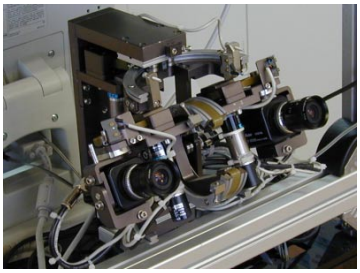
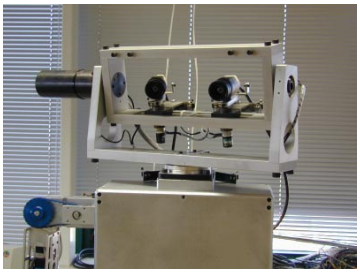
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO





Binocular Head Control with Foveal Vision : Methods and Applications

Alexandre José Malheiro Bernardino (Mestre)

Dissertação para a obtenção do Grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor José Alberto Rosado dos Santos Victor

Júri:

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutor Giulio Sandini
	Doutor João José dos Santos Sentieiro
	Doutor João Manuel Lage de Miranda Lemos
	Doutor Helder de Jesus Araújo
	Doutora Maria Isabel Lobato de Faria Ribeiro
	Doutor José Alberto Rosado dos Santos Victor

Abril de 2004

Agradecimentos

O primeiro e mais sentido agradecimento vai para o meu orientador científico, Prof. José Alberto Santos–Victor, pelo constante apoio e motivação dado ao longo destes anos, quer em termos profissionais quer pessoais. Em particular sinto-me honrado por ter beneficiado das suas invulgares qualidades de organização, relacionamento pessoal, pragmatismo, visão de futuro, e capacidade de identificar e perseguir as direcções mais promissoras de investigação. A sua dedicação ao laboratório de Visão (VisLab) permitiu criar um excelente relacionamento entre todos os membros, e com isso proporcionar trabalho científico de mérito reconhecido internacionalmente.

Agradeço vivamente à direcção do Instituto de Sistemas e Robótica (ISR), em particular ao seu director, Prof. João Sentieiro, por ter conseguido criar as condições necessárias e suficientes, quer em termos materiais quer humanos, ao desempenho de trabalho de investigação científica de excelência, como demonstrado pelas comissões de avaliação nos últimos anos.

A todos os colegas da Secção de Sistemas e Controlo do IST, e colegas de investigação no ISR, agradeço as excelentes relações profissionais e pessoais que sempre me proporcionaram, tornando a missão conjunta ensino/investigação extremamente motivadora e gratificante. Não posso deixar de agradecer individualmente aos colegas do VisLab, com os quais tive o prazer de trabalhar e conviver ao longo dos anos (a ordem é arbitrária): Gaspar *Labmate*, César *Kubrick*, Nuno *Homogracias*, Etienne *Marciano*, Manuel *Braço Forte*, Plinio *Arepas*, Raquel *Volta Sempre*, João Paulo *Multibody*, João Maciel *Matcher*, Ricardo *SpaceMan*, Niall *Omnidireccional*, Vitor *Party Animal*, Sjoerd *van der Blimp*, António *Fitipaldi*, Jordão *Xutos*, Matteo *Johny Bravo*, Vicente Javier *Castellon*, Roger e Anastácia *Capixaba*, *Zedufes* Viana, Rodrigo *Barbecue*, Cláudia *Chique*, Eval *Salsa*, Carlos *Ducados*, Diego *Maratona*, Lenildo *Moedinha*, Sandra *Smiley*, Javier *Globetrotter* e Jonas *Santa-Maria*. Aos outros, agradecerei quando encontrar alcunhas adequadas ;-)

Vorrei anche ringraziare per l'ospitalita' e l'aiuto tutti gli amici del Lira Lab. Giulio, Giorgio, Adonis, Carlos, Riccardo, Lorenzo e Ingrid. Spero di rivedervi presto!

A um título mais pessoal mas não menos importante, agradeço aos amigos de longa data, Marco e Cláudia, César e Cecília, Pedro e Susana, pela horas amizade e companhia que sempre me proporcionaram. Que a sorte nos acompanhe a todos, e que tenhamos oportunidade de a partilhar.

Aos meus pais, Mariana e António, e ao meu irmão, Sérgio, obrigado pelo apoio e amor que sempre me dedicaram, a todos os níveis, e que me esforcei por retribuir. Os laços familiares fortes que me proporcionaram foram, sem dúvida, essenciais para a motivação e energia com que realizei esta tese.

E, por último na lista, mas em primeiro no coração ...

à Helena

Abstract

The work in this thesis aims at the visual control of binocular robot heads with foveal images. Due to the complexity of visual processing in general settings, many biological systems have retinas with a small unique high resolution area called "fovea". To be able to perceive the whole environment, the observer uses attentional mechanisms to detect points of interest in the periphery of the visual field, and repositions the fovea to those points using eye movements. This strategy requires adequate oculomotor control mechanisms and efficient perceptual capabilities. The work in this thesis explores foveal vision, eye mobility, attentional mechanisms and efficient perceptual processing to develop a set of basic capabilities for the operation of a binocular head in realistic scenarios. We provide important contributions in the aspects of oculomotor control, foveal sensor design, depth perception, motion estimation and selective visual attention. In the overall, we demonstrate the applicability and efficiency of foveal vision in all involved perceptual aspects. Both computational and algorithmic advantages are illustrated along the thesis, and contribute toward the real-time operation of active artificial visual systems.

Keywords

Foveal Vision, Binocular Heads, Visual Servoing, Depth Estimation, Motion Estimation, Visual Attention.

Resumo

O trabalho descrito nesta tese visa o controlo visual de cabeças binoculares com imagens foveais. Devido à complexidade do processamento visual em situações genéricas, muitos sistemas biológicos apresentam retinas com apenas uma pequena região de alta acuidade visual, chamada *fovea*. Para que seja possível ter uma percepção global de todo o ambiente circundante, o observador utiliza mecanismos de atenção para detectar pontos de interesse na periferia, e movimentos oculares para reposicionar a fovea nesses pontos de interesse. Esta estratégia requer um controlo adequado dos movimentos oculares e um conjunto eficiente de capacidades perceptuais. O trabalho apresentado nesta tese explora os aspectos da visão foveal, da mobilidade ocular, dos mecanismos de atenção e do processamento eficiente da informação visual para desenvolver um conjunto de capacidades basicas que permitam o funcionamento das cabeças binoculares em cenários realistas. São apresentadas contribuições importantes ao nível do controlo oculomotor, do projecto de sensores foveais, da percepção de profundidade, da estimação de movimentos e da atenção visual selectiva. Em geral, demonstra-se a aplicabilidade e eficiência da visão foveal em todos os aspectos perceptuais abordados. Ao longo da tese são ilustradas as suas vantagens, quer computacionais, quer algorítmicas, que contribuem para o funcionamento em tempo-real de sistemas activos de visão artificial.

Palavras Chave

Visão Foveal, Cabeças Binoculares, Controlo Visual, Estimação de Profundidade, Estimação de Movimento, Atenção Visual.

Contents

Chapter 1

Introduction

The goal of this work is the development of techniques for the visual control of binocular heads. Binocular heads are agile visual systems designed to provide robots with the perceptual capabilities required for real-time operation in non-structured environments. However, due to the complexity of visual processing in general conditions, real-time functionality in artificial agents is currently difficult to achieve with reasonable computer power.

On the contrary, biological systems exhibit exceptional performances in hard natural environments due to a parsimonious and purposive allocation of visual resources to obtain only the relevant information for the task at hand. Probably, the main reason for such economy of resources is the non-uniform processing of the different parts of the visual field. In mammals, for instance, eyes have a single high acuity visual area in the center of the retina called *fovea*, and the remaining field of view is observed with much smaller resolution. This fact is compensated with attentional mechanisms that detect interesting points in low-resolution peripheral areas and trigger eye movements to reposition the gaze direction toward the interesting points.

This foveal, agile and attentive structure of the visual system is widely explored in this thesis. We show that the complementarity of these aspects contribute to the real-time operation of artificial systems in a diversity of visual processing tasks.

1.1 The Approach

The reason for exploring ideas motivated from biological systems is that living animals are a good example of robustness of operation in a multitude of situations, having visual behaviors highly competent in general purpose tasks. In particular we look at some biological facts from the primates' visual system to address many of the visual perception and control related aspects in this thesis. Examples are the decomposition of ocular movements in vergence, smooth-pursuit and saccades, the foveal structure of the retina, the disparity and orientation selective neuronal structures in the visual cortex, the operation of the visual attention system, among others.

A very simplified diagram of the basic visual control architecture addressed in this thesis can be seen in Fig. 1.1. We will dedicate a chapter to each of the modules presented. Foveation is the process that compresses image representation by lowering its resolution at the periphery. This strategy effectively reduces the computational complexity of the perceptual processes addressed in the thesis: depth perception, motion estimation and selective attention. Visual measurements obtained by these processes are then used to

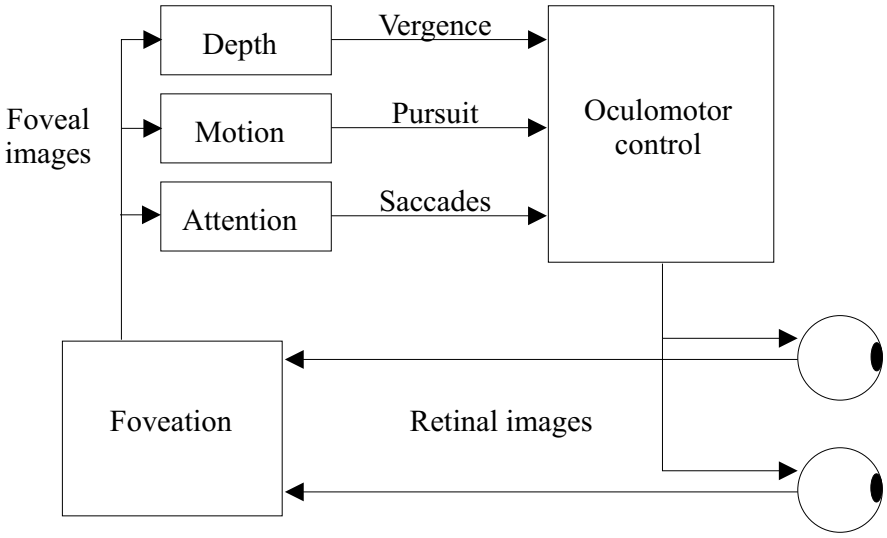


Figure 1.1: Basic architecture of binocular head control.

control, respectively, the vergence, smooth-pursuit and saccade eye movements of the binocular head. Depth estimation computes relative distance of objects from the camera, and is used to control vergence eye movements. Motion estimation obtains position and velocity measurements to control head pan and tilt motions. Selective attention mechanisms extract points of interest from the images that constitute candidates for triggering saccade gaze shifts. All algorithms use foveal images.

1.1.1 Foveation and Receptive Fields

The term *Foveation* comes from *Fovea*, the high resolution part of the retina of many animals. On such animals, high visual acuity only exist on a very small angular range in the center of the visual field. The space variant nature of the visual system is present not only in the retina but also in many of the higher level visual pathways through the brain. For example, a space variant allocation of resources also happens in visual attention - visual events happening on non-attended regions of the visual fields may remain unnoticed.

From the retinal level to highly complex regions in the visual cortex of primates, neurophysiology research has found biological computational elements that exhibit a "foveal" organization i.e. represent more densely and acutely certain parts of visual field. These computational elements are called *Receptive Fields* (RF) and, according to [51], are probably the most prominent and ubiquitous computational mechanism employed by biological information systems.

RF's can model many visual operations in the lower-level areas of the mammalian visual system and most of them seem to be related to sensory coding. For instance, some types of retinal ganglion cells have RF profiles that resemble Difference-of-Gaussians, coding image in terms of contrast. In some cells of the visual cortex, profiles are like Gabor functions that code image in terms of oriented edges.

The complexity of RF profiles seem to increase to higher levels of the visual pathway. In some areas of the inferior-temporal cortex of monkeys some cells resemble object-like features like faces or hands. Although some facts are still very controversial, it is accepted that RF's from a very important feature of brain information processing.

Photoreceptors

In the human eye, at the very first level, reflectance information is obtained by retinal photo receptor cells called **cones** and **rods**. *Cones* are color sensitive cells (exist in three different types with peak responses at different wavelengths) with very high concentration in the central area of the retina. They require medium illumination levels to be activated, being used mainly in daylight vision. There are approximately 5 million cones in the human retina. On the contrary, *rods* require very low illumination and are not color sensitive. They are more numerous than cones (about 100 million) and are distributed along the retina in a different fashion. Fig. 1.2 shows a microscope picture of *cones* and *rods* and their approximate radial distribution in the human retina. Notice that rods are absent from the fovea. This is the reason why astronomers look to very dim stars with the periphery of the eye, where rods exist in higher densities. Photo-receptors are very

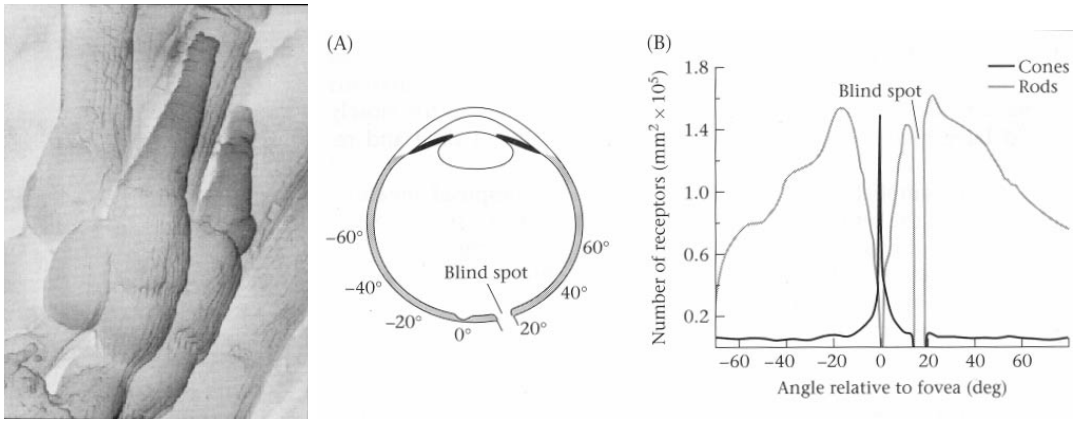


Figure 1.2: Left: *Rod* and *Cone* cells. *Rods* have cylindrical shape and *cones* have conical shape. Right: Radial distribution of *rods* and *cones* in the human retina. Reprinted from [155].

localized receptive fields and have an almost purely integrative function. They can be considered as point wise brightness and color sensors.

Ganglion Cells

Between the photo receptor level and the optic nerve (set of nerves transmitting optical information to the brain), there are several levels of neuronal structures. Bipolar, horizontal, amacrine and inter-plexiform cells convey information to the **Ganglion Cells**, that constitute the retinal output level, sending signals, through the optic nerve, to several brain areas for further processing. There may be as many as 20 visual pathways originating in the retina that may serve specialized computational goals [155].

Many types of ganglion cells exist. The best known types are the *midget* cells, with small dendritic fields, and the large *parasol* cells. They project information to brain areas via the *parvocellular* and *magnocellular* visual streams, respectively. The parvocellular stream seems to be specialized in high-resolution color vision, while the magnocellular stream has low-resolution and color blindness but high contrast sensitivity and fast dynamic response.

Midget and parasol ganglion cells can be modeled as receptive fields with a circular *center-surround* structure. The spatial support includes several photo receptors and is divided in two regions: a central region where the input enhances/inhibits the output;

and a surround region where the role of the input is reversed. Fig. 1.3 shows a model of the center-surround mechanism. Cells that are activated with a light center and dark surround are called *On* cells and in the opposite case are called *Off* cells. These cells, that compute the contrast between the center and the surround, are robust to changes in the average luminance of the scene and can be considered contrast feature extractors. Other types of cells just collect the average luminance within its support regions [133], or show highly non-linear responses to contrast reversing gratings [47], computing contrast magnitude over wide regions. This last type of receptive field can be modeled by full-wave rectification and averaging, and will be subject of further discussion later in this document.

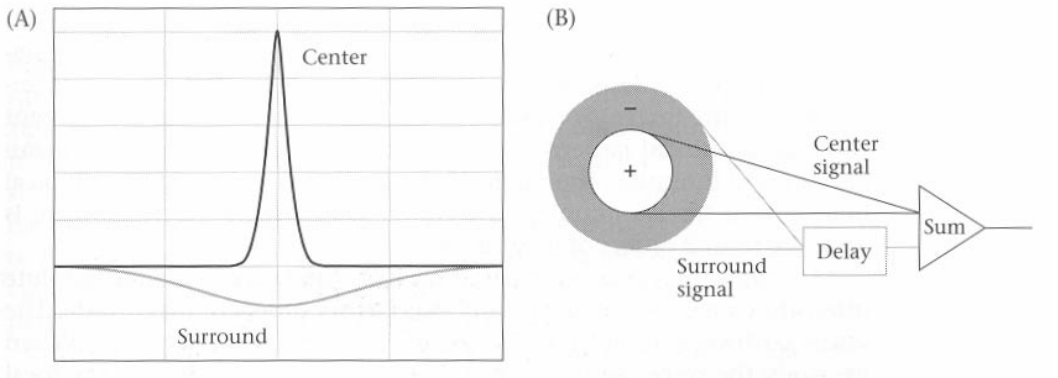


Figure 1.3: (A) The response of a linear retinal ganglion cell in the spatial (radially symmetric) domain and (B) temporal domain. Reprinted from [155].

The Primary Visual Cortex

While retinal neurons have circularly symmetric receptive fields, in the primary visual cortex most receptive fields exhibit some form **orientation selectivity**. They respond better to visual items of certain orientations and some of them are binocular (receive input from both eyes).

Two major types of cells can be found in V1 (one of the early areas of the primary visual cortex): *simple* and *complex* cells. Simple cells extract oriented features by collecting data from aligned circular center-surround cells in the LGN, an intermediate brain area between the retina and visual cortex (see Fig. 1.4). Complex cells are also orientation-selective but show a non-linear response. A common model to complex cells is the weighted spatial summation of the quadrature response of simple cells with same preferred orientation and spatial frequency [115]. These cells seem to be tuned to particular spatial gratings and may be involved in edge detection and texture segregation tasks.

Retino-Cortical Mapping

Signals reaching the cortex from the retina follow three basic organizational principles: the eye of origin, the class of ganglion cell and the spatial position of the ganglion cell within the retina [155]. In what concerns the design of foveation methods, we are mostly interested in the latter principle.

Biological findings in the visual cortex of monkeys [43] show that the displacement of a light stimulus in the retina produces displacements in the cortex that are inversely

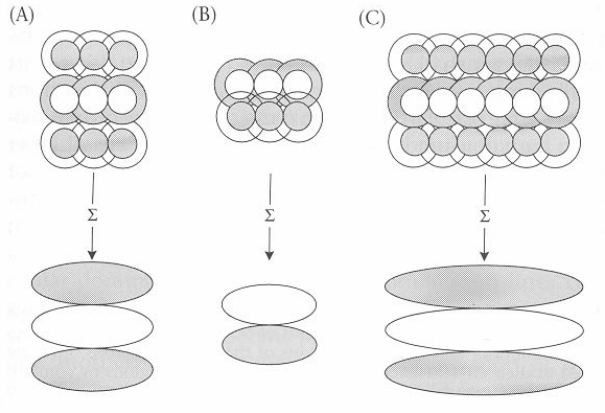


Figure 1.4: Simple cells are orientation-selective by linearly combining the responses of non-oriented receptive fields. Reprinted from [155].

proportional to distance to the fovea. This effect is also known by *cortical magnification*. This property indicates a general scaling behavior by which both RF spacing and size increase linearly with eccentricity, i.e., distance from the fovea [96]. Later it was found that responses to linear stimulus originating in the fovea lie roughly along lines in the cortex, and circular stimulus centered at the fovea produce linear responses in the cortex at approximately orthogonal orientations [144]. Thus, the information transmitted from the retina to the visual cortex is organized in an approximate logarithmic-polar law [131].

A good synthesis of the constraints on receptive field size distribution and retino-cortical mapping known from psychophysics, neuroanatomy and electro physiology can be found in [89], and are the following:

- The diameter of the smallest receptive field is proportional to eccentricity;
- At any eccentricity all diameters greater than the corresponding smallest unit are present;
- Mean receptive field size increases linearly with eccentricity;
- The transformation from the visual field to the cortex is logarithmic and the visual cortex seems rather homogeneous;
- At any retinal location many receptive field sizes are present but smaller fields are located more centrally;
- The relative overlap of receptive fields is independent of eccentricity.

Distributing receptive fields over the retina according to the previous assumptions, we obtain the "sunflower model" [89] shown in Fig. 1.5. In the visual cortex, these receptive fields are mapped into a constant size uniform distribution (also in Fig. 1.5).

These general principles have guided the main stream of research on biologically motivated foveation techniques. In the thesis, sensor design is motivated by human foveal vision to reduce the complexity of visual information processing. We introduce a computational foveation model, denoted *receptive field foveation*, that is able to describe a large percentage of the methods reported in the literature. In particular, we focus on the logpolar foveation model that is based on the retino-cortical mapping of the primates' visual system. Most of the existing methods disregard the analysis of aliasing distortions

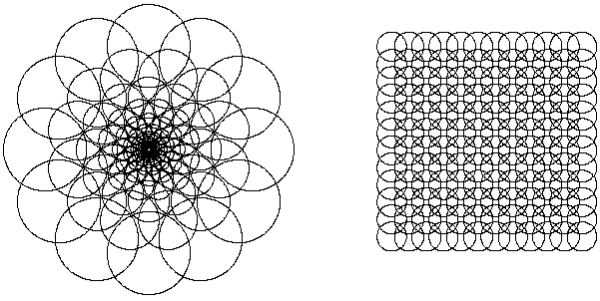


Figure 1.5: The "sunflower model". (Right) Receptive fields in the retina have sizes proportional to distance to the fovea (eccentricity) and the relative overlap between them is constant. (Left) In the visual cortex, retinal receptive fields are mapped into RF's with constant size and uniform distribution, according to a logarithmic-polar law.

caused by inadequate distribution and shape of receptive fields. Henceforth, we propose the *smooth logpolar transform*, a foveation method with highly overlapping receptive fields that significantly reduce aliasing distortions. Foveal vision is used in all visual processing methods developed in this work, and the demonstration of its usability and efficiency is one of the main contributions of the thesis. With respect to visual attention mechanisms, feature extraction methods are also motivated by the shape and function of retinal and cortical receptive fields. Spatial frequency and orientation features are extracted from the images using analysis elements similar to the non-linear ganglion cells in the retina and the directionally selective units in the visual cortex. Also, the principles of depth perception are deeply rooted on the function of disparity selective cells and the competitive facilitation/inhibition mechanisms existing in cortical binocular regions.

1.1.2 Oculomotor Control

The foveal structure of the eyes has a reduced visual acuity in the periphery of the visual field. Though it provides a significant economy of processing resources, visual events and objects in the periphery cannot be easily discriminated and recognized. Thus, to observe in detail peripheral items, the visual system requires the ability to frequently move the high resolution fovea to other places in the visual field. In the human visual system there are three main types of ocular movements by which the observer fixates and tracks interesting objects in the field of fiew: vergence, smooth–pursuit and saccades. For a detailed description of these movements see [34].

Vergence is the conjugate eye movement that controls the depth of the gaze point – eyes move in opposite directions and are controlled by depth cues such as disparity, focus, object size, and others. Observers can voluntarily engage or disengage on vergence eye movements. However, in normal circumstances, vergence is performed automatically on objects in the gaze direction, such that object depth and shape discrimination are stable and reliable.

Smooth–pursuit is a velocity driven eye movement that stabilizes the image of an object in the retina. It is used both for tracking moving objects and for fixating on static objects when the observer is moving. One particularity of this movement is the lack of voluntary control by the observer. In normal circumstances humans are not able to drive smooth-pursuit movements in the absence of proper velocity stimuli. This behavior happens automatically to compensate the retinal slip and stabilize perception on the fixated object,

unless a voluntary action drives attention away to other locations in the visual field.

Saccades are fast, ballistic eye movements that change the gaze direction in a step–like manner, and are driven by position based stimuli. They are used in small corrective movements to compensate velocity errors in smooth-pursuit movements, but their most interesting use is in overt visual searching. Due to the foveal nature of the retinas, objects in the periphery of the visual field cannot be recognized with precision. Thus, saccade movements change the gaze direction, several times each second, to reposition the high resolution fovea in potential objects of interest.

In terms of oculomotor control, this work follows the motion decomposition rules suggested by the human visual system. Vergence movements control the conjugate motion of the cameras. Version movements (composed of smooth-pursuit and saccades) control the head pan and tilt joints. In kinematics terms, vergence and version movements decompose the geometry of ocular movements in depth and fronto-parallel components. In dynamics terms, smooth-pursuit and saccades decompose the control strategy into velocity-based and position based. We apply these concepts in a *Visual Servoing* framework [53], and show that, under tracking conditions, system dynamics can be described directly in terms of image plane visual features in a decoupled fashion, which greatly simplifies the controller design problem. The simplicity and standardization of the approach allows extreme flexibility in the type of controllers that can be developed. We will illustrate the methodology with a simple proportional controller to drive saccade motions but, for smooth-pursuit control, we include a motion estimator to compensate steady-state tracking errors. The application of such controllers is tested and evaluated in a real robot binocular head.

In perceptual terms, the stimuli used to control vergence and smooth-pursuit movements is also motivated by their biological counterparts. Vergence is controlled by dominant disparity around the fixation point. Smooth-pursuit is controlled by dominant motion of retinal centered regions. The computation of such stimuli is naturally aided by the foveal geometry of the retina, that allocates a higher number of computational resources to the analysis of regions closer to the center of the image. Thus, on tracking and fixation scenarios, both depth perception and motion estimation are naturally focused on the object of interest, which attenuates the influence of background distracting elements and improves the robustness of the methods.

In this work, depth perception is addressed *via* dense disparity estimation using a Bayesian formulation similar to [22]. Besides adapting the formulation to foveal images, we propose a methodology to overcome local estimation ambiguities, using fast low-pass filters to propagate disparity information over neighboring regions in intermediate computational steps. This approach naturally favors smooth disparity surfaces but still allows the representation of depth discontinuities.

Motion estimation is developed with a parametric estimation technique similar to [72]. Besides adapting the algorithm to foveal images, several improvements are introduced. The optimization procedure is reparameterized such that gradient information is computed only once, thus saving significant online computational resources. Convergence range is increased through the use of a redundant parametrization of motion parameters, but more representative of the set of plausible image deformations. Robustness is improved *via* damped least squares and a hierarchical organization of the computations: more constrained (and stable) parameters are estimated first and serve as starting points for the estimation of more noise sensitive parameters. The algorithms make use of direct image gray level values, instead of optical flow, to avoid estimation drifts that are typical of velocity based techniques.

Although much is currently known about the driving stimuli for vergence and smooth–

pursuit movements (mostly involuntary), the way by which saccade movements are triggered is still an open issue. The reason is that many cognitive aspects, involving the motivational and emotional state of the observer, as well as task related aspects, are involved in saccade eye control. An interesting model is presented in [56], whose diagram is shown in Fig. 1.6. The model is divided vertically in spatial and temporal streams,

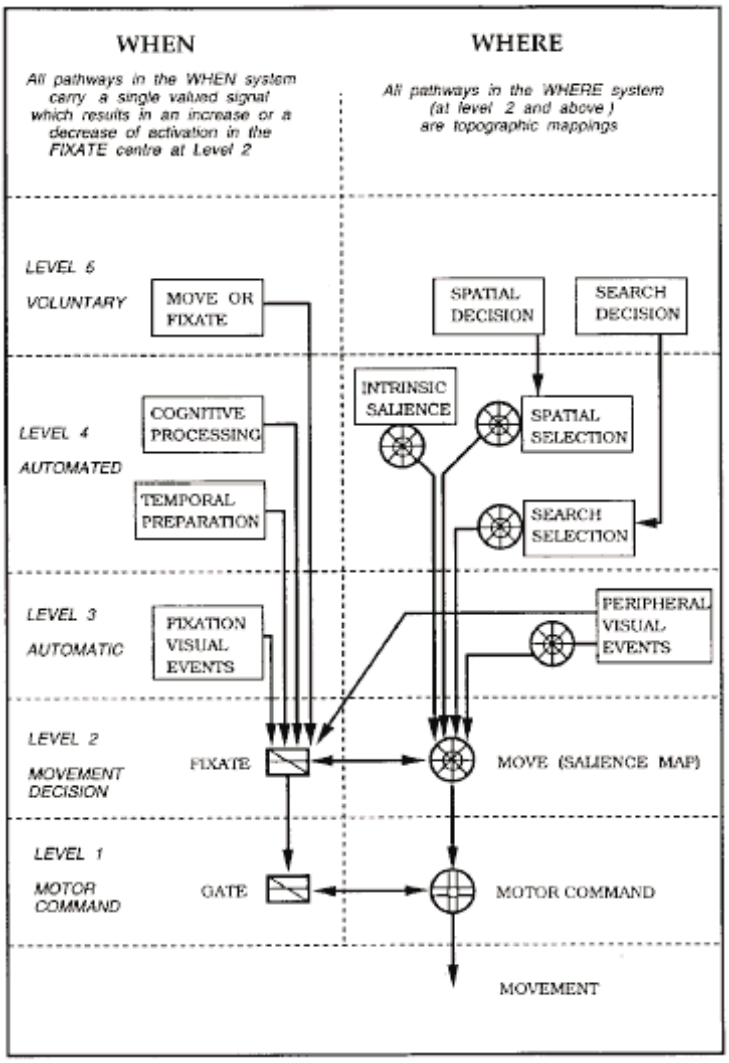


Figure 1.6: A model for the generation of saccade eye movements. From [56].

and horizontally by a hierarchy of levels going from involuntary to voluntary control. The decision of where and when to trigger a saccade movement is highly dependent on the voluntary levels of the model. Notwithstanding, the decision is aided by lower level modules that are mostly data driven. In this thesis, we address the automated spatial level of the model, where bottom-up influence, in the form of visual saliency, provides a means of detecting interesting objects to observe. This is the subject of selective visual attention mechanisms.

1.1.3 Visual Attention

Because foveal systems have reduced visual accuity in the periphery of the field of view, discrimination and recognition capabilities are diminished with respect to the fovea. Even

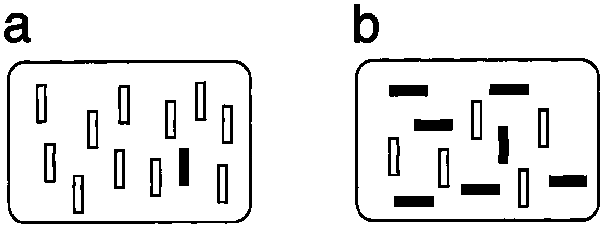


Figure 1.7: On searching the black vertical bar, subjects respond faster in case (a) than (b).

with coarse visual acuity, attention mechanisms are responsible for the detection and selection of potentially interesting regions, requiring detailed observation. The repositioning of the fovea to such "points of interest" proceeds to a detailed inspection of the object, confirming or rejecting possible expectations. This capability is fundamental both to accomplish a certain task or to react to unexpected events.

Two main types of attentional control are commonly found in the literature: **selective attention** and **divided attention**. Selective attention addresses the question of what features or locations attract attention, constituting candidates for further visual inspection. Divided attention focus on how visual resources are allocated to certain regions or objects in the visual field when different stimuli compete for attention. This mode of attention determines the regions where to allocate more attentive power, facilitating the detection of visual objects and events.

Selective Attention and Visual Search

Visual search is one of the main trends of research in psychophysics. The topic is of the uttermost importance in both biological and artificial vision systems because practically all visual tasks involve some sort of search. In picking parts of a bin or looking for someone in a crowd, efficient methods must exist to locate promising items and avoid exhaustively searching the scene. The main question is what are the features or locations in the visual field that minimize the search time for a certain object. The *Feature Integration Theory* of [146] was one of the first attempts to answer the question. The theory suggests that attention must be directed serially to each stimulus in a display whenever conjunctions of more than one separable feature are needed to characterize or distinguish possible objects. This means that, if an object in a display is the unique having a certain feature (it is "salient") then it is very easily detected (it "pops-out"), otherwise the observer must shift attention to other objects in the display searching for distinctive conjunction of features. In psychophysics, a common experiment to validate this theory is to ask for subjects to search for certain objects in a display, and measure their reaction times. For example, consider the situations depicted in Fig. 1.7, where subjects are asked to find the unique vertical black bar on the display. When a vertical black bar is among white bars (case (a)), reaction times are much faster that when it is among horizontal black bars and vertical white bars (case (b)). On the basis of this theory is was postulated that the visual attentional system is composed by two phases:

- A parallel pre-attentional phase builds a saliency map from binding several separable feature maps. If a visual item is the only one with strong activation in one of the

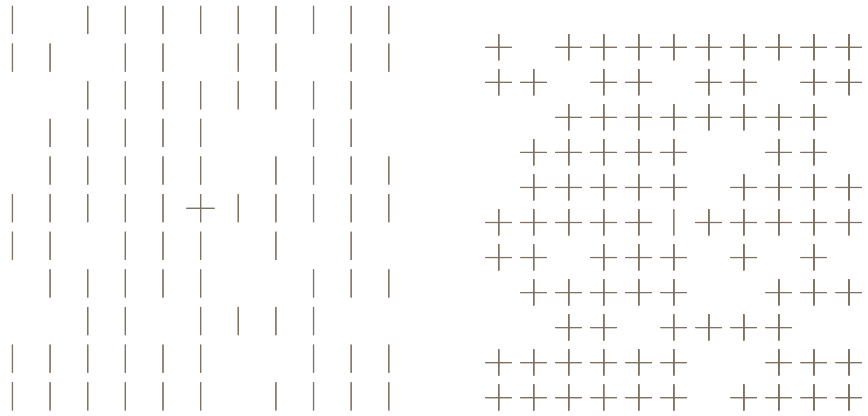


Figure 1.8: A cross among vertical bars is easier to find that a vertical bar among crosses. This happens because crosses and bars are not separable features (a cross contains a vertical bar in its shape).

feature maps, then it "pops-up" and is identified in constant time (independent of the number of distractors). This phase is fast, effortless and has a large field of view.

- A serial attentional phase scans out the saliency map sequentially searching for conjunction of features. This phase is effortful, has small visual field at each scan, and depends on prior knowledge and motivational issues.

It is important to notice that the theory is only valid for the conjunction of **separable** features. For example, in Fig. 1.8, the crosses and the vertical bars are not separable features because a cross contains also a vertical bar. In that case, it is easier to find a cross among bars than a bar among crosses because the cross is the only visual item containing a horizontal bar. Low-level features like color, orientation, curvature, motion and disparity are computed by distinct specialized areas of the cortex and, for that reason, are usually considered as separable. However, horizontal and vertical orientation features are represented in the same cortical area and interact closely. Hence, can not be considered separable.

Whenever the target is not unique in at least one of its features, a sequential search must be performed, leading, in the worst case, to exhaustively scanning all visual items. In the experimental studies of [160], it was found that human reaction times in the serial search phase were actually smaller that it was predicted by the *Feature Integration Theory* of [146]. It was proposed the *Guided Search Hypothesis*, where pre-attentional and attentional mechanisms are coupled – the parallel mechanism is used to guide the search, thus reducing the overall reaction time. The *Biased Competition* model of [48] also proposes the combination of pre-attentional and attentional mechanisms: bottom-up sources, that arise from sensory stimuli present in a scene, indicate where objects are located and which features are present at each location; and top-down sources, that arise from the current behavioral goals, weight the incoming bottom-up stimulus information to allow attention to be biased toward one bottom-up input over another. Top-down constraints are required to resolve the competition among the bottom-up inputs.

In this work, selective visual attention is based on the biologically motivated computational model of [88]. The goal is to detect interesting points in the visual field where to shift gaze to. Interesting points are defined as local maxima of a saliency map, that incorporates conspicuity information from many feature dimensions. We adapt the saliency computa-

tion algorithm to foveal images, but propose that low-level feature extraction should be made in the cartesian domain to preserve important high-frequency information. This is motivated by the existence of non-linear ganglion cells in the human retina, that compute spatial-frequency features even at peripheral retinal locations. Also, we develop a novel fast algorithm to extract directionally selective features using Gabor filters. Orientation, spatial frequency and luminance features are used to compute bottom-up visual saliency. We briefly describe how top-down attentional modulation can be performed, for simple objects, by heuristically selecting the more representative features of that class of objects. This topic will be further explored in future work.

Divided Attention

Another stream of research in visual attention mechanisms studies how attentional resources are distributed along spatial or feature dimensions. Its was know since the work of Helmoltz in 1865 that visual attention can be directed to certain parts of the visual field without moving the eyes but it was the work of [117] that started the modern theories of attention allocation. In his experiment, subjects were told to fixate the central part of a blank display. Then a peripheral cue was briefly flashed at one side of the display and after some time an object was finally displayed. It was found that reaction times were faster when the object appeared in the same part of the display as the cue was flashed. Thus the cue was able to attract the attentional focus to its neighborhood and facilitate the detection of posterior events in the same visual region. In another experiment the cue had the shape of an arrow and was placed in the central part of the display. Objects displayed in the part indicated by the arrow direction were detected faster than objects at the opposite side. Based on this fact it was postulated the *Spot-light* model of attentional control, composed by two orienting mechanisms:

- The exogenous or reflexive mechanism is engaged by peripheral cues, is very fast ($\approx$ 100 ms) and occurs even with uninformative cues.
- The endogenous or voluntary mechanism is engaged by central informative cues and is slower ($\approx$ 300 ms).

This model was refined by [92], that proposed a *variable spot-light* model where the area covered by the spot-light can be increased or decreased. In [52] it was shown that the efficiency of visual processing has an inverse relationship with the size of the the attentional focus (the *zoom-lens model*).

Many works, however, have shown results contradictory to spot-light kind of models. For example in [120] it was shown that humans can simultaneously track and index 4 to 5 moving objects among distractors, independent of their retinal location. Also [35] have shown that attention can be split is non-connected visual regions, and [84] have shown that subjects can attend to ring like formations, thus providing evidence against the spot-light models.

Other researchers are favorable to the hypothesis that, rather than spatial locations, are full objects and perceptual groups that attract attentional resources. For example, [50] displayed two overlapped objects (a box with a line struck to it) and asked subjects to identify two features of the objects (line texture, line orientation, box height or the location of a gap in the box). When the features were relative to the same object, decision was made without mutual interference, while decisions involving different objects were slower to perform. It was concluded that subjects cannot attend simultaneously to different objects, even though they occupy the same visual region. [9] used the flanker effect to show that

grouping factors influence the allocation of attention. The ability to discriminate letters among distractors in a display is affected if distractors share a common feature. Grouping effects spread attention over the whole group and less resources are allocated to individual elements in the group.

Recent works provide more and more evidence against pure spatial attention. In [19], two targets with different features are displayed at the same spatial location (overlapped). It is shown that humans can keep track of one of the targets solely on the basis of its trajectory along color, orientation, and spatial-frequency dimensions. It was concluded that attention is allocated to feature dimensions, instead of spatial locations.

In [151] the biased competition framework of [48] is extended for object-based segregation and attentional control. They show several examples that demonstrate that both bottom-up and top-down information cooperate to define perceptual groups and bias some groups to be more easily perceived than others. In the bottom-up stream, features group together based on the gestalt principles [158] of similarity, proximity, symmetry, good continuation and closure. In the top-down stream three sources of information influence object segregation and object selection: (1) recognition of familiar objects; (2) task bias; and (3) endogenous spatial attention processes (pre-cues). All these sources of information compete and cooperate to allocate visual attention to one or several objects in the scene.

There is still a lot of debate on this issue but it is clear that many segmentation and grouping processes influence the distribution of attention [49]. Although not formulated in terms of visual attention mechanisms, there are many computer vision techniques proposed to address the segmentation and grouping problem, like optimization in graphs [164, 3, 135, 61], histogram clustering [119], tensor voting [102], neural modeling [94] and level set methods [137]. Though promising, generic segmentation and perceptual grouping methods are still too time consuming to be used in real-time systems. However, for some particular tasks and scenarios, it is possible to develop simplified methodologies for divided attention. For example, in face detection algorithms, an initial step of color segmentation is performed to detect regions in the field of view having skin color. Then attention must be divided between these regions to search for other features that are characteristic of faces but not of other body regions (hands).

In this work we propose that, for some tasks, initial object segmentation can be performed by binocular disparity. This may be of use, for instance, on manipulation tasks, where objects are close enough to the system and can be reliably detected by stereo algorithms. Thus, beside controlling vergence eye movements, out depth perception algorithm is used for the initial segmentation of close range objects, that can be used to define the "attentional windows" where to perform additional operations.

1.2 Target Applications

The thesis deals with three fundamental perceptual aspects with important applications on robot behavior control:

- Depth perception – permits the estimation of the range of objects in the scene, with applications in classical robot behaviors like obstacle avoidance and short-range object manipulation.

- Motion Estimation – permits fixation on static or moving objects, with applications in tracking, ego-motion estimation or visual stabilization.

- Selective Attention – permits the selection of highly salient points in the environment where to anchor gaze or search for particular objects.

Though emphasis is put on binocular systems, the motion estimation and selective attention algorithms can be applied to systems with a single camera.

Search-and-rescue, service and entertainment are activities where robotics research is currently targeting applications. Contrary to industrial robots, where the lack of sensor capabilities and mechanical degrees of freedom require the preparation and modification of the working space, modern robots are characterized by rich perceptual systems and many mechanical degrees of freedom. Service robotics aims at the development of robots to assist humans in hospitals, museums, houses (for elderly and disable people). Search-and-rescue robotics explores the application of robots in catastrophe contexts, e.g. fires and earthquakes. Entertainment robots are being developed by major companies and research institutions as a means of marketing and pushing technological and scientific advances for future applications and commercialization. In every case, the dynamics and unpredictability of the scenarios require robots equipped with strong perceptual and motor resources, capable of navigating in the environment, avoiding obstacles, detecting and recognizing objects and events, interpreting situations and planning accordingly their actions.

As humanoid robots attract more and more research efforts, binocular robot heads become ubiquitous. The interest on humanoid robots is twofold: in one hand humanoid robots have an appearance that facilitates the interaction and acceptance in human social environments; in the other hand, anthropomorphic designs benefit from the accumulated knowledge in biological systems that, through evolution, have found "fine tuned" solutions to many problems of operation in natural environments. Large technological companies have humanoid robot prototypes, like the Sony's QRIO [166], Honda's ASIMO [167], ZMP's PINO [169] or Kawada Industries' HRP-2P [168]. Academic research is also exploring the field through laboratory prototypes like Cog [26], Babybot [103], Hermes [18], Robovie [86] or the Humanoid Robot of the Erato Kawato Dynamic Brain Project [5] (see Fig. 1.9).

In current artificial visual systems, despite the lowering price and growing capability of video cameras and processors, the huge amount of information provided by the visual sensor still pose real-time implementation problems. For example, Cog is operated by 32 PC 800 MHz computers [58], Hermes has a network of TMS 320C40 Digital Signal Processors, Babybot uses 4 PC's just for the visual processes, and Robotvie has 2 PC's at 2.4 GHz. However, the implemented visual functions are far from those expected and not robust enough to put into end-user applications. Better information processing strategies are required to extract the most of visual sensors without overwhelming the capabilities of current processor architectures. Through the use of foveal vision, the work described in this thesis pursuit the goal of real-time operation in realistic scenes with parsimonious computational resources.

1.3 Choices and Assumptions

Color Vision

The diversity of aspects related to visual perception sometimes forces researchers to simplify some aspects in order to concentrate on others. A common simplification is the use of distinctly colored objects to simplify segmentation and recognition problems. Though, in general we are not against this approach, in this work we chose **not** to simplify the vision

Figure 1.9: Humanoid Robots. From top-left to bottom-right: QRIO, ASIMO, PINO, HRP-2P, Cog, BabyBot, Hermes, Robovie, Erato Kawato Humanoid Robot.

problem. First, because we are addressing the applicability of robot visual systems in realistic, non-modified, environments. Second, because object segmentation and recognition are open problems far from being solved and assuming its simplification could render the problems trivial, hiding important issues on the gap between low and high-level vision.

Cognitive Influence

The behavior of biological visual systems is highly influenced by cognitive aspects, hard to model and replicate. Cognitive influences include the agent's historical record, and motivational or emotional states. An agent's behavior is affected by whether the agent has previously experienced similar situations and what outcomes those situations produced. The emulation of these aspects in artificial systems would require long operation times and complex learning methodologies. Though we present a model for the control of each individual movement, the way by which they are planned and orchestrated is still a open problem, and are not addressed in this thesis.

1.4 Organization of the Thesis

The thesis is divided in 7 chapters. The first and last chapters introduce and conclude, respectively, the work presented in the thesis. The middle chapters present the scientific work developed. There is not an individual chapter for the presentation of results. Each of the scientific chapters present their own results. Rather technical details or auxiliary results are put onto 5 appendices, at the end of the thesis.

The scientific work is organized in the following sequence. We begin, in Chapter 2 to address the process of creating non-uniform resolution representations of images, used in the remainder of the thesis. Chapter 3 concentrates on ocular movements and the pure control aspects of the binocular head. Chapter 4 is devoted to depth perception in binocular systems, where we present the dense disparity estimation algorithm in foveal images.

The problem of motion estimation and tracking is addressed in Chapter 5, with a parametric motion estimation algorithm improved and adapted to foveal images. Chapter 6 addresses the subject of selective visual attention, to identify points of interest in images where to drive saccade eye movements.

Besides presenting the main conclusions of this work, Chapter 7, also refers to open problems that will be subject to future research.

1.5 Summary of Contributions

The main contribution of the thesis is the demonstration of the efficiency and applicability of foveal vision in problems such as depth perception, motion estimation and selective visual attention. For this purpose, complete algorithms are developed and tested in realistic image sets and/or real robot setups.

Other important contributions include:

- Foveal sensor design – existing foveation methods have little concern with sampling theory aspects. We propose a foveation method that reduces the amount of aliasing in the process, based on overlapping Gaussian receptive fields. A fast approximate algorithm to implement the method is also presented.

- Robot dynamics control – under the assumption of small deviations from tracking, we show that robot dynamics can be controlled directly from image plane features, in a simple and decoupled fashion.

- Parametric motion estimation – we improve the computational efficiency of existing parametric motion estimation algorithms by reformulating the problem in time fixed coordinates. Convergence range and robustness are also improved by employing a redundant parameterization and organizing computations hierarchically.

- Local orientation analysis – we have developed a fast implementation of Gabor filters that extract oriented features from images. The method overcomes the efficiency of state-of-the-art algorithms.

- Retino–cortical saliency computation – we show that low-level feature extraction should be performed in retinal coordinates, mainly for features containing high spatial-frequency content. The remaining saliency computation steps should be performed in logpolar space due to computational efficiency.

Chapter 2

Foveation

Both natural and artificial visual systems have to deal with large amounts of information coming from the surrounding environment. When real-time operation is required, as happens with animals or robots in dynamic and unstructured environments, image acquisition and processing must be done in a few milliseconds in order to provide fast enough response to external stimuli. Appropriate sensor geometries and image representations are essential for the efficiency of the full visual processing stream.

Two parameters are of high importance in visual sensor design: **resolution** and **field of view**:

- resolution determines the scale of the smallest details that can be detected in the images;
- field of view determines how far from the central view point objects can be detected.

Computer vision systems usually control efficiency by controlling image size, either by reducing resolution or the field of view:

- image resolution is reduced uniformly according to some desired criteria;
- field of view is reduced by defining windows of interest around objects, where further processing is preformed.

These strategies have been applied successfully in some structured environments, where good models for object sizes and motions exist. However, they are too rigid to be applied in more unstructured situations. In the first case, resolution may not be enough to detect objects whose scale changes along time. In the second case, moving objects easily move away from the windows of interest. *Foveation* deals with methods to represent images efficiently, preserving both the field of view and maximum resolution, at the expense of reducing resolution at some parts of the image. With such a representation, the sensory strategy can allocate high resolution areas to objects of interest as they are detected in a wide field of view.

The visual system of many animals exhibits a foveated structure. In the case of mammals, where eyes are able to change gaze direction, retinas present a unique high resolution area in the center of the visual field, called *fovea*. The foveation geometry is fixed and the fovea can be redirected to other targets by ocular movements. The same structure is also commonly used in robot systems with moving cameras [16, 157, 111, 25, 10]. In computer systems with static cameras, some foveation geometries are dynamic, i.e. the fovea can move around to any point in the visual field [62, 7].

This chapter is devoted to the analysis and design of foveation methodologies, bearing in mind its application to artificial systems. The first section describes related work on foveation methods. The second section presents the Smooth Logpolar Transform. It uses highly overlapped receptive fields to reduce aliasing, i.e. image distortion due to inadequate distribution and shape of receptive fields. Third section presents a computationally efficient method, based on fast multi-scale transformations, to approximate the Smooth Logpolar Transform.

2.1 Related Work

In this section we review existing computational models for image foveation. The large majority of models are formulated directly in the spatial domain and are known by the name of *superpixel* methods. Methods in this class have been used mostly by robotics and computer vision people due to the biological motivation and simplicity of implementation. It is easy to customize both the size and geometry of the sensor to match the requirements of particular applications. Other class of methods are based on image multiresolution decompositions. These have been more used in image communication and visualization, and their strength is based on more efficient algorithms for compression and display.

2.1.1 Receptive-Field Foveation

The principle of superpixel foveation methods can be explained in few words: a uniform resolution image (from now on denoted *cartesian* or *original* image), is subdivided in compact regions, maybe overlapping, of arbitrary sizes and shapes. Then, the pixels of the original image belonging to each of those regions are "averaged" with predefined weighting functions and the result is stored in memory. The information contained in each of the regions is compacted into a single value - the *superpixel*.

Instead of *superpixels* we propose a formulation based on the concept of receptive-field, due to its closer biological interpretation. The term *Receptive Field* comes after neurophysiology experiments finding visual cells responsive only to stimulus in a confined region of the visual field. Also, not all parts of the RF contribute equally to the cell response, thus leading to the concept of *RF profile*. Profile functions have a limited spatial support (the region where a stimulus elicits RF responses) and a well defined center or location (where a stimulus elicits the maximal RF response). For a good review of these topics see [51].

In the *superpixel* model, regions subdividing the original image are put in correspondence with the RF spatial support, and the averaging operation is modeled as a inner product of the original image with the RF profile function. This formulation suits most of the methods of this class found in the literature. The notation is the following:

- Let $f(x, y)$ be the original image (a 2D real valued function).
- Let $\{\phi_i(x, y)\}_{i\in\Gamma}$ be a set of 2D real valued functions - the RF profile functions.
- Let $\{(x_i, y_i)\}_{i\in\Gamma}$ be the set of locations of all RF's.
- The output of a receptive field is modeled by its projection (inner product) on the original image:

$$c_i = \langle f, \phi_i \rangle \tag{2.1}$$

- The *Foveal Transform* is modeled by an operator $\mathcal{F}$ that applies to cartesian images and produces a set of coefficients (the foveal code) computed by the image projection in the set of all RF's:
$$\mathcal{F}(f) = \{\langle f, \phi_i \rangle\}_{i \in \Gamma} \tag{2.2}$$
- The *Inverse Foveal Transform* is modeled by the operator $\mathcal{F}^{-1}$ that applies to foveal codes and aims at reconstructing or approximating the original image. Its operation is left unspecified by now.
- The composite operation $\hat{\mathcal{F}} = \mathcal{F}^{-1} \circ \mathcal{F}$ is denoted *Foveal Filtering*.
- The image $\hat{f} = \hat{\mathcal{F}}(f)$ is called *foveated image* or *foveated approximation* to image f.

Supported by these definitions and notation, we will review existing foveation methods that fit in this class.

The Logpolar Transformation

The logpolar transformation is a class of models with some variants. Probably the majority of existing foveation methods fit in this class. It can also be found in the literature under the names *logpolar mapping* or $log(z)$ *model*. Its main properties are:

- Biological Plausibility - it is based on the receptive-field distribution and retino-cortical mapping in mammals' visual system.
- Image Mapping - the foveal transform coefficients are arranged in a 2D image (the so called *logpolar image*) preserving neighborhood relationships, almost everywhere, with respect to the original image.
- Rotation and Scaling "invariance" - when the original image is rotated or scaled with respect to its center, patterns in the *logpolar image* only suffer translations, thus preserving their shape.

The class of foveation methods known as "logpolar transformation" is based on the complex logarithmic function $\log(z)$, which, according to [131], can be used to approximate the retino-cortical mapping of primates. Let us consider the complex retinal and cortical planes, represented by the variables $z = x + jy$ and $w = u + jv$, respectively, where j is the complex imaginary unit:

$$w = \log(z) = \log(|z|) + j \arg(z) = \log(\rho) + j\theta \tag{2.3}$$

ρ is the input eccentricity and θ is the input angle. Image rotations and scalings in the center of the retinal plane become simple translations along the jv and ju axes in the cortical plane, respectively, as shown in Fig. 2.1. A weakness of the $log(z)$ model is the existence of a singularity in the origin such that points in the fovea can not be mapped. Two common solutions are used to overcome this problem: either using a different mapping for the fovea (e.g. the identity mapping) or applying the $log(z + a)$ model. This model was proposed in [132] as a better approximation to the retino-topic mapping of monkeys and cats. The $log(z+a)$ model transforms points in the first quadrant of the retinal plane via:

$$w = log(z + a) \tag{2.4}$$

The mapping for other quadrants is obtained by symmetry. The difference between the $log(z)$ and $log(z+a)$ models is illustrated in Fig. 2.2. The $log(z+a)$ lacks the exact scale

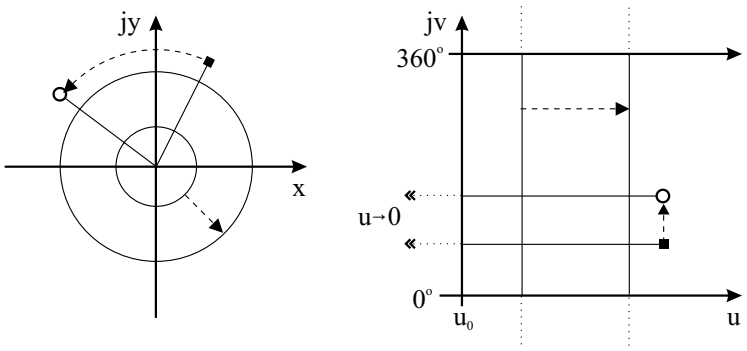


Figure 2.1: The $\log(z)$ model for retino-cortical mapping. The retinal plane (left) is mapped to the cortical plane (right) via $w = \log(z)$.

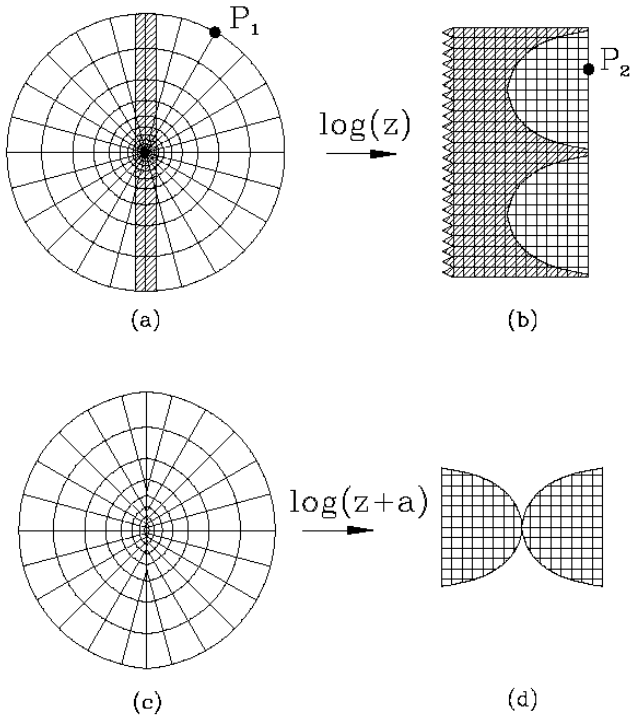


Figure 2.2: The $log(z + a)$ model can be seen as removing the shaded region in (a) from the $log(z)$ model. Reprinted from [154].

invariance property. However, this is often tolerated in practical applications.

The $log(z)$ and $log(z+a)$ are conceptual models defined in continuous coordinates. They tell us how position, size and shape of receptive fields relate between the retinal and cortical domains. In practice the mapping must be discretized and suitable shapes for RF's defined. The conventional approach considers the cortical plane uniformly discretized as a conventional CCD sensor, i.e. covered with dense grid of rectangular RF's with uniform profiles. Thus, let us consider a grid of $E \times A$ rectangular RF's with uniform profiles and boundaries at coordinates $w_{m,n} = \xi_m + i\eta_n, m = 0 \cdots E, n = 0 \cdots A$. Then, in the retinal plane, receptive fields are also uniform and shaped like sections of concentric annulus:

$$\phi_{p,q}(x,y) = \begin{cases} 1 & \text{if} \quad Re[log(x+iy)] \in [\xi_{p-1}, \xi_p] \wedge Im[log(x+iy)] \in [\eta_{p-1}, \eta_p] \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

where $p = 1 \cdots E$ and $q = 1 \cdots A$. This mapping is illustrated in Fig. 2.3.

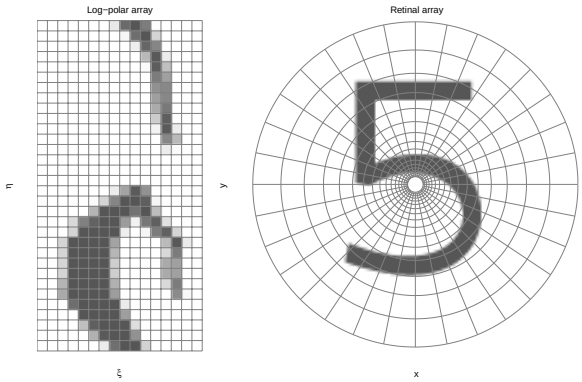


Figure 2.3: A rectangular grid in the cortical plane (left) is mapped to a grid composed of concentric circles and radial lines in the retinal plane (right).

