Optical Flow Estimation in the Complex Logarithmic Plane

Diplomarbeit
am Lehrstuhl für Kognitive Systeme
des Instituts für Informatik und Praktische Mathematik
der Christian–Albrechts–Universität zu Kiel

vorgelegt von
Volker Krüger

Betreuer: Dr. Kostas Daniilidis
Prof. Dr. Ing. Gerald Sommer

Dezember 1995

Abstract

Vision systems with spatially homogeneous resolution are not able to provide a real time response in a dynamically changing environment. A reactive behavior necessitates selective sensing in space. Such a selection can be accomplished by the combination of a space-variant resolution scheme and a sensor with controllable degrees of freedom. The field of view is split into a homogeneous high resolution area - the fovea - and the periphery with decreasing resolution. Both in neurobiology and in robot vision, models of the resolution decrease towards the image boundaries have been established. The most convincing model is the theory of logarithmic polar mapping. However, in order to exploit the rich properties of this mapping we have to tackle with a geometric deformation that is due to the polar mapping as well as with decimation effects that are due to the logarithmic scaling. In this thesis, we will study analytically and experimentally the polar deformation effects and the decimation effects due to the logarithmic subsampling on the computation of optical flow. We will first give an extensive introduction into the complex logarithmic mapping. We will then study in detail the deformation effects that are due to the complex mapping. This study will reveal a principal difference between optical flow that occurs in Cartesian and in complex logarithmically mapped images. We therefore propose two new methods for the estimation of the optical flow. We will continue our studies with an analysis of the logarithmic subsampling scheme and the resulting decimation effects on the image data. We will introduce a new analysis technique that will allow us to apply linear shift invariant analysis techniques to logarithmically subsampled data. This technique will reveal that logarithmically subsampled data contains so high frequencies that most derivative filters fail. We will therefore propose a frequency domain based derivative and smoothing filter that performs significantly better for the optical flow estimation on complex logarithmically mapped image-sequences than the usually applied binomial kernels.

Zusammenfassung

Visuelle Systeme mit örtlich homogener Auflösung besitzen aufgrund der zu geringen Rechnerkapazitäten heutiger Computersysteme nicht die Fähigkeit, in Echtzeit auf Veränderungen in ihrer Umgebung zu reagieren. Diese Fähigkeit, um sie trotzdem zu realisieren, verlangt ein zielbewußtes und ausgewähltes Beobachten der Umgebung. Solch ein zielbewußtes Beobachten läßt sich durch einen ortsvarianten Sensor mit steuerbaren Freiheitsgraden besonders unterstützen. In der Neurobiologie sowie in der Robotik haben sich jüngst Sensoren etabliert, die eine in Richtung der Peripherie des Sensors abnehmende Auflösung aufweisen. Das bekannteste Modell zeigt eine komplex-logarithmische Auflösung. Um den Informationsgehalt dieser visuellen Daten vollständig ausschöpfen zu können, müssen Probleme gelöst werden, die sich aus der geometrischen Deformation durch die polare Transformation und aus der logarithmischen Unterabtastung ergeben. In dieser Arbeit erörtern wir analytisch und experimentell die durch die logarithmisch polare Transformation eintretenden Probleme für die Berechnung des optischen Flusses. Zu aller erst werden wir eine umfangreiche Einleitung in die komplex-logarithmischen Transformation geben. Wir werden dann in allen Einzelheiten die polare Deformation sowie die daraus entstehenden Probleme für die Flußschätzung erörtern. Es wird deutlich werden, daß übliche kartesische Verfahren zur Flußschätzung nicht mehr angewendet werden können und daß stattdessen die Eigenschaften der polaren Transformation besonders berücksichtigt werden müssen. Wir werden deshalb zwei neue Verfahren zur optischen Flußschätzung vorstellen, die diese speziellen Eigenschaften aufgreifen und somit wesentlich bessere Ergebnisse liefern als herkömmliche Flußverfahren. Danach werden wir die logarithmische Unterabtastung und die daraus resultierenden Effekte für die Bilddaten analysieren. Hierfür werden wir eine neue Technik einführen, die uns erlauben wird, logarithmisch unterabgetastete Bilder auf einem kartesischen Träger ohne Informationsverlust darzustellen. Unter Anwendung dieser Analysetechnik werden wir sehen, daß logarithmisch unterabgetastete Bilder wesentlich höhere Frequenzen aufweisen als ihre Originale. Allgemeine Ableitungsfilter versagen bei hochfrequenten Bildern. Es werden deshalb neue Glättungs- und Ableitungsfilter eingeführt, die gute Filterantworten auch bei hohen Frequenzen zeigen und speziell zur Berechnung des optischen Flusses entwickelt werden.

Contents

Chapter 1

Introduction

This work is concerned with the analysis of visual data obtained from a newly invented type of camera sensor which is highly motivated by the primate's eye. In contrast to a conventional CCD camera the primate's eye shows a very high homogeneous resolution in the *fovea*. Towards the periphery and with increasing distance to the fovea the resolution decreases linearly. This so-called space-variant resolution scheme of the human's eye allows therefore a detailed information on the fixated objects and a wide field of view at the same time but still limits the amount of visual information to be processed.

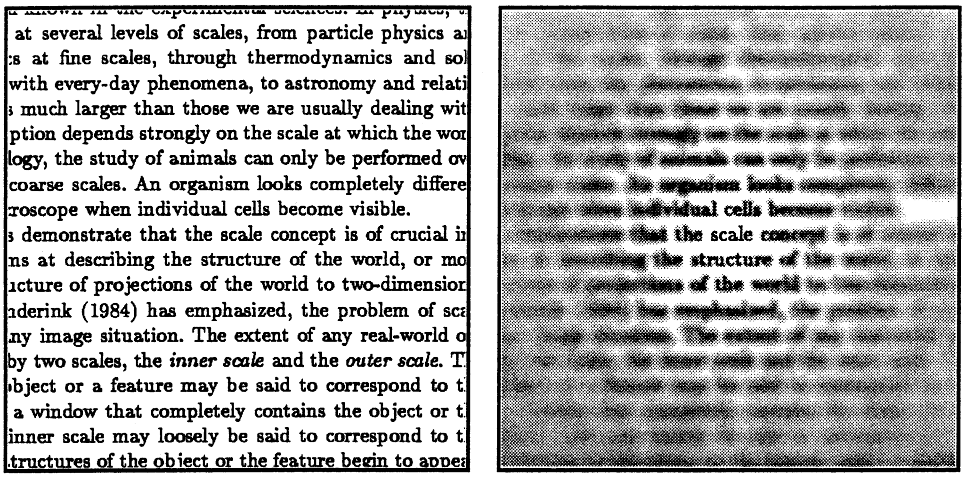


Figure 1.1: *A piece of text shown at a constant resolution (left) like a camera and the same text with linearly decreasing resolution (right) [Lindeberg and Florack, 1994].*

1.1 Biological Background

For a better understanding of this principle we give a basic and simplified introduction to the primate's eye. The primate's retina consists of three layers. In the first layer *photo-receptor-cells* (PRC) code the visual information in terms of nervous impulses. The PRC are connected either directly via bipolar-cells or indirectly via amacrin-, bipolar-, and horizontal-cells in the second layer to several types of *retinal ganglion cells* (RGC) in the third layer. The function of ganglion cells can be described by mainly two mechanisms. The roughly circular shaped center that produces a response either to the introduction (ON or *excitatory* response) or a with-drawl (OFF or *inhibitory* response) of the stimulus. The annular shaped surround shows a response in opposition to its center. Ganglion cells are thus divided into two classes of equal proportion: the ON-center cells with an ON center and an OFF surround and the OFF-center cells with an OFF center and an ON surround. Three further types of ganglion cells can be distinguished: the X, the Y and the W type.

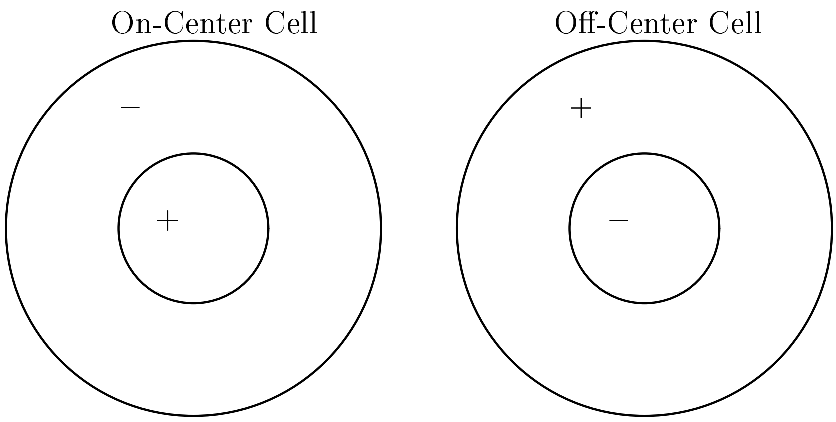


Figure 1.2: *The two major types of receptive fields of retinal ganglion cells are the* on-center cells *with an inhibitory surround and the* off-center cells *with an excitatory surround. The plus sign denotes the regions with on-reaction, the minus sign denotes the regions with off-reaction.*

The X-type cells show an approximately linear spatial summation over their inputs while for the Y- and W-type cells the summation is nonlinear. The W type is characterized by a slow conduction velocity of their inputs. Receptive fields of X-type ganglion cells, described in polar coordinates, realize difference-of-Gaussian (DoG) weighting on their inputs [Hawken and Parker, 1994; Kronauer and Zeevi, 1985]:

$$\begin{aligned} g_X(\rho, \sigma_i, \sigma_e) &= \frac{1}{\sqrt{2\pi}\sigma_e^2}\exp(-\rho^2/2\sigma_e^2) \\ &\quad -\frac{1}{\sqrt{2\pi}\sigma_i^2}\exp(-\rho^2/2\sigma_i^2) \\ &= G_e(\rho) - G_i(\rho), \ \ \sigma_i > \sigma_e \end{aligned} \tag{1.1}$$

where σ_e and σ_i denote the effective width of the excitatory center and the inhibitory surround of the Mexican-hat shaped receptive field. Such a simple model does not hold for Y and W type cells. Compared to receptive field dimensions the cell bodies of the PRC are small and packed so densely that it is possible to consider an approximately continuous cellular array. Denoting by (x, y) the X-cell's position, g_X the receptive field function and by e the stimulus then the response of the cell's array $r(d, \theta)$ can be written as

$$r(d,\theta) \;=\; \int_0^\infty \int_0^{2\pi} g_X(\rho)e(d\cos\theta - \rho\cos\phi, d\sin\theta - \rho\sin\phi)d\rho\, d\phi$$

with $d = \sqrt{x^2 + y^2}$ and $\theta = \arctan y/x$. Speaking in terms of spatial filtering (1.1) exhibits a band-pass characteristic. RGC react on local light intensity differences and are sensitive enough to react on intensity variations of only 2%. According to their physiology RGC transfer principally edge informations.

PRC are not uniformly spaced over the primate's retina: Roughly three regions can be distinguished [van Doorn *et al.*, 1972; Kronauer and Zeevi, 1985]: The first region is the *fovea centralis*, up to 0.5° of eccentricity, that contains the maximum receptor density and that shows therefore the best optical image. Above 0.5° the density decreases linearly with increasing eccentricity according to the globally constant *cortical magnification factor* [Wässle *et al.*, 1990]. The second region is an intermediate region and up to

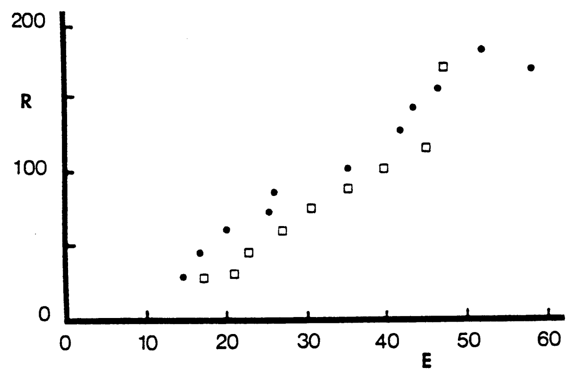


Figure 1.3: *The magnitude of the radius of the receptive fields in the cat's retina as a function of their retinal location. (Radius R in minutes of arc, eccentricity E in degrees) [van Doorn* et al.*, 1972].*

8° of eccentricity. Careful cell counts show that in primates the number of ganglion cells in the central retina (eccentricity $< 8°$) matches or slightly exceeds the number of photo receptors. Thus, while there may be a transform in which central field information is conveyed, there is probably no significant reduction in it. The third area is the peripheral region in which the visual information is compressed before transmission. Except for the fovea the ganglion cells are generally arranged in such a way that the individual cells are spatially coincident in the retinal coordinates with the photo receptors which drive them. Thus, receptive field size increases almost linearly with eccentricity, also according to the magnification factor [Wässle *et al.*, 1990]. This keeps the amount of visual information as received by $\approx 10^8$ cones and rods on the human retina low enough to be processed by only $\approx 10^6$ RGC and optic nerve fibers.

The next processing level is the *lateral geniculate nucleus* (LGN) where the responses of the RGC are projected onto. Hubel and Wiesel [Hubel and Wiesel, 1962] have shown that the LGN cells exhibit characteristics similar to those of the RGC but show a considerable reduction in the spontaneous activity of X-cells which corresponds, in terms of spatial filtering, to a more narrow pass-band. Therefore it is likely that a greater variety of preferred spatial frequencies are found at any single retinal location. The overall role of the LGN is barely known.

The LGN axons impinge on the *striate cortex* (area 17 of the visual cortex). The cortex is laid out in a kind of overall retinotopic mapping, but it differs from the retina on both, large and small scales. The large scale differences come from the fact that on the retina the ganglion cells are packed more densely at the fovea than in the periphery, while in the cortex the cells have uniform packing density. Thus, to allocate a fixed amount of cortical processing per spatial sample, the representation of the fovea occupies far more of the cortex that it does of the retina. The already mentioned *cortical magnification factor* quantifies this distorted projection [Schwartz, 1977] and indicates the amount of cortex associated with each degree of visual field.

The processing of the foveal data is restricted to either static or stabilized (i.e. tracked) objects because it can not provide detailed information on moving objects[1]. Therefore, in order to maintain the capability of perception the eye is equipped with three different types of *gaze control* for stabilization. One type of gaze control is a *reflex-like gaze control* that can, for example, be observed while looking out of a train window: the so called *optokinetic nystagmus* which is a composition of smooth eye movements (during which the eyes track a point of the environment) interrupted by fast *saccadic* movements which reset the direction of gaze. In this case it is practically impossible to keep the eyes steady, even if the subject in the scene is totally uninteresting. The resulting motion strategy maximizes the period during which "useful information" could be analyzed (i.e. information that is not degraded by motion blurring). This behavior is *preattentive* or, in other words, does not require "high-level processing". Another example here is the stabilization of gaze performed during motion of the head which is called *vestibulo-ocular* reflex. Also in this case it is impossible to stabilize the orientation of the visual axis by looking at an "abstract" point in the scene. Another type of gaze control is a *task driven gaze control*. A common example here is the strategy during walking: A task specific smooth eye movement is interrupted by saccadic eye movements that are directed towards the ground in order to support the "walking task". Whereas reflex-like gaze control is completely low-level

[1] e.g. it is impossible to read a moving news-paper

or preattentive, the smooth eye movement is in this case an interest dependent process. However, the saccadic eye movements are preattentive. The third type is a *voluntary gaze control*. This gaze control is a high-level and interest guided process. It allows humans, for example, to scan a row of text by gazing fast from letter to letter, to identify a letter and in general, if we are not dyslexic, to read.

All three types of gaze control have in common that they enable the eye to stabilize the important part of the visual data in the high resolving fovea. It is very important to understand that the space-variant resolution scheme depends on the possibility to control the gaze. The space-variant resolution scheme without gaze control can not provide any visual information.

1.2 First Findings on the Spatial Mapping in the Primates Sensory Projection

Schwartz [Schwartz, 1977] published in *Biological Cybernetics* the first results on the evidence that the density of receptor cells on the primate's retina and their *retinotopic mapping* onto the first area of the striate cortex (*area 17*) follow a simple mathematical rule: The density decreases linearly with the distance to the fovea. If we imagine a coordinate system with the origin in the center of the fovea the coordinates of the receptor cells could conveniently be written in polar coordinates. These two factors together give rise to the

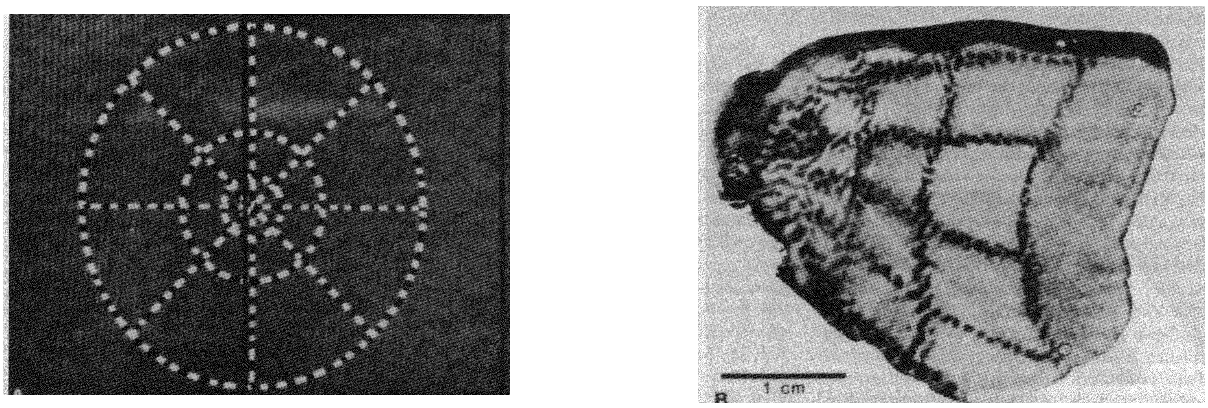


Figure 1.4: *Neural activity patterns: left, the retinal image; right, the area 17 [Spillmann and Werner, 1990].*

evidence of the well known conformal mapping

$$w = \ln(z) \tag{1.2}$$

where w represents a point in the striate cortex, z represents a point on the retina which may be written in polar coordinates. Schwartz denoted this mapping by the mathematical term *complex logarithmic mapping*. The grid formed by receptor cells on the retina is called *complex logarithmic grid*. In fact, experiments on the macaque's cortex (fig.: 1.4) [Hubel, 1989] have shown that the neural activity patterns that are due to a retinotopic mapping can be described in terms of the complex-logarithmic mapping. However, for other cortical and sub-cortical areas, the experimental data is much less comprehensive. The magnification factor for the (LGN) is the same as the one for the striate cortex, up to a scale factor and the complex logarithmic structure of the striate cortex seems to be effected in two ways: 1) the form of the density of the RGC, leading to a logarithmic radial structure in the optic tract and LGN, 2) the projection of the LGN onto the optic tract fibers that leads to the final form of the striate cortex retinotopic map, as in Equation (1.2).

