UNIVERSITÀ DEGLI STUDI DI GENOVA
Dottorato di Ricerca in Ingegneria Biomedica
XIX Ciclo

DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E TELEMATICA
VIA CAUSA 13 - 16145 - GENOVA (I)
TEL. 010-3532946 • FAX 010-3532948

LEARNING AND ADAPTATION IN COMPUTER VISION

Francesco Orabona

Relatore:
Giulio Sandini

Coordinatore:
Vittorio Sanguineti

Correlatore:
Giorgio Metta

DISSERTAZIONE PRESENTATA PER IL CONSEGUIMENTO
DEL TITOLO DI DOTTORE DI RICERCA

A.A. 2006/2007

If the human mind was simple enough to understand, we'd be too simple to understand it.

Emerson Pugh

. . . questo grandissimo libro [della natura] che continuamente ci sta aperto innanzi agli occhi (io dico l'universo), non si può intendere se prima non s'impara a intender la lingua, e conoscer i caratteri né quali è scritto. Egli è scritto in lingua matematica, e i caratteri son triangoli, cerchi, ed altre figure geometriche, senza i quali mezzi è impossibile a intendere umanamente parola; senza questi è un aggirarsi vanamente per un oscuro laberinto.

Galileo Galilei

Acknowledgments

I would like to thank all the persons that have helped me during my studies. The list would be extremely long, but all of them already know who they are! Among all I would like to thank especially my supervisor Giulio Sandini to have made everything possible. Moreover my PhD period at the LIRA-Lab would have been extremely different without the stimulating discussions with Giorgio Metta, continuous source of ideas and knowledge in every possible scientific fields!

I thank also the persons of the lab with which I have spent the greater part of my life in the last three years: Carlos Beltran, Fabio Berton, Matteo Brunettini, Claudio Castellini, Lorenzo Natale, Francesco Nori, Ingrid Sica. I would like also to thank Michael Bucko, Arjan Gijsberts, Antoine Pasquali, and Satyajit Rao for the infinite discussions in the kitchen about cognition, mathematics and... dinner!

A special thanks also to my best friends Miriam, Peppe, and Milena that have supported me during the difficult periods of these three years, and also to Violetta to have fought my laziness in writing anything different from mathematical formulas!

Finally a special “Thank” to my family who believed in me and in my dreams.

Abstract

In this thesis we analyze the topics of adaptation and learning in the context of computer vision. Until now the ability of humans to adapt and learn how to solve new tasks from their own experience remain impossible to replicate in an artificial system. Even if computers can beat humans on small, constrained domains, the generality of the human mind has no counterpart in the digital world.

The keys to understand and replicate a brain in a robot, could be to try to discover the general principles that govern our internal algorithms and to formalize them mathematically, and then to implement them in software.

If it is true that our brains are the product of an long process of adaptation to the environment, we could be able to “predict” our biology studying the world itself.

In this thesis we will show that, on one hand, it is possible to learn basic features of the processing of the neurons of the primary visual cortex from the row visual data and, on the other hand, we can learn such a high level visual skills as object classification.

The obtained results support the idea that these two aspects are critical for the comprehension of biological intelligence, and, hence, for creating an artificial cognitive agent.

To my grandfathers

Contents

1 Introduction **13**

1.1 Adaptation and learning 14
1.2 Putting things in context 15
1.3 Thesis outline 17

2 Learning Association Fields **19**

2.1 Gestalt laws, statistics and neurons 20
2.2 Learning from natural images 22
2.2.1 Feature extraction stage 23
2.2.2 Tensors 24
2.3 Preliminary results 24
2.3.1 The path across a pixel 26
2.4 Results 26
2.5 Using the fields 29
2.6 Discussion 30

3 Visual Attention **33**

3.1 Computational models of visual attention 33
3.1.1 A proto-object based model of visual attention 36
3.2 Setup 37
3.3 The model 38
3.3.1 Log-polar images 39
3.3.2 Feature extraction 40
3.3.3 Proto-objects 41
3.3.4 Inhibition of return 43
3.4 Learning about objects 45
3.5 Results 48
3.6 Discussion 49

4 Learning and Support Vector Machines **51**
4.1 Support Vector Machines 52
4.1.1 The importance of online learning 53
4.2 Background Mathematics 54
4.3 Previous work 56
4.4 Sparseness of the solution 58
4.5 Online Independent Support Vector Classification 59
4.5.1 Linear independence 60
4.5.2 Training the machine 62
4.6 Experimental Results 64
4.7 Discussion 67

5 Object Recognition and Categorization **69**
5.1 Two view-based models for object recognition 70
5.1.1 Standard Model 70
5.1.2 SIFT 71
5.2 Results on a categorization task 72
5.3 Adapting the features through selection 73
5.3.1 Unsupervised feature selection for SVM 74
5.3.2 Results 76
5.4 Discussion 78

6 Conclusions **79**

A How to include the bias term in OISVM **81**

B Kernels for SVM **83**
B.1 Some notes on polynomial kernels 83
B.2 The local matching kernel 84

References **91**

Chapter 1

Introduction

Contents

1.1 Adaptation and learning 14
1.2 Putting things in context 15
1.3 Thesis outline 17

IF we consider the aim to survive like a problem to solve, we can realize that animals and humans have already efficiently solved this problem using different means. Evolution on one hand has shaped bodies to solve different tasks, on the other hand has created minds able to solve new problems as they arise in everyday life.

The ultimate goal of Artificial Intelligence (AI) is considered to build an artificial agent with cognitive abilities: how is it related to the above considerations? What is an agent? What is cognition? Already Alan Turing had considered the difficulties of such definitions, and invented the idea of an operative test as a mean to evaluate artificial intelligence: requirement for an intelligent agent is to behave in such a way to fool a human interrogator, hence to be indistinguishable from a human. Thus it is easier to define the intelligence in relation to humans, instead of giving an absolute definition. Another way to try to define intelligence could be in the context of the tasks that a cognitive agent should be able to solve. But if we focus on specific abilities of humans and animals that we want to replicate, it is easy to find examples of softwares that are even better of their biological counter parts, *e.g*. chess softwares. An interesting observation on this has been done by Douglas Hofstadter (Hofstadter, 1999):

> It is interesting that nowadays, practically no one feels that sense of awe any longer - even when computers perform operations that are incredibly more sophisticated than those which sent thrills down spines in the early days. [...] There is a related “Theorem” about progress in AI: once some mental function is programmed, people soon cease

> to consider it as an essential ingredient of "real thinking". The ineluctable core of intelligence is always in that next thing which hasn't yet been programmed. This "Theorem" was first proposed to me by Larry Tesler, so I call it Tesler's Theorem: *"AI is whatever hasn't been done yet"*.

Thus the aim of creating an intelligent machine should be seen under another view: to create something that is able *to solve and learn to solve* different problems that it can meet in its environment. This different view offers us also the possibility to move the focus from the *performances* to the *reasons*. That is, it is more important to understand why, *e.g.*, the human retina has a spatial variant resolution, instead of focusing on gaining few percent points on an object recognition task. In the first case we could understand more general conditions that rules biological beings and that, if replicated in a computer, could make possible a quantum leap in the performances.

Borrowing ideas from another field, we could say that making intelligent artifacts can be seen like making mechanical flying machines. We really should take the time to understand how biological systems have solved the problem, even if we do not want to make an airplane with feathers. In fact the correct solution to mechanical flight was to take only a minimum amount of information from observation of birds and then figure out how to use available technology to make a machine fly. This scientific solution was clearly better than the prescientific idea that certain substances or forms could naturally rise or sink according to their essential properties (like in the Icarus legend in which bird-like wings give men the power of flight).

Thinking about the Tesler's "Theorem", one big differences between the most advanced artificial intelligence achievements and what animals and humans can do, could be the ability to solve problems, and generalizing from previous experience. Computer can be programmed to solve specific tasks or to learn to classify certain stimuli with a specific algorithms, but each problem requires a different method, that often uses specific information of the problem. Often this prior information is the result of a deep analysis done by the programmer, and not by the program itself. Again, the difference is the learning of living being is guided by general principles, that allow them to extract *autonomously* all the necessary information to solve specific tasks.

1.1 Adaptation and learning

If learning and adaptation seem to be the two key aspects that should be addressed to replicate the intelligent behaviors of animals and humans, however it is true that these cognitive abilities have sense only in relation to their particular needs and to ambient in which they live. It has no much sense to talk about adaptation without also considering what is the object of this adaptation. An agent can be seen as continuously adapting to the world, as perceived by its, and its own body can be

considered just as a special part of the world. Same reasoning can be done in a deeper level, taking into consideration the functional role of neurons and neural systems. A long standing hypothesis states that sensory systems are matched to the statistical properties of the signals to which they are exposed (Barlow, 1961).

The difference between adaptation and learning is quite narrow in a digital agent. We could consider the first as a product of a design process that has decided which abilities should be innate, and on which build more complex ones, through the second. Also in living beings we can do a similar distinction, in fact the expression Nature vs Nurture indicates the debate about the relative importance of an individual's innate qualities ("nature") versus personal experiences ("nurture") in determining or causing individual differences in physical and behavioral traits. In the same way we should distinguish between the innate abilities that a robot should have ready to be used, preprogrammed, and the abilities that it should be able to learn.

A first impulse would be to consider most of our high abilities as innate, but this is not the true. For example the same "concept of object" seems to develop across the first 6 months after birth (Johnson, 2005). In fact, it seems that infants are born without any means to perceive occlusion and, hence, no knowledge of objects. This example tells us that, maybe, most of the cognitive abilities, that we would like to see in an artificial machine, can be learnt from the experience, given enough time and an appropriate set of core abilities.

To summarize with an example, we are able to manipulate objects because we have hands, because we can learn how to use our hands, and because the world is made of objects.

1.2 Putting things in context

The human visual and attentive system will be taken as a case study. We can see, by analyzing the visual system, a clear example of how the evolution has shaped the bodies of many mammals. Typical visual tasks require both high acuity and a wide field of view. High acuity is needed for recognition tasks and for controlling precise visually guided movements. A wide field of view is needed for search tasks, for tracking multiple objects, being aware of possible source of dangers, *etc*. A common trade-off found by evolution in biological systems is to sample parts of the visual field at a high enough resolution (fovea) to support the first set of tasks, and to sample the rest of the field at an adequate level to support the second set. This is seen in animals with foveate vision, such as humans, where the density of photoreceptors on the retina is highest at the center and falls off dramatically toward the periphery. This space-variant visual system requires them to move their eyes, three times a second on average, in order to position their foveae onto interesting locations of the visual space. This allows taking a series of small "snapshots" at very high-resolution. The fact that this is the only way that allows clear "vision" implies the existence of an attention system which, at any moment

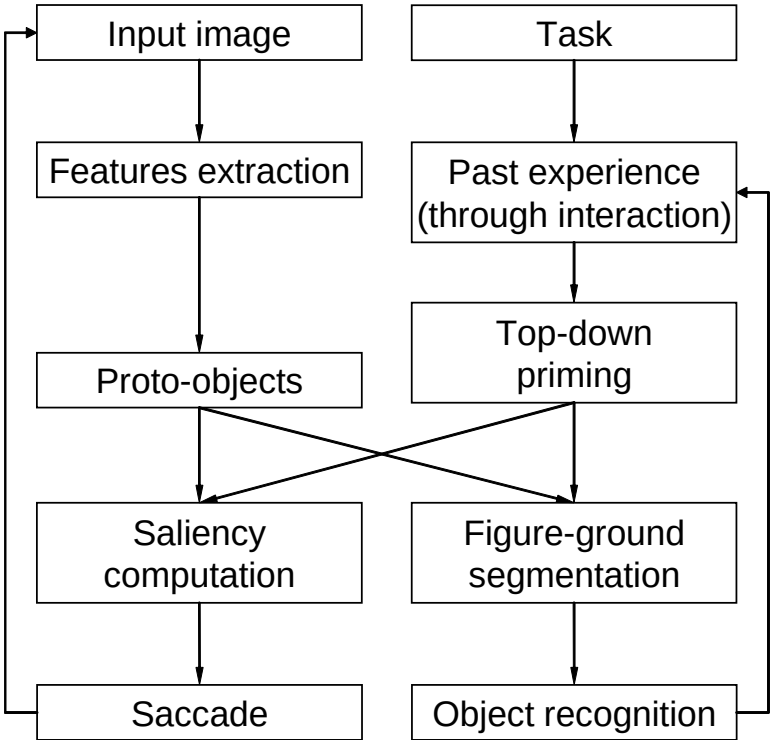

Figure 1.1: The relation between action, attention and experience.

in time, selects the point to fixate next. This leads to two sorts of questions: i) how to move the eyes efficiently to important locations in the visual scene, and ii) how to decide what is important and, as a consequence, where to look next. It is important to answer these questions to understand that there are two mechanisms that act at the same time. One is hard-coded in the brain, the other one has been learnt through the experience and the interaction with the world. In fact there are things that attract our attention instinctively, *e.g*. rapid change in the scene (Yantis and Jonides, 1984), and some other that are learnt from the experience, *e.g*. the image features that attract the attention of a radiologist viewing an X-ray image (Mylers-Worsley *et al.*, 1988). In general it has been shown that scanpaths for an individual are modified by the task presented (Yarbus, 1967).

From these considerations about the spatial density of the photoreceptors, the need for an active vision and, hence, for a mechanism of visual attention, we clearly see that the physiology of vision has shaped not only the way in which the image are scanned but our entire perception. In fact the external world is sensed continuously instead of maintaining and updating some complicated internal model. This idea has been summarized by O'Regan as: "The world as an outside memory" (O'Regan, 1992). The sentence remarks the fact that it is important to consider the problem of vision, and perception in general, deeply rooted in the physical

world. Given that, for example, changes in the world seem to be easily detectable, it is cheaper to store in memory a rough representation of the external world, directly accessing to it when a detailed information is needed and to keep track of the changes.

Moreover, it has no sense to talk about perception without talking about action, so it is logical to think that our perception is biased versus a representation that is useful to act on physical objects. In the case of visual attention this corresponds to ask if the attention is deployed on objects (object-based) or on space locations (space-based). This idea is supported by the discovery in monkeys of a class of neurons (*mirror neurons*) which not only fire when the animal performs an action directed to an object, but also when it sees another monkey or human performing the same action on the same object (Fadiga *et al.*, 2000). Indeed, this tight coupling of perception and action is present in in visual attention too. In fact, it has been shown in (Fischer and Hoellen, 2004) that more object-based attention is present during a grasping action, than space-based one. But how can we attend to objects before they are recognized? To solve this contradiction Rensink (Rensink, 2000b; Rensink, 2000a) introduced the notion of "proto-objects", that are volatile units of visual information that can be bound into a coherent and stable object when accessed by focused attention and subsequently validated as actual objects. In fact, it is generally assumed that the task of grouping pixels into regions is performed before selective attention is involved by perceptual organization and Gestalt grouping principles (Palmer and Rock, 1994).

All the above considerations can be summarized in the block diagram of Figure 1.1.

1.3 Thesis outline

The main focus of this thesis will be the development of abilities related to the visual system, in a biological inspired way. Starting from the above considerations various example of learning and adaptation will be taken under considerations. The thesis is organized as follows. Chapter 2 presents a method to adapt to the statistics of the world, learning second order relations between different edge detectors, and hence learning some of the Gestalt principles. Chapter 3 addresses the problem of modeling active vision, through the use of a proto-object based attentive system. In Chapter 4 an online algorithm for classification is presented, with a detailed mathematical analysis of his theoretical foundations. Finally, in Chapter 5 there is an application of a supervised learning procedure on an object categorization task. In the last chapter we draw the conclusions and discuss the future work. Finally in the Appendix there are some mathematical details that were not reported in the main text.

Chapter 2

Learning Association Fields

Contents

2.1 Gestalt laws, statistics and neurons 20
2.2 Learning from natural images 22
2.3 Preliminary results 24
2.4 Results 26
2.5 Using the fields 29
2.6 Discussion 30

THE term "grouping" (or "segmentation") is a common concept in the long research history of perceptual grouping by the Gestalt psychologists. In particular the Gestaltists tended to view their grouping phenomena as an illustration of the perceiver imposing a seemingly arbitrary (albeit systematic) organization upon the stimuli (*e.g.* (Wertheimer, 1923)). Nowadays the more typical view of such grouping demonstrations would be that they reflect non-arbitrary properties within the stimuli (similarity, common motion, *etc.*), which the visual system exploits heuristically because these properties are likely to reflect divisions into distinct objects in the real world. In particular these properties work because they reflect characteristics of the real world. In this sense it should be possible to learn these heuristic properties, that is, it should be possible to adapt to the statistics of natural images, learning the properties of the visual world. These properties can be then exploited to have a more efficient representation, *e.g.* (Buccigrossi and Simoncelli, 1999), and to complete missing information (Hyvärinen *et al.*, 2001).

Previous studies have shown that it is possible to learn certain properties of the responses of the neurons of the visual cortex, as for example the receptive fields of complex and simple cells, through the analysis of the statistics of natural images and by employing principles of efficient signal encoding from information theory, *e.g.* (Bell and Sejnowski, 1997). Here we want to go further and consider how the output signals of 'complex cells' are correlated and which information is likely to

be grouped together. We want to learn 'association fields', which are a mechanism to integrate the output of filters with different preferred orientation, in particular to link together and enhance contours. We used static natural images as training set and the tensor notation to express the learned fields. Finally we tested these association fields in a computer model to measure their performance.

This chapter is organized as follows: section 2.1 introduces the idea of the link between the Gestalt laws and the statistics of the world. Section 2.2 contains a description of the method, and section 2.3 describes a first set of experimental results and a method to overcome problems due to the non-uniform distribution of the image statistics. In section 2.4 we show the fields computed with this last modification and finally in sections 2.5 and 2.6 we show the performance of the fields in edge detection on a database of natural images and we draw some conclusions.

2.1 Gestalt laws, statistics and neurons

The goal of perceptual grouping in computer vision is to organize visual primitives into higher-level primitives thus explicitly representing the structure contained in the data. The idea of perceptual grouping for computer vision has its roots in the well-known work of the Gestalt psychologists back at the beginning of the last century who described, among other things, the ability of the human visual system to organize parts of the retinal stimulus into "Gestalten", that is, into organized structures. They formulated a number of so-called Gestalt laws (proximity, common fate, good continuation, closure, *etc*.) that are believed to govern our perception. It is logical to ask if these laws are present in the statistics of the world.

On the other hand it has been long hypothesized that the early visual system is adapted to the input statistics (Barlow, 1961). Such an adaptation is thought to be the result of the joint work of evolution and learning during development. Neurons, acting as coincidence detectors, can discover and use regularities in the incoming flow of sensory information, which eventually represent the Gestalt laws. It has been proposed that, for example, the mechanism that link together the elements of a contour is rooted in our biology, with neurons with lateral and feedback connections implementing these laws.

There is a large body of literature about computational modeling of various parts of the visual cortex, starting from the assumption that certain principles guide the neural code ((Simoncelli and Olshausen, 2001) for a review). In this view it is important to understand why the neural code is as it is. Bell and Sejnowski (Bell and Sejnowski, 1997), for example, demonstrated that it is possible to learn receptive fields similar to those of simple cells starting from natural images. In particular they demonstrated that it is possible to reproduce these receptive fields hypothesizing the sparsity and independence of the neural code. In spite of this, there is very little literature on learning an entire hierarchy of features, that is not only the first layer, and possibly starting from these initial receptive fields.

A step in the construction of this hierarchy is the use of 'association fields'

Figure 2.1: Sample input image from the Berkeley Segmentation Database. All the images were converted to grayscale before using the proposed method.

(Field *et al.*, 1993). In the literature, association fields are often hand-coded and employed in many different models with the aim to reproduce the human performance in contour integration. These fields are supposed to resemble the pattern of excitatory and inhibitory lateral connection between different orientation detector neurons as found, for instance, by Schmidt *et al*. (Schmidt *et al.*, 1997). In fact, Schmidt has shown that cells with an orientation preference in area 17 of the cat are preferentially linked to iso-oriented cells. Furthermore, the coupling strength decrease with the difference in the preferred orientation of pre- and post-synaptic cell. Models typically consider variations of the co-circular approach (Grossberg and Mingolla, 1985; Guy and Medioni, 1996; Li, 1998), that is two oriented elements are part of the same curve if they are tangent to the same circle. Others (Vonikakis *et al.*, 2006) have considered exponential curves instead of circles obtaining similar results.

Our question is whether it is possible to learn these association fields from the statistics of natural images. One of the first publication addressing second order relations of edge-like structures in images is from Krüger (Krüger, 1998). Then different authors have used different approaches to try to "learn" this fields: using a database of tagged images (Elder and Goldberg, 2002; Geisler *et al.*, 2001), using motion as an implicit tagger (Prodöhl *et al.*, 2003) or hypothesizing certain coding properties of the cortical layer (Hoyer and Hyvärinen, 2002).

Our approach is similar to to one of Sigman *et al*. (Sigman *et al.*, 2001), which uses images as the sole input. Further, we aim to obtain precise association fields, useful to link contours in a computer model.

2.2 Learning from natural images

We assume the existence of a first layer that simulates the behavior of the complex cells; in this paper we do not address the issue on how to learn them since we are interested in the next level of the hierarchy. Using the output of this layer we want to estimate the mean activity around points with a given orientation. For example it is likely that if a certain image position contains a horizontal orientation, then the adjacent pixels on the same line would be points with an orientation almost horizontal.

To have a precise representation of the orientations and at the same time something mathematically convenient we have chosen to use the tensor notation. Second order symmetric tensors can capture the information about the first order differential geometry of an image. Each tensor describes both the orientation of an edge and its confidence for each point. The tensor can be visualized as an ellipse, whose major axis represents the estimated tangential direction and the difference between the major and minor axis the confidence of this estimate. Hence a point on a line will be associated with a thin ellipse while a corner with a circle. Consequently given the orientation of a reference pixel, we estimate the mean tensor associated with the surrounding pixels. The use of the tensor notation give us the possibility to exactly estimate the preferred orientation in each point of the field and also to quantify its strength and confidence.

We have chosen to learn a separate association field for each possible orientation. This is done for two main reasons:

- It is possible to find differences between the association fields. For example, it is possible to verify that the association field for the orientation of 0 degrees is stronger than that of 45 degrees.

- For applications of computer vision, considering the discrete nature of digital images, it is better to separate the masks for each orientation, instead of combining the data in a single mask that has to be rotated leading to sampling problems. The rotation can be done safely only if there is a mathematical formula that represents the field, while on the other hand we are inferring the field numerically.

We have chosen to learn 8 association fields, one for each discretized orientation. The extension of the fields is chosen to be of 41x41 pixels taken around each point. It should be noted that even if we quantized the orientation of the (central) reference pixel to classify the fields, the information about the remaining pixels in the neighbor were not quantized, differently to (Geisler *et al.*, 2001; Sigman *et al.*, 2001). There is neither a threshold nor a pre-specified number of bins for discretization and thus we obtain a precise representation of the association field.