The coefficients of the foveal transform of image $f(x,y)$ are given by:

$$c_{p,q} =< f, \phi_{p,q} > \tag{2.6}$$

With uniform RF's, this operation represents the simple averaging of pixels within each receptive field. The coefficients are then stored in a 2D array with the coordinates (p,q), thus $\mathcal{F}(f)$ is also represented as an image, usually called *logpolar image*. Moreover, neighbor RF's in the retinal domain are also neighbors in the transform domain, except along the angular discontinuity and radial singularity. Shape invariance to centered rotations and scalings no longer holds perfectly for the discretized $log(z)$ model. However the approximation is good enough for practical applications, if discretization is not too coarse (see Fig. 2.4).

Overlapping RF models

Models following more closely biological data have overlapping receptive fields. They are computationally more expensive than non-overlapping ones but gain in smoothness of the foveal transform coefficients. The models presented in [159] and [126] have RF's with a log-polar distribution but with circular shapes and a moderate amount of overlap with the neighbors. [159] proposes a tessellation with a linear relation between receptive field size *vs* eccentricity, and a receptive field overlap of 50%, shown in Fig. 2.5. The proposal in [126]

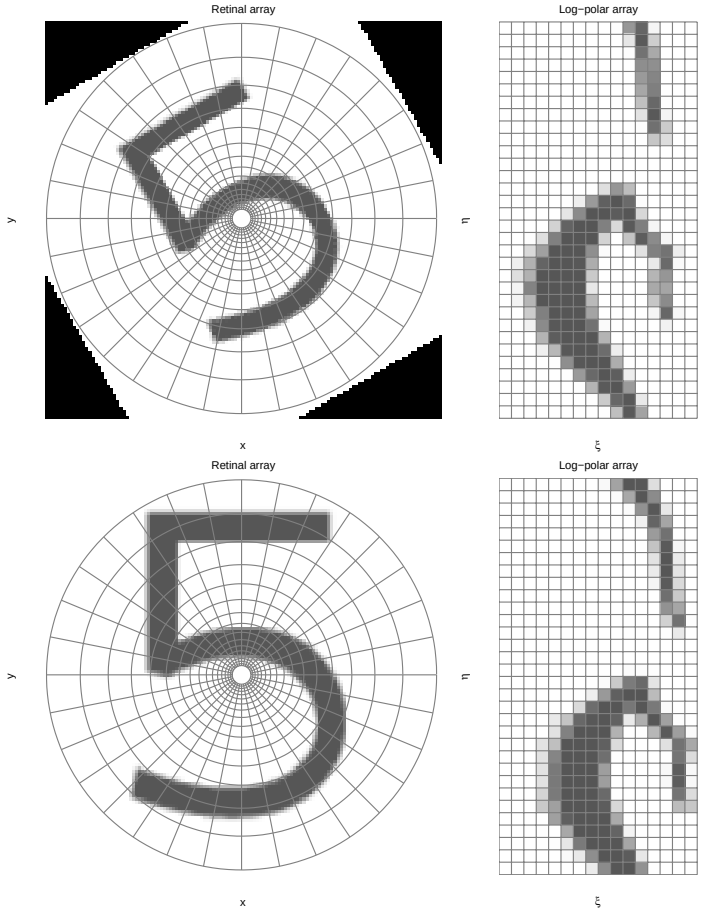


Figure 2.4: In the $log(z)$, the foveal transform coefficients are arranged in a 2D matrix indexed by angular and radial position. Rotated and scaled input images correspond to approximate translations of the foveal transform image.

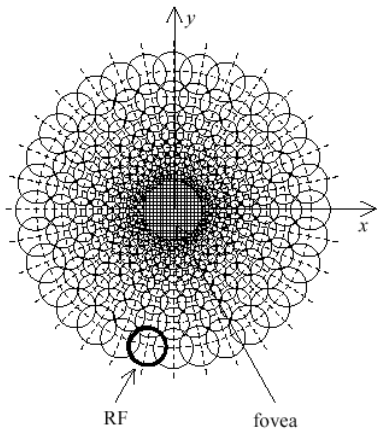


Figure 2.5: The overlapping RF model of [159] implemented in [20].

also uses circular receptive fields but tries to minimize the amount of overlap between them. For this goal they propose a slightly different organization of receptive fields where direct neighbors are not in the same ring – in two consecutive rings the angular positions are shifted by half the angular spacing in each ring.

Alternative Models

All methods described till now have been motivated by the log-polar transformation. However, other coordinate transformations have been used to design space-variant sensors. Though sometimes missing direct biological evidence, they are often very useful in engineering terms and have beneficial properties in particular applications.

An interesting coordinate transformation is the "Reciprocal Wedge Transform" (RWT) presented in [143]. In this method, coordinates (x, y) in the image domain are transformed to the (u, v) domain according to the following law:

$$u = x^{-1},\ v = y \cdot x^{-1} \tag{2.7}$$

Again assuming a uniform partitioning of the (u, v) plane with rectangular RF's, the distribution and shape of RF's in the image plane is as depicted in Fig. 2.6. One of the

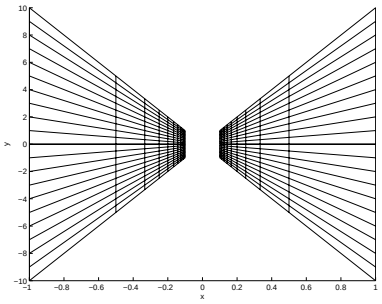


Figure 2.6: The RWT distribution of receptive fields. The central vertical stripe is excluded from the mapping due to the singularity at $x = 0$.

interesting properties of the RWT is the preservation of linear features. This is very useful e.g. for robot navigation where often is necessary to extract lines from the environment. With appropriate calibration, parallel lines in the environment on known planes are mapped to parallel lines in the sensor space. Applications in road navigation and motion stereo are described in [143]. The RWT can be easily implemented in hardware by using two common CCD's parallel to the optical axis and parallel to each other.

The *Dimensionally-Independent Exponential Mapping* (DIEM) [114] is a flexible space-variant sampling where horizontal and vertical image dimensions are sampled independently with exponential laws:

$$\begin{cases} x = \frac{(W-1)}{2}\left(\frac{2u}{S_h-1}\right)^{\gamma_h} \\ y = \frac{(H-1)}{2}\left(\frac{2v}{S_v-1}\right)^{\gamma_v} \end{cases} \tag{2.8}$$

W and H are respectively image height and width. S_h, S_v, u and v are the number of samples and corresponding indexes in the horizontal and vertical dimensions. Some of the advantages of the method are:

1. Flexibility - several topologies can be defined by adequately changing parameters γ_h

and γ_v (see Fig. 2.7).

2. Extensibility - since dimensions are considered separately, it can be easily extended to higher dimensional spaces.
3. Reconfigurability - due to its simplicity, topology can be changed on-the-fly.
4. Preserves horizontal and vertical lines.

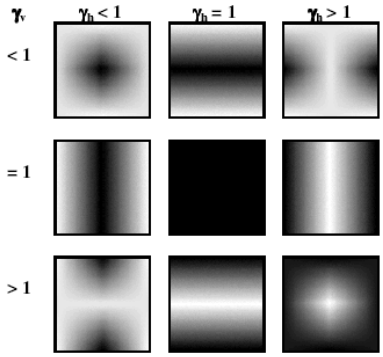


Figure 2.7: The DIEM method allows different foveation topologies depending on parameter values. Bright areas correspond to high sampling density areas.

The *Cartesian Variable Resolution* (CVR) method [7] follow a similar idea but extend the transformation to allow moving and multiple foveas. The base mapping for one fovea is:

$$\begin{cases} u = x_0 + s_x ln(\alpha(x - x_0) + 1) \\ v = y_0 + s_y ln(\alpha(y - y_0) + 1 \end{cases} \tag{2.9}$$

where (x_0, y_0) is the location of the fovea. Then, when multiple (N) foveas are present, the mapped coordinates of a point are computed as a weighted average of individual coordinates, where the weights decrease with distance to the foveas:

$$l_{\text{actual}} = \frac{1}{\sum_{j=1}^{N} \frac{1}{d_j^{\text{power}}}} \times \sum_{i=1}^{N} \frac{l_i}{d_j^{\text{power}}} \tag{2.10}$$

Here l_i are the coordinates calculated using fovea i and d_j is the distance to fovea j.

Irregular Transformations

In the foveation methods described till now, receptive field locations can be defined by 2D coordinate transformations of a rectangular uniform grid, and foveal transform samples can be arranged in matrices, whose indexes represent the lines (m) and columns (n) of the grid. This arrangement is very popular because foveal transform coefficients quasi-preserve neighborhood relationships and can be visualized and processed as regular images. However, there are situations where this organization is not possible. For instance, in the $\log(z + a)$ model, rings have a non constant number of samples (some foveal tessellation methods also have this property) thus not allowing a perfect image like representation. Another example is given in [6], where receptive field locations are defined by a self-organizing recursive method that produces a smooth transition between the foveal and peripheral areas, but cannot be described by a closed-form coordinate transformation.

To deal with situations like these, [154] proposes the Connectivity Graph (CG). A CG is a graph $G = (V, E)$ whose vertexes V stand for receptive field locations and edges E represent the adjacency relations between them. The CG is presented as a general framework for posing image operations in any kind of space variant sensor. The versatility of the representation is illustrated in [154] by developing algorithms for familiar image processing operations like convolution, edge detection, template matching, image translation, rotation, scaling, etc. Fig. 2.8 shows the connectivity graph for a sensor with randomly distributed receptive fields. The work of [10] uses the CG to implement a $\log(z)$ model

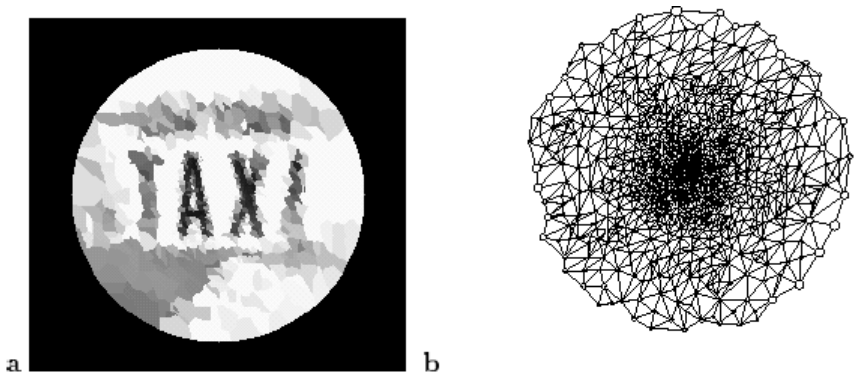


Figure 2.8: Image from a sensor having arbitrary pixel geometry (right). The connectivity graph for this sensor (left). Reprinted from [154].

and process images to control a miniature pan-tilt camera.

2.1.2 Multiscale Foveation

Real-world objects are projected on the retinas in a continuum of sizes, depending on their actual physical dimensions and distance to the observer. Also, each object itself may be composed of details with different sizes and shapes. This inherent "multi-scale" nature of objects has lead researchers to look for efficient image representations to store, access and compare scale-variant visual information. There are, currently, very fast algorithms to decompose one image in its several scales. Pyramids, wavelets and sub-band decompositions are standard image processing tools available in many software packages.

Foveation methods, described till now, are formulated directly in the spatial domain. In computational terms, direct implementation of the spatial domain methods is highly demanding, mainly in the case of overlapping receptive fields. A different class of methods explores the availability of fast multiscale image decompositions to simulate foveation. Applications are more frequent in the field of image transmission and visualization [90, 62, 156, 63] but there are some applications reported in robotics [136, 24, 87]. We denote these methods with the general name of *multiscale foveation* and can be subdivided the following steps:

- Multiscale Coding - image is coded in terms of a multiscale transformation e.g. pyramid, wavelet, or sub-band transforms. This may increase the size of the representation if the transforms are redundant.

- Weighting - multiscale transform coefficients are processed such that high frequencies are preserved in the fovea but attenuated in peripheral areas.

- Foveal Coding - Redundancy is removed in low-frequency areas either by discarding small coefficients or subsampling.
- Reconstruction - The foveated image is reconstructed for visualization.

Multiscale Transforms

Early works on multiscale coding addressed the problem in the context of image communication [30]. The task was to transmit image information at an initial coarse scale, where only the gist of large objects could be perceived, and then gradually transmit the remaining information, filling in the details. Images were decomposed into different frequency bands, with lowest frequencies being transmitted first. Efficient compression algorithms were developed to reduce as much as possible the amount of information to transmit at each frequency band. The Laplacian Pyramid was invented.

In a latter work [29], similar ideas were applied in the context of searching algorithms. Initial solutions were obtained first at coarse levels, providing the starting points for the finest levels. The concept of *Foveated Pyramid* arose. Search algorithms based on these ideas are now called "coarse-to-fine" and are successfully applied in many fields of computer vision, e.g. motion estimation [108], image segmentation [134] and stereo [98].

With the recent theoretical developments on multiresolution signal analysis, pyramids are now part of the more general *wavelet theory* [99]. Some foveation methods are based on wavelet decomposition of images, in particular the discrete wavelet transform (DWT). The wavelet foveation method [36] is of special importance because it establishes a framework for comparing multiscale methods with spatial methods. This method, and a brief introduction to wavelets will be the subject of the second part of this section.

The DWT has been applied very successfully in image coding and transmission, due to its economic orthogonal (non-redundant) signal decomposition. However, it is of limited use in pattern recognition applications due to lack of smoothness in the coefficients for small translations. Non-orthogonal representations based on the linear scale space, although redundant, have better behaved coefficients and are presented in the last part of this section.

The Foveated Pyramid

The **foveated pyramid** is derived from the standard Laplacian pyramid by applying, to each level of the pyramid, a weighting window of the size of the topmost level, so that pixels outside this window can be neglected (Fig. 2.1.2). Since the angular field-of-view decreases as the spatial resolution increases down the layers of the image stack, spatial foveation is thus simulated [136] (see Fig. 2.1.2).

The work in [90] extends the basic design to allow a moving fovea, which involves recomputing the weighting windows to use in each level of the pyramid. Their work aims at reducing image bandwidth for transmission and visualization, and the foveation point is obtained from the end user by a pointing device (mouse or eye tracker). They use a foveated pyramid like structure, whose resolution degradation is designed to match the human eye. However, due to the use of rectangular weighting windows with binary values (0-1), some visual artifacts are noticed. In their following work [62], this problem is addressed. They use a smooth window instead of a rectangular one. The spatial edge artifacts between the regions are eliminated by raised-cosine blending across levels of the pyramid.

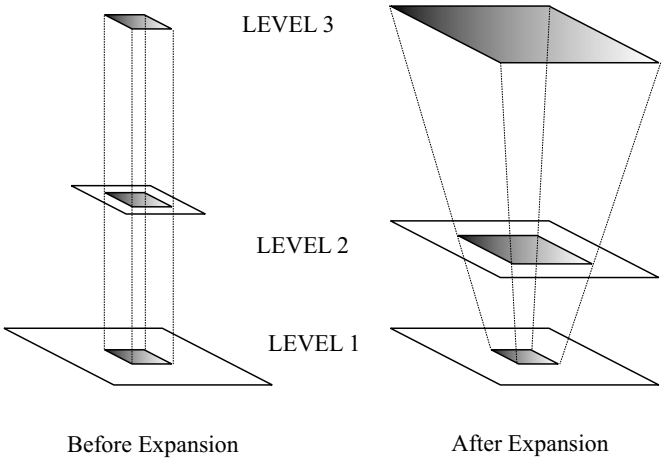


Figure 2.9: The Foveated Pyramid. In computational terms (left) a weighting window of the size of the topmost layer is applied in the center of each layer. Conceptually, this is equivalent to weight large regions in low resolution levels and small regions in high resolution levels, in the expanded pyramid (right).

The foveated pyramid has also found applications in computer vision and robotics. In [136], foveated Laplacian pyramid is used to perform 3D reconstruction and vergence using phase-based methods. A latter work [24] uses a correlation-based method to achieve real-time 3D reconstruction. A similar representation is used in [87], where a foveated stereo setup performs 3D reconstruction by actively fixating points in a surface, recovering depth of those points and integrating information over multiple successive fixations to build a multiresolution map of the surface. In their case, the advantage of a foveated pyramid representation is that vertical disparity is low at all pyramid levels, which facilitates matching algorithms.

Wavelet Foveation

In the last decades, wavelet theory had a significant expansion and is becoming ubiquitous in the signal processing field. It has a very elegant mathematical formulation and generalizes many of the multiresolution image transformations previously discovered. In particular, the Laplacian pyramid can be seen as a wavelet transform. Wavelet theory has derived a very efficient, non-redundant image transform, called Discrete Wavelet Transform (DWT). In [36] a new approach to foveation is introduced: the *Wavelet Foveation*. The technique is based on weighting the DWT coefficients of an image with appropriate functions such that a conventional foveation operator is well approximated. They model the process of foveation by an integral operator:

$$(Tf)(x) = \int_t f(x)k(x,t)dt \tag{2.11}$$

with the kernel:

$$k(x,t) = \frac{1}{w(x)}\phi(\frac{t-x}{w(x)}) \tag{2.12}$$

where ϕ is an averaging function and w a scale factor that depends on spatial position. This fits well into the *Receptive-Field* based methodology presented in the beginning of this section, where ϕ plays the role of a receptive field profile and w its size. The difference

here is that the representation is dense (without sampling), i.e., there exists a RF for each point of the image.

The operator T is denoted the *foveation operator* of *foveation filter* which, with discrete signals is written:

$$(Tf)(i) = \sum_n f(n)k(i,n) \tag{2.13}$$

Defining a logarithmic resolution fall-off, the kernel is given by:

$$k(i,n) = \frac{1}{\alpha|i|}\phi(\frac{n-i}{\alpha|i|}) \tag{2.14}$$

In Appendix C, we present some theoretical facts on the discrete wavelet transform (DWT). From equations (C.7) and (C.8), the DWT representation of a signal f is:

$$f(n) = \sum_i a_J(i)g_{i,J}(n) + \sum_{j=1}^{J}\sum_i d_j(i)h_{i,j}(n) \tag{2.15}$$

where a_j and d_j are the DWT approximation and detail coefficients at scale j, and $g_{i,j}/h_{i,j}$ are the approximation/detail wavelets at scale j and position i. Applying the foveation operator to the previous equation, we obtain:

$$(Tf)(n) = \sum_i a_J(i)(Tg_{i,J})(n) + \sum_j\sum_i d_j(i)(Th_{i,j})(n) \tag{2.16}$$

Thus, the foveated image can be represented as a linear combination of the foveated wavelets. The weights are the DWT transform coefficients of the original image.

The previous formula is not very helpful in computational terms since the the foveation operator does not preserve the translation invariance of the wavelets g and h. Since space invariant convolutions cannot be used, we would have to store all the foveated basis functions.

An efficient approximation can be obtained to the exact DWT foveation filtering expressed in (2.16). Let us also express the foveated basis in terms of its DWT:

$$(Tg_{i,J})(n) = \sum_k \alpha^i(k)g_{k,J}(n) + \sum_k\sum_l \beta_l^i(k)h_{k,l}(n) \tag{2.17}$$

$$(Th_{i,j})(n) = \sum_k \gamma_J^{i,j}(k)g_{k,J}(n) + \sum_k\sum_l \delta_l^{i,j}(k)h_{k,l}(n) \tag{2.18}$$

The $\alpha^i(k)$,$\beta_l^i(k)$,$\gamma^{i,j}(k)$ and $\delta_l^{i,j}(k)$ are the DWT coefficients of the foveated wavelets, given by:

$$\begin{cases} \alpha^i(k) = \langle Tg_{i,J}, g_{k,J}\rangle \\ \beta_l^i(k) = \langle Tg_{i,J}, h_{k,l}\rangle \\ \gamma^{i,j}(k) = \langle Th_{i,j}, g_{k,J}\rangle \\ \delta_l^{i,j}(k) = \langle Th_{i,j}, h_{k,l}\rangle \end{cases} \tag{2.19}$$

The approximation is derived by showing that [36]:

$$\begin{cases} \alpha^i(k) \approx 1, i = k \\ \alpha^i(k) \approx 0, i \neq k \\ \beta^{i,j}(k) \approx 0, \forall i, j, k \\ \gamma_l^i(k) \approx 0, \forall i, l, k \end{cases} \tag{2.20}$$

Therefore the foveated image can be approximated by:

$$(Tf)(n) \approx \sum_i a_J(i) g_{i,J}(n) + \sum_j \sum_i \sum_k \sum_l d_j(i) \delta_l^{i,j}(k) h_{k,l}(n) \tag{2.21}$$

Furthermore, the $\delta_l^{i,j}(k)$ are like shown in Fig. 2.1.2. Neglecting the off-diagonal terms, a

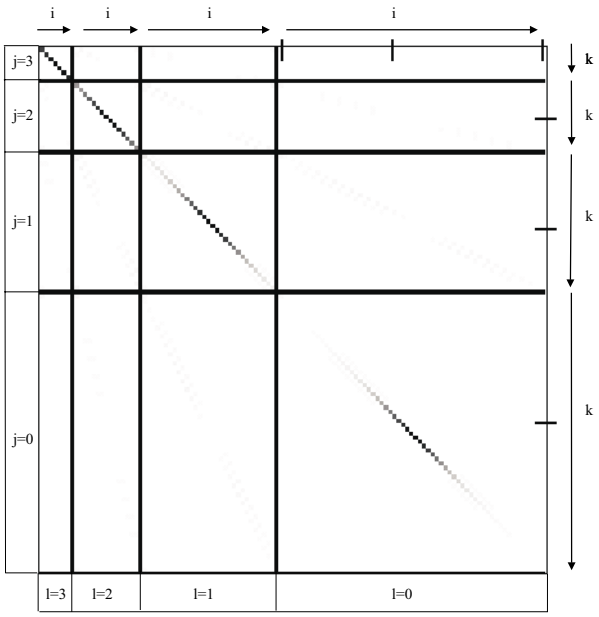


Figure 2.10: DWT coefficients of the foveated wavelets $\delta_l^{i,j}(k)$. Coefficients in the diagonal dominate the representation. Adapted from [36].

new approximation for the foveated image is:

$$(Tf)(n) \approx \sum_i a_J(i) g_{i,J}(n) + \sum_j \sum_i d_j(i) \delta_j^{i,j}(i) h_{i,j}(n) \tag{2.22}$$

Eqs. (2.15) and (2.22) suggest a computational procedure to obtain an approximation to image foveation: weight each scale j of the DWT coefficients by the "windows" $\delta_j^{i,j}(i)$. For 1D signals, their shapes are shown in Fig. 2.11. For images (2D signals), the foveation weighting windows are shown on Fig. 2.12.

An interesting point in this approach is that it provides a formal justification to a method used empirically before: windowing the different levels in a multiresolution representation to simulate foveation. However, [36] only derives the shape of the weighting functions and quality of the approximation to the case of orthogonal wavelet transforms. Also, a particular definition of the foveation operator is used. A different approach is proposed in [156], that determines the foveation masks for certain viewing conditions based

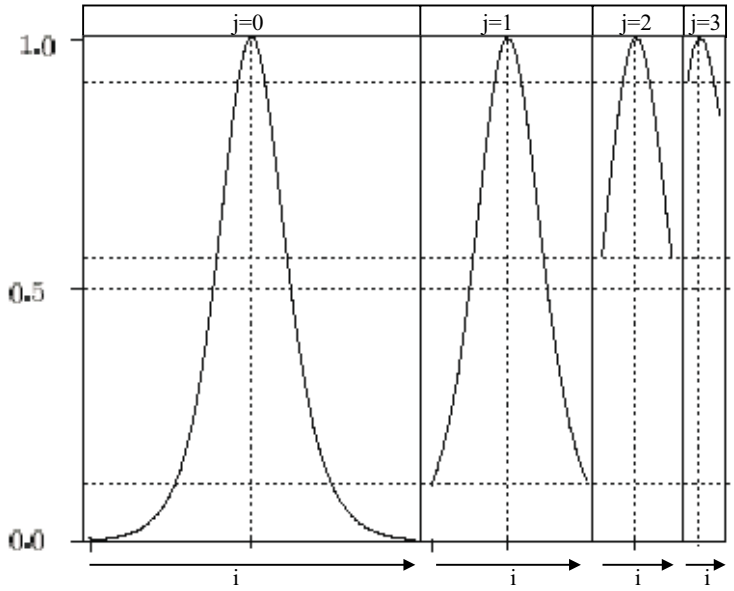


Figure 2.11: The foveation windows $\delta_j^{i,j}(i)$. Notice self-similarity along scales. Adapted from [36].

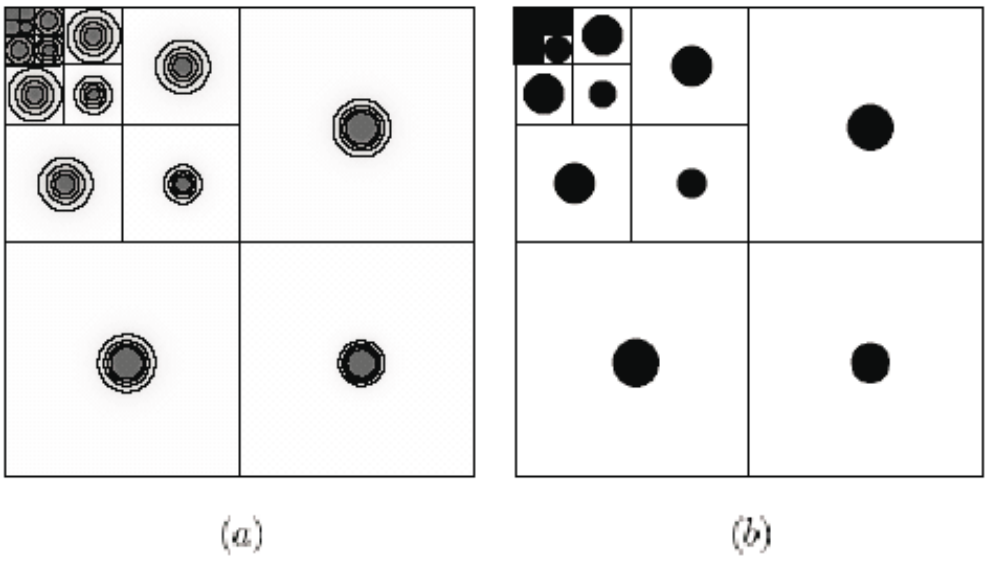


Figure 2.12: The foveation windows for 2D signals. The figure in the left shows the contour plot of the exact coefficients. In the right, the coefficients are rounded to binary values. Adapted from [36].

on the human contrast sensitivity function. This is used in image coding and transmission, to evaluate the perceptual importance of DWT coefficients - the most important wavelet coefficients are encoded and transmitted first.

DWT Shortcomings

The DWT is a non-redundant representation of a discrete signal, i.e. the number of transform coefficients is equal to the number of samples. It has found many applications in image coding and compression but its application to computer vision has been of limited success. One of the reasons is its sensitivity to image geometric deformations. For example, small translations of a visual pattern may produce large changes in the transform coefficients. This effect poses serious limitations to applications like pattern recognition or matching that rely on constancy assumptions. An example is illustrated in Fig. 2.13, where the vertical details at level j = 2 can be compared for images that are translated one pixel in both directions. The original image is the same as in Fig. C.1. Notice that many pixels change value abruptly from one case to the other.

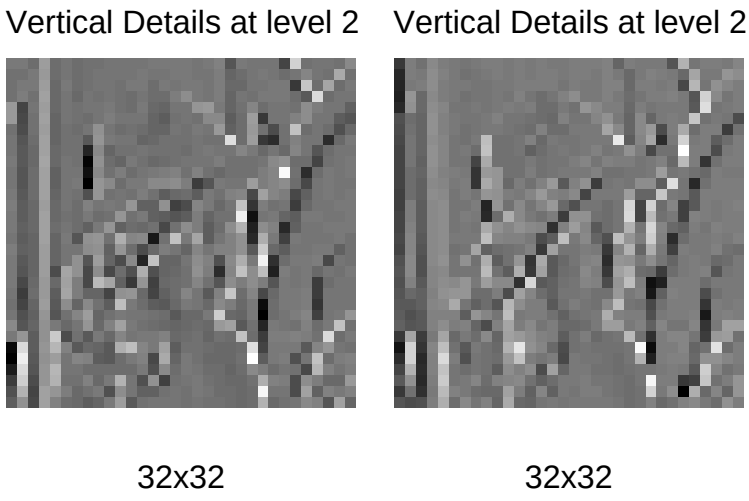


Figure 2.13: The detail coefficients at level 2 for slightly translated images.

If orthogonality is relaxed, it is possible to use smooth wavelets, that are less sensitive to geometric deformations. The cost to pay is to have some redundancy on the representation. The Laplacian Pyramid is an example of a redundant image representation. The redundancy factor is 2, because the pyramid has twice more coefficients than image pixels. It can be observed in Fig. 2.14 that the effect of translation is less drastic that in the DWT.

Other problem associated to non-redundant transforms is the lack of robustness to missing data and noise. If some coefficients of the transform are lost (e.g. in a communication process), redundancy on the representation may allow to estimate the values of the missing data [55]. Also, if noise is present in the coefficients, a redundant representation will reconstruct a less noisy signal [38].

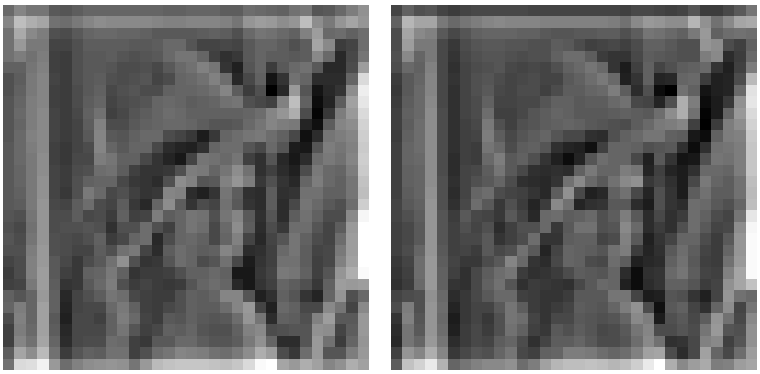


Figure 2.14: The Laplacian pyramid level 2 for slightly translated images.

2.2 Smooth Foveal Transforms

The major effort on the design of foveated systems has been centered around the topological (spatial) properties of the continuous mapping between the cartesian and foveated image spaces (e.g. the scale and rotation invariance in the logpolar transformation or the line feature preservation of the RWT). However, discretization requires certain cares to certify that foveal codes are proper representations of the original images. We have found that the majority of the works on foveal systems often disregards this aspect. Most applications use simple non-overlapping and uniform receptive-fields, that introduce aliasing effects in the foveal representation.

Aliasing (or frequency folding) is an effect of signal sampling. In a few terms, frequencies higher than a limit can not be represented by sampled signals. If those frequencies are not removed in the original signal, they will masquerade as lower frequencies and damage the signal representation. In signal processing applications, this effect should be avoided in order to be able to reconstruct the signal from its samples. In computer vision we are more interested in the stability and smoothness of the foveal code. Desirably, small changes in a visual pattern should not produce large changes in the foveal code, otherwise the representation is not suited for tasks like pattern recognition and matching.

In this section, we will motivate the problem and illustrate the distortions produced in foveated images by aliasing effects. Then we propose some guidelines for the distribution and profiles of receptive fields to reduce aliasing effects. Based on these guidelines, we design a foveated sensor with a logpolar distribution of Gaussian RF's and perform some experiments for comparison with other methods. To prepare the ground for it, we start with a frequency interpretation of the receptive field foveation process.

2.2.1 Frequency Interpretation of RF Foveation Methods

According to our formulation, the foveal code is obtained by the projection of the original image in a set of RF functions:

$$c_i =< f, \phi_i > \tag{2.23}$$

To compute the output of a particular receptive field we can either perform the operation directly as the previous equation suggests, or alternatively perform an equivalent two step operation:

1. **Filtering** - convolve the whole image with the RF weighting function:

$$f_f(x,y) = \sum_m \sum_n f(m,n)\phi(x-m, y-n) \tag{2.24}$$

2. **Subsampling** - pick the value at the RF location (x_i, y_i):

$$c_i = f_f(x_i, y_i) \tag{2.25}$$

In the first step, we consider the receptive field function as a filter that weights image frequency content by its Fourier transform. Thus, when sampling the filtered image we are taking a measure of local image frequency content within the passband of the receptive fields' frequency response.

Obviously, this procedure is rarely used in practical terms since often receptive fields have different shapes along the visual field and it would be a waste of processing power to filter the whole image with every different receptive field.

2.2.2 Aliasing

According to sampling theory, to avoid aliasing distortion the sampling rate (ω_s) should be higher than twice the maximum image frequency content ($\omega_{\max}$):

$$\omega_s > 2\omega_{\max} \tag{2.26}$$

This is also known as the *Nyquist criterion* and the value $2\omega_{\max}$ is known as the *Nyquist rate*. According to the frequency interpretation of the foveation procedure, the maximum image frequency content can be controlled by the RF shape. In a certain region, after filtering, image maximum frequency is less than or equal to the RF maximum frequency content. Therefore, the sampling distance between the receptive fields Δ must follow the law:

$$\Delta < \frac{\pi}{\omega_{\max}} \tag{2.27}$$

We have found that existing foveation systems rarely take this fact into account. In fact, computer vision tasks involving parameter estimation from large image regions, have not reported any problems with aliasing effects, because these operations involve a lot of redundant data and noise and distortions have a small influence in average. However, in applications requiring high reliability and stability of local measures, like pattern recognition, these distortions may negatively affect the performance of algorithms. Looking at the successful applications using foveal images, we find that most of them rely on the computation of a few motion parameters from full image data [157, 125, 14, 100, 16, 32, 17, 145, 1, 142, 138, 44], while very few have reported successful results on the detection and recognition of local patterns [54, 139].

To motivate the aliasing problem and realize its effects, we start with an example for one-dimensional signals.

1D example

Let us consider the very common uniform ($\pi(x)$) and Gaussian ($g(x)$) RF profiles, with unit area and similar spatial support:

$$\pi_\sigma(x) = \begin{cases} \frac{1}{6\sigma} \Leftarrow -3\sigma < x < 3\sigma \\ 0 \Leftarrow \text{otherwise} \end{cases}$$

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$$

where σ is a scale parameter. The corresponding Fourier transforms are:

$$G(\omega) = e^{-\sigma^2\omega^2/2}$$

$$\Pi(\omega) = \frac{\sin(\sigma\omega)}{\sigma\omega}$$

Both the RF profiles and corresponding Fourier transforms are shown in Fig. 2.15, for $\sigma = 20$. We can observe that the frequency amplitude of the smooth Gaussian profile

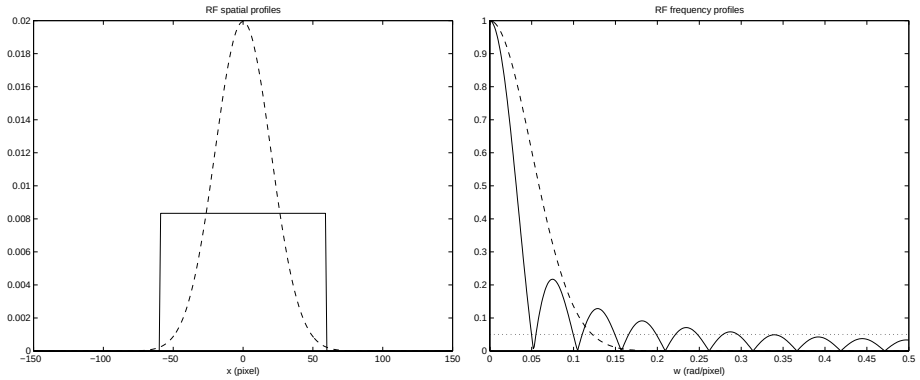


Figure 2.15: Spatial (left) and frequency (right) profiles of uniform (solid line) and Gaussian (dashed line) receptive fields. Dotted line in the frequency plot shows the 5% amplitude bandwidth.

stabilizes around zero much faster than the frequency response of the uniform receptive field. Considering a 5% tolerance, we can observe that the maximum frequencies are of about 0.12 rad/pixel for the Gaussian profile and 0.3 rad/pixel for the box like profile. Thus, according to sampling theory, spacing between receptive fields should be smaller than 26 pixel for Gaussian RF's and 10 pixel for the box-like RF's. In this case it is obvious the existence of overlap between neighbooring RF's.

Many existing foveation methods do not consider overlap between receptive fields. Commonly, this is due to computational considerations since the existence of overlap requires extra computations. What cost shall we pay by not following the *Nyquist criterion*? In general, problems arise when the input signal contains repetitive patterns of frequencies higher than the Nyquist rate. In these circumstances, high frequencies masquerade as low frequencies and generate "illusory" RF responses, as shown in Fig. 2.16. We can observe the output of Gaussian receptive fields with 10 pixel spacing but different amounts of overlap. With no overlap, there is some RF activity that disappears as RF overlap increases. This effect is more dramatic in the case of uniform RF's, where the "illusory" responses exist even for high overlap rates (Fig. 2.17). Obviously, this is a very artificial situation

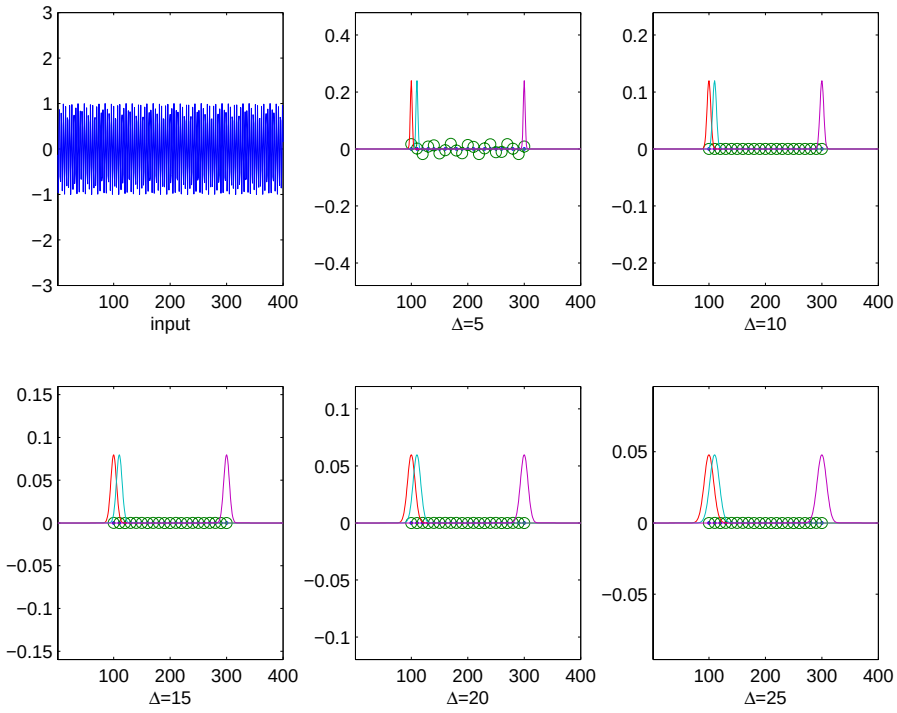


Figure 2.16: A high frequency signal (top-left) is analysed by several sets of Gaussian RF's located at the positions indicated by dots on the horizontal axes (10 pixels apart). Only the first two and the last receptive fields are shown explicitly. The spatial support of each RF is indicated by parameter Δ. The response of each receptive field is shown as a circle.

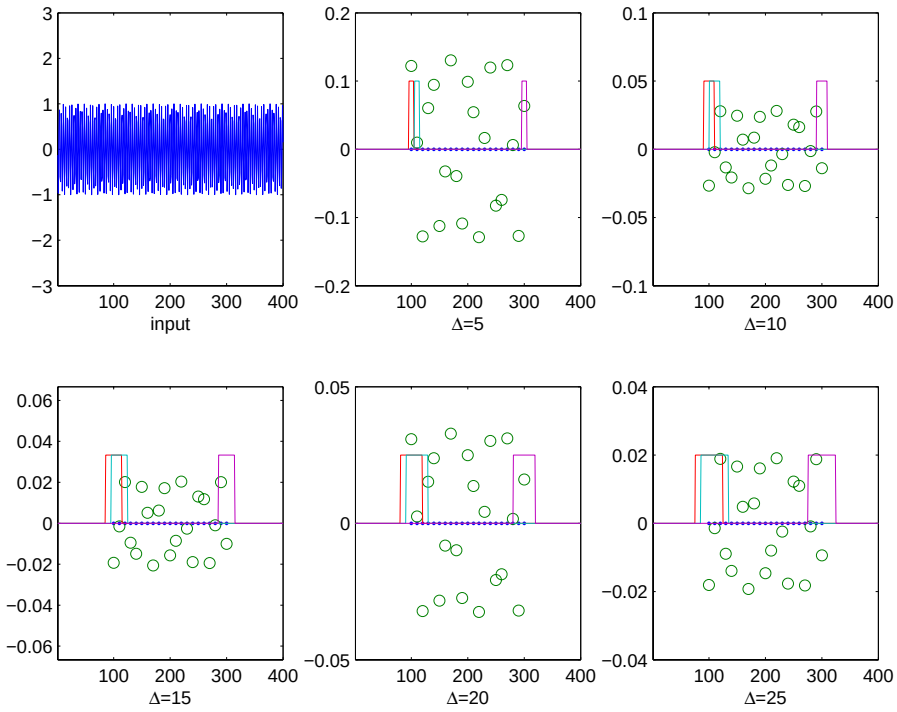


Figure 2.17: The same experiment of the previous Figure, with uniform RF's.

designed to introduce the problem. We will shown in next section that this effect also happens in more realistic situations.

2.2.3 The Smooth Logpolar Transform (SLT)

In this section we propose a rule for the location and width of log-polar distributed Gaussian RF's, in order to reduce the amount of aliasing distortion in the foveation process. The reasoning derives directly from the 1D case presented previously, to establish the size of RF's as a function of the distance to its neighbors. A log-polar distribution is chosen due to its wide use, desirable properties and biological support, already stated in the previous section. The RF Gaussian shape is chosen due to its smoothness both in space as in frequency.

Foveal Code Smoothness

Before going into the design phase, some criteria must be defined in order to evaluate the quality of the design. As previously stated, our main concern is centered on the stability of the foveal code, i.e. small coefficient change under small changes in the input representation.

A quantitative criterion could be formulated as a perturbation analysis method : evaluate the sensitivity of output coefficients as a function of input changes. However this is not easy to put into practice because input and output data are very high dimensional. Instead, we will have to rely on a qualitative criterion for the smoothness of the representation, namely visualizing the transform coefficients in a spatial arrangement preserving spatial neighborhood. This can be done both in the original domain (cartesian) or in the transform domain (logpolar) by generating images with values equal to the foveal code at the corresponding the RF locations. In the cartesian domain this results in very sparse images that are "filled in" using interpolation methods. In the logpolar domain, coefficients are dense almost everywhere because RFs form a dense uniform grid in transform coordinates. Exceptionally, some sparse regions exist, one very close to the fovea, where RFs corresponding to the same cartesian pixel are not represented, and the other in the far periphery due to the corners in the original cartesian images. To the cartesian and logpolar foveal code representations is also use to call the *Retinal* and *Cortical* images resp., due to the biological analogy.

An example is illustrated in Fig. 2.18. We show the original, retinal and cortical images corresponding to two types of receptive-fields. With highly overlapping Gaussian RF's, foveal coefficients relate smoothly with their neighbors, while with low overlapping uniform RFs it is possible to identify distortions or spurious values that degrade the low-pass information contained in the original image. The same information is shown also for a slightly translated original image. The smooth code is changed similarly to a translation of the same amount, while the non-smooth code presents large changes in the transform coefficients not modeled by a simple translation.

Having motivated the aliasing problem, we now proceed to sensor design. This amounts to define the distribution, size and shape of the receptive fields. The distribution will follow closely the logpolar model due to its already stated advantages. The shape of receptive fields is chosen as Gaussian due to its smoothness and localization both in space and in frequency. Their size will be defined to reduce aliasing distortion on the foveal code representation.

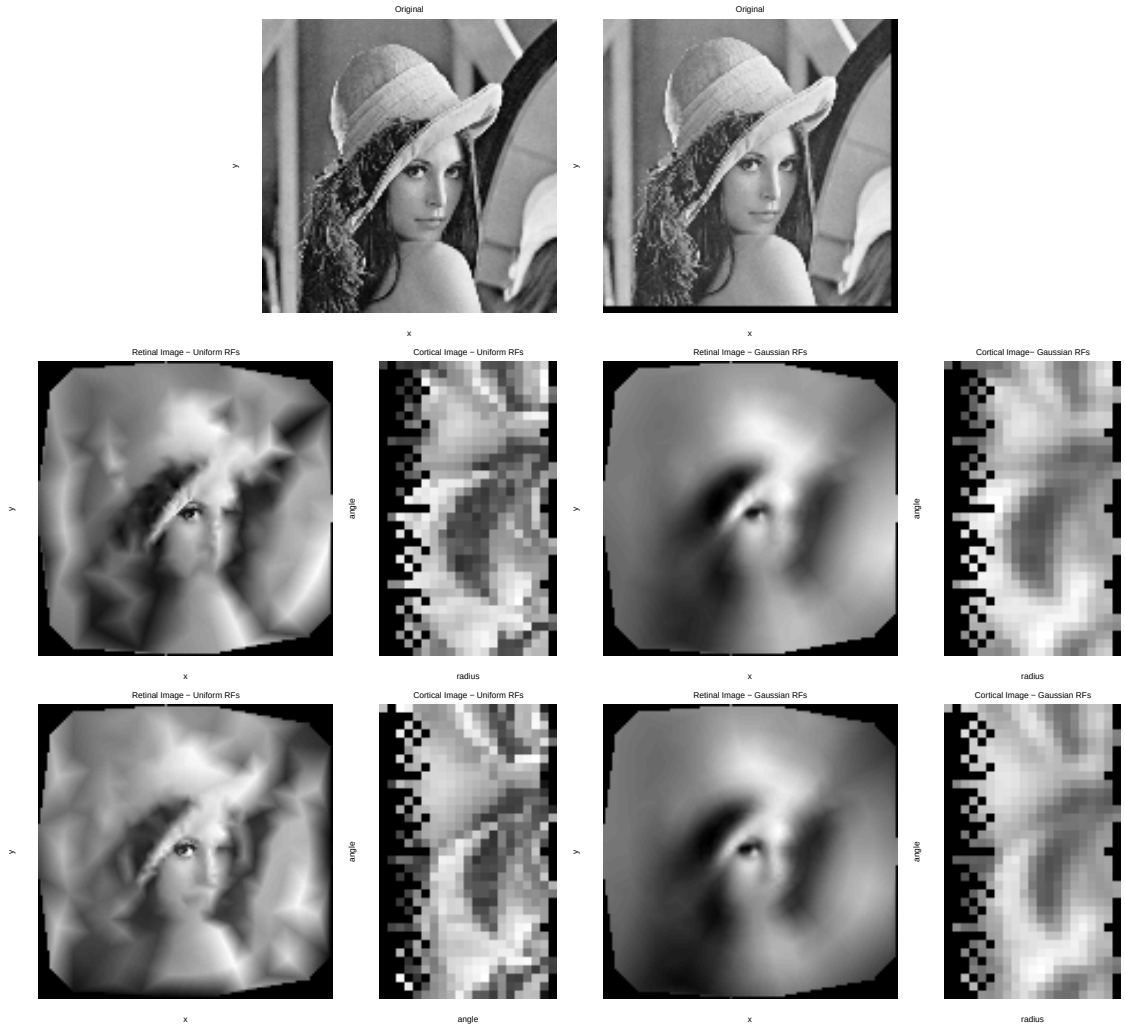


Figure 2.18: Foveal codes can be visualized both in the original domain, via image reconstruction by interpolation, or in the logpolar domain. The top row shows the input images. They are related by a translation of 3 pixels in the horizontal as vertical directions. The middle and bottom rows show the foveal codes. From left to right: retinal and cortical foveal codes acquired with Gaussian RFs (smooth codes), and retinal and cortical foveal codes obtained from uniform RFs (non-smooth codes). Notice that the smooth codes have similar appearances but the non-smooth code exhibits large changes in some regions (attend the differences in the nose, mouth and peripheral regions).

Smooth Radial Distribution

The definition of the radial distribution will take into account two design variables: the radius of the retina (R) and the radius of the fovea (f). We assume a uniform dense sampling inside the fovea (one RF per pixel) and a log-polar distribution outside the fovea. Let ρ_i be the radial coordinate of the i^{th} ring from center to periphery. The values of the ρ_i can be defines recursively by:

$$\rho_i = \begin{cases} \rho_{i-1} + 1, & i \leq f \\ k\rho_{i-1}, & i > f \end{cases} \tag{2.28}$$

where $\rho_0 = 0$ and k is a constant to define. To force a smooth transition between the uniform distribution inside the fovea and log-polar distribution outside the fovea we make the boundary RF's ρ_r and ρ_{r+1} to comply simultaneously with the uniform distribution in the fovea and the log-polar law outside:

$$\rho_{f+1} = \rho_f + 1 \tag{2.29}$$

$$\rho_{f+1} = k\rho_f \tag{2.30}$$

which results in:

$$k = \frac{f+1}{f} \tag{2.31}$$

A closed form expression for the ρ_i is given by:

$$\rho_i = \begin{cases} i, & i = 0, \cdots, f \\ fk^{i-f}, & i > f \end{cases} \tag{2.32}$$

whose graphical representation is shown if Fig. 2.19 for $f = 9$.

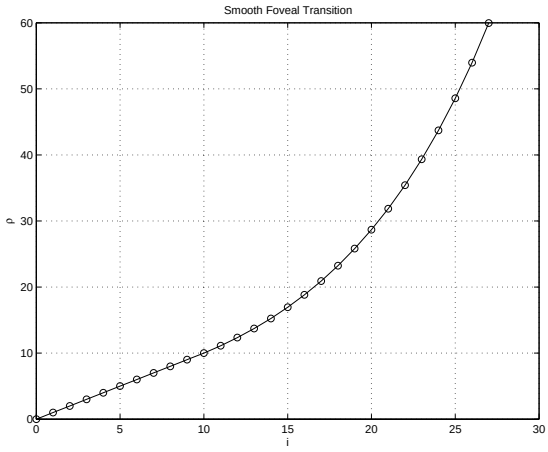


Figure 2.19: Radial distribution of RF's with smooth transition from a uniform resolution fovea to a logpolar distributed periphery. In this case, fovea radius is 9 pixel.

Balanced Angular Distribution

The angular distribution is defined to have similar distances between neighbor RF's, both in the radial and the angular directions. This definition only matters for the outer fovea

region. According to (2.32), the average radial distance between neighbor RF's at eccentricity i is given by:

$$\Delta_r = \frac{\rho_{i+1} - \rho_{i-1}}{2} = \rho_i \frac{k - k^{-1}}{2} \tag{2.33}$$

In the angular direction the distance between neighbor RF's at eccentricity i is given by:

$$\Delta_a = 2\rho_i \sin \frac{\Delta_\theta}{2} \tag{2.34}$$

where Δ_θ is the angular step. Forcing the two previous distances to be equal, we get:

$$\Delta_\theta = 2 \arcsin \frac{k - k^{-1}}{4} \tag{2.35}$$

This value is then corrected for an integer number of angles in the full circumference. The resulting distribution of RF's is called *balanced* and is shown in Fig. 2.20. In practical cases it may be useful to have the RF locations at integer values of pixels and remove any redundant RF's in the fovea (oversampling). Applying this correction to the previous case, we get a distribution also shown in Fig. 2.20.

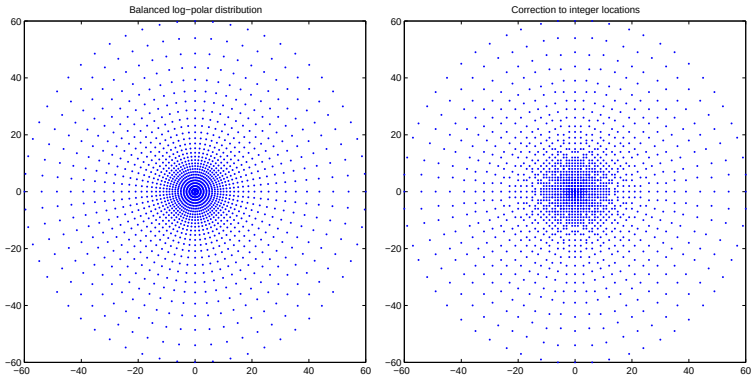


Figure 2.20: Distribution of RF's for a retina with 128 pixel diameter and a fovea with 9 pixel radius. RF's may be defined at sub-pixel locations (right) or at integer locations (right).

The plot in Fig. 2.21 shows the number of receptive fields as a function of fovea radius, with respect to full image size. The results were computed considering 128×128 images but the ratios shown are approximately independent of image size.

RF size

To complete the design, we will apply the Nyquist criterion to define the size of the Gaussian receptive fields, in order to attenuate possible aliasing effects. Here we assume smooth changes on distance between RF's, such that neighbor RF may have similar sizes. Also we consider that the spectral content of a Gaussian RF with variance σ^2 can be neglected for frequencies above $3/\sigma$ (see Fig. 2.15). Thus, according to Eq. (2.27), the relationship between RF spacing and RF size becomes:

$$\Delta < \frac{\sigma\pi}{3} \tag{2.36}$$

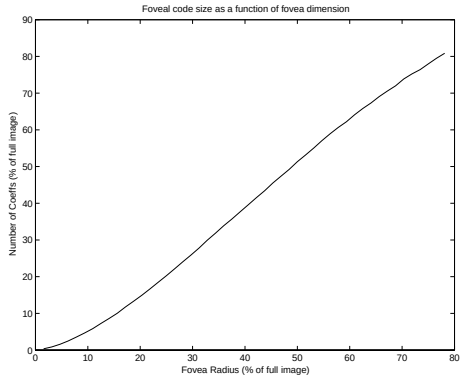


Figure 2.21: The ratio of receptive-fields to image pixels is shown as a function of fovea diameter to full image diameter.

or, rewriting to isolate RF size:

$$\sigma > \frac{3\Delta}{\pi} \tag{2.37}$$

Here Δ is the maximum spacing between the RF and its direct neighbors, which in the case of the balanced log-polar distribution, can be approximated by Eq. (2.33). Thus, in terms retinal radial position, RF size is constrained by:

$$\sigma(\rho) > \begin{cases} \frac{3}{\pi}, & \rho \leq f \\ 3\rho\frac{k-k^{-1}}{2\pi}, & \rho > f \end{cases} \tag{2.38}$$

In some situations, a better assignment for Δ can be obtained by explicitly computing the distance between each RF and its neighbors, according to some connectivity rule. This method is preferred if RF coordinates are corrected to integer pixel values, which changes slightly the balance of the distribution.

The RF overlap factor is computed by the fraction of the RF diameter overlapped by its neighbors and, for this design it is about 83% (Fig. 2.22). We approximate the diameter of a Gaussian RF by 6σ, since above this value, the RF profile function is close to zero.

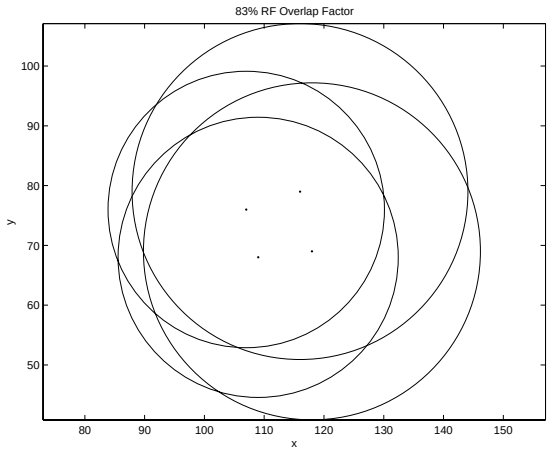


Figure 2.22: The RF overlap factor on the anti-aliased Gaussian design is about 83%.

Simulations

To illustrate the performance of the preceding design, we have made some simulations with three sensor designs, all with the same distribution of RF's but different profiles and overlap factors:

1. The anti-aliased Gaussian design, with an overlap factor of 83%.
2. A design with Gaussian RF's with 50% overlap factor.
3. A design with uniform RF's with 50% overlap factor.

All sensors were used to analyse a 128×128 pixel image. The fovea radius was defined as $f = 5$. The design produces a sensor with 581 RF's.

Image reconstructions with cubic interpolation are shown in Fig. 2.23. It is visible that sensors 2 and 3 exhibit spurious artifacts on the reconstructed images, which is typical of aliasing distortions.

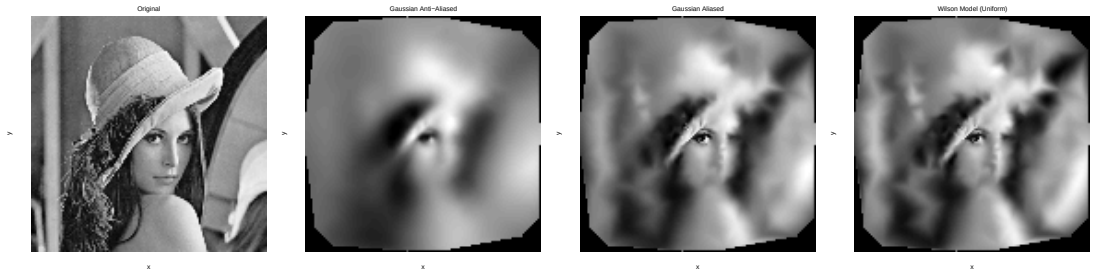


Figure 2.23: From left to right: original and reconstructed images from sensors with 83% overlapped Gaussian RF's, 50% overlapped Gaussian RF's and 50% overlapped uniform RF's.

2.3 The Fast Smooth Logpolar Transform (FSLT)

Most of the existing multiscale foveation methods are based on pyramid-like image decompositions. Transform coefficients of a certain level come from filtering and subsampling the previous level. Then, foveation is simulated by weighting each level of the decomposition with appropriate windows. However, subsampling in multiscale transformations usually conform to the cartesian space, which is restrictive in the definition of the sensor geometry. To overcome this limitation we propose an algorithm in two steps: first, foveation filtering is applied to the original image via a multiscale transformation without any subsampling involved; second, subsampling is performed according to a logpolar RF distribution.