1.3 The Complex Logarithmic Mapping in Computer Vision

Two years after the publication of Schwartz the complex logarithmic mapping was for the first time mentioned in terms of computer vision by [Weiman and Chaikin, 1979]. Weiman used a camera system whose sensor showed a complex logarithmic grid instead of the common Cartesian grid. Using this camera he derived a complex logarithmic mapped image which corresponds, in biological terms, to a retinotopic image mapped on the striate cortex. He analyzed the complex logarithmic grid and found that it has several advantages over the conventional Cartesian grid. As a camera moves in space the image of the projected scene undergoes changes that are related to the camera movement. In case of pan and tilt, for a small angle, and a reasonable distance from the object the imaged object moves without rotation, scaling or any distortion to a new position in the image plane. Such a motion is called an *image translation*. If the camera rotates about the visual axis the imaged object undergoes a *rotation* on the image plane. If the camera moves along the visual axis without rotation then the imaged object will be *scaled* in the image plane. As stated before, a space-variant resolution scheme does not make sense without tracking. Under the assumption that an object is foveated and tracked a translation does not occur. Weiman found further, that, again under the assumption of a foveated and tracked object, the complex logarithmic mapping reduces a scaling and a rotation to a simple shift in the log-polar mapped image. These properties are very important characteristics of the complex logarithmic mapping and simplify significantly many navigation techniques. We will study the properties in chapter 2.

1.4 Active Vision and Space–Variant Sensing

In 1990 the sensor was mentioned for the first time in terms of *active vision* by [Kreider *et al.*, 1990]. An *active vision system* is defined as a system which is able to manipulate its visual parameters in a controlled manner in order to extract useful data about the scene in time and space [Pahlavan *et al.*, 1993]. This definition says that an active vision system has the capability to change its visual parameters, e.g. to change the gaze or the focal length or to move in space (motion parallax). This capability is used to acquire the data from the scene which is of interest with respect to the specific task it has at the time. The active role of a vision system in its environment implies therefore the ability of an *interest guided perception* of the system and a dynamic interaction with the environment. The *behavior based* approach of active robot systems with visual capabilities [Sommer, 1995] poses an alternative to the traditional principle of a *passive* perception task that is totally independent from the specific interest. Further, such a *passive* system has to extract all information needed from a given image, possibly involving complicated reasoning and computations but cannot acquire more data which could facilitate the computations.

Only the relationship of the visual task with a behavior can facilitate a robust and a real-time interaction of a robot in a complex context. This relationship supposes a selective gathering of the sensorial data. Hence, one of the key aspects of active vision is the "where-to-look-next-aspect" for an interest guided exploration of the environment or during motor actions. However a major problem is, that gaze and attention control assume a resolution in the direction of gaze that is high enough for a quantitative analysis as well as a wide field of view. Using conventional Cartesian CCD cameras results in unacceptable burden for modern computers. Instead a space-variant resolution scheme (fovea – periphery) makes sense: Its field of view is split into a central region with a high resolution and the peripheral part of the visual field with a decreasing resolution which, on the whole, reduces the amount of data significantly. In fact, depending on the sensor specifications a data reduction down to only 4% of the original data amount is achieved which not only reduces the computational effort but solves also the well known bottle neck problem of a limited bus bandwidth between the camera and the framegrabber, too. Sensing the environment while using a space-variant sensor is referred to by the term *space-variant sensing*.

It has been proposed by many neuroethologists that most basic visual capabilities found in animals are based on motion [Horridge, 1986; Poggio and Reichardt, 1973]. Surely we do not have to copy biology, nevertheless the existence of vision in biological organisms gives rise to believe that artificial systems could possibly work in a similar way. Active robot systems that are equipped with visual sensors must possess basic visual motion capabilities for navigation tasks. Motion that is sensed through camera sensors results into apparent motion of the brightness patterns in the images, which is called *optical flow*. The robot has to evaluate this optical flow. Then, this optical flow is fed into navigational tasks in order to control the movements of the robot or to focus its attention on other objects that could injure or hinder it. Figure 1.4 shows simplified the dependency: Motion of the robot itself and motion of other objects result into an apparent motion which is sensed by a space-variant sensor. The sensed motion results in optical flow which in turn has to be estimated in order to feed the navigational task that controls the action (motion) of the robot. Even though space-variant sensing simplifies navigational tasks [Tistarelli and Sandini, 1993; Daniilidis, 1994] understanding the effects of space-variant sensing on the estimation of the optical flow is of major importance since this estimation is used as input to navigational tasks like *obstacle avoidance* which are undoubtfully of vital importance.

The scope of this work is to study the effects of the complex logarithmic mapping on the estimation of optical flow. We will restrict our studies on *differential techniques* since only those promise a flow estimation in real-time. In chapter 3.1 we give an extensive introduction to differential optical flow techniques. We will explain that differential flow

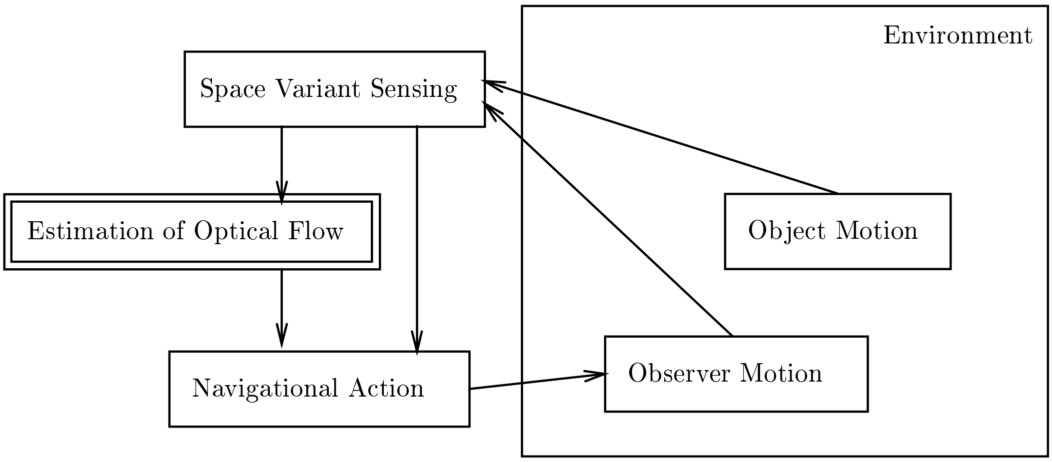


Figure 1.5: *The graph shows simplified the dependency of the navigational tasks on the optical flow: Motion is sensed through cameras. The resulting motion of brightness patterns in the images is used to estimate the optical flow. The optical flow feeds then the navigational tasks.*

techniques depend on the spatial and temporal derivatives of an image sequence and that the quality of the derivatives determines the quality of the estimated flow.

The thesis is split into two major parts. After giving an extensive introduction to the complex logarithmic mapping, we will prove in the first part

1. that the polar transformation introduces fictitious gray-values curvature that leads to an erroneous elimination of the *aperture problem*;

2. we will propose two new methods for the optical flow estimation in the log-polar domain that are superior to methods directly transferred from the Cartesian domain and we experimentally study their performance in a real sequence.

In the second part we will

1. propose a basis for an analysis of the logarithmic subsampling that allows spectral techniques for the design of the necessary low-pass and gradient filters;

2. we will show by exploiting the technique of 3. that logarithmic polar transformed images and sequences contain higher spatial frequencies than their corresponding Cartesian ones and that this is a source of errors for the filter response of filters that usually perform well on low frequency signals and

3. we will propose, as a first application for 4., a frequency domain based filter design for smoothing and for derivative filters that performs significantly better on the logarithmically transformed data than the common binomial approximations of the Gaussian derivatives. We will show the advantages in standard test sequences.

Chapter 2

The complex logarithmic mapping

2.1 Introduction

In this chapter we will present and describe in detail the complex logarithmic grid (log-polar grid) and the complex logarithmic mapping (log-polar mapping or log-polar transform) as well as their properties. The log-polar grid is supposed to replace the Cartesian coordinate grid which is traditionally used for image digitization. The discrete log-polar mapping describes the relationship between the Cartesian and the log-polar grid. We will see that the complex logarithmic grid shows a curvilinear topology that simplifies the mathematical formulations for many useful vision tasks because several important geometric linear transformations can be expressed by simple vector addition in complex logarithmic coordinates. In the next two sections we will give the mathematical formulation and definition for the continuous and the discrete complex logarithmic mapping. Following this definition we will derive in section 2.4 the layout of the log-polar grid. Then we will interpret in section 2.5 this definition geometrically and analytically in order to derive the major properties. A good understanding of the layout of the log-polar grid and its geometric properties is very important when interpreting log-polar mapped images. In the last section of this chapter we will introduce several sensor and software implementations.

2.2 Mathematical Formulation

We will use (x, y) to denote Cartesian coordinates and (ρ, η) to denote polar coordinates in the plane. A point (x, y) in the Cartesian plane can be represented by a single complex number z and its polar coordinates (ρ, η):

$$z = x + jy = \rho e^{j\eta}$$

with

$$\begin{array}{lcl} \rho & = & \sqrt{x^2 + y^2} \\ \eta & & \text{such that with } \left\{ \begin{array}{lcl} x & = & \rho\cos(\eta) \\ y & = & \rho\sin(\eta) \end{array} \right. \end{array}$$

The general complex logarithmic mapping (or log-polar mapping/transform) w of a complex point z is defined with the notations above as

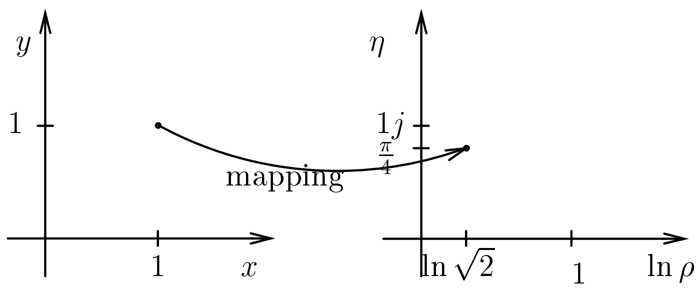
$$w = \ln(z) = \ln(\rho) + j\eta \tag{2.1}$$

for every $z \neq 0$ where $\Re(w) = \ln(\rho)$ and $\Im(w) = \eta + 2k\pi$. To exclude the periodicity of the imaginary part we assume $k = 0$ in order to constrain the range of $\Im(w)$ to $[0, \ldots, 2\pi)$.

With this the log-polar mapping describes the following relationship between Cartesian and log-polar coordinates:

$$(x, y) \overset{\text{mapping}}{\longrightarrow} (\ln(\rho), \eta)$$

The above definition is the most general definition possible. Usually the log-polar mapping



will only be applied to the non-foveal part of the retinal image to avoid a singularity in the origin and to allow a homogeneously sampled region. We thus define $\rho_0 < \rho < \rho_{\max}$, where ρ_0 is the radius of the fovea and $\rho_{\max}$ the half-size of the retinal image. For an arbitrary logarithmic basis a we obtain

$$z = x + jy = \rho a^{j\eta} \quad .$$

The base a is the counterpart to the cortical magnification factor [Wässle *et al.*, 1990].

Let ρ_0 be the radius of the fovea and $\rho_{\max}$ the half-size of the retinal image. For a ρ with $\rho_0 \leq \rho \leq \rho_{\max}$ the complex logarithmic mapping w of a point $z = \rho a^{j\eta}$ is then defined as

$$w = \log_a(z) = \log_a(\rho) + j\eta \tag{2.2}$$

for every $z \neq 0$ where $\Re(w) = \log_a(\rho)$ and $\Im(w) = \eta + 2k\pi$. Again, the range of $\Im(w)$ is constrained to $[0, \ldots, 2\pi)$.

In the following we will use as a definition for the complex logarithmic mapping the latter one. For convenience we will call the $(\log_a(\rho), \eta)$-plane with the term w-plane and we will denote the complex plane by the term z-plane. We excluded the fovea in both definitions because the fovea is sampled homogeneously and therefore not subject of discussion here but will be discussed in section 2.6.

2.3 Discretization

The definition as given in the last section applies for the continuous case and the components of the coordinates in the w-plane fall into the intervals $[\log_a(\rho_0), \log_a(\rho_{\max})]$ and $[0, 2\pi)$, respectively.

For the practical use in computer vision the w-plane has to be discretizised and should be adapted in such a way that it can be worked with as with a usual Cartesian image. This means that the radial component and the angular component are both discrete, start at minimal value 0 and finish at some previously defined maximal values. To systemize this idea let ξ and γ denote the radial and the angular component. The term log-polar plane will refer to the discretizised w-plane and has been chosen in contrast to the Cartesian plane. The variables N_r and N_a define the maximal radial and the maximal angular resolution of the log-polar plane. The mapping from the polar coordinates (ρ, η) to the coordinates (ξ, γ) in the log-polar plane given by [Tistarelli and Sandini, 1992] reads

$$\begin{aligned} \xi &= \log_a\left(\frac{\rho}{\rho_0}\right) = \log_a(\rho) - \log_a(\rho_0) && (2.3)\\ &= \log_a(\rho) - p \;\text{ with }\; p = \log_a(\rho_0) \\ \gamma &= \frac{N_a}{2\pi}\left(\eta + \frac{\pi}{2}\right) = \left(\eta + \frac{\pi}{2}\right)\cdot q \;\text{ with }\; q = \frac{N_a}{2\pi}\ . && (2.4)\end{aligned}$$

With these definitions ξ and γ satisfy $0 \le \xi \le N_r$ and $0 \le \gamma \le N_a$, respectively. Because (ξ, γ) refers to a point in the discrete log-polar plane, both components are always assumed to be natural numbers. Obviously, by looking at equations 2.3 and 2.4, only limited subsets of possible values for ρ and η meet this constraint. This problem is by no means trivial because ignoring the gray values on forbidden positions would mean a loss of relevant visual information. We will revisit this problem in section 2.6. The definition for γ is chosen according to the one employed on the retina-like CCD sensor of [Kreider *et al.*, 1990; Tistarelli and Sandini, 1993].

Because of $\xi = \log_a\left(\frac{\rho}{\rho_0}\right)$ the "magnification factor" a is given in dependency on N_r by

$$\begin{aligned} \rho_0 a^{N_r} &= \rho_{\max} \;\text{ or} && (2.5)\\ a &= \exp\left(\frac{1}{N_r}\ln\left(\frac{\rho_{\max}}{\rho_0}\right)\right)\ . \end{aligned}$$

With this definition of a exactly N_r values for ρ satisfy $\rho_0 \le \rho \le \rho_{\max}$: By numbering these ρ's with index $\xi = 0, \ldots, N_r - 1$, the very first ρ_0 (with index 0) is the same as the predefined ρ_0 which is the radius of the fovea. The last ρ_{N_r-1} equals $\rho_{\max}$. Furthermore exactly N_a values of η satisfy $0 \le \eta < 2\pi$. They will be indexed with η_γ, $\gamma = 0, \ldots, N_a - 1$.

2.4 The complex logarithmic grid

In this section we will study the question *"where is which pixel?"*. While looking at a log-polar mapped image an interpretation is usually very difficult because straight lines are mapped into curves. A good understanding of the geometric rules will help to answer the problem of which gray values of a log-polar mapped image correspond to which gray values in the Cartesian retinal image or, in short, how the visual information is coded.

Let us choose a point (ξ, γ) of the discrete log-polar plane. The polar coordinates ρ and η are obtained by $\rho = a^{\xi}\rho_0$ and $\eta = \frac{2\pi}{N_a}\gamma - \frac{\pi}{2}$ (2.3 - 2.4).

According to (2.2) the following relationship holds:

$$\begin{aligned} x &= \rho_0 a^{\xi} \cos(\eta) \\ y &= \rho_0 a^{\xi} \sin(\eta) \ . \end{aligned} \tag{2.6}$$

With these equations we can now visualize the geometric relationship between the log-polar coordinates and the Cartesian ones. An illustration is shown in Figure 2.1.

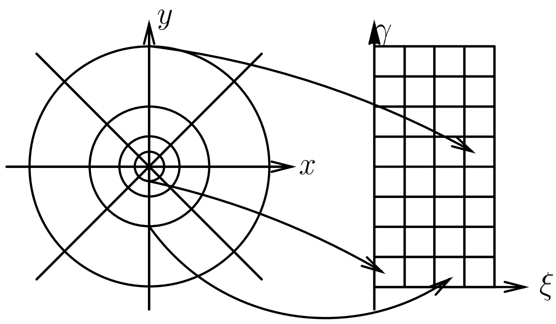


Figure 2.1: *The topology of the log-polar mapping in Cartesian(left) and log-polar coordinates(right) for $N_r = 4$ and $N_a = 8$.*

The complex logarithmic grid is shown on the right, the corresponding mapped version is shown on the left. This illustration is a little bit irritating: The cake-piece like areas are useful here **only** for visualization purposes. The Cartesian points that have corresponding points in the log-polar plane (on the right) are the line-crossings. The correspondences are symbolized with arrows: We start at $-\pi/2$ (or six o' clock position) and assign counter-clockwise to each Cartesian pixel a log-polar pixel. The distance of one log-polar pixel to another along a radius for a fixed angle increases with magnification factor a.

