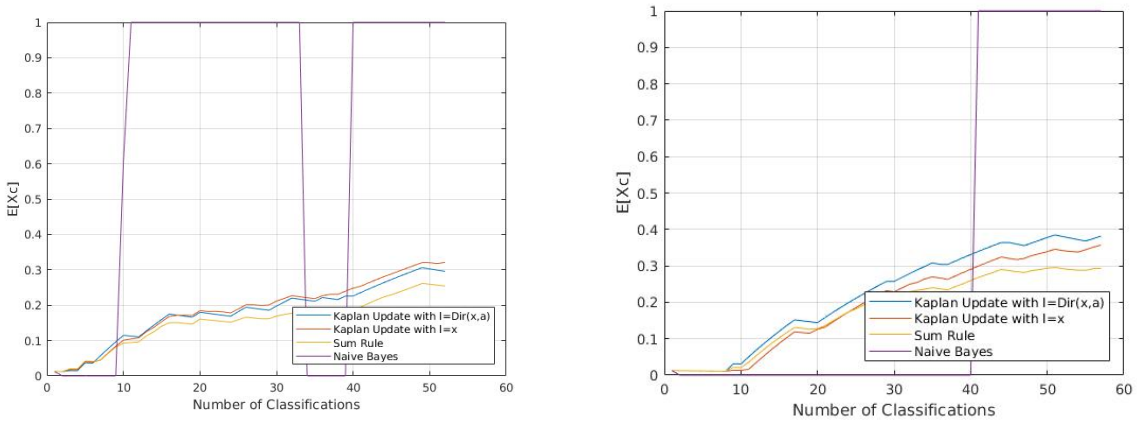


Figure 4.4: Two examples of the evolution of the expected value, for two different images, upon fusing the resulting bounding boxes of each images when foveated in different points, using the 4 algorithms mentioned on this section.

of averaging the results for each image, we present them for a single image, as in fig. 4.4. As one can see, the Naïve Bayes fusion algorithm only outputs expected values/probabilities for the ground-truth class of 0 or 1. This is due to having 80 different classes of objects (80 dimensions for the Dirichlet distributions of the foveal observation model). Remembering the pdf for the Dirichlet distribution of the foveal observation model (as in eq. (3.5)), the expected values/probabilities for the ground-truth class k for the output score vector of the foveal observation model are given by

$$s'_{t,l,k} = \frac{1}{D} Dir(\boldsymbol{S}_{t,l}|\boldsymbol{\alpha}_{k,d_{t,l}}) = \frac{1}{D}\frac{1}{B(\boldsymbol{\alpha}_{k,d_{t,l}})}\prod_{j=1}^{K} s_{t,l,k}^{\alpha_{k,d_{t,l},j}-1} \tag{4.4}$$

where

$$B(\boldsymbol{\alpha}_{k,d_{t,l}}) = \frac{\prod_{j=1}^{K}\Gamma(\alpha_{k,d_{t,l},j})}{\Gamma(\sum_{j=1}^{K}\alpha_{k,d_{t,l},j})}, \quad \boldsymbol{\alpha}_{k,d_{t,l}} = (\alpha_{k,d_{t,l},1}, ..., \alpha_{k,d_{t,l},K}). \tag{4.5}$$

and

$$\Gamma(\alpha_{k,d_{t,l},j}) = \int_0^\infty x^{\alpha_{k,d_{t,l},j}-1} e^{-x} dx \tag{4.6}$$

which is the factorial function of $\alpha_{k,d_{t,l},j} - 1$ for real positive numbers. For high dimensional data (like the 80 classes we have), $\Gamma(\sum_{j=1}^{K}\alpha_{k,d_{t,l},j} \gg \prod_{j=1}^{K}\Gamma(\alpha_{k,d_{t,l},j})$, thus, $B(\boldsymbol{\alpha}_{k,d_{t,l}}) \ll 0$, and consequently the Dirichlet pdf will take values with different order of magnitude, greatly depending on the $\boldsymbol{\alpha}$ parameters. Therefore, when comparing the resulting Dirichlet pdf for the different classes k (using different parameters $\boldsymbol{\alpha}_{k,d_{t,l}}$ to compute each $s'_{t,l,k}$ (eq. (4.4))) in order to model a given score vector $\boldsymbol{S}_{t,l}$ will result in a certain class having much bigger values than the other classes, explaining the Naïve Bayes results. The fact mentioned above also implies that the uncertainty when updating the map using the Naïve Bayes algorithm is always zero (the categorical distribution $\boldsymbol{p}$ has one class with probability 1, and all the others

with probability 0). That's why the uncertainty for the outputs of the Naïve Bayes fusion method is not represented on fig. 4.3c.

As for the other algorithms, both the expected value and the uncertainty have a positive evolution as new bounding boxes are fused, where the Kaplan updates present slightly better results, as in Kaplan experiments [29]. Since the state outputted by these fusion methods is a Dirichlet distribution, the uncertainty was measured by the KL divergence.

4.2.2 Fusion Example

On this section, we will present an example of the performance of the fusion methods when updating the map information as new detections appear whenever the focal point changes. This experiment will not only check how the performance evolves iteration-wise, but also spacial-wise.