Images used for the experiments were taken from the publicly available database

Figure 2.2: Complex cells output to the image in figure 2.1 for 0 degrees filter of formula (2.1).

(Berkeley Segmentation Database[1] (Martin *et al.*, 2001)) which consists of 300 color images of 321x481 and 481x321 pixels; 200 of them were converted to black and white and used to learn the fields, collecting 41x41 patches; an example image from the dataset is shown in figure 2.1.

2.2.1 Feature extraction stage

There are several models of the complex cells of V1, but we have chosen to use the classic energy model (Morrone and Burr, 1988) on the intensity channel. The response is calculated as:

$$E_\theta = \sqrt{\left(I * f_\theta^e\right)^2 + \left(I * f_\theta^o\right)^2} \tag{2.1}$$

where f_θ^e and f_θ^o are a quadrature pair of even and odd-symmetric filters at orientation θ. Our even-symmetric filter is a Gaussian second-derivative, and the corresponding odd-symmetric is its Hilbert transform. In figure 2.2 there is an example of the output of the complex cells model for the 0 degrees orientation.

Then the edges are thinned using a standard non-maximum suppression algorithm. This is equivalent to finding edges with a Laplacian of Gaussian and zero crossing. The outputs of these filters are used to construct our local tensor representation.

[1] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/`, last access 19/02/2007.

2.2.2 Tensors

In practice a second order tensor is denoted by a 2x2 matrix of values:

$$\mathcal{T} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{2.2}$$

It is constructed by direct summation of three quadrature filter pair output magnitudes as in (Knutsson, 1989):

$$T = \sum_{k=1}^{3} E_{\theta_k} \left(\frac{4}{3} \hat{n}_k^T \hat{n}_k - \frac{1}{3} I \right) \tag{2.3}$$

where E_{θ_k} is the filter output as calculated in (2.1), I is the 2x2 identity matrix and the filter directions $\hat{n}_k$ are:

$$\begin{aligned} \hat{n}_1 &= (1, 0) \\ \hat{n}_2 &= \left(\tfrac{1}{2}, \tfrac{\sqrt{3}}{2} \right) \\ \hat{n}_3 &= \left(-\tfrac{1}{2}, \tfrac{\sqrt{3}}{2} \right) \end{aligned} \tag{2.4}$$

The greatest eigenvalue λ_1 and its corresponding eigenvector e_1 of a tensor associated to a pixel represent respectively the strength and the direction of the main orientation. The second eigenvalue λ_1 and its eigenvector e_1 have the same meaning for the orthogonal orientation. The difference $\lambda_1 - \lambda_2$ is proportional to the likelihood that a pixel contains a distinct orientation.

2.3 Preliminary results

We have run our test only for a single scale, choosing the σ of the Gaussian filters equal to 2, since preliminary tests have shown that a similar version of the fields is obtained with other scales as well. Two of the obtained fields are in figures 2.3 and 2.4. It is clear that they are somewhat corrupted by the presence of horizontal and vertical orientations in any of the considered neighbors and by the fact that in each image patch there are edges that are not passing across the central pixel. On the other hand we want to learn association field for curves that do pass through the central pixel. Geisler *et al*. (Geisler *et al.*, 2001) used a human labeled database of images to infer the likelihood of finding edges with a certain orientation relative to the reference point. On the other hand, Sigman *et al*. (Sigman *et al.*, 2001) using only relative orientation and not absolute ones, could not have seen this problem. In our case we want to use unlabeled data to demonstrate that it is possible to learn from raw images and, as mentioned earlier, we do not want to consider only the relative orientations, but rather a different field for each orientation. We believe that this is the same problem that Prodöhl *et al*. (Prodöhl *et al.*, 2003) experienced using static images: the learned fields supported collinearity in the horizontal and vertical orientations but hardly in the oblique ones. They solved this problem using motion to implicitly tag only the important edges inside each patch. A similar approach is

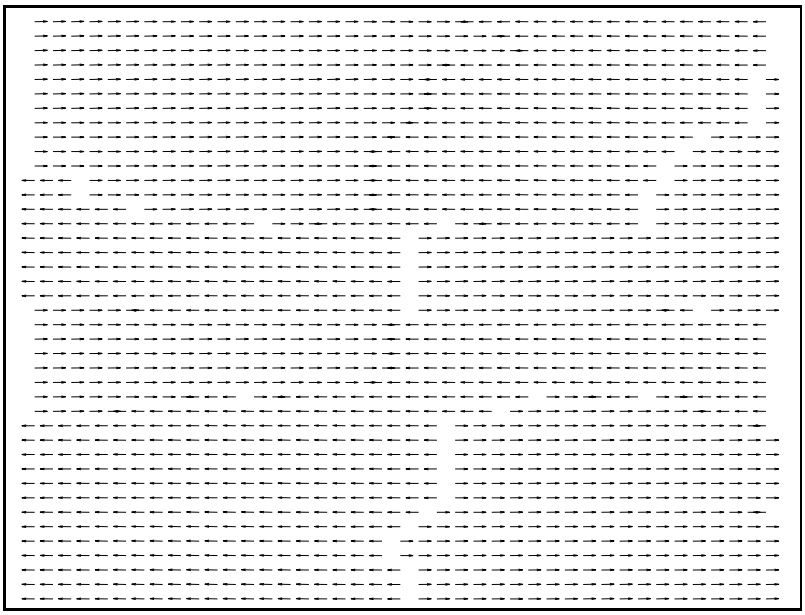

Figure 2.3: Main directions for the association field for the orientation of 0 degrees in the central pixel.

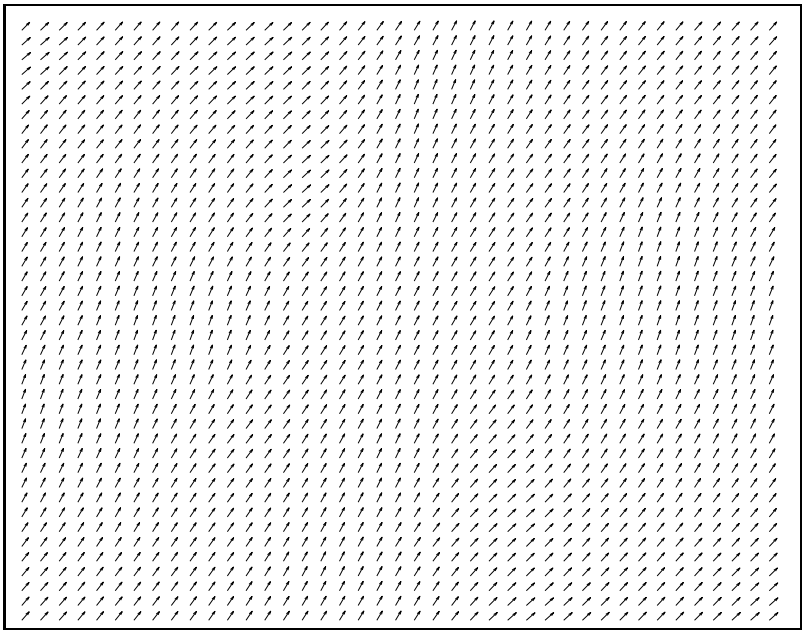

Figure 2.4: Main directions for the association field for the orientation of 67.5 degrees in the central pixel.

used by (Fitzpatrick and Metta, 2003) to disambiguate the edges of a target object from the other of the environment.

2.3.1 The path across a pixel

The neural way to solve the problem shown earlier is thought to be the synchrony of the firing between nearby neurons: if stimuli co-occur, then the neurons synchronize (Gray *et al.*, 1989). Inspired by this we considered in each patch only pixels that belong to a curve that goes through the central pixel. In this way the gathered data will contain only information about curves connected to the central pixel. Note that we select curves inside each patch, not inside the entire image. The simple algorithm used to select the pixels in each patch is the following:

1. put central pixel of the patch in a list;
2. tag first pixel in the list and remove it from the list. Put surrounding pixels that are active (non-zero) in the list;
3. if the list is empty quit otherwise go to 2.

With this procedure we remove the influence of horizontal and vertical edges that are more present in the images and that are not removed by the process of averaging. On the other hand, we are losing some information, for example about parallel lines, that in any case should not be useful for the enhancement of contours. Note that this method is completely parameter free; we are not selecting the curves following some specific criterion, instead we are just pruning the training set from some kind of noise. It is important to note that this method will learn the bias present in natural images versus horizontal and vertical edges (Coppola *et al.*, 1998), but it will not be biased to learn *only* these statistics, as in Prodöhl *et al.* (Prodöhl *et al.*, 2003) when using static images.

2.4 Results

We tested the modified procedure on the database of natural images and also on random images (results not shown), to verify that the results were not an artifact due to the method.

In figures 2.5, 2.6 there are respectively the main orientations, their strengths (eigenvalues) and the strengths in the orthogonal directions of the mean estimated tensors for the orientation of 0 degrees of the central pixel. Same for figures 2.7 and 2.8 for 67.5 degrees. The structure of the obtained association field closely resembles the fields proposed by others based on collinearity and co-circularity. We note that the size of the long-range connection far exceeds the size of the classical receptive field. We note also that the noisier regions in the orientation corresponds to very small eigenvalues so they do not influence very much the final result.

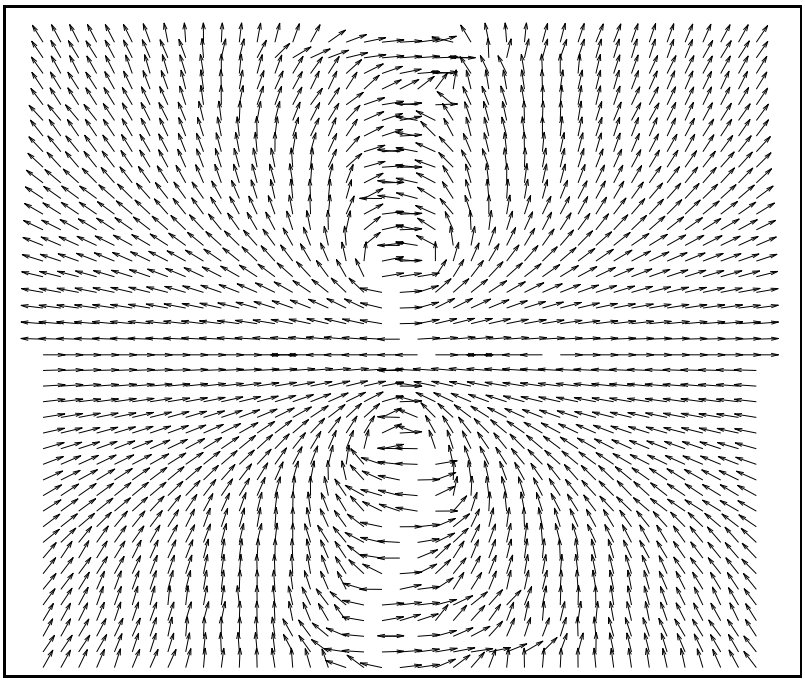

Figure 2.5: Main directions for the association field for the orientation of 0 degrees in the central pixel, with the modified approach.

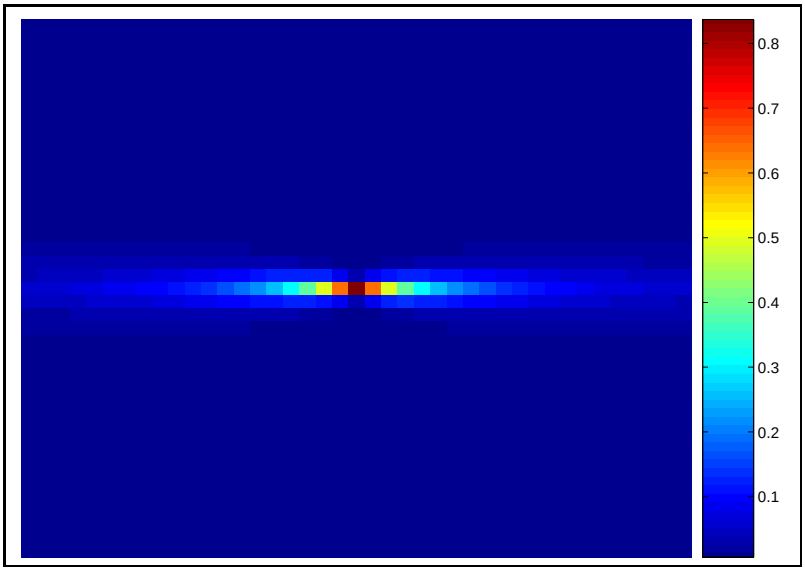

Figure 2.6: Difference between the two eigenvalues of the association field of figure 2.5.

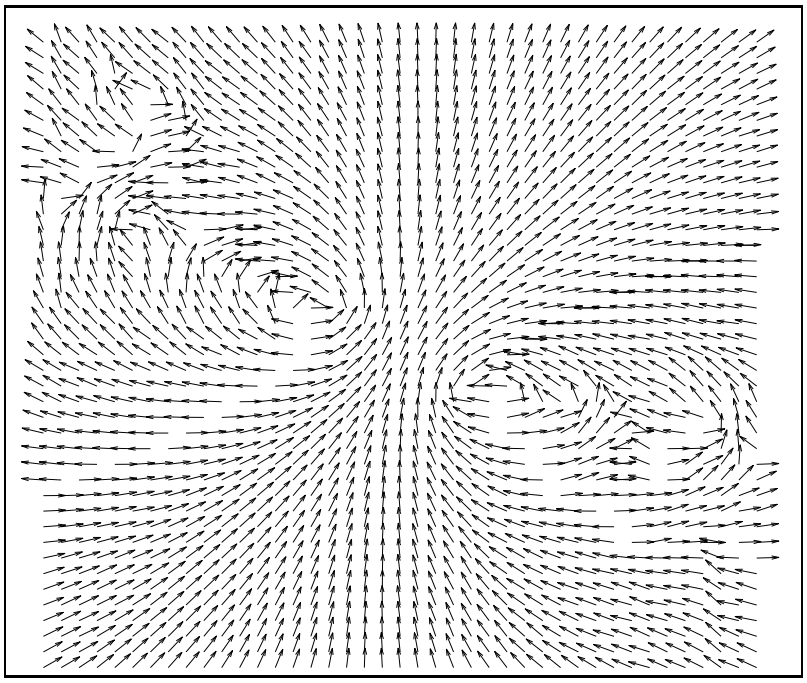

Figure 2.7: Main directions for the association field for orientation of 67.5 degrees, with the modified approach.

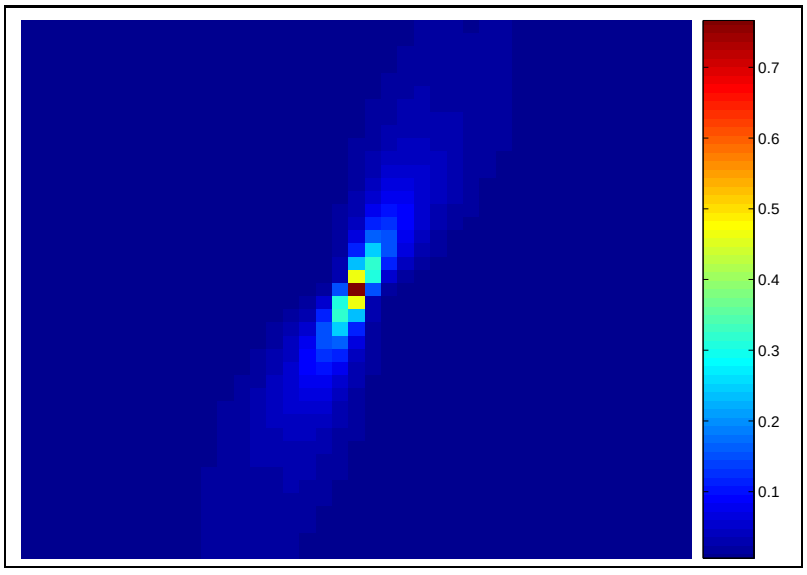

Figure 2.8: Difference between the two eigenvalues of the association field of figure 2.7.

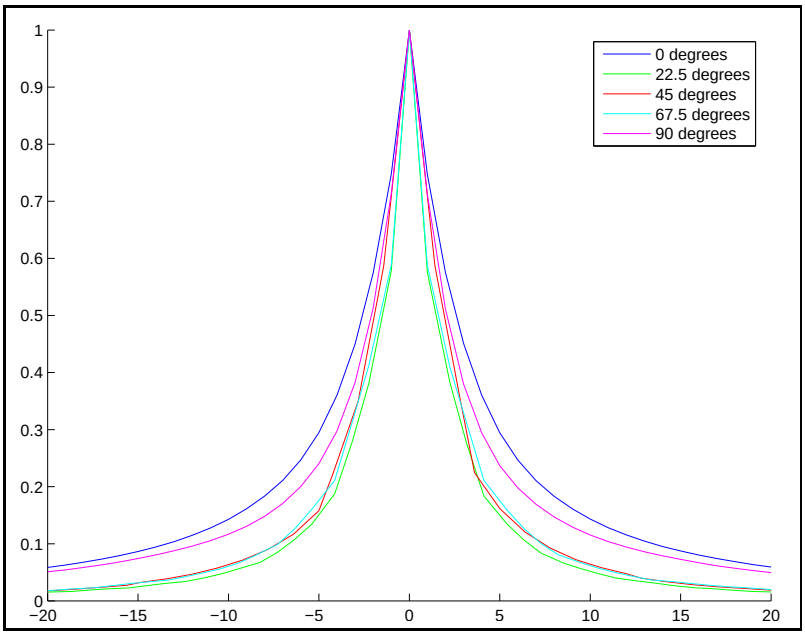

Figure 2.9: Comparison of the decay for the various orientations. On the y axis there are the first eigenvalues normalized to a maximum of 1, on the x axis is the distance from the reference point along the main field direction.

While all the fields have the same trend, there is a clear difference in the decay of the strength of the fields. To see this we have considered only the values along the direction of the orientation in the center, normalizing the maximum values to one. Figure 2.9 shows this decay. It is clear that fields for horizontal and vertical edges have a wider support, confirming the results of Sigman *et al*. (Sigman *et al.*, 2001).

2.5 Using the fields

The obtained fields can be used with any existing model of contour enhancement, but to test them we have used the tensor voting scheme proposed by Guy and Medioni *et al*. (Guy and Medioni, 1996). The choice is somewhat logical considering to the fact that the obtained fields are already tensors. In the tensor voting framework points communicate with each other in order to refine and derive the most preferred orientation information. Differently to the original tensor voting algorithm we don't have to choose the right scale of the fields (Lee and Medioni, 1999) since it is implicitly in the learnt fields. We compared the performances of the tensor voting algorithm using the learned fields versus the simple output of the complex cell layer, using the Berkeley Segmentation Database and the methodology proposed by Martin *et al*. (Martin *et al.*, 2004; Martin *et al.*, 2001). We can

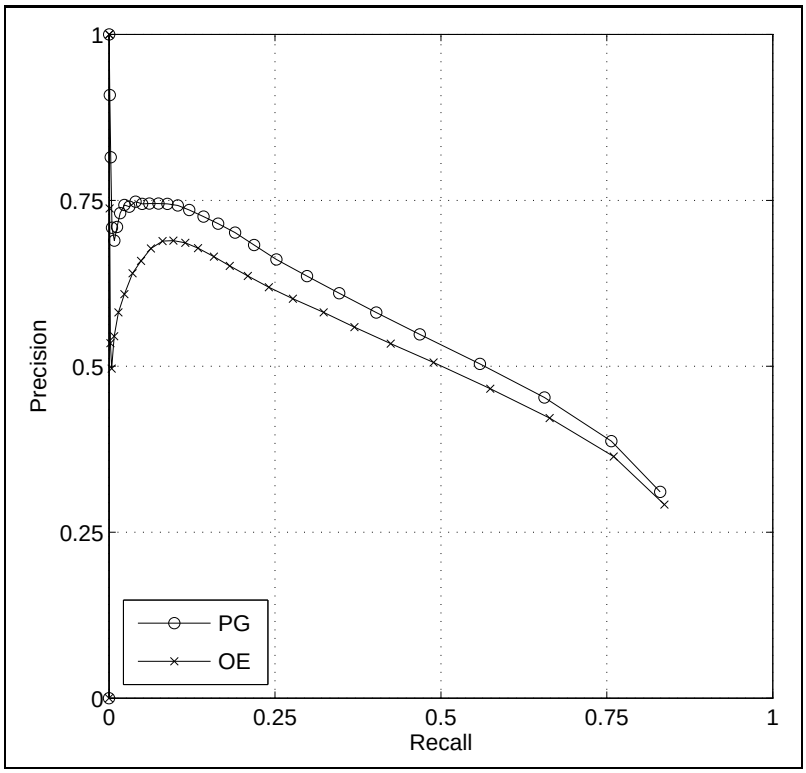

Figure 2.10: Comparison between tensor voting with learned fields (PG label) and the complex cell layer alone (OE label).

see the results in figure 2.10: there is a clear improvement using the tensor voting and the learned association fields instead of just using the simulated outputs of the complex cells alone. An example of the results on the test image in 2.1, after the non-maximum suppression procedure, are shown in figures 2.11 and 2.12.

2.6 Discussion

Several authors have studied the mutual dependencies of simulated complex cells responses to natural images. The main result from these studies is that these responses are not independent and they are highly correlated when they are arranged collinearly or on a common circle. In this chapter we have presented a method to learn precise association field from natural images. A bio-inspired procedure to get rid of the non-uniform distribution of orientations is used, without the need of a tagged database of images (Elder and Goldberg, 2002; Geisler *et al.*, 2001), the use of motion (Prodöhl *et al.*, 2003) or supposing the cortical signals sparse and independent (Hoyer and Hyvärinen, 2002). The learned fields were used in a computer model, using the tensor voting method, and the results were compared

Figure 2.11: Test image contours using the complex cell layer alone.

Figure 2.12: Test image contours using tensor voting with the learned fields. Notice the differences with the image 2.11: the contours are linker together and the gaps are reduced. Especially on the contour of back of the tiger the differences are evident.

using a database of human tagged images which helps in providing clear numerical results.

However the problem of learning useful complex features from natural images could in any case find a limit beyond these contour enhancement networks. At

least the knowledge of class to which the images belong is necessary as in (Fidler *et al.*, 2006), that have used a similar method to learn class specific combinations of basic features. Moreover the *usefulness* of a feature is not directly related to image statistics but supposes the existence of an embodied agent *acting* in the natural environment, not just perceiving it. In this sense in the future we would like to link strategies like the one used by Natale *et al*. (Natale *et al.*, 2005) and the approach described here, to link the first stages of unsupervised learning, to reduce the dimensionality of the inputs, to other stages of supervised learning for the definition of the extraction of useful features for a given task.

Chapter 3

Visual Attention

Contents

3.1	**Computational models of visual attention**	**33**
3.2	**Setup**	**37**
3.3	**The model**	**38**
3.4	**Learning about objects**	**45**
3.5	**Results**	**48**
3.6	**Discussion**	**49**

As said in the Introduction, one of the first steps of any visual system is that of locating suitable interest points, "salient regions", in the scene, to detect events, and eventually to direct gaze toward these locations. In the last few years, object-based visual attention models have received an increasing interest in the literature, the problem, in this case, being that of creating a model of "objecthood" that eventually guides a saliency mechanism. We present here an object-based model of visual attention and show its instantiation on a humanoid robot. The robot employs action to learn and define its own concept of objecthood.