For a practical "exercise" let us have a look at a real image (left) and its log-polar mapped version (right, but with the horizontal axis as γ-axis and the vertical axis, pointing downwards, as ξ-axis) in Figure 2.2. Superimposing the log-polar grid structure of Figure 2.1 onto the Cartesian image and starting at six o'clock position ($-\pi/2$) we can clearly identify the texture of the basement also in the log-polar image. At twelve o'clock position (or position $\pi/2$) we can identify the top of the central column in the low center of the log-polar image. The constant gray value of the top rows in the log-polar image results from the fact that they correspond to Cartesian pixels that lie on the central circles of the log-polar grid. These circles lie in the very center of the Cartesian image and the gray

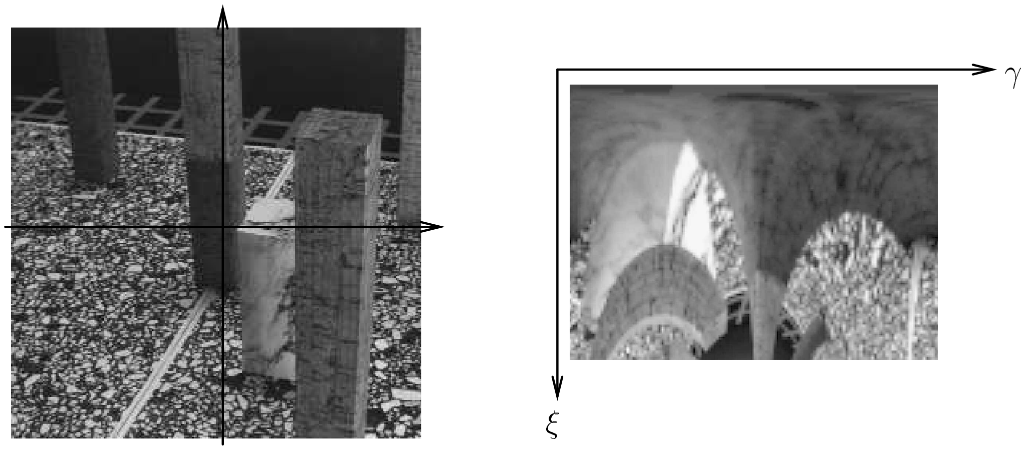


Figure 2.2: *The "Marbled Block" sequence. One original image of the sequence (left) and its log-polar mapped version (right).*

value of the top rows in the log-polar image result therefore from the gray values of the central column.

2.5 Properties

We stated already in the introduction that the complex logarithmic mapping has many global geometric properties and that among the most important ones are the invariance under scaling and the invariance under rotation. Now we will clarify these statements, prove and illustrate them for a better understanding.

2.5.1 Invariance under Scaling

Log-polar mapped images are invariant under scaling. If we let our camera model move along the visual axis towards or away from the fixated object the imaged object is expanded or contracted in the image plane. Let us assume that the image is scaled by some real factor c. If we write the Cartesian coordinates in polar form then the scaling can be described as:

$$\rho e^{j\eta} \quad \stackrel{\text{scaling}}{\longrightarrow} \quad \rho e^{j\eta} \cdot c \ \ .$$

Applying the log-polar mapping leads to

$$\rho e^{j\eta} \cdot c \quad \stackrel{\text{mapping}}{\longrightarrow} \quad \ln\rho + \ln c + j\eta \ \ .$$

Hence, the scaling by the factor c results in a shift along the ρ(real)-axis and the ξ-axis, respectively.

A geometric interpretation is shown in Figure 2.3: On top we see the image of the circular object and its log-polar mapped version. After the camera has moved towards the object it is enlarged as we can see in Figure 2.3, bottom left. The resulting effect in the log-polar plane is a noticeable shift along the ξ-axis.

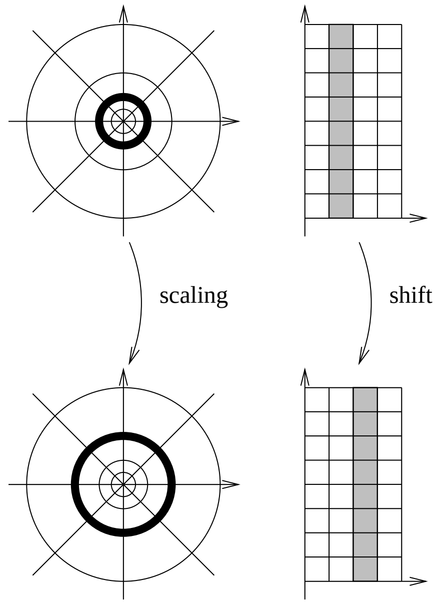


Figure 2.3: *A scaling of the radius results in a shift in the log-polar mapped image: On top left the original image, below the image scaled by some factor $c > 1$.*

2.5.2 Invariance under Rotation

Log-polar mapped images are invariant under rotation. If we let our camera model rotate about the visual axis the imaged object is rotated in the image plane.

Let us assume that the image is rotated by angle α. If we write the Cartesian coordinates again in polar form then the rotation can be described as:

$$\rho e^{j\eta} \quad \overset{\text{rotation}}{\longrightarrow} \quad \rho e^{j(\eta+\alpha)} \quad .$$

Applying the log-polar mapping leads to

$$\rho e^{j(\eta+\alpha)} \quad \overset{\text{mapping}}{\longrightarrow} \quad \ln \rho + j(\eta + \alpha) \quad .$$

Geometrically visualized, we can see that the rotation by angle α results in a shift along the η(imaginary)-axis and γ-axis.

A geometric interpretation is shown in Figure 2.4: On top we see the image of the object and its log-polar mapped version. After the camera has rotated by some angle α (Fig. 2.4, bottom left) the corresponding points in the log-polar plane are shifted along the γ-axis.

2.6 Implementation

Given a square digital Cartesian image $I(x, y)$ containing $2\rho_{\max} \times 2\rho_{\max}$ pixels and superimposing the log-polar grid of Figure 2.1 we can see that the mapping is not one-to-one

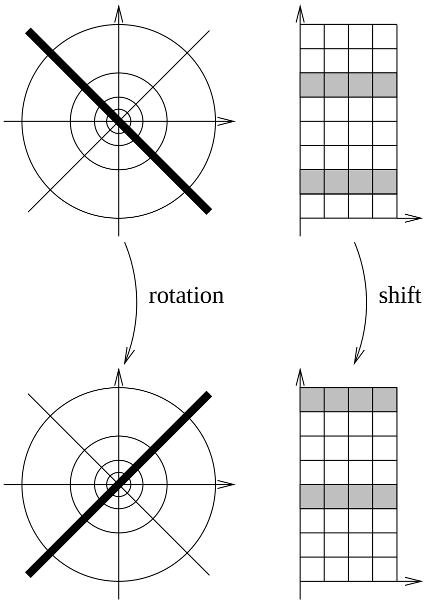


Figure 2.4: *A rotation of an object results in a shift in the log-polar mapped image: On top the original image, below the image rotated by some angle α.*

(see section 2.3). To solve this problem several ideas have been suggested.

2.6.1 Sandini's Sensor

According to [Kreider *et al.*, 1990; Tistarelli and Sandini, 1992] the sensor surface is tiled with cake-piece-like areas (Figure 2.5) in the non foveal part. The fovea is sampled homogeneously. The tile coordinates are given by their middle points which coincide with the marked points in Figure 2.1. Each cake-piece is understood as a single retinal receptor cell. Both the simulation software and the CCD chip version integrate over the area of a single tile the light that reaches it in order to compute the gray value of the corresponding pixel in the log-polar plane. The integration is done without weighting.

Let $\rho_0 \leq \rho_\xi < \rho_{\xi+1} \leq \rho_{\max}$ where ρ_ξ and $\rho_{\xi+1}$ denote the radii of two neighboring pixels ($\xi \in 0, \ldots, N_r - 2$) on the log-polar grid or respectively the ξ^{th} and the $(\xi + 1)^{\text{st}}$ eccentricity (see Fig.2.6.1). Setting the angle $\eta = 0$ equation (2.2) reads

$$\xi = \log_a(\rho)$$

from which follows for unitary dimension of the log-polar pixels

$$\log_a(\rho_{\xi+1}) - \log_a(\rho_\xi) = (\xi + 1) - \xi = 1$$

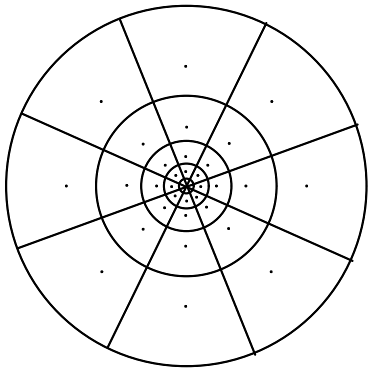


Figure 2.5: *The topology of the log-polar sensor according to [Kreidler* et al.*,1990] and for* $N_r = 4$ *and* $N_a = 8$

which can be rewritten as

$$\begin{aligned} log_a\left(\frac{\rho_{\xi+1}}{\rho_\xi}\right) &= 1 \qquad \text{or} \\ a &= \tfrac{\rho_{\xi+1}}{\rho_\xi} \qquad \text{or} \\ \rho_{\xi+1} &= a\rho_\xi \qquad . \end{aligned}$$

The last equation clarifies the role of the magnification factor a. Further relations follow right away:

$$\begin{aligned} \rho_\xi &= a^\xi \rho_0 \\ \Delta\rho_\xi &= \rho_{\xi+1} - \rho_\xi = a^\xi \Delta\rho_0 \\ \alpha &= \frac{2\pi}{N_a} = q^{-1} \end{aligned}$$

The ξ^{th} tile has thus length

$$\rho_0 a^{\xi+\frac{1}{2}} - \rho_0 a^{\xi-\frac{1}{2}} = \rho_0 a^\xi \left(\sqrt{a} - \frac{1}{\sqrt{a}}\right) = \rho_\xi \left(\sqrt{a} - \frac{1}{\sqrt{a}}\right)$$

The height of a tile (see figure 2.6.1) is given by

$$y_\xi = 2\rho_\xi \tan\left(\frac{1}{2q}\right)$$

where q was defined as $q = \frac{N_a}{2\pi}$.

For a hard- or software implementation attention has to be given to the sensor's specifications because incorrect specifications can result into an oversampling close to the fovea if $\Delta\rho_0$ is smaller that on Cartesian pixel or smaller than the smallest possible CCD cell.

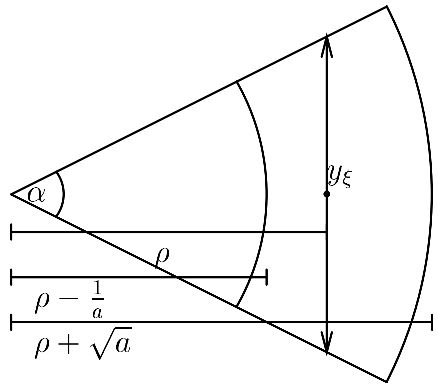


Figure 2.6: *The width of a tile is given by $\rho_{\xi+1} - \rho_\xi$, the height is given by y_ξ*

We say in case of a software implementation, that the parameter of the log-polar mapping are well defined, if neither subsampling nor oversampling occurs.

Figure 2.7 shows a new hardware implementation of a retina like sensor as developed by the IBIDEM project [Ferrari *et al.*, 1995] with $N_a = 64$ and $N_r = 128$.

2.6.2 The Sensor of Bolduc and Levine

The sensor of Bolduc and Levine [Bolduc and Levine, 1994] has the same log-polar grid as the previous implementation. The grid-pixel properties follow thus the same rules as explained in the previous subsection. But in opposite to the previous implementation, Bolduc and Levine were motivated by the retinal ganglion cells and implemented overlapping circular receptive fields with a difference-of-Gaussian weighting. The overlapping circular RF model permits each pixel to contribute to more than one RF and not all pixels contribute to the same number of RF. In contrast the previous model of [Kreider *et al.*, 1990] does not generate overlapping RF, their contents are uniformly averaged and each input contributes to only one RF value. Bolduc and Levine introduce the *overlap factor* o_v, which defines the percentage by which each receptive field is overlapped by its neighboring receptive fields. Further they introduce a *size-to-eccentricity ratio* $c_{d/e}$. The RF size is the diameter of the RF circle, and the eccentricity of an RF is the radial distance from the center of the RF to the center of the sensor. With this they define the angle α and the magnification factor a as

$$\alpha = \cos^{-1}\left(\frac{1 - c_{d/e}}{2}(1 - o_v)^2\right) \tag{2.7}$$

$$a = \frac{-c_{d/e}(1 - 2o_v) - \sqrt{(4 + c_{d/e}^2(1 - 2o_v)^2 - c_{d/e}^2)}}{c_{d/e} - 1} \tag{2.8}$$

This sensor implementation exists only as a software realization and the fovea is a copy of the central circle of the input image.

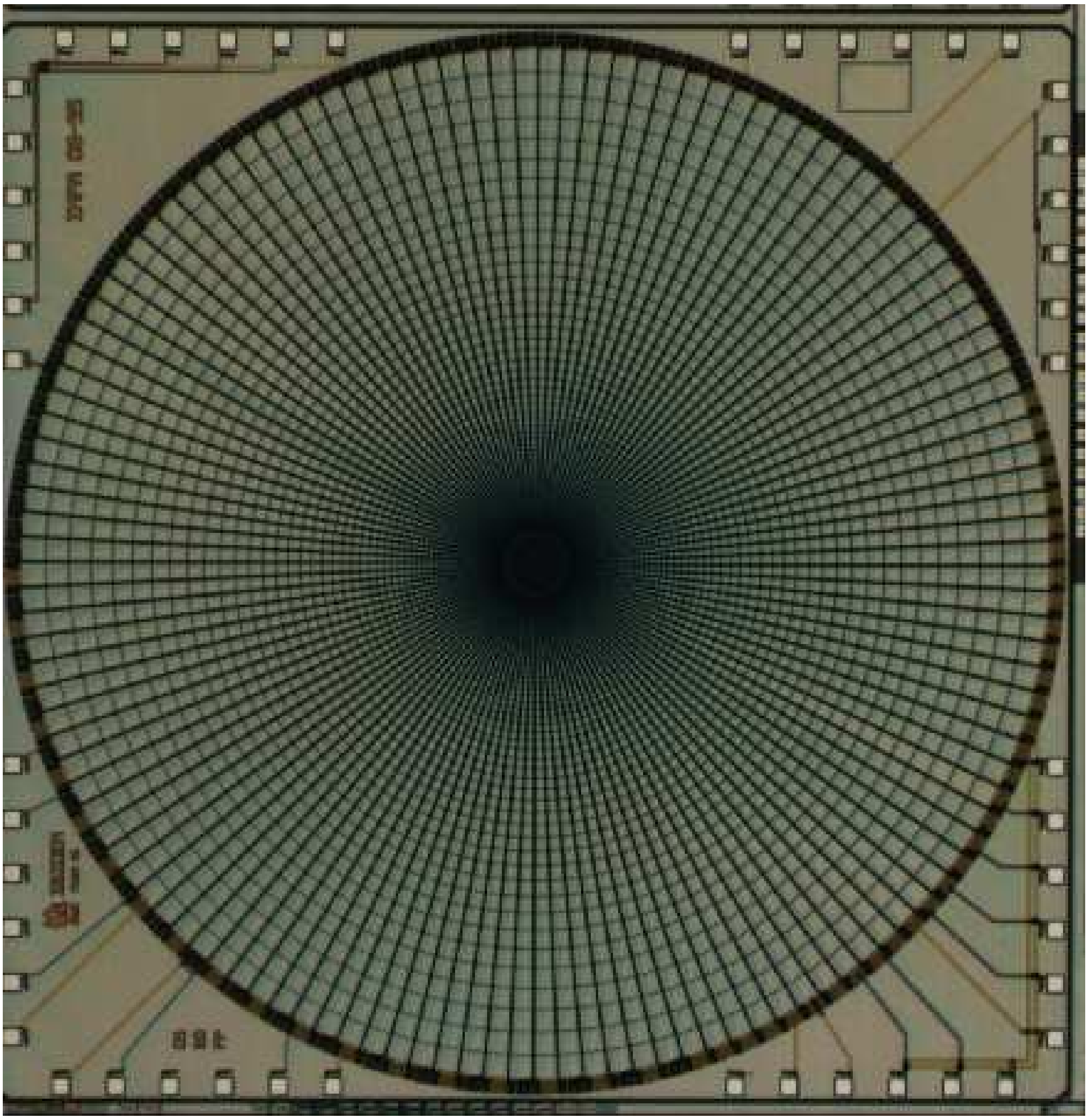


Figure 2.7: *The CCD realization for a retinal like sensor for the IBIDEM project [*Ferrari et al., *1995].*

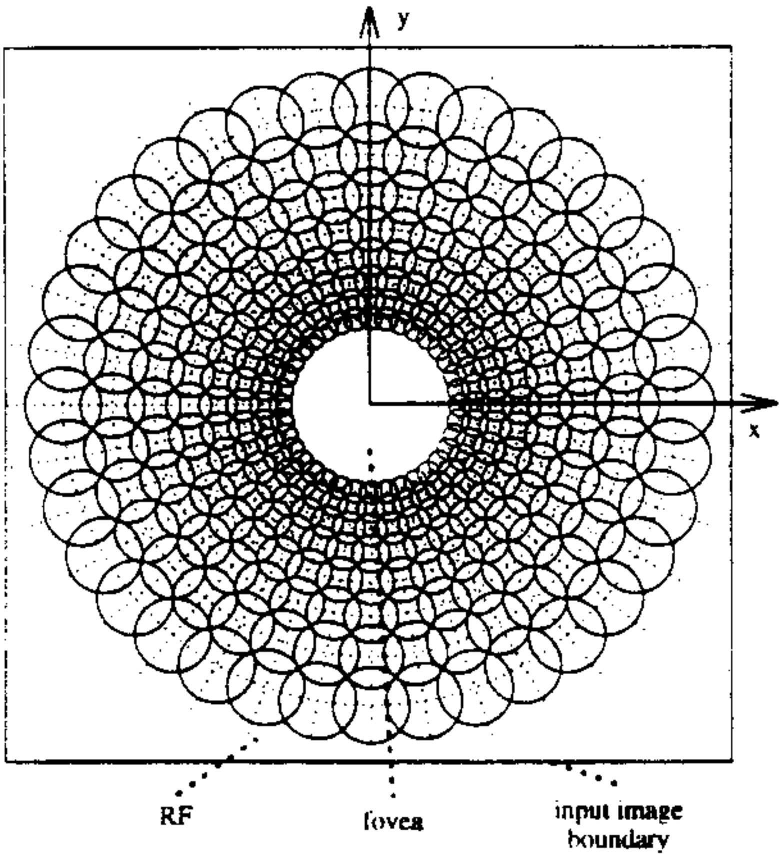


Figure 2.8: *Overlapping circular RF mapping template. The dashed lines represent the log-polar grid on which the RFs are centered [Bolduc and Levine, 1994].*

Chapter 3

Estimating Optical Flow in the Log-Polar Plane

A reliable computation of the optical flow is of vital importance for a robot system. Most navigation tasks depend on the optical flow, and its quality decides on success or failure of a navigation task as, for example, the estimation of the Time-To-Contact (TTC) with another object. Regarding the calculation of optic flow on the log-polar plane it is very important to understand the problems that are introduced by the log-polar mapping and that hinder a good estimation. As explained in section 2.4 the log-polar mapping deforms straight lines into curves. In this chapter we want to address the problems that are introduced by such deformations when computing optical flow. We will use the notion of *optical flow* for the apparent velocity of gray-value structures in the image in opposition to the pure geometric definition of the *motion field* which is defined as the velocity of the projected Cartesian scene points on the image. For the estimation process we will restrict our studies on differential techniques because they promise an optical flow estimation in real time.