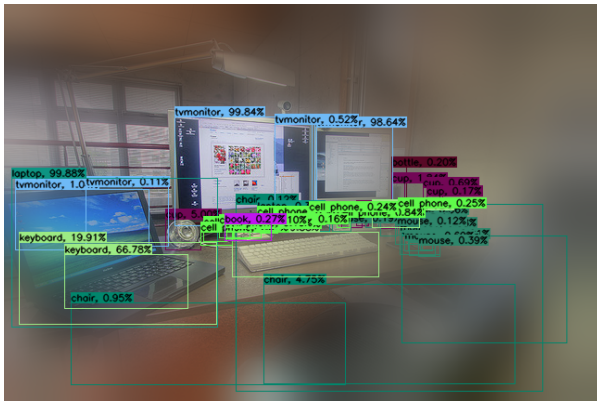
For this experiment a 50x50 grid map is put over the image, where each map cell contains the distribution that represents the state of each algorithm on that specific location. With that distribution, it is possible to evaluate the expected value of each class, on each cell.

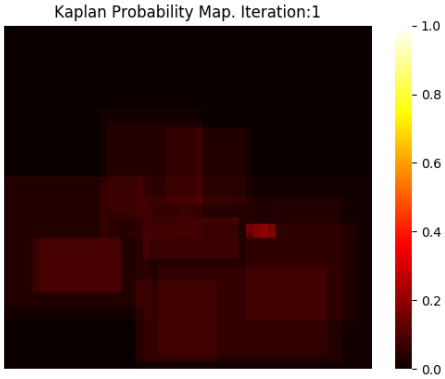
At each iteration a new focal point is randomly chosen, and the new foveated images (fig. 4.5 left) serve as input for the object detection algorithm (YOLOv3). The YOLOv3 algorithm then outputs several bounding boxes with the associated confidence scores, and, depending on the location of each bounding box, the corresponding map cells are updated, using the fusion methods.

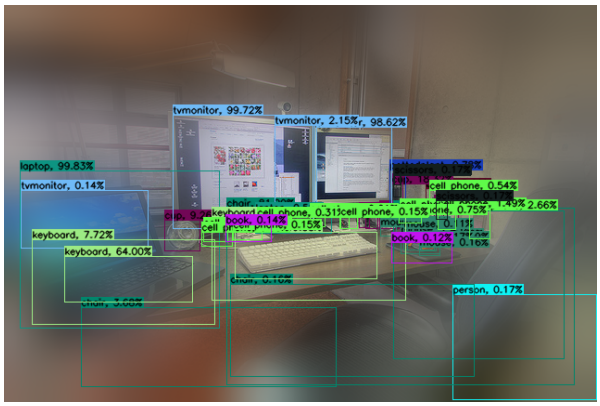
The heat-map on the right side of fig. 4.5 displays the expected value, at a given iteration, of the ground-truth class(es) of each map cell. Figure 4.5 represents the evolution of these expected values for the ground-truth class(es) throughout 10 iterations. Notice that the algorithm presents a positive evolution on the confidence of the object on a specific map cell being the ground-truth object, and also that the impact of each iteration depends on the location of the focal point. The other algorithms are not represented as their performance is similar, except for the Naïve Bayes where the probabilities are either 0 or 1 as explained on the previous section.

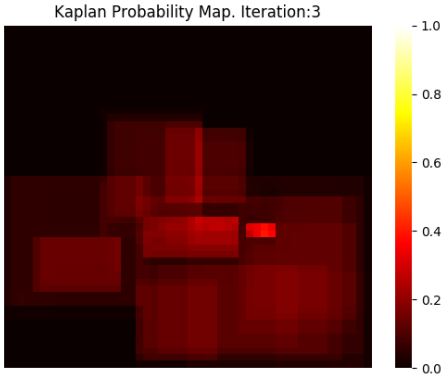
4.3 Next Best View Point

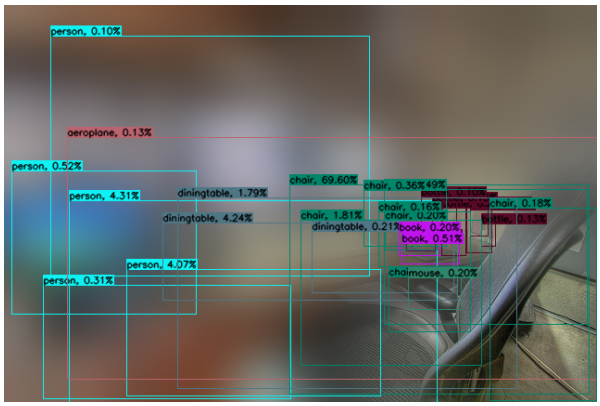
Up until now, every experiment considered either a fixed focal point or a set of randomly chosen focal points to check the performance of the algorithms used to implement the ”blocks” represented on fig. 3.1. On this section, we'll combine and take advantage of everything we've been experimenting until now plus an active search, to try to collect the most information relevant on the scene in the least number of gaze shifts. For that, knowing what is the most promising next view point is the key to achieve better performances than, per example, choosing a random point at each iteration.