The chapter is organized as follows: section 3.1 contains an introduction on the modeling of human visual attention. Section 3.2 describes the experimental setup used in the experiments. Section 3.3 details the robot's visual system and the implementation. Section 3.4 introduces the probabilistic object model and shows how this is used for object recognition. Finally in sections 3.5 and 3.6 we show experimental results and we draw some conclusions.

3.1 Computational models of visual attention

One way to study the phenomenology of visual attention is using the paradigm of visual search task. In such tasks the observer must tell the presence or absence of a target object among a number of other objects, "distractors". A dominant tradition

in visual search was initiated with a seminal paper by Treisman and Gelade (Treisman and Gelade, 1980). They argued that some primary visual properties allow a search in parallel across large displays. In such cases the target appears to 'pop out' of the display. For example there is no problem in searching for a red item amongst distractor items coloured green, blue or yellow. In other cases, the paradigmatic example being a 'feature conjunction' search for a target that is both green and a cross when distractors include red crosses and green circles, the task is much more difficult, suggesting the use of a different search strategy. Treisman and Gelade argued that in the pop-out tasks preattentional mechanisms permit rapid target detection, in contrast to the conjunction task, which was held to require a serial deployment of attention over each item in turn. They introduced an experimental paradigm that differentiated the different types of searches, measuring the time taken for an observer to make a speeded two choice decision concerning the presence or absence of a target in a visual display. Half of the displays contained a target and in the remaining the target was absent. The critical independent variable was the number of displayed items. The *search function* shows how the response time depends on this variable. The traditional interpretation of the search function is that the display-size-dependent increases shown in the search functions for conjunction searches come about through an item by item serial scan of covert attention through the display. If the display does not contain a target, it is assumed that every item in the display is scanned before a target-absent response is given. If the search is self-terminating in displays that do contain a target, then on average half the display items must be scanned before the target is found. This dichotomy serial/parallel has suggested the division of the attention in two stages: one 'preattentive' that is traditionally thought to be automatic, parallel, and to extract relatively simple stimulus properties, and other 'attentive' serial, slow, with limited processing capacity, able to extract more complex features. The 'preattentive' stage by definition is traditionally thought to precede the subsequent 'attentive' stage, with the latter by definition depending on the attentional state of the observer. Moreover, Treisman and Gelade proposed a model called Feature Integration Theory (FIT) (Treisman and Gelade, 1980), to justify their findings. The preattentive stage is modeled by a set of low-level feature maps that are extracted in parallel on the entire input image, than they are combined together by a spatial attention window operating on a master saliency map (Figure (3.1)).

The Treisman and Gelade's model is a representative of a class of models (space-based theories) that holds that attention is allocated to a region of space, with processing carried out only within a certain spatial window. Attention in this case could be directed to a region of space, even in absence of a real target. The most influential evidences for the spatial selection come from the experiments of Posner *et al*. (Posner *et al.*, 1980) and Downing and Pinker (Downing and Pinker, 1985). In a pointing experiment, they showed that anticipating the appearance of a target with a cue (for example an arrow) sped up the response of the subject. The opposite occurred, that is the subject's response was significantly slowed down, when the cue was in the wrong direction (invalid cue). This means that attention

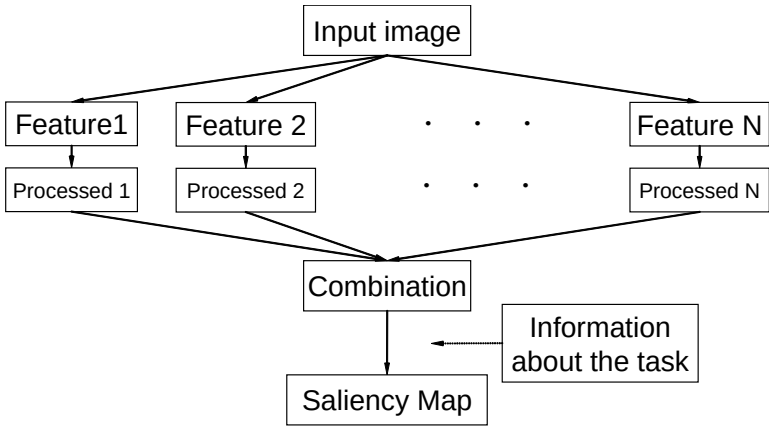

Figure 3.1: A simple schematization of the FIT model.

might be directed to a region of space even in absence of a real target. Moreover on invalid cues, the response slowed down monotonically as the distance between the cue and the actual target increased. These results suggest that attention is deployed as a spatial gradient, centered on a particular location. Hence this theory considers attention as a "spotlight", an internal eye or a sort of "zoom lens"; attention is deployed as a spatial gradient, centered on a particular location.

On the other hand there is a recent literature on the so-called 'object-based' visual attention, that represents the result of a fertile new cross-talk between two traditionally separate research fields, one concerning visual segmentation and grouping processes, and the other concerning selective attention. Object-based attention theories argue that attention is directed to an object or a group of objects, to process specific properties of the selected objects, rather than regions of space. There is a growing evidence both from behavioral and from neurophysiological studies that shows, in fact, that selective attention frequently operates on an object based representational medium in which the boundaries of segmented objects, and not just spatial position, determine what is selected and how attention is deployed (see (Scholl, 2001) for a review). This reflects the fact that the visual system is optimized for segmenting complex scenes into representations of (often partly occluded) objects to be used both for recognition and action, since perceivers must interact with objects and not with disembodied spatial locations. For example, attention to one part of an object confers an attentional advantage to other parts of that object (Egly *et al.*, 1994). Similarly, attention to one aspect of an object (*e.g.* its shape) enhances the cortical response to other aspects of that object (*e.g.* its color or motion); thus, all the attributes of an attended object seem to be bound together into a single entity. This concept holds even when the attended and ignored objects are spatially superimposed. O'Craven *et al.* (O'Craven *et al.*, 1999) have observed the effects of object-based attention using fMRI. In this study, observers looked at a display containing a sequence of semitransparent images of spatially

superimposed faces and houses. At any given moment, either the house or the face moved with an oscillatory motion. Observers were asked to decide whether the currently visible house (or face) matched the one immediately preceding it; this required them to attend closely to the relevant object type. A spatial 'spotlight of attention' could not select one of the two superimposed objects; it would necessarily select both or neither. The researchers found that activity in face- and house selective cortical regions mirrored the subject's state of attention (despite the fact that both a house and a face were present in the scene at all times), indicating that object-based selection was possible in this task. As predicted by an object-based account, all of the features of the attended object were selected, and the features of the ignored object were (relatively) suppressed.

Finally, another classification can be made depending on which cues are actually used in modulating attention. Bottom-up information, which comes only from the input image, includes basic features such as color, orientation, motion, depth, and their conjunction thereof. A feature or a stimulus catches attention if it differs from its immediate surrounding in some dimensions and the surround is reasonably homogeneous in those same dimensions. However, in attention, higher-level mechanisms are involved as well. A bottom-up stimulus, for example, may be ignored if attention is already focused elsewhere (Yantis, 1998). In this case attention is also influenced by top-down information relevant to the particular task at hand which is not necessarily available in the image (Yarbus, 1967).

In the literature a number of attention models that follow the first hypothesis have been proposed (Milanese *et al.*, 1995; Sela and Levine, 1997; Itti *et al.*, 1998), most of them being derived from Treisman and Gelade's FIT. Moreover the model proposed by Itti *et al*. (Itti *et al.*, 1998) is considered the state of the art, and, with some modifications, has been also implemented on humanoid robots, *e.g*. (Breazeal *et al.*, 2001). An important alternative model is given by Sun and Fisher (Sun and Fisher, 2003), which propose an combination of object-and feature-based theories. Presented with a manually segmented input image, their model is able to replicate human viewing behavior for artificial and natural scenes. The limit of the model is the human segmentation of the images: it supposes the use of information that could be not available in the preattentive stage, that is before the objects in the image are recognized.

For a complete review on this topic see (Itti and Koch, 2001a).

3.1.1 A proto-object based model of visual attention

The proposed model starts from the considerations that the human visual system extracts basic information from the retinal image in terms of lines, edges, local orientation etc. Vision though does not only represent visual features but also the *things* that such features characterize. In order to segment a scene in items, objects, that is to group parts of the visual field as coherent wholes, the concept of "object" must be known to the system.

The 'objects' which we will be concerned with are segmented perceptual units.

In particular, there is an intriguing discussion underway in vision science about reference to entities that have come to be known as "proto-objects" or "pre-attentive objects" (Rensink, 2000b; Rensink, 2000a; Pylyshyn, 2001), since they need not to correspond exactly with conceptual or recognizable objects. These are a step above the mere localized features, possessing some but not all of the characteristics of objects. Instead, they reflect the visual system's segmentation of current visual input into candidate objects (*i.e*. grouping together those parts of the retinal input which are likely to correspond to parts of the same object in the real world, separately from those which are likely to belong to other objects). They were introduced by Rensink in his interpretation of change blindness: observers were blind to big changes in a scene when a blank screen was shown for a few milliseconds before for the modified image (Rensink *et al.*, 1997)

The visual attention model proposed considers these first stages of the human visual processing, and employs a concept of salience based on proto-objects defined as blobs of uniform color in the image. Since we are considering an embodied system we will use the output of an instantiation of the model to control the fixation point of a robotic head. Moreover, through action, the attention system can go beyond proto-objects (Metta and Fitzpatrick, 2003). In fact, once an object is grasped, the robot can move and rotate it to build a statistical model of the features belonging to it, constructing a representation as a collection of proto-objects and their relative spatial locations. This internal representation then generates a top-down signal that bias attention toward known objects; as an example we will show how the top-down influence can be used to direct the attention of the robot to spot a specific object among other similar items lying on a table.

The proposed object-based model of visual attention integrates bottom-up and top-down cues; in particular, top-down information works as a priming mechanism for certain regions in the visual search task (*i.e*. when the robot seeks for a known object in the environment).

3.2 Setup

The experiments reported here were carried out on a robotic platform called Babybot. This is a humanoid upper torso which consists of a head, an arm and a hand. The head has 5 degrees of freedom, two of which control the neck pan and tilt, whereas the other three actuate two eyes to pan independently and tilt on a common axis. The arm is the well known Unimate PUMA 260, an industrial manipulator with 6 degrees of freedom; the hand (designed and realized at LIRA-Lab) has 5 fingers for a total of 6 degrees of freedom. From the point of view of the sensors, the head is equipped with two space-variant cameras (Sandini *et al.*, 2000) and two microphones for visual and auditory feedback. Proprioceptive information is provided to the robot by optic and magnetic encoders mounted on all joints of the head, arm and hand. More details about the Babybot can be found in (Natale, 2004).

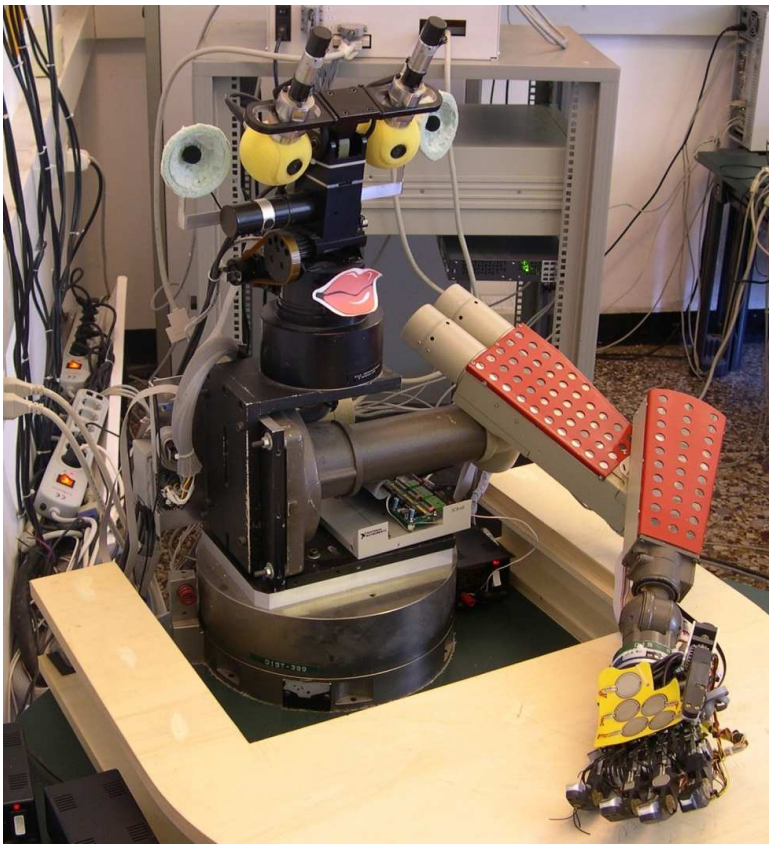

Figure 3.2: The robotic setup, Babybot. The experimental setup consists of a five degrees of freedom robot head, and an off-the-shelf six degrees of freedom robot manipulator, both mounted on a rotating base: *i.e.* the torso. The kinematics resembles that of the upper part of the human body although with less degrees of freedom.

3.3 The model

In Figure 3.3 there is a block diagram of the model; the input is a sequence of color log-polar images (Schwartz, 1977; Sandini and Tagliasco, 1980). The use of log-polar images comes from the observation that the distribution of the cones, i.e. the photoreceptors of the retina involved in diurnal vision, is not uniform. This distribution seems to influence the scanpaths during a visual search task and so it has to be taken into account to better model overt visual attention (Wolfe and Gancarz, 1996). In addition, the lower resolution of the periphery of the field of view reduces the images' size and thus reduces the computational load.

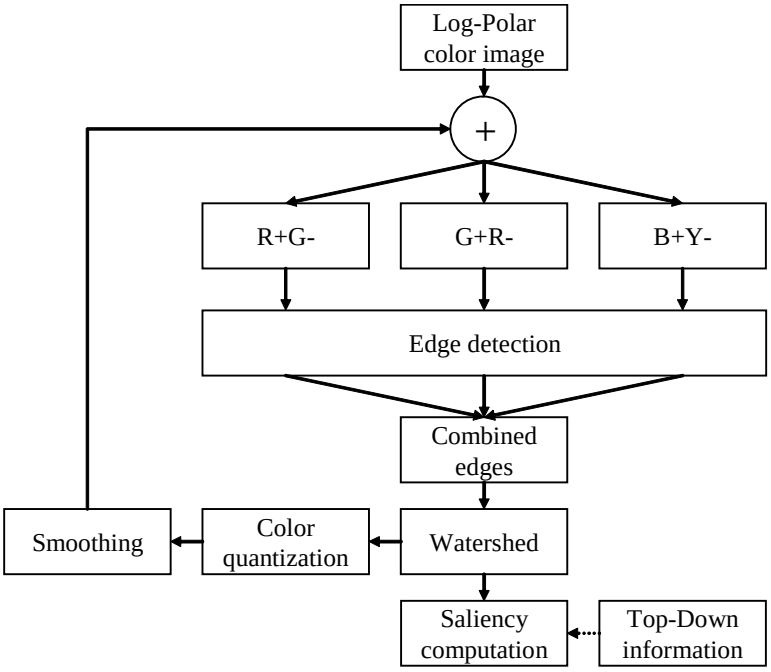

Figure 3.3: Block diagram of the model. The input image is first separated in the three color opponency maps, than edges are extracted. A watershed transform creates the clusters of uniform or uniform gradient of color (blobs). The saliency is defined on the blobs, and not on single pixels, taking into account top-down biases.

3.3.1 Log-polar images

The log-polar mapping is a model of the topological transformation of the primate visual pathways from the retina to the visual cortex. Cones have a higher density in the central region called fovea (approximately 2° of the visual field), while they are sparser in the periphery. Consequently, the resolution is higher and uniform in the center while it decreases in the periphery, moving away from the fovea. Moreover the cartesian image from the retina is deformed on the cortex through a transformation that can be well described as a logarithmic-polar (log-polar) mapping (Schwartz, 1977).

The main advantage of log-polar sensors is related to the small number of pixels and the comparatively large field of view. In fact the lower resolution of the periphery reduces the images' size and thus reduces the computational load of the visual processing, while the high resolution center can be used for standard visual algorithms (Sandini and Metta, 2002).

From the mathematical point of view the log-polar mapping can be expressed as a transformation between the polar plane (ρ, θ) (retinal plane), the log-polar plane (η, ξ) (cortical plane) and the Cartesian plane (x, y) (image plane), as follows

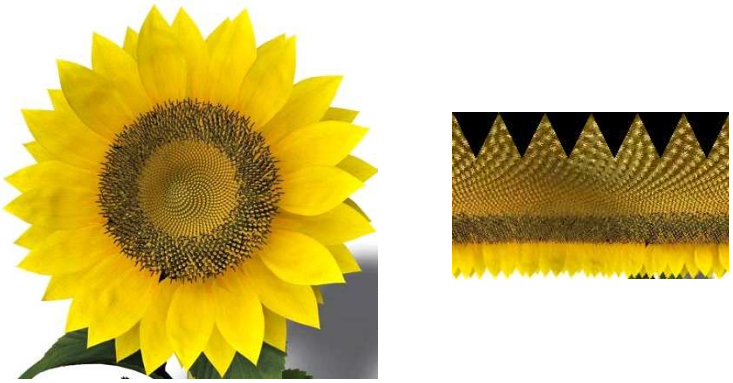

Figure 3.4: Log-polar transform of an image.

(Sandini and Tagliasco, 1980):

$$\begin{cases} \eta = q \cdot \theta \\ \xi = log_a \frac{\rho}{\rho_0} \end{cases} \tag{3.1}$$

where ρ_0 is the radius of the innermost circle, $1/q$ is the minimum angular resolution of the log-polar layout and (ρ, θ) are the polar co-ordinates. These are related to the conventional Cartesian reference system by:

$$\begin{cases} x = \rho \cdot cos\theta \\ x = \rho \cdot sin\theta \end{cases} \tag{3.2}$$

Figure 3.4 shows a Cartesian image and its log-polar counterpart as derived from Equations (3.1) and (3.2). It is worth noting that the flower's petals, that have a polar structure, are mapped vertically in the log-polar image. Circles, on the other hand, are mapped horizontally. Furthermore, the stamens that lie in the center of the image of the flower, occupy about half of the corresponding log-polar image (the cortical magnification).

3.3.2 Feature extraction

As a first step the input image at time t is averaged with the output of a color quantization procedure (see later) applied to the image at time $t-1$. This is to reduce the effect of the input noise. The red, green, blue channels of each image are then separated, and the yellow channel is constructed as the arithmetic mean of the red and green channels. Successively these four channels are combined to generate three color opponent channels, similar to those of the retina. Each channel, normally indicated as R^+G^-, G^+R^-, B^+Y^-, has a center-surround receptive field (RF) with spectrally opponent color responses. That is, for example, a red input in the center

of a particular RF increases the response of the channel R^+G^- , while a green one in the surrounding will decrease its response. The spatial response profile of the RF is expressed by a Difference-of-Gaussians (DoG) over the two sub-regions of the RF, "center" and "surround". A response is computed as there was a RF centered on each pixel of the input image, thus generating an output image of the same size of the input. This operation, considering for example the R^+G^- channel is expressed by:

$$R^+G^-(x,y) = \alpha \cdot R * g_c - \beta \cdot G * g_s \tag{3.3}$$

The two Gaussian functions, g_c and g_s , are not balanced: the ratio β/α is chosen equal to 1.5, consistent with the study of Smirnakis *et al*. (Smirnakis *et al.*, 1997). The unbalanced ratio preserves the achromatic information: that is, the response of the channels to a uniform gray area is not zero. Hence the model does not need to process achromatic information explicitly since it is implicitly encoded, similarly to what happens in the human retina's P-cells (Billock, 1995). The ratio σ_s/σ_c, the standard deviation of the two Gaussian functions, is chosen equal to 3. To be noted that by filtering a logpolar image with a standard space-invariant filter leads to a space-variant filtered image of the original cartesian image (Mallot *et al.*, 1990). Edges are then extracted on the three channels separately using a generalization of the Sobel filter due to (Li *et al.*, 2003), obtaining $E_{RG}(x,y)$, $E_{GR}(x,y)$ and $E_{BY}(x,y)$. A single edge map is generated combining the tree outputs:

$$E(x,y) = \max\left\{\left|E_{RG}(x,y)\right|, \left|E_{GR}(x,y)\right|, \left|E_{BY}(x,y)\right|\right\} \tag{3.4}$$

The log-polar transform has the side effect of sharpening the edges near the fovea due to the magnification factor of the mapping; this is compensated multiplying each pixel by a factor which is exponential on the eccentricity.

3.3.3 Proto-objects

It has been speculated, that synchronizations of visual cortical neurons might serve as the carrier for the observed perceptual grouping phenomenon (Eckhorn *et al.*, 1988; Gray *et al.*, 1989). The differences in the phase of oscillation among spatially neighboring cells are believed to contribute to the segmentation of different objects in the scene. We have used a watershed transform (rainfalling variant) (Vincent and Soille, 1991; De Smet and Pires, 2000) on the edge map to simulate the result of this synchronization phenomenon and to generate the proto-objects. The intuitive idea underlying this method comes from geography: a topographic relief is flooded by water, watershed are the divide lines of the domains of attraction of rain falling over the region. In our view the watershed transform simulates the parallel spread of the activation on the image, until this procedure fills all the spaces between edges. Differently from other similar methods the edges themselves will never be tagged as blobs and the method does not require complex membership functions either. Moreover the result does not depend on the order in which the points are examined like in standard region growing (Wan and Higgins, 2003). As a result,

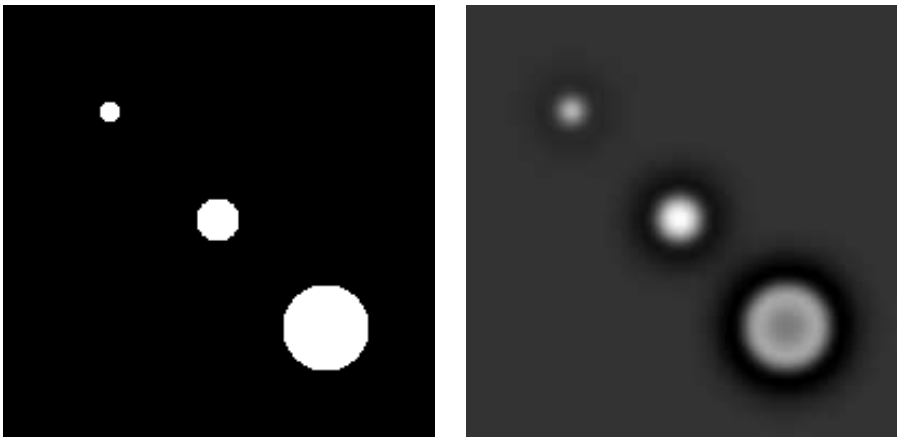

Figure 3.5: Filtering the image on the left with a Difference of Gaussians with the size of positive lobe equal to the size of the circle in the middle, we obtain the image on the right. Smaller blobs will be depressed while larger ones will be depressed in their centers.

the image is segmented into blobs with either uniform or uniform gradient of color. Hence from the choice of the feature maps come our definition of proto-objects as closed areas of uniform color of the image. Each blob is tagged with the average of the color of the pixels within its area (this leads to a sort of color quantized image). The result is blurred with a Gaussian filter and stored: this will be used to perform a time-smoothing by simple averaging with the frame at time $t + 1$ to reduce the effect of noise and increase the temporal stability of the blobs. After an initial startup time of about five frames, the number of blobs and their shape stabilize. If movement is detected in the image (as difference between two consecutive frames) then the smoothing procedure is halted and the bottom-up saliency map becomes the motion image.