After giving first a short introduction to differential flow techniques in section 3.1, we will apply these techniques in section 3.2 to log-polar transformed image sequences. Doing so, we are going to derive flow in terms of log-polar coordinates. However, common navigational techniques use Cartesian parameters, and even though the Cartesian optical flow parameters are uniquely related to the optical flow parameters in the log-polar plane, we will show in section 3.3 that *first* estimating the flow in the log-polar plane and *then* calculating the Cartesian flow components from the estimated log-polar flow components leads to erroneous results. An explanation of this fact is easy: Common differential flow techniques are developed for the Cartesian grid, and therefore every assumption made is expressed in terms of Cartesian coordinates. The assumption of constant or affine flow says that the velocities in x- and y-direction in the image plane, within a local neighborhood, remain constant or, respectively, vary linearly. This assumption is, mathematically spoken, nothing else than the Taylor expansion to the zeroth or first order, which shall describe $\dot{x}$ and $\dot{y}$ within the neighborhood. A differential technique which is applied to a log-polar sampled sequence makes thus assumptions that are not valid any more due to the polar deformation and the logarithmic subsampling scheme: We will see, that the assumption of local constancy or affinity of log-polar flow violates the constancy and affinity of Cartesian flow. In fact, the assumption of local constancy or affinity of log-polar flow implies, that the Cartesian flow varies with respect to the angle and the radius.

A way out of this problem is to relate *first* the log-polar flow components to the Cartesian ones. With this, all assumptions of the flow techniques are not anymore applied in terms of log-polar components but instead in terms of the Cartesian components. Estimating *then* the flow from the log-polar sequences leads to better results.

3.1 Introduction to Optical Flow

Let (u, v) describe the apparent optical flow of a Cartesian image $I(x, y)$ at point (x, y). Further we assume image translation:

$$I(x - ut, y - vt) = I(x, y, t) \ .$$

We assume the *Brightness Change Constrain Equation* (BCCE)

$$I(x - ut, y - vt) = I(x, y, t) \tag{3.1}$$

which means that a geometric point that moves with speed (u, v) does not change its gray value in time, or more formal, that the intensity is conserved over time. Using the Taylor expansion up to the first order with respect to t, we get

$$I_x u + I_y v = -I_t$$

where the subscripts denote the partial derivatives. The following subsections will introduce three common differential techniques by [Lucas and Kanade, 1981; Tretiak and Pastor, 1984; Campani and Verri, 1992]. The first two techniques assume constant flow within a local neighborhood. This is as a 1-D case sketched in figure 3.1: The dotted line shows the estimated flow I_t/I_x which ideally equals u. Alternatively we can relax the assumption of local constancy of the optical flow and allow a so-called *affine* flow which corresponds to the Taylor expansion up to the first order and permits the flow to vary linearly within a local neighborhood. We have to point out that both these assumptions are bounded to an image that is Cartesian.

3.1.1 Lucas and Kanade (1981)

This first method by [Lucas and Kanade, 1981] is an application of the BCCE which uses the gradients of the neighborhood $(x + \delta x, y + \delta y)$ of every considered pixel (x, y). It assumes locally constant optical flow:

$$\frac{d}{dt} I = 0 \ .$$

Let $w(\delta x, \delta y)$ denote a weight function that gives more influence to the constraints at the center of the neighborhood than those at the periphery. In our implementation we used

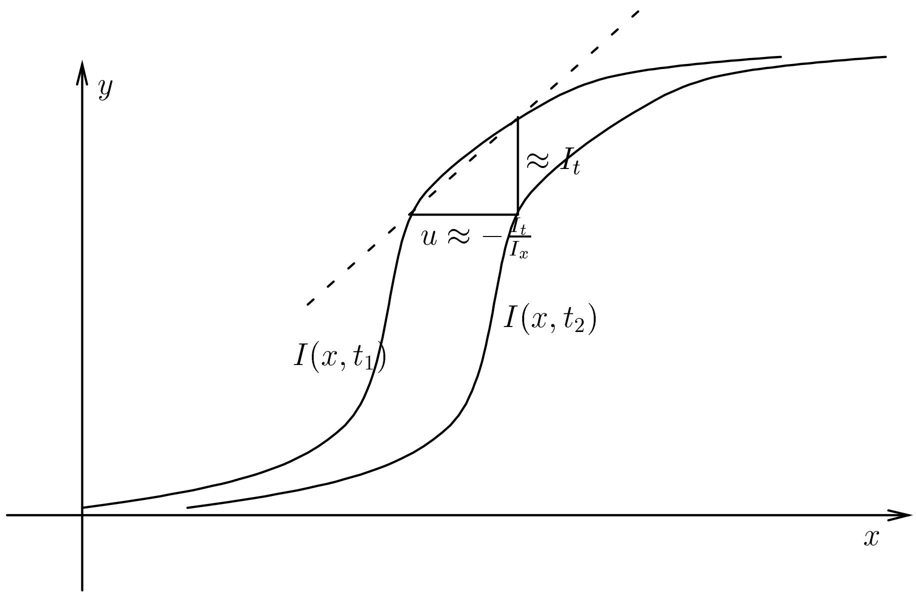


Figure 3.1: *The figure sketches the differential technique in the 1-D; flow is here assumed to be locally constant.*

the Gaussian as an isotropic weight function. The solution is then obtained by minimizing

$$\sum_{\delta x, \delta y} w(\delta x, \delta y)^2 \left(I_x(x+\delta x, y+\delta y)u + I_y(x+\delta x, y+\delta y)v + I_t(x+\delta x, y+\delta y)\right)^2 \qquad (3.2)$$

with respect to (u, v).

If $\mathbf{x}_i$, $i = 1 \ldots n$ denotes n points of a neighborhood at time t and if

$$\begin{aligned} W &= \operatorname{diag}[w(\mathbf{x}_1) \ldots w(\mathbf{x}_n)] \\ A &= [\nabla I(\mathbf{x}_1) \ldots \nabla I(\mathbf{x}_n)] \\ \mathbf{b} &= -[I_t(\mathbf{x}_1) \ldots I_t(\mathbf{x}_n)] \\ \mathbf{v} &= (u, v) \end{aligned}$$

then formula (3.2) may be solved by minimizing $\|W(A\mathbf{u} - \mathbf{b})\|$. The singular value decomposition may be used for a least-squares-estimation of $\mathbf{u} = (u, v)$.

3.1.2 Tretiak and Pastor (1984)

This method is a second derivative method which was first presented by [Tretiak and Pastor, 1984]. It is not, like the previous one, applied on a neighborhood but instead applied on a single pixel while using first and second derivatives of each single pixel. In this way the information of the neighborhood is encoded in the curvature of the gray value surface and thus in the second derivatives. This technique is very fast and was also used for optical flow estimation in the log-polar plane by [Tistarelli and Sandini, 1993]. It imposes the constancy of image brightness and image gradient over time :

$$\frac{d}{dt}I = 0 \qquad \frac{d}{dt}\nabla I = 0 \ .$$

The second spatio-temporal derivatives yield the over-determined system

$$\begin{aligned} I_x u + I_y v + I_t &= 0 \\ I_{xx} u + I_{yx} v + I_{tx} &= 0 \\ I_{xy} u + I_{yy} v + I_{ty} &= 0 \end{aligned} \tag{3.3}$$

With

$$A = \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \\ \frac{\partial^2 I}{\partial^2 x} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial^2 y} \end{bmatrix} \qquad b = \begin{bmatrix} -\frac{\partial I}{\partial t} \\ -\frac{\partial^2 I}{\partial x \partial t} \\ -\frac{\partial^2 I}{\partial y \partial t} \end{bmatrix}$$

we solve $A\mathbf{u} = \mathbf{b}$. Again $\mathbf{u}$ can be found by using the singular value decomposition.

3.1.3 Campani and Verri (1992)

This third method was proposed by [Campani and Verri, 1992] and is an extension of the first one. It allows the linear variation of the optical flow inside a neighborhood

$$\begin{aligned} u &= u_0 + u_x \delta x + u_y \delta y \\ v &= v_0 + v_x \delta x + v_y \delta y \ . \end{aligned}$$

This enables an estimation of the flow as well as its spatial derivatives. The equation

$$I_x(x+\delta x, y+\delta y)(u+u_x\delta x+u_y\delta y)+I_y(x+\delta x, y+\delta y)(v+v_x\delta x+v_y\delta y)+I_t(x+\delta x, y+\delta y) = 0 \tag{3.4}$$

is applied for every $(\delta x, \delta y)$ inside a neighborhood yielding an over-constrained system with the six unknowns

$$u, v, u_x, u_y, v_x, v_y.$$

wherever the Hessian of I is nonsingular. The flow derivatives can be used to compute the differential invariants of the flow like divergence $(u_x + v_y)$, rotation $(v_x - u_y)$, and shear $(u_x - v_y,\ u_y + v_x)$.

For our flow estimation we will use the first method presented and its extension as they showed better results than the method of [Tretiak and Pastor, 1984][1].

3.1.4 The Aperture Problem

For many reasons the computation of optical flow is an ill-posed problem. The only available constrain for the flow (u, v) is the brightness change constrain equation. This means that we can only compute the projection of the flow on the gradient direction

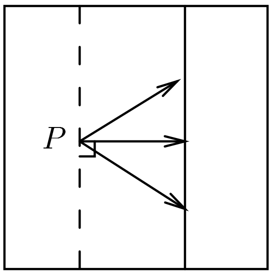


Figure 3.2: *The aperture problem: Only the projection of the flow on the image gradient can be computed.*

$((I_x, I_y) \cdot (u, v) = -I_t$. This problem is called *aperture problem* and is illustrated in figure 3.2: It is not clear where one point P moved to. In a situation like this the linear equation systems of the differential techniques become under-constrained because several points show the same gradient and (u, v) can therefore not be uniquely estimated. If the gradient in a local neighborhood varies then the Hessian matrix of the second derivatives, which was used by [Tretiak and Pastor, 1984], is regular. In points where the aperture problem appears this matrix becomes singular (see discussion in 3.3).

The aperture problem is crucial in case of log-polar mapped images: As shown in section 2.4 straight lines are mapped into curves. Since the aperture problem appears only in case of straight lines the log-polar mapping seems to eliminate this problem. This of course is not true as we will show in the next sections.

3.2 Optical Flow in the Log-Polar Plane

In this section, we apply two common Cartesian flow computation methods to a log-polar image sequence. We denote by (u^l, v^l) the optical flow vector in the log-polar plane and the motion field vectors in the log-polar plane is denoted by $(\dot{\xi}, \dot{\gamma})$, respectively. Let $J(\xi, \gamma)$ be a log-polar image. We first compute the optical flow (u^l, v^l) of the log-polar image $J(\xi, \gamma)$.

The motion field vectors $(\dot{\rho}, \dot{\eta})$ in the polar plane are related to the motion field vectors $(\dot{x}, \dot{y})$ in the Cartesian plane by

$$\begin{aligned} \dot{\rho} &= \dot{x}\cos\eta + \dot{y}\sin\eta \\ \dot{\eta} &= \frac{1}{\rho}\left(-\dot{x}\sin\eta + \dot{y}\cos\eta\right) \ . \end{aligned} \tag{3.5}$$

The relation of the log-polar plane to the polar plane is given by

$$\begin{aligned} \dot{\xi} &= \frac{1}{\ln a}\frac{\dot{\rho}}{\rho} \\ \dot{\gamma} &= \frac{N_a}{2\pi}\dot{\eta} \ . \end{aligned} \tag{3.6}$$

[1]The reason is that the estimation of the second derivatives is very sensitive to noise.

With these equations the Cartesian flow (u, v) can be reconstructed from the log-polar flow (u^l, v^l).

Assuming the brightness change constrain equation for the log-polar image $J(\xi, \gamma)$ we get

$$J_\xi u^l + J_\gamma v^l + J_t = 0$$

where J_ξ, J_γ and J_t are the spatio-temporal derivatives of the image. Applying the BCCE in a neighborhood $(\xi + \delta\xi, \gamma + \delta\gamma)$ and using the differential method of Lucas and Kanade from section 3.1.1 with appropriate weights $w(\delta\xi, \delta\gamma)$, we have to minimize the sum

$$\sum_{\delta\xi,\delta\gamma} w(\delta\xi, \delta\gamma)^2 \left(J_\xi(\xi + \delta\xi, \gamma + \delta\gamma)u^l + J_\gamma(\xi + \delta\xi, \gamma + \delta\gamma)v^l + J_t(\xi + \delta\xi, \gamma + \delta\gamma)\right)^2 , \quad (3.7)$$

with respect to (u^l, v^l). The sum can be rewritten in vector notation: If $\boldsymbol{\xi}_i$, $i = 1 \ldots n$ represent n points of a neighborhood (neighborhoods of 3×3 or 4×4 are realistic) at time t and

$$\begin{aligned} W &= \operatorname{diag}\left[w(\boldsymbol{\xi}_1) \ldots w(\boldsymbol{\xi}_n)\right] \\ A &= \left[\nabla J(\boldsymbol{\xi}_1) \ldots \nabla J(\boldsymbol{\xi}_n)\right] \\ \boldsymbol{b} &= -\left[J_t(\boldsymbol{\xi}_1) \ldots J_t(\boldsymbol{\xi}_n)\right] \\ \boldsymbol{v}^l &= (u^l, v^l) \ , \end{aligned} \quad (3.8)$$

then a solution for (3.7) is obtained by minimizing $\|W(A\boldsymbol{u}^l - \boldsymbol{b})\|$. We obtained a least-squares-estimation by using the singular value decomposition algorithm.

We extend this method by allowing the flow (u^l, v^l) to vary linearly inside a local neighborhood:

$$u^l = u_0^L + \delta u^l + u_\xi^l \delta\xi + u_\gamma^l \delta\gamma \quad (3.9)$$

$$v^l = v_0^L + \delta v^l + v_\xi^l \delta\xi + v_\gamma^l \delta\gamma \ . \quad (3.10)$$

This leads to

$$J_\xi(\xi+\delta\xi, \gamma+\delta\gamma)(u_0^l+u_\xi^l\delta\xi+u_\gamma^l\delta\gamma)+J_\gamma(\xi+\delta\xi, \gamma+\delta\gamma)(v_0^l+v_\xi^l\delta\xi+v_\gamma^l\delta\gamma)+J_t(\xi+\delta\xi, \gamma+\delta\gamma) = 0 \quad (3.11)$$

This extension is the same as the one done by [Campani and Verri, 1992] in section 3.1.3: it allows the estimation of the flow as well as its spatial derivatives. Applying (3.11) for every $(\delta\xi, \delta\gamma)$ in a neighborhood yields an over-constrained equation system with six unknowns which again we solve using singular value decomposition from [Press *et al.*, 1986].

Both presented methods use assumptions about the local constancy and local affinity of the optical flow in the log-polar plane. However, equations (3.5) and (3.6) show, that the flow components can *not* be assumed to be locally constant (or affine) as they are functions of ρ and η and therefore *must* vary according to some function of ρ and η. We explain mathematically in section 3.4, why this is so. Assuming the local constancy or the local affinity does not reflect the harmonic variations of the log-polar flow components

$\dot{\xi}$ and $\dot{\gamma}$ with respect to the angle η and the radius ρ and is, in a whole, a senseless assumption that leads to a wrong flow estimation. We will delve into this issue in the following sections. For further discussions, we will call the first method of this section the "Local-Constancy-in-the-Transformed-image-method" (LCT). The latter we will call the "Local-Affinity-in-the-Transformed-image-method" (LAT). These names are supposed to reflect the fact, that the assumptions of local constancy and affinity are applied to the transformed image.

3.3 The Polar Deformation

In this section we will restrict our studies to the polar mapping. The polar transformation is defined as the log-polar mapping, but without the application of the logarithm. By restricting our studies to the polar mapping we exclude errors that are introduced by the logarithmic subsampling. As explained earlier in section 2.4 which is the differential analogy of the technique by Lucas and Kanade, the polar transform maps straight lines and edges into curves[2] which eliminates the aperture problem (see section 3.1.4). This means that the aperture problem appears at points in the polar plane where the aperture problem does not occur in the corresponding points in the Cartesian image. Or, other way around, all of a sudden it is possible to compute both components of the optical flow at points in the polar plane where the corresponding Cartesian image does not show curvature. Of course, the superficial elimination of the aperture problem produces optical flow values that show a large error regarding the expected motion field.

To analyze this problem we will use the approach of [Tretiak and Pastor, 1984] from section 3.1.2. This technique is a second derivative method. It is constrained by the second-order derivatives that provide the neighborhood information: the Hessian matrix describes the local variance of the image gradient. In case of the aperture problem where the variance of the image gradient vanishes, the Hessian matrix becomes singular. It is therefore sufficient to study the rank of the Hessian matrix of a point (ρ, η) in the polar plane and the rank of the Hessian of the corresponding Cartesian point (x, y). According to our observation that the polar mapping enables the computation of both flow components at points without curvature in the original Cartesian image we expect that the Hessian in the polar plane has full rank at points where the Hessian matrix in the Cartesian plane is singular. In a whole the aperture problem is always eliminated by assuming local constancy and local affinity of the optical flow in the polar and log-polar plane, as we have done in LCT and LAT. A way out of this problem is to apply the assumptions of constancy and affinity to the back-transformed flow (u, v) in the Cartesian plane instead of applying it to the polar flow (u^p, v^p):

$$\frac{d}{dt}I \text{ instead of } \frac{d}{dt}E \ .$$

We denote by $E(\rho, \eta)$ the gray-value function of polar coordinates, $I(x, y)$ denotes the

[2]This is a property of the polar mapping and not a property of the logarithmic subsampling. Thus both, the log-polar and the polar mapping match lines into curves and we can restrict us, without loss of generality, to the polar transform.

gray-value function of Cartesian coordinates. With this we allow first a harmonic variation of the polar flow with the angle η. Second, we face the aperture problem in the Cartesian plane and the Hessian in the polar plane will have the same rank as the Cartesian Hessian.