[63] proposes a methodology for foveation filtering without subsampling. The purpose of their work is to simulate the vision of patients with various forms of visual field loss (due to eye disease), and can be used with completely arbitrary variable resolution geometries. Resolution is defined via a smooth function of space (the visual resolution map). The image is decomposed in several low-pass levels (a Gaussian decomposition) and each pixel in the foveated image is obtained by blending the pixels in the closest low-pass levels in terms of resolution. The weights are determined by the variable resolution map. They have developed a software that produces artifact free gaze contingent video at high frame rates in either 8-bit gray scale or 24-bit color. The low-pass levels are obtained by image

convolution with Gaussian kernels which is in close relationship with a linear scale-space decomposition of the images.

In our case, instead of a Gaussian decomposition, foveation filtering is obtained with a Laplacian Decomposition. This choice has two justifications. First, a Laplacian decomposition splits the frequency content of the original image into sub-bands, which will facilitate the definition of appropriate weighting windows to avoid aliasing effects. Second, the Laplacian decomposition models the output of ganglion cells in the human retina. Thus, we can get for free image features that represent image contrast. Appendix D introduces the unsampled Gaussian and Laplacian decompositions and presents a very efficient approximation to the Laplacian decomposition using the *à trous* algorithm.

2.3.1 The Foveation Filter

The foveation filter consists in simulating foveation without sampling by: i) weighting the several levels of the Laplacian decomposition with appropriate windows and, ii) reconstructing the foveated image. Since the representation is unsampled, this last step consists in the simple addition of all the levels in the decomposition (see Appendix D). The weighting windows for each level will be designed to minimize aliasing effects, assuming the balanced logpolar distribution of RF's as proposed in Sec. 2.2.3. From previous sections we have the following constrains:

- The *Nyquist criterion* imposes a constrain on the maximum sampling distance Δ between neighbor RF's as a function of maximum signal frequency ω_{max}, expressed in (2.27).
- The balanced logpolar distribution of RF's determines the sampling distance Δ as a function of eccentricity ρ, given by (2.33).
- The maximum frequency at each level of a dyadic Laplacian decomposition, $\omega_{max}(i)$, is given by (D.6).

Putting together the above constraints we derive an expression for the regions at each Laplacian level i where aliasing is avoided:

$$\begin{cases} \rho_{max}(0) < f \\ \rho_{max}(i) < \frac{2^i \pi}{3(k-k^{-1})} \end{cases} \tag{2.39}$$

where $k = (f+1)/f$ and f is the fovea radius. To avoid aliasing, in each level i we must annihilate the information contained in regions not following the above equation. Thus, the weighting windows at each level i should vanish where $\rho > \rho_{max}(i)$. Choosing Gaussian windows centered at the fovea, whose value is negligible for eccentricities greater than 3σ, we get the following window standard deviation for each Laplacian level:

$$\begin{cases} \sigma(0) < \frac{f}{3} \\ \sigma(i) < \frac{2^i \pi}{9(k-k^{-1})} \end{cases} \tag{2.40}$$

In practice we found that we can have slightly wider windows without having significant aliasing effects. We suggest to use Gaussian windows with the following standard deviations:

$$\begin{cases} \sigma(0) < \frac{f}{2} \\ \sigma(i) < \frac{2^i \pi}{6(k-k^{-1})} \end{cases} \tag{2.41}$$

Figure 2.24 shows weighting windows defined with the above rule for a fovea radius of 12 pixels, 128 × 128 images and 6 level Laplacian decomposition. The 90%, 50% and 10% level curves are shown superimposed to the distribution of the sampling points.

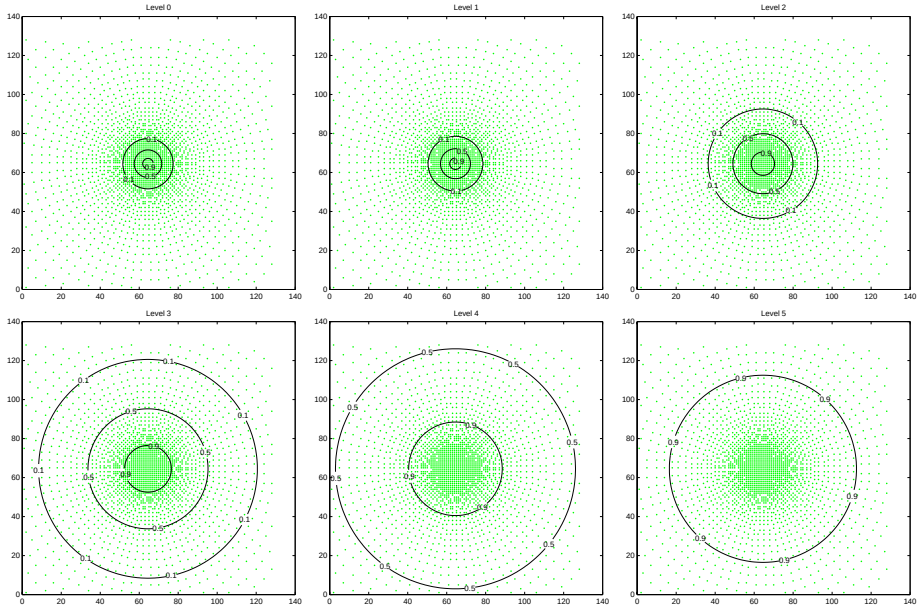


Figure 2.24: Level curves of the weighting windows for a 6 level decomposition of 128 × 128 images with a fovea radius of 12 pixels.

2.3.2 Sampling and Reconstruction

The proposed multiscale foveation algorithm is composed of the following operations:

1. Decomposition – Decompose the original image f into $S+1$ Laplacian levels l_j:

$$f(x,y) \rightarrow l_j(x,y), \quad j = 0 \cdots S \tag{2.42}$$

2. Weighting – Weight the decomposition levels with appropriate windows w_j:

$$l_j^w(x,y) = w_j(x,y) \cdot l_j(x,y) \tag{2.43}$$

3. Composition – Simulate the foveated image:

$$f_f(x,y) = \sum_{j=0}^{S} l_j^w(x,y) \tag{2.44}$$

4. Sampling – Sample the foveated image at the RF locations (x_i, y_i):

$$c(i) = f_f(x_i, y_i) \tag{2.45}$$

5. Visualization (optional) - Display the foveated image f_f.

All the steps are illustrated in the following set of figures. Fig. 2.25 shows the 6 level Laplacian decomposition of a 128 × 128 image. The set of weighting windows for each

level and the result of the weighting step are shown in Fig. 2.26. Finally, Fig. 2.27 shows the foveated image and the foveal code represented in retinal and cortical coordinates. Notice the good approximation between the foveated image and the retinal image, revealing the negligible amount of aliasing in the representation.

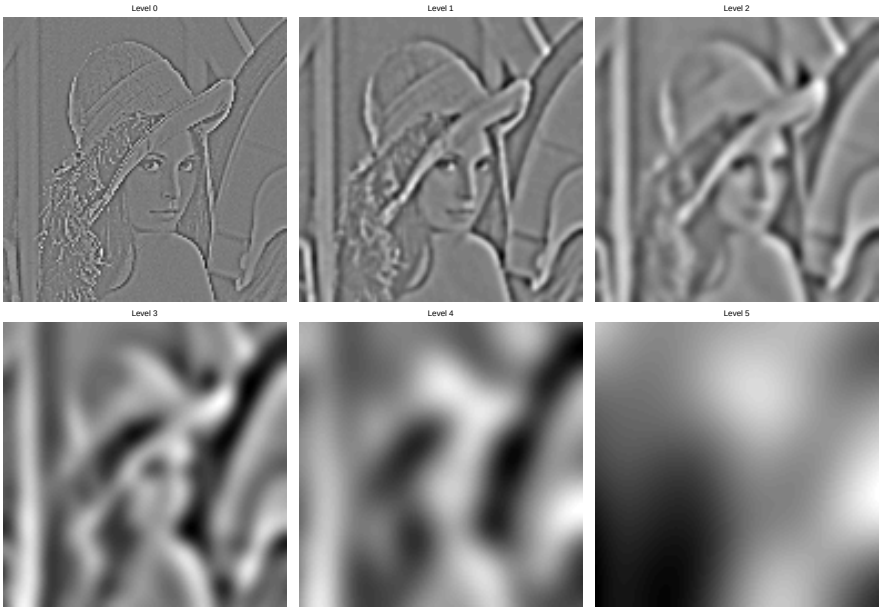


Figure 2.25: 6 level Laplacian decomposition of a 128 × 128 image.

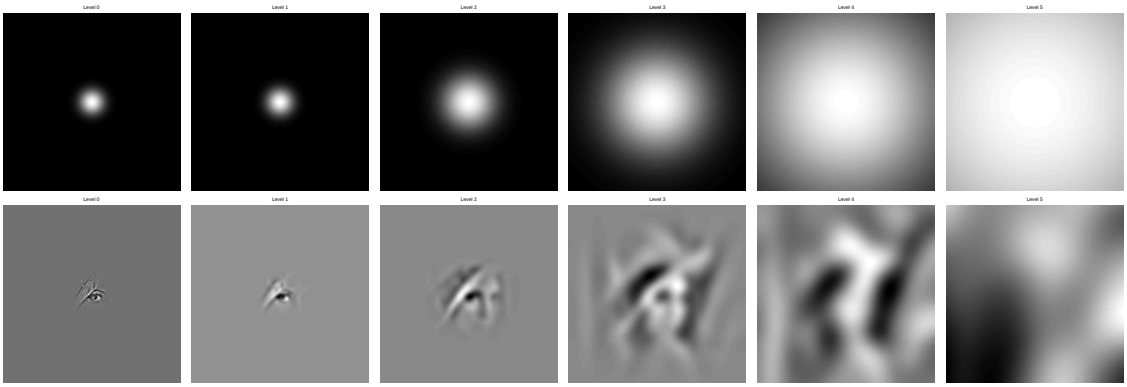


Figure 2.26: Weighting windows (top) and weighted Laplacian levels (bottom).

Because the composition step is a simple addition of all the weighted Laplacian levels, the above operations can be executed in a different order. For instance, it may be useful to have the foveal code separated in its different scales. In this case the last steps of the algorithm become:

3. Sampling – Sample each of the weighted Laplacian levels at the RF locations (x_i, y_i):

$$c_j(i) = l_j^w(x_i, y_i) \tag{2.46}$$

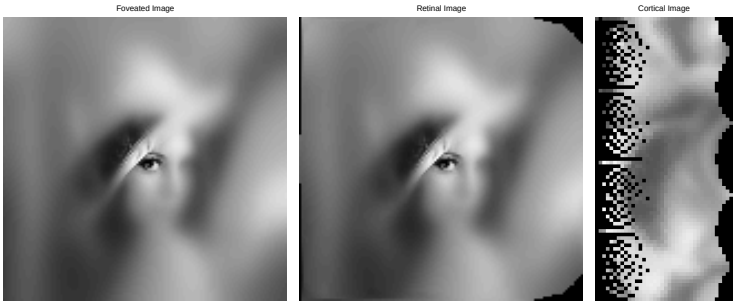


Figure 2.27: From left to right: The foveated image and corresponding reconstructed retinal and cortical codes.

4. Composition – Compose the foveal code:

$$c(i) = \sum_{j=0}^{S} c_j(i) \tag{2.47}$$

5. Visualization (optional) - Reconstruct the foveated image by interpolation of the foveal coefficients in the cartesian domain.

Fig. 2.28 shows the sampled coefficients at each level. In this alternate algorithm the foveated image is never explicitly computed and the composition step results on the cortical image shown before, in Fig. 2.27.

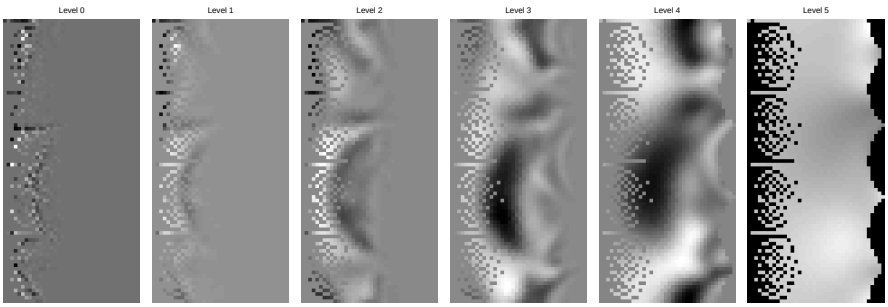


Figure 2.28: The foveal code can be stored in its composing sub-bands.

Another variation of interest is to have the coefficients unweighted and weight only as required, e.g. if different windowing functions must be used. In this case the weighting and sampling steps are reversed, in order, and the algorithm comes:

2. Sampling – Sample each of the unweighted Laplacian levels at the RF locations (x_i, y_i):

$$c_j^u(i) = l_j(x_i, y_i) \tag{2.48}$$

3. Weighting – Weight the sampled decomposition levels with the appropriate weights:

$$c_j(i) = c_j^u(i) \cdot w_j(x_i, y_i) \tag{2.49}$$

The unweighted foveal code levels can be observed in Fig. 2.29.

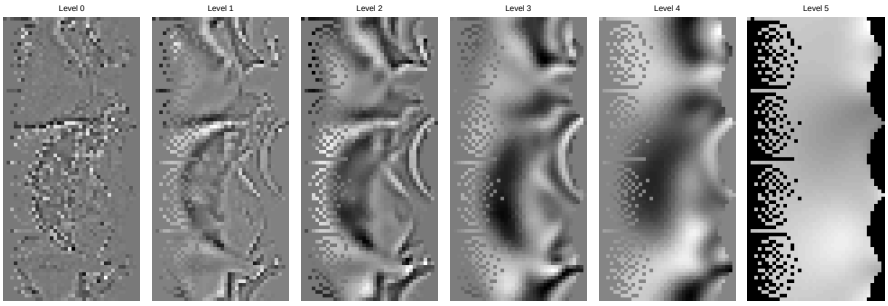


Figure 2.29: The 6 levels of foveal code, before the sampling step.

The previous variations of the basic algorithm show the versatility and flexibility of this approach. The particular order of operations may be chosen taken into account the particular use of the representation and the computational efficiency of each of its variations. For a general purpose application not requiring frequent visualization of the foveated image, we suggest the last form because it minimizes the number of multiplications required in the weighting step.

2.4 Final Remarks

This chapter was dedicated to the analysis of foveation techniques. An extensive literature review was presented, with special emphasis to log-polar based methods, which are the more frequent and biologically supported.

Foveation was formulated as the image projection on bases composed of spatially localized and finitely supported functions, the *receptive fields*. A frequency based interpretation of the process allowed to analyse sampling effects, often neglected in the literature, producing aliasing distortions. Such analysis led to a methodology to define appropriate RF's shape and size in order to reduce the aliasing effects on a logpolar-like distribution of receptive fields. A significant amount of receptive-field overlap is required to achieve so.

Direct implementation of foveation models with highly overlapping receptive fields require very intensive computations. Multiscale foveation methods are based on a different principle but allow efficient approximations to the foveation problem. We have reviewed existing algorithms but they usually consider subsampling on cartesian tessellations. We propose a new algorithm that executes logpolar sampling in a weighted Laplacian decomposition. The weighting windows were defined to reduce the amount of aliasing in the representation.

Chapter 3

Binocular Head Control

The objective of the work described in this thesis is the development of methodologies and algorithms for the visual based control of robotic binocular heads in arbitrary real environments. Many interacting aspects, from visual perception to motor control, are involved in this process. In this chapter we address the pure control related aspects. To this end we must assume simplifying assumptions, in particular that appropriate sensor inputs are computed by the perceptual parts of the system, and reference signals are properly defined by higher level decision modules. Latter chapters will refer to these issues.

Our approach to the control problem is based on the *Visual Servoing* framework [53, 73, 113, 78]. This theory integrates robot kinematics and image projection geometry to establish a general methodology for the design of visual based motion control schemes. It assumes that the pose of objects of interest is defined by visual features already computed. In this chapter we adopt the notation and methods provided by this framework to formulate our control strategies.

We present some original contributions related to the application of the visual servoing framework to active binocular systems, and the introduction of dynamics in direct image based controllers. Usually, visual servoing control strategies only consider the system kinematics and neglect dynamics [40]. For binocular head systems, with some linearization in equilibrium trajectories, it is possible to decouple the various degrees of freedom and "project" system dynamics to the image plane.

We start by formulating the binocular head control problem. In Section 3.1, we present the robotic systems that serve as test–beds for our experiments, the *Medusa* and *Blimunda* stereo heads, and model their kinematics and imaging aspects. Then, in Section 3.2, the *Visual Servoing* framework [78] is briefly described, providing the notation and basic theory to address our particular problem. We extend the analysis to consider moving objects and tracking systems. To this end, we analyse trajectory equilibrium conditions that lead to substantial simplifications in modeling and design. Then, in Section 3.3, we particularize this methodology to our binocular systems and derive suitable joint-independent control strategies. Additionally, we combine robot dynamics with the equilibrium conditions to develop dynamic controllers based directly in the image plane. Finally, in Section 3.5 we present some experiments on a real robotic system to illustrate the performance of our approach. The extraction of the appropriate visual features from real images will be the subject of following chapters.

3.1 Problem Formulation

Medusa and *Blimunda* are binocular systems with four degrees of freedom, designed to replicate the main motions performed by the human visual system. *Medusa* (see Fig. 3.1) was the first ISR-Lisboa[1] binocular head [129]. It was designed to perform accelerations

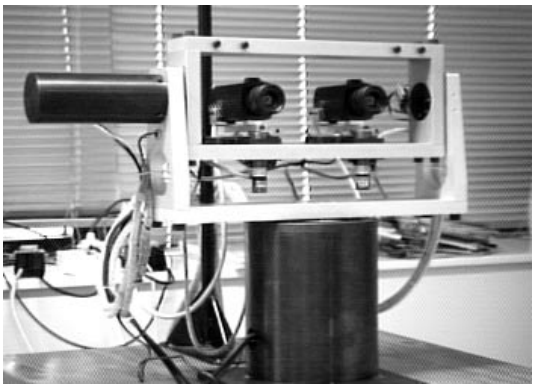


Figure 3.1: The *Medusa* binocular head.

matching those of the human eyes and neck, which require very powerful motors and heavy mechanical parts. A new binocular head, *Blimunda*, shown in Fig. 3.2, was acquired in the context of the Narval[2] project and weights a few kilograms, which allows its use on a mobile robot platform. Both heads have similar kinematics, with 4 degrees of freedom.

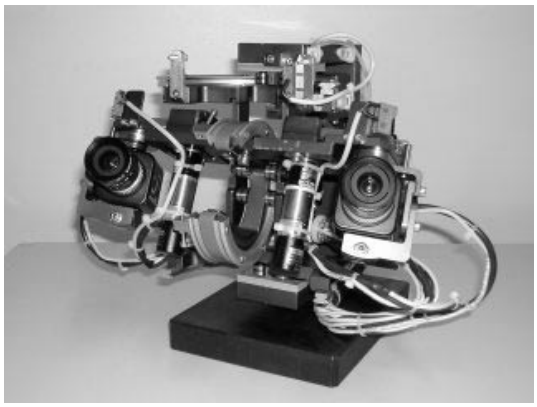


Figure 3.2: The *Blimunda* binocular head.

The revolution joints actuate cameras pan (2), neck pan (1) and neck tilt (1).

The head control problem has the general goal of detecting and tracking objects of interest in the visual field. After detecting an interesting object, the robot head should control its cameras in order to position and keep the target in the center of the images, canceling its relative motion. When prior information about target motion is not available, this strategy is optimal in minimizing the risk of target loss [152]. In the human visual system this is also the preferred strategy because eye resolution is higher in the center. Thus, we define the control goal as the minimization of object projection deviation from the center of the images. For this purpose we need to establish the relationships between 3D object positions and their projections in the image plane of both cameras.

[1] ISR-Lisboa is the Institute for Systems and Robotics at the Lisboa Technical University

[2] European research project ESPRIT-LTR 30185

3.1.1 Visual Kinematics

Fig. 3.3 represent the mechanics and geometry of our binocular heads, illustrating the four rotational degrees of freedom. Joint angles are denoted by θ_l, θ_r (left and right cameras), θ_p (pan) and θ_t (tilt). Cameras are mounted such that their optical axes always intersect. To the intersection point we denote *Fixation Point*. Objects at this location project exactly in the center of both images. Thus, to fixate and track an object, the fixation point must be controlled to coincide with object position. This will be attained indirectly by controlling the cameras in such a way that the object projection is kept in the center of the images. As explained latter, formulating the control problem in the image plane is less sensitive to calibration errors.

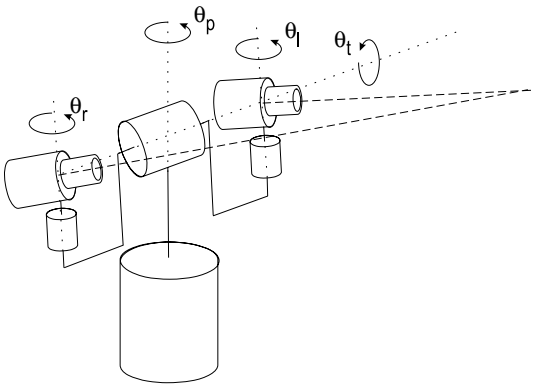


Figure 3.3: The mechanical structure of ISR-Lisboa binocular heads.

Because the robot head has 4 degrees of freedom, there are many possible joint angle configurations that correspond to the same fixation point. To remove the ambiguity and simplify the problem, we assume frontal vergence configurations, i.e. $\theta_v = \theta_r = -\theta_l$, as shown in Fig. 3.4. Since camera joint excursion is limited symmetrically, this strategy maximizes the range of possible vergence angles.

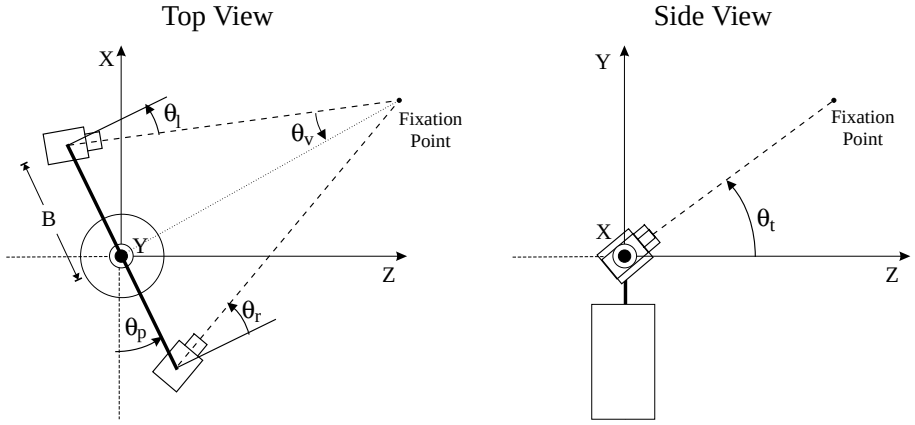


Figure 3.4: The robot configuration parameters $(\theta_v, \theta_p, \theta_t)$ and the world reference frame $\{X, Y, Z\}$. The top view shows a frontal vergence fixation $(\theta_v = \theta_r = -\theta_l)$.

In geometrical terms, cameras are relative position sensors. They measure the position of objects in the environment with respect to a camera-centered reference frame. Thus, to calculate coordinates of image projections we must first express the position of targets

in the coordinate frame of each camera. With this goal in mind, we have to analyse the following aspects:

1. How do the robot joint angles influence cameras position ?
2. What is the 3D object position with respect to both cameras ?
3. How do 3D positions project in the image plane of both cameras?

Robot Kinematics

Here we will express the pose of each camera as a function of robot joint angles. Consider an inertial frame, denoted *world frame*, as in Fig. 3.4. Cameras pose can be described by position and orientation with respect to the world frame. We describe positions and orientations by translation vectors $(\mathbf{t}_l, \mathbf{t}_r)$ and rotation matrices $(\mathbf{R}_l, \mathbf{R}_r)$, respectively. The subscripts l and r stand for *left* and *right* cameras. Both position and orientation are functions of the robot joint angles. *Robot configuration* is defined as a vector containing the vergence, tilt and pan angles:

$$\mathbf{q} = (\theta_v,\ \theta_p,\ \theta_t)' \tag{3.1}$$

Simple geometrical computations lead to the formulas that express cameras position and orientation as functions of the robot configuration.

$$\mathbf{R}_l = \begin{bmatrix} c_p c_v + s_p c_t s_v & s_t s_v & -s_p c_v + c_p c_t s_v \\ -s_p s_t & c_t & -c_p s_t \\ -c_p s_v + s_p c_t c_v & s_t c_v & s_p s_v + c_p c_t c_v \end{bmatrix}; \quad \mathbf{t}_l = \begin{bmatrix} -c_v B \\ 0 \\ s_v B \end{bmatrix} \tag{3.2}$$

$$\mathbf{R}_r = \begin{bmatrix} c_p c_v - s_p c_t s_v & -s_t s_v & -s_p c_v - c_p c_t s_v \\ -s_p s_t & c_t & -c_p s_t \\ c_p s_v + s_p c_t c_v & s_t c_v & -s_p s_v + c_p c_t c_v \end{bmatrix}; \quad \mathbf{t}_r = \begin{bmatrix} c_v B \\ 0 \\ s_v B \end{bmatrix} \tag{3.3}$$

In the expressions above, c_x and s_x are abbreviations for $\cos\theta_x$ and $\sin\theta_x$ respectively, and B represents half the camera baseline (fixed distance between the left and right cameras).

Relative Pose

In this chapter we consider that objects and their image projections can be described by points, i.e. 3D and 2D vectors, respectively. These points may correspond to the object geometrical center or to the positions of any other particular salient feature of interest. It will be the purpose of vision algorithms, to be presented in next chapters, to appropriately measure object position.

Let the 3D target position be represented by cartesian coordinates $\mathbf{P} = (X, Y, Z)'$ in the world reference frame. Then, target relative position with respect to the left and right cameras is denoted by vectors $\mathbf{P}_l$ and $\mathbf{P}_r$, respectively, that can be computed by:

$$\mathbf{P}_l = \mathbf{R}_l\mathbf{P} + \mathbf{t}_l \quad , \quad \mathbf{P}_r = \mathbf{R}_r\mathbf{P} + \mathbf{t}_r \tag{3.4}$$

where the rotations matrices $\mathbf{R}_l$, $\mathbf{R}_r$, and the translation vectors, $\mathbf{t}_l$, $\mathbf{t}_r$, are the ones given in (3.2) and (3.3).

Instead of parameterizing the target position directly with (X, Y, Z) values, we use spherical coordinates. Given the rotational nature of the robot head systems, a spherical parameter representation of the target will simplify the kinematics expressions. Thus, we

define the *target configuration* as a vector containing distance ρ, elevation angle ϕ and azimuth angle γ with respect to the world frame.

$$\mathbf{p} = (\rho,\ \phi,\ \gamma)' \tag{3.5}$$

Since we assume symmetric vergence configurations, elevation and azimuth angles are directly related to head tilt and pan, and distance is related non-linearly to the vergence angle (see Fig. 3.5). With this parameterization, the absolute target position is given by:

$$\mathbf{P} = \begin{bmatrix} \rho\cos\gamma\sin\phi \\ -\rho\sin\gamma \\ \rho\cos\gamma\cos\phi \end{bmatrix} \tag{3.6}$$

Using this expression in (3.4), we get the target position relative to both cameras:

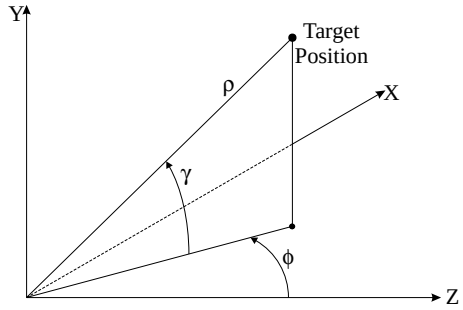


Figure 3.5: Target configuration parameters: distance (ρ), azimuth (ϕ) and elevation (γ)

$$\mathbf{P}_l = \begin{bmatrix} \rho\,\mathrm{c}_\gamma\,\mathrm{c}_v\,\mathrm{s}_{\phi-p} + \rho\,\mathrm{c}_\gamma\,\mathrm{c}_t\,s_v\,\mathrm{c}_{\phi-p} + \rho\,\mathrm{s}_\gamma\,\mathrm{s}_t\,\mathrm{s}_v - \mathrm{c}_v\,B \\ -\rho\,\mathrm{c}_\gamma\,\mathrm{s}_t\,\mathrm{c}_{\phi-p} + \rho\,\mathrm{s}_\gamma\,\mathrm{c}_t \\ -\rho\,\mathrm{c}_\gamma\,\mathrm{s}_v\,\mathrm{s}_{\phi-p} + \rho\,\mathrm{c}_\gamma\,\mathrm{c}_t\,\mathrm{c}_v\,\mathrm{c}_{\phi-p} + \rho\,\mathrm{s}_\gamma\,\mathrm{s}_t\,\mathrm{c}_v + \mathrm{s}_v\,B \end{bmatrix} \tag{3.7}$$

$$\mathbf{P}_r = \begin{bmatrix} \rho\,\mathrm{c}_\gamma\,\mathrm{c}_v\,\mathrm{s}_{\phi-p} - \rho\,\mathrm{c}_\gamma\,\mathrm{c}_t\,\mathrm{s}_v\,\mathrm{c}_{\phi-p} - \rho\,\mathrm{s}_\gamma\,\mathrm{s}_t\,\mathrm{s}_v + \mathrm{c}_v\,B \\ -\rho\,\mathrm{c}_\gamma\,\mathrm{s}_t\,\mathrm{c}_{\phi-p} + \rho\,\mathrm{s}_\gamma\,\mathrm{c}_t \\ \rho\,\mathrm{c}_\gamma\,\mathrm{s}_v\,\mathrm{s}_{\phi-p} + \rho\,\mathrm{c}_\gamma\,\mathrm{c}_t\,\mathrm{c}_v\,\mathrm{c}_{\phi-p} + \rho\,\mathrm{s}_\gamma\,\mathrm{s}_t\mathrm{c}_v + \mathrm{s}_v\,B \end{bmatrix} \tag{3.8}$$

Image projection

A projection in the image plane is defined as a 2D point, $\mathbf{p}_c = (x_c, y_c)'$, whose coordinates depend on the relative target position, $\mathbf{P}_c = (X_c, Y_c, Z_c)'$, and the particular image projection model for that camera, Π:

$$\mathbf{p}_c = \Pi\left(\mathbf{P}_c\right) \tag{3.9}$$

Conventional cameras, like the ones we use, can be well described by the *pinhole model*. An illustration of the pinhole model and the chosen reference directions is shown in Fig. 3.6. For simplicity, and without loss of generality, we use the normalized pinhole model, where the influence of the intrinsic parameters is removed. Taking into account the chosen reference directions, the coordinates of target projection in the image plane are:

$$\mathbf{p}_c = \Pi(\mathbf{P}_c) = (-X_c/Z_c,\ Y_c/Z_c)' \tag{3.10}$$

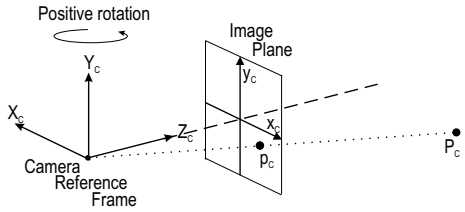


Figure 3.6: The camera pinhole model and chosen reference frames.

Replacing index c by l or r, and using the vectors $\mathbf{P}_l$ and $\mathbf{P}_r$ in (3.7) and (3.8), we can compute the target projection in both cameras as follows:

$$\left\{ \mathbf{p}_l = \Pi(\mathbf{P}_l)\mathbf{p}_r = \Pi(\mathbf{P}_r) \right. \tag{3.11}$$

3.2 The Visual Servoing Framework

In this section we present the visual servoing approach in its general formulation, and then particularize it for tracking problems. We will introduce the *equilibrium conditions* that model the control problem under small deviations from the desired trajectories. In the subsequent section we apply these concepts to our binocular head systems.

The visual servoing problem [53] is defined as the minimization of a task function depending on object's pose $\underline{P}$ (position and attitude of points, lines, surfaces) and cameras pose $\underline{Q}$. Often, pose information is represented in 6D cartesian coordinates (position and orientation), by means of a translation vector $\mathbf{t} \in \Re^3$ and an orientation matrix $\mathbf{R} \in SO^3$, both relative to a world coordinate frame $\mathcal{W}$.

For the sake of generality, we assume that each pose is parameterized by generalized coordinate vectors $\underline{p} \in \mathcal{C}_p \subseteq \Re^m$ and $\underline{q} \in \mathcal{C}_q \subseteq \Re^n$, where $\mathcal{C}_p$ and $\mathcal{C}_q$ are the target and camera configuration spaces. We denote target pose and camera pose by $\underline{P} = \mathcal{P}\left(\underline{p}\right)$ and $\underline{Q} = \mathcal{Q}\left(\underline{q}\right)$, respectively. For instance, $\underline{p}$ can contain spherical coordinates to represent position, and roll–pitch–yaw angles to represent orientation. Usually, the components of $\underline{q}$ directly represent the system's degrees of freedom (joint angles or displacements) and define uniquely the cartesian pose of the cameras. The task function (or error function) can be defined either in cartesian, configuration or feature spaces (image coordinates). The last choice is simpler and less sensitive to system calibration errors [78]. In this case, the visual servoing problem is stated as the minimization of:

$$e = \left\| \underline{f} - \underline{f}^0 \right\| \tag{3.12}$$

where $\underline{f}$ is the image feature parameter vector (e.g. position of points, orientation of lines, etc.) and $\underline{f}^0$ is the desired vector. The feature parameter vector is given by:

$$\underline{f} = \mathcal{F}\left(\mathcal{P}_c\left(\underline{q}, \underline{p}\right)\right) \tag{3.13}$$

where $\underline{P}_c = \mathcal{P}_c\left(\underline{p}, \underline{q}\right)$ is the relative pose between objects and cameras, and $\mathcal{F}$ is the projection mapping due to the imaging system.

3.2.1 Feature Sensitivity

Usually, in image-based visual servoing, the control strategy is computed through the linearization of (3.13) on the operating point:

$$\delta \underline{f} = \mathbf{J}_q \left(\underline{q}, \underline{p} \right) \cdot \delta \underline{q} + \mathbf{J}_p \left(\underline{q}, \underline{p} \right) \cdot \delta \underline{p} \tag{3.14}$$

where $\mathbf{J}_q$ and $\mathbf{J}_p$ are the *image jacobians* or *feature sensitivity matrices*:

$$\begin{cases} \mathbf{J}_q \left(\underline{q}, \underline{p} \right) = \frac{\partial \mathcal{F}}{\partial \mathcal{P}_c} \left(\mathcal{P}_c \left(\underline{q}, \underline{p} \right) \right) \cdot \frac{\partial \mathcal{P}_c}{\partial \underline{q}} \left(\underline{q}, \underline{p} \right) \\ \mathbf{J}_p \left(\underline{q}, \underline{p} \right) = \frac{\partial \mathcal{F}}{\partial \mathcal{P}_c} \left(\mathcal{P}_c \left(\underline{q}, \underline{p} \right) \right) \cdot \frac{\partial \mathcal{P}_c}{\partial \underline{p}} \left(\underline{q}, \underline{p} \right) \end{cases} \tag{3.15}$$

When the target configuration parameters and their variations are known (or can be estimated), a suitable kinematic control solution to compensate for the feature motion expressed by (3.14) is:

$$\delta \underline{q} = -\mathbf{J}_q^{-1} \cdot \delta \underline{f} + \mathbf{J}_q^{-1} \mathbf{J}_p \cdot \delta \underline{p} \tag{3.16}$$

For non-square jacobians, least-squares solutions can be obtained using appropriate generalized inverses [78]. Despite their simplicity, solutions based solely on kinematics are not adequate for high-performance applications [40]. In our case, robot and target dynamics will be considered.

3.2.2 Pose Estimation

The computation of the jacobians $\mathbf{J}_q$ and $\mathbf{J}_p$, requires that the current configuration parameters, $\underline{q}$ and $\underline{p}$ are known. Even though $\underline{q}$ can be obtained from the robot's internal encoders, the object pose $\underline{p}$ can only be estimated using $\underline{q}$ and the image features $\underline{f}$. In the visual servoing literature this process is called *Forward Visual Kinematics* or *Target Location Determination*, and has the same nature of the *Pose Estimation* or *3D reconstruction* problems, well known in the computer vision literature:

$$\underline{p} = \mathcal{K} \left(\underline{q}, \underline{f} \right) \tag{3.17}$$

From a single camera one can only obtain 2D information about the position of points in 3D (depth is undetermined). To obtain 3D information, we must use more cameras or have some *a priori* knowledge of the object structure [37]. Forward visual kinematics usually require a precise calibration of the robot-camera system and is very sensitive to modeling errors [73]. We will show that, in tracking systems this computation can be avoided.

3.2.3 The Equilibrium Conditions

During target tracking, we may assume that the system operates close to the desired configurations, i.e. $\underline{f} \approx \underline{f}^0$. This condition enforces a kinematics constraint (or virtual linkage [53]) between camera and target poses. Using (3.17), this constraint can be expressed in the configuration space as:

$$\underline{p}^0 = \mathcal{K} \left(\underline{q}, \underline{f}^0 \right) \tag{3.18}$$

We define the *equilibrium manifold*, $\mathcal{E}$, as the set of all parameters $\left(\underline{q}, \underline{p}\right)$ where tracking is perfect:

$$\mathcal{E} = \left\{ \left(\underline{q}, \mathcal{K}\left(\underline{q}, \underline{f}^0\right)\right) ; \forall \underline{q} \in \mathcal{C}_q \right\} \tag{3.19}$$

If these conditions hold, in (3.16) we can approximate the general jacobians, $\mathbf{J}_q$ and $\mathbf{J}_p$, by the jacobians in equilibrium, $\mathbf{J}_q^0 = \mathbf{J}_q\left(\underline{q}, \underline{p}^0\right)$ and $\mathbf{J}_p^0 = \mathbf{J}_p\left(\underline{q}, \underline{p}^0\right)$:

$$\delta \underline{q} \approx -\left[\mathbf{J}_q^0\right]^{-1} \cdot \delta \underline{f} + \left[\mathbf{J}_q^0\right]^{-1} \mathbf{J}_p^0 \cdot \delta \underline{p} \tag{3.20}$$

In the next section we show that these approximations lead to a substantial simplification of the control system.

3.3 Binocular Visual Servoing

Here we apply the visual servoing formulation to our particular system. Based on the robot kinematics and the geometry of image formation, we analyse the system equilibrium conditions. For small deviations from equilibrium it is possible to approximate the full model by a decoupled system. This allows the development of independent image based dynamic controllers.

We start by adapting the results on robot kinematics in Section 3.1 to the visual servoing notation of Section 3.2.

The generalized coordinate vectors parameterizing head and target pose can be clearly identified with the **q** and **p** vectors in (3.1) and (3.5). We have, thus:

$$\begin{cases} \underline{q} = (\theta_v,\ \theta_t,\ \theta_p)' \\ \underline{p} = (\rho,\ \phi,\ \gamma)' \end{cases} \tag{3.21}$$

To define the image feature vector we notice that object projection on both cameras provides a 4-tuple of information - horizontal and vertical image coordinates in two images. Because we only control 3 angles (vergence, pan ant tilt), a sufficient and more suitable representation is provided by a 3 DOF feature vector containing object disparity and average horizontal and vertical image positions. Thus we define the image feature vector as:

$$\underline{f} = (d,\ x,\ y) = \left(\frac{x_l - x_r}{2},\ \frac{x_l + x_r}{2},\ \frac{y_l + y_r}{2} \right) \tag{3.22}$$

For the geometry of our system, controlling this vector to zero is equivalent to control the target projection to the center of both images. We will see later that, selecting these particular features is an effective strategy to decouple the image jacobians in the equilibrium manifold.

Because of the binocular nature of our system, we have to adopt extended descriptions for other vectors and matrices involved in the formulation. Extended vectors and matrices are obtained by concatenating the descriptions from both cameras. For example, the relative target pose, $\underline{P}_c$, is now defined as:

$$\underline{P}_c = \begin{bmatrix} \mathbf{P}_l \\ \mathbf{P}_r \end{bmatrix} \tag{3.23}$$

Using (3.7) and (3.8), it becomes:

$$\underline{P}_c(\mathbf{q},\mathbf{p}) = \begin{bmatrix} X_l \\ Y_l \\ Z_l \\ X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} \rho\, \mathrm{c}_\gamma\, \mathrm{c}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, s_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{s}_v - \mathrm{c}_v\, B \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_t\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{c}_t \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{c}_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{c}_v + \mathrm{s}_v\, B \\ \rho\, \mathrm{c}_\gamma\, \mathrm{c}_v\, \mathrm{s}_{\phi-p} - \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{s}_v\, \mathrm{c}_{\phi-p} - \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{s}_v + \mathrm{c}_v\, B \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_t\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{c}_t \\ \rho\, \mathrm{c}_\gamma\, \mathrm{s}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{c}_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t \mathrm{c}_v + \mathrm{s}_v\, B \end{bmatrix} \tag{3.24}$$

The image projection function $\mathcal{F}$ now maps a 6 coordinate relative pose vector to a 3 coordinate feature vector. For the perspective image projection of (3.10), we obtain:

$$\mathcal{F}\left(\underline{P}_c(\mathbf{q},\mathbf{p})\right) = \begin{bmatrix} -\frac{X_l}{2Z_l} + \frac{X_r}{2Z_r} \\ -\frac{X_l}{2Z_l} - \frac{X_r}{2Z_r} \\ \frac{Y_l}{2Z_l} + \frac{Y_r}{2Z_r} \end{bmatrix} \tag{3.25}$$

To obtain the feature sensitivity matrices, we have to compute the partial derivatives in (3.15). They are derived in appendix A and, in the general case, have a rather complex structure. However, if we restrict our analysis to the equilibrium manifold $\mathcal{E}$ (3.19), their structure becomes very simple.

3.3.1 Fixation Kinematics

The goal of binocular tracking is to keep the target projection in the center of both images. In this case, the desired feature vector is $\underline{f}^0 = \underline{0}$. The choice of robot and target configuration parameters shown in Figs. 3.4 and 3.5, provide a very simple equilibrium condition for the binocular heads. Pan and tilt angles, (θ_p, θ_t) in Fig. 3.4, must be equal to target elevation and azimuth, (γ, ϕ) in Fig. 3.5. The fixation distance, (ρ) in Fig. 3.5, is obtained by simple trigonometric calculations. Thus, the equilibrium condition can be written as:

$$\underline{p}^0 = (\rho, \phi, \gamma) = (B \cot\theta_v, \theta_p, \theta_t)' \tag{3.26}$$

and relative target position at fixation is given by:

$$\underline{P}_c^0 = (0,\ 0,\ B/\mathrm{s}_v,\ 0,\ 0,\ B/\mathrm{s}_v)' \tag{3.27}$$

Substituting the above conditions in the general jacobian matrices (see Appendix A), we obtain the sensitivity matrices at fixation:

$$\mathbf{J}_q^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & \mathrm{c}_t\, \mathrm{c}_v^2 & 0 \\ 0 & 0 & -\mathrm{c}_v \end{bmatrix} \quad \mathbf{J}_p^0 = \begin{bmatrix} -\mathrm{s}_v^2/B & 0 & 0 \\ 0 & -\mathrm{c}_t\, \mathrm{c}_v^2 & 0 \\ 0 & 0 & \mathrm{c}_v \end{bmatrix} \tag{3.28}$$

In our case, vergence and tilt angles are always well below 90^o (see Fig. 3.4). Therefore $\mathbf{J}_q^0$ is always invertible and well conditioned. Additionally, the above sensitivity matrices are **diagonal** and decouple the different motion parameters. We can use separate controllers for the kinematic visual servoing of each joint, which is a major advantage in terms of mathematical simplification and computation time. In the next section we will see that the equilibrium conditions also simplify dynamic analysis.

3.4 Dynamic Control

Kinematics controllers have limited performance because they lack knowledge about actuator dynamics and robot inertial structure. Here we will consider robot dynamical aspects and include the effect of target motion in the control system. We will show that the equilibrium conditions simplify the analysis of system dynamics, allowing its representations directly in terms of image features.

In the absence of disturbances and friction, a general dynamic model for robots can be written as [41]:

$$\mathbf{M}\left(\underline{q}\right)\underline{\ddot{q}} + \underline{h}\left(\underline{q}, \underline{\dot{q}}\right) = \underline{\tau}$$

where $\mathbf{M}$ is the inertia matrix, $\underline{h}$ comprises centrifugal, coriolis and gravitational terms, and τ is the vector of applied torques. By mechanical design, gravitational, coriolis and centrifugal terms are negligible in the head dynamics and inertia terms are decoupled for each joint. Furthermore, we use high geared joints and velocity control for the motors, which is effective in linearizing the axis dynamics and eliminating much of the load variation [40]. In these circumstances, joint dynamics are well approximated by a second order model. Considering all joints together, we have:

$$\underline{\ddot{q}} = -\mathbf{\Lambda}^{-1} \cdot \underline{\dot{q}} + \mathbf{\Lambda}^{-1} \cdot \underline{u} \tag{3.29}$$

with

$$\underline{u} = \begin{bmatrix} u_v \\ u_p \\ u_t \end{bmatrix} \quad \text{and} \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_v & 0 & 0 \\ 0 & \lambda_p & 0 \\ 0 & 0 & \lambda_t \end{bmatrix}$$

where λ_j and u_j are the time constant and the velocity command for each joint j.

For image based visual servoing this model is of little use, as the dynamic model is not directly expressed in terms of the image feature vector. We consider the case of $dim\left(\underline{f}\right) = dim\left(\underline{q}\right) = n$, which includes our particular problem. Expressing (3.14) using time derivatives, we have:

$$\underline{\dot{f}} = \mathbf{J}_q\,\underline{\dot{q}} + \mathbf{J}_p\,\underline{\dot{p}}$$

Provided that $\mathbf{J}_q$ is invertible, we can write:

$$\underline{\dot{q}} = \mathbf{J}_q^{-1}\,\underline{\dot{f}} - \mathbf{J}_q^{-1}\mathbf{J}_p\,\underline{\dot{p}} \tag{3.30}$$

Time differentiating again, we obtain:

$$\underline{\ddot{f}} = \mathbf{J}_q\,\underline{\ddot{q}} + \mathbf{J}_p\,\underline{\ddot{p}} + \underline{\delta}$$

where $\underline{\delta}$ represents higher order terms that will be considered as external disturbances. Now, substituting (3.29) and (3.30) in the last expression, results:

$$\underline{\ddot{f}} = -\mathbf{J}_q\,\mathbf{\Lambda}^{-1}\,\mathbf{J}_q^{-1}\,\underline{\dot{f}} + \ + \mathbf{J}_q\,\mathbf{\Lambda}^{-1}\,\mathbf{J}_q^{-1}\,\mathbf{J}_p\,\underline{\dot{p}} + \mathbf{J}_q\,\mathbf{\Lambda}^{-1}\,\underline{u} + \mathbf{J}_p\,\underline{\ddot{p}} + \underline{\delta} \tag{3.31}$$

In general, this dynamic model is highly coupled and time-variant, because the image jacobians depend on the particular head configuration at a given time instant. However, in tracking systems, we can use the equilibrium condition, and rewrite it in a decoupled and time invariant fashion.

3.4.1 Fixation Dynamics

Specifying (3.31) for the fixation subspace, the jacobian matrix $\mathbf{J}_q^0$ becomes diagonal and can commute with $\mathbf{\Lambda}^{-1}$:

$$\ddot{\underline{f}} = -\mathbf{\Lambda}^{-1}\dot{\underline{f}} + \mathbf{J}_q^0\,\mathbf{\Lambda}^{-1}\underline{u} + \mathbf{\Lambda}^{-1}\mathbf{J}_p^0\,\dot{\underline{p}} + \mathbf{J}_p^0\,\ddot{\underline{p}} + \underline{\delta}$$

Defining a new control input $\underline{r} = \mathbf{J}_q^0\,\underline{u}$, and image target motion $\underline{w} = \mathbf{J}_p^0\,\dot{\underline{p}}$, both represented in local image plane coordinates, we have:

$$\ddot{\underline{f}} = -\mathbf{\Lambda}^{-1}\dot{\underline{f}} + \mathbf{\Lambda}^{-1}\underline{r} + \mathbf{\Lambda}^{-1}\underline{w} + \mathbf{J}_p^0\,\ddot{\underline{p}} + \underline{\delta} \tag{3.32}$$

This is a second order linear approximation of the dynamics, expressed in the image plane, that allows for the design of dynamic visual controllers directly from image measurements. Additionally, since all the matrices are diagonal, we can use separate controllers for each joint. Therefore, analysing the system at equilibrium, we can easily design a variety of closed–loop controllers, whereas using the general model (3.31) it would be difficult and cumbersome. In the next subsection, we develop a dynamic controller with target motion prediction, as a way to compensate for the unknown term $\underline{w}$ in (3.32).

3.4.2 Independent Joint Control and Motion Estimation

Considering smooth trajectories for the target, we can assume a local constant velocity model (small acceleration) and disregard the term $\ddot{\underline{p}}$ in (3.32). Additionally, with decoupled dynamics, we can define separate controllers for each joint:

$$\ddot{f}_i = -\frac{1}{\lambda_i}\,\dot{f}_i + \frac{1}{\lambda_i}\,r_i + \frac{1}{\lambda_i}\,w_i$$

Defining the state vector as $\underline{x}_i = \left(f_i,\ \dot{f}_i\right)$, a state-space linear representation can be written as:

$$\begin{cases} \dot{\underline{x}}_i = \mathbf{F}\cdot\underline{x}_i + \mathbf{G}\cdot r_i + \mathbf{G}\cdot w_i \\ \underline{y}_i = \mathbf{H}\cdot\underline{x}_i + \underline{n}_i \end{cases}$$

where $\underline{y}_i$ are the observations and $\underline{n}_i$ the observation noise vector. Also we have:

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\lambda_i} \end{bmatrix};\quad \mathbf{G} = \begin{bmatrix} 0 \\ \frac{1}{\lambda_i} \end{bmatrix};\quad \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

From this state-space model we can compute the regulator gains, $\underline{K}$, using standard regulator design techniques, and use the control law:

$$r_i = -\underline{K}\cdot\underline{x}_i$$

We may consider two distinct motion control situation in our system. The first one corresponds to what is known in biology as saccade motion. This is a ballistic motion of the eyes aiming at shifting the gaze direction toward a different object in the shortest possible time. In many circumstances the system is not able to predict where events on the visual field will trigger saccade motions[3]. The second motion correspond to the biological smooth-pursuit motion, where the system tracks a target moving in the scene.

[3]exceptions are, for example, reading tasks, where the locus of of sequential saccades is correlated in space.

In a general sense we can consider that such type of motions have some regularity and continuity (enforced by physical inertial laws) and can be predicted in a short term basis.

The motion control system should have a dual mode of operation, optimizing the time responses and tracking errors for both types of motions. In the first case, we can model saccades as the system response to step like inputs. Hence, the performance of the control system can be evaluated by simulating step like inputs and checking common performance indicators in the response signal, like settling time, overshoot and steady state error. For smooth pursuit motions, we will assume that target velocity is approximately constant between two consecutive sampling periods and design a predictor to estimate this velocity and a controller to minimize the tracking error. This strategy can provide good tracking performance to several types of motions, provided that motion is smooth along its trajectory.

For high sampling rates, an appropriate model for target motion is:

$$\dot{w}_i = 0$$

To estimate target motion, we augment the system model with the motion model, yielding:

$$\begin{bmatrix} \dot{\underline{x}}_i \\ \dot{w}_i \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{0} & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_i \\ w_i \end{bmatrix} + \begin{bmatrix} \mathbf{G} \\ \underline{0} \end{bmatrix} \cdot r_i$$

$$\underline{y}_i = \begin{bmatrix} \mathbf{H} & \underline{0} \end{bmatrix} \begin{bmatrix} \underline{x}_i \\ w_i \end{bmatrix} + \underline{n}_i$$

With this model, we are able to estimate (reconstruct) the state consisting of $\underline{x}_i$ and w_i, using a state-space structure [64]:

$$\begin{cases} \dot{\widehat{\underline{x}}}_i = \mathbf{F} \cdot \widehat{\underline{x}}_i + \mathbf{G} \cdot \underline{r}_i + \underline{L}_x \cdot \left(\underline{y}_i - \mathbf{H} \cdot \widehat{\underline{x}}_i \right) \\ \dot{\widehat{w}}_i = L_w \cdot \left(\underline{y}_i - \mathbf{H} \cdot \widehat{\underline{x}}_i \right) \end{cases}$$

The computation of the estimator gains, $\underline{L}_x$ and L_w can be done with standard state estimation techniques, where the system model is augmented by the target motion model. However notice that the previous model is not used for regulator design, which is obtained using the **F** and **G** matrices of the unaugmented system. Our plan is to use the estimated value of w_i in a feedforward control scheme to reduce errors. Fig. 3.7 presents a block diagram of the full control system.

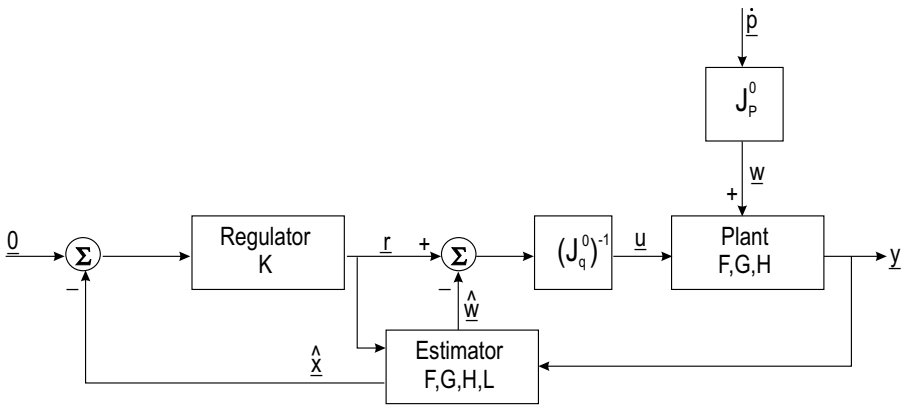


Figure 3.7: Block diagram of the dynamic control system.

To summarize this section, we have designed a control architecture for binocular heads using linear models and direct image plane measurements. The architecture permits the use independent dynamic controllers for each joint of the binocular head using, as inputs, the position and velocity of visual features in the image plane – no intermediate (full or partial) 3D reconstruction is required. In the remaining of this chapter we will present some results illustrating the performance of the proposed architecture, considering point-wise objects.

3.5 Performance Evaluation

Here we present experimental results that validate the applicability and advantages of our approach. The experiments compare the proposed controllers to others not having kinematics or dynamics compensation. We simulate 3D target trajectories to perform repeatable tests on the kinematic and dynamic closed loop control of the binocular head. *The real robot head is used inside the control loop.*

3.5.1 Kinematic Compensation

In equilibrium, image jacobians have a very simple structure, and can be easily used in the control law. To illustrate the effect of including the inverse jacobian in the visual control loop, we consider the following kinematic control strategies:

$$\delta \underline{q} = -\mathbf{K} \left[\mathbf{J}_q^0\right]^{-1} \cdot \delta \underline{f} \qquad versus \qquad \delta \underline{q} = -\mathbf{K} \cdot \delta \underline{f}$$

where $\mathbf{K}$ is a constant gain matrix, adjusted to obtain a good transient response to step inputs. Recall that the jacobian $\mathbf{J}_q^0$ has a strong effect in pan joint, reducing the loop gain when tilt and vergence angles are large:

$$\delta\theta_p = \cos(\theta_t) \cdot \cos(\theta_v)^2 \cdot \delta x$$

The results shown in Fig. 3.8 illustrate two cases: (1) a distant target moves in a step-like manner between the 3D points $(1, 0, 10)$ and $(-1, 0, 10)$, affecting only the x image coordinate and the pan angle; (2) a nearby target moves between points $(0.05, 0.4, 0.5)$ and $(-0.05, 0.4, 0.5)$, corresponding to large tilt and vergence angles. For the distant target both controllers behave similarly, since the jacobian gain is almost unitary. For targets close to the observer, kinematics compensation is advantageous and produces a faster response.

3.5.2 Dynamic Controller

In Section 3.4 we used the equilibrium conditions to derive an image–based decoupled time–invariant model that was extended to incorporate estimation of the target motion. Here we show that the dynamic controller developed in Section 3.4.2 is particularly suited to track slowly changing target motions. For comparison purposes we use the proportional controller with kinematics compensation as defined in the previous experiment. The adopted model parameters are $\lambda = 0.05$ s and the sampling period is $T = 0.02$ s. The regulator and estimator gains, K and L, were obtained from LQR and LQE design [64].

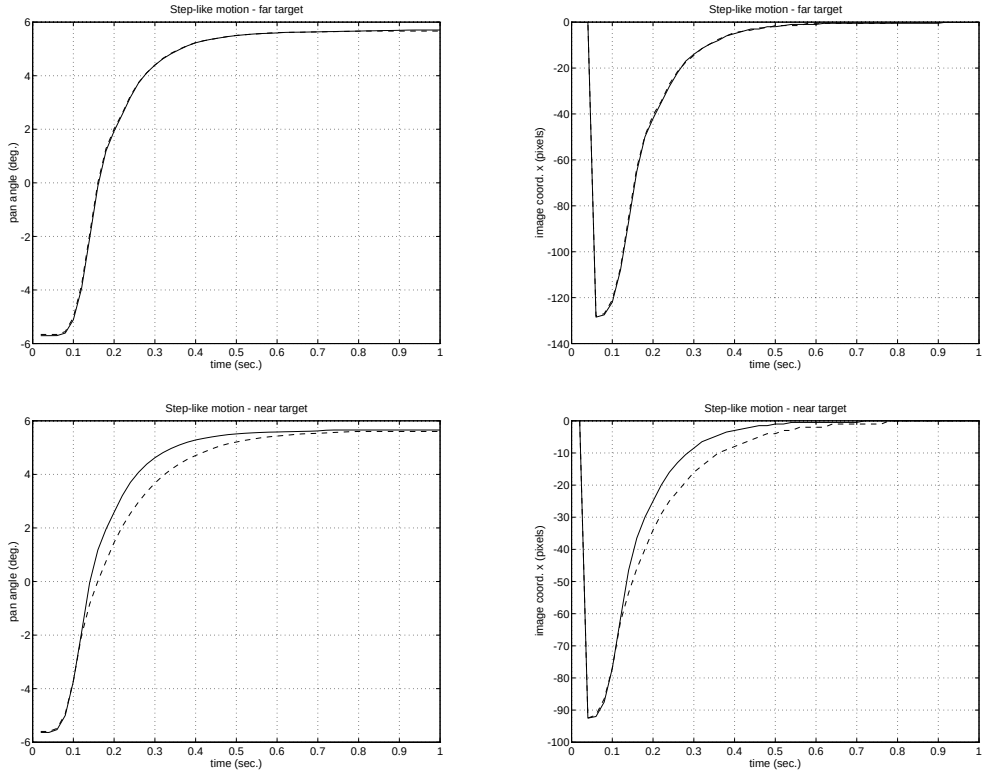


Figure 3.8: Evolution of the pan angle and x image feature to a step-like input. Top row: distant target. Bottom row: nearby target. Solid line: with kinematics compensation. Dashed line: without kinematics compensation.