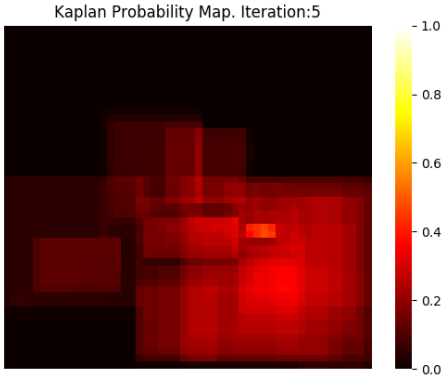

tvmonitor, 99.84%
tvmonitor, 0.52%
laptop, 99.88%
tvmonitor, 1.0
tvmonitor, 0.11%
bottle, 0.20%
cup, 0.69%
cup, 0.17%
cell phone, 0.24%
cell phone, 0.25%
book, 0.27%
mouse, 0.39%
keyboard, 19.91%
keyboard, 66.78%
chair, 0.95%


Kaplan Probability Map. Iteration:1
1.0
0.8
0.6
0.4
0.2
0.0


tvmonitor, 99.72%
tvmonitor, 2.15%
laptop, 99.83%
tvmonitor, 0.14%
cell phone, 0.54%
cell phone, 0.31%
cell phone, 0.15%
book, 0.14%
book, 0.12%
keyboard, 7.72%
keyboard, 64.00%
person, 0.17%


Kaplan Probability Map. Iteration:3
1.0
0.8
0.6
0.4
0.2
0.0


person, 0.10%
aeroplane, 0.13%
person, 0.52%
person, 4.31%
diningtable, 1.79%
diningtable, 4.24%
chair, 69.60%
chair, 0.36%
chair, 0.16%
chair, 0.20%
chair, 1.81%
diningtable, 0.21%
chair, 0.18%
book, 0.20%
book, 0.51%
person, 4.07%
person, 0.31%
chairmouse, 0.20%


Kaplan Probability Map. Iteration:5
1.0
0.8
0.6
0.4
0.2
0.0

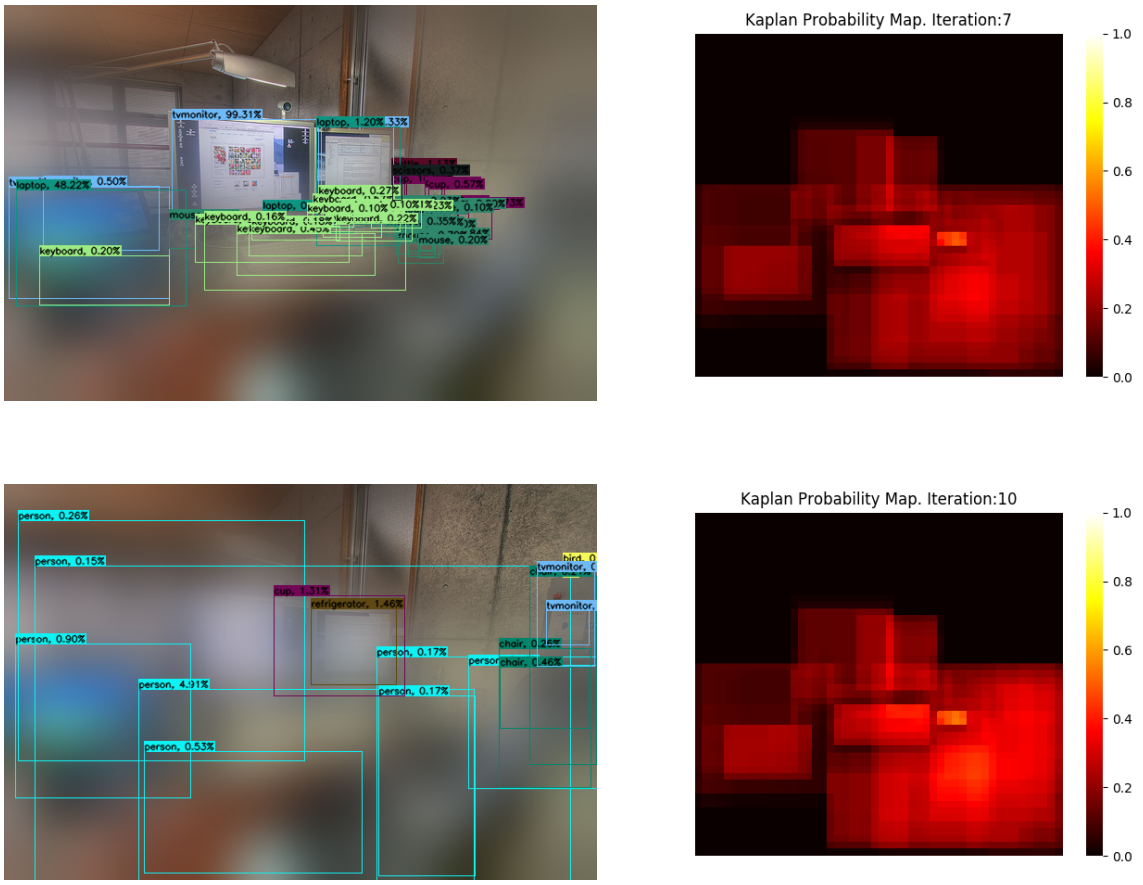


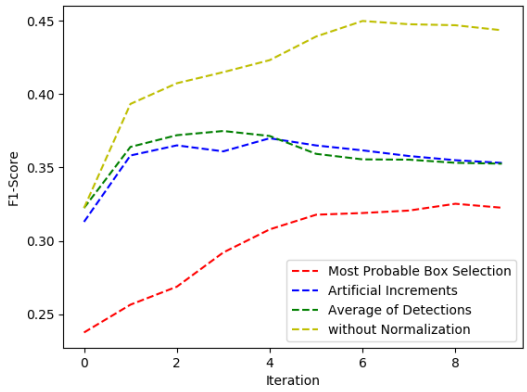
Figure 4.5: Performance evolution of the Kaplan algorithm throughout 10 iterations. Each line represents **Left:** the same image with a different focal point and **Right:** an heat-map with the expected value of the ground-truth object(s) given by the Kaplan method at a certain iteration.