Using the second derivative technique by [Tretiak and Pastor, 1984] in section 3.1.2 we get with these notations the following over-constrained equation system

$$\begin{aligned} E_\rho u^p + E_\eta v^p + E_t &= 0 \\ E_{\rho\rho} u^p + E_{\eta\rho} v^p + E_{t\rho} &= 0 \\ E_{\rho\eta} u^p + E_{\eta\eta} v^p + E_{t\eta} &= 0 \ . \end{aligned} \tag{3.12}$$

We are now interested in the rank of the matrix

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} \\ E_{\rho\eta} & E_{\eta\eta} \end{pmatrix} .$$

By twice differentiating $E(\rho,\eta)$ it can be easily proved (see Appendix C for the proof) that

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} \\ E_{\rho\eta} & E_{\eta\eta} \end{pmatrix} = \frac{\partial(x,y)}{\partial(\rho,\eta)}^T \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \frac{\partial(x,y)}{\partial(\rho,\eta)} + I_x \begin{pmatrix} \frac{\partial^2 x}{\partial\rho^2} & \frac{\partial^2 x}{\partial\rho\partial\eta} \\ \frac{\partial^2 x}{\partial\rho\partial\eta} & \frac{\partial^2 x}{\partial\eta^2} \end{pmatrix} + I_y \begin{pmatrix} \frac{\partial^2 y}{\partial\rho^2} & \frac{\partial^2 y}{\partial\rho\partial\eta} \\ \frac{\partial^2 y}{\partial\rho\partial\eta} & \frac{\partial^2 y}{\partial\eta^2} \end{pmatrix} \tag{3.13}$$

with

$$\frac{\partial(x,y)}{\partial(\rho,\eta)} = \begin{pmatrix} \frac{\partial x}{\partial\rho} & \frac{\partial x}{\partial\eta} \\ \frac{\partial y}{\partial\rho} & \frac{\partial y}{\partial\eta} \end{pmatrix} .$$

All the matrices with derivatives of (x, y) with respect to (ρ, η) have full rank up to two values of η for the second derivatives. Hence, the singularity of the Cartesian Hessian does not lead to the singularity of the polar Hessian which also depends on the Cartesian gradient.

It is plausible to suppose that the smallest singular value of the coefficient matrix – used as a confidence measure – will be higher in the system (3.12) than in the equivalent system in the Cartesian plane. This fact causes the acceptance of erroneous optical flow values.

We proceed by substituting the assumption of local flow constancy in (u^p, v^p) with local flow constancy in (u, v) before applying the polar transformation. We differentiate the Brightness Change Constraint Equation assuming that the spatial derivatives of (u, v) vanish and we obtain a system with the new coefficient matrix (see Appendix C for the proof)

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} - \frac{E_\eta}{\rho} \\ E_{\rho\eta} - \frac{E_\eta}{\rho} & E_{\eta\eta} + \rho E_\rho \end{pmatrix} = \frac{\partial(x,y)}{\partial(\rho,\eta)}^T \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \frac{\partial(x,y)}{\partial(\rho,\eta)} \tag{3.14}$$

which proves that the singularity of the Cartesian Hessian matrix is the necessary and sufficient condition for the singularity of the new coefficient matrix. Hence, the second derivative system with coefficient matrix (3.14) does not introduce erroneous values of the optical flow like the system (3.12). Based on this fact we are going to construct in the next section a gradient-sampling method similar to the *LCT* method, but with assumptions transferred to the Cartesian flow.

3.4 Assumptions Concerning the Cartesian Plane

In this section we transfer the assumptions about constancy and affinity of the *Cartesian* flow to the application of the BCCE in the neighborhood pixels $(\xi+\delta\xi, \gamma+\delta\gamma)$. Or in different words, we assume for the neighborhood $(\xi+\delta\xi, \gamma+\delta\gamma)$ that the corresponding Cartesian flow is constant. Hence, we start out with the BCCE

$$\begin{pmatrix} J_\xi(\xi+\delta\xi,\gamma+\delta\gamma) & J_\gamma(\xi+\delta\xi,\gamma+\delta\gamma) \end{pmatrix} \begin{pmatrix} u^l \\ v^l \end{pmatrix} = -J_t(\xi+\delta\xi,\gamma+\delta\gamma) \tag{3.15}$$

and we allow the log-polar flow to vary

$$\begin{pmatrix} u^l \\ v^l \end{pmatrix} = \frac{1}{\rho_0 a^{\xi+\delta\xi}} \begin{pmatrix} \frac{\cos(\gamma+\delta\gamma)}{\ln a} & \frac{\sin(\gamma+\delta\gamma)}{\ln a} \\ -q\sin(\gamma+\delta\gamma) & q\cos(\gamma+\delta\gamma) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \tag{3.16}$$

whereas the Cartesian flow shall remain constant. Combining these two equations we get

$$J_t = -\frac{1}{\rho a^{\xi+\delta\xi}} \begin{pmatrix} J_\xi & J_\gamma \end{pmatrix} \begin{pmatrix} \frac{\cos(\gamma+\delta\gamma)}{\ln a} & \frac{\sin(\gamma+\delta\gamma)}{\ln a} \\ -q\sin(\gamma+\delta\gamma) & q\cos(\gamma+\delta\gamma) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} . \tag{3.17}$$

Applying this equation in the same way as explained in section 3.1.1 we can solve for u and v.

Up to now the Cartesian flow was assumed to be constant within a local neighborhood. The next step is straight forward: We allow the Cartesian flow to vary linearly in the local neighborhood which leads to the system of linear equations

$$J_t = -\frac{1}{\rho_0 a^{\xi+\delta\xi}} \begin{bmatrix} g_x & g_y & g_x\delta x & g_x\delta y & g_y\delta x & g_y\delta y \end{bmatrix} \begin{bmatrix} u \\ v \\ u_x \\ u_y \\ v_x \\ v_y \end{bmatrix} \tag{3.18}$$

for every $(\xi+\delta\xi, \gamma+\delta\gamma)$ in the neighborhood of (ξ, γ) with

$$\begin{aligned} g_\xi &= \frac{1}{\ln a} J_\xi \Big|_{(\xi+\delta\xi,\gamma+\delta\gamma)} \\ g_\gamma &= q J_\gamma|_{(\xi+\delta\xi,\gamma+\delta\gamma)} \\ g_x &= g_\xi \cos(\eta+\delta\eta) - g_\gamma \sin(\eta+\delta\eta) \\ g_y &= g_\xi \sin(\eta+\delta\eta) + g_\gamma \cos(\eta+\delta\eta) \\ \delta x &= (\rho_0 a^{\xi+\delta\xi})\cos(\eta+\delta\eta) - \rho_0 a^\xi \cos(\eta) \\ \delta y &= (\rho_0 a^{\xi+\delta\xi})\sin(\eta+\delta\eta) - \rho_0 a^\xi \sin(\eta) \ . \end{aligned}$$

We obtain for every log-polar pixel of the neighborhood an equation with six unknowns u, u_x, u_y, v, v_x, v_y. In order to obtain the values for $\dot{\xi}$ and $\dot{\gamma}$, u and v were again transformed using (3.5) and (3.6). The Cartesian flow derivatives can be used to compute the differential invariants of the flow like divergence (u_x+v_y), rotation (v_x-u_y), and shear (u_x-v_y, u_y+v_x). By setting $\xi = 0$ and $\gamma = 0$ we obtain the equations derived by [Tunley and Young, 1994] as a special case.

Six or more measurements of J_t, J_ξ and J_γ in the local neighborhood lead to an (overconstrained) equation system. We expect that the resulting system does not have full rank at the positions where the aperture problem appears in the Cartesian plane and, in general, that the resulting coefficient matrix has lower singular values than the coefficient matrix A in (3.8).

It is worth to mention that the assumptions of a constant flow (u, v) or a affine flow $(u_0 + u_x\delta x + u_y\delta y, v_0 + v_x\delta x + v_y\delta y)$ are the Taylor expansions of $\dot{x}$ and $\dot{y}$ up to the zeroth and first order. A Taylor expansion up to the zeroth and first order of $\dot{\xi}$ and $\dot{\gamma}$, expressed in terms of Cartesian coordinates, leads exactly to the equation systems for the optical flow used in this section. In this sense, substituting $\dot{\xi}$ and $\dot{\gamma}$ with the equations (3.5) and (3.6) is not only justified by the merely intuitive explanation with the aperture problem but is also justified by the Taylor expansion as an approximation of $\dot{\xi}$ and $\dot{\gamma}$.

We will call the first method the "<u>L</u>ocal-<u>C</u>onstancy-in-the-<u>C</u>artesian-image-method" (*LCC*) and the latter we call "<u>L</u>ocal-<u>A</u>ffinity-in-the-<u>C</u>artesian-image-method" (*LAC*).

3.5 Experimental Comparison

In this section we present results that compare the four flow computation methods described in the previous sections and show the effect of the polar deformation on the singular values which were used as a confidence measure.

The local constancy and the local affinity assumptions are applied in 5×5 neighborhoods. The least squares problems are solved with the Singular Value Decomposition (SVD) and a threshold is applied to the smallest singular value as a reliability criterion. For a SVD we used the algorithm from [Press *et al.*, 1986]. Image gradients were derived using binomial filters.

We first tested the Local Constancy methods on the polar transform of an image sequence consisting of black squares on white background moving with uniform velocity $\vec{u} = (1, 1)$. The polar transformed image is shown on the left of Fig. 3.4. The images on the middle and right of Fig. 3.4 show the smallest singular value of the linear systems using the Local Constancy assumption in the polar (LCT) and the Cartesian (LCC) image, respectively. As we already expected from eq. (3.15) the LCC method produces a system with lower singular values. Thus, the aperture problem should not be overcome by introducing non-existing gray-value curvature.

To test the methods proposed in sections 3.2 (LCT and LAT, which applied the Cartesian constraints to the transformed flow) and in section 3.4 (LCC and LAC, which applied the Cartesian constraints to the Cartesian flow) we used the real sequence "Marbled Block"[3] [Otte and Nagel, 1994]. The original image of the sequence and its log-polar transform are shown in figure 3.5 left and middle, respectively. In Fig. 3.5 (right) we show

[3]Created by Michael Otte at University of Karlsruhe and FhG-IITB, Germany.

the flow field computed with the LAC method from section 3.4. The *ground-truth* values of the "Marbled Block" sequence for the *correct* 3D motion field were known. Furthermore, we estimated with conventional Cartesian flow techniques the Cartesian flow. This flow includes usual estimation errors due to bad gray value structure and noise. Both, the ground truth and the Cartesian estimate were transformed into log-polar flow components. These two types of reference flows will be called *transformed ground truth* and *transformed Cartesian estimate*.

We then estimated the optical flow in the log-polar transformed "Marbled Block" sequence. We compared the estimated flow to the transformed ground-truth flow. Further, we compared the estimated flow to the transformed Cartesian estimate. The error with respect to the ground truth contains errors that are due to the polar deformation, to the logarithmic subsampling as well as to bad texture and noise. The error with respect to the latter contains only errors that are due to the log-polar transform and can therefore be considered as a lower bound for the error to be expected.

The error measurements used here are the relative error, given in percentage, and the angular error, given in degrees and were also used by [Barron *et al.*, 1992]. Compared are, at a time, the two vectors $(u, v, 1)$ and $(\hat{u}, \hat{v}, 1)$, where (u, v) and $(\hat{u}, \hat{v})$ are the ground-truth (or the transformed Cartesian estimate) and the estimated flow, respectively. The length of a flow vector is given by its Euclidean norm. The relative error between two vectors is defined as the difference of length in percentage between a flow vector in the estimated flow field and the corresponding flow vector in the reference flows:

$$\frac{\|(\hat{u} - u, \hat{v} - v)\|_2}{\|(u, v)\|_2} \cdot 100 \ .$$

For example, if the Euclidean norm of an estimated vector is 10 pixels and the Euclidean norm of the corresponding correct vector is 9 pixels, then the result is an error of 10 %. The angular error between two vectors is defined as the difference in degrees between the direction of the estimated flow vector and the direction of the corresponding correct flow vector (see fig. 3.3).

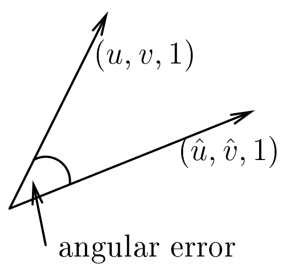


Figure 3.3: *The angular error, given in degrees, between* $(\hat{u}, \hat{v}, 1)$ *and* $(u, v, 1)$.

The comparisons of the estimated flow with the ground-truth flow and with the transformed Cartesian estimates were done using an error program that was also used by [Barron *et al.*, 1992] in a similar version. It performs the error calculations just explained for every flow vector for which the smallest singular value is above a given threshold. The threshold is given as a parameter to the flow algorithms. The error program takes for the calculation of the average errors into account only those estimated flow vectors for which the smallest singular value was above the given threshold. All other vectors are less reliable and are therefore ignored. The outputs of the error program are the average relative error, the average angular error and the density. The density is defined as the fraction of the estimates with smallest singular value above a threshold that varies for the four estimation techniques.

The following tables (3.1) and (3.2) show the column "Techniques", in order to compare the different flow techniques, and the columns "av. ang. err.", "av. rel. err." and "density", which are the outputs of the error program.

In table (3.1), we first present the angular- and relative errors for the *polar* transform of the sequence obtained with angular resolution of 512 samples per 360 degrees and radial resolution equal to the original (256 pels). Choosing the angular resolution so high ensures the suppression of an angular subsampling. We deal therefore only with the polar deformation. As it could be expected, the error is lower when we compare the polar estimates to the transformed Cartesian estimates. Regarding the local constancy assumption applied on the polar (LCT) and the Cartesian (LCC) plane the errors are about the same for the same density. However, this density is achieved for appropriately high chosen threshold for the LCT method. This is clear because we employed in both cases the same underlying estimation technique, but with the difference of applying the assumptions in the transformed and in the Cartesian plane. The erroneous vectors estimated with the LCT method can be filtered out by setting the threshold for the lowest singular values high enough. However, this can only be done by hand, and choosing the right threshold is a difficult procedure because a unique threshold for all sequences can not be found. In our tests we choose the threshold such, that the density of the LCC and the LCT methods were about the same. The superiority of the LCC method is shown if we compare it to the performance of the LCT method with the same threshold (LCT-thr). The difference in error is due to the falsely estimated vectors that are due to the incorrect assumptions and that are filtered out automatically by the LCC method.

The log-polar transform of the "Marbled Block" sequence is obtained with angular resolution of 128 samples/360 degrees and radial resolution of 45 samples for the radial range [32..356]. It should be noted that the angular resolution is one-fourth of the angular resolution of the polar transform, therefore the error is due to both the logarithmic and the polar subsampling. The errors with respect to the transformed ground-truth values are not significantly higher than to the transformed estimates what means that the effects of noise and poor gray-value structure in the original are inferior to the subsampling effects. The errors are shown for about the same density in order to compare the performance at points where the coefficient matrices are regular in all methods. We are thus strict to the methods using assumptions in the Cartesian plane. Even with such a high relative error we can use the log-polar transform for 3D-analysis. It is shown in [Daniilidis, 1994] that the 3D-translation direction can be computed with only 5 degrees of error.

Transformed ground truth			
Technique	av. ang. err.	av. rel. err.	density
LCT	5.64729	18.74557	0.54977
LCT-th	7.26917	23.65154	0.77566
LCC	6.02655	19.11052	0.48914
Transform of the Cartesian estimate			
LCT	3.68909	12.36976	0.54977
LCT-thr	3.78853	13.56477	0.77566
LCC	3.63529	12.04593	0.48914

Table 3.1: *Error statistics for the polar transform of the "Marbled Block" sequence (see text for explanation).*

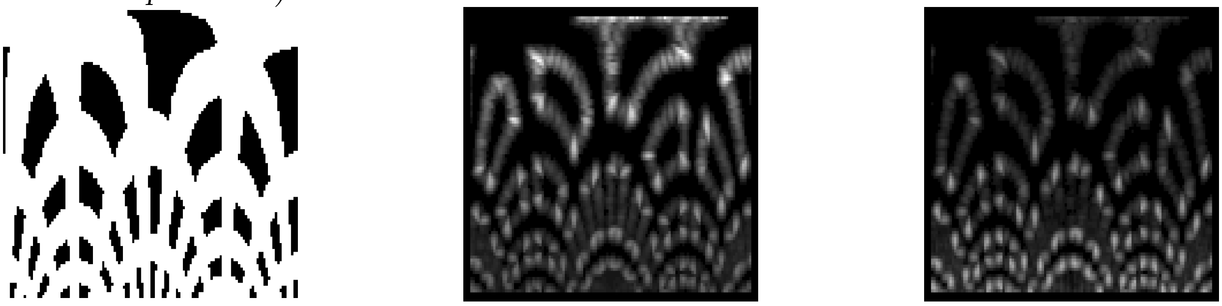


Figure 3.4: *The polar transform of an image with black rectangles (left), and the smallest singular value of the linear system resulting from the LCT (middle) and the LCC (right) assumptions, respectively. The horizontal axis is the angular axis and the vertical axis is the radial axis.*

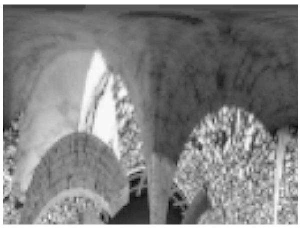
3.6 Summary and Conclusion

In the preceding chapter we addressed the problems that are introduced by the polar deformations for the computation of optical flow. After giving first a short introduction to differential flow techniques, we applied these techniques in section 3.2 to log-polar transformed image sequences. In section 3.3 we analyzed the polar deformation and explained that errors in the estimated flow are due to fictitious gray value structures that are introduced by the polar deformation. We explained further that assumptions that are done in conventional flow techniques are only valid on the Cartesian grid and that assumptions of constancy or affinity of optical flow are not justified for transformed images. As a way out of this problem we introduced the two new optical flow techniques *LCC* and *LAC* that related the Cartesian assumptions to the polar deformation and with this led to better results.

This chapter has shown that it is not easily possible to apply Cartesian vision techniques to log-polar mapped images because ignoring the characteristics and properties of the log-polar mapping leads to unjustified and even erroneous assumptions. The result is a failure

Transformed ground truth			
Technique	av. ang. err.	av. rel. er	density
LCT	5.34357	34.53929	0.71908
LCC	4.95797	28.85278	0.56144
LAT	5.30501	34.45435	0.75163
LAC	4.10992	26.02044	0.67358
Transform of the Cartesian estimate			
LCT	4.26153	32.02543	0.71908
LCC	3.80384	26.88554	0.56144
LAT	4.29278	32.22566	0.75163
LAC	3.28657	24.28787	0.67358

Table 3.2: *Error statistics for the log-polar transform of the "Marbled Block" sequence (see text for explanation).*



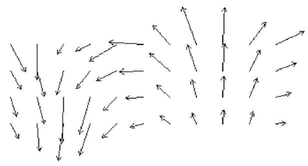


Figure 3.5: *The Cartesian original image 512x512 pixels(left), its log-polar transform 77x128 pixels shown magnified (middle), and the computed optical flow (right) of the "Marbled Block" sequence. The horizontal axis is the angular axis and the vertical axis is the logarithmic radial axis.*

of an incorrectly applied technique. Instead, care *can* and *has* to be taken if Cartesian techniques are applied in the log-polar domain so that assumptions are not violated.