As already mentioned above, a feature or a stimulus catches the attention of the system if it differs from its immediate surrounding. We chose to compute the bottom-up salience as the Euclidean distance in the color opponent space between each blob and its surrounding. The size of the spot or focus of attention is not constant: it changes depending on the size of the objects in the scene. To account for this fact the greater part of the visual attention models in literature uses a multi-scale approach filtering with some type of "blob" detector (typically a difference of Gaussian filter) at various scales (Itti and Koch, 2001a). We reasoned that this approach lacks continuity in the choice of the size of the focus of attention (see for example Figure (3.5)). We propose instead to dynamically vary the region of interest depending on the size of the blobs. That is the salience of each blob is calculated in relation to a neighborhood proportional to its size. In our implementation we consider a rectangular region 3 times the size of the bounding box of the

blob as surrounding region, centered on each blob. The choice of a rectangular window is not incidental, rather it was chosen because filters over rectangular regions can be computed efficiently by employing the integral image as in (Viola and Jones, 2004).

The bottom-up saliency is thus computed as:

$$S_{bottom-up} = \sqrt{\Delta RG^2 + \Delta GR^2 + \Delta BY^2} \tag{3.5}$$
$$\Delta RG = \langle R^+G^- \rangle_{blob} - \langle R^+G^- \rangle_{surround} \tag{3.6}$$
$$\Delta GR = \langle G^+R^- \rangle_{blob} - \langle G^+R^- \rangle_{surround} \tag{3.7}$$
$$\Delta BY = \langle B^+Y^- \rangle_{blob} - \langle B^+Y^- \rangle_{surround} \tag{3.8}$$

where $\langle\rangle$ indicates the average of the image values over a certain area (indicated in the subscripts). The top-down influence on attention is, at the moment, calculated in relation to the task of visually searching for a given object. In this situation a model of the object to search in the scene is given (see Section 3.4) and this information is used to bias the saliency computation procedure. In practice, the top-down saliency map is computed as the Euclidean distance in the color opponent space, between each blob's average color and the average color of the target:

$$S_{top-down} = \sqrt{\Delta RG^2 + \Delta GR^2 + \Delta BY^2} \tag{3.9}$$
$$\Delta RG = \langle R^+G^- \rangle_{blob} - \langle R^+G^- \rangle_{object} \tag{3.10}$$
$$\Delta GR = \langle G^+R^- \rangle_{blob} - \langle G^+R^- \rangle_{object} \tag{3.11}$$
$$\Delta BY = \langle B^+Y^- \rangle_{blob} - \langle B^+Y^- \rangle_{object} \tag{3.12}$$

with a notation similar to the one above. Blobs that are too small (1/550 of image area) or too big (1/4 of the image area) are discarded from the computation of salience and will not be considered as possible candidates to be part of objects. The blob in the center of the image (currently fixated) is also ignored because it cannot be the target of the next fixation. The total salience is simply calculated as the linear combination of the top-down and bottom-up contributions:

$$S = k_{td} \cdot S_{top-down} + k_{bu} \cdot S_{bottom-up} \tag{3.13}$$

and normalized in the range 0-255. The center of mass of the most salient blob is selected for the next saccade.

An example of the intermediate and final maps of bottom-up salience is shown in Figure 3.6. All the computations are done on log-polar images, but input and output images are shown remapped to cartesian for clarity.

3.3.4 Inhibition of return

In order to avoid being redirected immediately to a previously attended location, a local inhibition is transiently activated in the saliency map. This is called "inhibition of return" (IOR) and it has been demonstrated in human visual psychophysics.

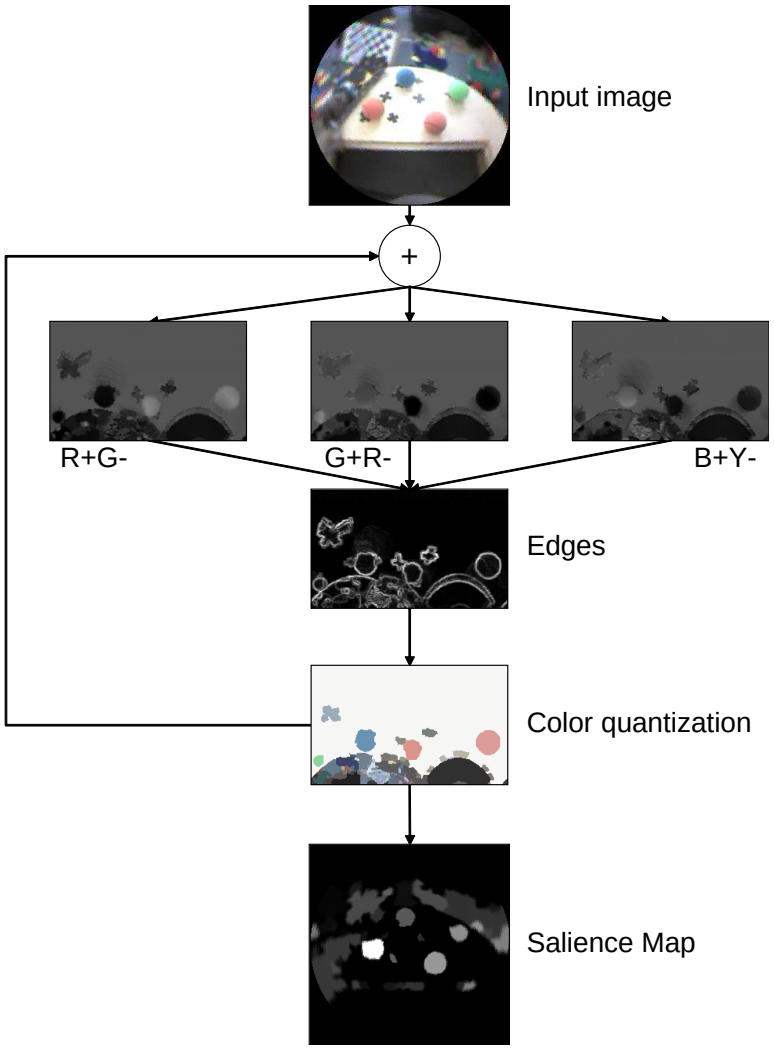

Figure 3.6: Example of model maps.

Posner and Cohen (Posner and Cohen, 1984), for example, demonstrated that the IOR does not seem to work in retinal coordinates but it is instead represented in an allocentric reference frame. Together with Klein (Klein, 1988), they proposed that the IOR is required to allow an efficient visual search by discouraging shifting the attention toward locations that have already been inspected. Static scenes, however, are seldom encountered in real life: objects move and a "tagging system" that merely inhibited environmental locations would be almost useless in any real situation. Tipper (Tipper, 1991) was among the firsts to demonstrate that the IOR could be attached to moving objects, and this finding has been replicated and extended ever since (Abrams and Dobkin, 1994; Gibson and Egeth, 1994; Tipper,

1994). These results bring to the conclusion that in humans the inhibition of return works by anchoring tags to objects as they move; in other words this process seems to be coded in an object-based reference frame.

Our system implements a simple object-based IOR. A list of the last five positions visited (Wolfe, 2003) is maintained in a head-centered coordinate system and updated with a FIFO (First In First Out) policy. The position of the tagged blob is stored together with the information about its color. When the robot gaze moves — for example by moving the eyes and/or the head — the system keeps track of the blobs it has visited. These locations are inhibited only if they show the same color seen earlier: so in case an inhibited object moves or its color changes, the location becomes available for fixation again.

3.4 Learning about objects

State of the art models of visual attention are usually used as sort of filters for object recognition systems, as in, *e.g.*, (Walther and Koch, 2006). In such systems the attention model and the object recognition one live in two different worlds, that is, they work on two different representations of the input images and few or none of the computation done by the first stage is used by the second one. Here we will show an attempt to build an object recognition system on the same basis of the visual attention, that is on the concept of proto-objects; it is clear that the performances will heavily depend on their definition. In Section 3.3.3 we said that the proto-objects are defined as closed areas of uniform color, hence the object representation is a collection of areas of uniform colors. The proposed method is just a proof of concept, for better object recognition systems see Chapter 5.

We assume the robot has already grasped the object; this can happen because a collaborative human has given the object to the robot or because it has autonomously grasped the object (even by chance initially). Both solutions are valid bootstrapping behaviors for the acquisition of an internal model of the object. When the robot holds the object it can explore it by moving and rotating it. Objects are represented by the blobs generated by the visual attention system and their relative positions (neighboring relations). The model is created statistically by looking at the same object for some time from different points of view. A histogram of the number of times a particular blob is seen is used to estimate the probability that the blob belongs to the grasped object. In the following, we use the probabilistic framework proposed by Schiele and Crowley (Schiele and Crowley, 1996). We want to calculate the probability of the object O given a certain local measurement M. This probability $P(O|M)$ can be calculated using Bayes' formula:

$$P(O|M) = \frac{P(M|O)P(O)}{P(M)} \tag{3.14}$$

where: $P(O)$ the a priori probability of the object O, $P(M)$ the a priori probability of the local measurement M, and $P(M|O)$ is the probability of the local

measurement M when the object O is fixated. In the following experiments we only carried out a detection experiment for a single object, there are consequently only two classes, one representing the object and another representing the background. Not knowing $P(O)$ and $P(\neg O)$ we set them to 0.5, in this way we do MAP estimation. Since a single blob is not discriminative enough, we considered the probabilities of observing pairs of blobs instead. To simplify the probability estimation (the number of possible combinations) we have chosen to observe only pairs composed of the central blob (taken as reference) and one surrounding blob as the local measurement M:

$$P(M|O) = P(B_i|B_c \; and \; (B_i \; adjacent \; B_c)) \tag{3.15}$$

where B_i is the i-th blob that surrounds the central blob B_c that belongs to the object O. That is, we exploit the fact the robot is fixating the object and assume the central blob will be constant across fixations. The color of the central blob will be stored and used to bias the visual search (see Section 3.3.3). The probabilities $P(M|\neg O)$ are estimated during the exploration phase by considering the blobs not adjacent to the central blob. The local measurements are considered independent because they refer to different blobs, so we factorize the total probability $P(M_1, \cdots, M_N|O)$ in the product of the probabilities $P(M_i|O)$. An object is considered 'found' if the probability $P(O|M_1, \cdots, M_N)$ is greater than a fixed threshold. When the object is found after visual search, a figure-ground segmentation is attempted: each blob is selected if it is adjacent to the central recognized blob and if its probability to belong to the object is greater than 0.5. In practice, we estimate the probability of all blobs adjacent to the central blob to belong to the object. This procedure, although requiring the "active participation" of the robot (through gazing) is faster than estimating all probabilities for all possible pairs of blobs of the fixated object. Estimation of the full joint probabilities would require a larger training set than the one we were able to use in our experiments. Our experimental scenario required the construction of the object model on the fly with the shortest possible exploration procedure, which naturally leads to estimating probabilities with few samples. It is likely that many bins in the histograms, used to estimate probabilities, are empty. To overcome this problem we have used a probability smoothing method. In particular we employed as zero count smoothing the Lidstone's law of succession:

$$P(M|O) = \frac{count(M \wedge O) + \lambda}{count(O) + \nu\lambda} \tag{3.16}$$

for a ν valued problem. With $\lambda = 1$ and a two valued problem ($\nu = 2$), we obtain the well-known Laplace's law of succession. Following the results of Kohavi *et al*. (Kohavi *et al.*, 1997), we choose $\nu = 1/n$ where n is equal to the number of images utilized during the training phase. A first use of the system is to create a visual model of the hand of the robot (a special object). By relying on this model the robot can distinguish the grasped object from parts of the hand that might still be visible.

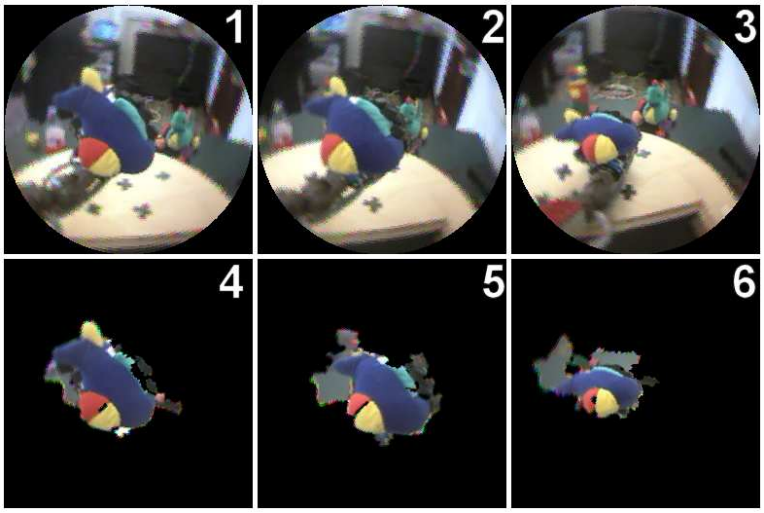

Figure 3.7: Some example images during exploration phase (1-3) and related segmentations (4-6) used to build the statistical model of the object. Note how the parts not of the object are not always detected, so their estimated probability to belong to the object will be low.

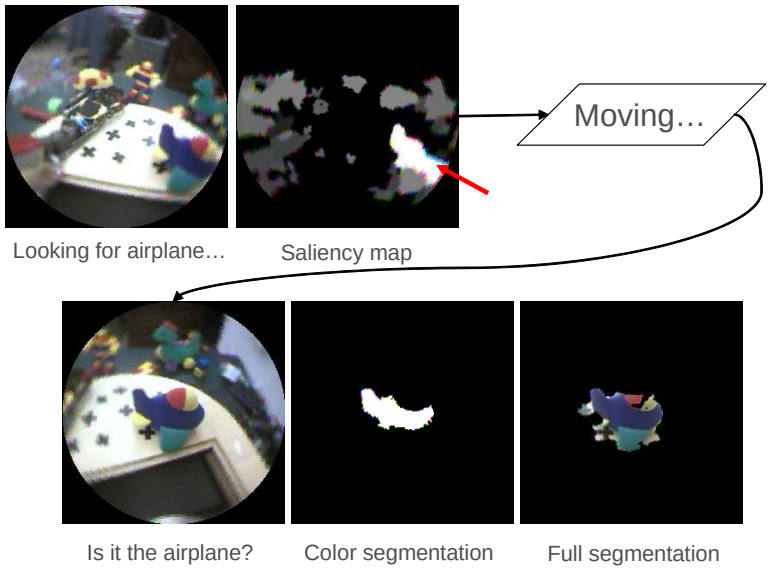

Figure 3.8: The flow chart of the visual search of an object (the toy airplane), recognition and segmentation. The saliency map is generated using the information about the color blue of the toy.

Object	Recognition rate	Number of saccades when recognized
Toy car	94%	3.19 ± 2.17
Toy airplane	88%	3.02 ± 2.84

Table 3.1: Performance of the recognition system measured from a set of 50 trials.

3.5 Results

The behavior of the robot during the learning phases is shown in Figure 3.7: all the blobs bordering the central one (blue) are used for learning the visual appearance of the object. Two examples of the saliency map are shown in Figure 3.9: in 3.9.4 there is a purely bottom-up ($k_{td} = 0$, $k_{bu} = 1$ in Equation (7)) map which is the result of the processing of the scene in 3.9.1; in 3.9.5 there is a purely top-down ($k_{td} = 1$, $k_{bu} = 0$) map output after the processing of 3.9.2. In the latter the robot was instructed to search for the toy airplane. After a saccade on the object and a successfully recognition the figure-ground segmentation is shown in Figure 3.9.6. The center of mass of the segmented object is used to guide the grasping action of the robot. Even if the result is not visually perfect, it has all the information to guide a manipulation task. In fact the perceptual system is not intended as stand alone, but strictly coupled with the action counterpart; however the segmented image could be improved with a stage of refinement of the borders. We have tested the attention system while guiding the recognition and grasping of objects in the Babybot. In table 3.5, results are shown when using a toy car and a toy airplane as target objects; 50 training/visual search sessions were performed for each object. The first column shows the recognition rate, the second the average number of saccades (mean $\pm$ standard deviation) it takes the robot to locate the target in case of successful recognition.

In order to compare the performance of the system with the state of the art model of Itti, we have done a comparison test of the bottom-up attention using the database of images by Itti and Koch (Itti and Koch, 2001b) (color images with an emergency triangle and relative binary segmentation masks of the triangle), which is freely available on the Internet[1]. First, the original images and segmentation masks are cropped to a square and transformed to the log-polar format (252x152 pixels) (see Figure 3.10.1 and Figure 3.10.2 for the cartesian remapped images). To simulate the presence of a static camera, the images are presented to the system continuously and, after five "virtual" frames, the bottom-up saliency map is confronted with the mask. In 49% of the images a point inside the emergency triangle is selected as the most salient (see an example in Figure 3.10.3). It is worth noting that a direct comparison with the results of Itti and Koch, by counting the number of false detection before the target object is found, is not possible since after each

[1] `http://ilab.usc.edu/imgdbs/`, last access 19/02/2007.

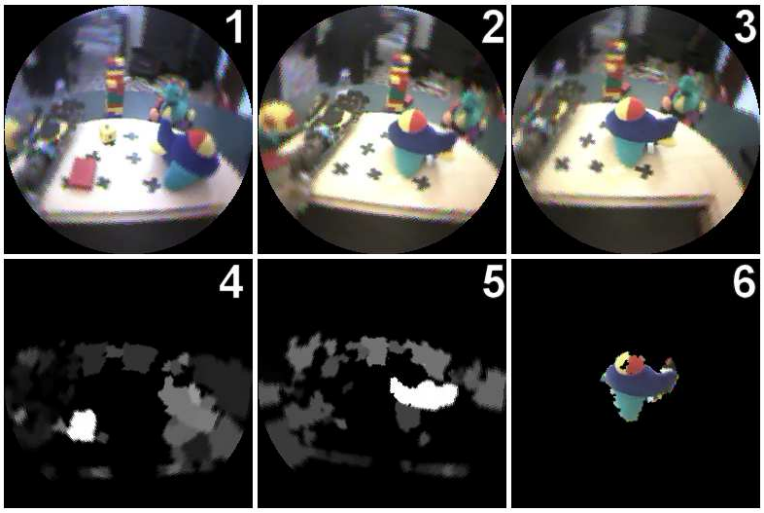

Figure 3.9: Example saliency maps. In (4) there is the bottom-up saliency map of the image (1). In (5) the top-down saliency map of (2), while searching for the blue toy airplane. Image (6) is the figure-ground segmentation of the image in (3), after having recognized the object.

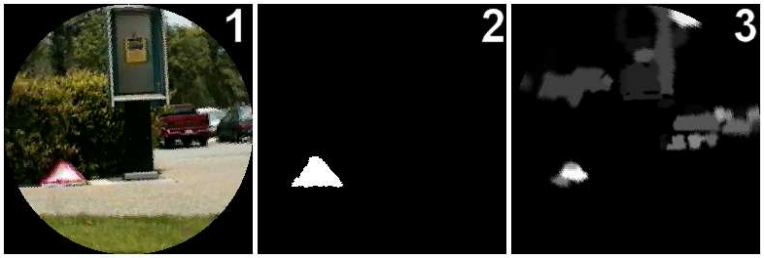

Figure 3.10: Result on a static example image taken from the database by Itti and Koch. Image (1) is the log-polar input image; image (2) is the binary mask used for to verify the correct localization of the target object and image (3) is the saliency map generated by the system.

saccade the log-polar image is heavily deformed.

3.6 Discussion

We have presented the implementation of a visual attention system employing both top-down and bottom-up information. It runs in real time on a standard Pentium class processor and it is used to control the overt attention mechanism of a humanoid robot. This eventually gives rise to a different sort of problems compared

to the more typical implementations that only generate scan paths on static images. The algorithm divides the visual scene in color blobs; each blob is assigned a bottom-up saliency value depending on the contrast between its color and the color of the surrounding area. The robot acquires information about objects through active exploration and uses it in the attention system as a top-down primer to control the visual search of that object. The model directs the attention on the proto-object's or segmented object's center of mass (see Section 3.3.3 and Section 3.5), similarly to the behavior observed in humans. In fact it has been observed that the first fixation to a simple shape that appears in the periphery tends to land on its center of gravity (Melcher and Kowler, 1999). When the camera moves, a new blob will appear in the image center. This active behavior simplifies the segmentation and the recognition task since there will always be a blob in the center that will be segmented from the background. A similar approach has been taken by Sun and Fisher (Sun and Fisher, 2003) but the main difference with this work is that they have assumed that a hierarchical set of perceptual groupings is provided to the attention system by some other means and considered only covert attention. On the other hand, our system has been shown in practice to be useful in guiding a humanoid robot in selecting objects to be grasped, by helping the visual search and recognition task. Moreover the framework introduced is general enough to work with other additional feature maps, extending the watershed transform to additional dimensions in feature space (e.g. local orientation) thus providing new ways of both segmenting and recognizing objects. As future work we want to integrate the associative fields learnt from natural images (see previous Chapter) to obtain better proto-objects.

Chapter 4

Learning and Support Vector Machines

Contents

4.1 Support Vector Machines 52
4.2 Background Mathematics 54
4.3 Previous work 56
4.4 Sparseness of the solution 58
4.5 Online Independent Support Vector Classification 59
4.6 Experimental Results 64
4.7 Discussion 67

THERE are many machine learning approaches that aim to reproduce the performances of humans, for example, in classification tasks. Generally speaking, considering the supervised learning framework, some samples with their labels (the identification of the class to which they belong) are fed to the machine as input. After a training on the given examples, the machine should be able to indicate the class of an unseen sample, possibly indicating also the prediction's degree of confidence. The training phase often consists in finding an optimal separating surface in the input space between the samples of different classes (Duda *et al.*, 2000). In general, it is possible to separate two clouds of points in infinite ways and different machine learning algorithms are defined by different optimality criterions.

Support Vector Machines (SVMs) are one of these methods, rooted in statistical learning theory. In the SVM framework the classification is done maximizing the margin separating both classes while minimizing the classification errors. One of their most interesting characteristics is that the solution achieved during training is *sparse*. This means that a few samples are usually considered "important" by the

algorithm (the so-called *support vectors*) and give account of the complexity of the classification/regression task.