Constant velocity

Fig. 3.9 shows the evolution of the pan and tilt angles and the (x, y) image coordinates, for a piecewise constant velocity trajectory between the 3D points $(1, -1, 6)$ and $(-1, 1, 6)$. The error in image features is smaller, on average, for the model based controller. Although during the target velocity discontinuities the error exhibits a short spike, during constant velocity periods it is much smaller than for the proportional controller. Hence, for the control of saccade motions it is preferable to use the simpler proportional controller, whereas for smooth pursuit motions, the model based controller will, on average, reduce the tracking error.

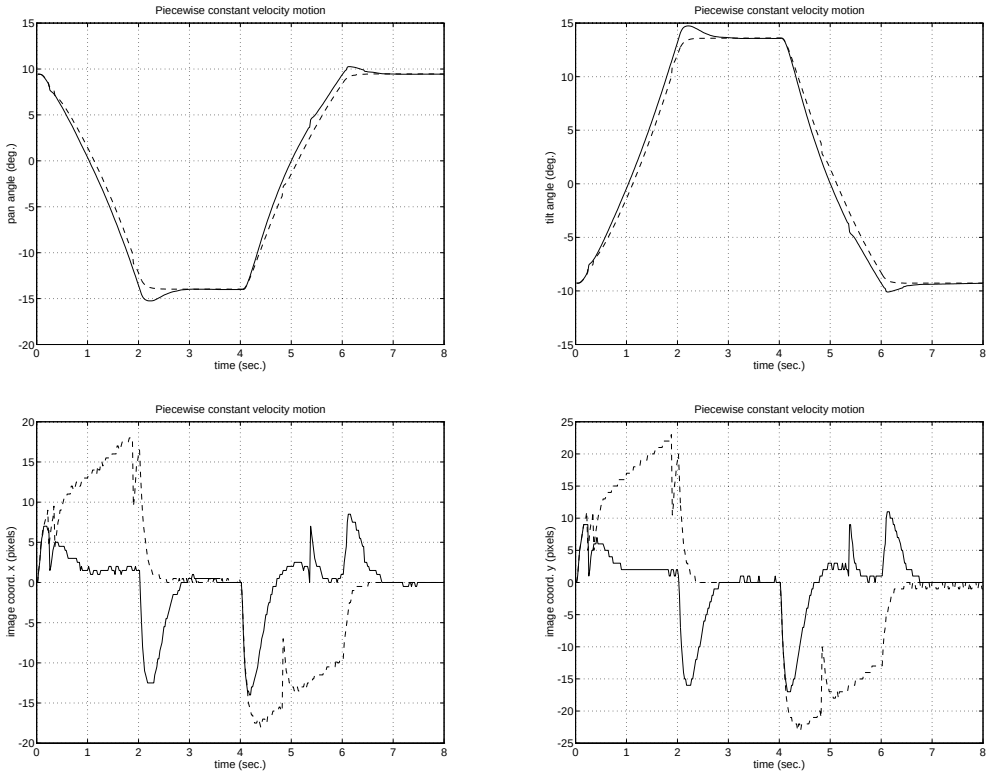


Figure 3.9: Evolution of the pan/tilt angles and x/y image features to a piecewise constant velocity input. Solid line: dynamic controller. Dashed line: proportional controller.

Slow elliptical motion

The target describes an elliptical trajectory with an angular velocity of 1 rad/s. Fig. 3.10 shows the error in the x image coordinate for each controller and also its estimated motion. In this situation, due to the absence of sudden changes in the target motion, the error for the model based controller is always much smaller than for the proportional controller.

Fast motions

Fig. 3.11 shows a 15 rad/s elliptical and step-like trajectories. These situations represent significant changes of target velocity and position discontinuities. The elliptical case represents a motion with a higher frequency than the bandwidth of the system and both

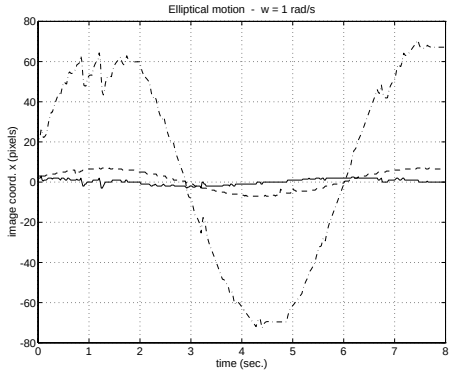


Figure 3.10: Evolution of the x feature in response to an elliptical target motion. Solid line: dynamic controller. Dashed line: proportional controller. Dash-dot line: estimated target motion.

controllers are unable to track the target. For step-like trajectories, the model based controller performs poorly because the slowly changing velocity model is no longer adequate at motion discontinuities.

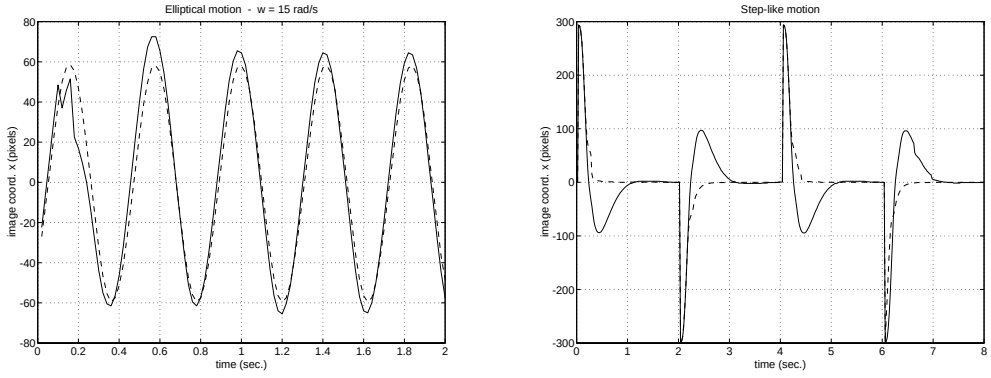


Figure 3.11: Evolution of the x feature in response to fast changing motions. Solid line: dynamic controller. Dashed line: proportional controller.

Chapter 4

Depth Perception

One of the most important capabilities of binocular systems is "depth perception". In passive photometric systems[1], monocular features like motion, focus, and shading have been used to address the problem. Binocular systems have the capability to compute depth via stereo, a conceptually simple and easy methodology.

Stereo involves the computation of disparity as a means for depth estimation. Disparity is the difference between the coordinates of object projections in the two images and can be computed by searching corresponding points. Depth is then computed through its nonlinear geometrical relationship to disparity.

In this chapter we describe methods and algorithms to estimate disparity using foveal images. In the first section we describe the stereo problem in verging systems and establish the relationships between depth, disparity and fixation distance. We briefly review the disparity estimation problem both for depth computation and vergence control. In the second section we present the proposed method for dense disparity estimation in foveal images. We adopt a Bayesian approach to the estimation problem and develop real-time algorithms to compute disparity maps. This information is then used for object segmentation and ocular vergence control. We propose a methodology to overcome local estimation ambiguities, using fast low-pass filters to propagate disparity information over neighboring regions in intermediate computational steps. This approach naturally favors smooth disparity surfaces but still allows the representation of depth discontinuities. Last section presents some results of the proposed disparity estimation algorithms in realistic setups.

4.1 Disparity and Stereo

Stereo depth perception is based on the existence of disparities between the projections of 3D points in both retinas. In the conventional parallel stereo setup[2], disparity only has horizontal components, that are inversely proportional to depth and directly proportional to camera distance (baseline). In the verging camera case[3], the relationship between disparity and depth is a bit more complex. With non null vergence angles, disparity has both horizontal and vertical components. Horizontal components of target projections alone are sufficient to estimate depth and in the following analysis we consider only coor-

[1] passive photometric systems do not emit any kind of energy to help the data acquisition process, as opposed to active photometric systems like laser and vision with artificial illumination.

[2] composed by two cameras with parallel optical axes mounted on a plane.

[3] the optical axes of the two cameras intersect in a 3D point called *fixation point*.

dinates in the plane of vergence (see Fig. 4.1). Once these coordinates are estimated, it is straightforward to compute the vertical components.

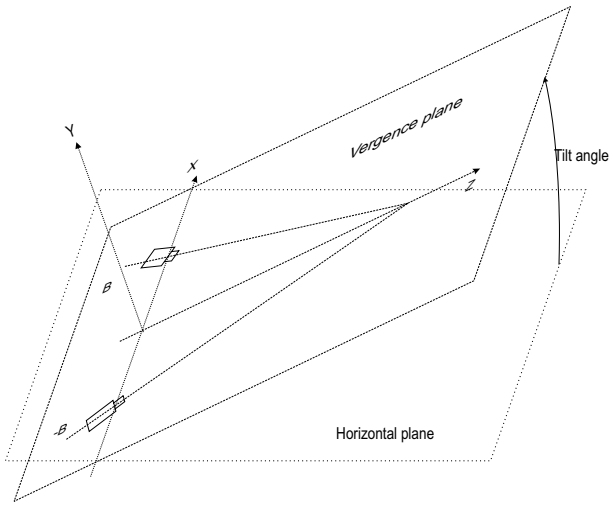


Figure 4.1: The vergence plane coordinate system is used for depth estimation.

The geometry of verging stereo in the vergence plane is represented in Fig. 4.2. The

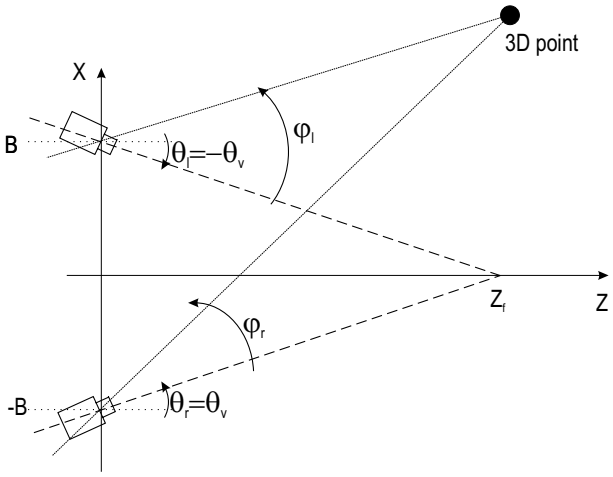


Figure 4.2: The geometry of vergent stereo.

angles ϕ_l and ϕ_r represent the horizontal angles of projection in the left and right cameras respectively, originating horizontal image coordinates $x_l = f{*}\tan(\psi_l)$ and $x_r = f{*}\tan(\psi_r)$, where f is the focal distance. Let us define horizontal disparity d and average horizontal position $\bar{x}$ as:

$$d = \frac{x_l - x_r}{2}, \quad \bar{x} = \frac{x_l + x_r}{2} \tag{4.1}$$

The 3D point coordinates in the vergence plane can be computed by:

$$X = -B\frac{2f\bar{x}}{(f^2 + \bar{x}^2 - d^2)\sin(2\theta_v) + 2fd\cos(2\theta_v)} \tag{4.2}$$

$$Z = Z_f\frac{\sin(2\theta_v)\left((f - \tan(\theta_v)d)^2 - \tan(\theta_v)^2x^2\right)}{(f^2 + \bar{x}^2 - d^2)\sin(2\theta_v) + 2fd\cos(2\theta_v)} \tag{4.3}$$

where B is half the camera distance and Z_f is the fixation distance. Similar formulas to recover depth in verging systems are presented in [109, 91]. In [109], depth is also represented as: i) the product of average horizontal projection and disparity; ii) fixation distance and; iii) a scale factor dependent of the vergence angle. It was noticed that the scale factor is much more stable for depths near the fixation point, where its value is close to unity. Thus, for an active binocular agent to acquire stable and reliable object shape representations, a reasonable strategy would be: i) to fixate at a distance close to the object and; ii) represent depth relative to fixation distance using the scale factor or the disparity itself. Then, if needed, relative depth can be transformed to absolute depth using the head kinematics expressions. In the remaining of the chapter we focus on the disparity estimation problem for object segmentation and vergence control. In none of these applications explicit depth information is necessary, and disparity, viewed as "qualitative" depth, is sufficient for our needs.

4.1.1 Disparity Estimation

The key issue for depth perception in stereo systems is the estimation of disparity from the world information projected in the cameras. Given the coordinates of P corresponding points in the left and right images $\left\{(x_l^i, y_l^i)\right\}_{i=1\cdots P} \rightarrow \left\{(x_r^i, y_r^i)\right\}_{i=1\cdots P}$, we compute the horizontal disparity by:

$$d^i = \frac{x_l^i - x_r^i}{2} \tag{4.4}$$

Several methods have been proposed to measure disparity in stereo images. For a recent review see [130]. Disparity estimation methods can be classified into *sparse* and *dense*. *Sparse* methods estimate disparity in a few points on the images. Usually, points like corners and edges are chosen because of their stability to changes in illumination and view angle. *Dense* methods compute disparity at every point in the image. In this case it is easier to represent disparity as a map, $d(x, y)$, containing the corresponding disparity value for each observed point in one image. For instance, considering a left dominant eye configuration, the disparity map is represented as $d(x_l, y_l)$, and a dense depth map can be computed by using the expressions (4.2) and (4.3), with $\bar{x} = x_l - d(x_l, y_l)$. The same analysis is valid for right dominant eye configurations.

Sometimes, there is no need to represent disparity at each point. For example, in binocular systems, vergence control can be performed by minimizing global (or dominant) horizontal disparity. Dominant horizontal disparity is the amount of translation that maximizes the correlation between the two images. For this purpose, frequency domain approaches like *cepstral* filtering [161, 39] and phase correlation [59, 141, 74] are the preferred methods because of the shift invariance property of Fourier techniques. However, in cases where the object of interest is small, the disparity of the background dominates and induces non desirable results. An alternative to enforce the importance of objects centered in the field of view is the use of foveal images [14, 33, 100], where the background information is attenuated and the target region becomes dominant.

In this chapter we propose a disparity estimation algorithm based on foveal images, that serves the two purposes:

- It computes dense disparity maps that are used for object depth segmentation.
- It computes dominant horizontal disparity that is used for head vergence control.

The algorithm is based on multiple disparity hypothesis testing, in a Bayesian framework, and use foveal images that favor objects centered in the field of view (under tracking).

Before describing the use of foveal images, in the remaining of this section we review the Bayesian formulation for the disparity estimation problem, in a regular cartesian image representation.

4.1.2 Bayesian Formulation

We formulate the disparity estimation problem in a discrete Bayesian framework similar to [22]. The method can be summarized in the following steps:

1. Define a finite discrete set of possible disparities and corresponding prior probabilities.
2. Given each disparity in the set, compute the likelihood of each pixel in the stereo pair, using a generative probabilistic model.
3. Use the Bayes rule to compute the posterior probability of each disparity value at every pixel, given the image data.
4. Identify, for each pixel, the disparity value with highest posterior probability.

In the following paragraphs we will describe in more detail each one of these steps.

The Prior Model

Taking the left image as the reference, disparity at point $\mathbf{x}_l$ is given by $\mathbf{d}(\mathbf{x}_l) = \frac{\mathbf{x}_l - \mathbf{x}_r}{2}$, where $\mathbf{x}_l = (x_l, y_l)$ and $\mathbf{x}_r = (x_r, y_r)$ are the locations of matching points in the left and right images, respectively. If a pixel at location $\mathbf{x}$ in the reference image is not visible in the right image, we say the pixel is occluded and disparity is undefined ($\mathbf{d}(\mathbf{x}) = \varnothing$). Let us consider a discrete finite set of disparities D, representing the disparity values which are more likely to exist in a certain environment:

$$D = \{\mathbf{d}_n\}, n = 1 \cdots N \tag{4.5}$$

For each location $\mathbf{x}$ in the left eye, we define the following set of hypothesis:

$$H = \{h_n(\mathbf{x})\}, n = 0 \cdots N \tag{4.6}$$

where h_n represent particular disparity values, $\mathbf{d}(\mathbf{x}) = \mathbf{d}_n$. Hypothesis $h_0(\mathbf{x})$ represents the occlusion condition ($\mathbf{d}(\mathbf{x}_l) = \varnothing$). We make the following assumptions for the prior likelihood of each disparity hypothesis:

- We define a prior probability of occlusion with a constant value for all sites:

$$Pr(h_0) = q \tag{4.7}$$

- We do not favor any *a priori* particular value of disparity. A constant prior is considered and its value must satisfy $Pr(h_n) \cdot N + q = 1$, which results in:

$$Pr(h_n) = (1 - q)/N \tag{4.8}$$

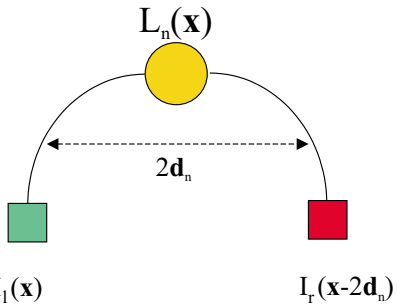


Figure 4.3: The likelihood function computes the similarity between pixels in the left and right images for each disparity hypothesis $\mathbf{d}_n$.

The Likelihood Model

This phase of the algorithm consists in evaluating the likelihood of each pixel in the acquired pair of images, $L_n(\mathbf{x})$, for each of the possible disparity hypothesis. We can imagine having, for each pixel, a set of computational units tuned to each of the disparities $\mathbf{d}_n$, that compute the degree of match between a pixel at location $\mathbf{x}$ in the left image and a pixel at location $\mathbf{x} - 2\mathbf{d}_n$ in the right image. This is illustrated in Fig. 4.3.

The disparity likelihood function $L_n(\mathbf{x})$ is defined according to the following assumptions:

- The appearance of object pixels do not change with view point transformations (Lambertian surfaces) and cameras have the same gain, bias and noise levels. This corresponds to the well known *Brightness Constancy Assumption* [76]. Considering the existence of additive noise, η, we get the following stereo correspondence model:

$$I_l(\mathbf{x}) = I_r(\mathbf{x} - 2\mathbf{d}(\mathbf{x})) + \eta(\mathbf{x}) \tag{4.9}$$

- In the unoccluded case, the probability of a certain gray value $I_l(\mathbf{x})$ is conditioned by the value of the true disparity $\mathbf{d}(\mathbf{x})$ and the value of I_r at position $\mathbf{x} - 2\mathbf{d}(\mathbf{x})$: Restricting disparity values to the set D, we write:

$$Pr(I_l|h_n, I_r) = Pr(I_l(\mathbf{x})|\mathbf{d}_n, I_r(\mathbf{x} - 2\mathbf{d}_n))$$

- Noise is modeled as being independent and identically distributed with a certain probability density function, f. Thus, the above likelihood is given by:

$$P_r(I_l|h_n, I_r) = f(I_l(\mathbf{x}) - I_r(\mathbf{x} - 2\mathbf{d}(\mathbf{x})))$$

  Generally, zero-mean Gaussian white noise is accepted as a reasonable model to work with, thus having $f(t) = 1/\sqrt{2\pi\sigma^2}e^{-t^2/2\sigma^2}$, where σ^2 is the noise variance.

- If a pixel at location $\mathbf{x}$ is occluded in the right image, its gray level is unconstrained and can have any value in the set of M admissible gray values,

$$Pr(I_l|h_0, I_r) = \frac{1}{M} \tag{4.10}$$

Given the previous assumptions, the disparity likelihood function is given by:

$$L_n(\mathbf{x}) = Pr(I_l(\mathbf{x})|h_n(\mathbf{x}), I_r(\mathbf{x})) = \begin{cases} f(I_l(\mathbf{x}) - I_r(\mathbf{x} - 2\mathbf{d}_n)) & \Leftarrow \quad n \neq 0 \\ \frac{1}{M} & \Leftarrow \quad n = 0 \end{cases} \tag{4.11}$$

The Posterior Model

With the previous formulation, the disparity estimation problem fits well in a Bayesian inference framework. The probability of a certain hypothesis given the image gray levels (posterior probability) is given by the Bayes' rule:

$$Pr(h_n|I_l, I_r) = \frac{Pr(I_l|h_n, I_r)Pr(h_n, I_r)}{\sum_{i=0}^{N} Pr(I_l|h_i, I_r)Pr(h_i, I_r)} \tag{4.12}$$

where we have dropped the argument $\mathbf{x}$ because all functions are computed at the same point. Since the unconditioned random variables h_n and I_r are independent, we have $Pr(h_n, I_r) = Pr(h_n)Pr(I_r)$. Using this fact and (4.11), the above equation simplifies to:

$$Pr(h_n|I_l, I_r) = \frac{L_n Pr(h_n)}{\sum_{i=0}^{N} L_i Pr(h_i)} \tag{4.13}$$

Now, substituting the priors (4.10), (4.7) and (4.8) in (4.13), we get the disparity posterior probability:

$$Pr(h_n|I_l, I_r) = \begin{cases} \frac{L_n}{\sum_{i=1}^{N} L_i + qN/(M-qM)} & \Leftarrow n \neq 0 \\ \frac{qN/(M-qM)}{\sum_{i=1}^{N} L_i + qN/(M-qM)} & \Leftarrow n = 0 \end{cases} \tag{4.14}$$

Maximum *A Posteriory* Estimation

The choice of the hypothesis that maximize the above equations leads to the MAP (*maximum a posteriori*) estimate of disparity[4]. However, without any further assumptions, there may be many ambiguous solutions. It is known that in the general case, the stereo matching problem is under-constrained and ill-posed [130], especially in image regions with uniform brightness. On a pixel by pixel basis, in low-textured image areas, the disparity posterior probability may have similar values for many disparity hypothesis. One way to overcome this problem is to assume that neighbor pixels tend to have similar disparities. In this work we will assume that the scene is composed by piecewise smooth surfaces, and will allow spatial interactions between neighboring pixels.

4.2 Dense Disparity Estimation on Foveal Images

In this section we extend the Bayesian framework presented before to cope with foveal images and to remove ambiguities in low-textured regions. Few approaches have been proposed to compute disparity maps in foveated active vision systems. Existing ones rely on the foveated pyramid representation [87, 136, 24]. To our knowledge, the only work to date addressing the computation of stereo disparity in *logmap* images is [71]. In that work, disparity maps are obtained by matching Laplacian features in the two views (zero crossings), which results in sparse disparity maps. In this paper we describe a

[4]The terms in the denominator are normalizing constants and do not need to be computed explicitly.

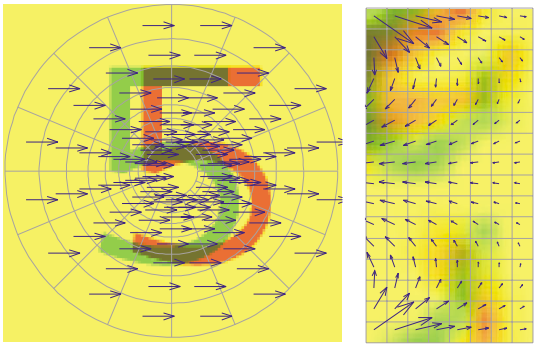


Figure 4.4: A space invariant shift in retinal coordinates (left) corresponds to a space variant warping in the foveal array.

stereo algorithm to compute dense disparity maps on foveal images. In our context, dense representations are advantageous for object segmentation and region of interest selection. We apply the method to a *logmap* based system, but its principle is valid for any imaging geometry.

4.2.1 Adaptation to Foveal Images

In cartesian coordinates the likelihood functions $L_n(\mathbf{x})$ compute the degree of match between a pixel at location $\mathbf{x}$ in the left image and a pixel at location $\mathbf{x} - 2\mathbf{d}_n$ in the right image. A computationally efficient way to obtain the likelihood functions is to create *disparity shifted* images $I_r^n(\mathbf{x})$ computed from the right image by shifting all pixels by an amount $2\mathbf{d}_n$:

$$I_r^n(\mathbf{x}) = I_r(\mathbf{x} - 2\mathbf{d}_n), \quad n = 1 \cdots N \tag{4.15}$$

Now, the likelihood function can be interpreted as an image and can be obtained, in the non-occluded case, by:

$$L_n(\mathbf{x}) = f(I_l(\mathbf{x}) - I_r^n(\mathbf{x})) \tag{4.16}$$

However, in foveal coordinates, the disparity shifts are different for each pixel. For example, with the *logmap* transformation, the correspondence map is shown in Fig. 4.4.

Let $z = l(\mathbf{x})$ represent the foveal coordinate transformation. For a given disparity hypothesis $\mathbf{d}_n$, the pixel shifts in foveal coordinates are different for each image location and can be computed by:

$$\mathbf{z}_r^n(\mathbf{z}_l) = \mathbf{l}\Big(\mathbf{l}^{-1}(\mathbf{z}_l) - 2\mathbf{d}_n\Big) \tag{4.17}$$

This map can be computed off-line for all foveal locations and stored in a look-up table to speed-up on-line calculations. To minimize discretization errors, the weights for intensity interpolation can also be pre-computed and stored. A comprehensive explanation of this technique, for the *logmap* case, can be found in [100].

Once the correspondence maps $\mathbf{z}_r^n(\mathbf{z}_t)$ have been computed, the disparity estimation procedure is equivalent to what was described before in cartesian coordinates. In summary:

- Let $I_{fov_l}(\mathbf{z})$ and $I_{fov_r}(\mathbf{z})$ be the left and right foveal images. Compute the foveal *disparity warped* images, $I_{fov_r}^n(\mathbf{z})$, by:

$$I_{fov_r}^n(\mathbf{z}) = I_{fov_r}\big(\mathbf{z}_r^n(\mathbf{z})\big)$$

- Compute the $N+1$ *foveal likelihood images*, $L^n_{fov}(\mathbf{z})$, that express the likelihood of a particular hypothesis at foveal location $\mathbf{z}$:

$$L^n_{fov}(\mathbf{z}) = f(I_{fov_l}(\mathbf{z}) - I^n_{fov_r}(\mathbf{z}))$$

- Using (4.14), compute foveal posterior probabilities:

$$Pr_{fov}(h_n|I_{fov_l}) \propto \begin{cases} L^n_{fov}(\mathbf{z}) & \Leftarrow n \neq 0 \\ qN/(M - qM) & \Leftarrow n = 0 \end{cases} \tag{4.18}$$

4.2.2 Dealing with Ambiguity

In a biological perspective, the value of the likelihood images L^n_{fov} at each foveal location $\mathbf{z}$ can be interpreted as the response of disparity selective binocular neurons in the visual cortex, expressing the degree of match between corresponding locations in the right and left retinas. When many disparity hypothesis are likely to occur (e.g. textureless areas) several neurons tuned to different disparities may be simultaneously active. In a computational framework, this "aperture" problem is usually addressed by allowing neighborhood interactions between units, in order to spread information from non-ambiguous regions to ambiguous regions. A Bayesian formulation of these interactions leads to Markov Random Field techniques [23], whose existing solutions (annealing, graph optimization) are still very computationally expensive. Neighborhood interactions are also very commonly found in biological literature and several cooperative schemes have been proposed, with different facilitation/inhibition strategies along the spatial and disparity coordinates [101, 118, 116]. For the sake of computational complexity we propose a spatial-only facilitation scheme whose principle is to reinforce the output of units at locations whose coherent neighbors (tuned for the same disparity) are active. This scheme can be implemented very efficiently by convolving each of the *foveal likelihood images* with a low-pass type of filter, resulting on $N+1$ *Facilitated Foveal Likelihood Images*, F^n_{fov}. We use a fast IIR isotropic separable first order filter, which only requires two multiplications and two additions per pixel. Filters of large impulse response are preferred because information is spread to larger neighborhoods and favor larger objects, at the cost of missing small or thin structures in the image. Also, due to the space-variant nature of the foveal map, regions on the periphery of the visual field will have more "smoothing" than regions in the center. At this point, it is worth noticing that since the 70's, biological studies show that neurons tuned to similar disparities are organized in clusters on visual cortex area V2 in primates [77], and more recently this organization has also been found on area MT [46]. Our architecture, composed by topographically organized maps of units tuned to the same disparity, agrees with these biological findings.

4.2.3 Computing the Solution

Replacing in (4.18) the *foveal likelihood images* L^n_{fov} by their filtered versions F^n_{fov} we obtain $N+1$ *foveal disparity activation images*:

$$D^n_{fov} = \begin{cases} F^n_{fov}(\mathbf{z}) & \Leftarrow n \neq 0 \\ qN/(M - qM) & \Leftarrow n = 0 \end{cases} \tag{4.19}$$

The disparity map is obtained by computing the hypothesis that maximizes the *foveal*

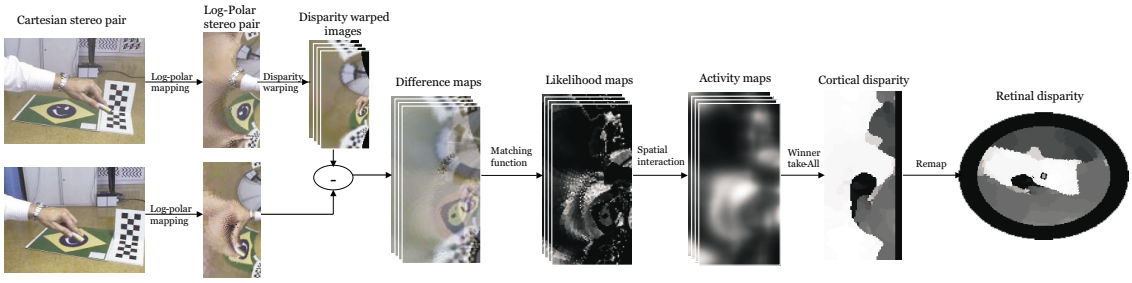


Figure 4.5: The disparity estimation algorithm with foveal images is schematically represented in the above diagram.

disparity activation images for each location:

$$\hat{\mathbf{d}}(\mathbf{z}) = \arg\max_n (D^n_{fov}(\mathbf{z}))$$

In a *neural networks* perspective, this computation is analogous a winner-take-all competition with inhibitory connections between non-coherent units at the same spatial location [2]. A block diagram of the full algorithm is shown in Fig. 4.5.

4.2.4 Dominant Disparity and Vergence Control

In previous works [14, 13, 12] we have shown that vergence control under tracking can be robustly attained using, as feedback signal, the dominant disparity in foveal images. Dominant disparity was computed by testing, from several global disparity hypothesis, the one maximizing correlation between images. The disparity estimation algorithm proposed here also compute the dominant disparity in a similar fashion. Recall that the foveal activity maps D^n_{fov} represent the activation of all cells tuned to the particular disparity $\mathbf{d}_n$. Thus, a method to estimate the dominant disparity $\mathbf{D}$ is simply obtained by accumulating the activation of such maps and choosing the one with maximal total activation:

$$\mathbf{D} = \arg\max_n \sum_{\mathbf{z}} D^n_{fov}(\mathbf{z}) \tag{4.20}$$

Head vergence angle can be controlled from the dominant disparity estimate using the methods presented in Chapter 3.

4.3 Results

We have tested the proposed algorithm on a binocular active vision head in general vergence configurations, and on standard stereo test images. In Figs. 4.6, 4.7 and 4.8 we show the original stereo pairs and the computed disparity maps. Bright and dark regions correspond to near and far objects, respectively. The innermost and outermost rings present some noisy disparity values due to border effects than can be easily removed by simple post-processing operations.

In the latter example, we show also the result of disparity segmentation. Notice that the uniform area at the right of the hand gets "fused" with the hand because it corresponds to a real ambiguity - the data in the images cannot tell if it belongs to the hand or to the floor plane. If the background is more textured, like in Fig. 4.9, this merging effect does not happen. Such type of ambiguity can only be removed by assuming that similar

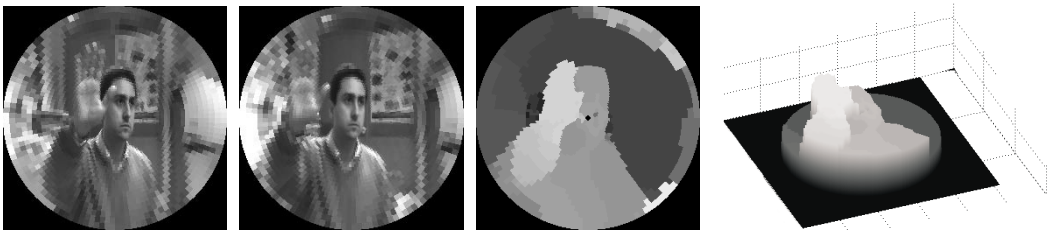


Figure 4.6: The images in the right show the raw foveated disparity map computed from the pair of images shown in the left, taken from a stereo head verging on a point midway between the foreground and background objects.

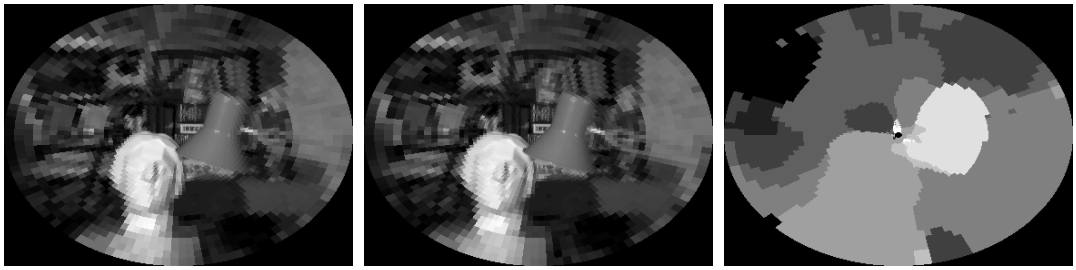


Figure 4.7: The disparity map on the right was computed from the well known stereo test images from Tsukuba University. In the left we show the foveated images of the stereo pair. Notice that much of the detail in the periphery is lost due to the space variant sampling. Thus, this result can not be directly compared with others obtained from uniform resolution images.

colors/gray levels are likely to correspond to the same object. The proposed method only groups regions of similar disparity.

Some intermediate results of the first experiment are presented in Fig. 4.10, showing the output of the foveal likelihood and the foveal activation for a particular disparity hypothesis. In the likelihood image notice the great amount of noisy points corresponding to false matches. The spatial facilitation scheme and the maximum computation over all disparities are essential to reject the false matches and avoid ambiguous solutions.

A point worth of notice is the blob like nature of the detected objects. As we have pointed out in Section 4.2.2, this happens because of the isotropic nature and large support of the spatial facilitation filters. Also, the space variant image sampling, blurs image detail in the periphery of the visual field. This results in the loss of small and thin structures like the fingertips in the stereo head example and the lamp support in the Tsukuba images. However note that spatial facilitation do not blur depth discontinuities because filtering is not performed on the disparity map output, but on the likelihood maps before the "max" operation.

The lack of detail shown in the computed maps is not a major drawback for applications like people tracking, obstacle avoidance and region of interest selection. As a matter of fact, it has been shown in a number of works that many robotics tasks can be performed with coarse sensory inputs if combined with fast control loops [127].

The parameters used in the tests are the following: log-polar mapping with 128 angular sections and 64 radial rings; retinal disparity range from -40 to 40 pixels (horizontal) and from -6 to 6 pixels (vertical), both in steps of 2; $q = 0.1$ (prior probability of occlusion); $M = 256$ (number of gray values); $\sigma = 3$ (white noise standard deviation); facilitation

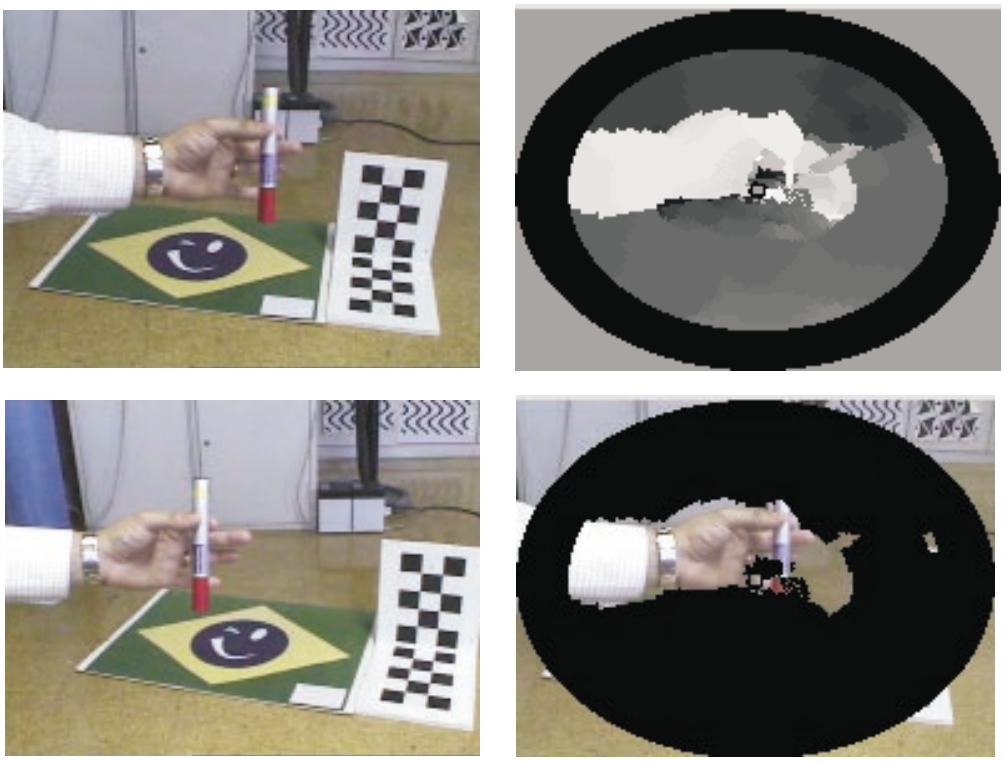


Figure 4.8: This figure show the application of the disparity estimation algorithm to color images. In the column we show the images composing the stereo pair. In the left column we show the computed disparity map (top) and object segmentation based on disparity.

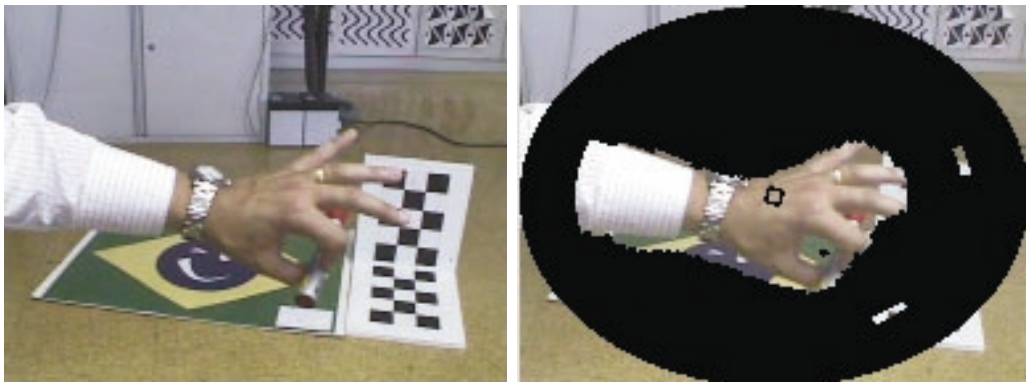


Figure 4.9: One example of foreground object segmentation with more textured background.

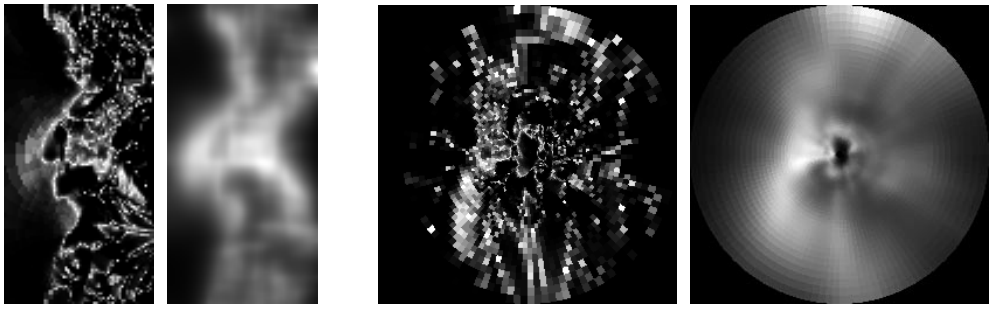


Figure 4.10: Intermediate results for the experiment in Fig. 4.6. This figure shows the foveal maps tuned to retinal disparity $d_i = 26$, for which there is a good match in the hand region. In the left group we show the likelihood images L^i_{fov} (left) and D^i_{fov} (right) corresponding to the foveal activation before and after the spatial facilitation step. In the right group, the same maps are represented in retinal coordinates, for better interpretation of results.

filtering with zero-phase forward/reverse filter $y(n) = 0.8y(n-1)+0.2x(n)$. The algorithms were implemented in C++ and run at 4Hz in a P4 computer at 2.66MHz.

Chapter 5

Motion Estimation and Tracking

This chapter focuses on the tracking problem, i.e. maintaining gaze on an object of interest while it moves around in the environment. For this purpose we must compute reliable information about the relative target motion (position and/or velocity), with respect to the robot head. In previous works [15, 12, 16], we have addressed the problem in a purely reactive fashion, composed by two steps. First, the disparity segmentation algorithm selected points belonging to the closest object in front the binocular head. Second, centroid and average velocity estimates of the segmented region were used for controlling the pan and tilt joints of the stereo head. This strategy has shown good performance for initiating a tracking process but revealed two main problems: (i) it does not keep a model for the tracked object which, in cluttered environments, may produce undetectable target losses or shifts; (ii) the object centroid estimates are very noisy and optic flow measurements accumulate errors along time, thus leading to difficulties in the control system.

Other research groups with robot heads have addressed the tracking problem in similar fashions. For example, [39], though not using foveal images, use disparity segmentation to segregate the target from the background, *via* a zero-disparity-filter (ZDF) and then compute the centroid of the ZDF features to control the pan and tilt angles of the stereo head. In [8] both vergence and tracking are controlled from optical flow derived measurements. Target detection is obtained by motion segmentation assuming a static background and discounting the optical flow produced by head rotation. In [122], ego-motion is also discounted to segment clusters of independently moving corners. Affine structure of the corner clusters is used to improve robustness to corner drop-out and reappearance, thus improving the stability of target position estimation. In [149], both motion and disparity segmentation are used to identify the target to track. The background is modeled as having an affine motion, which is estimated and discounted to the whole image to identify independently moving objects. Motion in the target is also assumed to have affine velocity, and is estimated to provide the control system with the appropriate reference signals. All these works lack object persistence, i.e. no model of object shape or appearance is employed, which often results in target losses or shifts.

In this work we present methods to improve the tracking system and overcome some of the problems described above. In one hand, we provide a model for the object being tracked, by the means of a deformable template. In the other hand, we estimate position measurements instead of velocity, thus allowing head control without target drifting. We assume a parametric model for object motion such that the proposed algorithm is able to track objects whose changes in appearance follow some *a priori* geometric deformation model. In principle the system copes with any parametric deformation model, but the best performance is obtained with few degrees of freedom (d.o.f.) models, including the

often used *affine*, *scaled-euclidean* and *euclidean* models.

The problem of parametric motion estimation has been very debated in the computer vision community. Due to the complexity of object shapes in the world, it is often considered that scenes are composed by approximately planar patches. For example, in aerial mapping or ocean floor exploration, the ground can be approximated by a plane if the camera is distant enough [69]. Therefore, planar surfaces provide a simple and manageable model to work with. A planar patch moving in the 3-dimensional world suffers deformations in the image that can be described by a 8 d.o.f projective model. The affine model (6 d.o.f) provides a good approximation that is sufficient in most cases. In other cases, simpler models like the scaled-euclidean (4 d.o.f), or euclidean (3 d.o.f.) can represent well enough the motion of regions in the images. Therefore many methods have been proposed to track the motion of planar patches in the images. Many rely on optic flow techniques [128, 11, 106] that accumulate errors along time and drift from the expected solution. Instead, our approach is similar to [72]. An initial template of the object is registered with the incoming images at every time step, and is not prone to the velocity estimation bias characteristic of optic flow methods.

Besides adapting the motion estimation algorithm to foveal images, we propose three main improvements to the algorithm described in [72]. First, we obtain large gains in computation time because we specify the optimization problem in a time-fixed coordinate frame, where most of the computation can be done only once at initialization. Second, the convergence range is increased by using a redundant parameterization of the geometric deformations. Finally, robustness is improved by a hierarchical organization of the computations, estimating first the more stable deformation parameters, and later the more noise sensitive parameters.

This chapter is organized in the following way: Section 5.1 describes the visual tracking algorithm, where for the sake of simplicity, the usual cartesian representation of images is considered. This algorithm is easily extended to the log-polar representation, which is summarized in Section 5.2. The adopted geometric deformation models and the hierarchical structure of the algorithm are described in Section 5.3. Experiments with simulated and real setups are presented in Section 5.4.

5.1 Parametric Motion Estimation

To address the tracking problem we assume that an object of interest has been previously detected, and the image region containing its pixels has been segmented. The segmentation algorithm can be based in stereo, like the one presented in the previous chapter, or any other feature (color, texture, etc.). The tracking problem consists in estimating the motion of the segmented image region along time and use the computed measurements to control the robot head such as to keep the object in the center of the images. For a system like ours, only the gaze direction can be controlled and it is enough to extract the translation parameters from the whole motion description. However, for reliable motion estimation, a complete enough motion model must be employed.

The proposed approach has the following characteristics:

- We assume that target motion in the image plane can be described by parametric geometrical image transformations. In particular, we consider transformations that approximate the motion of planar surfaces in the environment.
- The problem is formulated as the optimization of an objective function in the space of motion parameters. An appropriate choice of reference frames allows the opti-

mization to be performed in time-invariant coordinates, improving the on-line computational efficiency of the algorithm.

- Motion is represented in a redundant basis, allowing the representation of image deformations with more degrees of freedom, which augments the range of convergence of the optimization algorithm.

This section is organized as follows. First we review the parametric motion estimation problem, and develop two similar formulations in parallel. The first formulation is equivalent to the one presented in [72], that consists in a conventional iterative minimization algorithm. The objective function to optimize considers the difference of the current image to the previous image deformed by the assumed motion model. At each time step, gradient information has to be computed in order to iterate the algorithm, and the "best" motion parameters are the ones that minimize the objective function. In the second formulation (the one we propose) motion is decomposed in *predicted* and *residual* motion fields, and the optimization of the objective function is made in time–fixed coordinates. The computation of gradient information for the iterative minimization algorithm is made only once at initialization, thus saving significant on-line computations. The two algorithms are derived simultaneously to clearly distinguish their differences, and realize how the proposed formulation results in significant computational improvements. The use of a redundant parameterization for geometric deformations allows further improvements to the optimization algorithm. Instead of computing an approximation to the partial derivatives of the objective function at a single scale, we build an augmented descriptions of differential information at several scales. Simulations show that such a strategy extends the convergence range of the algorithm.

5.1.1 Problem Formulation

Notation and problem formulation are similar to those presented in [72].

Notation

Let $I_t(\mathbf{x})$ denote image brightness of a pixel located at point $\mathbf{x} \in \mathbb{R}^2$ and at time t. We model image motion by the differentiable and invertible motion field $\mathbf{f}(\mathbf{x};\mu)$, where $\mu = (\mu_1, \mu_2, \cdots, \mu_n)^T$ is the motion parameter vector. A motion field maps 2D points to 2D points, representing point displacements (motion). The inverse motion field, $\mathbf{f}^{-1}$,maps back points to their original locations, as illustrated in Fig. 5.1, and verifies:

$$\mathbf{f}^{-1}(\mathbf{f}(\mathbf{x};\mu);\mu) = \mathbf{f}(\mathbf{f}^{-\mathbf{1}}(\mathbf{x};\mu);\mu) = \mathbf{x}$$

The solution of the "motion estimation problem" consists in recovering the motion parameter vector μ for each time instant. The ground truth value is denoted by $\mu^*(t)$ and the corresponding estimate by $\mu(t)$. Initially, at time $t = 0$, a set of image pixels is selected, defining a reference region $\mathcal{R} = \{\mathbf{x}_1, \cdots, \mathbf{x}_m\}$. The reference template $R(\mathbf{x})$ is defined as the pixel gray-level values of region $\mathcal{R}$ at time $t = 0$:

$$R(\mathbf{x}) = I_0(\mathbf{x}) \quad \text{for all} \quad \mathbf{x} \in \mathcal{R}$$

Assuming brightness constancy [76], all changes in image brightness in subsequent time steps can be described by the motion parameters $\mu^*(t)$ and the motion field $\mathbf{f}$, (see

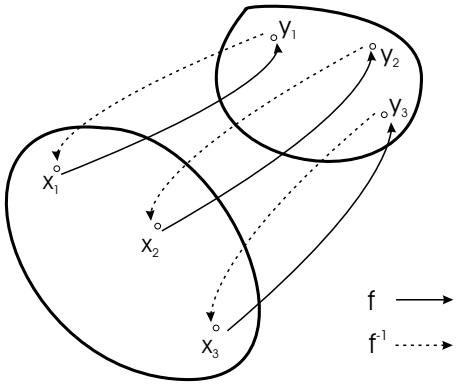


Figure 5.1: Point coordinates $\mathbf{x}_i$ are mapped to points $\mathbf{y}_i$ according to motion field f, and can be mapped back to original coordinates with the inverse motion field f^{-1}

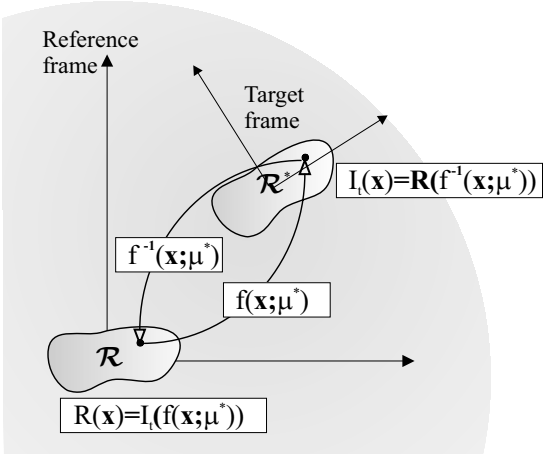


Figure 5.2: Representation of region and image motion using coordinate transformations.

Fig. 5.2):

$$I_t(\mathbf{f}(\mathbf{x};\mu^*(t))) = R(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{R} \tag{5.1}$$

or equivalently by the **inverse** motion field $\mathbf{f}^{-1}$:

$$I_t(\mathbf{x}) = R(\mathbf{f}^{-1}(\mathbf{x};\mu^*(t))) \quad \forall \mathbf{x} \in \mathbf{f}(\mathcal{R},\mu^*) \tag{5.2}$$

Warping operators

Notice that regions are mapped from the original space to the target space *via* the direct motion field, f, whereas the reference template is mapped to the current image by the inverse map, f^{-1}. Things get simpler if we define a pair of reciprocal warping operators, w and w^{-1}, that can be applied to both images and coordinates and always map the input to output spaces with the same transformation. Thus the *image warping* operators are defined as:

$$\begin{cases} w_\mu(I(\mathbf{x})) = I\left(f^{-1}(\mathbf{x};\mu)\right) \\ w_\mu^{-1}(I(\mathbf{x})) = I\left(f(\mathbf{x};\mu)\right) \end{cases} \tag{5.3}$$

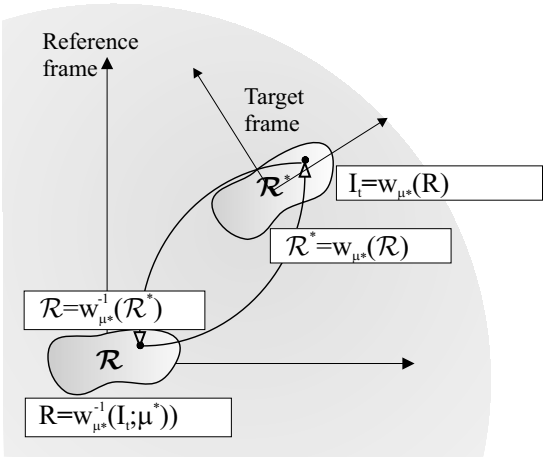


Figure 5.3: Representation of region and image motion using warping operators.

and the region warping operators are:

$$\begin{cases} w_\mu(\mathcal{R}) = f(\mathcal{R};\mu) \\ w_\mu^{-1}(\mathcal{R}) = f^{-1}(\mathcal{R};\mu)) \end{cases} \tag{5.4}$$

Thus, equalities (5.1) and (5.2) can be rewritten like:

$$R = \quad w_\mu^{-1}(I_t) \quad \forall \mathbf{x} \in \mathcal{R} \tag{5.5}$$

$$I_t = \quad w_\mu(R) \quad \forall \mathbf{x} \in w_\mu(\mathcal{R}) \tag{5.6}$$

Fig. 5.3 illustrates the representation of geometric transformations using the warping operators.

The Optimization Framework

Usual optimization techniques can recover the motion parameters by minimizing a least squares objective function. The objective function can either be expressed in the difference between the reference template and the inverse warped current image:

$$O_1(\mu) = \sum_{\mathbf{x}\in\mathcal{R}} \left[w_\mu^{-1}(I_t(\mathbf{x})) - R(\mathbf{x})\right]^2 \tag{5.7}$$

or express the difference between the current image and the direct warped reference template:

$$O_2(\mu) = \sum_{\mathbf{x}\in w_\mu(\mathcal{R})} \left[I_t(\mathbf{x}) - w_\mu(R(\mathbf{x}))\right]^2 \tag{5.8}$$

Note that in the first formulation the optimization parameters are applied to a **time-varying** image while in the second formulation they parameterize the **time-fixed** reference template. Notwithstanding, both formulations are equivalent.

5.1.2 Motion Decomposition

An usual assumption in tracking problems is to consider that motion is "smooth", i.e. it is continuous and does not suffer abrupt changes in time. This is not an unrealistic

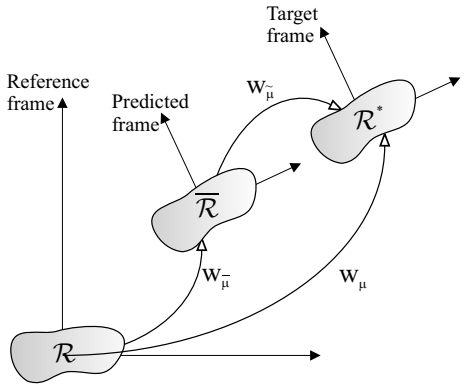


Figure 5.4: The full motion transformation, w_μ, is composed by a known predicted transformation, $w_{\bar{\mu}}$, and a unknown residual transformation, $w_{\tilde{\mu}}$.

assumption since motion arises from the displacement of physical objects, which is constrained by inertial physical laws. Thus, a good starting point for the search of target motion at time instant t can be obtained using information of past time steps. We denote this starting point as "initial guess" or "motion prediction" and represent it as $\bar{\mu}$. It can be obtained simply as the estimate of the motion field parameters in the previous time instant $\bar{\mu(t)} = \mu(t-1)$ or by a suitable prediction based on the past time information like in a Kalman Filter [64].

In general, the prediction will not coincide with the true motion and a residual error $\tilde{\mu}$ remains to be estimated. The residue is also called "innovation term" because it contains the component of motion that can not be predicted and must be computed by image processing algorithms. Using, as components, the prediction ($\bar{\mu}$) and innovation ($\tilde{\mu}$) terms, we define the composition rule that generates the full motion field (see Fig. 5.4):

$$\mathbf{f}(\mathbf{x};\mu) = \mathbf{f}(\mathbf{f}(\mathbf{x};\bar{\mu});\tilde{\mu})$$

or, in terms of warping operators:

$$\begin{cases} w_\mu(\cdot) = w_{\tilde{\mu}} \circ w_{\bar{\mu}}(\cdot) \\ w_\mu^{-1}(\cdot) = w_{\bar{\mu}}^{-1} \circ w_{\tilde{\mu}}^{-1}(\cdot) \end{cases} \tag{5.9}$$

where the operator $\circ$ represents warping composition.

Given the predicted motion vector at time t, we can recast the tracking problem as one of determining the "innovation term" $\tilde{\mu}(t)$. This can be obtained by applying the decomposition rule directly in (5.7):

$$O_1(\tilde{\mu}) = \sum_{\mathbf{x}\in\mathcal{R}} \left[w_{\bar{\mu}}^{-1} \circ w_{\tilde{\mu}}^{-1}(I_t(\mathbf{x})) - R(\mathbf{x}) \right]^2$$

or equivalently, in (5.8), pre-apply the operator $w_{\bar{\mu}}^{-1}$ to the images and summation regions:

$$O_2(\tilde{\mu}) = \sum_{\mathbf{x}\in\tilde{\mathcal{R}}} \left[w_{\bar{\mu}}^{-1}(I_t(\mathbf{x})) - w_{\tilde{\mu}}(R((\mathbf{x})) \right]^2$$

In the latter case the region of summation is given by $\tilde{\mathcal{R}} = w_{\tilde{\mu}}(\mathcal{R})$.