Chapter 4

Spatial–Temporal Derivatives in the Log–Polar Plane

4.1 Introduction

Two types of errors have to be considered in optical flow estimation. Errors of the first type are introduced by the geometric deformation due to the polar mapping, errors of the second type are due to the logarithmic subsampling. Errors that are introduced by the polar transformation have been analyzed in the previous chapter. In this chapter we will address the filtering problems that arise from the application of derivative filters to sequences transformed with the logarithmic polar mapping. The goal of this chapter is to study the effects of the logarithmic polar transformation on image sequences and thus on all subsequent filtering and computation steps. In order to focus the studies on the errors introduced by the logarithmic subsampling scheme, we restrict our considerations on logarithmically subsampled 1-D signals.

In the following we first introduce an *intermediate lossless* signal. This lossless signal is a logarithmic version of a homogeneously sampled signal and is defined such that the reconstruction of the original signal is possible without loss. This will give us the possibility to use linear shift invariant filtering and analysis methods on the logarithmic data. The logarithmically subsampled signal as it is defined in chapter 2 is related to the lossless signal by only a scaling factor. To obtain the logarithmically subsampled version from the lossless signal it is sufficient to employ the regular pyramid steps: low-pass filtering, subsampling and shrinking [Burt, 1984; Meer *et al.*, 1987].

After its introduction we will use the lossless signal to focus on three points:

1. We will show that logarithmically subsampled 1-D images and sequences contain higher spatial frequencies than their corresponding Cartesian ones and that this is a source of errors for the filter response of FIR filters that usually perform well on synthetic and noise free signals of low frequency content.

2. We will study explicitly the effects of these errors on the calculation of optical flow. In this context we will show that the reliability of the estimated flow does not only

depend on the correct estimation of the derivatives but that it also depends on the smoothing in space and in time and, furthermore, that it depends on *how* the smoothing is done. This will lead to the conclusion that if the flow is to be estimated correctly, the smoothing and derivative filters that are applied can not be designed independently from each other but instead have to be designed together so that they fulfill some important requirements.

3. Based on these observations we will show that conventional approximations of Gaussian derivative filters, like binomial filters, do not fulfill these requirements for higher spatial frequencies and that they will lead therefore to an erroneous estimation of the optical flow in logarithmically subsampled signals which as stated under 1. contain higher frequencies than usually. Therefore we will propose two filters for smoothing and for differentiation that are based on frequency space considerations as well as on these requirements. These filters perform significantly better for the estimation of optical flow on the logarithmically transformed signals that the common binomial filters as we will show in standard test sequences.

A similar notion of the intermediate lossless log-polar image was introduced by [Porat and Zeevi, 1988] but with respect to the reconstruction of a non-uniformly subsampled signal. Most derivative filters applied until now in computer vision are not based on frequency domain considerations [Hashimoto and Sklansky, 1987]. [Jähne, 1993] proposed a Taylor expansion in the frequency domain that preserves the low frequencies. Approaches oriented to differential geometric description of the gray-value surface [Karabassis and Spetsakis, 1995; Vieville and Faugeras, 1994] try to preserve the local polynomial image structure, therefore they are based on local fitting of polynomial models. The most related to our work is done by Simoncelli [Simoncelli, 1994] who approximates the smoothing and the derivative filter in the frequency domain giving, however, low weights to the high frequencies. This is not the case in immensely subsampled images like the peripheral parts of the log-polar image, therefore we try to equally treat the entire spectrum. Furthermore our filter design method takes explicitly in consideration the requirements needed of a correct flow estimation. All approaches will be presented and extensively discussed in section 4.4.

4.2 The Logarithmic Subsampling

In order to focus our studies on errors introduced by the logarithmically subsampling scheme we restrict ourselves to a discrete 1D gray-value signal $g(\rho)$, $\rho = \rho_0 \dots \rho_{\max}$ and its logarithmically subsampled version $\gamma(\xi)$, $\xi = 0 \dots N_r$. The mapping is done according to the definitions in chapter 2, the base a of the logarithm is given by (2.5).

We now introduce an intermediate function $\lambda(\xi)$ [Porat and Zeevi, 1988] that is obtained by exact coordinate transformation

$$\lambda(\xi) = g\left(\rho_0 b^{\xi}\right) \tag{4.1}$$

and which is illustrated in Figure 4.1. We see that, according to the definition of the

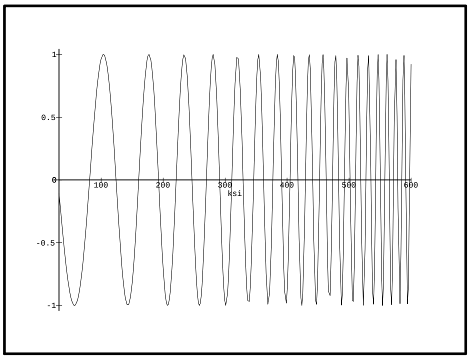


Figure 4.1: *Lossless transform of a sine.*

logarithmic subsampling, the sine occupies for one period more space close to the origin than at greater distance. In order to transform the signal g without any loss the basis b has to be chosen in such a way that even the most peripheral pixel is exactly transformed: $b\rho_{\max} - 1 = \rho_{\max}$. We therefore choose $b = \frac{\rho_{\max}}{\rho_{\max}-1}$.

It is interesting to mention that calculating the logarithmic signal $\gamma(\xi)$ directly from $g(\rho)$ while using the definition of chapter 2 is completely equivalent to calculating first the intermediate lossless signal $\lambda(\xi)$ and then applying regular pyramid steps in order to obtain $\gamma(\xi)$. Since the lossless signal $\lambda(\xi)$ contains all visual information of $g(\rho)$, the loss of information caused by the logarithmic subsampling of $g(\rho)$ is the same as the information loss caused by the pyramid steps applied on the lossless $\lambda(\xi)$.

The lossless signal $\lambda(\xi)$ has dimension M. If the logarithmically subsampled signal $\gamma(\xi)$, as defined in chapter 2, has dimension N_r, then the lossless signal $\lambda(\xi)$ is related to the logarithmic signal $\gamma(\xi)$ by a scaling factor M/N_r. Thus $\gamma(\xi)$ can be obtained from $\lambda(\xi)$ by the three transition steps in a regular pyramid with subsampling interval M/N_r: The linear shift invariant low-pass filtering step, the subsampling step and the shrinking step. As an example, for a signal $g(\rho)$ with $\rho_0 = 32 < \rho < \rho_{\max} = 256$ and $N_r = 64$ we have $b = 1.00392$ and $M = 531.30$. The subsampling factor is ≈ 8 octaves.

To understand the logarithmic subsampling effects it is therefore sufficient to study the effects on the lossless signal caused by the regular pyramid steps. This we will do in the following three sections.

4.2.1 Frequential Properties and Relationships

We will now describe the main frequency relationships between the original signal g and its logarithmically-subsampled version γ. This is important for a correct application of the pyramid steps.

Because the signal γ is subsampled logarithmically, the sampling distance depends on the distance ρ from the origin and it increases linearly with increasing ρ. For an arbitrary ρ with $\rho_0 \leq \rho \leq \rho_{\max}$ the local sampling distance of signal g at radius ρ is given by $\delta\rho = \rho\left(\sqrt{a} - \frac{1}{\sqrt{a}}\right)$ (appendix B, lemma 1; see also chapter 2) and $\delta\rho$ is *not* constant as

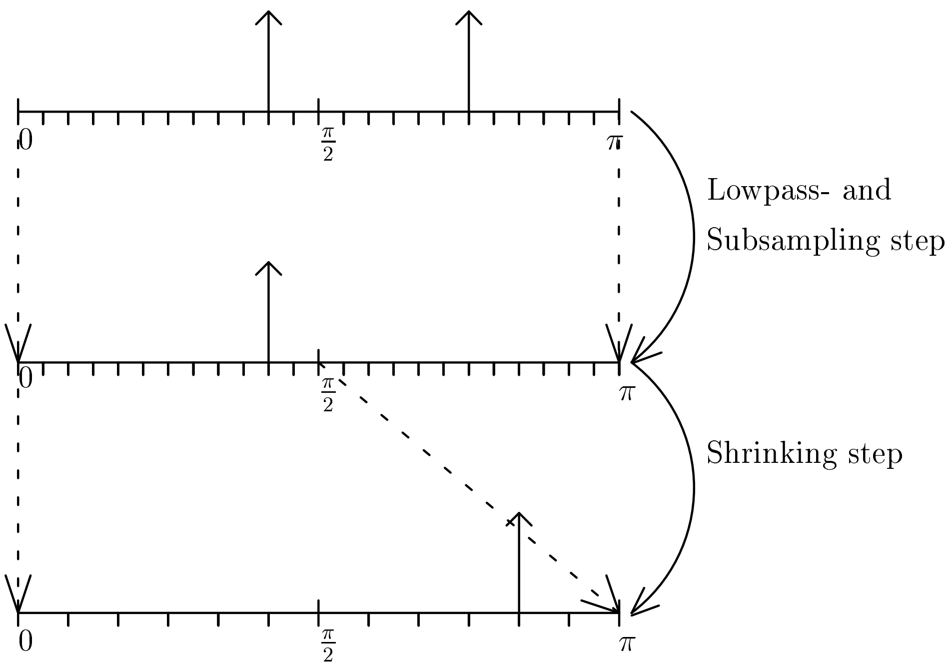


Figure 4.2: *The subsampling and the shrinking steps for one octave demonstrated in the frequency domain: We see two spikes in the frequency space. The low-pass filter and the subsampling step cuts all frequencies above $\pi/2$. The shrinking step stretches the signal.*

usual. The local sampling frequency of g at radius ρ is given by the reciprocal of the local sampling distance $\frac{1}{\delta\rho}$. During transformation of $g(\rho)$ into the intermediate lossless $\lambda(\xi)$ the signal $g(\rho)$ is stretched out by the factor $\frac{\ln a}{\ln b}$ if ρ is close to ρ_0. In the periphery for a ρ close to $\rho_{\max}$ the signal $g(\rho)$ is practically not deformed (appendix B, lemma 3), as required by the definition of the basis b.

As an interpretation, edges in g that are close to ρ_0 are smoothed, so that the frequencies are somewhat "slowed down" by a factor $\geq (\frac{\ln a}{\ln b})^{-1}$ which is the same as the local sampling distance. High frequencies of structures in the periphery of g are preserved in λ because for $\rho = \rho_{\max}$ the local sampling distance is the same as the one of g.

4.2.2 The First Pyramid Step: Smoothing the Lossless Signal

The first step in a regular pyramid is the low-pass filtering which is needed in order to suppress all frequencies above half the Nyquist frequency. It ensures that the sampling theorem is satisfied during the subsampling and shrinking steps. We will use a Gaussian filter as a low-pass filter for the signal g that allows all frequencies less equal to $f_{\max} = \frac{1}{2\delta\rho}$ $(\rho_0 \leq \rho \leq \rho_{\max})$. In order to find in a ordinary logarithmically sampled signal a suitable σ for some ρ so that the low-pass-cut-frequency of the Gaussian is close to the desired $f_{\max}$, we solve $\exp(-2\pi^2\sigma^2 f_{\max}^2) = \epsilon$ for σ^2, assuming $\epsilon > 0$. Thus

$$\sigma^2 = -\ln(\epsilon)\frac{1}{2\pi^2 f_{\max}^2} \quad . \tag{4.2}$$

Substituting $\frac{1}{2\delta\rho}$ for $f_{\max}$ in (4.2) we get σ^2 as a function of ρ:

$$\sigma^2(\rho) = -\ln(\epsilon)\frac{(2\delta\rho)^2}{2\pi^2} \quad . \tag{4.3}$$

However, all these considerations do *only* apply for signals on the logarithmic lattices and they reflect the difficulties for a low-pass filtering that come with the logarithmic sampling scheme. In order to design a Gaussian filter for the lossless signal such that λ has the desired low-pass-cut-frequency, a sufficient σ_λ is much easier to find and is given as $\sigma_\lambda = \sigma(\rho_{\max})$ (appendix B, lemma 3) which is the same σ as the one that would have had to be applied on the original signal since the sampling distance of the original signal and the sampling distance at $\rho_{\max}$ of λ is the same.

4.2.3 The Last Pyramid Steps: Subsampling and Shrinking the Lossless

During the subsampling and shrinking step the signal is stretched out in the frequency space due to the normalization to the new Nyquist frequency of the shrinked image (see Figure 4.2 for an explanation in the frequency domain). The frequency based representation of a discrete signal is always done in relation to its pixel size and to its absolute dimension (e.g. given in meters). Let us assume that the original image $g(x)$ contains only frequencies up to half the Nyquist frequency, and that the signal has dimension (in meters) N_m, discretizised by N samples or pixels. The signal $g_s(x) = g(2x)$, pixel wise $x = 0, \ldots, M-1$, is assumed to have the same dimension (in meters) N_m, but it is discretizised by only $M = N/2$ pixels (samples). If $g_s(x)$ has the same dimension (in meters) as g then the pixels of g_s must have double the size of the pixels of g. Therefore, g_s can only show a maximal frequency that is half the maximal frequency of g.

For g, the sampling interval δx equals one, but for g_s, $\delta x_s = 2\delta x = 2$. Therefore, in this case, the maximal ω for g_s is defined in terms of δx and $\omega_{\max}$ equals thus $\pi/2$, while the following equations are all given in terms of the constant size N:

$$\begin{aligned} \hat{g}(\omega) &= \sum_{x=0}^{N-1} g(x) e^{\frac{\omega x}{N}} \\ &= \sum_{x=0}^{M-1} g(2x) e^{\frac{\omega 2x}{N}} \\ &= \sum_{x=0}^{M-1} g_s(x) e^{\frac{\omega 2x}{N}} \quad . \end{aligned}$$

However, relating the maximal ω for g_s to $\delta x_{ss} = 1$ means to divide the dimension (in meters) of g_s by two. We get a signal with dimension (in meters) N_n, which we may denote by g_{ss}. For this signal g_{ss}, $\omega_{\max}$ equals π while the number of pixels is $M = N/2$:

$$\begin{aligned}\hat{g}_{ss}(\omega) &= \sum_{x=0}^{M-1} g_s(x) e^{\frac{\omega x}{M}} \\ &= \sum_{x=0}^{M-1} g_{ss}(x) e^{\frac{\omega x}{M}} \quad ,\end{aligned}$$

Therefore, even though $g(2x) = g_s(x) = g_{ss}(x)$, while assuming $\delta x = \delta x_{ss} = 1$, $\hat{g}_{ss}(\omega)$ contains frequencies that are twice as high as the frequencies of $\hat{g}_s(\omega)$. This is, as explained, due to the normations to different dimensions (in meters) but constant pixel sizes: $g_s(x)$ is the subsampled version of $g(x)$ and $g_{ss}(x)$ is the shrinked version of $g_s(x)$. Summarizing, we can say that subsampling equals a reduction in pixel resolution by some factor k while keeping the dimension (in meters) of the signal constant:

$$\begin{aligned}\delta x &\overset{\text{subsample}}{\longrightarrow} \delta x_k = k\delta x \\ g(x) &\overset{\text{subsample}}{\longrightarrow} g_s(x) \quad .\end{aligned}$$

The shrinking step corresponds to a reduction of the dimension (in meters) by the same factor k:

$$\begin{aligned}N_m &\overset{\text{shrink}}{\longrightarrow} N_n = N_m/k \\ g_s(x) &\overset{\text{shrink}}{\longrightarrow} g_{ss}(x) \quad .\end{aligned}$$

A signal that has frequencies only up to $\frac{\pi}{2}$ contains thus after the shrinking step of one octave frequencies up to π (if, certainly, the signal is properly low-pass filtered).

As another and more intuitive explanation, let us imagine a sine with a period length of four pixels: $(0\ 1\ 0\ -1\ 0\ 1\ 0\ -1\ 0\ \ldots)$. After subsampling and shrinking the cosine as an interval length of two: $(1\ -1\ 1\ -1\ \ldots)$, the frequency is twice the one before subsampling and shrinking.

It is very important to understand the frequency responses of space invariant filters are the same before and after the transition steps because they are always related to $\delta x = 1$ and the dimension (in meters) of the image they are applied on. Therefore, the frequency response of a linear shift invariant filter is not effected by the subsampling steps.

Assuming a noise free and band-limited signal g with small frequencies, the transformation of g into the lossless signal λ does not introduce higher frequencies (see section 4.2.1). However, the pyramid steps, applied to the lossless signal, do, as just explained. Since a log-polar transformed signal corresponds to a lossless signal that has been subsampled and shrinked several times, a logarithmically subsampled signal contains, in a whole, higher frequencies than the original signal g[1].

[1]This is true if the parameters of the log-polar mapping are well defined (section 2.6).

4.3 The Effects on Spatio-Temporal Filtering

In this chapter we will study and explain effects of the logarithmic subsampling scheme on the estimation of optical flow. The optical flow is assumed to be calculated with differential techniques from chapter 3.

As explained above, a signal is stretched out in frequency space by the subsampling and shrinking step of a regular pyramid. Assuming a noise free and band-limited signal that contains only low frequencies, the signal represented in a lower resolution layer of the pyramid contains therefore higher frequencies than the same signal represented in a higher resolution layer. This is also true for the lossless signal and its subsampled versions.

We have mentioned above that the frequency response of a filter is the same for all resolution layers. However, the frequency response of the filter will not be optimal for all frequencies under its frequency support. For example, common filters for smoothing and differentiating are the binomial filters that approximate the Gaussian and its derivatives. They have been studied and derived by [Hashimoto and Sklansky, 1987] and will be explained precisely in section 4.4.3. These filters show a good performance for low frequencies, but they show a bad one for higher frequencies. Applying the binomial filters on different resolution layers will therefore deliver poorer results in the lower pyramid layers where the signals contain higher spatial frequencies and where therefore higher frequencies occur under the frequency support of the filter. In higher layers of the pyramid the occurring frequencies under the frequency support of the filters are low so that the frequency response of the binomial filters are sufficiently good.

Differentiation of a signal in image space corresponds to a multiplication of the signal with $(j\omega)$ in frequency space. The frequency response of the binomial derivative filter approximates therefore the product $(j\omega) \cdot \hat{D}_0$, where $\hat{D}_0$ is the Fourier transform of the Gaussian. In Figure 4.3 (left), we can see that the approximation of the ideal first derivative $(j\omega)$ is acceptable only for lower frequencies. Apart from the derivative also the Gaussian itself is approximated in the frequency response of the binomial derivative filter as part of the product with $(j\omega)$. We will see in the following, that it is important for our purposes that the frequency response of the binomial filter is approximated instead of $\hat{D}_0$ or, in other words, it is important that the frequency response of the binomial derivative filter approximates the product "$(j\omega)\times$ *the frequency response of the binomial smoothing filter*". However, figure 4.3, (right) shows, that the frequency response of the binomial filter is not preserved well during differentiation. A bad approximation of the Gaussian smoothing causes problems especially in situations where we have to make comparisons of a smoothed signal with its derivative as, for example, in differential techniques for the estimation of optical flow. We will delve into this issue below.