4.3.1 Comparison of Update Normalization Methods

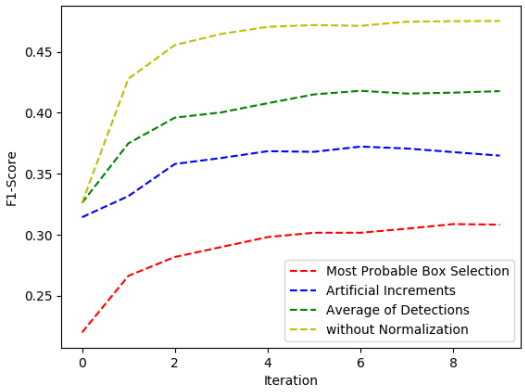
Let's first analyse the bounding box selection methods described and explained in section 3.2.3, in order to find which method achieves better results. The three normalization methods: Average of Detections, Most Probable Box Selection, and Artificial Increments, and not using any normalization method, have to be compared using the three acquisition functions defined in section 2.4.1, the KL Divergence Gain (eq. (3.32)), the Classification Entropy Loss (eq. (3.33)), and the Absolute Gain on the Difference Between Two Peaks (eq. (3.34)).

For this experiment, 150 random images from the COCO dataset were used and each one foveated 10 times (10 iterations). The foveation points were chosen by the acquisition functions and differ from method to method, nevertheless, the starting focal point was randomly chosen for each different image, but is the same on every method. The detections at each iteration were pre-processed by each of the

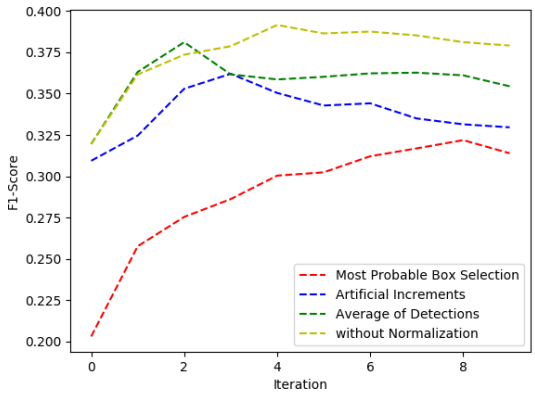
bounding box selection methods and then used to update the information of the map using the Modified Kaplan Update (eq. (3.13) with eq. (3.16)). The evolution of the F1-Score, iteration-wise, is graphically represented on fig. 4.6.



(a) F1-Score comparison using the absolute gain of the difference between two peaks as the acquisition function.



(b) F1-Score comparison using the KL divergence gain as the acquisition function.



(c) F1-Score comparison using the classification entropy loss as the acquisition function.

Figure 4.6: Performance metrics of 4 different fusion algorithms, analysed time-wise as new bounding boxes are detected.

It is important to first clarify that the only fusion algorithm used was the Modified Kaplan, since only the Modified Kaplan and the Naïve Bayes approach depend on the distance of the detection to the center of the fovea. The other fusion methods would make the same predictions for every next focal point and, therefore, would not be possible to choose the ”next best view point”. The Naïve Bayes approach was not tested since the results obtained on the last section were not satisfying.

Regarding the F1-Scores represented on fig. 4.6, the relative performance varies with the metric/acquisition function used. Nevertheless, on every acquisition function, there is a consensus that the normalization methods do not contribute with better performances upon scene exploration.

The Most Probable Box Selection method is the one that has the worst performance in terms of

the F1-Score. The most plausible explanation is that it is the only normalization method where we are actually losing information, since, in order to choose the bounding box with highest probability, all the others have to be ignored. The blur imposed by the foveal sensor (which makes the objects boundaries harder to define), and the possibility of having overlapping objects, do not cope with this normalization method, since the ignored boxes might have helped the algorithm to choose the next best view point.

For the Artificial Increments method, where we are giving strength to the update on map cells with less detections in order to balance the number of updates, dealing with foveated images (where some parts of the image are blurred, thus the detection algorithm does not perform as well), might be increasing the strength of wrong detections, lowering the performance of the algorithm.

The performance of the Average of Detections, although better than the other normalization methods, is still lower than do not using any normalization. This fact might be due to the distortion imposed on the confidence scores, upon averaging them. Whenever we average several confidence scores, the new score array does not correspond to a detection, therefore, since the foveal observation model were trained using directly the outputs of the object detector (YOLOv3), an array corresponding to the average of the confidence scores will not be correctly modeled. Consequently, the classification performance with the Average of Detections selection method decreases when using fusion methods that take advantage of the foveal observation model, just like the one used in this experiment, the Modified Kaplan Update.

4.3.2 Comparison of Acquisition Functions

On fig. 4.6 one can check the performance of bounding box selection methods for each acquisition function used in this work (section 2.4.1): Absolute Gain on the Difference between Two Peaks, KL Divergence Gain, and Classification Entropy Loss. We can now select the best bounding box selection method and check which acquisition function achieves better results.

On fig. 4.7 there is a clear difference between using the Classification Entropy Loss and the other acquisition functions. Although both the entropy and the KL Divergence measure similarly the amount of confusion on a map cell, the KL Divergence combines that confusion with the amount of updates done in that particular cell, more updates mean less uncertainty even if the probability of every class is the same. We can then say that the KL Divergence is the most suited metric to measure the uncertainty of the Dirichlet distributions that characterize the state of the map.

The difference between two peaks saturates quicker than the KL Divergence, and do not take into consideration if a given cell has more than one object. This might explain the drop in performance, when compared to the KL divergence gain.

It is now time to compare the benefits of active gaze selection with respect to random search.