The rationale for the latter formulation is to first remove the known part of the transfor-

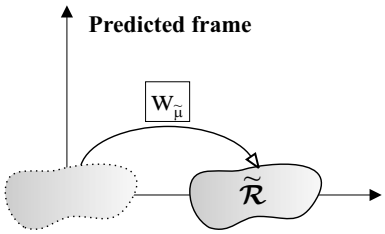


Figure 5.5: If the current image is inverse warped at the beginning of the optimization algorithm, according to the predicted motion transformation, the remaining transformation parameters can be estimated by local search around the origin.

mation and express the objective function in the predicted coordinate frame, as illustrated in Fig. 5.5. Notice that, in both objective functions, one of the images is known and fixed while the other is a variable dependent of the optimizing parameters $\tilde{\mu}$. The advantage of the second formulation is that the image depending on the unknown parameters is the reference template, thus allowing many operations to be precomputed at initialization, once the reference template is defined.

Differential Approximation

To formulate the optimization problem in terms of differential approximations we first have to adopt a vectorial representation of the involved images. Let us define the following vectors:

- The registered image
$$\bar{\mathbf{I}}_t(\tilde{\mu}) \triangleq \text{vec}\left[w_{\tilde{\mu}}^{-1} \circ w_{\bar{\mu}}^{-1}(I_t)\right]$$
- The warped template
$$\mathbf{R}(\tilde{\mu}) \triangleq \text{vec}\left[w_{\tilde{\mu}}(R)\right]$$

In the above expressions, the operator **vec** represents the stacking of all image pixels into a long column vector. The objective functions can be rewritten as:

$$O_1(\tilde{\mu}) = \sum_{\mathcal{R}} \left[\bar{\mathbf{I}}_t(\tilde{\mu}) - \mathbf{R}(\mathbf{0})\right]^2$$

$$O_2(\tilde{\mu}) = \sum_{\tilde{\mathcal{R}}} \left[\bar{\mathbf{I}}_t(\mathbf{0}) - \mathbf{I}_0(\tilde{\mu})\right]^2$$

Image $\bar{\mathbf{I}}_t(\mathbf{0}) = \mathbf{I}_t(\bar{\mu})$ is called the *predicted registration* and can be computed once per time step, by inverse warping the acquired image $I_t(\mathbf{x})$ with the predicted motion vector $\bar{\mu}$.

Assuming small magnitude for the components of $\tilde{\mu}$ we can make the following approximations:

- the *predicted image* $\bar{\mathbf{I}}_t(\tilde{\mu})$ at *fixed time t* can be approximated by a first order McLaurin series expansion with respect to the motion parameters:
$$\bar{\mathbf{I}}_t(\tilde{\mu}) \approx \bar{\mathbf{I}}_t(\mathbf{0}) + \bar{\mathbf{M}}_t \cdot \tilde{\mu} \tag{5.10}$$
where the matrix $\bar{\mathbf{M}}_t$ is the $m \times n$ matrix of partial derivatives of the *predicted*

registration $\bar{\mathbf{I}}_t(\mathbf{0})$ written in column form:

$$\bar{\mathbf{M}}_t = \left[\frac{\partial \bar{\mathbf{I}}_t}{\partial \tilde{\mu}_1}(\mathbf{0}) \mid \cdots \mid \frac{\partial \bar{\mathbf{I}}_t}{\partial \tilde{\mu}_n}(\mathbf{0}) \right]$$

- the image $\mathbf{R}(\tilde{\mu})$ can be approximated by a first order McLaurin series expansion:

$$\mathbf{R}(\tilde{\mu}) \approx \mathbf{R}(\mathbf{0}) + \mathbf{M}_0 \cdot \tilde{\mu}$$

where the constant matrix $\mathbf{M}_0$ is the $m \times n$ matrix of partial derivatives of the *reference image*, $\mathbf{R}(\mathbf{0})$, written in column form:

$$\mathbf{M}_0 = \left[\frac{\partial \mathbf{R}}{\partial \tilde{\mu}_1}(\mathbf{0}) \mid \cdots \mid \frac{\partial \mathbf{R}}{\partial \tilde{\mu}_n}(\mathbf{0}) \right]$$

With these assumptions we can rewrite the objective functions as follows:

$$O_1(\tilde{\mu}) = \sum_{\mathcal{R}} \left[\mathbf{D}_t + \bar{\mathbf{M}}_t \cdot \tilde{\mu} \right]^2 \tag{5.11}$$

$$O_2(\tilde{\mu}) = \sum_{\tilde{\mathcal{R}}} \left[\mathbf{D}_t - \mathbf{M}_0 \cdot \tilde{\mu} \right]^2$$

where, in both cases, $\mathbf{D}_t = \bar{\mathbf{I}}_t(\mathbf{0}) - \mathbf{R}(\mathbf{0})$ is the difference between the predicted registration and the reference template.

Again notice that, in the first formulation, the matrix of partial derivatives is time–varying while, in the second formulation, it is fixed for all time instances.

5.1.3 Computing the Solution

Both objective functions derived above are quadratic functions of the residual motion parameters, thus solutions to the optimization problem can be obtained in closed form by solving the set of equations $\nabla O = 0$. The solution yields in the first case:

$$\tilde{\mu} = -\left(\bar{\mathbf{M}}_t^T \bar{\mathbf{M}}_t \right)^{-1} \bar{\mathbf{M}}_t^T \mathbf{D}_t \tag{5.12}$$

and in the second formulation:

$$\tilde{\mu} = \left(\mathbf{M}_0^T \mathbf{M}_0 \right)^{-1} \mathbf{M}_0^T \mathbf{D}_t \tag{5.13}$$

Care should be taken with possible singularities in the motion covariance matrices $\bar{\mathbf{M}}_t^T \bar{\mathbf{M}}_t$ and $\mathbf{M}_0^T \mathbf{M}_0$. This may happen when there is not sufficient texture in the interest region and certain object motions may not be observable from the image gray-level variations. This is a generalization of the aperture problem [76].

Minimizing Online Computations

The work presented in [72] is based on an objective function of type O_1. Although derived with a different motion decomposition rule and temporal analysis, their objective function is equivalent to (5.11). Functions O_1 and O_2 have a similar aspect but the former contains a jacobian matrix $\bar{\mathbf{M}}_t$ that is time dependent while the latter contains a **constant**

Framework 1	**Framework 2**
offline steps	
Define the target region	Define the target region
Acquire and store the reference template	Acquire and store the reference template
–	Compute $\mathbf{M}_0$
–	Compute $\mathbf{M}_0^+ = \left(\mathbf{M}_0^T\mathbf{M}_0\right)^{-1}\mathbf{M}_0^T$
online steps	
Acquire new image	Acquire new image
Use a suitable motion prediction $\bar{\mu}$ to rectify the target region into the current image	Use a suitable motion prediction $\bar{\mu}$ to rectify the target region into the current image
Compute $\mathbf{D}_t$ by taking the difference between the predicted registration and the reference template	Compute $\mathbf{D}_t$ by taking the difference between the predicted registration and the reference template
Compute $\bar{\mathbf{M}}_t$	–
Compute $\bar{\mathbf{M}}_t^+ = \left(\bar{\mathbf{M}}_t^T\bar{\mathbf{M}}_t\right)^{-1}\bar{\mathbf{M}}_t^T$	–
Compute $\tilde{\mu} = \bar{\mathbf{M}}_t^+\mathbf{D}_t$	Compute $\tilde{\mu} = \mathbf{M}_0^+\mathbf{D}_t$
Compute μ by composing transformations $\bar{\mu}$ and $\tilde{\mu}$	Compute μ by composing transformations $\bar{\mu}$ and $\tilde{\mu}$

Table 5.1: Functional comparison between the proposed tracking algorithms

jacobian matrix $\mathbf{M}_0$. We propose a formulation based on objective function O_2. An objective function of this type is computationally advantageous since the jacobian matrix can be computed at initialization of the reference template, while the previous formulation requires the computation of the jacobian (or part of it) at run time[1]

Algorithmic efficiency

In an algorithmic point of view, there are some operations that can be performed in a **offline** phase (initialization) and other are performed **online** (at each time step). Since the derivation of the algorithm is based on local linearization, for good run-time performance the sampling period should be kept at minimum. Therefore, the online computation should be as fast as possible. Table 5.1 describes two computational algorithms that implement the solutions expressed in Eqs. (5.12) and (5.13).

Image Warping

Image warping is the process of obtaining a new image by applying a geometric transformation (motion field) to an original image. This process is needed to compute the *predicted registration* image. It is also used to compute image partial derivatives. A very simple means of implementing this procedure is by using look-up tables expressing the

[1]In [72], the jacobian is decomposed in the product of: image spatial gradient; motion field spatial derivatives; and motion field derivatives with respect to the motion parameters. Image spatial gradients can be calculated offline, on the reference template. Additionally, for some particular motion models, the jacobian matrix can be written as a product of a constant $m \times k$ matrix and a time varying $k \times n$ matrix, saving some online computation. However, there is always part of the jacobian that must be computed online.

Step	Operation	Alg.	Complex.
1	Acquire image	1, 2	–
2	Rectify image	1, 2	$O(m)$
3	Compute $\mathbf{D}_t$	1, 2	$O(m)$
4	Compute $\bar{\mathbf{M}}_t$	1	$O(m \times n)$
5	Compute $\bar{\mathbf{M}}_t^T \bar{\mathbf{M}}_t$	1	$O(m \times n^2)$
6	Compute $\left(\bar{\mathbf{M}}_t^T \bar{\mathbf{M}}_t\right)^{-1}$	1	$O(n^3)$
7	Compute $\bar{\mathbf{M}}_t^+$	1	$O(m \times n^2)$
8	Compute $\tilde{\mu}$	1, 2	$O(m \times n)$
9	Compute μ	1, 2	$O(n)$

Table 5.2: Online computational complexity

correspondences between pixel locations in the original and target image regions. More sophisticated methods can employ some kind of interpolation to improve the resulting image quality. In terms of computational complexity, image warping is $O(m)$, where m is the number of pixels in the interest region.

Discrete Derivatives

In both formulations we must compute partial derivatives of images with respect to the motion parameters. In framework 1 we must compute:

$$\bar{\mathbf{I}}_t^{(i)}(\mathbf{0}) = \frac{\partial \bar{\mathbf{I}}_t(\tilde{\mu})}{\partial \tilde{\mu}_i}\Big|_{\tilde{\mu}=0} \quad i = 1 \cdots n$$

and, in framework 2:

$$\mathbf{R}^{(i)}(\mathbf{0}) = \frac{\partial \mathbf{R}(\tilde{\mu})}{\partial \tilde{\mu}_i}\Big|_{\tilde{\mu}=0} \quad i = 1 \cdots n$$

A possible way to obtain discrete approximations to the partial derivatives consists in applying the formulas:

$$\bar{\mathbf{I}}_t^{(i)}(\mathbf{0}) \approx \frac{\bar{\mathbf{I}}_t(h \cdot \mathbf{e}_i) - \bar{\mathbf{I}}_t(0)}{h}$$

$$\mathbf{R}^{(i)}(\mathbf{0}) \approx \frac{\mathbf{R}(h \cdot \mathbf{e}_i) - \mathbf{R}(0)}{h}$$

where h is a "small" constant and $\mathbf{e}_i$ is a vector with value 1 at position i and 0 at all other positions. Using this method, to compute each partial derivative we must perform one image warping and one image difference. Therefore, the computation of $\mathbf{M}_0$ has complexity $O(n \times m)$.

Online complexity

Let us concentrate on the online complexity of the algorithms, which is the most important for real–time performance. Table 5.2 presents the required online operations as well as the computational complexity of each step. Assuming fixed dimension for the motion parameter vector, the overall complexity for each case is $O(m)$. However we can observe that algorithm 2 saves a great amount of computation because it does not need to compute online steps 4 to 7 (they are computed offline). Figure 5.6 presents the total number of arithmetic operations performed on the online steps of each algorithm as a function of

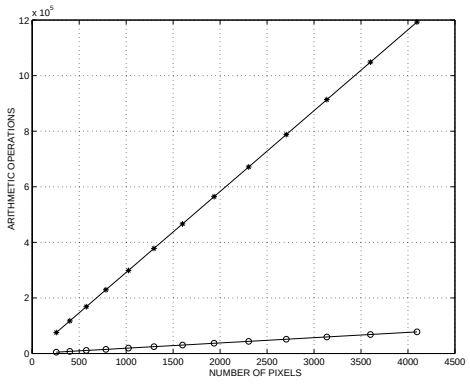


Figure 5.6: Number of operations for algorithms 1 ($*$) and 2 ($\circ$) as function of region dimension. Results obtained with $n = 8$

the number of pixels. Data is obtained for $n = 8$ (planar motion model). The gain in efficiency is obvious. Algorithm 2 online performance is about 15 times faster.

5.1.4 Redundant Parameterization

The motivation to propose a redundant parameterization for the motion vector comes from realizing that the representation of image deformations in a Taylor series expansion like Eq. (5.10) may not be complete enough. Considering that image regions may have hundreds of pixels (i.e. a vector in a very high dimension space) and that common motions models have few parameters, then representing one image by a linear combination of a few basis images (the partial derivatives) can lead to very bad approximations. Obviously, the approximation quality depends on image texture content but in general the approximation is only valid for very small perturbations. We propose to improve the approximation by enriching the linear combination with discrete derivatives at several directions and scales. A similar approach is proposed in [66].

Let us define a set of redundant motion vectors $\mathcal{V} = \{\tilde{\mu}_i, i \in (1 \cdots s)\}, s \gg n$. This set must be complete, such that any motion vector can be represented as a linear combination vectors on $\mathcal{V}$:

$$\tilde{\mu} = \sum_{i=1}^{s} k_i \cdot \tilde{\mu}_i \tag{5.14}$$

Since this new basis is redundant, multiple solutions may exist for the coefficients of the linear combination. Coefficients, $\mathbf{k} = (k_1, \cdots, k_s)^T$, represent a new set of parameters for the geometric deformation and implicitly define a new representation for image warping:

$$\mathbf{R}(\mathbf{k}) = \mathbf{R}(\tilde{\mu}) = \mathbf{R}(\sum_{i=1}^{s} k_i \cdot \tilde{\mu}_i)$$

The discrete partial derivatives of this new representation come:

$$\frac{\partial \mathbf{R}(\mathbf{k})}{\partial k_i}\Big|_{\mathbf{k}=0} = \frac{\mathbf{R}(h \cdot \tilde{\mu}_i) - \mathbf{R}(0)}{h}$$

If the discretization step h is unitary, the magnitude of the sample vectors $\tilde{\mu}_i$ represent the discretization scale. With this new parameterization, the image partial derivatives can be interpreted as derivatives at multiple directions and scales in the original representation

(direction and scale of each $\tilde{\mu}_i$).

The proposed motion estimation algorithm can be applied with no changes to this new representation. The jacobian matrix comes:

$$\mathbf{M}_0 = \left[\frac{\partial \mathbf{R}(\mathbf{k})}{\partial k_1}\Big|_{\mathbf{k}=0} \mid \cdots \mid \frac{\partial \mathbf{R}(\mathbf{k})}{\partial k_s}\Big|_{\mathbf{k}=0} \right]$$

which has dimension $m \times s$. The optimization process computes a solution for $\mathbf{k}$, and $\tilde{\mu}$ is given by Eq. (5.14). The computation time increases with the number of sample vectors, but since most of the computations are done offline, real–time performance can still be obtained. It is worth noticing that the basis images, $\mathbf{R}(\tilde{\mu}_i)$, are not linearly dependent because their dimension is in general much higher than the number of motion vectors in $\mathcal{V}$. However, this depends on image texture and care should be taken when running the optimization algorithm. We use a damped least-squares method [79] to compute the jacobian matrix pseudo–inverse.

One of the advantages of the redundant parameterization over the standard one is the ability to customize the set $\mathcal{V}$ of basis vectors according to the kind and range of expected image deformations. For instance, in a companion work [150], the basis vector set was updated on-line according to the latest estimated motion parameters, and we have developed a fast algorithm to compute recursively the jacobian matrix $\mathbf{M}_0$ using the matrix inversion lemma [67].

Also with the increase of computational power, we can easily add new sample vectors to improve the estimation results. The algorithm can be customized in order to estimate the larger and more constrained motions in the first iterations and the finer and more generic transformations in the last iterations, which improve its robustness. This will be further explained latter in this chapter.

Experiment Let us consider one simple example and compare the performance of standard and redundant parameterizations. The experiment consists in simulating image translations from -30 to 30 pixels in small steps (0.1 pixels). For each translation we apply both approaches and compare the solutions. Results are presented in Fig. 5.7. The first plot is relative to estimating translation with the standard representation. Several curves are presented, corresponding to the estimated translation with different number of iterations $(10, 20, \cdots 150)$. We can observe that in this case the convergence interval is limited to about ± 8 pixels. Another aspect of concern is the convergence speed. The number of iterations depends on the required precision – if translation is small, good estimates can be obtained with a few iterations, but more iterations are required in the limits of the convergence interval. The second plot shows the performance with the redundant parameterization. Again several curves are presented for the evolution of the estimation process with different number of iterations $(2, 4, \cdots 14)$. We can observe that the convergence interval is much larger in comparison with the previous method. Also, the convergence rate is higher – the algorithm reaches a stable solution in about 10 iterations.

Before we go onto foveal images, it is worth mentioning that this algorithm has been applied successfully in cartesian images for the station keeping of underwater and aerial vehicles, in the context of the European Project NARVAL[2]. Work related with this project is described in [70, 150].

[2]Navigation of Autonomous robots via Active Environmental Perception, Esprit-LTR Proj. 30185

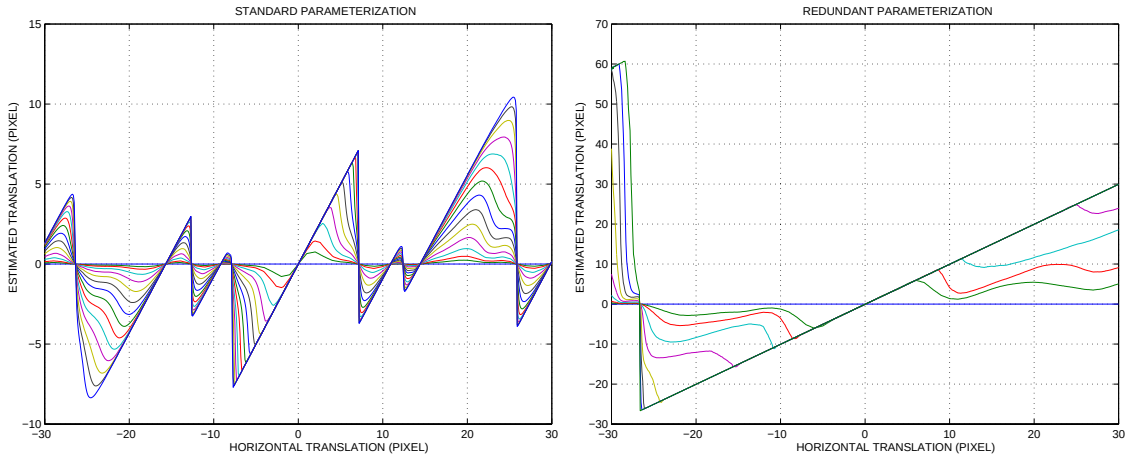


Figure 5.7: Comparison of different parameterizations for geometric deformation: **(left)** Standard parameterization. Notice the limited convergence range (about ± 8 pixels). Different lines correspond to different number of iterations $(10, 20, \cdots, 150)$. **(right)** Redundant parameterization. Different lines correspond to different number of iterations $(2, 4, \cdots, 14)$.

5.2 Adaptation to Foveal Images

Although the derivation of the motion estimation algorithm was done considering cartesian coordinates, its extension to log-polar coordinates is straightforward. The process to adapt the motion estimation algorithm to foveal images is very similar to the one described in Section 4.2.1. It consists in describing the deformation motion fields in foveal rather than cartesian coordinates.

Again, let $\mathbf{z} = \mathbf{l}(\mathbf{x})$ be the foveal coordinate transformation. The deformation motion fields are now given by:

$$\mathbf{f}^{fov}(\mathbf{z}; \mu) = \mathbf{l}\Big(\mathbf{f}\big(\mathbf{l}^{-1}(\mathbf{z}); \mu\big)\Big) \tag{5.15}$$

In terms of computation complexity, the transformation of an image according to a deformation field is the same in cartesian and foveal images. However, since foveal images have less pixels, these transformations are faster to compute, which is important to achieve high sampling rates and better tracking performance. An example of the application of a motion field including cartesian translations and rotations is shown in Fig. 5.8 for log-polar images.

5.3 Algorithm Implementation

To implement an algorithm based on the proposed method, some design choices must be taken, such as: (i) the deformation model (ii) the number and distribution of sample vectors $\tilde{\mu}_i$; (iii) the iterative (or not) structure of the algorithm, i.e. the number of iterations and/or the stopping criteria.

The choice of a deformation model depends on the considered application. Target shape and motion should be taken into account when deciding this point. One thing to take into consideration is that more constrained transformations(with less parameters) are more robust to non modeled aspects of the deformations. Less constrained models (with mode degrees of freedom) are in general less stable but for some applications may

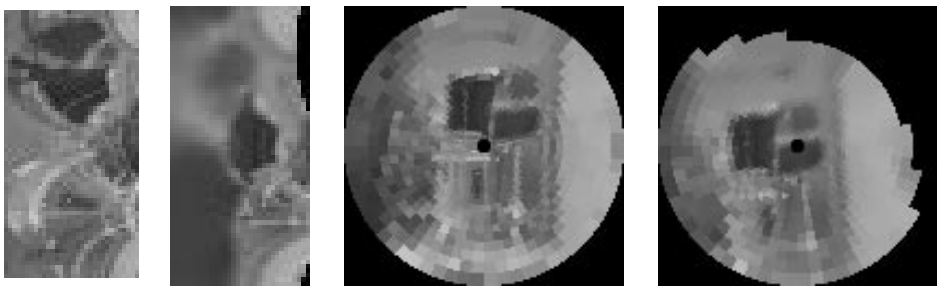


Figure 5.8: The original log-polar (on the left) is warped according to a transformation that includes translation and rotation (second from the left). The corresponding retinal images are also shown (right).

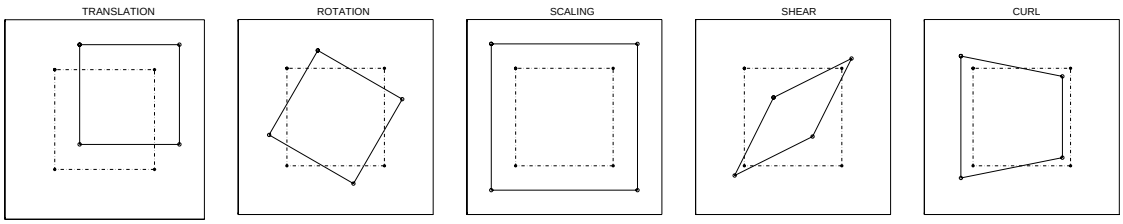


Figure 5.9: 2D projective transformations can be decomposed in translation, rotation, scaling, shear and curl.

be required. Planar surfaces are often used in robotic applications, since they can be found in many human made environments and represent good approximations for other types of surfaces. In such cases, the use of a projective motion model can provide very good motion estimates which may be needed for precise pose computation or trajectory generation. In this paper we use two types of motion models. In the simulations we use the projective model, thus being able to estimate all deformations of planar surfaces motion. The projective model has eight degrees of freedom and includes deformations such as translation, rotation, scaling, shear and curl, as illustrated in Fig. 5.9. In experiments with real images we use a rigid motion model composed by translations and rotations.

In terms of number of iterations, we chose to have a fixed number. Since we are mostly interested in real-time implementations it is more important to have a fixed computation time than a very precise tracking, therefore we fix the number of iterations at 3 for the planar model an at 2 for the rigid model. Regarding the choice of the sample vectors, and after many experiments we chose to create three sets for the planar model: one with translation vectors, one with affine vectors and one with projective vectors. Each set is composed by 48 vectors with non–uniform distributions, sampling more densely small deformations but still considering large deformations. For instance, the sample translation set is composed by vectors that translate the template by amounts $(x, y) \in \{-6, -3, -1, 1, 3, 6\}^2$. The idea is to have good precision when the deformations are small but still be able to detect large deformations. The three iterations are organized in the following way:

- The first iteration uses **sample translation vectors**. Since in the beginning a large deformation is likely to exist, it is more robust to estimate more constrained transformations. This iteration is intended to center the template with the current image and leave to the next iterations the estimation of the remaining deformations.

- The second iteration uses **sample affine vectors**. This iteration should estimate most of the rotation, scaling and shear present in the transformation.
- The last iteration uses **sample projective vectors**. It should estimate the remaining deformations and make the final fine adjustment to the template. Since this set spans 8 degrees of freedom, it should be used only with small deformations, otherwise it is likely to produce erroneous results.

Following the same ideas, for the rigid model we use two sets of vectors (one for each iteration). The first one uses the same 48 translation vectors. The second use 24 rotations, also with non–uniform distribution around the origin.

5.4 Evaluation of Results

Several experiments are shown to evaluate the performance of the proposed methodologies. In particular we are interested in testing the motion estimation algorithm, evaluating the benefits of using foveated images and checking the system in real situations, in particular with objects deviating from the assumed deformation models. In the first experiment we simulate camera motions that produce, in the retinal plane, increasing image translations, rotations and scalings. This experiment shows the range limitations of the algorithm, i.e., the maximal image deformations allowed between two consecutive images. In the second set of experiments we compare the use of cartesian and log-polar images. Again we simulate image motion for objects of different sizes in order to have ground truth data. In the third experiment we use the real setup and evaluate qualitatively the performance of the full system and its ability to cope with non modeled aspects.

5.4.1 Performance Evaluation

In these experiments we evaluate the algorithm convergence range with translations, rotations and scalings. The algorithm is applied to increasingly larger motions and at each iteration the estimated and real transformations are compared. The performance criterion is given by the L2–norm of the vector that contains the corner displacements (in pixel) of a polygonal window defining the template. The results are presented in Fig. 5.10 and show that, for the images and transformations used, the algorithm is able to estimate with precision (error less than half pixel per window corner) about 10 pixels translation, 11 degrees rotation, 18% zoom–in or 24% zoom–out. These values correspond to the biggest motions between images that the algorithm can cope with.

5.4.2 Advantages of Foveal Images

To evaluate the performance of the algorithms with ground truth data we developed a simulator for the system. We assume a simple first order dynamic model for the velocity of the pan and tilt joints with a time constant of 200 msec and the sampling frequency is 10Hz (100 msec), which define a relatively slow dynamics. Therefore the control is not "one step" but instead has a lag that depends on target velocity and the dynamic model parameters. In this case pan and tilt joints are controller from the target motion estimated from only one camera (dominant eye).

We use two planar surfaces to simulate the environment : one is the background located $10m$ away from the camera and the other is the target at $0.5m$. We tested different scales for the target, from 36% to 2.25% of the full image area (see Fig. 5.11).

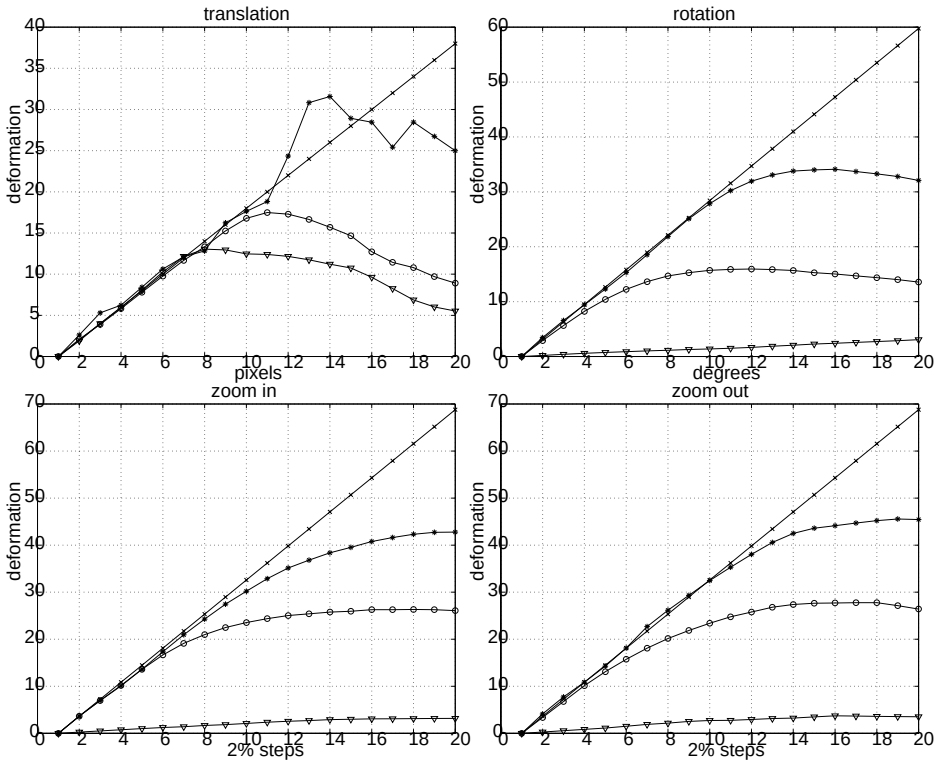


Figure 5.10: Performance of the algorithm for translation, rotation, zoom–in and zoom–out transformations. $\times$ – ground truth. ∇ – first iteration (translation sample vectors). $\circ$ – second iteration (affine sample vectors). $*$ – third iteration (planar sample vectors)

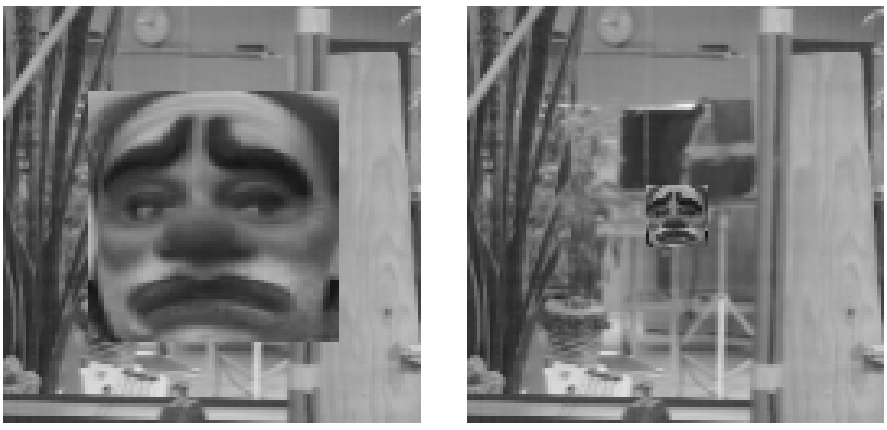


Figure 5.11: Simulated images with targets of scales 36% (left) and 2.25% (right).

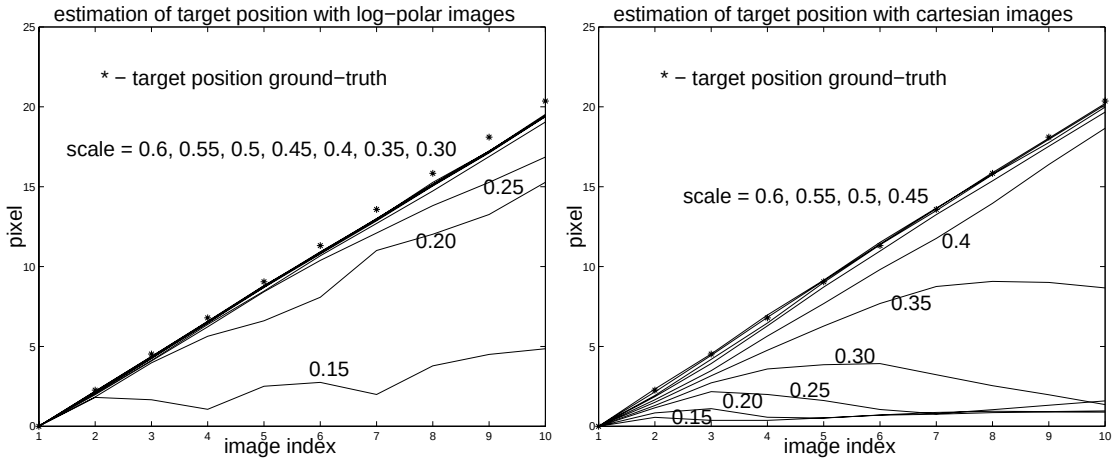


Figure 5.12: Comparison between log-polar **(left)** and cartesian **(right)** versions of the open-loop experiment. The true and estimated target position are represented for targets of several dimensions.

Open-loop test In this simulated experiment the cameras do not move. We compare the use of log-polar and cartesian images with objects of different sizes. The actual dimension of the object **is not known** *a priori*, therefore the system selects an initial template that occupies the full image except a small border region in the periphery on the view field. The target translates linearly in 3D space. At each time instance the algorithm estimates target position, which is used as initial guess to the next time step. In Fig. 5.12 we present plots of the estimated template position for the log-polar and cartesian versions of the algorithm. From these plots we can observe that the performance of both versions is good for large objects but degrades when target size diminishes. Notwithstanding, the log-polar version copes with smaller objects than the cartesian version.

Closed-loop test This is also a simulated experiment and illustrate the integration of motion estimation and active camera control. Simulated pan and tilt angles are controlled to keep the observation direction on the center of the target. The target moves with constant velocity during the first 15 time steps and then stops. In this case the displacements can be larger than in the previous experiment because the target is actively kept inside the field of view. Results are shown in Fig. 5.13. Notice that a 9% size object is not tracked by the cartesian algorithm. Even for 36% size, cartesian tracking is not very stable and sometimes looses track of target motion. The log-polar algorithm performs very well in both cases, presenting a tracking error less than two pixels in the image plane.

5.4.3 Active Tracking of Real Objects

In this experiment we use a real pan/tilt setup and illustrate qualitatively the performance of the system tracking a face. The face moves in front of the system in a natural fashion, performing translations and rotations, both in depth and fronto–parallel to the system. The 2D motion model considered in this case consists in translation and rotation (3 degrees of freedom). We show plots with the estimation of the motion parameters along the sequence. For each plot we signal some special frames, also shown as images, for which the motion amplitude is large. Fig. 5.14 corresponds to the estimation of horizontal translation. Notable points are signaled at frames 275, 524, 560, 592, where target velocity

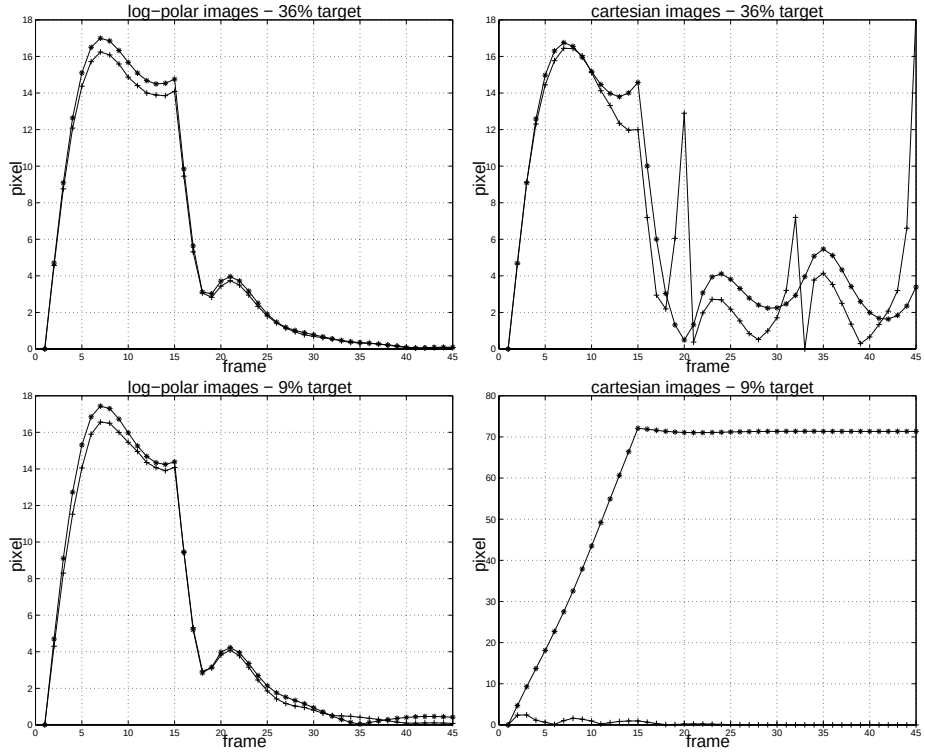


Figure 5.13: Estimated (+) *versus* true (∗) position of the target. Comparison between log-polar and cartesian versions of the algorithm with 36% and 9% size objects

attain high values in the horizontal direction. Figs. 5.15 and 5.16 show results for vertical translation and rotation. A qualitative analysis shows good robustness to non modeled deformations such as pose changes, scale changes and background motion.

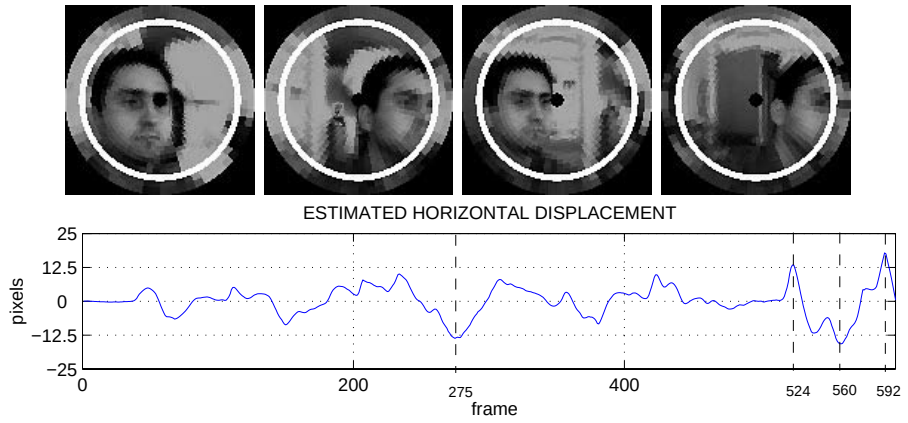


Figure 5.14: Estimation of horizontal translation along the active tracking experiment. The images shown on top correspond to the notable points signaled in the plot.

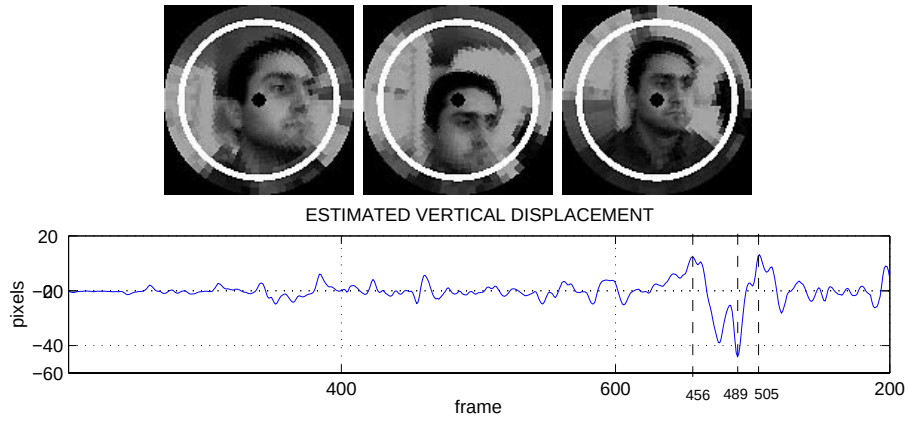


Figure 5.15: Estimation of vertical translation along the active tracking experiment.

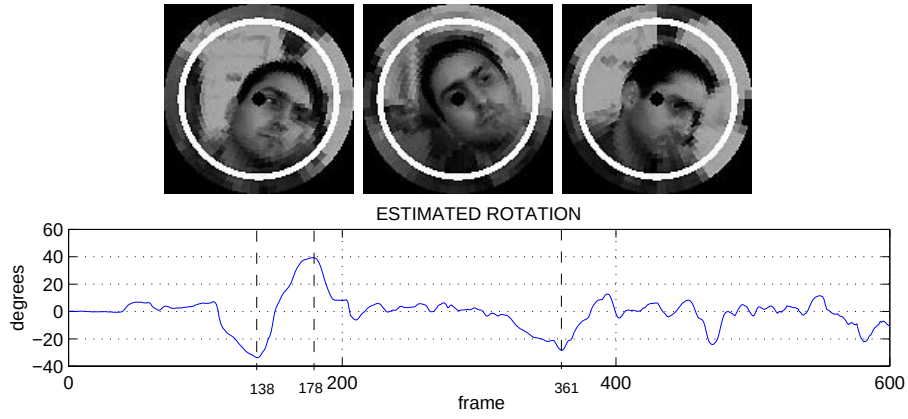


Figure 5.16: Estimation of rotation along the active tracking experiment.

Chapter 6

Visual Attention

Visual Attention addresses the problems of detecting regions of interest in the visual field and allocating visual resources to objects or events in the scene. Real world scenes are too complex to be fully understood in a short time range, either by biological or artificial visual systems. Hence, the visual system first pre-selects a set of interesting points, based on their "visual saliency", and sequentially inspects them to build a representation of the external environment, skipping non-interesting regions to reduce the computational requirements of the task.

Our main contribution in this chapter is threefold. Firstly, we address the problem of saliency computation in the log-polar space and propose a stratified approach: early feature extraction is performed in retinal coordinates, instead of log-polar, to avoid the loss of high frequency information; subsequent saliency operations are performed in log-polar space to improve computational efficiency. Secondly, we develop algorithms for extracting directionally selective features from images, based on Gabor filtering, with higher computational efficiency that state-of-the-art methods. Finally, we illustrate purely data-driven saliency and top-down saliency biasing, using directional, luminance and spatial-frequency features. Saliency computations are performed in log-polar and top-down modulation is hard coded to enhance saliency of particular structures in the images.

This chapter is organized as follows. The first section reviews the main computational models of visual attention in the literature. The second section addresses the problem of selective attention in log-polar space and show that usual feature extraction methods in log-polar space do not provide rich enough information for attentional mechanisms. Motivated by the finding of non-linear ganglion cells in the human retina, we propose that features containing significant frequency content should be extracted in cartesian coordinates, prior to foveation. The third section addresses the computation of directional features with Gabor wavelets and describes a novel fast method to compute such features. In the fourth section, we illustrate log-polar saliency computation with directional, luminance, and spatial frequency features, both in a purely data-driven context and in the search for eyes in face images.

6.1 Computational Models of Attentional Control

Modern theories of visual attention have started in the 1980s in experimental psychophysics studies. The pioneering works of [146] and [117] have inspired an avalanche of research in these topics and have served as the basis for ongoing theoretical developments. Experimental psychophysics findings have motivated (and still motivate) much of the work in

computational models and implementations of visual attentions mechanisms. For example, the computational modeling works of [88] and [148], and computer implementations of [82], [81], [105] and [147], follow very closely ideas arising from human psychophysics experiments.

The recently emerged two-component framework for attentional deployment suggests that attention is directed to particular points in a scene using both image-based saliency cues (bottom-up) and cognitive task-dependent cues (top-down). Most computational models of visual attention address mainly the bottom-up component, where the concept of *saliency* is central and constitutes the basis for selecting visual items. Saliency is related to the uniqueness or distinctiveness of an object among its neighbors, and several methods have been proposed to quantify its value : center-surround operations in feature maps [82, 81, 105]; scale-space maxima [83]; feature symmetry [123, 97]; local entropy maxima [85], amongst others.

Most common implementations of selective visual attention are motivated by the computational model of [88], which addresses the bottom-up stream of processing in visual attention. The top-down volitional component of attention has not been computationally modeled in detail and is still a controversial topic.

The main principles of the model of [88] are summarized in the following lines:

- Early visual features are computed pre-attentively from the original image and are topographically stored in feature maps. Frequently used features include luminance, color, edges, spatial-frequency and orientation.

- Conspicuity maps are created by non-linear interactions across distant spatial locations, aiming to promote feature maps with a small number of salient and isolated peaks.

- A unique saliency map is obtained by combining the conspicuity maps, and encodes stimuli saliency at every location in the visual scene in a topographically organized manner.

- The saliency map provides an efficient control strategy for visual search : the focus of attention simply scans the saliency map in order of decreasing saliency.

Particular implementations usually differ in the way each of the computational steps are implemented, but follow the general architecture. For example, the implementation in [82] has the following characteristics:

1. Feature Extraction – several feature maps are computed from the original color image, defined by its (r, g, b) components: (i) the intensity map is obtained by averaging the original color channels, $I = (r+g+b)/3$; (ii) local orientation features are derived from the convolution of the intensity image with Gabor filters tuned to 4 different orientations; (iii) color–opponent features are computed by first transforming the color space to RGBY, $R = r - (g + b)/2$, $G = g - (r + b)/2$, $B = b - (r + g)/s$, $Y = (r + g)/2 - |r - g|/2$, and then computing the color–opponent maps, $R - G$ and $B - Y$.

2. Conspicuity Maps – the feature maps are processed by center-surround units composed by Differences-Of-Gaussians at several scales. This is implemented by building Gaussian pyramids for each feature map and differencing levels 2, 3 and 4 from levels 3 and 4 scales above in the pyramid. A total of 6 contrast maps are built for each feature, in a total of 42 (6 for intensity + 12 for color + 24 for orientation). Then,

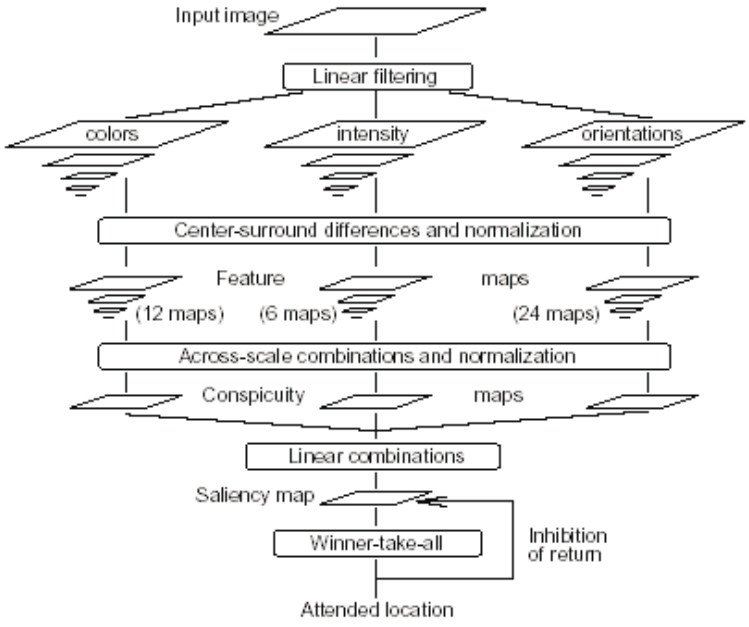


Figure 6.1: The attentional model of [82].

each the 42 contrast maps are processed to promote maps in which a small number of strong activation peaks are present. This is achieved by a normalization operation consisting of the following operations : (i) scale values to a fixed range, $[0 \cdots M]$; (ii) compute the average of all local maxima ($\bar{m}$) except the global one (M); (iii) multiply the whole map by $(M - \bar{m})^2$.

3. Saliency Maps – intensity, color and orientation conspicuity maps are combined by simple addition. Then the final saliency map is obtained by normalizing and summing the combined conspicuity maps.

A diagram of this model is shown in Fig. 6.1. In a latter implementation [81], the computation of the conspicuity maps is performed directly from the raw features in a iterative process emulating long-range inhibition mechanisms in the visual cortex. Each of the feature maps is repeatedly convolved by center-surround operators and half-wave rectified. This non-linear feature enhancement technique is more time consuming than the closed-form normalization of [82], but has the advantage of promoting blob like isolated peaks. It has shown very good results in the visual search phase, outperforming humans in the time required (number of gaze shifts) to reach the target [81]. However, the model at that stage did not have any recognition capabilities, thus comparison with human performance is not completely fair, because humans spend time with recognition processes at each gaze.

In a latter work [104], the bottom-up method described above was integrated with top-down models for searching people in outdoor scenes. Two top-down trainable methods were tried: one was based on Support Vector Machines (SVM) [112] and the other on the biologically motivated HMAX model [124]. Good performance was obtained with the SVM model trained to recognize people, with significant computational improvements over exhaustive search strategies. However, the SVM recognition module is still too slow to be used in real-time applications. Other criticism to the method is the complete separation between the bottom-up and top-down processes. Bottom-up information serves to guide

the recognition module to promising locations in the visual field but top-down information is never used to help the data-driven process.

In the model of [105], bottom-up and top-down modules are more tightly integrated. The feature extraction phase is very similar to the one in [82], but they consider also local curvature features and use 16 directions in the orientation filters. The non-linear feature enhancement phase is significantly different from the previous implementations. Center-surround operations are performed with elliptic, rather than isotropic, Difference-Of-Gaussians, with 8 different orientations and 3 scales. The center-surround maps are then rectified and squared, and the maximum along each orientation and scale is selected, forming one conspicuity map per feature. Then, an iterative relaxation process is employed, consisting in the minimization of an energy functional by gradient descent. The energy functional contains terms that penalize inter-map incoherence (when several center-surround maps enhance different conflicting image regions), intra-map incoherence (the existence of many non-convex regions), overall activity (to avoid energy growing without bound) and a final term to force the values in the map to be either a maximum or minimum of the considered range of values. The method is computationally very demanding but allows the introduction of top-down information in intermediate iterations of the relaxation procedure. Objects to be searched for, are given to the system in the form of templates, that are used to train a distributed associative memory. After two iterations of the relaxation process, the best bottom-up candidates are used to evaluate the likelihood of being objects of interest. Then the gradient descent rule of the minimization algorithm is changed to penalize locations with low likelihood, and the process is repeated in the remaining iterations. This way, bottom-up and top-down information are intertwined in the attentional process and cooperate for the computation of saliency.

An additional feature of the work of [105] is the use of an alerting mechanism, based on motion, to trigger attention. It uses a pyramid representation of the input to provide a coarse but fast detection of objects moving against a static background. When an object enters the field of view, the alerting mechanism takes control over the rest of the system and directly elicits an attentional shift. A diagram of the full model is shown in Fig. 6.2.

The selective tuning model of [147] is one of the few computational models to provide an implementation of non-spotlight focus of attention. The method can also be interpreted as a biologically motivated segmentation and grouping methodology. The basis of the method is a multi-resolution pyramid, with small receptive field units in the bottom layers and large receptive fields in the top layers. At each site and scale, the information is collected by "interpretive units" of different types. Each interpretive unit may receive feedback, feedforward and lateral interactions from other units. The method assumes that values of "saliency" or "interestingness" are already available at the input layer. Information is propagated upwards averaging the activation of children nodes, as usual. Once the information reaches the top level, the most salient items are identified and information is propagated downwards: the units not on the winner's receptive field are inhibited and the process continues to the bottom of the pyramid. As the pruning of connections proceeds downward, interpretive units are recomputed and propagated upward. The result of these feedback/feedforward connections is the segmentation of regions that are coherently salient at all scales. The attentional focus is directed coarsely at the pyramid top level and routed to the lower layers in a guided manner (see Fig. 6.3). To privilege certain parts or features in the visual field, top-down information can be externally introduced with *bias* units in the pyramid. Results have been shown for different features: luminance, oriented edges and optical flow patterns.

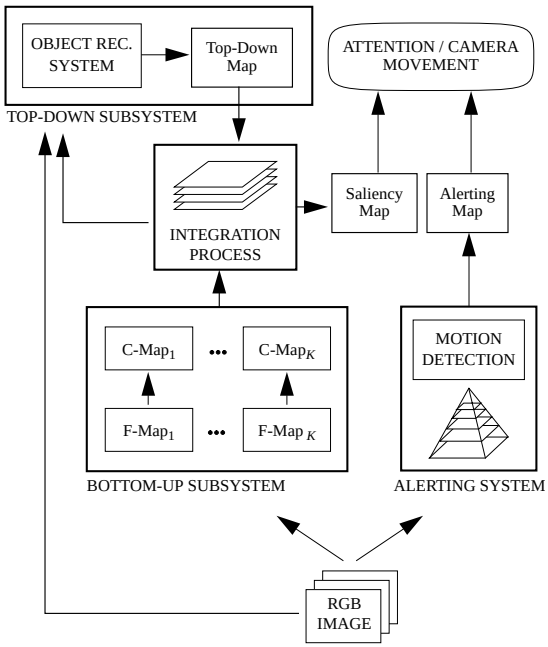


Figure 6.2: The attentional model of [105].

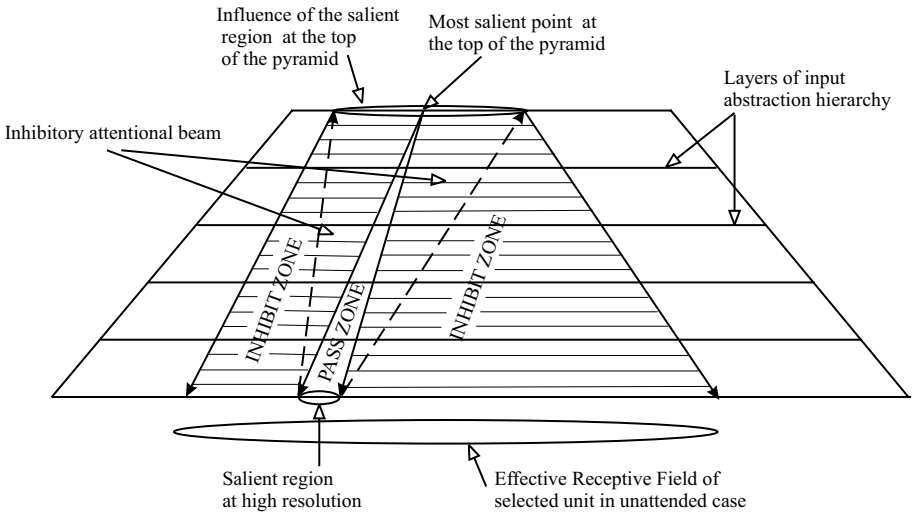


Figure 6.3: Information routing in the selective tuning model. Adapted from [147]

6.2 Visual Attention in Foveal Systems

As described the previous section, computational visual attention architectures rely on massive extraction and processing of visual features at multiple dimensions (e.g. color, scale, orientation, motion). Despite the existence of some fast techniques for multi-scale feature extraction and processing, the amount of information to process is too large for the real-time implementation of generic visual attention architectures in full resolution images.

A built-in mechanism to reduce computational load is provided by the foveal structure of the eyes (see Chapter 2). A foveal system inspects just a small part of the scene at each time step, and shifts the gaze direction in order to observe other locations. Instead of a fully parallel interpretation of the scene, foveal systems employ a sequential process, trading off computational resources by gaze shifts (time). The location of new regions to attend are determined by **selective attention mechanisms**. A *overt shift of attention* is generated if the visual system decides to move the eyes to selected locations. However, the visual system can also focus attention in a particular location of the scene without moving the eyes (*covert shift of attention*).

Feature extraction in foveal images is a non-trivial issue due the the space variant nature of the representation. Some methods have been proposed for feature extraction in log-polar images. For instance, [68] uses a supervised learning mechanism to represent low-level features (blobs, edges) in log-polar images, for posterior detection. In [153], a set of space-variant filters is used to detect edge points in log-polar images. In general, the image analysis elements must be space-variant to cope with the foveal geometry. This fact severely limits the usage of existing fast feature extraction methods in foveal images.

Another problem related to feature extraction in space-variant representations is that part of the image spatial frequency content is destroyed in the foveation process. For example, a large object consisting of an high spatial frequency luminance pattern in the periphery of the visual field, may simply disappear after transformed to a foveal image (an example of this effect is presented latter). Henceforth, we propose that low-level feature extraction should be performed in the original domain, before image foveation. In fact, in biological vision systems, a large amount of low-level feature extraction is performed directly in retinal coordinates. It is widely known that linear retinal ganglion cells extract luminance contrast features at several scales. Also, [47] shows that some non-linear retinal ganglion cells respond to the local frequency content of the image, by computing the average absolute response of contrast features at several scales.

In the first part of this section we adapt the saliency computation mechanism of [81] to the log-polar space. Because saliency computation involves an iterative relaxation procedure, performing the computations in log-polar space is a means of significantly improving computational efficiency. In the second part of the section, we show why features should be extracted before the foveation process, and propose the computation of spatial-frequency features with units resembling non-linear retinal ganglion cells.

6.2.1 Saliency Maps in Log-Polar Images

Here we will show the viability of using the log-polar representation to compute the saliency of features in the visual field. We will use the saliency computation technique proposed by [81] and show that performing the iterative computation of conspicuity in the log-polar space, leads to significant computation time reduction, with a small penalty in the quality of the results (saliency is slightly biased toward the center of the image). The architecture

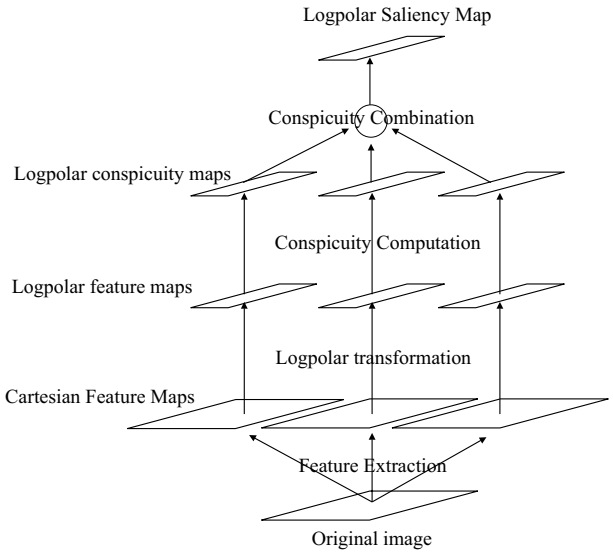


Figure 6.4: The proposed attentional architecture for log-polar saliency detection

for saliency computation is composed by a first phase of feature extraction in the original cartesian domain (this will be further justified latter), a second phase of feature foveation and conspicuity computation in log-polar space, and a final phase for the composition of conspicuity into a saliency map in log-polar coordinates (see Fig. 6.4). To abstract the feature extraction process, we first illustrate the idea with luminance features, that are independent of imaging geometry. Let $I(x, y)$ represent the original image. Positive and negative luminance features are defined as:

$$\begin{cases} L^+(x, y) = I(x, y) \\ L^-(x, y) = \max I(x, y) - I(x, y) \end{cases} \tag{6.1}$$

Consider the particular example given in Fig. 6.5, where saliency is computed in cartesian space as in [81]. Luminance features are repeatedly convolved with center-surround operators composed by Difference-Of-Gaussians at 4 different scales, resulting in 8 conspicuity maps. We have fixed the number of iterations to 8. All the conspicuity maps are added to generate the global saliency map. In Fig. 6.6, the same is shown for the log-polar space. The cartesian images have 380×380 pixels, while the log-polar images have 40×80 pixels. Center-surround operations are performed at only 3 levels, with smaller scales, because log-polar images are significantly smaller than the corresponding cartesian images. Consequently, the log-polar space reduces about 60 times the computational load. By visual comparison, the most salient points in both representations are similar, although in the log-polar case, the global ordering is biased toward objects near the center of the image. However we should notice that the purpose of attentional mechanisms in foveal systems is to select points of interest for overtly shifting the gaze direction. Thus, in real situations, the actual search order will be different from the ordering presented in the figures.