To study the consequences of these observations on our logarithmically subsampled signals in detail we consider the optical flow u_λ in the lossless signal as a lower error bound for the error in the optical flow u_γ of the subsampled signal γ (λ and γ are defined as in chapter 4.2). Using gradient-based techniques the flow of the 1D lossless and subsampled signals is $u_\lambda = -\frac{\lambda_t}{\lambda_\xi}$ and $u_\gamma = -\frac{\gamma_t}{\gamma_\xi}$, respectively. The temporal gradient γ_t of the logarithmic signal is the response of the lossless temporal gradient to the low-pass filtering

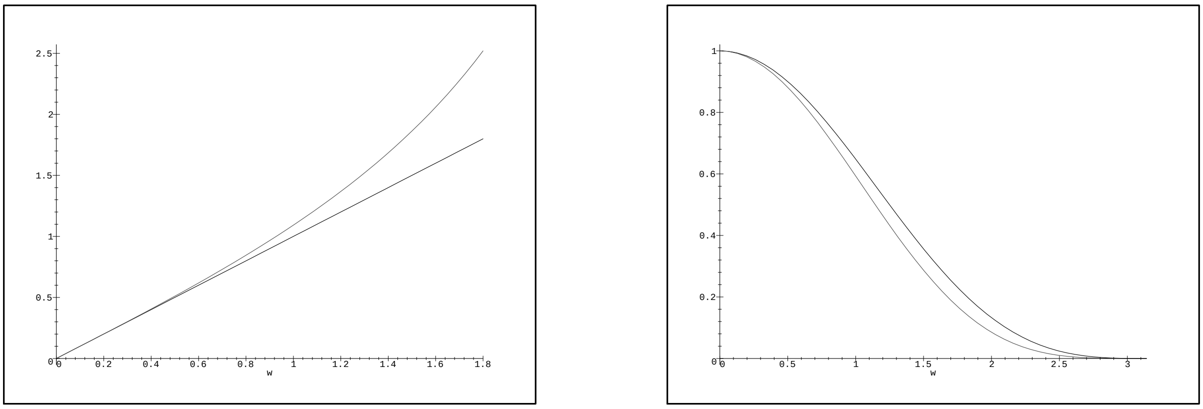


Figure 4.3: *The graph on the left shows the binomial approximation of the first Gaussian derivative in frequency space of which the binomial approximated Gaussian has been divided out (upper curve) in comparison to the ideal differentiator ($j\omega$)(shown are the absolute values for convenience). The graph on the right shows in frequency space the binomial filter and in comparison to the first derivative of binomial Gaussian from which the ideal differentiator ($j\omega$) has been divided out (lower curve).*

step. Studying the spatial gradient γ_ξ we will assume that all filters have the appropriate anti-aliasing characteristic. Then the error in the estimated u_γ is due to two reasons. The first reason of an error is that the spectrum of the lossless image is stretched out by the subsampling and shrinking steps whereas the spectrum of the spatial gradient is the same before and after subsampling and shrinking. The error in the spatial gradient depends on the quality of the approximation of the ideal differentiator ($j\omega$) under the frequency support of the filter itself. In case of the binomial filters the ideal differentiator ($j\omega$) is approximated well only for lower frequencies. Therefore the derivatives are approximated worse in lower resolution layers. The second reason for an error appears when comparisons are made between a smoothed and a differentiated signal as it happens in differential flow techniques. The error is introduced by the inequality of the Gaussian that was applied for smoothing and the Gaussian that is approximated by its derivative filter. In Figure 4.3 (right) a clear difference can be seen[2].

For a better understanding of the second error type let us consider without loss of generality a moving 1D continuous signal $g(x,t) = \cos(2\pi F(x-ut))$ in space x and time t. Its derivatives g_x and g_t are derived by convolution with the first Gaussian derivative in space, or in time respectively, while smoothed with a Gaussian in the corresponding orthogonal direction. Two Gaussian filters are applied here (see appendix A): $(D^x_{1x}D^t_0)(x,t)$ differentiates in direction x with standard deviation σ^x_x and smoothed in direction t with standard deviation σ^x_t; $(D^x_0 D^t_{1t})(x,t)$ differentiates in direction t with standard deviation

[2]Note that the graph in Figure 4.3 (left) is shown only for $\omega < 1.8$. All the other graphs of Figure 4.3 and 4.10 are normalized to $[0..\pi)$.

σ_t^t and smoothed in direction x with standard deviation σ_x^t:

$$\left(D_{1x}^x D_0^t\right)(x,t) = -\frac{x}{2\pi{\sigma_x^x}^3\sigma_t^x} e^{\frac{x^2}{2{\sigma_x^x}^2}} e^{\frac{t^2}{2{\sigma_t^x}^2}} \tag{4.4}$$

$$\left(D_0^x D_{1t}^t\right)(x,t) = -\frac{t}{2\pi\sigma_x^t{\sigma_t^t}^3} e^{\frac{x^2}{2{\sigma_x^t}^2}} e^{\frac{t^2}{2{\sigma_t^t}^2}} \quad . \tag{4.5}$$

The Fourier transforms of (4.4) and (4.5) are given by

$$\left(\hat{D}_{1x}^x \hat{D}_0^t\right)(\omega_x,\omega_t) = (j\omega_x) e^{-\frac{1}{2}\omega_x^2{\sigma_x^x}^2} e^{-\frac{1}{2}\omega_t^2{\sigma_t^x}^2} \tag{4.6}$$

$$\left(\hat{D}_0^x \hat{D}_{1t}^t\right)(\omega_x,\omega_t) = (j\omega_t) e^{-\frac{1}{2}\omega_t^2{\sigma_t^t}^2} e^{-\frac{1}{2}\omega_x^2{\sigma_x^t}^2} \quad . \tag{4.7}$$

The Fourier transform of the example function is

$$\hat{g}(\omega_x,\omega_t) = \frac{\pi^2}{2}\left\{\delta(\omega_x - 2\pi F, \omega_t + 2\pi F u) + \delta(\omega_x + 2\pi F, \omega_t - 2\pi F u)\right\} \quad . \tag{4.8}$$

Using the convolution theorem which says that the convolution in image space is equivalent to the product in frequency space we get by multiplying (4.8) with (4.6) and (4.7):

$$\begin{aligned}\left(\hat{D}_{1x}^x \hat{D}_0^t\right)\hat{g}(\omega_x,\omega_t) = {} & j\frac{\pi^2}{2}\left\{\delta(\omega_x - 2\pi F, \omega_t + 2\pi F u) + \delta(\omega_x + 2\pi F, \omega_t - 2\pi F u)\right\}\omega_x \\ & e^{-\frac{1}{2}\omega_x^2{\sigma_x^x}^2} e^{-\frac{1}{2}\omega_t^2{\sigma_t^x}^2}\end{aligned} \tag{4.9}$$

$$\begin{aligned}\left(\hat{D}_0^x \hat{D}_{1t}^t\right)\hat{g}(\omega_x,\omega_t) = {} & j\frac{\pi^2}{2}\left\{\delta(\omega_x - 2\pi F, \omega_t + 2\pi F u) + \delta(\omega_x + 2\pi F, \omega_t - 2\pi F u)\right\}\omega_t \\ & e^{-\frac{1}{2}\omega_t^2{\sigma_t^t}^2} e^{-\frac{1}{2}\omega_x^2{\sigma_x^t}^2} \quad .\end{aligned} \tag{4.10}$$

The inverse Fourier transformation leads to the desired derivatives g_x and g_t:

$$g_x = 2\pi F \sin(2\pi F(x-ut)) e^{-2\pi^2 F^2\left({\sigma_x^x}^2 + u^2{\sigma_t^x}^2\right)} \tag{4.11}$$

$$g_t = -2\pi F u \sin(2\pi F(x-ut)) e^{-2\pi^2 F^2\left({\sigma_x^t}^2 + u^2{\sigma_t^t}^2\right)} \quad . \tag{4.12}$$

The estimated velocity $u_{\mathrm{est}} = -\frac{g_t}{g_x}$ equals the correct velocity u iff

$$e^{-2\pi^2 F^2\left({\sigma_x^x}^2 + u^2{\sigma_t^x}^2\right)} = e^{-2\pi^2 F^2\left({\sigma_x^t}^2 + u^2{\sigma_t^t}^2\right)} \quad . \tag{4.13}$$

In practice, the Gaussians in formulas (4.4-4.5) are approximated[3], as done in chapter 3, by a binomial and its discrete difference. The two Gaussians in formula (4.13) are then each approximated by the frequency response product of the smoothing binomial and the derivative binomial in the orthogonal direction. Even if $\sigma_x^x = \sigma_x^t$, the approximations of the

[3] Of course, the standard deviations have to be chosen appropriately: $({\sigma_x^x}^2 + u^2{\sigma_t^x}^2) = ({\sigma_x^t}^2 + u^2{\sigma_t^t}^2)$.

spatial derivative frequency response $e^{-2\pi^2 F^2 {\sigma_x^x}^2}$ (lhs of (4.13)) and the spatial smoother frequency response $e^{-2\pi^2 F^2 {\sigma_x^t}^2}$ (rhs of (4.13)) by the corresponding binomial filters are not equal. The same is the case for the second terms of the exponentials in both sides of (4.13). Hence, $u_{\text{est}} = -\frac{g_t}{g_x} = u \cdot e^{-F^2\left({\sigma_x^t}^2 + u^2 {\sigma_t^t}^2\right)} / e^{-F^2\left({\sigma_x^x}^2 + u^2 {\sigma_t^x}^2\right)}$ will always be u times a factor $c \neq 1$. As mentioned it is important that the binomial derivative filter shows a well approximation of the frequency response of the gaussian filter. However, as shown in figure 4.3, this is not true. Therefore the factor c will surely not be equal one. In fact, setting $u = 1$, $F = 2\pi/16$ in g and subsampling and shrinking g by four octaves leads to $u_{\text{est}} = 0.198$ instead of $u = 0.25$ (see section 4.6) which leads to a factor $c \approx 1.30$ and an error of 21%(!) of u_{est} with respect to u.

Another fact is shown by the formulas (4.11) and (4.12): Usually a derivative filter is applied only in combination with a low-pass filter in order to suppress its high-pass characteristic. However the observation that $u = u_{\text{est}} \iff \exp^{-2\pi^2 F^2\left({\sigma_x^x}^2 + u^2 {\sigma_t^x}^2\right)} = \exp^{-2\pi^2 F^2\left({\sigma_x^t}^2 + u^2 {\sigma_t^t}^2\right)}$ shows that the low-pass effects of the filters are canceled out and that the Gaussian, in fact, serves only as a good "carrier" for the derivative (see section 4.4).

Convolving a signal with the derivative of a Gaussian ideally equals, since the derivative is linear, a convolution of the differentiated signal with the Gaussian. Or, in general, convoluting a signal with a differentiated other arbitrarily chosen signal ideally equals the convolution of the differentiated signal with the arbitrarily chosen one:

$$g * \partial d = \partial\left(g * d\right) \tag{4.14}$$

The right side of eqn. (4.14) is needed but unknown, the left side is known and calculated. The eqn. 4.14 demonstrates the "carrier-effect" of d: The signal g is multiplied in frequency space with ∂d, while d is divided out later, so that the differentiated signal ∂g remains. However, the equality in (4.14) holds *iff* the frequency response of the filter ∂d, $\widehat{\partial d}$, equals the product $(j\omega) \cdot \hat{d}$, where $\hat{d}$ is the frequency response of d. Only then, the commutativity of the convolution holds:

$$\begin{array}{rrcl} & g * \partial d & = & \partial\left(g * d\right) \\ \iff & \hat{g} \cdot \widehat{\partial d} & = & \hat{g}\left[(j\omega)\hat{d}\right] \\ \iff & \widehat{\partial d} & = & (j\omega)\hat{d} \end{array}$$

This principle we may call "carrier-principle". For the binomial filters and its derivatives this "carrier-principle" does not hold, as we have seen. However, the inequality is visible only if g contains high frequencies and is therefore acceptable for low-frequency signals.

It has to be mentioned that the equality in (4.14) can never hold. Still, the error

$$E(g, d) = g * \partial d - \partial(g * d) \tag{4.15}$$

can be minimized. In this sense we have the possibilities of

1. using a discrete approximation of the continuous derivative or

2. using a discrete approximation of a continuously differentiated function and to ensure the carrier-principle. In this case we have to make sure during the construction of

the filter that the differentiated function is preserved well in the product with $(j\omega)$ so the carrier function may be divided out again, later. For example, if we use the Gaussian, the frequency responses of the Gaussian filter and the derivative filter have to be such that the frequency response of the derivative filter equals the product of $(j\omega)$ with the frequency response of the Gaussian filter.

In the next section we will discuss these two possibilities in greater detail. Also we will follow the initial idea of using a Gaussian as a carrier function for the continuous derivative and we will construct a filter that shows a good approximation of the derivative and ensures at the same time that the equality in equation (4.13) holds.

4.4 Derivative Filters

4.4.1 Background on Frequency Domain Based Filter Design

In this section we will introduce some laws that we will be using in the following. Here we show that a band-limited signal can be recovered by interpolation with a sinc-function. Further, we show that it is possible to reduce studies of periodic signals to just one period.

Consider a continuous aperiodic function $f(x)$, which is assumed to extend from $-\infty$ to ∞. Further let $f(x)$ be band-limited such that, without loss of generality, its Fourier transform $\hat{f}(\omega)$ vanishes outside the open interval $(-\pi, \pi)$. In order to obtain a sampled and discrete version $f_s(x)$ of $f(x)$, $f(x)$ has to be multiplied with a train of Dirac impulses:

$$f_s(x) = f(x) \sum_{k=-\infty}^{\infty} \delta(x-k)$$

Applying the Fourier transform (see Appendix A) defined by

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-j\omega x} dx \quad ,$$

the Fourier transform of the sampled $f_s(x)$ reads

$$\tilde{\hat{f}}_s(\omega) = \hat{f}(\omega) * \sum_k \delta(\omega - 2k\pi) \quad . \tag{4.16}$$

The signal $\tilde{\hat{f}}_s(\omega)$ is continuous and 2π periodic and each period is a replica of the original signal $\hat{f}(\omega)$. Let us now define a window function $w(\omega)$ as

$$w(\omega) = \begin{cases} 1 & -\pi \leq \omega \leq \pi \\ 0 & \text{elsewhere.} \end{cases}$$

By multiplying the signal $\tilde{\hat{f}}_s(\omega)$ with the window function $w(\omega)$, we get

$$\begin{aligned} \hat{f}_s(\omega) &= \tilde{\hat{f}}_s(\omega) \cdot w(\omega) \\ &= \left[\hat{f}(\omega) * \sum_k \delta(\omega - 2k\pi) \right] \cdot w(\omega) \quad . \end{aligned} \tag{4.17}$$

The Fourier transform of the window function is the sinc-function (Figure 4.4)

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \quad . \tag{4.18}$$

According to the convolution theorem which says that a convolution in image space cor-

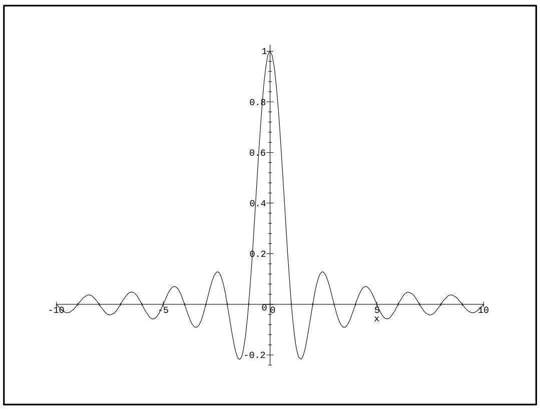


Figure 4.4: *The sinc function.*

responds to a multiplication in Fourier space, (4.17) implies that

$$\begin{aligned} f(x) &= \left[f(x) \sum_k \delta(x-k) \right] * \operatorname{sinc}(x) \\ &= \sum_{k=-\infty}^{\infty} f_s(k) \operatorname{sinc}(x-k) \quad . \end{aligned} \tag{4.19}$$

This last equation shows that it is possible to completely recover a band-limited function from appropriately spaced samples if $\hat{f}_s(\omega) = \hat{f}(\omega)$. Equation (4.19) is known as the *Whittaker–Shannon sampling theorem.*

Sampling in Fourier space a continuous function $\hat{g}(\omega)$, at positions $\frac{2\pi u}{N}$ with

$$\hat{g}_s(\omega) = \sum_{u=-\infty}^{\infty} \hat{g}(\omega)\delta\left(\omega - 2\pi\frac{u}{N}\right)$$

we obtain a discrete version $\hat{g}_s(\omega)$ of the frequency response $\hat{g}(\omega)$. According to (4.16), the corresponding spatial function $\tilde{g}_s(x)$ is continuous periodic with period N:

$$\tilde{g}(x) = g(x) * \sum_k \delta(x - kN) \quad .$$

where $g(x)$ is the inverse Fourier transform of $\hat{g}(\omega)$. If the support (or length) of the original spatial function $g(x)$ is smaller than N, then multiplying $\tilde{g}(x)$ with a window $w(x)$ that extends over the interval $[-\frac{N}{2}, \frac{N}{2}]$ allows a recovery of $g(x)$:

$$g(x) = \tilde{g}(x) \cdot w(x)$$

This means that we can reconstruct a spatial signal from the sampled Fourier corresponding if the support is smaller than inverse sampling distance in frequency space.

Suppose now that we have a discrete signal $g_s(x)$ that vanished outside the interval $[0, N-1]$. The Fourier transform reads

$$\tilde{\hat{g}}_s(\omega) = \sum_{k=0}^{N-1} g_s(x) e^{-j\omega\frac{x}{N}} \quad . \tag{4.20}$$

and is 2π periodic as already shown. Sampling again at positions $\frac{2\pi u}{N}$, we obtain

$$\widehat{\tilde{(g_s)}}_s(u) = \sum_{x=0}^{N-1} g_s(x) e^{-j2\pi\frac{ux}{N}} \quad . \tag{4.21}$$

which is called the *discrete Fourier transform* (DFT).