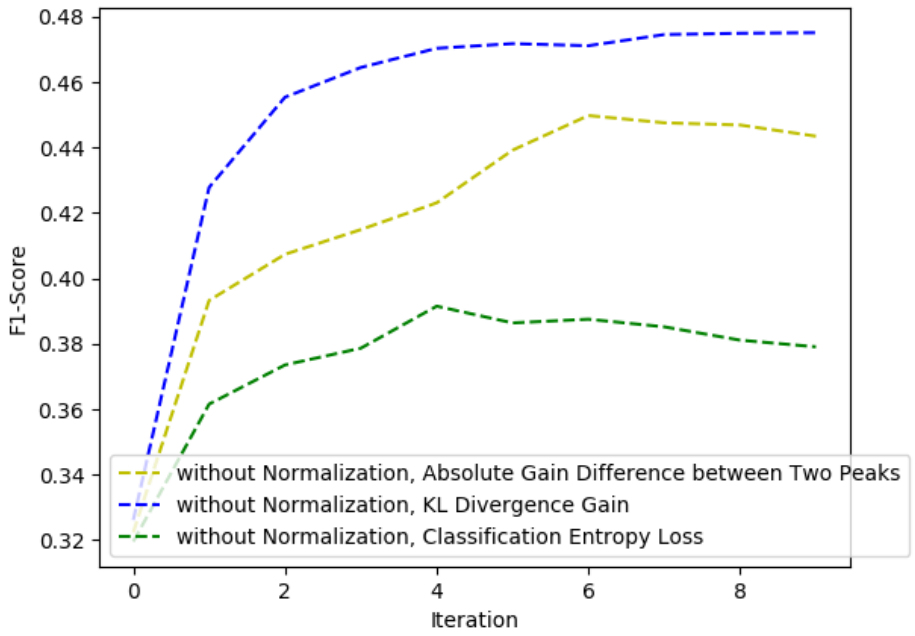


Figure 4.7: Comparison between the F1-Score of the algorithm, using the three different acquisition functions combined with the best bounding box selection method for each one.

4.3.3 Active Gaze Selection Performance

Using Active Perception methods to find and classify objects aims to choose the best location where to look at, in a way that the exploration is as fast as possible. For that, in order to check if the algorithm proposed is promising, one has to check the performance of actively choosing the next focal point against when choosing the focal point randomly.

As mentioned previously, the Modified Kaplan fusion method is the one that allows us to predict the evolution of the information on the map, depending on the next position of the center of the fovea. Thus, one can compare the performance of this method, combined with an acquisition function to choose the next point where to look at, against the performance of this method, choosing randomly the next focal point, and against the performance of the other fusion methods.

The experiment is as in section 4.3.1. In order to reduce possible biases, the starting focal point of a given image (the location of the center of the fovea on the first iteration), is chosen randomly and it's the same for every method, only varies from image to image. After that, the next focal points depend on whether we are using an acquisition function to choose them, or if we are choosing them randomly.

The analysis of fig. 4.8 will mainly focus on the performance of actively choosing the focal point against choosing it randomly, since the analysis of the fusion methods alone was already done on section 4.2.

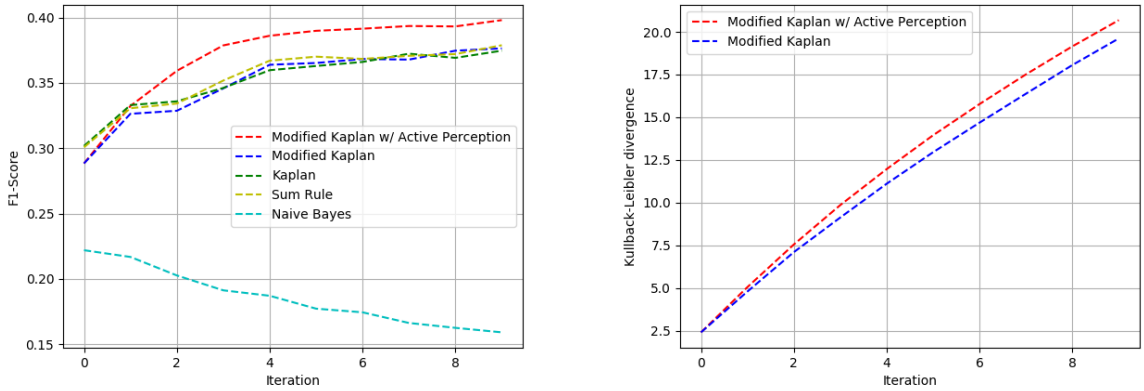


Figure 4.8: Left: F1-Score of the algorithm using the acquisition function "KL Divergence Gain" in red, against the F1-Score of all fusion methods mentioned in section 3.2 when choosing the focal point randomly. **Right:** KL Divergence evolution with and without using Active Perception (acquisition function "KL Divergence Gain") to choose the next best view point.

The results are promising, on the left side of fig. 4.8 one can immediately notice that the Modified Kaplan update jointly with the best Active Perception method analysed achieves better F1-Score at almost every iteration than all other fusion algorithms when choosing randomly the next focal point.

One other important aspect is the growth rate of the performance on classifying the objects on the image. Since the goal is to find and classify every object on the image, in the least number of gaze shifts, analysing how fast the algorithm can detect and correctly classify most of the objects is a key factor.

As one can see, choosing the next focal point by maximizing the predicted gain on the average KL divergence of the map, achieves an F1-Score around the third iteration that can not be surpassed by any of the methods that use random search. This means that predicting the next best view point, taking advantage of the difference of resolutions in the center of the fovea and on the peripheries, makes the algorithm analyse the scene more efficiently, going more often to the points of interest, whilst randomly choosing the focal point takes, in average, more gaze shifts to visit the places that contribute to increase the knowledge of the scene.