6.2.2 Feature Extraction in Cartesian Space

Many saliency computation mechanisms rely on the detection of regions with high frequency content, e.g. corners [75], edges [31], curves [4] and local image entropy [85]. However, foveation uses low bandwidth RF shapes in the periphery of the visual field,

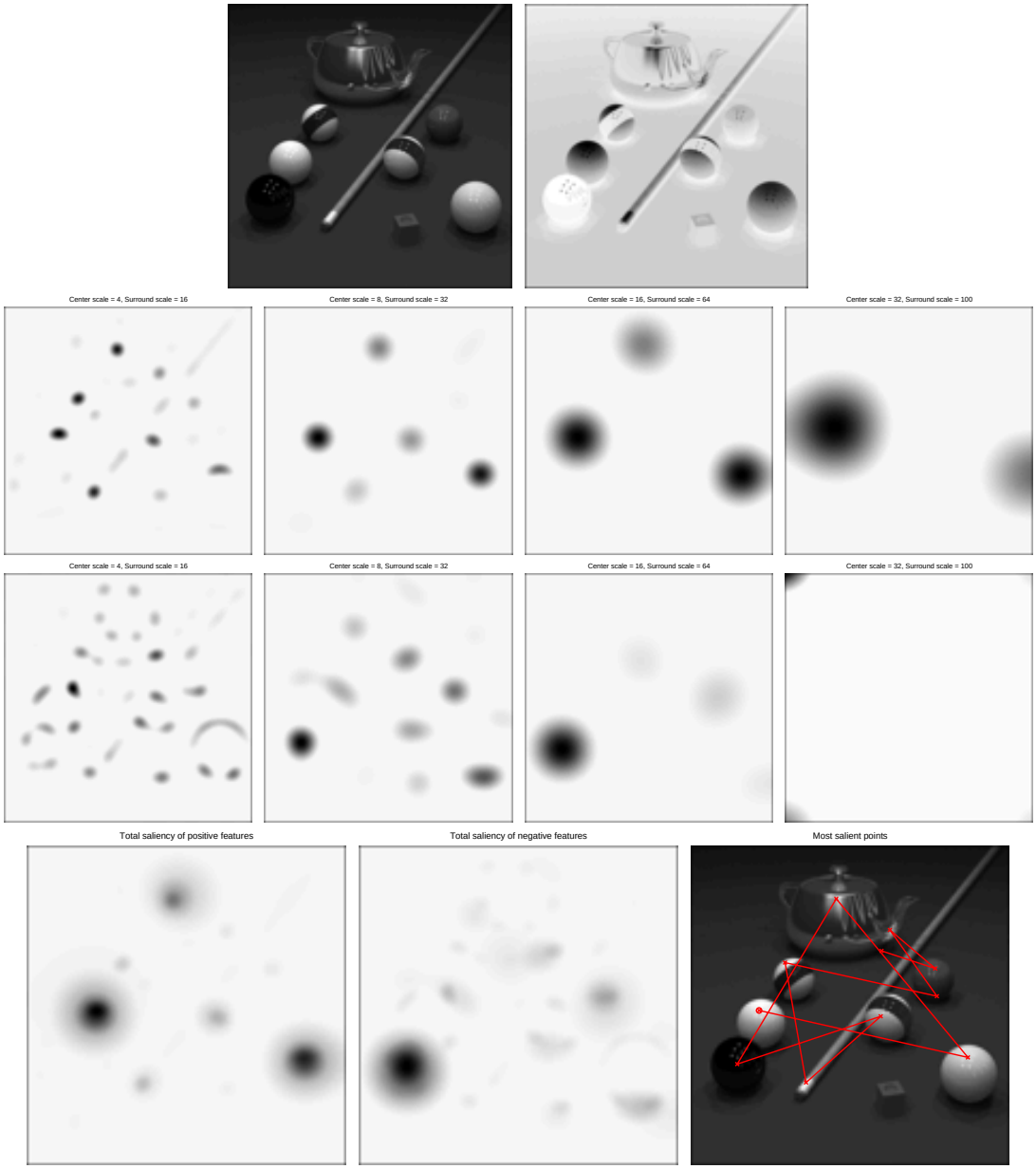


Figure 6.5: Computation of luminance saliency in cartesian space. Illustration for a particular test image. From top to bottom, left to right we have: positive luminance; negative luminance; positive conspicuity with center-surround scales 4-16, 8-32, 16-64, 32-128; negative conspicuity with center-surround scales 4-16, 8-32, 16-64, 32-128; total positive conspicuity; total negative conspicuity and; the most salient points plotted in decreasing saliency order. The most salient point is marked with a circle.

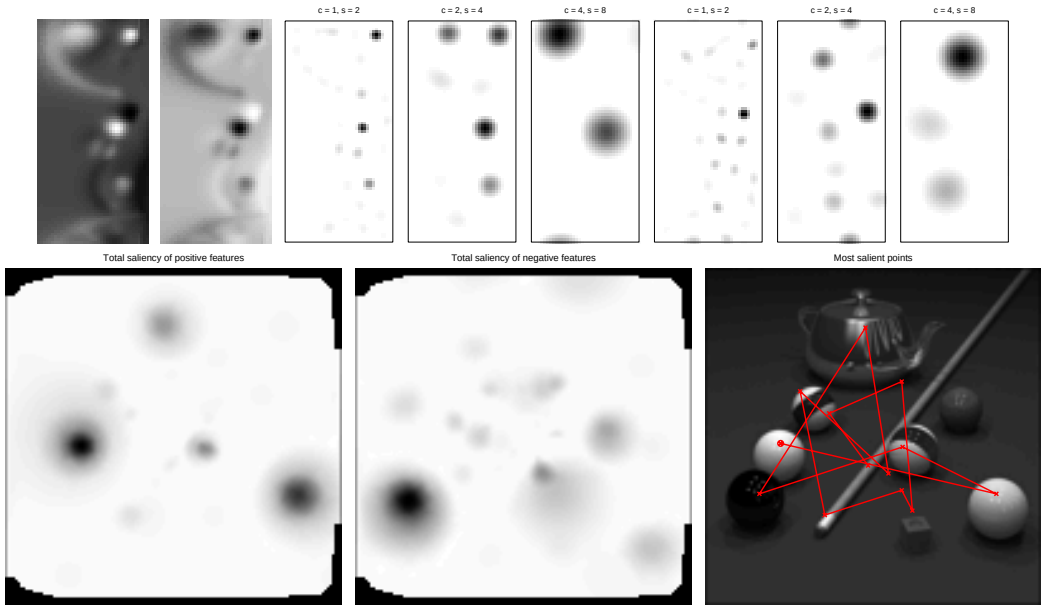


Figure 6.6: Computation of luminance saliency in log-polar space. From top to bottom, left to right we have: log-polar positive luminance; log-polar negative luminance; positive conspicuity with center-surround scales 1-2, 2-4, 4-8; negative conspicuity at scales 1-2, 2-4, 4-8; retinal representation of total positive conspicuity; total negative conspicuity and; the most salient points plotted in decreasing order of saliency. The most salient point is marked with a circle.

that irreversibly destroy high frequency information.

Let us consider the test image shown in Fig. 6.7, composed by textured patterns of different spatial frequencies. Since the average gray level of the patterns is equal to background luminance, peripheral (large) receptive fields average out the high-frequency patterns and output the background level. Thus, important information for attentional control is lost in the foveation process.

Our approach to the problem consists in first performing the feature extraction process and then foveating the resulting feature maps. In the human visual system it is known that, although the density of cones (photoreceptors for high acuity vision) decreases very rapidly, the density of rods (photoreceptors very sensitive to contrast and able to detect a single photon of light) is maximal at about 20 degrees and decrease slowly to the periphery [21]. This suggests that high acuity tasks and attentional tasks require different hardware and sampling strategies, in accordance with our proposal. The proposed low-level feature extraction process is motivated by the existence of non-linear ganglion cells in the retina [47]. These cells have large receptive fields and accumulate the output modulus of small linear ganglion cells, as schematically represented in Fig. 6.8.

Let us mathematically model the linear ganglion cell as Laplacian filters $l_{(c,s)}$, with center scale c and surround scale s, and the averaging operation as a Gaussian filter g_a, with scale $a > s$. Then, the feature map representing the local image frequency "content" at scale s, and corresponding to the output of the non-linear ganglion cells, can be computed by:

$$F_s = g_a * |l_{(c,s)} * I| \tag{6.2}$$

where I is the original luminance image.

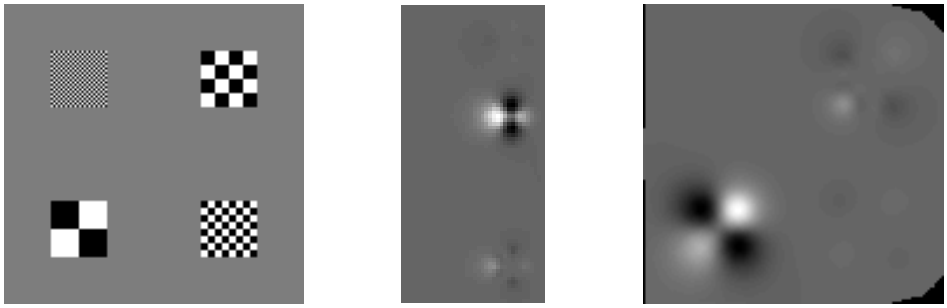


Figure 6.7: Test pattern mapped to log-polar coordinates (left) and mapped back to cartesian coordinates (right). Notice that high frequency patterns in the periphery become invisible in the foveal image, while lower frequency patterns can still be detected.

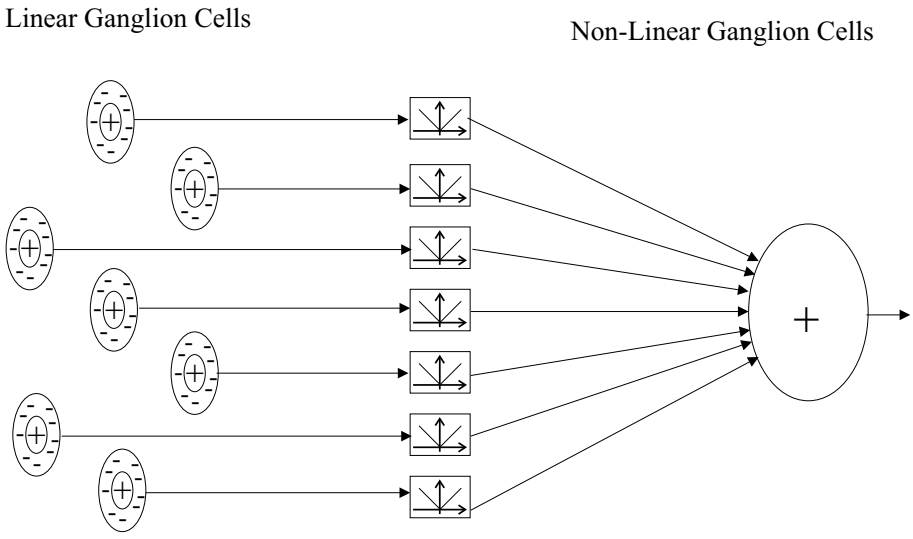


Figure 6.8: Non-linear ganglion cells extract the local high frequency content in the images, thus responding strongly to edges and textured patches of arbitrary directions.

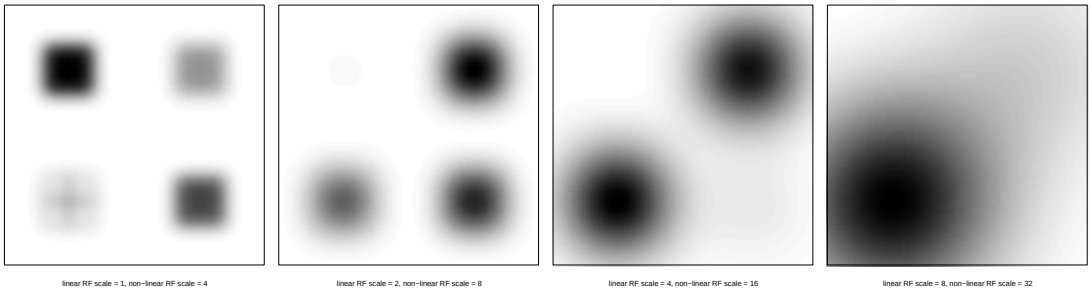


Figure 6.9: Spatial-frequency feature maps. From left to right spatial-frequency tuning of receptive fields decrease. In every case, the non-linear receptive field size is 4 times the surround scale of the linear receptive fields.

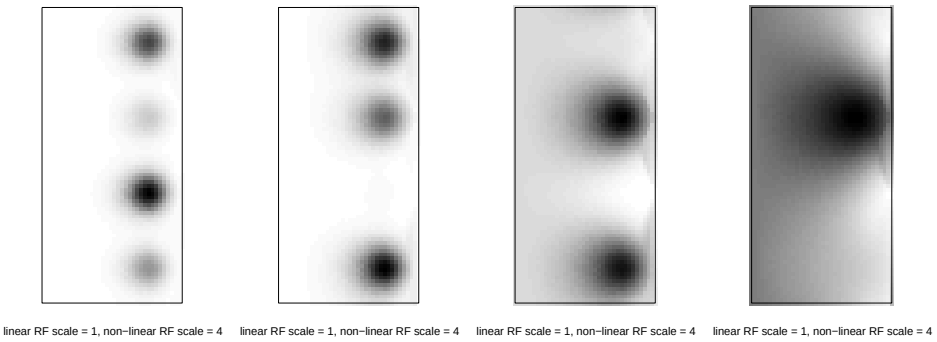


Figure 6.10: Spatial-frequency feature maps in log-polar space.

Consider again the test image pattern in Fig. 6.7. The local non-linear feature maps are represented in Fig. 6.9, for center-surround scales $(c, s) = \{(0.5, 1); (1, 2); (2, 4); (4, 8)\}$ and fixed parameter $a = 4 \times s$.

After the feature maps have been computed, they (and their negatives) are converted to foveal images for further processing. The log-polar positive spatial-frequency feature maps are shown in Fig. 6.10. The computation of conspicuity maps is performed in foveal coordinates, according to the architecture presented in Fig. 6.4. Again, comparing the cartesian and log-polar spaces (see Fig. 6.11), we can observe that the most salient regions are correctly identified in both cases, though the order of saliency is different.

Another example of spatial-frequency saliency computation is shown in Fig. 6.12, for a negative saliency case. A high-frequency pattern composed of random white noise, contains a low-frequency inner region. In both geometries, the saliency computation mechanism is able to identify the positive an negative salient points in a similar fashion.

An interesting aspect of our approach is that, despite the high frequency patterns can be identified in foveal images, individual elements constituting these patters can not be spatially localized within the pattern group. This is similar to what happens with humans when reading – we can detect the existence of texture corresponding to letters and words in peripheral locations of the visual field, but can not recognize or count individual items. In [80], the human capability of individuating items in the visual field was tested. Individuation is a property required to encode object location, track its position or count objects in sets of 4 or more items [80]. Fig. 6.13 shows the difference between "seeing" an "individuating" objects. When fixating at the cross, one can "see" vertical bars to the right, but it is not possible to count the bars. Intriligator and Cavanagh [80] have made a

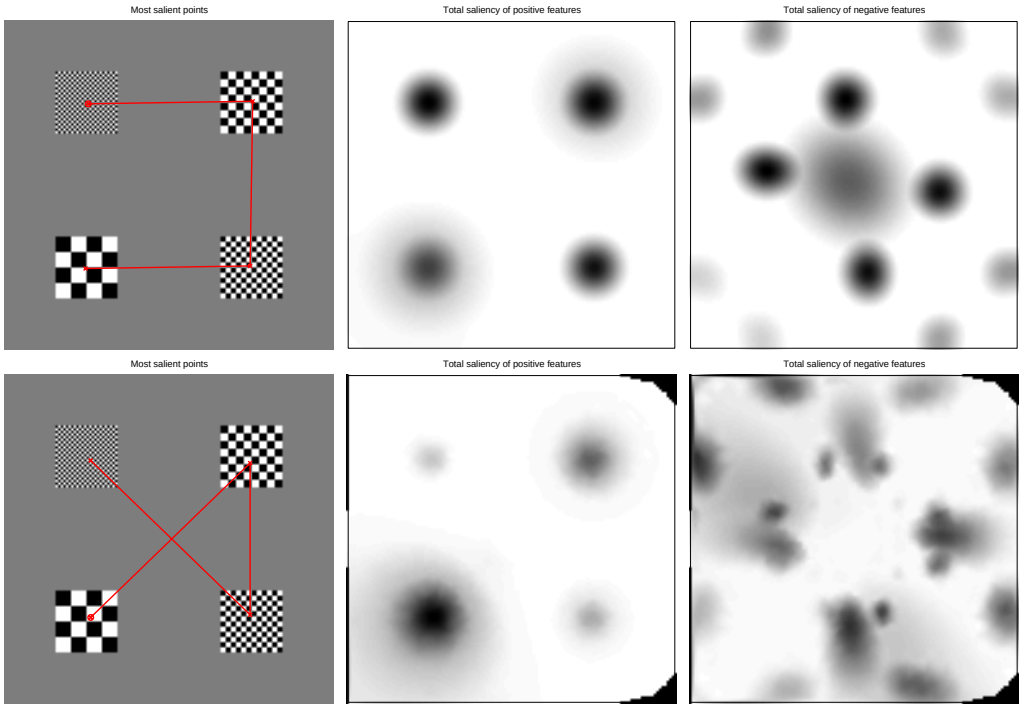


Figure 6.11: Computation of saliency from spatial-frequency features. The top and bottom rows correspond, respectively, to the cartesian and log-polar cases. From left to right we show the most salient points in decreasing order of saliency, and the positive and negative feature conspicuities.

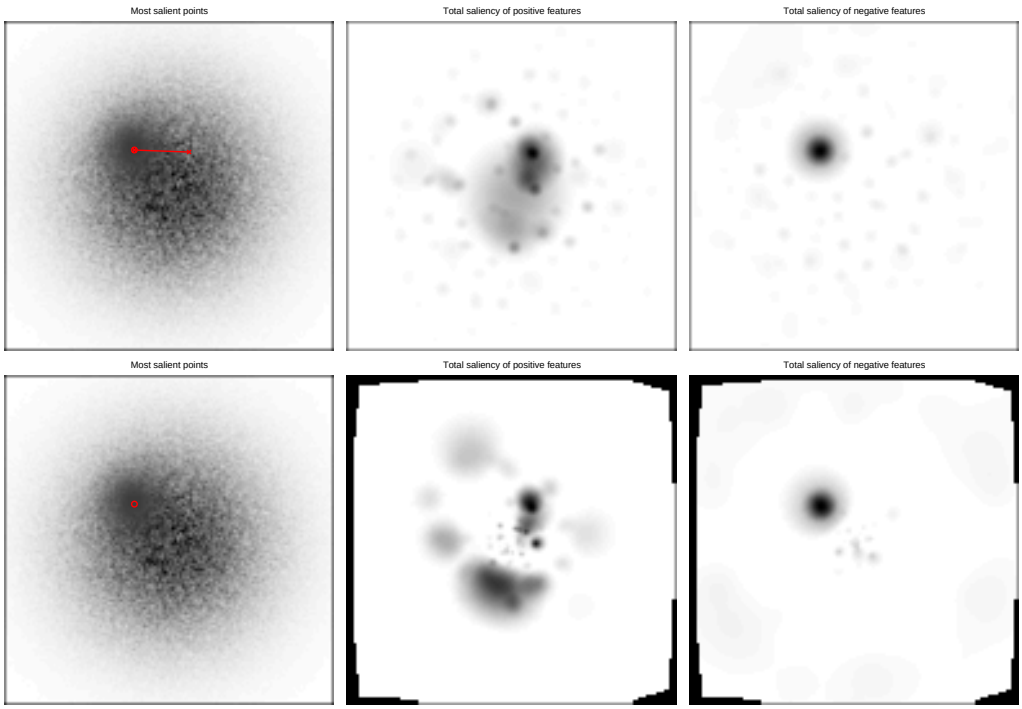


Figure 6.12: Example containing spatial-frequency negative saliency. The top and bottom rows correspond, respectively, to the cartesian and log-polar cases.

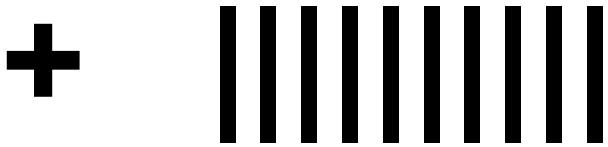


Figure 6.13: Seeing *vs* individuating. Adapted form [80].

series of experiments with observers looking at the center of a display with several objects with varying distances from each other. Observers where not allowed to move their eyes and only covert attentional shifts were permitted. They found that there is an attentional resolution limit, much coarser than visual acuity, such that objects spaced more finely cannot be selected individually for further processing. However, some properties of the objects can still be discriminated, like motion and texture. For instance, when observing a group of moving dots whose spacing is lower than the resolution limit for that eccentricity, one may not be able to count the dots but can resolve their shape and motion. These results support our approach, where certain features are extracted at higher resolution than the one used for higher level tasks.

Since feature extraction is performed in the cartesian images, our proposal involves a high computational cost in the initial processing phase. However, low-level feature extraction has linear complexity with regard to image size. Costly operations, like the saliency computation, are processed with much smaller foveal images. This strategy can be extended to other visual processes of high order complexity, like object segmentation or recognition, where the fact of having reduced image sizes results in more significant computational improvements.

6.3 Directional Features

In the previous section we have used local frequency features to illustrate the caveats of conventional attentional mechanisms in foveal images. These features are useful to attend and discriminate non-oriented visual information. However, the orientation of visual structures has a great importance in the representation of objects. An obvious example was given in Fig. 1.8 where bars and crosses can only be distinguished attending to the differences of orientation on its constituting parts.

We have seen in Section 6.1 that existing attentional models propose early image representations based on multiscale decompositions of different features like color, intensity and orientation [81]. The highest computational load is usually involved in the extraction of orientation features. In this section we describe a fast algorithm for the computation of multi-scale directional features, that outperforms state-of-the-art algorithms. The method is based on Gabor filters, that extract information related to the local orientation, scale and frequency of image structures. The oriented features will then be used for searching for generic points of interest (bottom-up saliency), and for guided search of simple visual items (top-down modulation).

Both information theoretic and biological arguments have motivated the wide use of Gabor filters for image analysis. Gabor [60] has discovered that Gaussian-modulated complex exponentials provide the best trade-off between spatial and frequency resolution, allowing simultaneously good localization and description of signal structures. Neurophysiological studies have shown that visual cortex simple cells are well modeled by families of 2D Gabor functions [45]. These facts raised considerable interest because they suggests

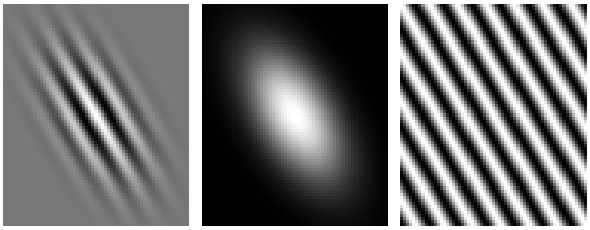


Figure 6.14: A Gabor function (left) resulting from the product of a Gaussian envelope (middle: $\alpha = 30$ degrees, $\sigma_1 = 8$ pixels, $\sigma_2 = 16$ pixels), by a complex exponential carrier (right: $\lambda = 8$ pixels, $\theta = 30$ degrees). Only real parts are shown.

that neuronal structures may indeed develop toward optimal information coding.

Two-dimensional Gabor filters are very utilized on image analysis systems for applications such as image compression [57], texture classification [121], image segmentation [140] and motion analysis [27]. Fast algorithms for Gabor filtering exist [163, 107], and take advantage of the separability of isotropic Gabor functions in the horizontal and vertical directions. However multi-scale/multi-feature representations require analysis with many Gabor wavelets, tuned to different orientations, scales and frequencies, hence any improvement in computational efficiency has significant effects in the performance of the algorithms. We have developed a fast algorithm for isotropic Gabor filtering that outperforms current implementations, based on three facts: Gabor functions can be decomposed in convolutions with Gaussians and multiplications by complex exponentials; isotropic Gaussian filtering can be implemented by separable 1D horizontal/vertical convolutions; appropriate boundary conditions are derived to deal with boundary effects without having to extend image borders. Our proposal reduces to about one half the number of required operations with respect to state-of-the-art approaches.

The section is organized as follows. First we review some of the underlying theory of Gabor filtering. Then we briefly describe the proposed algorithm for fast Gabor image analysis (a detailed description is provided in Appendix E) and compare our approach with state-of-the-art methods. Finally we show how Gabor features can be used for the saliency computation of directional information.

6.3.1 Gabor Wavelets for Image Analysis

Gabor functions are defined by the multiplication of a complex exponential function (the carrier) and a Gaussian function (the envelope). Gabor wavelets are Gabor functions with zero-mean values. The Gabor wavelet satisfies the admissibility condition for multiresolution image analysis, [99] and, apart from a scale factor, is equivalent to the Morlet Wavelet.

Image analysis by convolution with Gabor wavelets has been extensively studied in the literature, and provides a method to estimate the oriented local frequency content of image regions. In practical terms, the convolution output modulus will be high whenever the local image structure is similar to the Gabor wavelet shape, in terms of scale (σ), wavelength (λ), and orientation (θ). Figure 6.14 shows the real part of a two dimensional Gabor function, the corresponding Gaussian envelope $\mathbf{w}_\sigma$ and carrier $\mathbf{c}_{\lambda,\theta}$.

When the Gaussian envelope is isotropic ($\sigma_1 = \sigma_2 = \sigma$), image convolution with Gaussian and Gabor functions can be implemented efficiently by one-dimensional sequential convolutions in the horizontal and vertical directions. In another hand, fast one-dimensional infinite impulse response filters, approximating Gaussian and Gabor func-

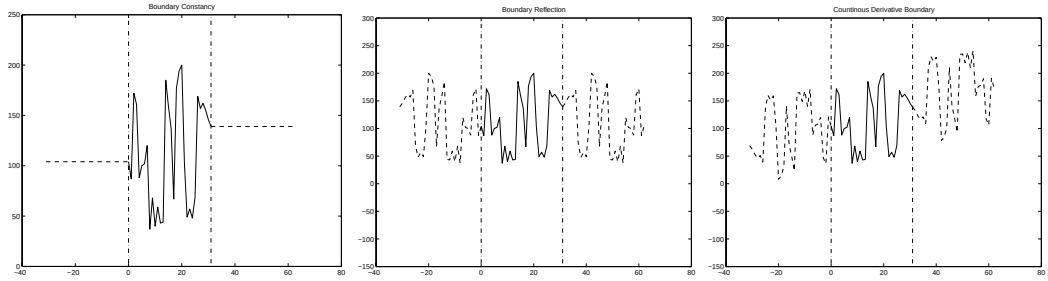


Figure 6.15: Boundary extension methods to avoid transient responses in the signal limits. From left to right, boundary extension methods are: constant value, reflection and continuous derivative. Dashed lines represent the extended parts of the signal.

tions, have been developed recently [162, 163]. With these techniques it is possible to implement 2D isotropic Gaussian and Gabor filtering with 26 and 108 operations per pixel, respectively.

6.3.2 Fast Implementation

The first improvement we propose to Gabor filtering involves rewriting the Gabor wavelets as multiplications with complex exponentials and convolutions with Gaussian functions. The motivation for this decomposition is the fact that state-of-the-art Gaussian convolution is more efficient than Gabor convolution, and compensates the extra multiplications with complex exponentials. In Appendix E, we describe in detail the approach and show that this decomposition allows 35% reduction in computational complexity, with respect to the work in [163]. We focus on the isotropic case, where vertical/horizontal separable implementations exist for Gaussian filtering, but the method can also be applied to the anisotropic case. In fact, a separable implementation of anisotropic Gaussian filtering has recently been proposed, consisting in two 1D convolutions performed in non-orthogonal directions [65].

The second improvement is related to the existence of redundant computations when multiple orientations and wavelengths are computed on a single scale. If, for example, 4 orientations and 2 wavelengths are used, the total number of operations can be reduced by 42%.

Another important contributions is the derivation of appropriate initial conditions for the initialization of the filtering operations. Without appropriate initial conditions, any IIR filtering operation may present undesirable transient responses near the signal boundaries. In the case of Gaussian and Gabor filtering, the transients will be larger for higher scale filters. These effects are undesired because they generate artificial responses corresponding to step edges in the boundary of the image. In the domain of signal processing, several approaches are common to address this problem, often involving the extension of the signal boundary and making certain assumptions on signal properties, for instance piecewise constancy, continuity in the derivative or reflexion (see Fig. 6.15). To allow for complete transient extinction, the boundary should be extended by more than 3 times the scale of the filter. Thus for large scale filters this would imply a significant increase in computation time. A better solution is to derive adequate conditions to initialize the filter state at the boundaries. This way we avoid explicitly running the filter at the extended boundaries. Our case however, consists in cascaded passes of forward, backward, horizontal and vertical filtering, requiring a careful analysis of the initial condition in each

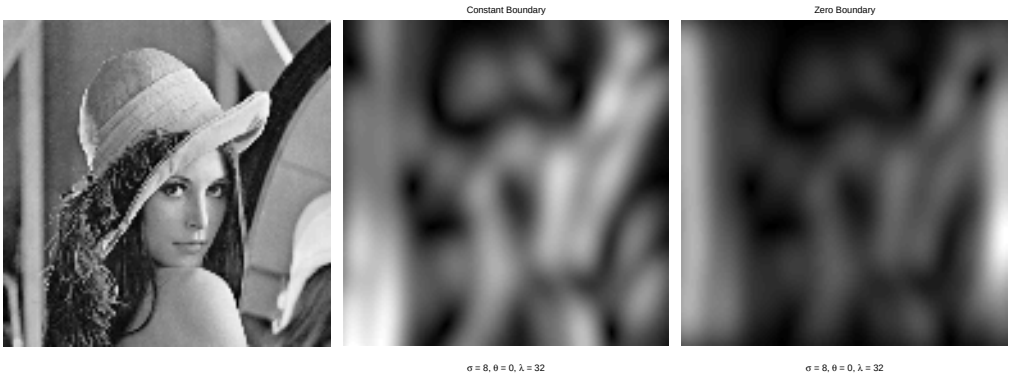


Figure 6.16: Boundary effects on Gabor filtering. The image in the left is filtered with a Gabor wavelet tuned for vertical edges. If appropriate initial conditions are not used, boundary effects are visible at vertical image borders (right). With the initial conditions provided by our method, boundary effects vanish (center).

pass. Also, part of the Gaussian filtering operations are made on complex exponentially modulated images, whose boundary extension assumptions are different than usual. The full derivation of the initial conditions is done in Appendix E.

Fig. 6.16 shows the result of image convolution with a Gabor wavelet tuned for vertical edge detection. The initial conditions computed with our method are compared with zero initial conditions. Notice that without adequate initial conditions, spurious responses arise in the image vertical boundaries.

6.3.3 Biologically Plausible Gabor Wavelets

It has been found that simple and complex cells in the primary visual cortex have receptive fields that resemble Gabor functions of particular combinations and ranges of parameters (see [93] for a review). In particular the half-amplitude frequency bandwidth (β) of the Gabor filters range from 0.5 to 2.5 octaves. This parameter depends only on the values of scale and wavelength, as follows. In the radial direction, the frequency response of an isotropic Gabor function is given by the expression:

$$\widetilde{\mathbf{g}}(|\Omega|) = e^{-\frac{1}{2}\sigma^2(|\Omega|-\frac{2\pi}{\lambda})^2} \tag{6.3}$$

Half-amplitude points are, in octaves:

$$\Omega_{1,2} = \frac{2\pi}{\lambda} \pm \frac{\sqrt{2\log(2)}}{\sigma}$$

and, the half-amplitude bandwidth is given by:

$$\beta = \log_2 \frac{2\pi\sigma + \lambda\sqrt{2\log(2)}}{2\pi\sigma - \lambda\sqrt{2\log(2)}} \tag{6.4}$$

Let us consider the first 4 scales of a dyadic decomposition, $\sigma = \{1, 2, 4, 8\}$, and the wavelength values $\lambda = \{3.7, 7.4, 14.8, 29.6\}$. The half-amplitude bandwidth values of each scale/wavelength combination are show in Table 6.1. We may choose to use wavelets whose half-frequency bandwidth is approximately within biologically plausible values (bold entries in Table 6.1). If all wavelets satisfying the former criterion are chosen, in the present

	$\lambda = 3.7$	7.4	14.8	29.6
$\sigma = 1$	**2.47 (E)**	-	-	-
2	**1.04 (ST)**	**2.47 (E)**	-	-
4	**0.51 (LT)**	**1.04 (ST)**	**2.47 (E)**	-
8	0.26	**0.51 (LT)**	**1.04 (ST)**	**2.47 (E)**

Table 6.1: Half-amplitude bandwidth values (in octaves) for each pair scale/wavelength. Bold face entries are within biologically plausible range. Symbols in parenthesis indicate the appearance of the Gabor wavelet: E – "edge" wavelet, ST – "small texture" wavelet, LT – "large texture" wavelet.

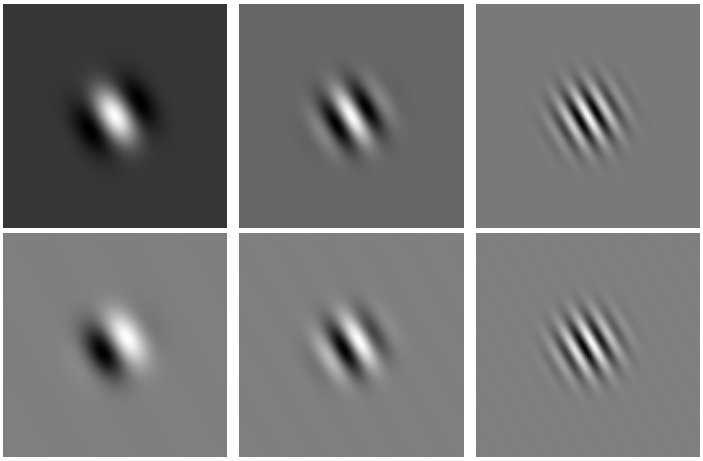


Figure 6.17: Real (top) and imaginary (bottom) parts of: (left) an "edge" (**E**) Gabor wavelet with half-frequency bandwidth in octaves $\beta = 2.47$; (center) a "small texture" (**ST**) wavelet having $\beta = 1.04$; (right) a "large texture" (**LT**) wavelet with $\beta = 0.51$

example we have a total number of 36 wavelets. This choice corresponds to the wavelet shapes shown in Fig. 6.17, and resemble units tuned to edges, small texture patches and large texture patches, respectively. Roughly speaking, "edge" wavelets will respond equally well in image locations corresponding to edges and textures with appropriate scale and orientation. "Texture" wavelets will respond better in textured areas with the matched direction and wavelength. Jointly using both types of features it is possible to distinguish between edges and textured regions [115].

Computationally, our implementation to Gabor filtering requires the following number of operations (see Appendix E):

$$26 \times S + 2 \times C + 60 \times K \tag{6.5}$$

where S is the number of scales to compute, C is the number of carriers (orientation-wavelengths) and K is the total number of wavelets. In the exemplified decomposition, we have $S = 4$, $C = 16$, and $K = 36$, which leads to 2296 operations per pixel. In a processor at 2.66Ghz, performing this decomposition on 128×128 grayscale images, takes about 0.150 seconds. For the sake of comparison, if the state-of-the art IIR filters of [162] and [163] where used, the number of required operations per pixel would be $26 \times S + 110 \times K$, increasing to 4064. Figure 6.18 shows the output modulus of the proposed filter, applied to a common test image.

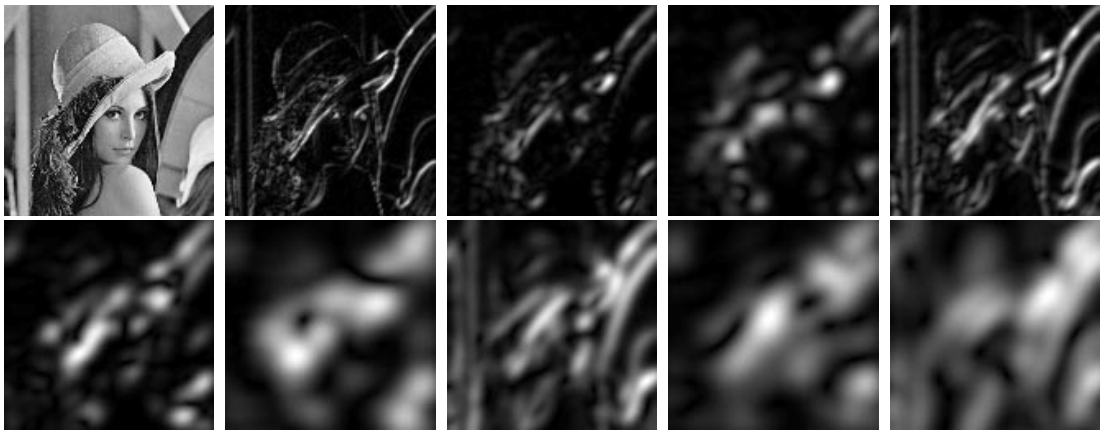


Figure 6.18: Modulus of the Gabor wavelet decomposition (for 45 degrees orientation only) applied to the "Lenna" test image (shown on the top-left). Image contrast has been normalized for visualization purposes. From top-left to bottom-right, the wavelet parameters (σ, λ) are, respectively: (1,3.7), (2,3.7), (4,3.7), (2,7.4), (4,7.4), (8,7.4), (4,14.8), (8,14.8), (8,29.6).

6.3.4 Foveal Saliency of Directional Features

In this section we evaluate the performance of the proposed log-polar saliency computation mechanism, applied to local orientation features derived from Gabor wavelets. The rationale is to detect points of local orientation differing from its neighbors. Such points represent potentially interesting locations where to direct attention.

Following the architecture presented in Section 6.2.1, the first step of saliency computations involves extracting features on the original cartesian images. As in the case of spatial-frequency feature extraction, we propose a linear/non-linear receptive field structure. Linear receptive fields compute Gabor features at certain scales, wavelengths and orientations. The non-linear receptive field computes the absolute value of the Gabor features and accumulates their value in a larger spatial extent. This structure is illustrated in Fig. 6.19, where the real and imaginary parts of the Gabor wavelets are shows as separate receptive fields. This model is very close to some types of complex cells present on the visual cortex of mammals (areas V1 and V2) [115]. However, in the retina, cells of such structure have not been reported in the literature.

A new rule for orientation contrast

Mathematically, the non-linear Gabor features maps are computed by:

$$D_{\sigma,\theta,\phi,a} = g_a * |\gamma_{\sigma,\theta,\phi} * I| \tag{6.6}$$

where I is the original luminance image, $\gamma_{\sigma,\theta,\phi}$ is a complex Gabor wavelet, and g_a is a Gaussian filter with scale $a > \sigma$.

With respect to the luminance and spatial frequency features, positive and negative orientation feature maps are derived in a significantly different way:

$$\begin{cases} D^{+}_{\sigma,\theta_i,\phi,c} = D_{\sigma,\theta_i,\phi,c} * \left(D_{\sigma,\theta_i,\phi,c} - D_{\sigma,\theta_i,\phi,s} + \sum_{j \neq i} D_{\sigma,\theta_j,\phi,s} \right) \\ D^{-}_{\sigma,\theta_i,\phi,c} = D_{\sigma,\theta_i,\phi,c} * \left(D_{\sigma,\theta_i,\phi,c} - \sum_{j \neq i} D_{\sigma,\theta_i,\phi,c} + \sum_{j \neq i} D_{\sigma,\theta_j,\phi,s} \right) \end{cases} \tag{6.7}$$

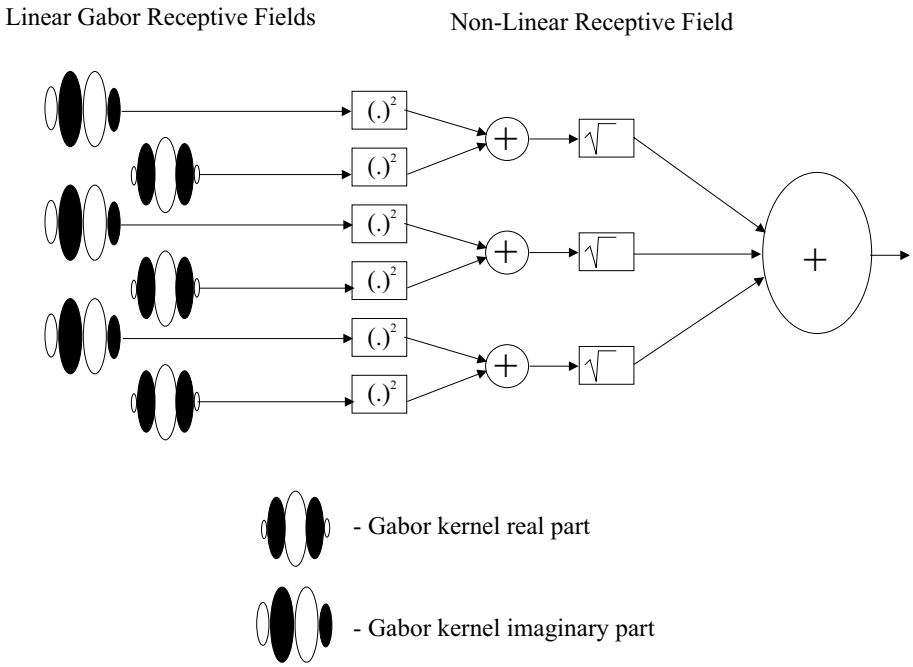


Figure 6.19: Gabor features are extracted by a non-linear receptive field structure that accumulates the modulus of linear Gabor filters over extended spatial regions. Such structure respond strongly to edges and textured patches of particular orientations, regardless of their exact position and phase.

where $s > c$. The above formulas are motivated by the following reasons:

- The positive map for orientation θ should be high in locations where features with orientation θ exist, and the surroundings do not contain features with orientation θ but must contain other orientations.
- The negative map for orientation θ should be high in locations where features with orientation θ exist, but there are different orientations at surrounding locations not present in the central location.

This strategy aims to reduce high saliency values in points surrounded by featureless areas, as would happen if the rule for luminance and spatial-frequency features was used. Positive and negative orientation feature maps, obtained by this method, are displayed in Fig. 6.20, for a particular test image.

Cartesian *vs* Logpolar Saliency

Here we compare the results of applying the iterative saliency computation method in cartesian and log-polar geometries. Although feature maps are differently derived with respect to spatial-frequency or luminance features, the saliency computation method is analogous:

1. values of the positive and negative features maps are normalized to a fixed range;
2. each normalized feature map is repeatedly convolved (8 times) with center-surround operators of scales $(c, s) = (\sigma, 4\sigma)$, in the cartesian case, and $(c, s) = (\sigma/2, \sigma)$, in the log-polar case.

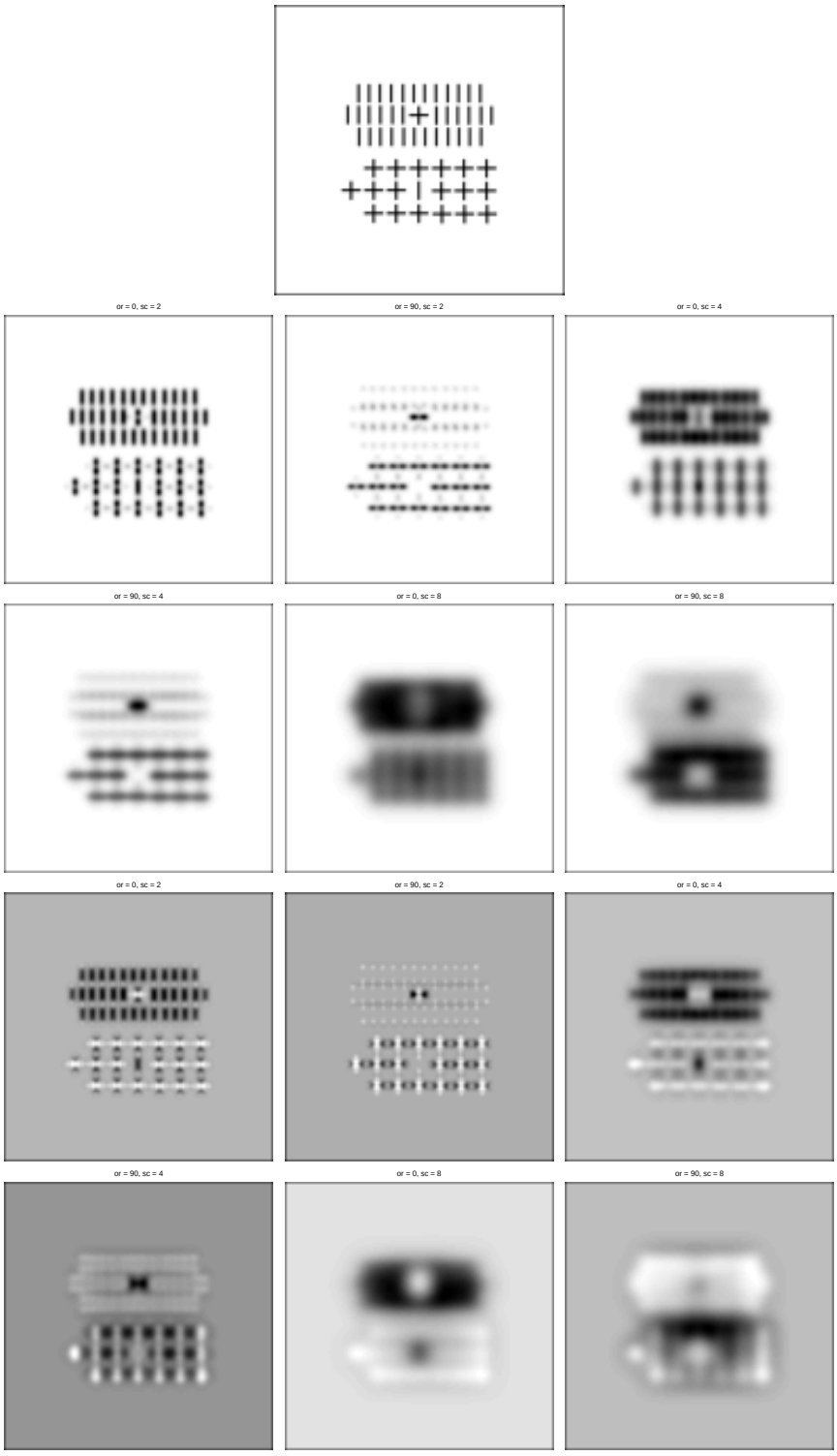


Figure 6.20: Orientation features extracted from the test image shown on top. The second and third rows show the positive orientation features of scale and orientation $(\sigma, \theta) = \{(2, 0^0), (2, 90^0), (4, 0^0), (4, 90^0), (8, 0^0), (8, 90^0)\}$. The fourth and fifth rows show the negative orientation features. In every case, the wavelength value is $\lambda = 3.7\sigma$ and the non linear-receptive fields center and surround scales are $c = 2\sigma$, $s = 8\sigma$.

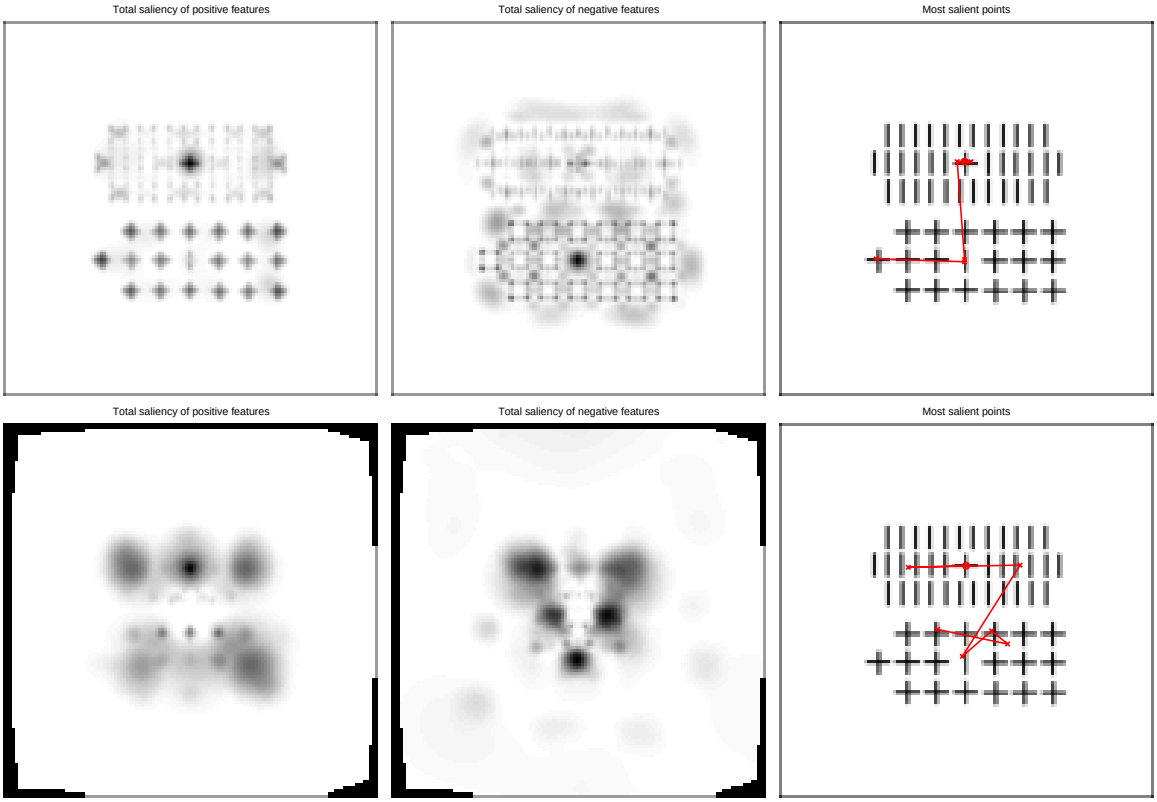


Figure 6.21: Orientation saliency computation in cartesian (top) and log-polar spaces (bottom). The left column shows the total saliency of positive features. In the middle column it is show the total negative saliency. The most salient points, sorted in descending saliency order, are shown in the right column.

3. saliency is computed from the conspicuity maps by simple summation.

We illustrate the results with a cartesian test pattern with 512×512 pixels. The corresponding log-polar map has 80×42 pixels. Hence, if saliency computations are performed in log-polar space we obtain about 80 times computational savings. Results are shown in Fig. 6.21 and it can be observed that both the bar-among-crosses and the cross-among-bars are within the most salient points in both geometries. Also, its is visible that the log-polar geometry naturally privileges points closer to the center of the images.

6.4 Bottom-up and Top-down Selective Attention

Random exploration of visual scenes by scanning points of high global saliency, is a means to build a sparse representation of the environment. At each scan, the visual system grabs relevant information of the observed item and summarizes its appearance by means of a symbol (if the system has recognition capabilities) or, more simply, by a template image. In some circumstances, an object may become more relevant for the task at hand, triggering a different behavior (e.g. tracking). Without prior task bias, all features contribute equally to the definition of saliency values.

A different search strategy is applied when the system should look for particular objects in the scene. Here, a pure data-driven strategy may select a high number of salient locations and drive the system to exhaustively search all salient items. Some guidance

can be provided by weighting differently the conspicuity maps, enhancing features that are more representative of the target object.

In this section we present some results of bottom-up and top-down selective attention in log-polar images. Experiments were made with natural images but the methodologies have not yet been applied to real-time scenarios. Though significant computational improvements are achieved by using foveal images, the system still lacks the ability of real-time performance. Also, further research must be devoted to the integration of the attentional mechanism with other components of the system. Anyway, we think that the results presented in this chapter illustrate the fundamental issues related to visual attention mechanisms and motivate future research work.

6.4.1 Bottom-up Saliency from Multiple Features

In our architecture, purely data-driven saliency is implemented by linearly combining all conspicuity maps with the same weight. In Fig. 6.22 we show some results of global saliency computation in a set of natural images, both using cartesian and log-polar representations. The images contain some salient objects that are equally well detected in both cases. The log-polar version tends to detect fewer blobs corresponding to the most salient large-scale objects, whereas the cartesian version tends to pick more small-scale salient points.

In terms of computational complexity, the log-polar saliency computation is 80 times faster than its cartesian counterpart, thus compensating the sparseness and blob like nature of detected points. We should notice again that selective attention is just a "filtering step" to remove unpromising parts of the scene and speed-up image scanning. At each gaze shift the system should inspect the objects more carefully according to its task, and recompute saliency values in the whole view field.

6.4.2 Top-down Saliency Biasing

We illustrate top-down saliency biasing with the search for eyes in face images. In normal circumstances (up-front faces), eyes can be well represented by blobs of small size and horizontal edges. Thus, if we bias saliency by weighting favorably these features, we expect to raise the relative importance of eyes in the global saliency map. The results shown in Fig. 6.23 illustrate this idea in a set of face images. We compare the case of full saliency, obtained summing all the conspicuity maps, and eye-biased saliency, obtained by summing only conspicuity maps with medium spatial frequencies and horizontal Gabor wavelets at all scales. Saliency computations are performed in log-polar space. In the figures, we show all local maxima above 10% of the global maximum. We can observe that top-down bias effectivelly reduces the number of maxima to search for, excluding salient points that are unlikely to represent eyes.

Obviously, more elaborate weighting strategies can further enhance the saliency of particular visual items in the images. In this experiment we just wanted to illustrate that even with very simple empirical rules, top-down saliency bias can be effective in reducing problem complexity. Also, other features, like color, can filter out additional visual regions not conforming to the representation of faces and eyes. We do not further explore this issue here, but it will be subject of future research efforts.

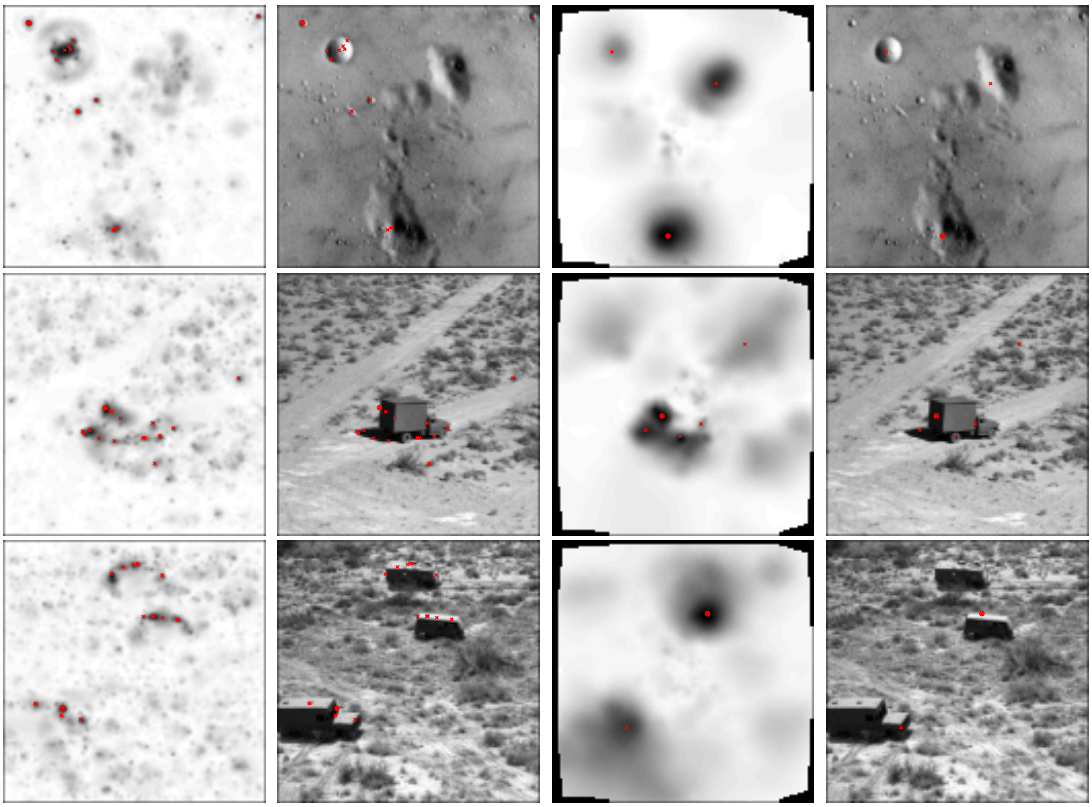


Figure 6.22: Global saliency computed in cartesian space (columns 1 and 2) and log-polar space (columns 3 and 4).