It is natural to ask if humans use a similar internal method to learn from examples and to classify new stimuli. At least for face recognition task, it is possible to answer to this question, in fact it has been demostrated that using SVM the distance of a face to the separating hyperplane is an almost perfect predictor of the human classification performance (Graf *et al.*, 2006). In that study SVM resulted the best candidates to model human internal classification algorithms, while the prototype classifier, as well as its piecewise linear extension seemed to be least adapted for the task. Moreover the prototype classifier behaved in the least human-like manner. Hence it seems that algorithms such as the SVM better capture the human internal face space. A classification algorithm using the center of the classes, such as for the prototype classifier, seems thus less adapted to model human classification behavior than a classifier maximizing the margin between the classes such as the SVM.

Even if SVM has been applied to different domains with excellent results and it seems to be close to human learning algorithm, it has the disadvantage to "grow" for ever. That is the number of support vectors grows proportionally with the number of training samples, thus it is impossible to have a lifelong training like in humans. Due to the big number of support vectors they can be up to 50 slower of other specialized approaches with similar performances (Burges and Schölkopf, 1996). Given that both the training and testing time crucially depend on the number of support vectors, and it is then very important to keep it small. In recent literature this has become a key issue in order to speed up SVMs without losing accuracy. We propose a new algorithm called Online Independent Support Vector Machines (OISVM), an incremental way of building the minimal solutions, based upon linear independence in the feature space. Experiments reveal that our machines achieve a dramatic reduction in the number of support vectors without losing accuracy, and mathematically assuring to reach a limit in the number of support vectors.

This chapter is structured as follows: in Section 4.1 and 4.2 we introduce SVM and their mathematical background, then in Section 4.3 there is a review of the relevant literature; in Section 4.4 some considerations on the sparseness of SVM solutions are stated. In Section 4.5 then, we describe OISVM; in Section 4.6 we show some experimental results, and lastly in Section 4.7 conclusions are drawn.

4.1 Support Vector Machines

In a number of research fields as diverse as, e.g., bioinformatics, data mining and robotics, it is crucial to be able to reconstruct an unknown function given a finite set of samples and the values the function assigns them. Given this very general problem, statistical learning theory (Vapnik, 1998; Poggio and Smale, 2003) can tell us how close our approximation is to the original function, and give us an indication of how well it will work. Usually, a set of samples of the unknown function

is available, and then a machine learning algorithm is employed to interpolate the data.

Introduced in the early 90s by Boser, Guyon and Vapnik (Boser *et al.*, 1992), *Support Vector Machines* (SVMs) are a class of machine learning algorithms deeply rooted in Statistical Learning Theory (Vapnik, 1998), able to classify data taken from an unknown probability distribution, given a set of training examples. As opposed to analogous methods such as, e.g., artificial neural networks, they have the main advantages that (a) training is guaranteed to end up in a global minimum, (b) their generalization power is theoretically well founded, (c) they can easily work with highly dimensional, non-linear data, and (d) the solution achieved is sparse. Due to these good properties, they have been now extensively used in, e.g., speech recognition, object classification and function approximation (Cristianini and Shawe-Taylor, 2000). On the other hand, one of their main drawbacks is their alleged inability to cope with large datasets (Keerthi *et al.*, 2006).

4.1.1 The importance of online learning

Yet, in most real-life applications, datasets *are* large, for example when online learning must be performed. Online learning is a scenario in which training data is provided one example at a time, as opposed to the batch mode in which all examples are available at once (see (Laskov *et al.*, 2006) and citations therein). In fact, the classical approach to machine learning is to use all the available data at once, train on this data and then use the trained machine. Note that after the predictor is obtained, it stays fixed and is not updated as new data arrive. In contrast, an on-line prediction algorithm can take advantage of the fact that the training set is augmented one sample at a time and continues to update and improve the model as more data arrive. Hence, in the case of, e.g., non-stationary data, online algorithms will generally perform better since ambiguous information (i.e., whose distribution varies over time) is present, and couldn't possibly be taken into account by the batch algorithm. Online algorithms allow to incorporate additional training data, when it is available, without re-training from scratch. Moreover using online learning it is possible to exploit to possibility of active learning algorithms, *e.g.* (Li and Sethi, 2006), that actively select the new data point that will be added to the training set. Active learning algorithm are known to require less samples to converge to a good solution (Duda *et al.*, 2000). This last topic is very interesting if seen in the light of the tight coupling between action and perception (see Section).

Moreover in an online setting there is no guarantee that the flow of data will *ever* cease; therefore, applying SVMs here looks appealing but we need a way to cope with large datasets. One of the keys to the problem seems to lie in the sparseness of their solution. That an SVM solution is *sparse* means that usually just a few samples account for its complexity; in fact, SVMs can be seen as a way of compressing data by selecting "the most important" samples (*support vectors*, SV) among those in the training set. Keeping the number of SVs small without losing accuracy is therefore a major challenge. This is even more relevant since

a recent result (Steinwart, 2003) shows that this number grows indefinitely with the number of training samples, and the testing time — a central issue in online learning, since one might want to test in real time — crucially depends on it.

Following related literature, we propose a method of selecting support vectors based upon *linear independence in the feature space*: support vectors which are linearly dependent on already stored ones are rejected, and a smart, incremental minimization algorithm is employed to find the new minimum of the cost function. Our experiments indicate that SVMs employing this idea, that we will call *Online Independent Support Vector Machines* (OISVMs), do not grow linearly with the training set but reach a limit size and then stop growing, *while keeping the full accuracy of standard SVMs* in the case of finite-dimensional feature spaces and with a negligible loss in accuracy in the infinite-dimensional case.

4.2 Background Mathematics

Assume $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, with $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \{-1, 1\}$, is a set of samples drawn from an unknown probability distribution; we want to find a function $f(\mathbf{x})$ such that $sgn(f(\mathbf{x}))$ best determines the category of any future sample $\mathbf{x}$. Assuming the data are linearly separable, according to the standard approach, a *separating hyperplane* in $\mathbb{R}^m$ is sought for:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{4.1}$$

with $\mathbf{w} \in \mathbb{R}^m$ and $b \in \mathbb{R}$. In this case, the hyperplane must respect the constraints $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geq 0$, for all $i = 1, \ldots, l$ (from now on, this will be implicit whenever a subscript i appears free in a formula). In the general, more likely and realistic case in which the data are not linearly separable, we introduce l slack variables ξ_i and rather require that $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \geq 0$, with $\xi_i \geq 0$. In order to find such a hyperplane, we wish to maximize the hyperplane's distance from both groups of samples (*margin*), minimizing at the same time the values of the slack variables. The margin is easily determined to be $\frac{2}{||\mathbf{w}||}$, so we are left with the problem of minimizing $||\mathbf{w}||$ and ξ_i subject to the above constraints. The problem is then usually solved minimizing the following expression:

$$\min_{\mathbf{w},b} \left(||\mathbf{w}||^2 + C \sum_{i=1}^{l} \xi_i^p \right) \tag{4.2}$$

subject to the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \tag{4.3}$$

where $C \in \mathbb{R}^+$ is an error penalty coefficient and p is usually 1 or 2 (Cristianini and Shawe-Taylor, 2000). Since both the problem and the constraints are convex,

(4.2) and (4.3) can be compactly expressed in Lagrangian form by introducing l pairs of coefficients α_i, μ_i and then minimizing the objective function

$$L_P = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i\right) + C\sum_{i=1}^{l} \xi_i^p - \sum_{i=1}^{l} \mu_i \xi_i^p \quad (4.4)$$

subject to the constraints that $\alpha_i, \mu_i \geq 0$. Using the KKT conditions (Cristianini and Shawe-Taylor, 2000), that gives us *necessary and sufficient* conditions for $\mathbf{w}, b$ and α_i to be be a solution, we obtain for the case $p = 1$

$$\begin{aligned}
\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i &= 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i && (4.5)\\
\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i &= 0 && (4.6)\\
\frac{\partial L_P}{\partial b} = \sum_{i=1}^{l} \alpha_i y_i &= 0 && (4.7)\\
\alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i\right) &= 0 && (4.8)\\
\xi_i(\alpha_i - C) &= 0 && (4.9)
\end{aligned}$$

and for $p = 2$ the condition (4.9) disappears while condition (4.6) becomes

$$\frac{\partial L_P}{\partial \xi_i} = C\mu_i - \alpha_i = 0 \quad (4.10)$$

Substituting Equation (4.5) in (4.1), gives

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x} \cdot \mathbf{x}_i + b \quad (4.11)$$

An example of the optimal separating hyperplane for a simple 2-dimensional problem is shown in Figure 4.1.

Notice that, in the last Equation and in Equation (4.4), the $\mathbf{x}$'s only appear in the form of inner products; in order to boost the expressive power of SVMs then, the $\mathbf{x}_i$s are usually mapped to a highly, possibly infinite-dimensional space (the *feature space*) via a non-linear mapping $\Phi(\mathbf{x})$; the core of the SVM becomes then the so-called *kernel function* K such that $K(\mathbf{x}_1, \mathbf{x}_2) = \Phi(\mathbf{x}_1) \cdot \Phi(\mathbf{x}_2)$. This idea is called *kernel trick* and is standard in SVM literature; it avoids the necessity of explicitly knowing Φ (see section B in the Appendix for more details). Equation (4.11) then becomes

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \quad (4.12)$$

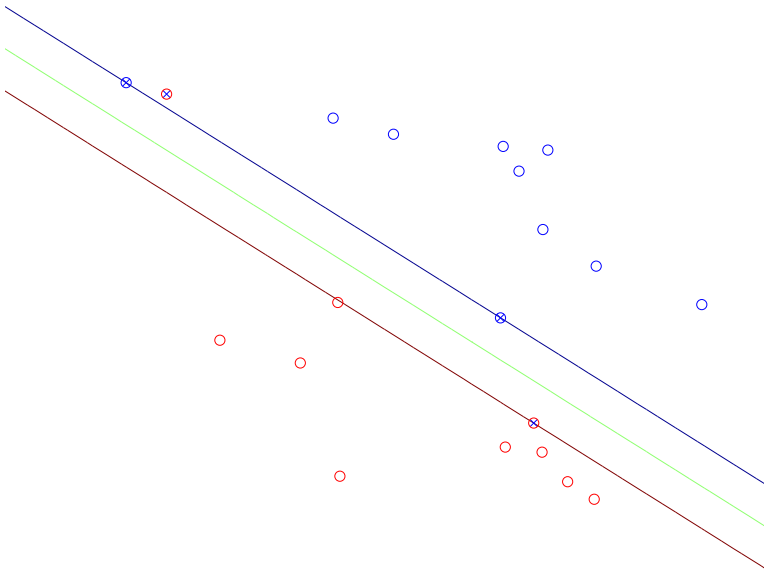

Figure 4.1: Optimal linear separating hyperplane (in green), it corresponds to the implicit curve defined by $f(\mathbf{x}) = 0$, while the blue and red line corresponds to the curves defined by $f(\mathbf{x}) = 1$ and $f(\mathbf{x}) = -1$. The support vectors are marked with an 'x'. The distance between the red and blue line is the margin. Notice how the misclassified red sample is a support vector.

An example of the optimal separating hyperplane for the same 2-dimensional problem of Figure 4.1 is shown in Figure 4.2.

After training, that is after the minimization of L_P, some of the α_is (actually most of them in many practical applications) are zero; those $\mathbf{x}_i$s for which this does *not* hold are somehow crucial to the solution and are called *support vectors*, hence the name of the approach. This phenomenon is known as *sparseness* of the solution, meaning that only a subset of the training data is usually really needed to build it. This is a quick account of SVMs — the interested reader is referred to (Burges, 1998) for a tutorial, and to (Cristianini and Shawe-Taylor, 2000) for a comprehensive introduction to the subject.

4.3 Previous work

An exact simplification of the decision function (4.12) is proposed in (Downs *et al.*, 2001), based upon linear independence of the SVs in the feature space, performed *after* the training is done. In particular they observed that if a support vector is

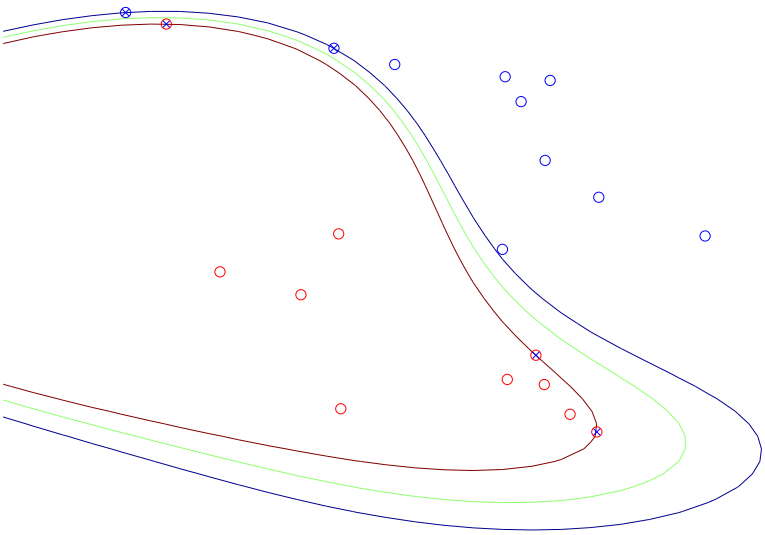

Figure 4.2: Optimal separating hyperplane using a Gaussian kernel (in green). The support vectors are marked with an 'x'. The use of a non-linear kernel makes possible to separate the two classes without misclassified samples.

dependent on the other support vectors in the feature space, *i.e.*

$$\exists \mathbf{x}_k : K(\mathbf{x}, \mathbf{x}_k) = \sum_{i=1, i \neq k}^{l} c_i K(\mathbf{x}, \mathbf{x}_i) \tag{4.13}$$

then the decision function (4.12) found after training can be written as

$$f(\mathbf{x}) = \sum_{i=1, i \neq k}^{l} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + \alpha_k y_k \sum_{i=1, i \neq k}^{l} c_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{4.14}$$

Hence it is possible to remove the dependent support vector $\mathbf{x}_k$, update the other coefficients, and obtain a new smaller representation of the decision function, without changing it in any way. Notice that the new coefficients may not respect the KKT constraints.

This can be seen as a simple consequence of the fact that, if the feature space has dimension n, at most $n + 1$ SVs are required to build the solution (Pontil and Verri, 1998). The idea is useful in reducing the testing time, but it is unfeasible in an online setting, since the simplification should be performed every time a new sample is acquired. The same consideration applies, *e.g.*, to the after-training simplification proposed in (Nguyen and Ho, 2005). On the other hand, discarding from the sample set the linearly dependent SVs will result in an approximation; other methods to heuristically select a subset of the support vectors have been proposed, *e.g.*, in (Lee and Mangasarian, 2001; Keerthi *et al.*, 2006; Wu *et al.*, 2006). Besides

this, these methods require the knowledge of the full training set, and therefore are not suited for online learning.

In order to keep the solution compact without losing accuracy, the key is to build a low-rank approssimation of the kernel matrix. Unsupervised rank reduction methods have been proposed, *e.g.* (Baudat and Anouar, 2003), as well as supervised ones, *e.g.* (Bach and Jordan, 2005), but no application of these ideas appears so far, to the best of our knowledge, in online settings.

A different method has been proposed by Collobert *et al.* (Collobert *et al.*, 2006): they have used a non-convex formulation of the learning problem where training errors are no longer support vectors thus dramatically reducing the growth rate of the support vectors with the training samples. Anyway in the paper it is not clear if the number of support vectors reaches a limit or if it will grow indefenitely, even if less than with standard SVM.

The exact solution to online SVM learning was given by Cauwenberghs and Poggio in 2000 (Cauwenberghs and Poggio, 2000), but their idea has received little attention in the community so far (Laskov *et al.*, 2006).

4.4 Sparseness of the solution

The time required by an SVM to train and predict is, in turn, cubic and linear in the number of support vectors (Keerthi *et al.*, 2006). Moreover, a recent result by Steinwart (Steinwart, 2003) indicates that the number of support vectors, r, increases linearly with the number l of training samples (given a kernel function K, r tends to $2B_K l$, where B_K is the smallest classification error achievable with the kernel K). Therefore, although support vectors somehow code all the information required by the solution, their number grows indefinitely as the input space is sampled. It is then highly desirable that the number of support vectors is kept as small as possible, without losing accuracy. Surprisingly, even if the machine keeps growing, usually the generalization power reaches a plateau after a while.

In general, the possibility to obtain an alternative, equivalent, and possibly more compact representation of the SVM solution follows from the fact that the solution of an SVM problem is not unique if the kernel matrix K, where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, does not have full rank, which is equivalent to some of the support vectors being linearly dependent on the others *in the feature space*. In fact, as pointed out in (Burges, 1998), given a vector $\boldsymbol{\alpha}$ solution of Equation (4.2) and (4.3), consider $\boldsymbol{\delta}$ that belongs to the null space of K, orthogonal to the vector all of whose components are 1 and satisfing $\sum_{i=1}^{l} \delta_i y_i = 0$. If $0 \leq \alpha_i + \delta_i \leq C$ then $\boldsymbol{\alpha} + \boldsymbol{\delta}$ is also a solution. However it is possible to show that the space of possible solutions to an SVM problem is even larger. In fact using the Representer Theorem (Kimeldorf and Wahba, 1970; Cox and O'Sullivan, 1990), Equation (4.5) can be written as follows:

$$\mathbf{w} = \sum_{i=1}^{l} \beta_i \mathbf{x}_i \tag{4.15}$$

for a set of generic coefficients β_i. Substituting Equation (4.15) in (4.4) and using the kernel trick, we get

$$L'_P = \sum_{i,j}^{l} \left(\frac{1}{2}\beta_i - \alpha_i y_i \right) \beta_j K_{ij} - \sum_{i=1}^{l} \alpha_i (b y_i - 1 + \xi_i) + \sum_{i=1}^{l} (C - \mu_i)\xi_i^p \tag{4.16}$$

Now, enforcing the KKT conditions on *this*, more general version of the problem, one obtains that

$$\frac{\partial L'_P}{\partial \beta_i} = \sum_{i=1}^{l} (\beta_i - \alpha_i y_i) K_{ij} = 0 \tag{4.17}$$

Clearly, in order for (4.17) to hold, the vector whose components are $\beta_i - \alpha_i y_i$ must be in the null space of K. Now if K has full rank, the null space only consists of the null vector, and therefore $\beta_i = \alpha_i y_i$ (this particular result already appears in (Keerthi *et al.*, 2006)). Otherwise, there are infinite solutions to the SVM problem, and the β_is are not constrained at all: this agrees with Downs *et al.*'s method and generalizes it.

4.5 Online Independent Support Vector Classification

To avoid simplifying the solution each time a new sample is acquired, we need a way to use independent SVs only. Hence, the main idea is to decouple the concept of "basis" vectors, that is the vectors $\mathbf{x}_i$ that we constrain to be allowed to have a β_i different from zero in (4.15), from the samples used to find out the actual values of these β_is. If the selected basis vectors span the same subspace as the whole sample set, the solution found will be equivalent — that is, we will not lose any precision.

Following Keerthi *et al.* (Keerthi *et al.*, 2006) then, and inspired by the above considerations, we explicitly choose a subset of the support vectors to form a basis for the solution. In that paper, two heuristics are proposed to select an appropriate subset of support vectors; we hereby propose *to online select the set of support vectors that are linearly independent in the feature space and to build the solution only using them*. The solution found this way is *the same* as if using all the training samples as basis set, that is the classical SVM formulation. No approximation whatsoever is involved, unless one gives it up in order to obtain even less support vectors. See below, especially Section 4.6, for a discussion on this point. Moreover *the training procedure is incremental*, after each new sample the coefficients are updated without recalculating the entire solution from scratch.

We assume that a set of l training samples is available and that the machine has been trained on them. The indexes of the vectors in the current basis are denoted

by $\mathcal{B}$, and $\mathbf{x}_{l+1}$ denotes the new sample under judgement. Since the procedure is incremental, we also assume that the vectors indexed by $\mathcal{B}$ are linearly independent in the feature space, that is, that $K_{\mathcal{BB}}$ has full rank. The algorithm can then be summed up as follows:

- check whether $\mathbf{x}_{l+1}$ is linearly independent from the basis in the feature space; if it is, add it to $\mathcal{B}$; otherwise, leave $\mathcal{B}$ unchanged.
- incrementally re-train the machine.

In the following, the notation A_{IJ} and $\mathbf{v}_I$, where A is a matrix, $\mathbf{v}$ is a vector and $I, J \subset \mathbb{N}$ denote in turn the sub-matrix and the sub-vector obtained from A and $\mathbf{v}$ by taking the indexes in I and J. The next two Subsections detail the linear independence test and the training method.

4.5.1 Linear independence

In general, checking linear independence in a matrix is done via some decomposition, or by looking at the eigenvalues of the matrix; but here we want to check whether a *single* vector is linearly independent from a set of vectors which are already known to be independent. Inspired by the definition of linear independence (Engel *et al.*, 2002), we check how well the vector can be approximated by a linear combination of the vectors in the set. Let $d_j \in \mathbb{R}$ with $j \in \mathcal{B}$; then let

$$\Delta = \min_{\mathbf{d}} \left|\left| \sum_{j \in \mathcal{B}} d_j \phi(\mathbf{x}_j) - \phi(\mathbf{x}_{l+1}) \right|\right|^2 \tag{4.18}$$

If $\Delta > 0$ then $\mathbf{x}_{l+1}$ is linearly independent with respect to the basis, and $l+1$ is added to $\mathcal{B}$. In practice, we check whether $\Delta \leq \eta$ where $\eta > 0$ is a tolerance factor, and we expect that larger values of η lead to worse accuracy, but also to smaller bases. As a matter of fact, if η is set at machine precision then OISVMs retain the exact accuracy of SVMs. Notice also that if the feature space has finite dimension n, then no more than n linearly independent vectors can be found, and $\mathcal{B}$ will never contain more than n vectors.

Expanding equation (4.18) we get

$$\Delta = \min_{\mathbf{d}} \left(\sum_{i,j \in \mathcal{B}} d_j d_i \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_i) - 2 \sum_{j \in \mathcal{B}} d_j \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_{l+1}) + \phi(\mathbf{x}_{l+1}) \cdot \phi(\mathbf{x}_{l+1}) \right) \tag{4.19}$$

that is, applying the kernel trick,

$$\Delta = \min_{\mathbf{d}} \left(\mathbf{d}^T K_{\mathcal{BB}} \mathbf{d} - 2\mathbf{d}^T \mathbf{k} + K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) \right) \tag{4.20}$$

where $k_i = K(\mathbf{x}_i, \mathbf{x}_{l+1})$ with $i \in \mathcal{B}$. It is apparent from Equation (4.20) that the range of η is related to the kernel used; for example for Gaussian kernels $\Delta \leq 1$ and hence good values of η range in $\{0, 1\}$.