Besides the improved growth rate, we can also see on the left side of fig. 4.8 that choosing the next focal point by maximizing the KL Divergence Gain contributes to an overall performance improvement (on average) of around 2-3% on the F1-Scores after the 10 iterations of the experiment.

It's also interesting to note the evolution of the actual average KL divergence of the map when choosing the next view point by trying to maximize this metric against choosing it randomly (fig. 4.8 Right). When trying to maximize the gain of the KL divergence, we achieve a better KL divergence than when choosing randomly the focal point. This results once again validate our proposed Foveal Observation Model (section 3.1), meaning that the Observation Model gives us good estimations of the evolution of the KL divergence, for each possible next position of the focal point, predicting well the evolution of the

map based on the current knowledge.

A fact that one has to take into consideration upon analysing these results, is that the algorithm is performing a scene exploration in a small environment (one image). This fact reduces the gap between the performance of actively choosing the next best view point and choosing the next focal point randomly. If the scene were bigger, the number of points to choose from would also grow and, therefore, the performance when choosing randomly the next center of the fovea would be considerably smaller. Thus, actively choosing the next view point would be even more crucial, and the difference between performances is expected to grow. For future work, it would be interesting to analyse this prediction.

5

Conclusion

Contents

5.1 Conclusions 67
5.2 System Limitations and Future Work 68

5.1 Conclusions

In this thesis we propose a computational framework, inspired by human vision, that incorporates the combination of foveal vision and a state-of-the-art object detector with recent approaches on fusion of classifiers, to perform an active exploration for objects.

The main goal was to find and correctly classify as many objects as possible in one image, in the least number of gaze shifts. For this purpose, the work was divided in three major components. First, the Foveal Observation Model that corrects the outputs of the detections performed in foveated images (section 3.1). Secondly, the update of the knowledge about the world, where, at each saccade new detections are fused with the information already known (section 3.2). And, finally, the prediction of the next best view point, in order to orient the gaze to the most promising places (section 3.3).

Regarding the first component, object detectors were built to locate and classify objects on Cartesian images, thus, one of our contributions was to train, develop, and analyse a Foveal Observation Model that post-processes the results of the object detector, taking advantage of the confusion imposed by the blur on the periphery of the image to try to classify the objects in one passage.

The classification performance of the Foveal Observation Model was validated in our tests (as presented on section 4.1), reducing the uncertainty imposed on the classification scores while achieving a similar accuracy when compared to the object detector itself. From these results, we can conclude that the confusion between classes, due to the increasing blur as we go to the peripheries of the fovea, can be modeled. This means that each class of object produces a certain spectrum of confidence scores outputted by the object detector, depending on the level of distortion of the object, that can be used to have better knowledge of what is the class of the object that the detector is dealing with.

We also concluded that the Observation Model could make good predictions about the evolution of the map on the next iteration (section 4.3.3), depending on the location of the fovea, based only on the current knowledge. This is one of the most important features and contributions of this Observation Model, since it allows one to use this predictions to then choose the most promising point where to look next, in order to maximize or minimize a certain metric, depending on the task to perform.

About the fusion of classifiers, we extended Kaplan's work to our application, and corroborated his conclusions on the performance of the fusion algorithms [29], where both the sum rule and Kaplan's approach are suitable and have similar performances for the fusion of classifiers, although Kaplan's approach have a slight performance advantage as the number of classifications increase, both in terms of reducing the uncertainty as well as increasing the average expected value for the ground-truth class (see section 4.2.1). Then, we proposed a modified version of Kaplan's fusion algorithm, combining it with the outputs of the Foveal Observation Model instead of using directly the outputs of the object detector (section 3.2.2). We concluded on section 4.2.1 that the outputs of the Foveal Observation Model remain valid when combined with the fusion algorithm, and that the greediness of the classification using

the observation model is compensated by the limits imposed by the fusion algorithm on each update, something that doesn't happen when fusing these outputs with the Naïve Bayes approach.

For the last component, we wished to predict the best point where to look next. In order to choose one point over another, the expected influence on the map had, of course, to depend on the distance of each cell to the new focal point. This dependence could only be achieved using the fusion algorithm combined with the Foveal Observation model, the Modified Kaplan Update.

The results obtained on the last component are significant, and validate the proposed framework as being the first exploration algorithm with foveal vision that takes advantage of the performance of a state-of-the-art detector. The algorithm achieved a performance more than three times faster by trying to shift the gaze to the location that maximizes the KL divergence gain, and also contribute with an overall improvement of 2-3% of the performance (F1-Score), than when choosing randomly the next focal point (section 4.3.3).

Therefore, the results prove, that the newly trained and developed Foveal Observation Model is useful and valid to predict the map evolution in places where we have less information and/or higher uncertainty, meaning that it characterizes well the influence of the blur imposed by the foveated image on the objects. When combined with the other components of the work, the results show that it is possible to take advantage of the uncertainty imposed by this kind of images to optimize the exploration of a scene.

We can finally conclude that this work contributes with a promising new approach on active exploration, since it is a first step on taking advantage of the performance of a state-of-the-art object detector, trained on Cartesian images, to develop a searching algorithm using foveal vision. By modelling the uncertainty imposed by the image on the detections we showed that it was possible to perform a search for objects on a given environment without resorting to more specific and limited heuristics.