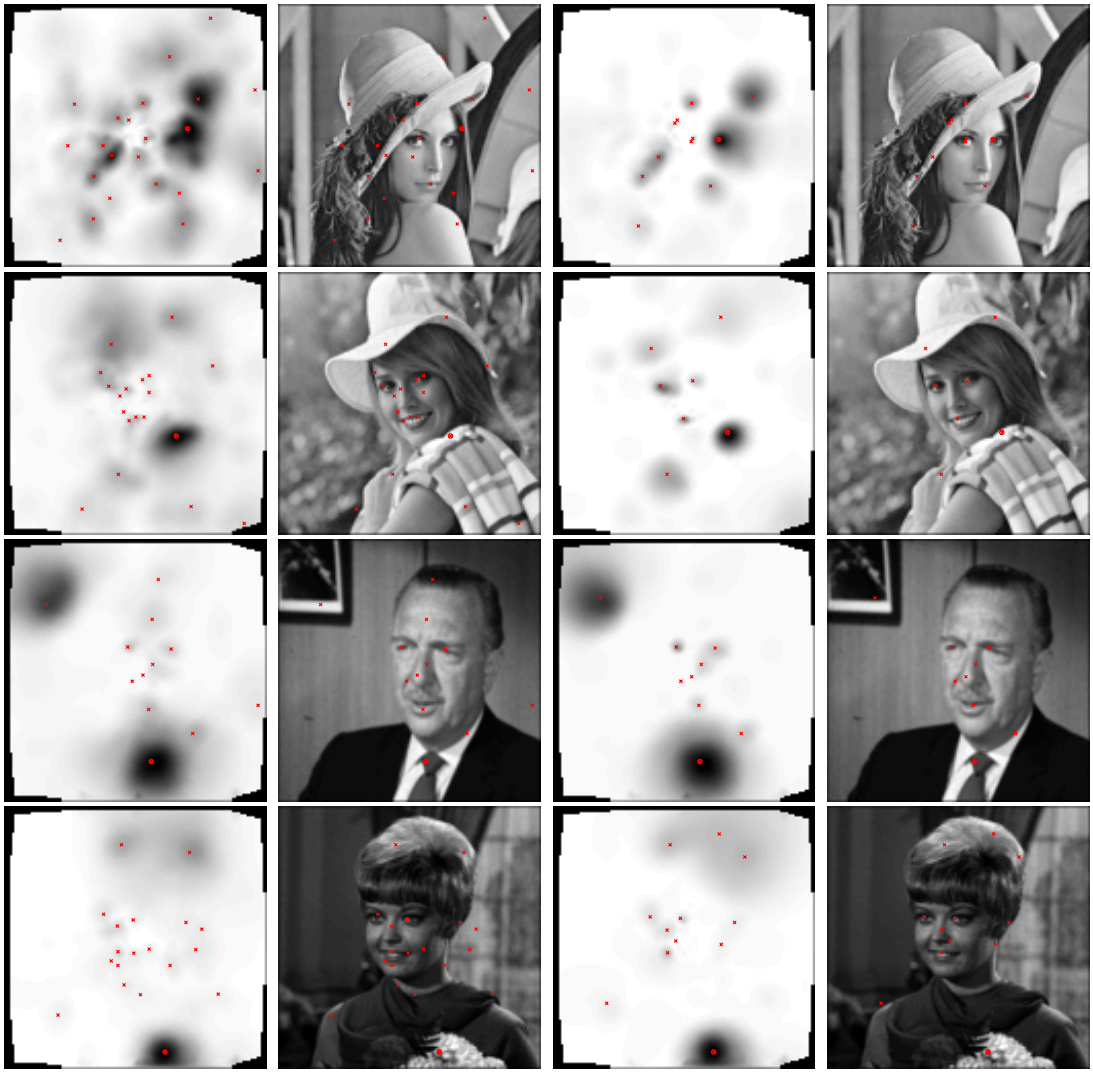


Figure 6.23: Global saliency (columns 1 and 2) and eye-biased saliency (columns 3 and 4) computed in log-polar space.

6.5 Final Remarks

In this chapter we have explored aspects related to visual attention mechanisms with potential application in the control of visual systems. In particular, we addressed the computation of image saliency in log-polar images and have shown that it achieves significant computational improvements with low performance loss with respect to the cartesian geometry. Saliency is essential in reducing the complexity of visual search, both for exploratory tasks (driven basically by bottom-up information) and goal directed tasks (involving top-down modulations).

Many aspects still require further research in order to use the attentional system in the visual control of binocular heads. In one hand, it is essential to define high-level control mechanisms that guide system behavior in terms of task specific actions and events. The decision of engaging on a random visual search in the scene or the search for particular objects is highly dependent of the particular agent goal. On the other hand there are still some visual capabilities that require further developments in order to integrate the attentional system in a working device. In the following chapter we point out some directions for future research that will address these issues.

Chapter 7

Conclusions

As stereo heads become more and more frequent in the bodies of humanoid robots, it is likely that active binocular vision will play a fundamental role in future robotics. This thesis has contributed, in several aspects, to support the application of active visual perception in the control of robot behavior.

One of the central issues of the thesis was the application of foveal vision in visual information processing. In one hand we have proposed a foveation methodology based on highly overlapping receptive fields, that reduces the amount of aliasing in the process. This aspect is often disregarded in conventional foveation methods. On the other hand, we have demonstrated the application and efficiency of foveal vision in three essential capabilities for robot visual perception : depth estimation, tracking and selective attention. The use of logpolar images reduces the hard computational needs of such capabilities, in the order of a logarithm factor. Additionally, the logpolar geometry is a natural focus of attention in the center of the visual field, favoring perceptual processes in tracking scenarios.

One exception to full foveal processing is given by our selective attention model, where we have proposed a cartesian implementation of low-level feature extraction before the foveation step. The motivation is the existence of non-linear ganglion cells in the human retina, that extract high-frequency spatial information before projecting to cortical areas. The aim is to preserve spatial-frequency content in the input visual flow that, otherwise, would be lost. Due to linear complexity on the initial feature extraction phase, this strategy does not involve a high performance penalty, and its influence in the selection of interest points is illustrated experimentally.

We have performed a series of improvements in the particular depth estimation, tracking and selective attention algorithms *per se*, independently of foveal geometry aspects. Improvements were centered in the efficiency, reliability and robustness of the algorithms.

To control head pan and tilt movements, motion estimation is performed in a parametric optimization framework, using run-time selected object templates. The optimization algorithm was formulated such that a great part of the computations are performed at template initialization, reducing the on-line computational cost. The use of a redundant parameterization and a hierarchical "coarse-to-fine" estimation strategy, improve the robustness and convergence range of the method. The use of object templates has the advantage of providing estimates of absolute image displacements, rather than relative displacements given by optical-flow-like methods. The aim is to avoid drifts in the estimation due to error accumulation at each time step. A limitation is the assumption of a particular model for deformations in the object template, but we have shown experimentally that some deviations from the model are tolerated by the algorithm. However, future work should address object representations less "rigid" than templates, to cope

with deformable and articulated objects, as well as view point changes.

To identify interest points for saccade motion control, we have adapted an existing multi-feature saliency computation methodology to logpolar space. Due to the iterative nature of the method, the use of foveal images save significant computational processing. As previously stated, low-level feature extraction is performed in retinal coordinates to preserve image spatial-frequency content. This is illustrated with the use of feature extractors similar to non-linear ganglion cells present in the human retina. These feature extractors accumulate the rectified output of smaller scale linear ganglion cells. The same principle is adopted to the extraction of oriented features, in the form of Gabor wavelets. We have proposed a novel fast algorithm to Gabor filtering, which is about 40% more efficient than existing methods. The importance of oriented features in both bottom-up and top-down selective attention was experimentally illustrated. However, a deep evaluation of the method was not performed. Although there are similarities among the saliency computation model and human neuronal structures, it is not yet clear if the model is capable of exhibiting human-like performance. Recent work [110] has tried to quantitatively assess the plausibility of the selective visual attention model in comparison with human behavior. Though results were encouraging, the full assessment of such correlation has not been conclusive.

In terms of depth estimation, a disparity algorithm based on a Bayesian approach was adapted to logpolar images. The algorithm works by maximizing the likelihood of multiple disparity hypothesis. We have proposed a fast methodology to deal with ambiguities due to aperture problems in the disparity estimates, consisting of long-range spatial reinforcement of units tuned to similar disparities. A real-time implementation was presented and tested in realistic scenarios, to control eye vergence movements and segment close objects in front of the binocular system. Both the multiple hypothesis testing and the long-range reinforcement process are motivated by the operation of binocular neuronal structures in the human visual cortex. However, the particular disparity sensitive units employed in this work use directly the image gray levels, while biological units have receptive fields resembling Gabor kernels. Future work will address this issue, which, we think, will improve smoothness in the solution and robustness to illumination differences in the stereo pair.

In pure control aspects we have proposed methods that greatly simplify the design of head motion controllers. Assuming small deviations from tracking and providing an appropriate parameterization, it is possible to express both robot kinematics and dynamics directly in terms of image features. We have proposed a simple proportional controller for saccade motion control and a dynamical controller with motion prediction for smooth-pursuit control, but the model is general and flexible enough to support many other types of controllers. In future work we intend to develop predictive controllers tuned to several types of motions (periodic, parametric, etc.) and a scheme to coordinate their operation.

7.1 Future Directions

Though we have already pointed out some future research goals related to each of the subjects addressed in this thesis, there are many other points that deserve special attention and were left unaddressed.

The first of them is, naturally, the development of an integrated control system involving all the developed skills. We have not addressed this point here because the coordination and sequencing of the different behaviors is highly dependent on cognitive, motivational and task related aspects of robot state, which are issues that require *per*

se further advances. Though we think that task-specific behavior coordination can be achieved with a moderate amount of effort, to achieve full autonomy and adaptability is essential to have the capability of integrating knowledge in long-term operation periods (learning from experience), and this should be the main focus of research in future work. Better machine learning, knowledge acquisition and representation methods are needed to provide robots with the means of adapting, learning and developing their capabilities from their interaction with the environment. Based on this reasoning, we have identified some open problems where future research should be centered on, aiming at long-term flexibility, autonomy and adaptability of robot visual control systems.

Object Representation and Recognition

Most of our daily visual activity aims at identifying and recognizing objects in the environment for the planning and execution of actions. However, theory on object representation and recognition in realistic situations is still in early research stages. Usually, research works simplify this issue by using unambiguous features, easily identified by current vision algorithms. In non-modified environments, new advances on object recognition theory will play a major role on robot perceptual capabilities.

Short-Term Memory and Perceptual Stability

All the methods presented in this work use an image based reference frame, which is strongly affected by robot motion. The planning of robot actions require a representation of the external world in a stable reference frame. This representation, like a short-term memory, must be updated whenever a visual item appears or disappears from the robot surrounding environment. It may not be exhaustive (like a visual mosaic), but must allow the representation of important visual items so that the robot can reason and plan its actions.

Task Definition and Coordination

Agent's behavior is strongly determined by task related aspects. For example, the decision to shift the gaze to a particular items in the field of view or to fixate some other image region, is dependent on the current behavioral needs of the agent. If an agent is in unvisited places, it may desire to look at all salient points to get the gist of the environment. By the contrary, if the agent is engaged in social activity its behavior is biased toward detecting and tracking other individuals. In this work we have not assumed any task related control of the visual activity. The developed methods for depth perception, motion estimation and selective attention can be used for the control of vergence, smooth-pursuit and saccade eye movements, but the means of planning and sequencing these behaviors was not addressed in the thesis. Future work should aim at researching methodologies for appropriate representation of tasks and task-related knowledge, such that perceptual strategy can be guided toward task accomplishment.

Supervised and Unsupervised Learning

Aiming at adaptable and "growing" systems, it is fundamental that information from past experience (unsupervised) or exemplified by a "teacher" (supervised) can be incorporated in the system. Beside having to address the issue of representing this information, there is the need to develop efficient and scalable supervised and unsupervised learning methods.

We envisage a visual system where the identification of correlations between classes of objects and their task relevance can be estimated along time and used to optimize system behavior. Future work should aim at the improvement of scalability and flexibility of learning methodologies, in order to promote systems operating autonomously in long-term temporal ranges, adapting to environmental changes and improving continuously its performance.

Bibliography

[1] I. Ahrns and H. Neumann. Real-time monocular fixation control using the log-polar transformation and a confidence-based similarity measure. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, 1998.

[2] S. Amari and M. Arbib. Competition and cooperation in neural nets. In J. Metzler, editor, *Systems Neuroscience*, pages 119–165. Academic Press, 1977.

[3] A. Amir and M. Lindenbaum. A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):168–185, 1998.

[4] H. Asada and M. Brady. The curvature primal sketch. *PAMI*, 8(1):2–14, 1986.

[5] C. G. Atkeson, J. G. Hale, F. Pollick, M. Riley, S. Kotosaka, S. Schaal, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar, and M. Kawato. Using humanoid robots to study human behavior. *IEEE Intelligent Systems*, 1562:52–87, 1999.

[6] L. S. Balasuriya and J. P. Siebert. An artificial retina with a self-organised retinal receptive field tessellation. In *Proceedings of the Biologically-inspired Machine Vision, Theory and Application symposium, Artificial Intelligence and the Simulation of Behaviour Conventions*, April 2003.

[7] A. Basu and K. Wiebe. Enhancing videoconferencing using spatially varying sensing. *IEEE Trans. on Systems, Man, and Cybernetics*, 38(2):137–148, March 1998.

[8] J. Batista, P. Peixoto, and H. Araujo. Real-time vergence and binocular gaze control. In *Proc. IROS'97*, pages 7–11, Grenoble, France, September 1997.

[9] G. C. Baylis and J. Driver. Visual parsing and response competition: The effects of grouping. *Perceptual Psychophysics*, 51:145–162, 1992.

[10] B. Bederson. *A Miniature Space-Variant Active Vision System*. PhD thesis, New York University, 1992.

[11] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. ECCV*, pages 237–252, Santa Margherita Ligure, Italy, May 1992.

[12] A. Bernardino. Seguimento binocular de alvos móveis baseado em imagens log-polar. Master's thesis, Instituto Superior Técnico, Lisbon, Portugal, December 1996.

[13] A. Bernardino and J. Santos-Victor. Sensor geometry for dynamic vergence. In *Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, April 1996.

[14] A. Bernardino and J. Santos-Victor. Vergence control for robotic heads using log-polar images. In *Proc. IROS*, pages 1264–1271, Osaka, Japan, November 1996.

[15] A. Bernardino and J. Santos-Victor. Visual behaviours for binocular tracking. *Robotics and Autonomous Systems*, 25(3-4), 1998.

[16] A. Bernardino and J. Santos-Victor. Binocular visual tracking: Integration of perception and control. *IEEE Trans. on Robotics and Automation*, 15(6):137–146, December 1999.

[17] A. Bernardino, J. Santos-Victor, and G. Sandini. Foveated active tracking with redundant 2d motion parameters. *Robotics and Autonomous Systems*, 39(3-4):205–221, June 2002.

[18] R. Bischoff and V. Graefe. Hermes: an intelligent humanoid robot, designed and tested for dependability. In B. Siciliano and P. Dario, editors, *Springer Tracts in Advanced Robotics (STAR): Experimental Robotics VIII*, volume 5. Springer, Heidelberg, 2003.

[19] E. Blaser, Z. Pylyshyn, and A. Holcombe. Tracking an object through feature space. *Nature*, 408:196–199, 2000.

[20] M. Bolduc and M. Levine. A real-time foveated sensor with overlapping receptive fields. *Real-Time Imaging: Special Issue on Natural and Artifical Imaging and Vision*, 3(3):195–212, 1997.

[21] M. Bolduc and M. Levine. A review of biologically motivated space-variant data reduction models for robotic vision. *CVIU*, 69(2):170–184, February 1998.

[22] Y. Boykov, R. Zabih, and O. Veksler. Disparity component matching for visual correspondence. In *Proc. CVPR*, pages 470–475, 1997.

[23] Y. Boykov, R. Zabih, and O. Veksler. Markov random fields with efficient approximations. In *Proc. CVPR*, pages 648–655, 1998.

[24] T. Boyling and J. Siebert. A fast foveated stereo matcher. In *Proc. of Conf. on Imaging Science Systems and Technology (CISST)*, pages 417–423, Las Vegas, USA, 2000.

[25] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati. Active vision for sociable robots. *Proc. IEEE Trans. Systems, Man and Cybernetics- A*, 31(5):443–453, September 2001.

[26] R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. M. Williamson. The cog project: Building a humanoid robot. *Lecture Notes in Computer Science*, 1562:52–87, 1999.

[27] E. Bruno and D. Pellerin. Robust motion estimation using gabor spatial filters. In *Proc. of the 10th European Signal Processing Conference*, Sept. 2000.

[28] B. Burt and L. Florack. Front end vision: A multiscale geometry engine. In *Proc. IEEE Intl' Workshop on Biologically Motivated Computer Vision*, Seoul, Korea, May 2000.

[29] P. Burt. Smart sensing within a pyramid vision machine. *Proc. IEEE*, 76(8):1006–1015, August 1988.

[30] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, 4(31):532–540, April 1983.

[31] J. Canny. A computational approach to edge detection. *PAMI*, 8(6):679–698, 1986.

[32] C. Capurro, F. Panerai, and G. Sandini. Vergence and tracking fusing log-polar images. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, Vienna, August 1996.

[33] C. Capurro, F. Panerai, and G. Sandini. Dynamic vergence using log-polar images. *IJCV*, 24(1):79–94, August 1997.

[34] R. Carpenter. *Movements of the eyes*. Pion, London, 1988.

[35] C. Castiello and C. Umiltà. Splitting focal attention. *Journal of Experimental Psychology: Human Perception and Performance*, 18(3):837–848, 1992.

[36] E. Chang, S. Mallat, and C. Yap. Wavelet foveation. *J. Applied and Computational Harmonic Analysis*, 9(3):312–335, October 2000.

[37] F. Chaumette, P. Rives, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In *Proc. ICRA*, pages 2248–2253, 1991.

[38] A. Cohen and J. Kovačević. Wavelets: the mathematical background. *Proceedings of the IEEE*, 84(4), 1996.

[39] D. Coombs and C. Brown. Real-time binocular smooth pursuit. *IJCV*, 11(2):147–164, October 1993.

[40] P. Corke and M. Good. Dynamic effects in visual closed-loop systems. *IEEE Trans. Robotics and Automation*, 12(5):671–683, October 1996.

[41] J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1986.

[42] J. Crowley, O. Riff, and J. Piater. Fast computations of characteristic scale using a half-octave pyramid. In *CogVis 2002, International Workshop on Cognitive Computing*, Zurich, October 2002.

[43] P. Daniel and D. Whitteridge. The representation of the visual field on the cerebral cortex in monkeys. *Journal of Physiology*, 159:203–221, 1961.

[44] K. Daniilidis. Attentive visual motion processing: computations in the log-polar plane. *Computing*, 11:1–20, 1995.

[45] J. Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision Research*, 20:847–856, 1980.

[46] G. DeAngelis and W. Newsome. Organization of disparity-selective neurons in macaque area mt. *The Journal of Neuroscience*, 19(4):1398–1415, 1999.

[47] J. Demb, L. Haarsma, M. Freed, and P. Sterling. Functional circuitry of the retinal ganglion's cell nonlinear receptive field. *The Journal of Neuroscience*, 19(22):9756–9767, 1999.

[48] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience*, 18:193–222, 1995.

[49] J. Driver and G. C. Baylis. Attention and visual object segmentation. In R. Parasuraman, editor, *The Attentive Brain*, pages 299–325. The MIT Press, Cambridge and London, 1998.

[50] J. Duncan. Selective attention and the organization of visual information. *Journal of Experimental Psychology: General*, 113:501–517, 1984.

[51] S. Edelman. Receptive fields for vision: from hyperaccuity to object recognition. In R. J. Watt, editor, *Vision*. MIT Press, 1996.

[52] C. Eriksen and S. James. Visual attention within and around the field of focal attention: a zoom lens model. *Perception and Psychophysics*, 40(4):225–240, 1986.

[53] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robotics and Automation*, 8(3):313–326, June 1992.

[54] S. Excel and L. Pessoa. Space-variant representation for active object recognition. In *Proc. of the International Symposium on Computer Graphics, Image Processing, and Vision (SIBGRAPI'98)*, Rio de Janeiro, Brasil, October 1998.

[55] P. Ferreira. Discrete finite frames and signal reconstruction. In J. Byrnes, editor, *Signal Processing for Multimedia*. IOS Press, 1999.

[56] J. M. Findlay and R. Walker. A model of saccade generation based on parallel processing and competitive inhibition. *Behavioral and Brain Sciences*, 22:661–721, 1999.

[57] S. Fischer and G. Cristóbal. Minimum entropy transform using gabor wavelets for image compression. In *Proc. of Int. Conf. on Image Analysis and Processing*, Palermo, Italy, Sept. 2001.

[58] P. M. Fitzpatrick. *From First Contact to Close Encounters: A Developmentally Deep Perceptual System For a Humanoid Robot*. PhD thesis, MIT, 2003.

[59] D. Fleet, A. Jepson, and M. Jenkin. Image matching using the windowed fourier phase. *CVGIP: Image Understanding*, 53(2):198–210, March 1991.

[60] D. Gabor. Theory of communication. *J. IEE*, 93:429–459, 1946.

[61] Y. Gdalyahu, D. Weinshal, and M. Wermank. Self organization in vision: stochastic clustering for image segmentation, perceptual grouping and image database organization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1053–1074, 2001.

[62] W. S. Geisler and J. S. Perry. A real-time foveated multi-resolution system for low-bandwidth video communication. In *Human Vision and Electronic Imaging, SPIE Proceedings 3299*, pages 294–305, August 1998.

[63] W. S. Geisler and J. S. Perry. Real-time simulation of arbitrary view fields. In *Proc. of the ACM Symposium on Eye Tracking Research and Applications*, 2002.

[64] A. Gelb. *Applied Optimal Estimation*. The MIT Press, 1994.

[65] J.-M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. In *ECCV (1)*, pages 99–112, 2002.

[66] M. Gleicher. Projective registration with difference decomposition. In *Proc. CVPR'97*, pages 331–337, June 1997.

[67] G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore and London, 89.

[68] H. M. Gomes and R. B. Fisher. Primal sketch feature extraction from a log-polar image. *Pattern Recognition Letters*, 24(7):983–992, 2003.

[69] N. Gracias and J. Santos-Victor. Trajectory reconstruction using mosaic registration. In *Proc. SIRS'99*, Coimbra, Portugal, July 1999.

[70] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, October 2003.

[71] E. Grosso and M. Tistarelli. Log-polar stereo for anthropomorphic robots. In *Proc. 6th European Conference on Computer Vision (ECCV)*, pages 299–313, Dublin, Ireland, July 2000.

[72] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[73] G. Hager, W. Chang, and A. Morse. Robot hand-eye coordination based on stereo vision. *IEEE Control Systems Magazine*, 15(1):30–39, 1995.

[74] M. Hansen and G. Sommer. Active depth estimation with gaze and vergence control using gabor filters. In *Proc. ICPR'96*, pages 287–291, 1996.

[75] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.

[76] B. Horn. *Robot Vision*. MIT Press, McGraw Hill, 1986.

[77] D. Hubel and T. Wiesel. Stereoscopic vision in macaque monkey. cells sensitive to binocular depth in area 18 of the macaque monkey cortex. *Nature*, 225:41–42, 1970.

[78] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. Robotics and Automation*, 12(5):651–670, October 1996.

[79] C. W. II. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans. Systems, Man and Cybernetics*, 16(1):93–101, 1986.

[80] J. Intriligator and P. Cavanagh. The spatial resolution of visual attention. *Cognitive Psychology*, 2001.

[81] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.

[82] L. Itti, C. Kock, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE PAMI*, 20:1254–1259, 1998.

[83] M. Jagersand. Saliency maps and attention selection in scale space and spatial coordinates: An information theoretic approach. In *Proc. of ICCV*, pages 195–202, 1995.

[84] J. F. Juola, D. J. Bouwhuis, E. E. Cooper, and C. B. Warner. Control of attention around the fovea. *Journal of Experimental Psychology: Human Perception and Performance*, 17(1):125–141, 1991.

[85] T. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[86] T. Kanda, N. Miralles, M. Shiomi, T. Miyashita, I. Fasel, J. Movellan, and H. Ishiguro. Face-to-face interactive humanoid robot. Submitted to ICRA'04.

[87] W. N. Klarquist and A. C. Bovik. Fovea: A foveated vergent active stereo system for dynamic three-dimensional scene recovery. *IEEE Trans. Robotics and Automation*, 14(5):755 – 770, Oct. 1998.

[88] C. Koch and S. Ullman. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[89] J. Koenderink and A. van Doorn. Visual detection of spatial contrast; influence of location in the visual field, target extent and illuminance level. *Biol.Cybern.*, 30:157–167, 1978.

[90] P. Kortum and W. Geisler. Implementation of a foveated image coding system for image bandwidth reduction. *SPIE Proceedings*, 2657:350–360, 1996.

[91] E. Krotkov, K. Henriksen, and R. Kories. Stereo ranging with verging cameras. *IEEE PAMI*, 12(12):1200–1205, December 1990.

[92] D. LaBerge. Spatial extent of attention to letters and words. *Journal of Experimental Psychology: Human Perception and Performance*, 9:371–379, 1983.

[93] T. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(10), October 1996.

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

[95] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[96] T. Lindeberg and L. Florack. *Foveal scale-space and the linear increase of receptive field size as a function of eccentricity*. Tech. Report ISRN KTH/NA/P-94/27-SE, 1994.

[97] G. Loy and A. Zelinsky. A fast radial symmetry transform for detecting points of interest. In *Proc. of ECCV*, pages 358–368, 1995.

[98] S. Mallat. Wavelets for a vision. *Proceedings of the IEEE*, 84(4):604–614, 1996.

[99] S. Mallat. *A Wavelet Tour of Signal Processing, 2nd Ed.* Academic Press, 1999.

[100] R. Manzotti, G. Metta, A. Gasteratos, and G. Sandini. Disparity estimation on log-polar images and vergence control. *Computer Vision and Image Understanding*, 83:97–117, April–June 2001.

[101] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[102] G. Medioni. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[103] G. Metta, F. Panerai, and G. Sandini. Babybot: A biologically inspired developing robotic agent. In *Proc. of the AAAI Fall Symposium Symposium*, Cape Cod, USA, October 2000.

[104] F. Miau, C. Papageorgiou, and L. Itti. Neuromorphic algorithms for computer vision and attention. *Proc. SPIE 46 Annual International Symposium on Optical Science and Technology*, 4479:12–23, 2001.

[105] R. Milanese. *Detecting salient regions in an image: from biological evidence to computer implementation*. PhD thesis, University of Geneva, Switzerland, 1993.

[106] C. Morimoto and R. Chellappa. Fast electronic digital image stabilization. In *Proc. ICPR*, Vienna, Austria, August 1996.

[107] O. Nestares, R. Navarro, and J. Portilla. Efficient spatial-domain implementation of a multiscale image representation based on gabor functions. *Journal of Electronic Imaging*, 7(1):166–173, Jan. 1998.

[108] J. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *International Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.

[109] T. J. Olson. Stereopsis for verging systems. In *Proc. CVPR'93*, New York, USA, June 1993.

[110] N. Ouerhani, R. von Wartburg, H. Hügli, and R. Müri. Empirical validation of the saliency-based model of visual attention. *Electronic Letters on Computer Vision and Image Analysis*, 3(1):13–24, 2004.

[111] F. Panerai, C. Capurro, and G. Sandini. Space variant vision for an active camera mount. In *Proc. SPIE AeroSense95*, Florida, USA, April 1995.

[112] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38:15–33, 2000.

[113] N. Papanikolopoulos, B. Nelson, and P. Kosla. Six degree-of-freedom hand/eye visual tracking with uncertain parameters. *IEEE Trans. Robotics and Automation*, 11(5):725–732, October 1995.

[114] M. Peters and A. Sowmya. A real-time variable sampling technique: Diem. In *Proc. 14th International Conference on Computer Vision*, Brisbane, Australia, August 1998.

[115] N. Petkov and P. Kruizinga. Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells. *Biological Cybernetics*, 76:83–96, 1997.

[116] S. Pollard, J. Mayhew, and J. Frisby. Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[117] M. I. Posner. Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32:3–25, 1980.

[118] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52:93–99, 1985.

[119] J. Puzicha, T. Hofmann, and J. Buhmann. Histogram clustering for unsupervised segmentation and image retrieval. *Pattern Recognition Letters*, 20:899–909, 1999.

[120] Z. Pylyshyn and R. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3(3):1–19, 1988.

[121] T. Randen and Husøy. Image representation using 2d gabor wavelets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, April 1999.

[122] I. Reid and D. Murray. Active tracking of foveated feature clusters using affine structure. *IJCV*, 18(1):41–60, 1996.

[123] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context free attentional operators: The generalized symmerty transform. *IJCV*, 14:119–130, 1995.

[124] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat Neurosci*, 2:1019–1025, 1999.

[125] G. Salgian and D. Ballard. Visual routines for vehicle control. In D. Kriegman, G. Hager, and S. Morse, editors, *The Confluence of Vision and Control*. Springer Verlag, 1998.

[126] G. Sandini and V. Tagliasco. An antropomorphic retina-like structure for scene analysis. *Computer Vision, Graphics and Image Processing*, 14(3):365–372, 1980.

[127] J. Santos-Victor and A. Bernardino. Vision-based navigation, environmental representations, and imaging geometries. In *Proc. 10th International Symposium of Robotics Research*, Victoria, Australia, November 2001.

[128] J. Santos-Victor and G. Sandini. Visual behaviors for docking. *CVIU*, 67(3), September 1997.

[129] J. Santos-Victor, F. van Trigt, and J. Sentieiro. Medusa - a stereo head for active vision. In *Proc. of the Int. Symposium on Intelligent Robotic Systems*, Grenoble, France, July 1994.

[130] D. Scharstein and R. Szelisky. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, April–June 2002.

[131] E. Schwartz. Spatial mapping in the primate sensory projection: Analytic structure and relevance to perception. *Biological Cybernetics*, 25:181–194, 1977.

[132] E. Schwartz. Computational anatomy and functional architecture of the striate cortex. *Vision Research*, 20:645–669, 1980.

[133] S. Shah and M. D. Levine. *Information Processing in Primate Retinal Cone Pathways: A Model.* Tech. Report TR-CIM-93-18, Centre for Intelligent Machines, McGill University, Canada, 1998.

[134] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Proc. CVPR*, 2000.

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

[136] J. Siebert and D. Wilson. Foveated vergence and stereo. In *Proc. of the 3rd Int. Conf. on Visual Search (TICVS)*, Nottingham, UK, August 1992.

[137] E. Sifakis, C. Garcia, and G. Tziritas. Bayesian level sets for image segmentation. *JVCIR*, 13(1/2):44–64, March 2002.

[138] C. Silva and J. Santos-Victor. Egomotion estimation using log-polar images. In *Proc. of the International Conference of Computer Vision (ICCV)*, Bombay, India, January 1998.

[139] F. Smeraldi, N. Capdevielle, and J. Bigun. Face authentication by retinotopic sampling of the gabor decomposition and support vector machines. In *Proc. of the 2nd International Conference on Audio and Video Based Biometric Person Authentication (AVBPA'98)*, Washington DC, USA, March 1999.

[140] T. Tangsukson and J. Havlicek. Am-fm image segmentation. In *Proc. IEEE Int. Conf. on Image Processing*, pages 104–107, Vancouver, Canada, Sept. 2000.

[141] W. M. Theimer and H. A. Mallot. Phase-based binocular vergence control and depth reconstruction using active vision. *CVGIP: Image Understanding*, 60(3):343–358, November 1994.

[142] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of the time-to-impact from optical flow. *IEEE Trans. on PAMI*, 15(8):401–411, April 1993.

[143] F. Tong and Z. Li. Reciprocal-wedge transform for space-variant sensing. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 17(5):500–511, May 1995.

[144] R. Tootell, M. Silverman, E. Swikes, and R. DeValois. Deosyxlucose analysis of retinotopic organization in primate striate cortex. *Science*, 218:902–904, 1982.

[145] V. Traver. *Motion Estimation Algorithms in Log-Polar Images and Application to Monocular Active Tracking.* PhD thesis, Universitat Jaume I, Castellón, Spain, September 2002.

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

[147] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davies, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78:507–546, 1995.

[148] J. K. Tsotsos. An inhibitory beam for attentional selection. In Harris and Jenkin, editors, *Spatial Vision in Humans and Robots*, pages 313–331. Cambridge University Press, 1991.

[149] T. Uhlin, P. Nordlund, A. Maki, and J.-O. Eklundh. Towards an active visual observer. In *Proc. of ICCV*, pages 679–686, 1995.

[150] S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39:145–155, 2002.

[151] S. P. Vecera and M. Behrmann. Attention and unit formation: A biased competition account of object-based attention. In T. Shipley and P. Kellman, editors, *From fragments to objects*, pages 145–180. Elsevier, New York, 2001.

[152] M. Vincze and C. Weiman. A general relationship for optimal tracking performance. *SPIE Proceedings*, 2904:402–412, 1996.

[153] A. M. Wallace and D. J. McLaren. Gradient detection in discrete log-polar images. *Pattern Recognition Letters*, 24(14):2463–2470, 2003.

[154] R. Wallace, P. Ong, B. Bederson, and E. Schwartz. Space variant image processing. *IJCV*, 13(1):71–90, September 1995.

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

[156] Z. Wang and A. C. Bovik. Embedded foveation image coding. *IEEE Transactions on Image Processing*, 10(10):1397–1410, October 2001.

[157] C. Weiman. Log-polar vision for mobile robot navigation. In *Proc. of Electronic Imaging Conference*, pages 382–385, Boston, USA, November 1990.

[158] M. Wertheimer. Laws of organization in perceptual forms. In S. Yantis, editor, *Key Readings in Cognition: Visual Perception*, pages 216–224. Psychology Press, Philadelphia, USA, 2001.

[159] S. Wilson. On the retino-cortical mapping. *International Journal on Man-Machine studies*, 18:361–389, 1983.

[160] J. M. Wolve, K. R. Cave, and S. L. Franzel. Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology*, 15:419–433, 1989.

[161] Y. Yeshurun and E. Schwartz. Cepstral filtering on a columnar image architecture: A fast algorithm for binocular stereo segmentation. *IEEE Trans. PAMI*, 11(7):759–767, July 1989.

[162] I. Young and L. van Vliet. Recursive implementation of the gaussian filter. *Signal Processing*, 44:139–151, 1995.

[163] I. Young, L. van Vliet, and M. van Ginkel. Recursive gabor filtering. *IEEE Trans. on Signal Processing*, 50(11):2798–2805, 2002.

[164] C. T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1), 1971.

[165] The usc-sipi image database. http://sipi.usc.edu/services/database.

[166] Sony dream robot qrio. http://www.sony.net/SonyInfo/QRIO.

[167] Honda asimo. http://world.honda.com/ASIMO/.

[168] Humanoid robotics: Promet. http://www.kawada.co.jp/english/aircraft_project1.html.

[169] Humanoid robot: Pino. http://www.zmp.co.jp/e_html/products.html.

Appendix A

Binocular Image Jacobians

Image Jacobians, or Feature Sensitivity Matrices, express the differential kinematics relationships between given motor commands and the corresponding motion of visual features. According to Section 3.2, image jacobians can be computed by the following expression, in the general case (also Eq. (3.15)):

$$\begin{cases} \mathbf{J}_q\left(\underline{q},\underline{p}\right) = \frac{\partial \mathcal{F}}{\partial \mathcal{P}_c}\left(\mathcal{P}_c\left(\underline{q},\underline{p}\right)\right) \cdot \frac{\partial \mathcal{P}_c}{\partial \underline{q}}\left(\underline{q},\underline{p}\right) \\ \mathbf{J}_p\left(\underline{q},\underline{p}\right) = \frac{\partial \mathcal{F}}{\partial \mathcal{P}_c}\left(\mathcal{P}_c\left(\underline{q},\underline{p}\right)\right) \cdot \frac{\partial \mathcal{P}_c}{\partial \underline{p}}\left(\underline{q},\underline{p}\right) \end{cases} \tag{A.1}$$

Also, in Section 3.3, we obtained the expressions for computing the relative target position. In our particular case this is given by (3.24):

$$\underline{P}_c = \mathcal{P}_c\left(\underline{q},\underline{p}\right) = \begin{bmatrix} X_l \\ Y_l \\ Z_l \\ X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} \rho\, \mathrm{c}_\gamma\, \mathrm{c}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, s_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{s}_v - \mathrm{c}_v\, B \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_t\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{c}_t \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{c}_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{c}_v + \mathrm{s}_v\, B \\ \rho\, \mathrm{c}_\gamma\, \mathrm{c}_v\, \mathrm{s}_{\phi-p} - \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{s}_v\, \mathrm{c}_{\phi-p} - \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t\, \mathrm{s}_v + \mathrm{c}_v\, B \\ -\rho\, \mathrm{c}_\gamma\, \mathrm{s}_t\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{c}_t \\ \rho\, \mathrm{c}_\gamma\, \mathrm{s}_v\, \mathrm{s}_{\phi-p} + \rho\, \mathrm{c}_\gamma\, \mathrm{c}_t\, \mathrm{c}_v\, \mathrm{c}_{\phi-p} + \rho\, \mathrm{s}_\gamma\, \mathrm{s}_t \mathrm{c}_v + \mathrm{s}_v\, B \end{bmatrix} \tag{A.2}$$

Also, we derived the image projection function (3.25):

$$\mathcal{F}\left(\underline{P}_c\right) = \begin{bmatrix} -\frac{X_l}{2Z_l} + \frac{X_r}{2Z_r} \\ -\frac{X_l}{2Z_l} - \frac{X_r}{2Z_r} \\ \frac{Y_l}{2Z_l} + \frac{Y_r}{2Z_r} \end{bmatrix} \tag{A.3}$$

To compute the image jacobians $\mathbf{J}_q$ and $\mathbf{J}_p$, we need to derive the partial derivative matrices $\frac{\partial \mathcal{F}}{\partial \mathcal{P}_c}$, $\frac{\partial \mathcal{P}_c}{\partial \underline{q}}$ and $\frac{\partial \mathcal{P}_c}{\partial \underline{p}}$.

Sensitivity of $\mathcal{F}$ with respect to $\underline{P}_c$

$$\frac{\partial \mathcal{F}}{\partial \underline{P}_c}(\underline{P}_c) = \begin{bmatrix} -\frac{1}{2Z_l} & 0 & \frac{X_l}{2Z_l^2} & \frac{1}{2Z_r} & 0 & -\frac{X_r}{2Z_r^2} \\ -\frac{1}{2Z_l} & 0 & \frac{X_l}{2Z_l^2} & -\frac{1}{2Z_r} & 0 & \frac{X_r}{2Z_r^2} \\ 0 & \frac{1}{2Z_l} & -\frac{Y_l}{2Z_l^2} & 0 & \frac{1}{2Z_r} & -\frac{Y_r}{2Z_r^2} \end{bmatrix} \tag{A.4}$$

Sensitivity of $\mathcal{P}_c$ with respect to $\underline{q}$

$$\frac{\partial \mathcal{P}_c}{\partial \underline{q}}(\underline{q},\underline{p}) = \begin{bmatrix} -\rho c_\gamma s_v s_\delta + \rho c_\gamma c_t c_v c_\delta + \rho s_\gamma s_t c_v + s_v B & -\rho c_\gamma c_v c_\delta + \rho c_\gamma c_t s_v s_\delta & -\rho c_\gamma s_t s_v c_\delta + \rho s_\gamma c_t s_v \\ 0 & -\rho c_\gamma s_t s_\delta & -\rho c_\gamma c_t c_\delta - \rho s_\gamma s_t \\ -\rho c_\gamma c_v s_\delta - \rho c_\gamma c_t s_v c_\delta - \rho s_\gamma s_t s_v + c_v B & \rho c_\gamma s_v c_\delta + \rho c_\gamma c_t c_v s_\delta & -\rho c_\gamma s_t c_v c_\delta + \rho s_\gamma c_t c_v \\ -\rho c_\gamma s_v s_\delta - \rho c_\gamma c_t c_v c_\delta - \rho s_\gamma s_t c_v - s_v B & -\rho c_\gamma c_v c_\delta - \rho c_\gamma c_t s_v s_\delta & \rho c_\gamma s_t s_v c_\delta - \rho s_\gamma c_t s_v \\ 0 & -\rho c_\gamma s_t s_\delta & -\rho c_\gamma c_t c_\delta - \rho s_\gamma s_t \\ \rho c_\gamma c_v s_\delta - \rho c_\gamma c_t s_v c_\delta - \rho s_\gamma s_t s_v + c_v B & -\rho c_\gamma s_v c_\delta + \rho c_\gamma c_t c_v s_\delta & -\rho c_\gamma s_t c_v c_\delta + \rho s_\gamma c_t c_v \end{bmatrix} \tag{A.5}$$

where $\delta = \phi - \theta_p$.

Sensitivity of $\mathcal{P}_c$ with respect to $\underline{p}$

$$\frac{\partial \mathcal{P}_c}{\partial \underline{p}}(\underline{q},\underline{p}) = \begin{bmatrix} c_\gamma c_v s_\delta + c_\gamma c_t s_v c_\delta + s_\gamma s_t s_v & \rho c_\gamma c_v c_\delta - \rho c_\gamma c_t s_v s_\delta & -\rho s_\gamma c_v s_\delta - \rho s_\gamma c_t s_v c_\delta + \rho c_\gamma s_t s_v \\ -c_\gamma s_t c_\delta + s_\gamma c_t & \rho c_\gamma s_t s_\delta & \rho s_\gamma s_t c_\delta + \rho c_\gamma c_t \\ -c_\gamma s_v s_\delta + c_\gamma c_t c_v c_\delta + s_\gamma s_t c_v & -\rho c_\gamma s_v c_\delta - \rho c_\gamma c_t c_v s_\delta & \rho s_\gamma s_v s_\delta - \rho s_\gamma c_t c_v c_\delta + \rho c_\gamma s_t c_v \\ c_\gamma c_v s_\delta - c_\gamma c_t s_v c_\delta - s_\gamma s_t s_v & \rho c_\gamma c_v c_\delta + \rho c_\gamma c_t s_v s_\delta & -\rho s_\gamma c_v s_\delta + \rho s_\gamma c_t s_v c_\delta - \rho c_\gamma s_t s_v \\ -c_\gamma s_t c_\delta + s_\gamma c_t & \rho c_\gamma s_t s_\delta & \rho s_\gamma s_t c_\delta + \rho c_\gamma c_t \\ c_\gamma s_v s_\delta + c_\gamma c_t c_v c_\delta + s_\gamma s_t c_v & \rho c_\gamma s_v c_\delta - \rho c_\gamma c_t c_v s_\delta & -\rho s_\gamma s_v s_\delta - \rho s_\gamma c_t c_v c_\delta + \rho c_\gamma s_t c_v \end{bmatrix} \tag{A.6}$$

where $\delta = \phi - \theta_p$.

Sensitivity matrices at equilibrium

In equilibrium, the following constraints hold (Eqs. (3.26) and (3.27)):

$$\underline{p}^0 = (\rho, \phi, \gamma) = (B \cot \theta_v, \theta_p, \theta_t)' \tag{A.7}$$

$$\underline{P}_c^0 = \mathcal{P}_c(\underline{q}, \underline{p}^0) = \begin{bmatrix} X_l \\ Y_l \\ Z_l \\ X_r \\ Y_r \\ Z_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ B/\mathrm{s}_v \\ 0 \\ 0 \\ B/\mathrm{s}_v \end{bmatrix} \tag{A.8}$$

Substituting the above conditions in the sensitivity matrices of (A.4), (A.5), and (A.6) we can compute those matrices at the equilibrium manifold:

$$\frac{\partial \mathcal{F}}{\partial \underline{P}_c}(\underline{P}_c^0) = \begin{bmatrix} -\frac{s_v}{2B} & 0 & 0 & \frac{s_v}{2B} & 0 & 0 \\ -\frac{s_v}{2B} & 0 & 0 & -\frac{s_v}{2B} & 0 & 0 \\ 0 & \frac{s_v}{2B} & 0 & 0 & \frac{s_v}{2B} & 0 \end{bmatrix} \tag{A.9}$$

$$\frac{\partial \mathcal{P}_c}{\partial \underline{q}}(\underline{q}, \underline{p}^0) = \begin{bmatrix} \frac{B}{s_v} & -\frac{B}{s_v}c_v^2 c_t & 0 \\ 0 & 0 & -\frac{B}{s_v}c_v \\ 0 & Bc_v c_t & 0 \\ -\frac{B}{s_v} & -\frac{B}{s_v}c_v^2 c_t & 0 \\ 0 & 0 & -\frac{B}{s_v}c_v \\ 0 & -Bc_v c_t & 0 \end{bmatrix} \tag{A.10}$$

$$\frac{\partial \underline{P}_c}{\partial \underline{p}}(\underline{q}, \underline{p}^0) = \begin{bmatrix} s_v & \frac{B}{s_v}c_v^2 c_t & 0 \\ 0 & 0 & \frac{B}{s_v}c_v \\ c_v & -Bc_v c_t & 0 \\ -s_v & \frac{B}{s_v}c_v^2 c_t & 0 \\ 0 & 0 & \frac{B}{s_v}c_v \\ c_v & Bc_v c_t & 0 \end{bmatrix} \tag{A.11}$$

Finally, the jacobian matrices at equilibrium are:

$$J_q(\underline{q}, \underline{p}^0) = \begin{bmatrix} -1 & 0 & 0 \\ 0 & c_t c_v^2 & 0 \\ 0 & 0 & -c_v \end{bmatrix} \tag{A.12}$$

$$J_p(\underline{q}, \underline{p}^0) = \begin{bmatrix} -s_v^2/B & 0 & 0 \\ 0 & -c_t c_v^2 & 0 \\ 0 & 0 & c_v \end{bmatrix} \tag{A.13}$$

Appendix B

The Laplacian Pyramid

The Laplacian Pyramid was introduced in [30] as a technique for fast image encoding and decoding. The technique found many applications other than image coding and transmission because of its multiscale representation. The method consists on very simple operations, allowing easy and fast computer implementations: convolutions with Gaussian-like low-pass filters, subtractions, downsampling and upsampling. A pyramid structure with N levels is built recursively from bottom to top. The lowest level on the pyramid has the same number of pixels as the original image and codes high frequency components of the image. The resolution decreases by one quarter each level up in the pyramid, while the represented frequencies decrease in octaves. Once the pyramid is constructed, either the original image or any or its intermediate scales can be easily reconstructed.

Let $f(x, y)$ be an image and $g(x, y)$ a low-pass filter. Two functions mediate the pyramid operations:

- REDUCE - Low-Pass filter and subsample by two in each dimension:

$$\text{REDUCE}\{f(x,y)\} = (f * g)(2x, 2y) \tag{B.1}$$

- EXPAND - Upsample by a factor of two in each dimension (with zero insertion) and low-pass filter:

$$\text{EXPAND}\{f(x,y)\} = 4\Big[\sum_k \sum_l f(k,l)\delta(x-2k, y-2l)\Big] * g(x,y) \tag{B.2}$$

The construction of the pyramid consists in computing successive low-pass approximations of the image and storing the approximation errors at each level. The application of an EXPAND after a REDUCE operation is a way to produce a low-pass approximation of the image. The residual information, obtained by subtracting the approximation from the original image, is stored. The process is iterated with the reduced image of level $i-1$ as the input of level i:

1. Compute the low-pass approximation of the image $f_i(x, y)$ by:

$$\hat{f}_i = EXPAND\{REDUCE\{f_i(x,y)\}\} \tag{B.3}$$

2. Compute the residual:

$$\tilde{f}_i(x,y) = f_i(x,y) - \hat{f}_i(x,y) \tag{B.4}$$

3. Repeat the procedure for the next level with the reduced image $f_{i+1}(x, y)$ as input:

$$f_{i+1}(x, y) = REDUCE\{f_i(x, y)\} \tag{B.5}$$

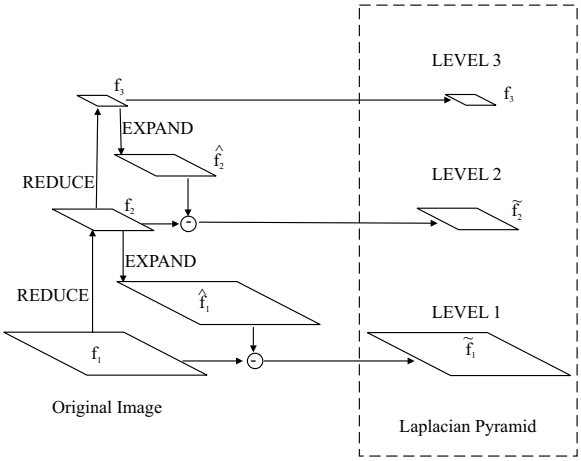


Figure B.1: Diagram for the construction of a Laplacian pyramid

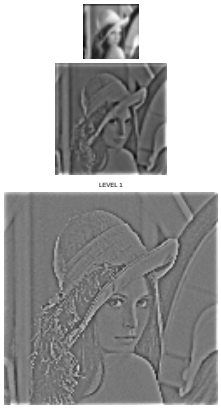


Figure B.2: The images of a Laplacian pyramid with 3 levels.

The whole process is illustrated in Fig. B.1 and its application to a test image is shown in Fig. B.2. The reason for denominating *Laplacian* to the pyramid is that each level can be obtained equivalently by Difference-Of-Gaussian filters, which resemble the Laplacian operator.

After the pyramid is built, it is fairly straightforward to reconstruct the original image or its approximation at any level. One just have to recursively add the residuals to the approximation in the previous level (see Fig. B.3):

$$f_i = \text{EXPAND}\{f_{i+1}\} + \tilde{f}_i \tag{B.6}$$

An equivalent way to obtain the original image is to expand each level of the pyramid to its full size (prior to pyramid construction) and then sum all the layers into a single image. This is illustrated in Fig. B.4.

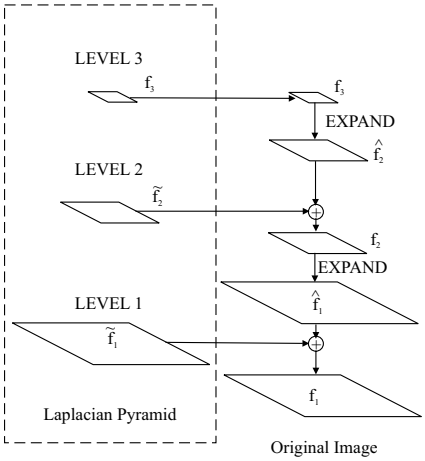


Figure B.3: Diagram for the reconstruction of images from a Laplacian pyramid.

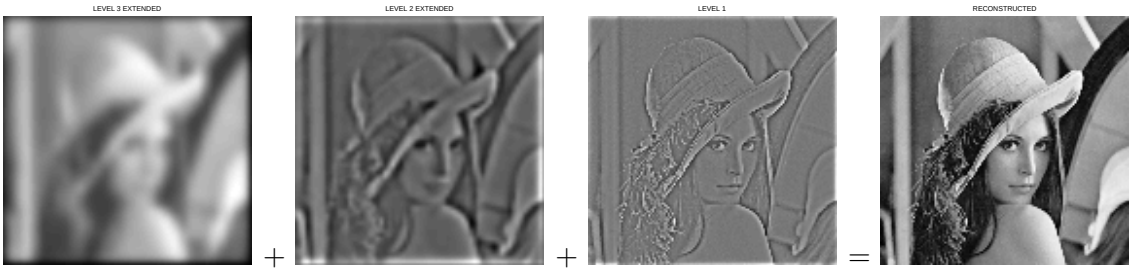


Figure B.4: The original image can be reconstructed from the sum of all pyramid levels expanded to full size.

Appendix C

Wavelet Theory

Wavelet theory proposes an alternative spectral decomposition to the usual Fourier analysis. In the Fourier domain, a signal is represented as a linear combination of trigonometric functions (sines and cosines) which have infinite support in time. From the Fourier coefficients one can just analyse the global spectral content of the signal and not information from localized regions. Instead, the wavelet theory proposes to represent signals as linear combination of functions, compactly supported in time and in frequency. In a Fourier decomposition, coefficients are indexed by frequency, whereas a wavelet decomposition indexes its coefficients by location τ and scale σ. The continuous wavelet transform (CWT), is defined as:

$$CWT\{f(t);\tau,\sigma\} =< f,\psi_{\tau,\sigma} >= \int_{-\infty}^{+\infty} f(t)\psi^*_{\tau,\sigma}(t)dt \tag{C.1}$$

where:

$$\psi_{\tau,\sigma}(t) = \frac{1}{\sqrt{\sigma}}\psi(\frac{t-\tau}{\sigma}) \tag{C.2}$$

and ψ is a band-pass "prototype" function called the *mother wavelet*. In practical applications, wavelet transforms must be computed on discrete grids. Wavelet theory provides the necessary technical conditions such that a signal can be completely represented by its samples.

Depending on the way the parameters τ, σ are discretized, wavelet transforms appear in different types. With practical applications in mind, the most useful type is the Discrete Wavelet Transform, which apply to discrete time and scales. Before we go into the details, let us introduce the concept of multiresolution spaces.

C.1 Multiresolution spaces

The whole idea behind a wavelet decompositions is to have a hierarchical multiresolution representation of signals, consisting of, at the first level, a coarse representation and, at the following levels, the details at consecutively higher resolutions.

Let us consider a low-pass function $\phi(t) \in L^2(R)$ of unit scale and define V_0 as the subspace generated by the basis $\{\phi(t-i), i \in Z\}$. Any signal $f(t)$ in V_0 can be expressed as a linear combination of the basis functions:

$$f_0(t) = \sum_{i=-\infty}^{+\infty} a_0(i)\phi(t-i) \tag{C.3}$$

Now, if we dilate each function of the basis and their locations by 2 we create a subspace V_1 of "coarser" or "lower resolution" signals. Iterating this procedure we generate a sequence of subspaces $\{V_j\}, (j \in Z)$ that constitute a *multiresolution analysis* of $L^2(R)$. Multiresolution theory shows that each V_j has basis $\phi_{i,j}(t) = 2^{-j/2}\phi(2^{-j}t - i)$ and is called the *approximation space* at resolution 2^{-j} or scale 2^j.

The *wavelet spaces* W_j are the orthogonal complements of V_j in V_{j-1}. They contain the necessary information to go from scale 2^j to 2^{j-1}. Each W_j has a basis composed of translated and dilated versions of the mother wavelet $\{\psi_{i,j}(t) = 2^{-j/2}\psi(2^{-j}t - i)\}$. Suppose we have, for scale j, a set of coefficients $\{a_j(i)\}$ that represent the approximation of a signal in V_j, and also the set of coefficients $\{d_j(i)\}$ representing the details of the signal in W_j. The approximation of the signal at scale $j-1$ can be computed by:

$$f_{j-1}(t) = \sum_{i=-\infty}^{+\infty} a_j(i)\phi_{i,j}(t) + \sum_{i=-\infty}^{+\infty} d_j(i)\psi_{i,j}(t) \tag{C.4}$$

If $\{\phi(t-i)\}$ and $\{\psi(t-i)\}$ are orthogonal bases of V_0 and W_0, resp., then $\{\phi_{i,j}(t)\}$ and $\{\psi_{i,j}(t)\}$ are orthogonal bases of V_j and W_j, resp. In this case, the approximation and detail coefficients of a signal at a certain scale, can be obtained by projecting the signal into the basis sets:

$$\begin{cases} a_j(i) = \langle f, \phi_{i,j}\rangle = \int_{-\infty}^{+\infty} f(t)\phi_{i,j}(t)dt \\ d_j(i) = \langle f, \psi_{i,j}\rangle = \int_{-\infty}^{+\infty} f(t)\psi_{i,j}(t)dt \end{cases} \tag{C.5}$$

Otherwise, if $\{\phi(t-i)\}$ and $\{\psi(t-i)\}$ are not orthogonal bases of V_0 and W_0, the analysis subspaces are different from the synthesis subspaces. The analysis subspaces and bases are called "dual".

Given a scaling function for V_0, multiresolution theory has developed methods to create the orthogonal wavelet basis for W_0, or the "dual" bases in the non-orthogonal case. An extensive set of wavelets has been created in the last decades. Their choice follows criteria like the number of filter coefficients or the number of vanishing moments, but we will not enter into detail here.

The orthogonal discrete wavelet transform is a very efficient signal decomposition, with many applications in signal compression and transmission, that has raised considerable interest in the computer vision area, due to the existence of very fast implementations. In the following section we will describe the main properties on the discrete wavelet transform.

C.2 The Discrete Wavelet Transform

The discrete wavelet transform (DWT) applies to discrete-time signals, and both time and scale are discretized in a dyadic fashion. The role of the approximation and wavelet functions is now played by the discrete analysis filters $g(i)$ and $h(i)$, computed from the *two-scale equations*:

$$\begin{cases} \frac{1}{\sqrt{2}}\phi(\frac{t}{2}) = \sum_i g(i)\phi(t-i) \\ \frac{1}{\sqrt{2}}\psi(\frac{t}{2}) = \sum_i h(i)\phi(t-i) \end{cases} \tag{C.6}$$

The DWT computes a set of approximation coefficients $a_J(i)$ at a scale $j = J$, and the detail coefficients $d_j(i)$ at scales $j = 1 \cdots J$:

$$\begin{cases} a_J(i) = APP(f(n); i2^J, 2^J) = \langle f, g_{i,J}\rangle = \sum_n f(n)g_{i,J}(n) \\ d_j(i) = DWT(f(n); i2^j, 2^j) = \langle f, h_{i,j}\rangle = \sum_n f(n)h_{i,j}(n) \end{cases} \tag{C.7}$$

where $g_{i,J}(n) = g(n - i2^J)$ and $h_{i,j}(n) = h(n - i2^j)$. Because the filters are translation and scale invariant, computations can be performed very efficiently through signal convolutions.

The signal can be represented by its transform coefficients as:

$$f(n) = \sum_i a_J(i)\bar{g}_{i,J}(n) + \sum_{j=1}^{J}\sum_i d_j(i)\bar{h}_{i,j}(n) \tag{C.8}$$

where the synthesis filters $\bar{g}$ and $\bar{h}$ are derived from the dual scaling function by the two-scale equation (C.6). If the bases are orthogonal, then the analysis an synthesis filters are the same.

C.3 Extension to 2D

The application of wavelets to 2D signals (images) is made with three mother wavelets and a scaling function, obtained by tensor product of the 1D functions in the horizontal and vertical directions:

$$\begin{cases} \Psi^v_{m,n,j}(x,y) = \psi_{m,j}(x)\phi_{n,j}(y) \\ \Psi^h_{m,n,j}(x,y) = \phi_{m,j}(x)\psi_{n,j}(y) \\ \Psi^d_{m,n,j}(x,y) = \psi_{m,j}(x)\psi_{n,j}(y) \\ \Phi_{m,n,j}(x,y) = \phi_{m,j}(x)\phi_{n,j}(y) \end{cases} \tag{C.9}$$

The 2D wavelets compute the vertical, horizontal and diagonal details (v, h, and d indexes, resp.), while the scaling function computes the approximation of the signal as in the one-dimensional case. Fig. C.1 shows the discrete wavelet transform of a popular test image.