Solving (4.20), that is, applying the extremum conditions with respect to $\mathbf{d}$, we obtain

$$\tilde{\mathbf{d}} = K_{\mathcal{B}\mathcal{B}}^{-1}\mathbf{k} \tag{4.21}$$

and, by replacing (4.21) in (4.20) once,

$$\Delta = K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) - \mathbf{k}^T\tilde{\mathbf{d}} \tag{4.22}$$

In general it is possible to prove that, given $\eta > 0$, the number of basis vectors will reach a finite number and then will stop growing: this is obvious for finite dimensional feature space but the same result holds also for infinite dimensional spaces (Engel *et al.*, 2004).

Note that $\mathcal{B}$ can be safely inverted since, by incremental construction, it is full-rank. An efficient way to do it, exploiting the incrementality of the approach, is that of updating it recursively:

$$K_{\mathcal{B}\mathcal{B}}^{-1} \leftarrow \begin{bmatrix} & & & 0 \\ & K_{\mathcal{B}\mathcal{B}}^{-1} & & \vdots \\ & & & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} + \frac{1}{\Delta}\begin{bmatrix} \tilde{\mathbf{d}} \\ -1 \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{d}}^T & -1 \end{bmatrix} \tag{4.23}$$

where $\tilde{\mathbf{d}}$ and Δ are already evaluated during the test. This method matches the one used in Cauwenberghs and Poggio's incremental algorithm (Cauwenberghs and Poggio, 2000), in turn similar to on-line recursive estimation of the covariance of sparsified Gaussian processes (Csató and Opper, 2001). Thanks to this incremental evaluation, the time complexity of the linear independence check is $O(|\mathcal{B}|^2)$, as one can easily see from Equation (4.21).

To gain other insights to what is going on using this active sparsification method consider the case in which a Gaussian kernel is used. Consider the expression (4.18) with $\mathcal{B} = \{i\}$, that is, as the i-th element is the only one in the base

$$\Delta_i = \min_{d_i} ||d_i\phi(\mathbf{x}_i) - \phi(\mathbf{x}_{l+1})||^2 \tag{4.24}$$

Obviously $\Delta_i \geq \Delta, \forall i \in \mathcal{B}$, so if $\Delta_i \leq \eta$ then we have that $\Delta \leq \eta$ and the sample $l + 1$ will not be added to the basis set. Remembering Equations (4.19)-(4.22), last equation can be expanded in

$$\Delta_i = K(\mathbf{x}_{l+1}, \mathbf{x}_{l+1}) - \frac{K(\mathbf{x}_{l+1}, \mathbf{x}_i)^2}{K(\mathbf{x}_i, \mathbf{x}_i)} \tag{4.25}$$

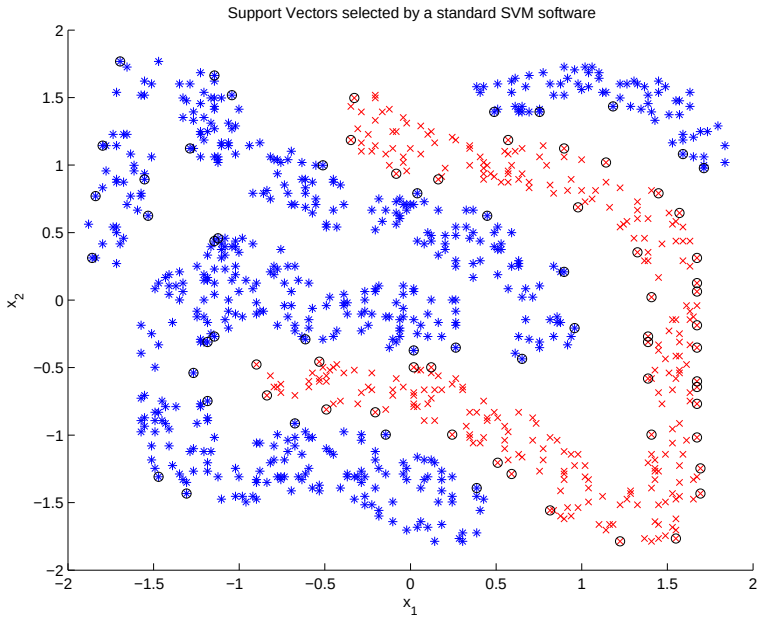

Figure 4.3: Classification problem *fourclass*. The support vectors selected by the a standard SVM with gaussian kernel are circled.

If we consider the case in which the kernel is Gaussian we have that $K(\mathbf{x}, \mathbf{x}) = 1, \forall \mathbf{x}$ and we can write

$$\begin{aligned} \Delta_i \leq \eta &\Leftrightarrow 1 - K(\mathbf{x}_{l+1}, \mathbf{x}_i)^2 \leq \eta \\ &\Leftrightarrow K(\mathbf{x}_{l+1}, \mathbf{x}_i) \geq \sqrt{1-\eta} \\ &\Leftrightarrow \exp\left(-\gamma \left|\left|\mathbf{x}_{l+1} - \mathbf{x}_i\right|\right|^2\right) \geq \sqrt{1-\eta} \\ &\Leftrightarrow \left|\left|\mathbf{x}_{l+1} - \mathbf{x}_i\right|\right|^2 \leq -\frac{1}{2\gamma}\log\left(1-\eta\right) \end{aligned} \tag{4.26}$$

Hence if at least one point $\mathbf{x}_i$ of the basis set is too near to the new point $\mathbf{x}_{l+1}$, it will be not added to the basis set. In other words when we use a Gaussian kernel, fixing a certain value of η implies imposing a minimum distance between the points selected as basis vectors. An example of this is shown in Figures 4.3 and 4.4. In this case γ is equal to 5 and η is 0.4, hence the minimum squared distance from Equation (4.26) is ≈ 0.0511, while the minumum distance for the support vectors selected is ≈ 0.0639.

4.5.2 Training the machine

The training method largely follows Keerthi *et al*. (Keerthi and DeCoste, 2005; Keerthi *et al.*, 2006), that we have adapted for online training. The algorithm

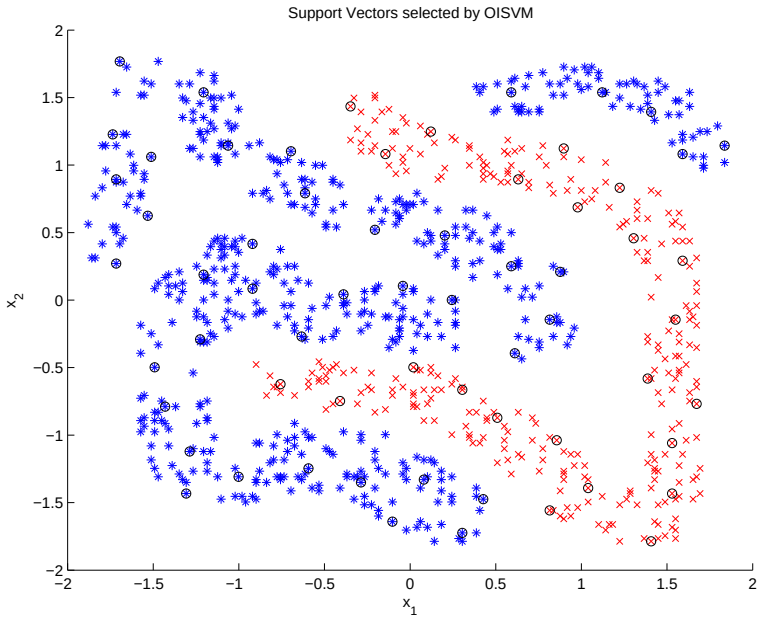

Figure 4.4: Same problem of the figure 4.3. The circled samples are the ones selected by our sparsification procedure as basis vectors. It is possible to see the effect of the minimal distance imposed by the constant η.

directly minimizes problem (4.2) as opposed to the standard way of minimizing its dual Lagrangian form, allowing to select explicitly the basis vectors to use. Let $\mathcal{D} \subset \{1, \ldots, l\}$, setting $p = 2$ in (4.2) we can write it as an unconstrained problem

$$\min_{\boldsymbol{\beta}} \left(\frac{1}{2} \boldsymbol{\beta}^T K_{\mathcal{D}\mathcal{D}} \boldsymbol{\beta} + \frac{1}{2} C \sum_{i=1}^{l} max\left(0, 1 - y_i K_{i,\mathcal{D}} \boldsymbol{\beta}\right)^2 \right) \tag{4.27}$$

where $\boldsymbol{\beta}$ is the vector of the Lagrangian coefficients involved in $f(\mathbf{x})$, analogously to the α_is in the original formulation. For convenience the bias term has not been included, but the analysis presented in this section can be simply extended to include it (see chapter A in the Appendix). Then, we explicitly set $\mathcal{D} = \mathcal{B}$, assuring thus that the solution to the problem is unique, since $K_{\mathcal{B}\mathcal{B}}$ is full rank by construction. Newton's method as modified by Keerthi *et al.* (Keerthi and DeCoste, 2005; Keerthi *et al.*, 2006) can then be used to solve (4.27) after each new sample. When the new sample $\mathbf{x}_{l+1}$ is received the method goes as follows:

1. use the current value of $\boldsymbol{\beta}$ as starting vector;

2. let $o_{l+1} = K_{l+1,\mathcal{B}} \boldsymbol{\beta}$, if $1 - y_{l+1} o_{l+1} \geq 0$ stop: the current solution is already optimal;

3. let $\mathcal{I} = \{i : 1 - y_i o_i > 0\}$ where $o_i = K_{i,\mathcal{B}}\boldsymbol{\beta}$ is the output of the i-th training sample;

4. update $\boldsymbol{\beta}$ with a Newton step: $\boldsymbol{\beta} - \gamma \mathbf{P}^{-1}\mathbf{g} \rightarrow \boldsymbol{\beta}$ where $\mathbf{P} = K_{\mathcal{B}\mathcal{B}} + CK_{\mathcal{B}\mathcal{I}}K_{\mathcal{B}\mathcal{I}}^T$ and $\mathbf{g} = K_{\mathcal{B}\mathcal{B}}\boldsymbol{\beta} - CK_{\mathcal{B}\mathcal{I}}(\mathbf{y}_\mathcal{I} - \mathbf{o}_\mathcal{I})$;

5. let $\mathcal{I}^{new} = \{i : 1 - y_i o_i > 0\}$ where o_i are ricalculated using new $\boldsymbol{\beta}$. If $\mathcal{I}^{new}$ is equal to $\mathcal{I}$ stop; otherwise $\mathcal{I} = \mathcal{I}^{new}$ and go to step 4.

In Step 4 above, γ is set to one, without any convergence problem. With this choice the update of $\boldsymbol{\beta}$ is $C\mathbf{P}^{-1}K_{\mathcal{B}\mathcal{I}}\mathbf{y}_\mathcal{I} \rightarrow \boldsymbol{\beta}^{new}$. In order to speed up the algorithm, we maintain an updated Cholesky decomposition of $\mathbf{P}$ and a vector with the product $K_{\mathcal{B}\mathcal{I}}\mathbf{y}_\mathcal{I}$: every time a sample enters or exits from the set $\mathcal{I}$ these two quantities are updated. It turns out that the algorithm converges in very few iterations, usually 0 to 2; the time complexity of the re-training step is $O(|\mathcal{B}|l)$, as well as its space complexity; hence, keeping $\mathcal{B}$ small will speed up the training time as well as the testing time.

4.6 Experimental Results

In order to test the effectiveness of OISVMs with respect to standard SVMs, we have chosen a set of databases commonly used in the machine learning community[1] and have then run comparative tests on them. In order to check our predictions about the linear independence tolerance constant, η, we have chosen finite- and infinite-dimensional kernels, namely polynomial kernels of degree 1 (linear) and cubic, and Gaussian kernel. We expect, in the finite-dimensional case, η to be essentially irrelevant, and the machine to stop growing once a certain number of l.i. support vectors have been found. This is exactly due to the feature space being finite-dimensional, and therefore only a finite number of l.i. vectors can be found. In the case of the infinite-dimensional kernel, we have run the OISVM with η at different values, expecting, as foretold, bigger values of η to cause the accuracy to degrade, but also the size of the machine to remain smaller than with smaller values.

OISVM is implemented in Matlab hence CPU times cannot be used.

For each benchmark, we display the mean number of retained support vectors on 10 random $75\%/25\%$ train/test runs. We compare against LIBSVM (Chang and Lin, 2001) (straight line), a standard SVM implementation. The coefficients γ and C have been found by cross-validation and employed in both LIBSVM and OISVMs. For the sake of comparison, LIBSVM has been also modified as suggested by its Authors in order to set $p = 2$ in equation (4.2), therefore in the following it is called LIBSVM-2. In the case of finite-dimensional kernels, we only show the performance of LIBSVM-2 against OISVMs with η at machine precision;

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`, last access 19/02/2007.

Benchmark name	Classification rate loss	% SVs vs. LIBSVM-2	% SVs vs. LIBSVM
Breast	0.47 ± 0.82	10.2 ± 0.87	22.1 ± 1.77
Diabetes	-0.52 ± 2.1	40.2 ± 2.1	55.2 ± 2.73
German	0.40 ± 1.15	6.1 ± 0.23	9.2 ± 0.35
Heart	-0.45 ± 1.01	10.3 ± 0.56	15.5 ± 0.94

Table 4.1: Comparison of OISVM and LIBSVM on standard benchmarks, solved using a Gaussian kernel. For each benchmark, we report the difference in classification rate with respect to LIBSVM-2 and the percentage of the number of SVs with respect to LIBSVM and LIBSVM-2. The values of η for each dataset have been chosen in order not to loose more than 0.5% accuracy.

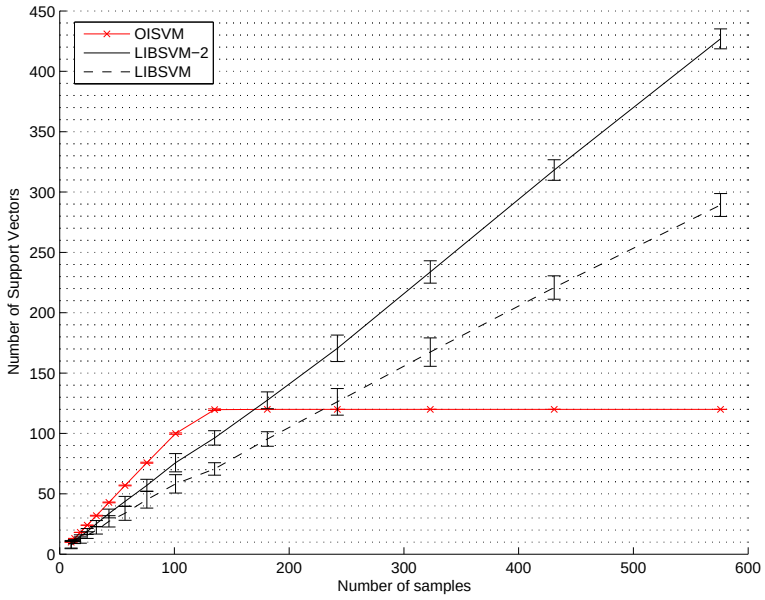

Figure 4.5: Comparison of OISVM and LIBSVM on the *Diabetes* benchmark, it is solved using a homogeneous polynomial kernel with degree 3.

in the case of the infinite-dimensional kernel, we show one curve for a value of η that guarantees a good trade-off between performance and sparseness.

Consider Figure 4.5: when all samples have been loaded, LIBSVM-2 has about 427 SVs, and LIBSVM about 290, confirming the fact that the norm-2 formulation is known to be less sparse of the norm-1. The kernel used is a homogeneous polynomial with degree 3 and the benchmark has 8 features, therefore the dimension of the feature space is $\binom{10}{3} = 120$ (see, *e.g.*, (Burges, 1998)); and, as expected, OISVM stops acquiring new SVs when there are exactly 120, although it loads a

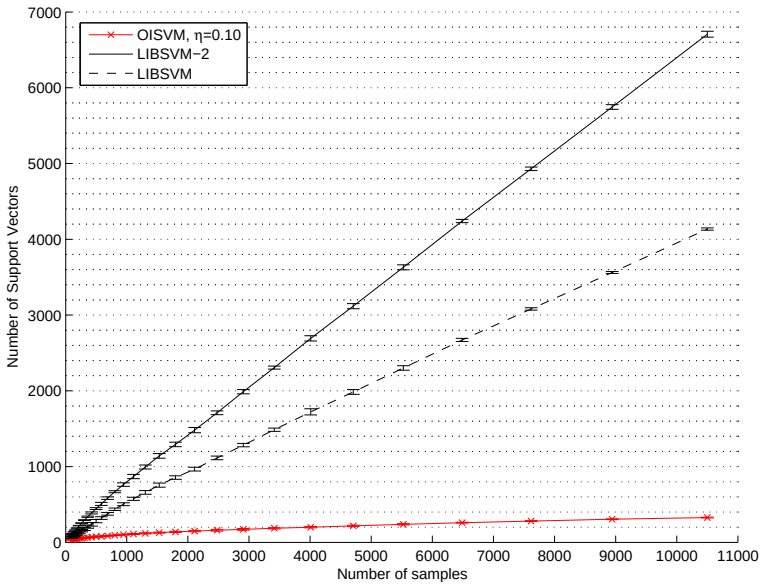

Figure 4.6: Comparison of OISVM and LIBSVM on the *Adult7* benchmark, it is solved using a Gaussian kernel.

few more before reaching the limit, with respect to the other approaches. The accuracy (not displayed) is exactly the same of LIBSVM-2, because the two solutions found are completely equivalent. Again, notice that, after having acquired 120 SVs, OISVM will never acquire any more ever, while keeping the same accuracy, whereas the LIBSVMs do, as theoretically proved in (Steinwart, 2003).

Consider now Figure 4.6: the kernel used is Gaussian and the dimension of its feature space is infinite. The benchmark is relevantly large (16100 samples) and complex (123 features). Nevertheless, with η as small as 0.1, at the end OISVM has less than 5% of the SVs used by LIBSVM-2 and less than 8% with respect to LIBSVM. The accuracy is $0.063\% \pm 0.14$ worse than that of LIBSVM-2.

Lastly, consider Table 4.1, which shows the very same data in compact form for 4 more standard databases. OISVM attains a number of SVs which is about 6% to slightly more than 55% of LIBSVM, whereas the accuracy is basically the same, being slightly better than LIBSVM in two cases (*Diabetes*, this time solved via a Gaussian kernel, and *Heart*).

As a final remark, notice that in general the number of support vectors chosen by OISVMs could be higher than that obtained by SVMs. An example of this phenomenon is visible in Figure 4.5, between x-values 0 and 150.

4.7 Discussion

A new method is presented to keep Support Vector Machines small, called OISVMs (Online Independent Support Vector Machines). OISVMs avoid inserting into their kernel matrix support vectors which are linearly dependent of previous ones in the feature space — in other words, the kernel matrix is always kept at full rank. The primal SVM problem is then solved via an incremental algorithm which benefits of the small size of the kernel matrix.

Experimental results show that (i) in the case of finite-dimensional kernels, OISVMs attain the theoretical limit of linearly independent support vectors allowed by the feature space; (ii) in the case of infinite-dimensional kernels, they dramatically reduce the number support vectors at the price of a negligible degradation in the accuracy. Notice that, in this latter case also, they can be used to obtain full precision, choosing the tolerance threshold to be equal to machine precision.

Chapter 5

Object Recognition and Categorization

Contents

5.1 Two view-based models for object recognition 70
5.2 Results on a categorization task 72
5.3 Adapting the features through selection 73
5.4 Discussion 78

EARLY approaches to object recognition in static images were influenced predominantly by the idea of the create a faithful description of the world, reconstructing the 3-D structure of objects as proposed by Marr (Marr, 1982). The difficulties to detect simple characteristics in images like edges and vertices have challenged these early models, favoring the recent idea of recognition systems that make use of viewpoint-dependent descriptions. Moreover there is psychophysical evidence supporting these approaches (Tarr and Bülthoff, 1998). In a view-based approach, each object is represented by a number of images taken from different viewpoints, then these model images are compared to the test images. However objects can appear in images in different positions, orientations and scales. Hence to reliably recognize objects, we should extract from the images features that are independent from the translation, rotation and scale transformations. Such an object recognition system should be used after a visual attention system, that would select a region of the image at once (Walther and Koch, 2006). Similar considerations can be done for a categorization system.

The reminder of the chapter is organized as follows: section 5.1 contains a description of the two state-of-the-art models for object recognition: the "standard model" and the SIFT model. Section 5.2 describes the experimental results of a comparison between the two methods. In section 5.3 we show how to improve the performance of the standard model and finally in sections 5.4 we draw some

conclusions.

5.1 Two view-based models for object recognition

In the following two view-based state of the art models for object recognition and categorization are compared: the so called "standard model" and the SIFT.

5.1.1 Standard Model

The so called "standard model" of object recognition has been proposed by (Riesenhuber and Poggio, 1999) and then improved by (Serre *et al.*, 2005). It can be thought as the natural evolution of previous hierarchical model for object recognition (Fukushima, 1980; Lecun *et al.*, 1998). In the model there are two types of layers, *cells*, that are alternated in the hierarchy. The "simple cells" extract local features from the previous level and are tuned to specific stimuli; the "complex cells" pool a number of specific simple cells, to have a local form of invariance, while simultaneously maintaining specificity to the stimuli. In particular the complex cells use a MAX operation between the inputs, that seems to have a biological justification (Lampl *et al.*, 2004). In the model two couple of layers of simple/complex cells are implemented, for a total of 4 layers. The first layer of simple cells (S1) extracts local orientations with a set of Gabor filters with different scales and orientations tuning. (Gabor filters, which are the product of a cosine grating and a 2D Gaussian envelope, has been used to approximate the receptive field sensitivity profile of orientation-selective neurons in primary visual cortex (Leventhal, 1991).) The layer of complex cells (C1) pools over a local neighborhood of space and scale the outputs of the simple cells of the previous layer. The filters used in layer S2, instead, are learnt directly from the images: a big number of random patches of the output of C1, of different sizes, are taken while the system "sees" different natural images. Each of those patches is set as a prototype of the S2 unit which are radial basis function (RBF) units. That is the output of i-th unit S2

$$\exp\left(-\frac{||X - C_i||^2}{2\sigma^2}\right) \tag{5.1}$$

at all the spatial positions X. The last layer of complex cells, C2, pools the maxima over all the scales and locations. Hence the output of the system is a feature vector of size equal to the number of S2 detectors, which is independent of the size of the input image. Hence the output of the system, in the limit of the digital implementation, is not carrying any information about the scale or the location of a certain local image feature. The system is purely feed-forward, without any feedback connections (Behnke, 2003), and consequently it is only appropriate to model fast decisions of object presence or absence (Hung *et al.*, 2005). We have used the Matlab source code of the model that is available of the website of the authors[1], the

[1] `http://cbcl.mit.edu/software-datasets`, last access 19/02/2007.

only modification that we have made is to set σ in Equation (5.1) dependent on the size of the patches, as proposed in (Mutch and Lowe, 2006), because it improves the performances at no additional computational cost. At the end of the model there is an SVM trained to classify the extracted features in object classes.