5.2 System Limitations and Future Work

One of the major limitations of this work is that we are searching for objects within an image. In the future, we intend to expand the approach to real-world scenarios, where the search space is bigger, and the objects are not always on the field of view, giving more importance to correctly predicting the next best view point.

Also, the reduction on the number of saccades is considerable, reflecting the value of the approach, but there's still the need to analyse if this reduction is translated into computational gains, *i.e.*, if using foveal vision translates in an improvement on the computational efficiency of the exploration of the scene, against using full-resolution vision (Cartesian images).

Thus, it would also be of interest to train the algorithm for foveated images that do explore the

characteristics of the foveal vision. As for now, the transformation to foveated images is done by applying a filter on top of the original Cartesian image, without changing the resolution of the pixels. This type of foveated images do not present computational gains, since the amount of stored data is the same as a full-resolution image. Therefore, although our work serves as a first step on using foveal vision to explore a scene for objects, in the future we wish to use foveated images leveraging log-polar transformations (following Ozimek & Siebert's work [4], for example) to reduce the required computational resources and increase the exploration speed.

Bibliography

[1] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, 1995.

[2] C. Melício, R. Figueiredo, A. F. Almeida, A. Bernardino, and J. Santos-Victor, "Object detection and localization with Artificial Foveal Visual Attention," in *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2018, pp. 101–106.

[3] R. Figueiredo, "Space-Variant Vision Mechanisms for Resource-Constrained Humanoid Robot Applications," Ph.D. Thesis, Instituto Superior Técnico, Universidade de Lisboa, 2020.

[4] P. Ozimek and J. P. Siebert, "Integrating a Non-Uniformly Sampled Software Retina with a Deep CNN Model," in *BMVC 2017 Workshop on Deep Learning on Irregular Domains*, 2017.

[5] J. P. Siebert, P. Ozimek, L. Balog, N. Hristozova, and G. Aragon-Camarasa, "Smart Visual Sensing Using a Software Retina Model," in *IROS2018 Workshop: Unconventional Sensing and Processing for Robotic Visual Perception, at 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[6] Adrian Rosebrock, "Intersection over Union (IoU) for object detection - PyImageSearch," p. 1, 2016. [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

[7] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, feb 2018.

[9] J. H. Krantz, "The Stimulus and Anatomy of the Visual System," in *Experiencing Sensation and Perception*, 2012, ch. 3, pp. 3.1 – 3.36.

[10] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, Eds., *Principles of Neural Science*, 3rd ed. New York: Elsevier, 1991.

[11] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, “Search in the real world: Active visual object search based on spatial relations,” *2011 IEEE International Conference on Robotics and Automation*, pp. 2818–2824, 2011.

[12] A. Aydemir, A. Pronobis, M. Gobelbecker, and P. Jensfelt, “Active visual object search in unknown environments using uncertain semantics,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 986–1002, 2013.

[13] A. F. Almeida, R. Figueiredo, A. Bernardino, and J. Santos-Victor, “Deep Networks for Human Visual Attention: A Hybrid Model Using Foveal Vision,” in *ROBOT 2017: Third Iberian Robotics Conference*, A. Ollero, A. Sanfeliu, L. Montano, N. Lau, and C. Cardeira, Eds. Cham: Springer International Publishing, 2018, pp. 117–128.

[14] F. B. Colavita, “Human sensory dominance,” *Perception & Psychophysics*, vol. 16, no. 2, pp. 409–412, 1974.

[15] S. Clippingdale and R. Wilson, “Self-similar Neural Networks Based on a Kohonen Learning Rule,” *Neural Networks*, vol. 9, no. 5, pp. 747–763, jul 1996.

[16] V. J. Traver and A. Bernardino, “A review of log-polar imaging for visual perception in robotics,” *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 378–398, 2010.

[17] J. Yan, Z. Lei, L. Wen, and S. Z. Li, “The Fastest Deformable Part Model for Object Detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[21] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective Search for Object Recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[23] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[24] T. Minka, “Estimating a Dirichlet distribution,” *Technical report, M.I.T.*, 2000.

[25] J. Liu, S. Lipschutz, and M. Spiegel, *Schaum’s Outline of Mathematical Handbook of Formulas and Tables, 4th Edition*. <country>US</country>: McGraw-Hill, 2012. [Online]. Available: https://mhebooklibrary.com/doi/book/10.1036/9780071795388

[26] J. Huang, “Maximum Likelihood Estimation of Dirichlet Distribution Parameters,” *CMU Technique Report*, 2005.

[27] L. Montesano and M. Lopes, “Learning grasping affordances from local visual descriptors,” in *2009 IEEE 8th International Conference on Development and Learning*, 2009, pp. 1–6.

[28] R. P. de Figueiredo, A. Bernardino, J. Santos-Victor, and H. Araújo, “On the advantages of foveal mechanisms for active stereo systems in visual search tasks,” *Autonomous Robots*, vol. 42, no. 2, pp. 459–476, 2018.

[29] L. M. Kaplan, S. Chakraborty, and C. Bisdikian, “Fusion of classifiers: A subjective logic perspective,” in *2012 IEEE Aerospace Conference*, 2012, pp. 1–13.

[30] B. Settles, “Active Learning Literature Survey,” University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[31] D. H. Ballard, “Active Perception,” *Encyclopedia of Neuroscience*, no. March, pp. 31–37, 2009.

[32] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, “Active vision,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[33] M. Grotz, T. Habra, R. Ronsse, and T. Asfour, “Autonomous view selection and gaze stabilization for humanoid robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1427–1434.