A good computational property of the DWT is that both the analysis and the synthesis can be made recursively with fast convolution operations. The sequence of analysis starts with the original signal and computes the detail and approximation coefficients scale $j = 1$. Then, the detail and approximation coefficients at scale $j = 2$ are computed from the approximation coefficients at the previous level, thus, at consecutive steps, the number of coefficients to compute reduces to an half. The synthesis phase is done on reverse order.

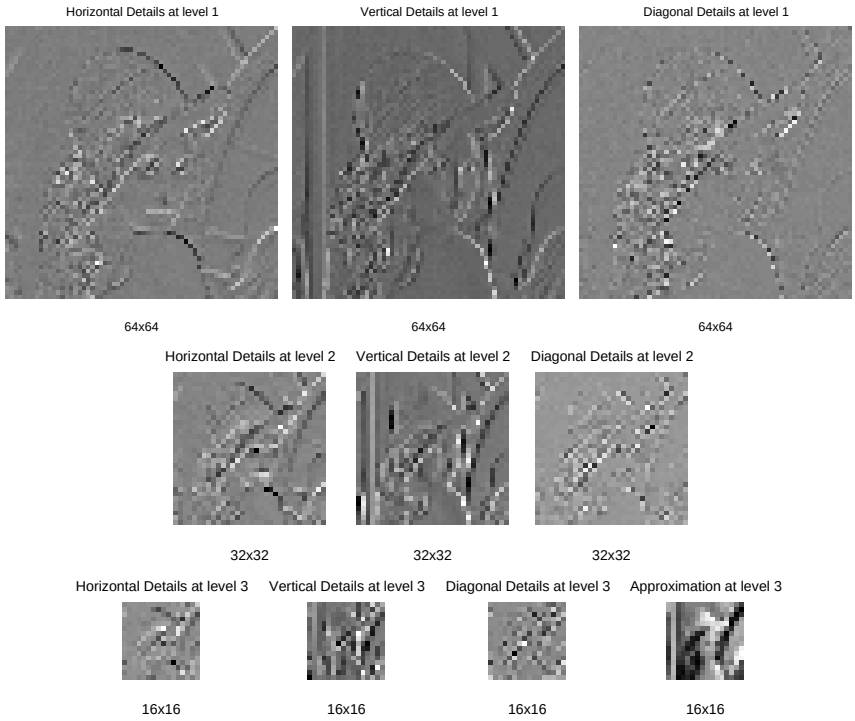


Figure C.1: Discrete Wavelet Transform coefficients of a 128×128 pixel image at 3 resolution levels, computed with a Daubechies (db1) wavelet. The total number of coefficients is equal to the number of pixels in the original image.

Appendix D

Unsampled Image Decompositions

D.1 Gaussian Decomposition and the Scale-Space

Gaussian functions are frequently used in computer vision as filters and weighting windows for diverse purposes, due to their smoothness and compactness both in space and in frequency.

A 2D normalized isotropic Gaussian function is defined as:

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2})$$

and its Fourier transform is given by:

$$\widetilde{g}(x, y, \sigma) = \exp(-\frac{(\omega_x^2 + \omega_y^2)\sigma^2}{2})$$

We will denote the parameter σ the **scale** of the function. The points of half amplitude are at distance $r = \sqrt{2\log(2)}\sigma$ from the origin in the spatial domain, and $\rho = \sqrt{2\log(2)}\sigma^{-1}$ in the frequency domain. An isotropic Gaussian function with scale 4 is illustrated in Fig. D.1.

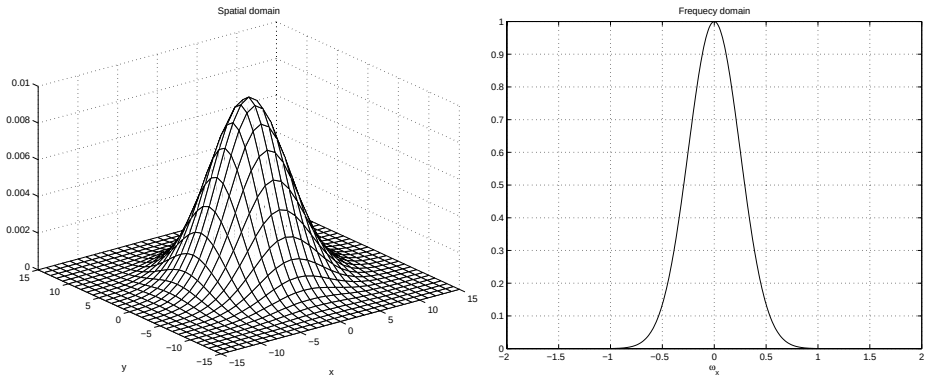


Figure D.1: Isotropic Gaussian function with scale 4. Left: 2D spatial representation. Right: 1D profile in the frequency domain.

Recently, scale-space theory [95] has formally supported the use of Gaussian functions in early image processing. A fundamental property of these functions is that combinations of convolutions with Gaussian functions can still be written as a Gaussian convolution with

a different scale. Thus, Gaussian functions with scale free parameter generate a scale-space structure under the convolution operation. In practice, this means that Gaussian functions generate no *spurious resolution effects* when zooming in and out on pixels [28], providing a sort of scale invariance.

Measurements from an image are taken not only at a spatial position, but also at a certain scale. If for example one is searching for large sized objects in the image, one should look at parts of the scale-space with high σ. On the contrary, if one in interested in analysing fine structure details, one should take measurements of lower scales. The scale-space of an image is a 3 dimensional volume indexed by spatial coordinates and a scale coordinate. It can be represented mathematically by:

$$F(x, y, \sigma) = f(x, y) * g(x, y, \sigma) \tag{D.1}$$

where $f(x, y)$ is the image to analyse and $*$ is the linear convolution operator.

The scale-space of an image has a continuous domain. For practical purposes, on a discretized version is used instead. The Gaussian Decomposition of an image is composed by samples of the continuous scale-space at discrete values of scale:

$$G(x, y, \sigma_i) = F(x, y, \sigma_i), i = 1 \cdots S \tag{D.2}$$

where S is the number of sampled scales. Fig. D.2 shows a Gaussian decomposition with scales $2^i, i = 0, \cdots, 4$. When scale values are related by powers of 2, we call the

Figure D.2: Example of a isotropic Gaussian decomposition with dyadic scales.

decomposition **dyadic**.

In the previous example, the frequency content of all levels of the decomposition have a low-pass nature. Considering the bandwidth of an isotropic Gaussian function as the value of the radial frequency where the amplitude halves, then level i has bandwidth $\beta_i = \sqrt{2\log(2)}2^{-i}$. Fig. D.3 shows the frequency response of the given filter bank. Notice

that in a dyadic decomposition the bandwidth of each level halves with increasing scale, i.e, with a logarithmic frequency graphical representation, the cut-off frequencies are equally spaced.

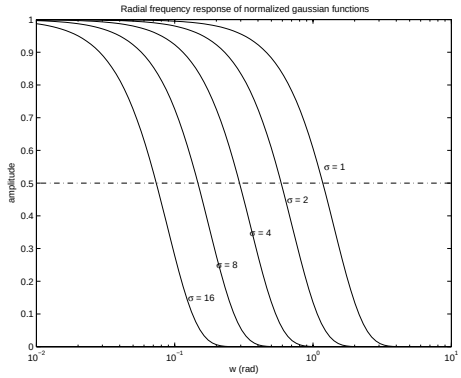


Figure D.3: Fourier transform magnitude of a set of dyadic Gaussian functions.

Although in a Gaussian decomposition, the notion of different scales and resolutions is clearly visible, this decomposition is very redundant in spectral terms, since the image frequency content of a certain level is present in all the lower levels. Next section is about a more efficient class of decompositions.

D.2 Sub-Band Decompositions

In a general setting, sub-band decomposition methods consist in image filtering with sets of band-pass filters tuned to different parts of the spectrum. Let $f(x, y)$ be a 2D signal with Fourier transform $\widetilde{f}(\omega_x, \omega_y)$ and $\mathcal{H} = \{h_i(x, y), i = 1 \cdots N\}$ be a set of band-pass filters with Fourier transform $\widetilde{h_i}(\omega_x, \omega_y)$. The image sub-band decomposition is composed by the set $\mathcal{F} = \{f_i(x, y), i = 1 \cdots N\}$ where each $f_i(x, y)$ is a *sub-band image*:

$$f_i(x, y) = f(x, y) \star h_i(x, y)$$

with Fourier transform:

$$\widetilde{f_i}(\omega_x, \omega_y) = \widetilde{f}(\omega_x, \omega_y) \cdot \widetilde{h}_i(\omega_x, \omega_y)$$

If all frequencies are well represented by the filter set, then it is said to be **complete**. If each filtering step can be implemented by cascaded convolutions in the x and y directions with 1D filters, the filter set is said to be **separable**. Separable filter sets are computationally more efficient.

A very interesting choice for the filter set is when $\sum \widetilde{h_i} = 1$. In this case the filter set is **complementary** and a very efficient algorithm exists to reconstruct the image from its sub-bands: just add all levels of the decomposition:

$$f(x, y) = \sum_i f_i(x, y)$$

The Laplacian decomposition is motivated on the Laplacian Pyramid but works on unsampled domains. It consists on differencing consecutive levels of the *Gaussian Decomposition*. The filters that generate this decomposition are composed by *Difference-*

Of-Gaussians (DOG), that split the image frequency content into sub-bands. A practical algorithm to generate the Laplacian decomposition can be as follows:

1. Split image $f(x, y)$ into S Gaussian levels, by convolution with Gaussian kernels $\{g_i(x, y)\}, i \in \{1, \cdots, S\}$, of variance σ_i, creating a Gaussian decomposition $f_i(x, y)$. A common choice is to have dyadic levels, where the Gaussian kernel size doubles from level to level.

2. Subtract consecutive Gaussian levels, thus obtaining separate frequency bands:

$$\begin{cases} l_0(x, y) = f(x, y) - f_1(x, y) \\ l_i(x, y) = f_i(x, y) - f_{i+1}(x, y), \quad i \in \{1, \cdots, S-1\} \\ l_S(x, y) = f_S(x, y) \end{cases} \tag{D.3}$$

The additional low-pass (l_0) and high-pass (l_S) levels are included for spectral completeness.

Fig. D.4 shows the referred sub-band decomposition applied to the same image as in Fig. D.2.

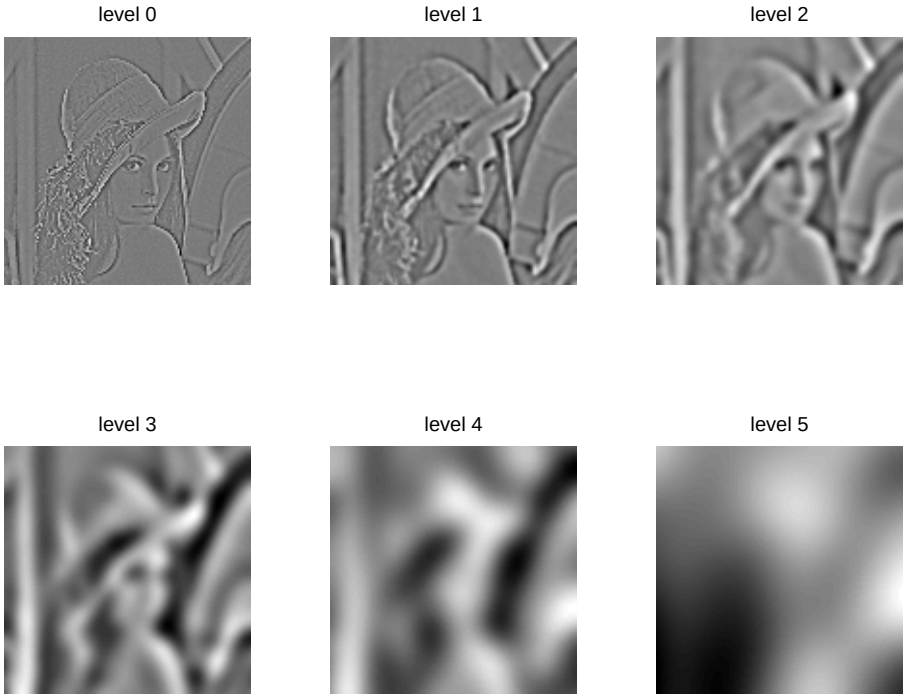


Figure D.4: Sub-band decomposition of test image. Levels $1-4$ are the sub-band images while levels 0 and 5 are the high-pass and low-pass residual levels, resp.

The set of operations in the previous algorithm is equivalent to image filtering with a set of band-pass filters of the type *Difference-Of-Dyadic-Gaussians* (DODG) and additional low-pass and high-pass filters:

$$\begin{cases} h_0(x, y) = \delta(x, y) - g_1 x, y \\ h_i(x, y) = g_i(x, y) - g_{i+1}(x, y), \quad i \in \{1, \cdots, S-1\} \\ h_S(x, y) = g_S(x, y) \end{cases} \tag{D.4}$$

Given that a Gaussian function of variance σ^2 has negligible frequency content after frequency $3/\sigma$, the maximum spatial frequency of each filter i is given by:

$$\begin{cases} \omega_{max}(0) = \pi \\ \omega_{max}(i) = \frac{3}{\sigma_i}, \quad i \in \{1, \cdots, S\} \end{cases} \tag{D.5}$$

In a dyadic decomposition, where $\sigma_i = 2^{i-1}, \quad i = 1 \cdots S$, we have:

$$\begin{cases} \omega_{max}(0) = \pi \\ \omega_{max}(i) = \frac{3}{2^{i-1}}, \quad i \in \{1, \cdots, S\} \end{cases} \tag{D.6}$$

Fig. D.5 shows the radial frequency response of such filter set. This image decompo-

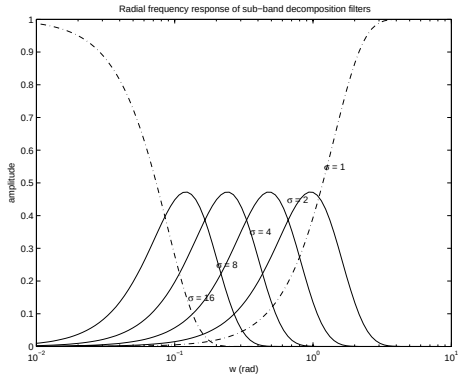


Figure D.5: Frequency response of a sub-band decomposition obtained from a set of *Difference-Of-Gaussians* filters, with residual low-pass and high-pass filters for spectral completeness (the sum of all *spectra* is the unit function).

sition is commonly named "Laplacian" for historical reasons. The reason is that DODG filters are similar in shape to the differential operator *Laplacian* applied to Gaussian functions.

D.3 Fast Approximations

Efficient recursive algorithms have been proposed to decompose one image into dyadic Gaussian or Laplacian decompositions with sub-sampling [30]. In the usual approach, an image pyramid is created, by successively filtering the previous level with fixed size separable Gaussian-like FIR filters, and sub-sampling by 2 both dimensions of the image.

In pyramid implementations, a great deal of computation reduction is due to the sub-sampling step, where at each level image size is reduced by a factor of 4 and filters are small. In the unsampled case, not only image size is kept constant from level to level but also the size of the filters must increase to generate the large scales. An efficient algorithm to address this problem is the *à trous* algorithm [99]. The *à trous* algorithm is a recursive technique to implement filters of increasing scale but with a constant number of coefficients. Is is based on upsampled filters, obtained from the base filter by inserting zeros between samples, and applied recursively on the original signal. Although not all filter sets can be implemented by this technique, if the base filter coefficients are properly chosen, we can obtain good approximations to quasi-dyadic Gaussian filters.

To illustrate this, let us consider an image $f(x, y)$ and low-pass filter $q^0(x, y)$ with Fourier transform $\widetilde{q}^0(\omega_x, \omega_y)$. The first step of the unsampled *à trous* algorithm consists in obtaining $\mathbf{f}^1$, the low-pass version of $\mathbf{f}^0$:

$$\mathbf{f}^1 = \mathbf{f}^0 * \mathbf{w}^0 \tag{D.7}$$

In the next decomposition level a new filter is created by expanding the previous one with zero insertion:

$$q^1(x, y) = \begin{cases} q^0(\frac{x}{2}, \frac{y}{2}), & x, y \text{ even} \\ 0, & \text{otherwise} \end{cases} \tag{D.8}$$

which, in the frequency domain, corresponds to a spectral compression:

$$\widetilde{q}^1(\omega_x, \omega_y) = \widetilde{q}^0(2\omega_x, 2\omega_y) \tag{D.9}$$

The new low-pass signal computed by:

$$\mathbf{f}^2 = \mathbf{f}^1 * \mathbf{q}^1 \tag{D.10}$$

and the procedure goes on recursively until the last scale level is reached. Since the convolution operation is linear, the low-pass signal at the $i + 1$ level can be written as:

$$\mathbf{f}^{i+1} = \mathbf{f}^0 * \mathbf{q}^0 * \mathbf{q}^1 * \cdots * \mathbf{q}^i \tag{D.11}$$

This is equivalent to filter the original signal $\mathbf{f}^0$ with filters $\mathbf{w}^i$ resulting from successive convolutions of the several $\mathbf{q}^k$:

$$\mathbf{w}^i = \prod_{k=0}^{i} * \mathbf{q}^k \tag{D.12}$$

where the symbol $\prod *$ represents the composition of convolution operations. The Fourier transforms of the equivalent filters are given by:

$$\widetilde{w}^i(\omega_x, \omega_y) = \prod_{k=0}^{i} \widetilde{q}^0(2^k\omega_x, 2^k\omega_y) \tag{D.13}$$

In [30], several base filters $\mathbf{q}^0$ are tested. Not all choices are unimodal or resemble Gaussian functions. The 2D base filters are generated by the tensor product of 1D filters:

$$q^0(x, y) = q^0_x(x) \cdot q^0_y(y)$$

The following 1D filter is proposed to generate a set of equivalent filters similar to dyadic Gaussian functions:

$$q^0_x(x) = \begin{cases} 0.05, x \in \{-2, 2\} \\ 0.25, x \in \{-1, 1\} \\ 0.40, x = 0 \\ 0, \text{otherwise} \end{cases} \tag{D.14}$$

In the frequency domain, this filter has Fourier transform:

$$\widetilde{q_x}^0(\omega_x) = 0.4 + 0.5\cos(\omega_x) + 0.1\cos(2\omega_x) \tag{D.15}$$

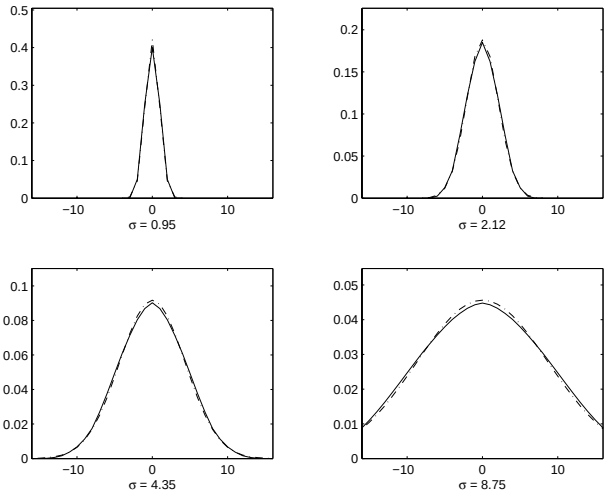


Figure D.6: Solid lines represent the equivalent filters $\mathbf{w}^i$ of the *à trous* decomposition with base filter $\mathbf{q}_x^0$. Dotted lines show Gaussian filters with equivalent variances.

Scale	1	2	4	8
Bandwidth (*à trous*)	1.21	5.46×10^{-1}	2.67×10^{-1}	1.33×10^{-1}
Bandwidth (Gaussian)	1.24	5.54×10^{-1}	2.71×10^{-1}	1.35×10^{-1}

Table D.1: Half-amplitude bandwidth values (in rad/sec) for the first 4 dyadic scales. Comparison between Gaussian filters and the *à trous* decomposition.

Although not being exactly dyadic (standard deviations are $0.95, 2.12, 4.35, 8.75, \cdots$), these filters have similar half frequency bandwidth and sufficient attenuation at high frequencies. In Figures D.6 and D.7, we can compare their shapes with Gaussian filters of equivalent variance, in the spatial and frequency domains, for the 1D case. Table D.3 compares the half-amplitude frequency bandwidths for the first 4 levels. Thus, to create an approximate Gaussian decomposition, we apply the unsampled *à trous* algorithm with base filter defined before. The approximate DOGD decomposition is obtained using Eq. D.3

Performance Analysis

To date, the fastest method for isotropic Gaussian filtering is presented in [162]. It is based on a 2 pass, cascaded forward-backward recursive implementation, with separable infinite impulse response (IIR) filters. In the 2-dimensional case, its most economic version requires 26 operations per pixel (3 multiplications and 3 additions per dimension per pass). This cost is independent of the scale parameter and any scale is allowed. By the contrary, the proposed unsampled *à trous* algorithm with 5 tap symmetric base filter requires only 14 operations per pixel (3 multiplications and 4 additions per dimension), thus resulting in 45% computational savings. To evaluate the quality of the proposed approximation, we have compared the approximate method (with the 4 level *à trous* decomposition) and the method of [162] with true dyadic scales ($\sigma = 1, 2, 4, 8$). We applied both methods to the test images from the *miscellaneous*, *aerial* and *texture* classes of the USC-SIPI image database [165], converted to grayscale and resized to 128×128 pixel sizes. We computed the signal to approximation ratios, and the results are shown in Fig. D.8. For all images, the approximation error is smaller than 30 dB ($\approx 3\%$).

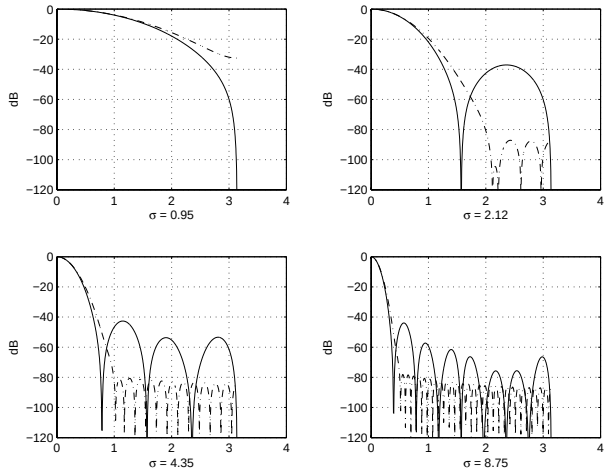


Figure D.7: Solid lines represent the spectra of the equivalent filters $\mathbf{w}^i$. Dotted lines show the spectra of Gaussian filters with equivalent variances.

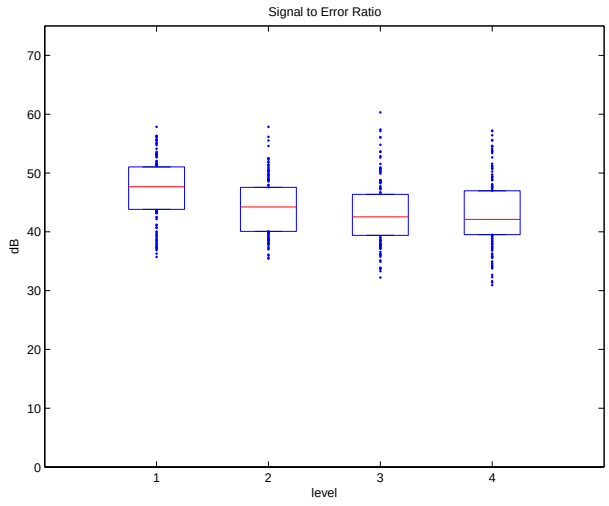


Figure D.8: Ratio between image energy and approximation error energy for Gaussian filtering with the proposed method. Energy is computed as sum-of-squares. Each box corresponds to a level of the decomposition and shows the median, upper and lower quartiles of the data. Other values are shown as small dots.

Appendix E

Fast Gabor Filtering

E.1 Definition and Background

Let x, y be the discrete spatial coordinates and $w_\sigma(x, y)$ be a two dimensional Gaussian envelope with scale parameter $\sigma = (\sigma_1, \sigma_2, \theta)$. The standard deviations σ_1 and σ_2 are oriented along directions θ and $\theta + \pi/2$, respectively:

$$w_\sigma(x, y) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left[-\frac{(x\cos\theta + y\sin\theta)^2}{2\sigma_1^2} - \frac{(y\cos\theta - x\sin\theta)^2}{2\sigma_2^2}\right] \tag{E.1}$$

Let $c_{\lambda,\theta}(x, y)$, be a complex exponential carrier representing a plane wave with wavelength λ and orientation θ:

$$c_{\lambda,\theta}(x, y) = \exp\left[i\frac{2\pi}{\lambda}(x\cos\theta + y\sin\theta)\right] \tag{E.2}$$

To simplify notation, we will drop the pixel coordinates (x, y) whenever they are not required and write in bold face all functions of the spatial coordinates. With this notation, a two dimensional Gabor function is written as:

$$\mathbf{g}_{\sigma,\lambda,\theta} = \mathbf{w}_\sigma \cdot \mathbf{c}_{\lambda,\theta} \tag{E.3}$$

This function has non zero mean value (is a non admissible wavelet), which is not desirable for the purpose of feature extraction and multi-scale analysis. In practice, the zero-mean version is preferred [93]:

$$\gamma_{\sigma,\lambda,\theta} = \mathbf{w}_\sigma \cdot (\mathbf{c}_{\lambda,\theta} - k_{\sigma,\lambda,\theta}) \tag{E.4}$$

The parameter $k_{\sigma,\theta,\lambda}$ is set to remove the Gabor function DC value, i.e. $\widetilde{\gamma}(0, 0) = 0$:

$$k_{\sigma,\theta,\lambda} = \frac{\widetilde{w}_\sigma(-\frac{2\pi\cos\theta}{\lambda}, -\frac{2\pi\sin\theta}{\lambda})}{\widetilde{w}_\sigma(0, 0)}$$

where $\widetilde{w}$ denotes the Fourier transform of w. To distinguish between the two functions, we call **Gabor function** to the non-zero-mean function and **Gabor wavelet** to the zero-mean function.

Mathematically, the convolution of an image $\mathbf{f}$ with a Gabor wavelet $\gamma_{\sigma,\lambda,\theta}$ is written as:

$$\mathbf{z}_{\sigma,\lambda,\theta} = \mathbf{f} * \gamma_{\sigma,\lambda,\theta} \tag{E.5}$$

and can be computed by the discrete convolution formula:

$$z_{\sigma,\lambda,\theta}(x,y) = \sum_{k,l} f(k,l) \cdot \gamma_{\sigma,\lambda,\theta}(x-k, y-l) \tag{E.6}$$

Replacing in Eq. (E.5) the definition of the Gabor wavelet (E.4), we get:

$$\mathbf{z}_{\sigma,\lambda,\theta} = \mathbf{f} * \mathbf{g}_{\sigma,\lambda,\theta} - \mathbf{f} * k_{\sigma,\lambda,\theta}\mathbf{w}_{\sigma} \tag{E.7}$$

Thus, image convolution with a Gabor wavelet can be implemented by subtracting two terms: the convolution with a Gabor function and the convolution with a scaled Gaussian function.

E.2 The Isotropic Case

In the isotropic case we have $\sigma_1 = \sigma_2$. The isotropic Gaussian envelope is defined by:

$$w_{\sigma}(x,y) = \frac{1}{2\pi\sigma^2} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \tag{E.8}$$

Thus, the isotropic Gabor wavelet of scale σ, orientation θ and wavelength λ is defined in the spatial domain as:

$$\gamma_{\sigma,\theta,\lambda}(x,y) = w_{\sigma}(x,y) \cdot (c_{\theta,\lambda} - k_{\sigma,\theta,\lambda})$$

and, in the frequency domain, has the following representation:

$$\widetilde{\gamma}_{\sigma,\theta,\lambda}(\Omega_x,\Omega_y) = \widetilde{w}_{\sigma}\left(\Omega_x - \frac{2\pi\cos\theta}{\lambda}, \Omega_y - \frac{2\pi\sin\theta}{\lambda}\right) - k_{\sigma,\theta,\lambda}\widetilde{w}_{\sigma}\left(\Omega_x,\Omega_y\right) \tag{E.9}$$

The motivation to consider the isotropic case comes from the fact that efficient separable implementations exist for convolution with Gaussian and Gabor functions. Both can be written as the tensor product of vertical and horizontal 1D filters. Gaussian functions are decomposed by:

$$w_{\sigma}(x,y) = \overbrace{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{x^2}{2\sigma^2}\right]}^{w'_{\sigma}(x)} \cdot \overbrace{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{y^2}{2\sigma^2}\right]}^{w'_{\sigma}(y)} \tag{E.10}$$

and Gabor functions are written as:

$$g_{\sigma,\lambda,\theta}(x,y) = \underbrace{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{x^2}{2\sigma^2} + i\frac{2\pi x\cos\theta}{\lambda}\right]}_{g'_{\sigma,\lambda,\theta}(x)} \cdot \underbrace{\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{y^2}{2\sigma^2} + i\frac{2\pi y\sin\theta}{\lambda}\right]}_{g'_{\sigma,\lambda,\theta}(y)} \tag{E.11}$$

Image convolution with such functions can be performed with two cascaded (horizontal and vertical) 1D convolutions with complexity $O(N \cdot M)$ each, where N is the number of image pixels and M is the number of filter coefficients. For example, Gaussian filtering can be implemented by:

$$f(x,y) * w_{\sigma}(x,y) = w'_{\sigma}(y) * w'_{\sigma}(x) * f(x,y) \tag{E.12}$$

The fastest (to date) implementations of convolution with Gaussian and Gabor functions are described in [162] and [163]. In [162] a recursive separable Gaussian filter requires 7 real multiplications and 6 real additions per pixel per dimension. The extension to 2-dimensional signals thus needs 26 operations. In [163] a recursive separable Gabor filter was developed, requiring 7 complex multiplications and 6 complex additions per pixel per dimension. With 2-dimensional signals, this implementation needs about 108 operations[1]. Therefore, image convolution with Gabor wavelets consists in 1 Gaussian filtering, 1 Gabor filtering, 1 multiplication and one addition and corresponds to a total of 136 operations per pixel[2]

E.3 Filter Decomposition

Let $\mathbf{z}^c_{\sigma,\lambda,\theta}$ denote the result of image convolution with Gabor functions:

$$z^c_{\sigma,\lambda,\theta}(x,y) = \sum_{k,l} f(k,l) \cdot w_\sigma(x-k, y-l) \cdot c_{\lambda,\theta}(x-k, y-l) \tag{E.13}$$

Using the definition of the Gabor wavelet (E.3) in (E.5) we get:

$$\mathbf{z}_{\sigma,\lambda,\theta} = \mathbf{f} * (\mathbf{w}_\sigma \cdot \mathbf{c}_{\lambda,\theta}) - k_{\sigma,\lambda,\theta} \cdot \mathbf{f} * \mathbf{w}_\sigma \tag{E.14}$$

The Gaussian convolution in the last term, $\mathbf{f} * \mathbf{w}_\sigma$, is denoted by $\mathbf{z}^w_\sigma$ and can be computed via:

$$z^w_\sigma(x,y) = \sum_{k,l} f(k,l) \cdot w_\sigma(x-k, y-l) \tag{E.15}$$

The first term, $\mathbf{f} * (\mathbf{w}_\sigma \cdot \mathbf{c}_{\lambda,\theta})$, corresponds to a convolution with a Gabor function and is denoted $\mathbf{z}^c_{\sigma,\lambda,\theta}$:

$$z^c_{\sigma,\lambda,\theta}(x,y) = \sum_{k,l} f(k,l) \cdot w_\sigma(x-k, y-l) \cdot c_{\lambda,\theta}(x-k, y-l) \tag{E.16}$$

Since the complex exponential function $\mathbf{c}_{\lambda,\theta}$ is separable, we can expand the previous expression into:

$$z^c_{\sigma,\lambda,\theta}(x,y) = c_{\lambda,\theta}(x,y) \cdot \sum_{k,l} \bar{c}_{\lambda,\theta}(k,l) \cdot f(k,l) \cdot w_\sigma(x-k, y-l) \tag{E.17}$$

where $\bar{c}$ denotes complex conjugation. Writing in compact form, we have:

$$\mathbf{z}^c_{\sigma,\lambda,\theta} = \mathbf{c}_{\lambda,\theta} \cdot [(\mathbf{f} \cdot \bar{\mathbf{c}}_{\lambda,\theta}) * \mathbf{w}_\sigma] \tag{E.18}$$

Finally, the full filtering operation (E.7) can be written:

$$\mathbf{z}_{\sigma,\lambda,\theta} = \mathbf{c}_{\lambda,\theta} \cdot [(\mathbf{f} \cdot \bar{\mathbf{c}}_{\lambda,\theta}) * \mathbf{w}_\sigma] - k_{\sigma,\lambda,\theta} \cdot (\mathbf{f} * \mathbf{w}_\sigma) \tag{E.19}$$

A graphical representation of the method is depicted in Fig. E.1.

[1] we consider 1 complex multiplication equal to 4 real multiplications plus 2 real additions

[2] Notice that the implementation in [163] reports to **non zero mean** Gabor functions, while we are interested in **zero mean** Gabor wavelets. No direct separable implementation of zero mean Gabor wavelets has been described in the literature.

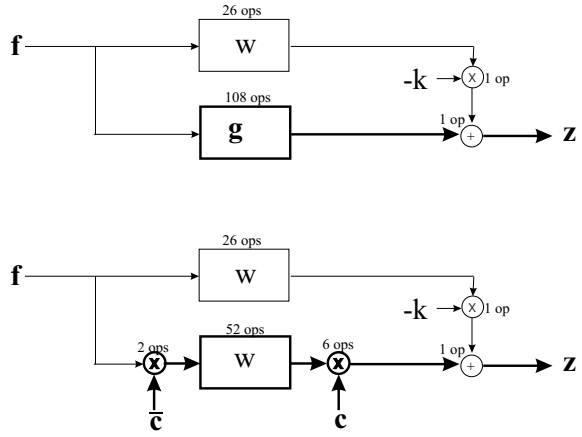


Figure E.1: The single-scale, single carrier case: convolution with Gabor wavelets, using state-of-the-art Gabor and Gaussian filters, require 136 operations per pixel (top), while the proposed equivalent decomposition method (bottom), only requires 88. Thick/Thin lines and boxes represent complex/real signals and filters, respectively.

Considering the isotropic case, we can adopt the IIR Gaussian filtering implementation of [162] (26 operations per pixel), and the required computations on Eq. (E.19), are:

- A **modulation** (product of $\mathbf{f}$ with $\bar{\mathbf{c}}_{\lambda,\theta}$) is computed by multiplying one real image and one complex image, corresponding to **2 operations** per pixel.
- A **complex Gaussian filtering** (convolution of $\mathbf{w}_\sigma$ with $\mathbf{f} \cdot \bar{\mathbf{c}}_{\lambda,\theta}$) requires **52 operations** per pixel.
- A **demodulation** operation (product of $\mathbf{c}_{\lambda,\theta}$ with $(\mathbf{f} \cdot \bar{\mathbf{c}}_{\lambda,\theta}) * \mathbf{w}_\sigma$) requires 1 complex multiplication per pixel, corresponding to **6 operations** per pixel.
- A **real Gaussian filtering** ($\mathbf{f} * \mathbf{w}_\sigma$) requiring **26 operations** per pixel.
- A **real scaling** by $k_{\sigma,\lambda,\theta}$, requires **1 operation** per pixel.
- The **final subtraction**, corresponds to only **1 operation** per pixel because only the real part of Gabor functions have non zero DC value.

Altogether we have 88 operations which, in comparison with the reference value of 136 operations, correspond to about 35% savings in computation.

When multiple carriers (orientations/wavelengths) are considered, the term $\mathbf{f} * \mathbf{w}_\sigma$ in (E.19) is common to all of them. A graphical representation of the method is shown in Fig. E.2 for the single-scale-multiple-carrier case. With regard to the number of operations, image Gaussian filtering contributes with 26 operations per pixel and each carrier contributes with additional 62 operations per pixel, in our proposal, or 110 operations per pixel, with direct Gabor filtering. If, for example, 4 orientations and 2 wavelengths are used, the total number of operations is $8 \times 62 + 26 = 522$ per pixel *vs* $8 \times 110 + 26 = 906$ per pixel, representing about 42% savings. It is also worth mentioning that multi-scale image decomposition architectures most often compute image Gaussian expansions to support further processing [30, 42], and the intermediate Gaussian filtered images $\mathbf{f} * \mathbf{w}_\sigma$ may already have been computed by the system, thus saving extra 26 operations per pixel.

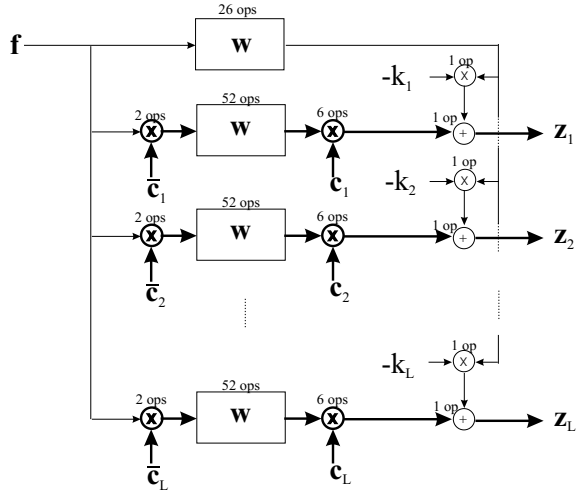


Figure E.2: Proposed Gabor filtering scheme in the single-scale multi-carrier case. Thick/Thin lines and boxes represent complex/real signals and filters, respectively. Close to each computational element we indicate the number of real operations required.

E.4 Isotropic Gaussian Filtering

Our implementation of one-dimensional Gaussian filtering is based on [162]. Convolution with Gaussian functions is approximated by a cascaded two pass filtering operation. First, the original signal $f(t)$ is filtered in the forward direction by causal filter $w_\sigma^f(t)$:

$$f^f(t) = f(t) * w_\sigma^f(t) \tag{E.20}$$

Second, the resulting signal $f^f(t)$ is convolved in the backward direction with the anti-causal filter $w_\sigma^b(t)$:

$$f^b(t) = f^f(t) * w_\sigma^b(t) \tag{E.21}$$

We use forward and backward infinite-impulse-response (IIR) filters with 3 poles each, defined by the Z transforms:

$$\begin{cases} \widetilde{w}_\sigma^f(z) = \frac{b_0}{1+a_1 z^{-1}+a_2 z^{-2}+a_3 z^{-3}} \\ \widetilde{w}_\sigma^b(z) = \frac{b_0}{1+a_1 z+a_2 z^2+a_3 z^3} \end{cases} \tag{E.22}$$

Thus the full 1D filter is represented by:

$$\widetilde{w}'(z) = \frac{b_0^2}{\left(1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}\right)\left(1 + a_1 z + a_2 z^2 + a_3 z^3\right)} \tag{E.23}$$

or, factorizing the denominator into first order terms:

$$\widetilde{w}'(z) = \frac{b_0^2}{\left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right)\left(1 - p_3 z^{-1}\right)\left(1 - p_1 z\right)\left(1 - p_2 z\right)\left(1 - p_3 z\right)} \tag{E.24}$$

The filter coefficients, b_0, a_1, a_2 and a_3, and filter poles, p_1, p_2 and p_3, are function of the scale σ. Formulas to compute their values are provided in [162].

In the time domain, the filtering operation is implemented recursively:

$$\begin{cases} f^f(t) = b_0 f(t) - a_1 f^f(t-1) - a_2 f^f(t-2) - a_3 f^f(t-3) & \text{(forward pass)} \\ f^b(t) = b_0 f^f(t) - a_1 f^b(t-1) - a_2 f^b(t-2) - a_3 f^b(t-3) & \text{(backward pass)} \end{cases} \tag{E.25}$$

2D Gaussian filtering is implemented by cascading horizontal and vertical 1D filters. The full 2D Gaussian filter is represented in the time domain by:

$$w_\sigma(x,y) = w'_\sigma(x) * w'_\sigma(y) \tag{E.26}$$

and, in the frequency domain, by:

$$\widetilde{w}_\sigma(e^{i\Omega_x}, e^{i\Omega_y}) = \widetilde{w}'_\sigma(e^{i\Omega_x})\widetilde{w}'_\sigma(e^{i\Omega_y}) \tag{E.27}$$

E.5 Boundary Conditions

Let us define:

$$f^{00}(x,y;\omega_x,\omega_y) = f(x,y) \cdot e^{i\cdot(\omega_x x + \omega_y y)} \tag{E.28}$$

where $f(x,y)$ is the original image and ω_x, ω_y are the horizontal and vertical frequencies of the complex exponential carrier:

$$\begin{cases} \omega_x = \frac{2\pi}{\lambda}\cos(\theta) \\ \omega_y = \frac{2\pi}{\lambda}\sin(\theta) \end{cases} \tag{E.29}$$

Our approach to Gabor filtering involves Gaussian filtering of images $f^{00}(x,y;\omega_x,\omega_y)$ for several particular values of ω_x and ω_y. Here we will derive the boundary conditions to initialize the Gaussian filters, for the general class of images $f^{00}(x,y;\omega_x,\omega_y)$.

The full 2D Gaussian filtering operation is implemented by cascaded forward-backward passes in the horizontal and vertical directions, using the one dimensional forward and backward filters $w^f(t)$ and $w^b(t)$, respectively. Each one-dimensional filtering operation is implemented recursively by convolving the filter with a signal defined in the domain $t \in 0, \cdots, N-1$. The following operations and boundary conditions are used in our implementation:

- Forward pass

$$\begin{cases} y(t) = b_0 x(t) - a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3), \quad t \geq 0, \\ y(-1) = y_{-1} \\ y(-2) = y_{-2} \\ y(-3) = y_{-3} \end{cases} \tag{E.30}$$

- Backward pass

$$\begin{cases} z(t) = b_0 y(t) - a_1 z(t+1) - a_2 z(t+2) - a_3 z(t+3), \quad t \leq N-1, \\ z(N) = z_N \\ z(N+1) = z_{N+1} \\ z(N+2) = z_{N+2} \end{cases} \tag{E.31}$$

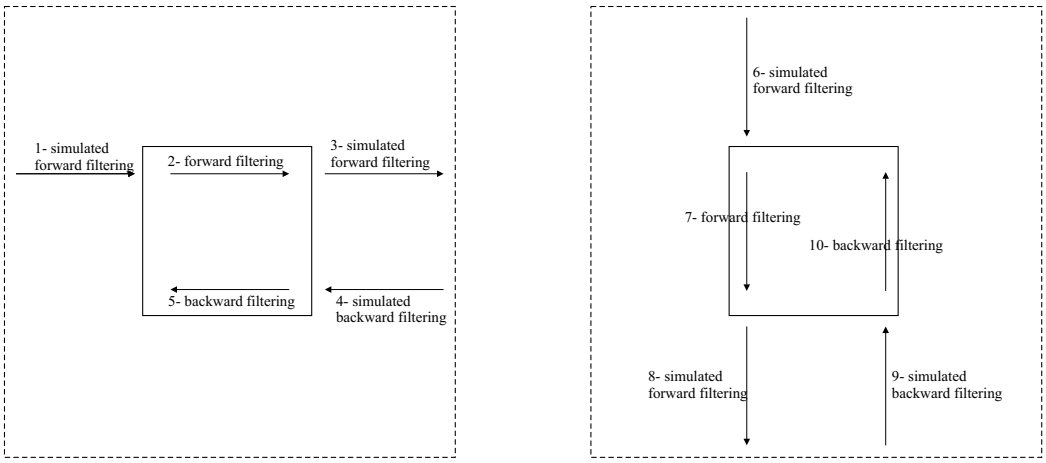


Figure E.3: Sequence of filtering operations, first in the horizontal directions (left) and then in the vertical direction (right). In boundary regions, simulated filtering operations are performed to compute the initial conditions for real filtering operations. It is considered that the boundary values replicate the first and last values at each line/column of the original image, $f(x, y)$.

In the 2D case, the filtering operations are applied to the image $f^{00}(x, y; \omega_x, \omega_y)$, sequentially in the directions left-right (horizontal-forward pass), right-left (horizontal-backward pass), top-down (vertical-forward pass) and bottom-up (vertical-backward pass). For each pass we need to compute appropriate initial conditions in the boundary and this will be made by virtually extending the boundaries of the original image from $-\infty$ to $+\infty$, and computing the response of the filters to the boundary signals using frequency domain and Z transform methods, instead of explicit filtering. The sequence of operations is illustrated in Fig. E.3.

In the following, we will denote the filtering dimension by t, and the other (fixed) dimensions by x or y. The cascaded filtering operations generate the sequence of images:

$$\begin{cases} f^{f0}(t, y) = w^f(t) * f^{00}(t, y), & \text{horizontal forward pass} \\ f^{b0}(t, y) = w^b(t) * f^{f0}(t, y), & \text{horizontal backward pass} \\ f^{bf}(x, t) = w^f(x) * f^{b0}(x, t), & \text{vertical forward pass} \\ f^{bb}(x, t) = w^b(x) * f^{bf}(x, t), & \text{vertical backward pass} \end{cases} \tag{E.32}$$

where we have dropped arguments ω_x and ω_y, to simplify notation. The purpose of the current analysis is to determine the initial conditions in each case, i.e., the values:

$$\begin{cases} f^{f0}(-1, y), f^{f0}(-2, y), f^{f0}(-3, y), & \text{horizontal forward pass} \\ f^{b0}(N, y), f^{b0}(N+1, y), f^{b0}(N+2, y), & \text{horizontal backward pass} \\ f^{bf}(x, -1), f^{bf}(x, -2), f^{bf}(x, -3), & \text{vertical forward pass} \\ f^{bb}(x, N), f^{bb}(x, N+1), f^{bb}(x, N+2), & \text{vertical backward pass} \end{cases} \tag{E.33}$$

Horizontal Forward Pass

The initial conditions for the horizontal forward pass are obtained considering constancy in the boundary of the original image. This means that the boundary values if the input

image can be seen as complex exponential functions:

$$f^{00}(t,y) = f(0,y) \cdot e^{i\cdot(\omega_x t+\omega_y y)},\ t<0 \tag{E.34}$$

Thus, the output signal in the boundary can be determined by multiplying the input signal by the frequency response of the forward pass filter:

$$f^{f0}(t,y) = |\widetilde{w}^f(e^{i\omega_x t})| f(0,y) \cdot e^{i(\omega_x t+\omega_y y+\angle\widetilde{w}^f(e^{i\omega_x t}))},\ t<0 \tag{E.35}$$

In conclusion, the initial conditions in the forward horizontal pass are given by;

$$\begin{cases} f^{f0}(-1,y) = |\widetilde{w}^f(e^{-i\omega_x})| f(0,y) \cdot e^{i(-\omega_x+\omega_y y+\angle\widetilde{w}^f(e^{-i\omega_x}))} \\ f^{f0}(-2,y) = |\widetilde{w}^f(e^{-2i\omega_x})| f(0,y) \cdot e^{i(-2\omega_x+\omega_y y+\angle\widetilde{w}^f(e^{-2i\omega_x}))} \\ f^{f0}(-3,y) = |\widetilde{w}^f(e^{-3i\omega_x})| f(0,y) \cdot e^{i(-3\omega_x+\omega_y y+\angle\widetilde{w}^f(e^{-3i\omega_x}))} \end{cases} \tag{E.36}$$

Horizontal Backward Pass

First we will consider the simulated forward pass in the right boundary. Again assuming boundary constancy in the original image, the right boundaries can be described by:

$$f^{00}(t,y) = f(N-1,y) \cdot e^{i\cdot(\omega_x t+\omega_y y)},\ t>N-1 \tag{E.37}$$

Now, let the forward filtering pass continue through the right boundary. The resulting signal can be computed by:

$$f^{f0}(t,y) = w^f(t) * \left[f(N-1,y) \cdot e^{i\cdot(\omega_x t+\omega_y y)} \right],\ t>N-1 \tag{E.38}$$

The initial conditions for this filtering step are provided by the values already computed of $f^{f0}(t,y)$, for $t \in \{N-3, N-2, N-1\}$. The solution can be obtained in a elegant way with the unilateral Z transform. To simplify notation let us represent the input and output signals by $x(t)$ and $y(t)$ respectively, and shift the origin of coordinates to the right boundary ($t=N$). The new input signal is defined by:

$$x(t) = x_0 \cdot e^{i\cdot(\omega_x t-\omega_x N+\omega_y y)},\ t\geq 0 \tag{E.39}$$

where $x_0 = f(N-1,y)$. Now, the forward filtering operation on the right boundary can be represented by the following difference equation:

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) = b_0 x(t) \tag{E.40}$$

with initial conditions $y_1 = f^{f0}(N-1,\cdot)$, $y_2 = f^{f0}(N-2,\cdot)$ and $y_3 = f^{f0}(N-3,\cdot)$. In the unilateral Z transform domain, this is equivalent to:

$$\begin{aligned} Y(z) + a_1\left(z^{-1}Y(z)+y_1\right) + a_2\left(z^{-2}Y(z)+z^{-1}y_1+y_2\right) + \\ a_3\left(z^{-3}Y(z)+z^{-2}y_1+z^{-1}y_2+y_3\right) = b_0 X(z) \end{aligned} \tag{E.41}$$

Collecting terms, the expression can be rewritten as:

$$Y(z) = \frac{b_0}{Q(z)}X(z) - y_1\frac{a_1+a_2z^{-1}+a_3z^{-2}}{Q(z)} - y_2\frac{a_2+a_3Z^{-1}}{Q(z)} - y_3\frac{a_3}{Q(z)} \tag{E.42}$$

Now we will consider the simulated backward pass in the right boundary. Let us denote by $v(t)$ the signal obtained at the right boundary by backward filtering signal $y(t)$ from $t = \infty$ to $t = 0$. Because the filter starts at ∞ we can consider zero initial condition at this stage (any transient vanishes before reaching the boundary). The Z transform $V(z)$ can be obtained by multiplying $Y(z)$ by $\widetilde{w}^b(z) = b_0/Q(z^{-1})$, leading to:

$$V(z) = \frac{b_0^2}{W(z)}X(z) - y_1 b_0 \frac{a_1 + a_2 z^{-1} + a_3 z^{-2}}{W(z)} - y_2 b_0 \frac{a_2 + a_3 Z^{-1}}{W(z)} - b_0 y_3 \frac{a_3}{W(z)} \tag{E.43}$$

where $W(z) = Q(z)Q(z^{-1})$. This can be decomposed in a natural term $V_n(z)$ depending only on the initial conditions and a forced term $V_f(z)$ depending only on the input signal:

$$V_f(z) = \frac{b_0^2}{W(z)}X(z) \tag{E.44}$$

$$V_n(z) = -y_1 b_0 \frac{a_1 + a_2 z^{-1} + a_3 z^{-2}}{W(z)} - y_2 b_0 \frac{a_2 + a_3 Z^{-1}}{W(z)} - b_0 y_3 \frac{a_3}{W(z)} \tag{E.45}$$

To compute $v_n(t)$ and $v_f(t)$, their Z transform must be inverted. This can be done by performing a partial fraction expansions in first order terms. Let p_1, p_2 and p_3 be the poles of $Q(z)$. In terms of these poles, function $W(z) = Q(z)Q(z^{-1})$ is given by:

$$W(z) = \left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right)\left(1 - p_3 z^{-1}\right)(1 - p_1 z)(1 - p_2 z)(1 - p_3 z) \tag{E.46}$$

and its inverse can be written as:

$$\frac{1}{W(z)} = \frac{b_0^2 a_3^{-1} z^{-3}}{\prod_{i=1}^{3}(1 - p_i z^{-1}) \prod_{i=1}^{3}\left(1 - p_i^{-1} z^{-1}\right)} \tag{E.47}$$

Performing a partial fraction expansion of the previous function, we obtain:

$$\frac{1}{W(z)} = \sum_{i=1}^{3} \frac{R_i}{1 - p_i z^{-1}} + \sum_{i=1}^{3} \frac{R_i'}{1 - p_i^{-1} z^{-1}} \tag{E.48}$$

where the residues of causal terms R_i and anti-causal terms R_i' can be computed by:

$$\begin{cases} R_i = \frac{1}{W(z)}\left(1 - p_i z^{-1}\right), \; z = p_i \\ R_i' = \frac{1}{W(z)}\left(1 - p_i^{-1} z^{-1}\right), \; z = p_i^{-1} \end{cases} \tag{E.49}$$

Now, the natural term $V_n(z)$ can be written as:

$$V_n(z) = -b_0 \sum_{i=1}^{3} \sum_{t=1}^{3} y_t \left(\frac{r(i,t)}{1 - p_i z^{-1}} + \frac{r'(i,t)}{1 - p_i^{-1} z^{-1}} \right) \tag{E.50}$$

where the residues $r(i,n)$ and $r'(i,n)$ are given by:

$$\begin{cases} r(i,1) = R_1\left(a_1 + a_2 p_i^{-1} + a_3 p_i^{-2}\right) \\ r(i,2) = R_2\left(a_2 + a_3 p_i^{-1}\right) \\ r(i,3) = R_3 a_3 \\ r'(i,1) = R'_1\left(a_1 + a_2 p_i + a_3 p_i^2\right) \\ r'(i,2) = R'_2\left(a_2 + a_3 p_i\right) \\ r'(i,3) = R'_3 a_3 \end{cases} \tag{E.51}$$

Only values of $v_n(t)$ for $t \geq 0$ are needed to compute the initial conditions. Thus, to obtain time function $v_n(t)$, only the causal residues are used[3]:

$$v_n(t) = b_0 \sum_{i=1}^{3} p_i^t y_t r(i,t) \tag{E.52}$$

Considering now the forced response, the partial fraction expansion of $V_f(z)$ involves the pole of the input signal:

$$X(z) = \frac{x_0 \exp\left[i(\omega_y y - \omega_x N)\right]}{1 - p_4 z^{-1}} \tag{E.53}$$

where $p_4 = \exp\left[i\omega_x\right]$. Thus, we have:

$$V_f(z) = x_0 b_0^2 \frac{r_4}{1 - p_4 z^{-1}} + x_0 b_0^2 \sum_{i=1}^{3} \frac{r_i}{1 - p_i z^{-1}} + \frac{r'_i}{1 - p_i^{-1} z^{-1}} \tag{E.54}$$

where the residues are given by:

$$\begin{cases} r_i = R_i \exp\left[i(\omega_y y - \omega_x N)\right] / \left(1 - p_4 p_i^{-1}\right) \\ r_4 = \exp\left[i(\omega_y y - \omega_x N)\right] / W(z),\ z = p_4 \\ r'_i = R'_i \exp\left[i(\omega_y y - \omega_x N)\right] / \left(1 - p_4 p_i\right) \end{cases} \tag{E.55}$$

Consequently, the forced response $v_f(t)$ for $t \geq 0$ is given by:

$$v_f(t) = x_0 b_0^2 r_4 e^{i\omega_x t} + x_0 b_0^2 \sum_{i=1}^{3} r_i p_i^t \tag{E.56}$$

Finally, the initial conditions for the horizontal backward pass can be computed by:

$$v(t) = v_n(t) + v_f(t),\ t = 1, 2, 3 \tag{E.57}$$

Vertical Forward Pass

With the constant boundary assumption, the top boundary is a complex exponential signal described by:

$$f^{00}(x,y) = f(x,0) \cdot e^{i \cdot (\omega_x x + \omega_y y)},\ y < 0 \tag{E.58}$$

[3]the response associated to non-causal terms only exist for $t < 0$.

After the horizontal forward and backward filtering passes, the top boundary is transformed by the frequency response of the cascaded forward and backward filters. Before vertical filtering, the top boundary is given by:

$$f^{b0}(x,y) = |\widetilde{w}'(e^{i\omega_x})| f(x,0) \cdot e^{i(\omega_x x + \omega_y y))},\ y < 0 \tag{E.59}$$

In the last expression it was taken into consideration that the $w'(\cdot)$ is a zero phase filter. The initial conditions for the vertical forward pass can be computed by multiplying the above signals by the frequency response of the vertical forward filter at the adequate frequencies:

$$f^{bf}(x,t) = f(x,0)|\widetilde{w}^f(e^{i\omega_y})| \cdot |\widetilde{w}'(e^{i\omega_x})| \cdot e^{i(\omega_x x + \omega_y t + \angle\widetilde{w}^f(e^{i\omega_y})))},\ t < 0 \tag{E.60}$$

Vertical Backward Pass

In a analogous way, the bottom boundary is described by:

$$f^{b0}(x,y) = |\widetilde{w}'(e^{i\omega_x})| f(x,N-1) \cdot e^{i(\omega_x x + \omega_y y))},\ y > N-1 \tag{E.61}$$

Shifting the origin of the vertical coordinates to $y = N$, we redefine the input signal:

$$x(t) = x_0|\widetilde{w}'(e^{i\omega_x})| \cdot e^{i(\omega_x x + \omega_y t - \omega_y N))},\ t \geq 0 \tag{E.62}$$

where $x_0 = f(x, N-1)$. Again, we decompose the result of the vertical filtering in the boundary as a signal $v(t)$ that correspond to a natural response to initial condition of the forward pass, and a forced response to the signal in the boundary:

$$v(t) = v_n(t) + v_f(t) \tag{E.63}$$

In a similar fashion to what shown in Section E.5, the natural part of the response is given by (E.50) with the residues in (E.51), where the initial conditions are now given by:

$$\begin{cases} y_1 = f^{bf}(x, N-1) \\ y_2 = f^{bf}(x, N-2) \\ y_3 = f^{bf}(x, N-3) \end{cases} \tag{E.64}$$

With regard to the forced response, the derivation is also very similar to what was presented in Section E.5. It can be computed by (E.56), with the new residues given by:

$$\begin{cases} r_i = R_i \exp\left[i(\omega_x x - \omega_y N)\right] / \left(1 - p_4 p_i^{-1}\right) \\ r_4 = |\widetilde{w}'(e^{i\omega_x})| \exp\left[i(\omega_x x - \omega_y N)\right] / W(z),\ z = p_4 \\ r_i' = R_i' \exp\left[i(\omega_x x - \omega_y N)\right] / \left(1 - p_4 p_i\right) \end{cases} \tag{E.65}$$