Beside the SVM classifier, the other non-fixed part of the system are the filters of the S2 layer, that can be thought as adapting to the images statistics. In fact uniformly sampling a stochastic variable, we obtain another random variable that has more or less the same probability density function of the original one. In section 5.3 we will discuss with greater details about the disadvantages of this method. Note that it is possible to learn class specific detectors, that are expected to have better performances on a single class, or universal detectors, that are expected to work equally well on all the possible image classification tasks. In the latter case a set of generic natural images is used to learn the filters. In the following tests we have used the set of universal features available on the website of the Authors.

5.1.2 SIFT

The Space Invariant Feature Transform (SIFT) (Lowe, 1999) are descriptors of stable image patches (keypoints) designed to be invariant to local image transformations as rotations, scale warpings, illumination changes and noise. Here an object, like in the standard model, is coded as a combination of SIFT points. The SIFTs have been shown to excel in the re-detection of a previously seen object under new image transformations.

Usually single SIFTs are matched one to the other and the class of the observed object is decided with the majority of the votes. Instead in our comparison we have decided to use an SVM classifier, like in the standard model. In this way, it is possible to have a fair comparison with the standard model and, at the same time, to enhance the generalization power of the SIFT. The input to the SVM are sets of SIFT point, each being a vector in $\mathbb{R}^{128}$ (the standard SIFT descriptor). Given that for each image a different number of SIFT can be found, each set associated with each image will have a variable number of points. Hence a special kernel must be used to calculate the scalar product between sets, and we have chosen the Matching Kernel proposed by Wallraven *et al*. (Wallraven *et al.*, 2003) (see also section B.2 in the Appendix). This kernel has been designed to match set of features of variable dimensions, in particular it has been used to match SIFT features, and it can also take into account the spatial information of each points. We have chosen not to use this possibility in order to give to the classifier the same type of information produced by the standard model. Note that no adaptation whatsoever is the model; even if prior information about the images has been implicitly used in the design of the optimal way to detect the keypoints and to code them.

The original software made by Lowe has been used[2].

[2] `http://www.cs.ubc.ca/~lowe/keypoints/`, last access 19/02/2007.

Figure 5.1: Sample images from the Caltech-101 database. In the first row they are taken, respectively from the class *Airplanes*, *Car side*, *Faces*, *Leaves* and *Motorbikes*; this subset has been used to compare the standard model and the SIFT. In the second row there are example images from the other categories (*Elephant*, *Gramophone*, *Umbrella*, *Yin Yang* and *Ibis*). Note that all the images are of different sizes, but in the test they have been normalized to have all the same height of 140 pixels, and the width has been rescaled proportionally.

5.2 Results on a categorization task

The Caltech datasets, containing 101 objects plus a background category (used as the negative set) and available at `http://www.vision.caltech.edu`[3], has been used for our tests. These datasets contain the target object embedded in a large amount of clutter and the challenge is to learn from unsegmented images and discover the target object class automatically. We have tested both approaches on a subset of the 101-object datasets plus an additional leaf database as in (Serre *et al.*, 2005) for a total of five datasets. Example images of these datasets are shown in Figure 5.1.

The system was trained with 15 examples from the each object class. From the remaining images, we extracted 50 images for each category to test the system's performance, averaging over 5 random splits. All images were normalized to 140 pixels in height (width was rescaled accordingly so that the image aspect ratio was preserved) and converted to gray values before processing like in (Serre *et al.*, 2005); this was done also for the images used with the SIFT, to have a fair comparison.

The parameters of the Matching Kernel and the C, for the classification with SIFT have been found with 5 random splits of 15/50 images for training/testing, for each category. Instead for the standard model a linear SVM has been used, with a value of C equal to 1, as in (Serre *et al.*, 2005). In fact the dimensionality of the input space is big enough compared to the number of training samples to have a separable problem, so the value of C is not critical. The paradigm of "one-vs-all" is

[3]Last access 19/02/2007.

Database name	Classification rate SIFT	Classification rate standard model
Airplane	92.00 ± 3.16	87.2 ± 5.40
Car side	88.80 ± 4.15	94.80 ± 3.35
Faces	88.40 ± 3.29	94.80 ± 3.03
Leaves	90.80 ± 3.35	88.80 ± 6.72
Motorbikes	88.40 ± 7.27	92.00 ± 5.48
Overall	89.68 ± 2.66	91.52 ± 1.61

Table 5.1: Comparison of SIFT and standard model classification performances on a subset on the 101-Caltech database. The mean classification rates $\pm$ standard deviation are shown for each datasets, on 5 random splits $15/50$ of training/test images.

used for the multi-class classification. The performances using 5 different random splits is summarized in table 5.2.

It is interesting to see that there is not a clear winner between the two methods. In fact the standard deviation are too high to say that there is a real difference between the performances. It is also interesting to note that the two methods appear complementary in their performances: it seems that easy datasets for one method are difficult for the other and vice versa. These results are in opposition with the result of (Serre *et al.*, 2005), that claim that the C2 features are better than SIFT features in the same classification task. In our opinion the main difference is that they do not use the right classifier for the SITFs. Indeed it is possible to obtain similar results using an appropriate classifier as the SVM plus the Matching Kernel. These findings make us believe that the classifier is the most critical part of an object recognition system, given two equally good feature extraction systems. As a further example of this claim, (Mutch and Lowe, 2006) have demonstrated that it is possible to gain more than 3% of classification performance with the supervised feature selection method in (Mladenić *et al.*, 2004).

5.3 Adapting the features through selection

In the spirit of adapting the feature extraction system to the image statistics, we want to address the possibility to select only a subset of the features extracted by the system. As in the learning of association fields (see Chapter 2) we are interested in using an unsupervised strategy, at least to select a subset of the available information, that could then refined using external (supervised) knowledge in later stages.

Given than the test time required to test a new point is proportional to the number of features that must be calculated on each image, reducing the number of features will proportionally reduce the testing time. On the other hand it is possible

that removing some of the features we improve also the classification performance. In fact in the machine learning literature it is well known that a classifier with a small number of informative features can work better than a classifier with tons of redundant or useless features. In general given a certain classification task it is possible to select some of the features that have more information and discard redundant or even useless ones (Mladenić *et al.*, 2004). Considering the case of online learning, it is not easy to select the features knowing the labels, that is in a supervised way, because the samples are available only one at time. An optimal feature selection could be done only after having acquired enough samples, but in this way the online behaviour would be disrupted.

In the following we introduce a simple algorithm to select a subset of feature in unsupervised way. The results will show that the subsets selected will be always better than a random selection.

5.3.1 Unsupervised feature selection for SVM

Consider the case of a linear SVM or in general any learning algorithm that depends on the scalar products in the input space. Let the case in which two features are identical, in this case we could remove one of the two and multiply the other by 2:

$$\exists p, q : a_p = a_q \forall \mathbf{a} \in \mathbb{R}^n \Rightarrow \tag{5.2}$$

$$\Rightarrow \mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = \sum_{i \neq p,q} a_i b_i + 2 a_p b_p = \tag{5.3}$$

$$= \sum_{i \neq p,q} a_i b_i + \sqrt{2} a_p \sqrt{2} b_p \tag{5.4}$$

Hence it is the same as being in a space with $n - 1$ dimensions, with one of them multiplied by a factor of $\sqrt{2}$. In general to have good results in machine learning it is important to give to all the input features the same importance, regardless of the input range. This is the reason because the range of the inputs are usually normalized to have the same maximum-minimum ranges or to have the same standard deviations. But in the case of two features identical this is impossible, because the extra weight is not in the feature itself but in the duplication. The case of a repeated feature can appear trivial, but it can be the case that two features carry the same information even if they are not exactly the same. Consider the case of two features, corresponding to the index p and q, that has a correlation index approximately equal to 1. In this case we can write

$$r(a_p, a_q) \approx 1 \forall a \in \mathbb{R}^n \Rightarrow a_p \approx \alpha a_q + \beta \tag{5.5}$$

If we remove the mean from all the features and normalized them to have unitary standard deviation then β can be considered 0 and $\alpha \approx 1$, and we can write again

$$\sum_{i=1}^{n} a_i b_i \approx \sum_{i \neq p,q} a_i b_i + \sqrt{2} a_p \sqrt{2} b_p \tag{5.6}$$

The same happens if the correlation is approximatively -1. Given the above considerations we claim that the presence of the correlated features can worse the classification performance, and removing some of them can improve the classifier and, at the same time, speed up the system because less features need to be computed. This is, again, similar to the idea proposed by Barlow that the neural responses are statistically independent, because the redundancy in the sensory input is removed by early sensory systems (Barlow, 1961).

Notice that this somewhat different from applying Principal Component Analysis (PCA) because it is true the with PCA redundant features are removed, but no gain in speed is obtained because the principal components are linear combination of *all* the original features, so all the features must be calculated in any case.

Hence we propose to build the reduced set of uncorrelated features starting from the sample correlation matrix between all the input features. We consider only the absolute values of the sample correlation matrix, because positive and negative correlations count in the same way. Then iteratively select the most correlated couple of features and discard one of them. In particular we have chosen to discard between of the two, the feature that has the smallest sum of the sample correlations with the other features. The rationale behind this approach is to keep the features that are mostly uncorrelated with the other. The row and column associated with the removed feature are removed from the sample correlation matrix and the removed feature is tagged as number n. Then the second removed feature is tagged as number $n - 1$ and so on, until all the rows and columns of the matrix are removed. At the end, the tag associated with each feature will give a sort of ranking of the uniqueness of the information carried by that particular features. Of course if there is a feature containing pure noise, uncorrelated to the other features it will be ranked as the most important. However this is a limit intrinsic to any unsupervised feature selection: without knowing the labels it is impossible to understand if the noise is useless to the classification task. It could be the case that the output is only function of the noise, *e.g.* the label of the sample being equal to the sign of the noise.

Then we select increasingly sets of features, considering the ranking obtained from the above method. For example the first set has the features from number 1 to number 100, the second from 1 to 200 and so on until the last set that contains all the features. The performance of the method is confronted with a baseline obtained using a random ranking of the features, and building the same sets of increasing sizes.

Note that even if a big number of input samples is needed to estimate the sample correlation matrix, their labels are not needed. Hence the selection takes place before the learning phase, thus it can be used in an online learning framework.

The feature selection method proposed works very well with the random features collected during the training phase by the standard model. It is worth to note that sampling randomly an input space we select more points in the areas more densely populated. Samples that are near in the input space it very likely that will be highly correlated. Hence the feature set randomly selected will contain many

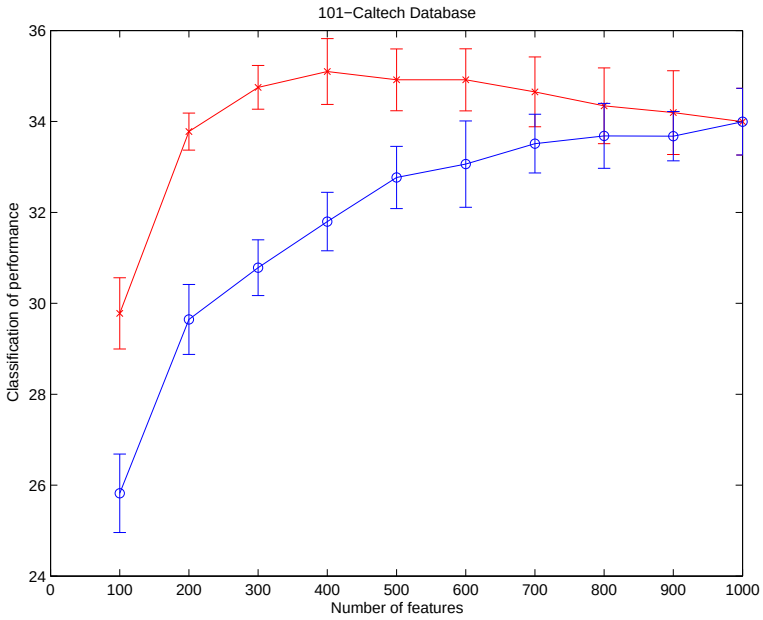

Figure 5.2: Classification performances using the unsupervised feature selection method proposed, compared with a random selection of subsets of features, on 5 different 15/50 training/test splits.

similar features belonging to the most common image patches. On the other hand it also is very likely that these very common features will have low discriminative power in a classification task, because they are present in all the classes. Note that the use of random features has been proposed even by others, *e.g.* (Nowak *et al.*, 2006), and it is very likely that this method could be applied successfully to other object recognition system.

The validity of the proposed method is demonstrated by the tests done on the 101-Caltech database using the standard model and a linear SVM.

5.3.2 Results

As in Section 5.2 the system was trained with 15 examples from the each object class, this time using all the 102 classes. From the remaining images, we extracted 50 images for each category to test the system's performance, averaging over 5 random splits. The images were preprocessed as described in 5.2. The kernel used is linear and the parameter C is equal to 1.

In Figure 5.2 there is the comparison between the random selection of features and the proposed method. As said above, using less features that the total it is possible to increase the accuracy of the classifier, at the same time reducing the computational cost. In this case it is possible to gain approximatively 1% of

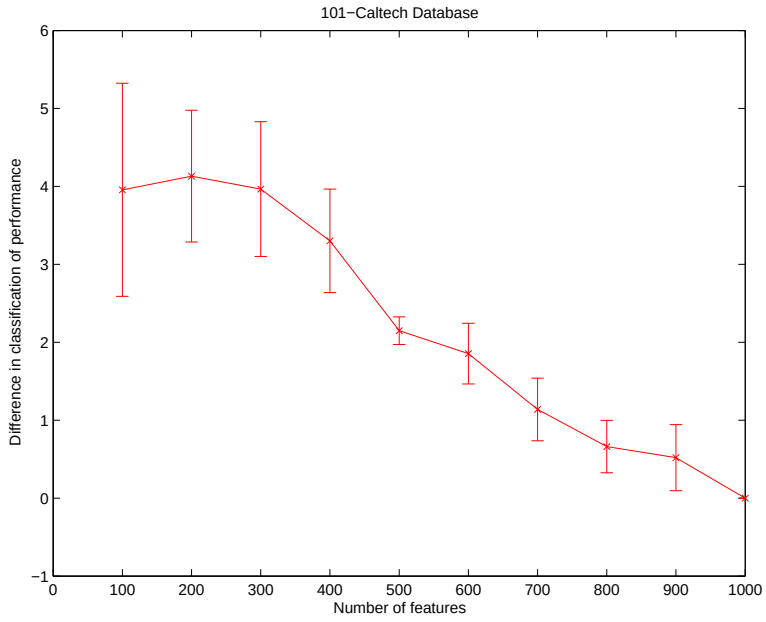

Figure 5.3: Difference of classification performances between the random method and the proposed method.

classification performances, having 2.5 times less features and, hence, 2.5 times faster.

Moreover selecting a subset of features with the proposed method is *always* better than selecting a random subset. This can be seen clearly in Figure 5.3 where it is plotted the difference in the performances between the two methods. Of course when all the features are selected the performances of the two methods are exactly the same, being equal to the performances of the system without any feature selection.

Analyzing Figure 5.2, we can see that the rate of improvement adding more random features is very slow. This suggest that adding more features is unlikely to be the right way to improve the performances. This can be explained from the fact that new features are likely to bring redundant information, correlated to already added features. At the same time unique and independent features, being more rare, will have a small weight (see Equation 5.6).

We have also combined OISVM (see Chapter 4) with the random feature selection and the unsupervised feature selection, and the results are shown in Table 5.2. The kernel used in the classification task is linear, hence the dimensionality of the feature space is equal to the dimensionality of the input space, that is to the number of features used. We can see that using OISVM the number of support vectors is more or less equal to the dimensionality of the space (the difference is due to numerical approximations), while using standard SVM the number of SVs

	100	300	600	1000
Feature sel.	1518 ± 0.8	1518 ± 2.8	1518 ± 1.8	1520 ± 2.1
Random features	1519 ± 1.7	1521 ± 2.6	1521 ± 3.1	1520 ± 2.1
OISVM $\eta = 0$, f. sel.	100 ± 0	302 ± 3.0	602 ± 3.3	1008 ± 2.1
OISVM $\eta = 0$, random	101 ± 2.2	310 ± 3.8	611 ± 5.5	1006 ± 1.5

Table 5.2: Mean number of support vectors for different numbers of features. The number of SVs for the unsupervised feature selection and random selection are more or less the same, while for OISVM it is exactly equal to the number of features because a linear kernel is used.

is independent from the number of features. Moreover η is set to 0, so the solution obtained is exactly the same of the one obtained with SVM. Considering the the time to evaluate the linear kernel function is proportional to the number of features, we obtain a speed-up of 6.25 times combining OISVM and the feature selection method.

5.4 Discussion

A comparison of two state of the art algorithms for object categorization has been made, stressing the importance of the learning part. The performance between the two methods are not statistically significant, given that the classifier has been correctly tuned. Hence it seems that learning is critical subsystem, given two equally good feature extraction systems.

Moreover, talking about the possibility to make the feature extraction system adapt to the statistics of input image, an unsupervised feature selection system has been introduced. The method is able to improve the performance of the classifier and at the same time to speed up the system. Object of the future work will be the integration of the model of visual attention system proposed in Chapter 3 with one of these models for object recognition.

Chapter 6

Conclusions

In this thesis we have presented some studies on the topics of learning and adaptation in computer vision.

Starting from raw images, we have shown that it is possible to adapt to the statistics of the world to have a better internal coding and to complete missing information in the input. Going up in the visual hierarchy, we have taken under consideration the mechanism of visual attention. Remembering the link between perception and action, and the simple fact that every biological system has an aim, we have proposed a proto-object based model of visual attention. Hence the model does not work on disembodied locations or meaningless pixels but with perceptual groupings that are the building blocks of the concept of visual object. Moreover the idea of proto-object has then been exploited to build an object recognition system, coupled with the attentive system.

Considering learning, we have introduced a new general online algorithm based on SVM, able to produce very sparse solutions retaining almost all the accuracy of the original SVMs. This different formulation of the SVM guarantees a finite number of support vectors, regardless of the number of training samples.

We have than applied this algorithm to an object classification task. We have also shown the advantages of adapting the feature extraction stage to the input statistics, gaining speed and classification accuracy at the same time.

The obtained results support our idea that learning and adaptation are critical for the comprehension of biological intelligence, and, hence, for creating an artificial cognitive agent.

Appendix A

How to include the bias term in OISVM

The formulation is exactly with the following substitutions:

$$\begin{bmatrix} K_{\mathcal{BB}} & 0 \\ 0 & 0 \end{bmatrix} \rightarrow K'_{\mathcal{BB}} \tag{A.1}$$

$$\begin{bmatrix} K_{\mathcal{BI}} \\ \mathbf{1} \end{bmatrix} \rightarrow K'_{\mathcal{BI}} \tag{A.2}$$

where $\mathbf{1}$ is a row vector of all 1, with $|\mathcal{I}|$ elements.

$$\begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} \rightarrow \boldsymbol{\beta}' \tag{A.3}$$

With these substitutions the regularization term $\frac{1}{2}\boldsymbol{\beta}'^T K'_{\mathcal{DD}}\boldsymbol{\beta}'$ is equal to $\frac{1}{2}\boldsymbol{\beta}^T K_{\mathcal{DD}}\boldsymbol{\beta}$, while the output of the i-th training sample $o'_i = K'_{i,\mathcal{B}}\boldsymbol{\beta}'$ is equal to $K_{i,\mathcal{B}}\boldsymbol{\beta} + b$.

Notice that $K_{\mathcal{BB}}$ in Equation (4.21) is always the same, and it is not changed by the use of a bias term.

Appendix B

Kernels for SVM

To understand what is a kernel can be useful to directly build it passing through an explicit formulation of the function Φ. Consider $\mathbf{x} \in \mathbb{R}^2$ and $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$\Phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \tag{B.1}$$

We have that

$$\begin{aligned} \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}) &= x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 = \\ &= (\mathbf{x} \cdot \mathbf{y})^2 \end{aligned} \tag{B.2}$$

Hence it is possible to calculate the scalar product in the new space without knowing explicitly the function Φ. In some other cases it is not possible to work with Φ because the space induced by the kernel is infinite dimensional as for the Gaussian kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \left|\left|\mathbf{x} - \mathbf{y}\right|\right|^2\right) \tag{B.3}$$

B.1 Some notes on polynomial kernels

One of the most used kernel is the polynomial one

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^p \tag{B.4}$$

In many text and publications often it is written without the constant term c (even in LIBSVM the default value for c is 0). It is obvious that the homogeneous polynomial kernel function, that is with $c = 0$, will be even or odd, depending on the degree p. Maybe it is not so obvious that the entire decision or regression function found by an SVM very likely will also be odd or even! In fact the SVM solution, both in regression and classification, can be written as

$$f(x) = \sum \beta_i K(\mathbf{x}, \mathbf{x}_j) + b = \sum \beta_i \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)^p + b \tag{B.5}$$

If the bias term b is 0 then $f(x)$ is even or odd, depending on p, being linear combination of odd or even functions.

B.2 The local matching kernel

Denoting with $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^n$ and $\mathcal{L} = \{\mathbf{L}_i\}_{i=1}^m$ two sets of local features associated with two images, we define

$$K(\mathcal{I}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^{M} \widehat{K}\left(\mathbf{I}_{p_i}, \mathbf{L}_{q_i}\right) \tag{B.6}$$

$$M = \min(N, \max(m, n)) \tag{B.7}$$

where

$$\widehat{K}\left(\mathbf{I}_{p_l}, \mathbf{L}_{q_k}\right) = \exp\left(-\gamma \left|\left|\mathbf{I}_{p_l} - \mathbf{L}_{q_k}\right|\right|^2\right) \tag{B.8}$$

$$\widehat{K}\left(\mathbf{I}_{p_1}, \mathbf{L}_{q_1}\right) \geq \widehat{K}\left(\mathbf{I}_i, \mathbf{L}_j\right) \tag{B.9}$$

$$\widehat{K}\left(\mathbf{I}_{p_2}, \mathbf{L}_{q_2}\right) \geq \widehat{K}\left(\mathbf{I}_i, \mathbf{L}_j\right)\ i \notin \{p_1\}\ j \notin \{q_1\} \tag{B.10}$$

$$\widehat{K}\left(\mathbf{I}_{p_3}, \mathbf{L}_{q_3}\right) \geq \widehat{K}\left(\mathbf{I}_i, \mathbf{L}_j\right)\ i \notin \{p_1, p_2\}\ j \notin \{q_1, q_2\} \tag{B.11}$$

$$\cdots \tag{B.12}$$

$$\widehat{K}\left(\mathbf{I}_{p_M}, \mathbf{L}_{q_M}\right) \geq \widehat{K}\left(\mathbf{I}_i, \mathbf{L}_j\right)\ i \notin \{p_1, \cdots, p_{M-1}\}\ j \notin \{q_1, \cdots, q_{M-1}\} \tag{B.13}$$

$$p_i \neq p_j, q_i \neq q_j\ \ \forall i, j = 1, \cdots, M \tag{B.14}$$

and N and γ are parameters of the kernel. This definition is slightly different from the definition given in (Wallraven *et al.*, 2003), but it has better performances[1]. The idea behind this formulation is to consider the M best matching couples of local features, without considering the ones that we have already matched, starting from the best couple. Even if this kernel is non-Mercer (Boughorbel *et al.*, 2004), it has been largely used with good performances in various object recognition and classification tasks.