[34] E. Rivlin and H. Rotstein, “Control of a Camera for Active Vision: Foveal Vision, Smooth Tracking and Saccade,” *International Journal of Computer Vision*, vol. 39, pp. 81–96, 2000.

[35] S. Kullback and R. A. Leibler, “On Information and Sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 1–164, 1951.

[36] B. Kurt, “Kullback-Leibler Divergence Between Two Dirichlet (and Beta) Distributions,” 2013. [Online]. Available: http://bariskurt.com/kullback-leibler-divergence-between-two-dirichlet-and-beta-distributions/

A

List of Symbols

Table A.1: A list of the symbols used on this document are synthesized here to serve as an auxiliary reader guide. $\mathbb{P}^N$ will be considered as the probability simplex for dimension N, which is a vector of $\mathbb{R}^N$ whose elements are all positive and sum to unit.

Symbol	Type	Definition
$\mathcal{O} \subseteq [1, ..., N]$	Set	Set of objects represented on the scene
N	Integer	Number of objects represented on the scene
$O_m \in \mathcal{C}$	Label	Label of object $m \in [1, ..., N]$
$\mathcal{C} \subseteq [1, ..., K]$	Set	Set of possible classes
K	Integer	Number of possible classes
$c_k \in \mathcal{C}$	Label	Label for class $k \in [1, ..., K]$
$\mathcal{I}_t \subseteq [I_{t,1}, ..., I_{t,L_t}]$	Set	Set of detected objects at instant $t \in [1, ..., T]$
T	Integer	Total number of time instants
L_t	Integer	Number of detected objects at instant t
$I_{t,l} = (\boldsymbol{B}_{t,l}, \boldsymbol{S}_{t,l})$	Tuple	l-th detection at instant t, $l \in [1, ..., L_t]$
$\boldsymbol{B}_{t,l} \in \mathbb{R}^4$	Vector	Bounding box coordinates of the l-th detection at instant t
$\boldsymbol{S}_{t,l} \in \mathbb{P}^K$	Vector	Score vector of the l-th detection at instant t
$s_{t,l,k} \in \boldsymbol{S}_{t,l}$	Probability	Score for the class k of the l-th detection at instant t
$(u_{t,l}, v_{t,l}) \in (\mathbb{R}, \mathbb{R})$	Scalars	Relative coordinates of the l-th detection at instant t to the focal point
$(x_t, y_t) \in (\mathbb{R}, \mathbb{R})$	Scalars	Global coordinates of the focal point at instant t
$(x, y) \in (\mathbb{R}, \mathbb{R})$	Scalars	Global coordinates

Continued on next page

Table A.1 – continued from previous page

Symbol	Type	Definition
$C_{x,y} \in \mathcal{C}$	Label	Random variable for the class label at pixel (x, y)
$d_{t,l} \in \mathbb{R}$	Scalar	Distance of the relative coordinates $(u_{t,l}, v_{t,l})$ to the focal point (x_t, y_t)
$f_{t,l}(x, y)$	Function	Returns the l for a given instant t, and coordinates (x, y)
$\boldsymbol{p} \in \mathbb{P}^K$	Vector	Vector of parameters of a categorical distribution
$p_k \in \boldsymbol{p}$	Probability	Expected value for the class k of the categorical distribution
$\boldsymbol{\alpha}_{k,d_{t,l}} \in \mathbb{R}^K$	Vector	Parameters of the Dirichlet distribution that models the observation model
$\alpha_{k,d_{t,l},j} \in \boldsymbol{\alpha}_{k,d_{t,l}}$	Scalar	Parameter $j \in [1, ..., K]$ of the Dirichlet distribution
$\mathcal{L} \subseteq [L_1, ..., L_T]$	Set	Total set of observation vectors
$\boldsymbol{L}_t \in \mathbb{R}^K$	Vector	t-th observation likelihood vector
$l_{t,j} \in \boldsymbol{L}_t$	Scalar	Likelihood of observation t belonging to the class j
$\boldsymbol{M}_t(x, y) \in \mathbb{R}^K$	Vector	Vector of parameters that encodes the resulting distribution of the fusion of all observations that overlap the pixel (x, y), up until instant t
$(x*, y*) \in (\mathbb{R}, \mathbb{R})$	Scalars	Coordinates of the predicted next best focal point
$(x_m, y_m) \in (\mathbb{R}, \mathbb{R})$	Scalars	Map cell coordinates
$\boldsymbol{p}_t^{x_m,y_m} \in \mathbb{P}^K$	Vector	Vector of parameters, at the instant t, of the categorical distribution that characterized the generation of objects on the map cell (x_m, y_m)
$p_{t,k}^{x_m,y_m} \in \boldsymbol{p}_t^{x_m,y_m}$	Probability	Expected value of the class k, at instant t, on the map cell (x_m, y_m)
$\boldsymbol{\beta}_t^{x_m,y_m} \in \mathbb{R}^K$	Vector	Vector of parameters of the resulting Dirichlet distribution of the fusion algorithms at instant t for the map coordinates (x_m, y_m)
$\beta_{t,k}^{x_m,y_m} \in \boldsymbol{\beta}_t^{x_m,y_m}$	Scalar	Parameter of the state of the map for the class k, at instant t, on the map cell (x_m, y_m)
$\boldsymbol{S}'_{t,l} \in \mathbb{P}^K$	Vector	l-th score vector at instant t outputted by the object detector after being modelled by the observation model
$s'_{t,l,k} \in \boldsymbol{S}'_{t,l}$	Probability	l-th score (of the class k) at instant t outputted by the object detector after being modelled by the observation model