[1] Personal communication by B. Caputo.

List of Figures

1.1 The relation between action, attention and experience. 16

2.1 Sample input image from the Berkeley Segmentation Database. All the images were converted to grayscale before using the proposed method. 21
2.2 Complex cells output to the image in figure 2.1 for 0 degrees filter of formula (2.1). 23
2.3 Main directions for the association field for the orientation of 0 degrees in the central pixel. 25
2.4 Main directions for the association field for the orientation of 67.5 degrees in the central pixel. 25
2.5 Main directions for the association field for the orientation of 0 degrees in the central pixel, with the modified approach. 27
2.6 Difference between the two eigenvalues of the association field of figure 2.5. 27
2.7 Main directions for the association field for orientation of 67.5 degrees, with the modified approach. 28
2.8 Difference between the two eigenvalues of the association field of figure 2.7. 28
2.9 Comparison of the decay for the various orientations. On the y axis there are the first eigenvalues normalized to a maximum of 1, on the x axis is the distance from the reference point along the main field direction. 29
2.10 Comparison between tensor voting with learned fields (PG label) and the complex cell layer alone (OE label). 30
2.11 Test image contours using the complex cell layer alone. 31
2.12 Test image contours using tensor voting with the learned fields. Notice the differences with the image 2.11: the contours are linker together and the gaps are reduced. Especially on the contour of back of the tiger the differences are evident. 31

3.1 A simple schematization of the FIT model. 35
3.2 The robotic setup, Babybot. The experimental setup consists of a five degrees of freedom robot head, and an off-the-shelf six degrees of freedom robot manipulator, both mounted on a rotating base: *i.e.* the torso. The kinematics resembles that of the upper part of the human body although with less degrees of freedom. 38
3.3 Block diagram of the model. The input image is first separated in the three color opponency maps, than edges are extracted. A watershed transform creates the clusters of uniform or uniform gradient of color (blobs). The saliency is defined on the blobs, and not on single pixels, taking into account top-down biases. 39
3.4 Log-polar transform of an image. 40
3.5 Filtering the image on the left with a Difference of Gaussians with the size of positive lobe equal to the size of the circle in the middle, we obtain the image on the right. Smaller blobs will be depressed while larger ones will be depressed in their centers. 42
3.6 Example of model maps. 44
3.7 Some example images during exploration phase (1-3) and related segmentations (4-6) used to build the statistical model of the object. Note how the parts not of the object are not always detected, so their estimated probability to belong to the object will be low. . . . 47
3.8 The flow chart of the visual search of an object (the toy airplane), recognition and segmentation. The saliency map is generated using the information about the color blue of the toy. 47
3.9 Example saliency maps. In (4) there is the bottom-up saliency map of the image (1). In (5) the top-down saliency map of (2), while searching for the blue toy airplane. Image (6) is the figure-ground segmentation of the image in (3), after having recognized the object. 49
3.10 Result on a static example image taken from the database by Itti and Koch. Image (1) is the log-polar input image; image (2) is the binary mask used for to verify the correct localization of the target object and image (3) is the saliency map generated by the system. 49

4.1 Optimal linear separating hyperplane (in green), it corresponds to the implicit curve defined by $f(\mathbf{x}) = 0$, while the blue and red line corresponds to the curves defined by $f(\mathbf{x}) = 1$ and $f(\mathbf{x}) = -1$. The support vectors are marked with an 'x'. The distance between the red and blue line is the margin. Notice how the misclassified red sample is a support vector. 56
4.2 Optimal separating hyperplane using a Gaussian kernel (in green). The support vectors are marked with an 'x'. The use of a non-linear kernel makes possible to separate the two classes without misclassified samples. 57

4.3 Classification problem *fourclass*. The support vectors selected by the a standard SVM with gaussian kernel are circled. 62
4.4 Same problem of the figure 4.3. The circled samples are the ones selected by our sparsification procedure as basis vectors. It is possible to see the effect of the minimal distance imposed by the constant η. 63
4.5 Comparison of OISVM and LIBSVM on the *Diabetes* benchmark, it is solved using a homogeneous polynomial kernel with degree 3. 65
4.6 Comparison of OISVM and LIBSVM on the *Adult7* benchmark, it is solved using a Gaussian kernel. 66

5.1 Sample images from the Caltech-101 database. In the first row they are taken, respectively from the class *Airplanes*, *Car side*, *Faces*, *Leaves* and *Motorbikes*; this subset has been used to compare the standard model and the SIFT. In the second row there are example images from the other categories (*Elephant*, *Gramophone*, *Umbrella*, *Yin Yang* and *Ibis*). Note that all the images are of different sizes, but in the test they have been normalized to have all the same height of 140 pixels, and the width has been rescaled proportionally. 72
5.2 Classification performances using the unsupervised feature selection method proposed, compared with a random selection of subsets of features, on 5 different 15/50 training/test splits. 76
5.3 Difference of classification performances between the random method and the proposed method. 77

List of Tables

3.1 Performance of the recognition system measured from a set of 50 trials. 48

4.1 Comparison of OISVM and LIBSVM on standard benchmarks, solved using a Gaussian kernel. For each benchmark, we report the difference in classification rate with respect to LIBSVM-2 and the percentage of the number of SVs with respect to LIBSVM and LIBSVM-2. The values of η for each dataset have been chosen in order not to loose more than 0.5% accuracy. 65

5.1 Comparison of SIFT and standard model classification performances on a subset on the 101-Caltech database. The mean classification rates $\pm$ standard deviation are shown for each datasets, on 5 random splits $15/50$ of training/test images. 73

5.2 Mean number of support vectors for different numbers of features. The number of SVs for the unsupervised feature selection and random selection are more or less the same, while for OISVM it is exactly equal to the number of features because a linear kernel is used. 78

References

Abrams, R. and Dobkin, R. Inhibition of return: effects of attentional cuing on eye movement latencies. *Journal of Experimental Psychology: Human Perception and Performance*, 20(3):467–477, 1994

Bach, F. R. and Jordan, M. I. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*. 2005

Barlow, H. B. Possible principles underlying the trasformations of sensory messages. In W. A. Rosenblith, ed., *Sensory Communication*, pages 217–234. MIT Press, 1961

Baudat, G. and Anouar, F. Feature vector selection and projection using kernels. *Neurocomputing*, 55(1-2):21–38, 2003

Behnke, S. *Hierarchical Neural Networks for Image Interpretation*, vol. 2766 of *Lecture Notes in Computer Science*. Springer, 2003

Bell, A. J. and Sejnowski, T. J. The 'indipendent components' of natural scenes are edge filters. *Vision Research*, 37:3327–3338, 1997

Billock, V. A. Cortical simple cells can extract achromatic information from the multiplexed chromatic and achromatic signals in the parvocellular pathway. *Vision Research*, 35:2359–2369, 1995

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In D. Haussler, ed., *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM press, 1992

Boughorbel, S., Tarel, J.-P., and Fleuret, F. Non-mercer kernels for SVM object recognition. In *Proceedings of British Machine Vision Conference (BMVC'04)*, pages 137–146. London, England, 2004

Breazeal, C., Edsinger, A., Fitzpatrick, P., and Scassellati, B. Active vision for sociable robots. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 31(5):443–453, 2001

Buccigrossi, R. W. and Simoncelli, E. P. Image compression via joint statistical characterization in the wavelet domain. *IEEE Transactions on Image Processing*, 8(12):1688–1701, 1999

Burges, C. J. C. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998

Burges, C. J. C. and Schölkopf, B. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems*, pages 375–381. 1996

Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*, pages 409–415. 2000

Chang, C.-C. and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Collobert, R., Sinz, F., Weston, J., and Bottou, L. Trading convexity for scalability. In *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*, pages 201–208. ACM Press, New York, NY, USA, 2006

Coppola, D. M., Purves, H. R., McCoy, A. N., and Purves, D. The distribution of oriented contours in the real world. *PNAS*, 95:4002–4006, 1998

Cox, D. and O'Sullivan, F. Asymptotic analysis of penalized likelihood and related estimators. *Ann. Statist.*, 18:1676–1695, 1990

Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. Cambridge University Press, 2000

Csató, L. and Opper, M. Sparse representation for gaussian process models. *Advances in Neural Information Processing Systems*, 13, 2001

De Smet, P. and Pires, R. L. V. Implementation and analysis of an optimized rainfalling watershed algorithm. In B. Vasudev, T. R. Hsing, A. G. Tescher, and R. L. Stevenson, eds., *Proceedings SPIE Vol. 3974, Image and Video Communications and Processing 2000*, pages 759–766. 2000

Downing, C. and Pinker, S. The spatial structure of visual attention. In M. I. Posner and O. S. M. Marin, eds., *Attention and Performance XI: Mechanisms of attention*, pages 171–187. Erlbaum, Hillsdale, NJ, 1985

Downs, T., Gates, K. E., and Masters, A. Exact simplification of support vectors solutions. *Journal of Machine Learning Research*, 2:293–297, 2001

Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000

Eckhorn, R., Bauer, R., Jordan, W., Brosch, M., Kruse, M., Munk, W., and Reitboeck, H. J. Coherent oscillations: A mechanism of feature linking in the visual cortex? *Biological Cybernetics*, 60:121–130, 1988

Egly, R., Driver, J., and Rafal, R. Shifting visual attention between objects and locations: evidence for normal and parietal subjects. *Journal of Experimental Psychology: General*, 123:161–177, 1994

Elder, J. H. and Goldberg, R. M. Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2:324–353, 2002

Engel, Y., Mannor, S., and Meir, R. Sparse online greedy support vector regression. In *Proceedings 13th European Conference on Machine Learning*. 2002

Engel, Y., Mannor, S., and Meir, R. The kernel recursive least squares algorithm. *IEEE Transactions on Signal Processing*, 52(8), 2004

Fadiga, L., Fogassi, L., Gallese, V., and Rizzolatti, G. Visuomotor neurons: ambiguity of the discharge or 'motor' perception? *Internation Journal of Psychophysiology*, 35(2–3):165–177, 2000

Fidler, S., Berginc, G., and Leonardis, A. Hierarchical statistical learning of generic parts of object structure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 182–189. IEEE Computer Society, Washington, DC, USA, 2006

Field, D. J., Hayes, A., and Hess, R. F. Contour integration by the human visual system: evidence for local "association field". *Vision Research*, 33(2):173–193, 1993

Fischer, M. H. and Hoellen, N. Space- and object-based attention depend on motor intention. *The Journal of General Psychology*, 131:365Ű–378, 2004

Fitzpatrick, P. and Metta, G. Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society: Mathematical, Physical and Engineering Sciences*, 361(1811):2615–2185, 2003

Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193Ű–202, 1980

Geisler, W. S., Perry, J. S., Super, B. J., and Gallogly, D. P. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41:711–724, 2001

Gibson, B. S. and Egeth, H. Inhibition of return to object-based and environment-based locations. *Perception and Psychophysics*, 55(3):323–339, 1994

Graf, A. B. A., Wichmann, F. A., Bülthoff, H. H., and Schölkopf, B. Classification of faces in man and machine. *Neural Computation*, 18:143–165, 2006

Gray, C. M., König, P., Engel, A. K., and Singer, W. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338:334–336, 1989

Grossberg, S. and Mingolla, E. Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. *Perceptual Psychophysics*, 38:141–171, 1985

Guy, G. and Medioni, G. Inferring global perceptual contours from local features. *Int. J. of Computer Vision*, 20:113–133, 1996

Hofstadter, D. R. *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books, 1999

Hoyer, P. O. and Hyvärinen, A. A multilayer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593–1605, 2002

Hung, C., Kreiman, G., Poggio, T., and DiCarlo, J. Fast readout of object identity from macaque inferior temporal cortex. *Science*, 310:863–866, 2005

Hyvärinen, A., Hoyer, P. O., and Oja, E. Image denoising by sparse code shrinkage. In S. Haykin and B. Kosko, eds., *Intelligent Signal Processing*. IEEE Press, 2001

Itti, L. and Koch, C. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001a

Itti, L. and Koch, C. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, 2001b

Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998

Johnson, S. P. Building knowledge from perception in infancy. In L. Gershkoff-Stowe and D. Rakison, eds., *Building object categories in developmental time*, pages 33–62. Erlbaum, 2005

Keerthi, S. S., Chapelle, O., and DeCoste, D. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, 8:1–22, 2006

Keerthi, S. S. and DeCoste, D. A modified finite newton method for fast solution of large scale linear SVMs. *Journal of Machine Learning Research*, 6:341–361, 2005

Kimeldorf, G. and Wahba, G. A correspondence between bayesian estimation of stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41:495–502, 1970

Klein, R. M. Inhibitory tagging system facilitates visual search. *Nature*, 334:430–431, 1988

Knutsson, H. Representing local structure using tensors. In *Proceedings 6th Scandinavian Conference on Image Analysis*, pages 244–251. Oulu, Finland, 1989

Kohavi, R., Becker, B., and Sommerfield, D. Improving simple bayes. In *Proceedings of the European Conference on Machine Learning*. 1997

Krüger, N. Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129, 1998

Lampl, I., Ferster, D., Poggio, T., and Riesenhuber, M. Intracellular measurements of spatial integration and the max operation in complex cells of the cat primary visual cortex. *Journal of Neurophysiology*, 92(5):2704–2713, 2004

Laskov, P., Gehl, C., Krüger, S., and Müller, K.-R. Incremental support vector learning: analysis, implementation and applications. *Journal of Machine Learning Research*, 7:1909–1936, 2006

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998

Lee, M.-S. and Medioni, G. Grouping ., -, $\rightarrow$, θ, into regions, curves and junctions. *Journal of Computer Vision and Image Understanding*, 76(1):54–69, 1999

Lee, Y. J. and Mangasarian, O. L. RSVM: Reduced support vector machines. In *Proceedings of the SIAM International Conference on Data Mining*. 2001

Leventhal, A. *The Neural Basis of Visual Function: Vision and Visual Dysfunction, vol. 4*. CRC Press, 1991

Li, M. and Sethi, F.-I. K. Confidence-based active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(8):1251–1261, 2006

Li, X., Yuan, T., Yu, N., and Yuan, Y. Adaptive color quantization based on perceptive edge protection. *Pattern Recognition Letters*, 24:3165–3176, 2003

Li, Z. A neural model of contour integration in the primary visual cortex. *Neural Computation*, 10:903–940, 1998

Lowe, D. G. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, vol. 2, pages 1150–1157. IEEE Computer Society, Washington, DC, USA, 1999

Mallot, H. A., von Seelen, W., and Giannakopoulos, F. Neural mapping and space-variant image processing. *Neural Networks*, 3(3):245–263, 1990

Marr, D. *Vision: a computational investigation into the human representation and processing of visual information*. W. H. Freeman, San Francisco, 1982

Martin, D., Fowlkes, C., and Malik, J. Learning to detect natural image boundaries using local brightness, color and texture cues. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 26(5):530–549, 2004

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings 8th Int'l Conf. Computer Vision*, vol. 2, pages 416–423. 2001

Melcher, D. and Kowler, E. Shapes, surfaces and saccades. *Vision Research*, 39:2929–2946, 1999

Metta, G. and Fitzpatrick, P. Early integration of vision and manipulation. *Adaptive Behavior*, 11:109–128, 2003

Milanese, R., Gil, S., and Pun, T. Attentive mechanisms for dynamic and static scene analysis. *Optical Engineering*, 34:2428–2434, 1995

Mladenić, D., Brank, J., Grobelnik, M., and N.Milic-Frayling. Feature selection using linear classifier weights: Interaction with classification models. In *In The 27th Annual International ACM SIGIR Conference (SIGIR 2004)*, pages 234–241. 2004

Morrone, M. and Burr, D. Feature detection in human vision: A phase dependent energy model. *Proc. Royal Soc. of London B*, 235:221–245, 1988

Mutch, J. and Lowe, D. G. Multiclass object recognition with sparse, localized features. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 11–18. IEEE Computer Society, Washington, DC, USA, 2006

Mylers-Worsley, M., Johnston, W., and Simons, M. The influence of expertise on x-ray image processing. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 14:553–557, 1988

Natale, L. *Linking action to perception in a humanoid robot: a developmental approach to grasping*. Phd, University Of Genoa, 2004

Natale, L., Orabona, F., Metta, G., and Sandini, G. Exploring the world through grasping: a developmental approach. In *Proceedings 6th CIRA Symposium*, pages 27–30. 2005

Nguyen, D. and Ho, T. An efficient method for simplifying support vector machines. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 617–624. ACM Press, New York, NY, USA, 2005

Nowak, E., Jurie, F., and Triggs, B. Sampling strategies for bag-of-features image classification. In *ECCV06*, pages IV: 490–503. 2006

O'Craven, K., Downing, P., and Kanwisher, N. fMRI evidence for objects as units of attentional selection. *Nature*, 401:584–587, 1999

O'Regan, J. Solving the "real" mysteries of visual perception: the world as an outside memory. *Canadian Journal of Psychology*, 46:461–488, 1992

Palmer, S. and Rock, I. Rethinking perceptual organization: the role of uniform connectedness. *Psychonomic Bulletin & Review*, 1(1):29–55, 1994

Poggio, T. and Smale, S. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society*, 50(5):537–544, 2003

Pontil, M. and Verri, A. Properties of support vector machines. *Neural Computation*, 10:955–974, 1998

Posner, M. I. and Cohen, Y. Components of visual orienting. In H. Bouma and D. G. Bouwhuis, eds., *Attention and Performance X: Control of language processes*, vol. 531–556. Erlbaum, 1984

Posner, M. I., Snyder, C. R. R., and Davidson, B. J. Attention and the detection of signals. *Journal of Experimental Psychology: General*, 109:160–174, 1980

Prodöhl, C., Würtz, R. P., and von der Malsburg, C. Learning the gestalt rule of collinearity from object motion. *Neural Computation*, 15:1865–1896, 2003

Pylyshyn, Z. W. Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80(1-2):127–158, 2001

Rensink, R. A. The dynamic representation of scenes. *Visual Cognition*, 7(1/2/3):17–42, 2000a

Rensink, R. A. Seeing, sensing, and scrutinizing. *Vision Research*, 40(10–12):1469–1487, 2000b

Rensink, R. A., O'Regan, J. K., and Clark, J. J. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997

Riesenhuber, M. and Poggio, T. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999

Sandini, G. and Metta, G. Retina-like sensors: motivations, technology and applications. In T. Secomb, F. Barth, and P. Humphrey, eds., *Sensors and Sensing in Biology and Engineering*. Springer Verlag, New York, NY, 2002

Sandini, G., Questa, P., Scheffer, D., and Mannucci, A. A retina-like cmos sensor and its applications. In *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2000)*. Cambridge, USA, 2000

Sandini, G. and Tagliasco, V. An anthropomorphic retina-like structure for scene analysis. *Computer Vision, Graphics and Image Processing*, 14:365–372, 1980

Schiele, B. and Crowley, J. Where to look next and what to look. In *Proceedings of the Conf. on Intelligent Robots and Systems (IROS'96)*, pages 1249–1255. 1996

Schmidt, K., Goebel, R., Löwel, S., and Singer, W. The perceptual grouping criterion of collinearity is reflected by anisotropies of connections in the primary visual cortex. *European Journal of Neuroscience*, 5(9):1083–1084, 1997

Scholl, B. J. Objects and attention: the state of the art. *Cognition*, 80:1–46, 2001

Schwartz, E. L. Spatial mapping in the primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, 25(4):181–194, 1977

Sela, G. and Levine, M. D. Real-time attention for robotic vision. *Real-Time Imaging*, 3:173–194, 1997

Serre, T., Wolf, L., and Poggio, T. Object recognition with features inspired by visual cortex. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 994–1000. IEEE Computer Society, Washington, DC, USA, 2005

Sigman, M., Cecchi, G. A., Gilbert, C. D., and Magnasco, M. O. On a common circle: Natural scenes and gestalt rules. *PNAS*, 98(4):1935–1940, 2001

Simoncelli, E. and Olshausen, B. Natural images statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001

Smirnakis, S. M., Berry, M. J., Warland, D. K., Bialek, W., and Meister, M. Adaptation of retinal processing to image contrast and spatial scale. *Nature*, 386:69–73, 1997

Steinwart, I. Sparseness of support vector machines. *Journal of Machine Learning Research*, 4:1071–1105, 2003

Sun, Y. and Fisher, R. Object-based visual attention for computer vision. *Artificial Intelligence*, 146:77–123, 2003

Tarr, M. J. and Bülthoff, H. H. Image-based object recognition in man, monkey and machine. *Cognition*, 67:1–20, 1998

Tipper, S. P. Object-centred inhibition of return of visual attention. *Quarterly Journal of Experimental Psychology*, 43A:289–298, 1991

Tipper, S. P. Object-based and environment-based inhibition of return of visual attention. *Journal of Experimental Psychology: Human Perception and Performance*, 20(3):478–499, 1994

Treisman, A. M. and Gelade, G. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980

Vapnik, V. N. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998

Vincent, L. and Soille, P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991

Viola, P. and Jones, M. J. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004

Vonikakis, V., Gasteratos, A., and Andreandis, I. Enhancement of perceptually salient contours using a parallel artificial cortical network. *Biological Cybernetics*, 94:194–214, 2006

Wallraven, C., Caputo, B., and Graf., A. Recognition with local features: the kernel recipe. In *Proceedings of ICCV'03*. 2003

Walther, D. and Koch, C. Modeling attention to salient proto-objects. *Neural Networks*, 19:1395–1407, 2006

Wan, S. and Higgins, W. Symmetric region growing. *IEEE Transactions on Image Processing*, 12(9):1007–1015, 2003

Wertheimer, M. Untersuchungen zur lehre von gestalt. *Psycholgishe Furschung*, 4:301–350, 1923

Wolfe, J. M. Moving towards solutions to some enduring controversies in visual search. *Trends in Cognitive Sciences*, 7:70–76, 2003

Wolfe, J. M. and Gancarz, G. Guided search 3.0 basic and clinical applications of vision science. In V. Lakshminarayanan, ed., *Basic and Clinical Applications of Vision Science*, pages 189–192. Kluwer Academic, Dordrecht, Netherlands, 1996

Wu, M., Schölkopf, B., and Bakir, G. A direct method for building sparse kernel learning algorithms. *Journal of Machine Learning Research*, 7:603–624, 2006

Yantis, S. Control of visual attention. In H. Pashler, ed., *Attention*, pages 223–256. Psycology Press, 1998

Yantis, S. and Jonides, J. Abrupt visual onsets and selective attention: Evidence from visual search. *Journal of Experimental Psychology: Human Perception and Performance*, 10:601–621, 1984

Yarbus, A. L. *Eye movements and vision*. Plenum Press, New York, 1967