For the inverse DFT

$$\tilde{g}_s(x) = \sum_{u=0}^{N-1} \widehat{\tilde{(g_s)}}_s(u) e^{j2\pi\frac{ux}{N}} \quad .$$

it is sufficient to sum only over one period of length N. The Fourier series is a sum of complex exponentials with frequencies that are integer multiples of $2\pi/N$. A discrete N-periodic sequence has thus only N distinct complex exponentials. Function $\tilde{g}_s(x)$ is N-periodic.

Summarized, the information of a band-limited signal is conserved in the N samples of the DFT if the length of the signal is smaller than N. Furthermore, the fact that there exists a unique correspondence between a band-limited function with finite support and its Fourier transform that is described by the DFT, allows us to describe a continuous function $f(x)$ $(\hat{f}(\omega))$ by discrete samples of its (inverse) Fourier corresponding function $\hat{f}_s(\omega)$ $(f_s(x))$. However this ideal case of a band-limited function with finite support does not exist. Taking the Gaussian function as an example, its periodic copies will never be perfectly orthogonal, but a small dependency of the coefficients is tolerable at least if it is below the resolution of the amplitude of the coefficients. If the support of a function is larger than N, then a restriction to only N samples corresponds to a convolution of the Fourier corresponding function with a sinc. A “cutting” of the support greater N leads thus to an oscillation of the function in frequency space. This oscillation is, however, related to the amount of energy that is being cut off and is small in case of the Gaussian.

4.4.2 Differentiation of the Interpolation Template

In this section we will discuss the approach by [Karabassis and Spetsakis, 1995]. They proposed to differentiate instead of a discrete signal a continuous interpolation function of this discrete signal which we shall then resample at the points of the original lattice.

The most common way to perform the interpolation is to use the *Lagrange-Based Approach* [Stoer, 1985]: A polynomial of certain degree is to be found that fits the discrete samples of the signal. It is well known that when the degree of the polynomial goes to

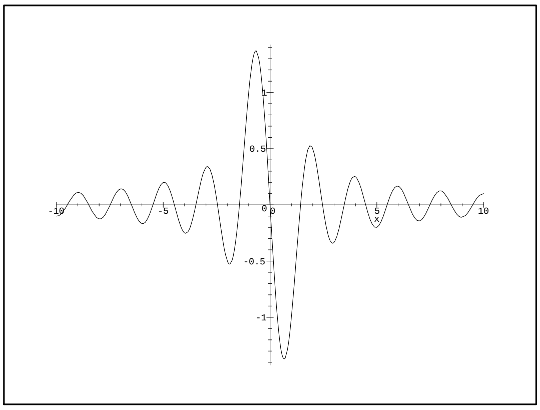


Figure 4.5: *The* $\frac{d}{dx}$ *sinc function.*

infinity, the interpolation polynom converges to the sinc-interpolation function. A 1D band-limited continuous signal $f(x)$ with $\hat{f}(\omega) = 0$ for all $\omega > \pi$, can thus be recovered from its sampled discrete version $f_s(x)$ by

$$f(x) = \sum_{k=-\infty}^{\infty} f_s(k)\, \mathrm{sinc}\,(x-k) \tag{4.22}$$

In order to differentiate $f(x)$ it is straight forward to differentiate the interpolation template $\mathrm{sinc}\,(x)$. We get

$$h(x) = \frac{d}{dx}\left(\frac{\sin(\pi x)}{(\pi x)}\right) = \frac{1}{x}\left[\cos(\pi x) - \frac{\sin(\pi x)}{(\pi x^2)}\right] \quad . \tag{4.23}$$

Sampling (4.23) at $x = k$ for $k = 0, \pm 1, \pm 2 \ldots$ leads to

$$h(k) = \begin{cases} 0 & k = 0 \\ \frac{1}{k}\left[\cos(\pi k) - \frac{\sin(\pi k)}{(\pi k^2)}\right] & \text{else} \end{cases} \quad . \tag{4.24}$$

The coefficients $h(k)$ are given by $\ldots \frac{1}{3}, -\frac{1}{2}, 1, 0, -1, \frac{1}{2}, -\frac{1}{3}, \ldots$ and the mask has infinite length. The information of the sinc-based derivative is not coded in N samples because the length of the impulse response is infinite. Taking a finite length mask corresponds to convolution in the frequency domain with a sinc. This leads to oscillation of the frequency response as shown in figure 4.6.

4.4.3 A Discrete Approach for Derivative Estimation

In this section we want to discuss an approach by [Hashimoto and Sklansky, 1987] that we mentioned already in the previous section while discussing binomial filters. [Hashimoto and Sklansky, 1987] introduce a discrete approximation of the Gaussian derivative estimation

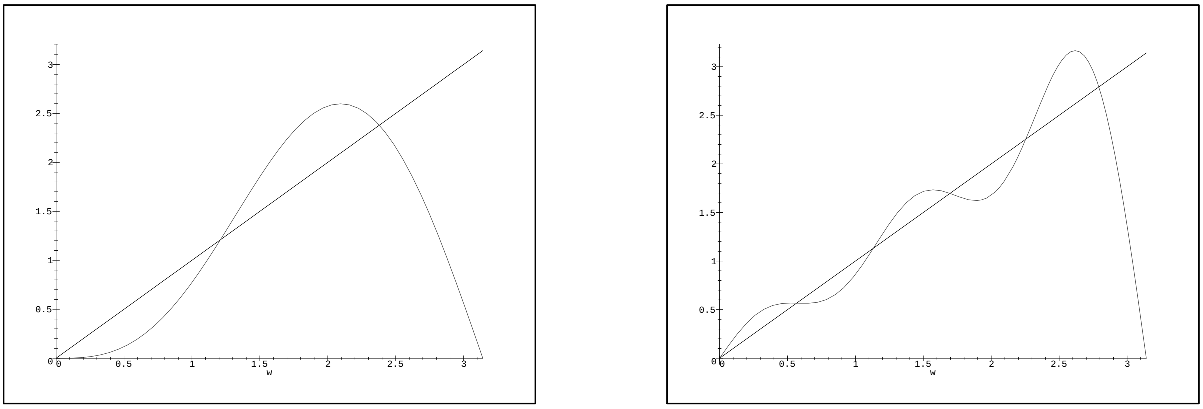


Figure 4.6: *The frequency response of* $\frac{d}{dx}$ *sinc* (x) *with mask length* $N = 5$ *(left) and* $N = 11$ *(right).*

filters by first interpreting the Gaussian filters by the Hermite polynomials, then applying this interpretation to the discrete case. The 1D Gaussian filter is given by (see appendix A)

$$D_0(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}}$$

and its Fourier transform

$$\hat{D}(\omega) = e^{-\frac{1}{2}\sigma^2\omega^2} \quad .$$

The Gaussian function is infinitely differentiable. The impulse response of the kth derivative estimation filter is

$$D_k(x) = \frac{d^k}{dx^k} D_0(x) \quad ,$$

with transfer function

$$\hat{D}_k(\omega) = (j\omega)^k \hat{D}_0(\omega) \quad .$$

These Gaussian derivative estimation filters have continuous form. To apply them to a digital image, [Hashimoto and Sklansky, 1987] approximate the continuous filters in the spatial domain. The nth Hermite polynomial $H_n(x)$ is derived by Rodrigues' formula

$$H_n(x) = \frac{(-1)^n}{\exp(-x^2)} \frac{d^n}{dx^n} \exp(-x^2) \quad .$$

This formula relates the Gaussian derivative estimation filters to the Hermite polynomials (see appendix A). Further, it has been shown that the nth Krawtchouk polynomial $K_n(x)$

$$K_n(x) = \frac{1}{(-1)^n 2^n} \frac{(N-x)!x!}{(N-n)!n!} \Delta^n \binom{N-n}{x-n} \quad .$$

approaches the nth Hermite polynomial as the number N of points increases; Δ^n is the nth forward difference operator. From the latter equation [Hashimoto and Sklansky, 1987] derive

$$D_k(x - N/2) \approx d_{N,k}(x) = \frac{1}{2^{N-k}} \sum_{i=0}^{k} (-1)^i \binom{N-k}{x-i} \binom{k}{i} \quad . \tag{4.25}$$

Formula (4.25) describes in a general form the discrete differentiation of the binomial [Jähne, 1993]. This approach is often used because of its simplicity[4] but it shows a good performance only for low frequencies. The reason here is again a cut of the spatial support which results in oscillation in frequency space. We used this operator already during our discussions in the previous chapter and we will revisit the binomial operator and its derivatives again, later.

4.4.4 Design of Derivative Filters Using Taylor Coefficients

Another technique for constructing a derivative filter is presented in [Jähne, 1993]. It is motivated by the observation that a convolution mask with odd length corresponds to a derivative filter of odd order. Given a signal $f(x)$. We are interested in a mask

$$d = \begin{pmatrix} d_0 & -d_1 & -d_2 & \dots & -d_N & \dots & d_N & \dots & d_2 & d_1 \end{pmatrix} \tag{4.26}$$

of length $L = 2N + 1$ such, that

$$f'(x) \approx \sum_{k=-N}^{N} d_k f(x + k\Delta x) \quad .$$

The values for $f(x)$ are represented by their derivatives:

$$\begin{aligned} f(x - \Delta x) &= f(x) - f'(x)\Delta x + \frac{1}{2} f''(x)(\Delta x)^2 - \frac{1}{6} f'''(x)(\Delta x)^3 + \dots \\ f(x) &= f(x) \\ f(x + \Delta x) &= f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)(\Delta x)^2 + \frac{1}{6} f'''(x)(\Delta x)^3 + \dots \quad . \end{aligned}$$

With derivatives up to the Lth order we get

$$\begin{aligned} \sum_{k=-N}^{N} d_k f(x + k\Delta x) &\overset{(\Delta x = 1)}{=} \sum_{k=-N}^{N} d_k \sum_{j=0}^{L-1} \frac{1}{j!} f^{(j)}(x) k^j \\ &= \sum_{j=0}^{L-1} f^{(j)}(x) \sum_{k=-N}^{N} \frac{1}{j!} k^j d_k \\ &= \sum_{j=0}^{L-1} f^{(j)}(x) \sum_{k=-N}^{N} p_{jk} d_k \\ &\overset{!}{=} f^{(1)}(x) \quad . \end{aligned} \tag{4.27}$$

[4] Fix point arithmetic is sufficient for a calculation of $d_{N,k}(x)$. An implementation of such a filter is, for example, easily done even on a DataCube.

This results in an equation system

$$\sum_k p_{jk} d_k = \begin{cases} 1 & j = 1 \\ 0 & \text{else} \end{cases}$$

which is to be solved for d_k.

[Jähne, 1993] applied this technique in frequency space to the function $\hat{f}(\omega) = j\omega$. The mask (4.26) has the transfer function

$$^{(N)}\hat{d} = j \sum_{u=1}^{N} d_u \sin(u\omega) \ .$$

The Taylor expansion of the sine functions is done up to the Lth order. This results in N coefficients for L powers and therefore N equations with N unknown variables d_u. The equation system is given by

$$\begin{pmatrix} 1 & 2 & 3 & \cdots & N \\ 1 & 8 & 27 & \cdots & N^3 \\ 1 & 32 & 243 & \cdots & N^5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1^{2N} & 2^{2N} & 3^{2N} & \cdots & r^{2N} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_N \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \ .$$

For $L = 2$ the mask is given as

$$\frac{1}{12}\begin{pmatrix} -1 & 8 & 0 & -8 & 1 \end{pmatrix} \ .$$

As it can be seen in figure 4.7 this filter shows a significant improvement over the discrete

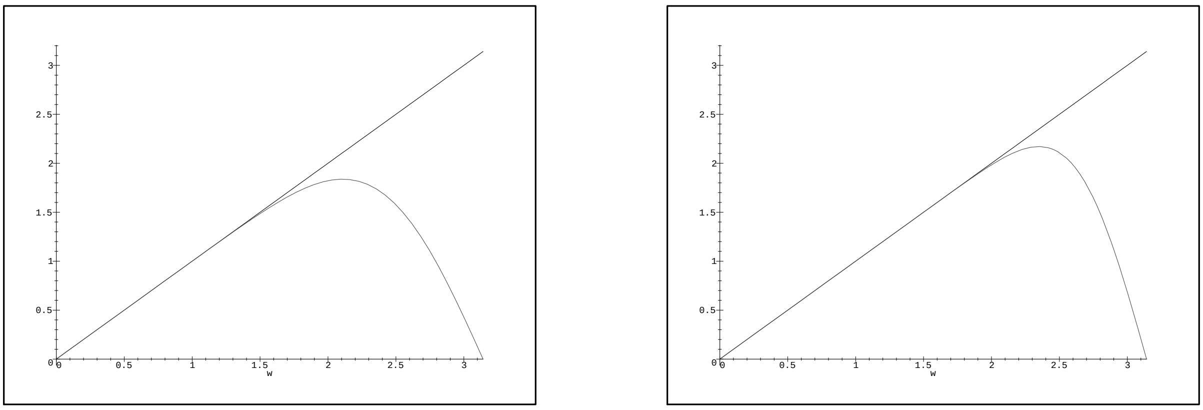


Figure 4.7: *The frequency response of the filter with mask length $N = 5$ (left) and $N = 11$ (right).*

derivative filter Δ^k.

4.5 Frequency Domain Based Filter Design

The derivative filters presented in the last sections can, unfortunately, not be applied to logarithmically subsampled data. As we have demonstrated some of the filters show a frequency response that insufficiently approximates the ideal differentiator ($j\omega$) for high frequencies. Others, as the binomial filters, do not show the desired behavior for higher frequencies. In this section we will propose a further filter design method. The goal of the frequency domain based filter design is, given a desired frequency response $\hat{h}_d(\omega)$, to find filter coefficients $h_0 \ldots h_{N-1}$ such, that the corresponding frequency response $\hat{h}(\omega)$ minimizes

$$E(\hat{h}_d, \hat{h}) = \int_{-\infty}^{\infty} \left[\hat{h}_d(\omega) - \hat{h}(\omega)\right]^2 d\omega \ . \tag{4.28}$$

If $\vec{h}$ is the vector of the filter coefficients h_k we can express $\hat{h}(\omega)$ in terms of its inverse DFT:

$$\hat{h}(\omega) = \sum_{k=0}^{N-1} h_k e^{-j\omega k} = \vec{h} \begin{pmatrix} 1 \\ e^{-j\omega} \\ \vdots \\ e^{-j\omega(N-1)} \end{pmatrix} \tag{4.29}$$

We can rewrite equation 4.28 with

$$E(\hat{h}_d, \hat{h}) = \int_{-\infty}^{\infty} \left[\hat{h}_d(\omega) - \left(\sum_{k=0}^{N-1} h_k e^{-j\omega k}\right)\right]^2 d\omega \ . \tag{4.30}$$

Since $\hat{h}(\omega)$ is 2π periodic, equation 4.30 can again be simplified by limiting the integral to the interval $[0..\pi]$:

$$E(\hat{h}_d, \hat{h}) = \int_{0}^{\pi} \left[\hat{h}_d(\omega) - \left(\sum_{k=0}^{N-1} h_k e^{-j\omega k}\right)\right]^2 d\omega \ . \tag{4.31}$$

According to equation (4.17) and (4.21) we discretizise ω within the interval $[0..\pi]$ so that equation (4.31) may be written discretely:

$$E(\vec{h}_d, \vec{h}) = \left|\vec{\hat{h}}_d - F\vec{h}\right|^2 \ , \tag{4.32}$$

where F denotes matrix whose columns are the first N Fourier basis functions:

$$F = \begin{pmatrix} W_N^{(0,0)} & \ldots & W_N^{(0,N-1)} \\ \vdots & \ddots & \vdots \\ W_N^{(N-1,0)} & \ldots & W_N^{(N-1,N-1)} \end{pmatrix} \quad \text{where } W_N^{(n,m)} = e^{-j2\pi\frac{mn}{N}}$$

Equation 4.32 says that we look for the mask coefficients h_k with a frequency response that approximates best the desired frequency response.

This filter design technique is not a regular recipe because this approximation criterion leads to adverse behavior at discontinuities of $\hat{h}_d(\omega)$: Consider as $\hat{h}_d(\omega)$ the ideal low-pass filter with a discontinuity at $\omega_c - \pi/2$. Figure 4.8 (left) shows the desired frequency response. We see the resulting frequency response obtained from samples in figure 4.8 (right). We already mentioned earlier that this abnormal behavior is due to a "cut-off"

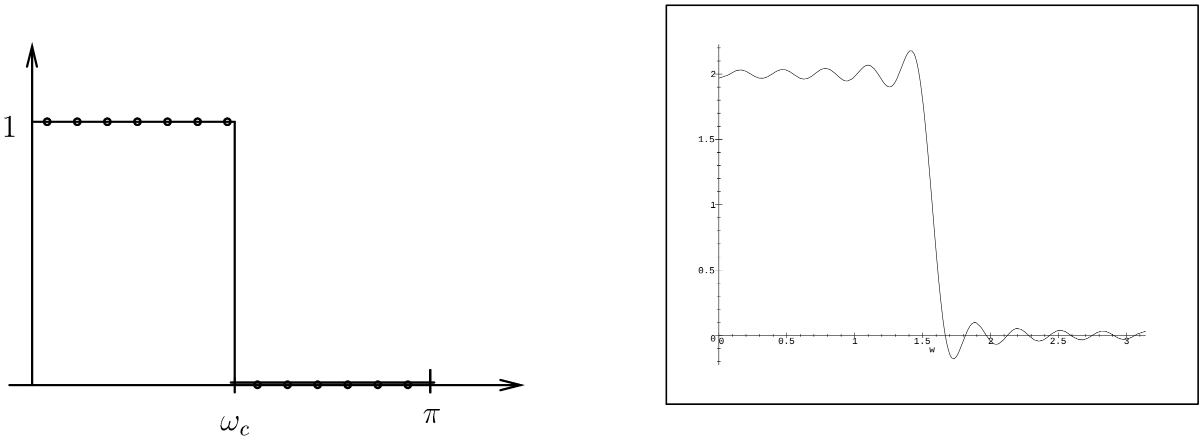


Figure 4.8: *Fixed samples of the ideal log-pass filter frequency response (left), the oscillation effects due to the discontinuity at ω_c (right).*

of the spatial signal outside the window of size N. The application of this window to the spatial signal leads to the oscillation of the frequency response. An application of this technique for the construction of a derivative filter will lead to similar results for the same reasons.

To construct a derivative filter that shows the carrier-principle we apply this frequency domain based filter design enhanced with a further and special constrain which we formulated in section 4.3. It says that the frequency response of the smoothing filter has to be included as a factor in the frequency response of the derivative filter:

$$E((j\omega)\hat{\vec{d}}_0, \hat{\vec{d}}_{1x}) = \int_{-\infty}^{\infty} \left[(\hat{j\omega})\hat{\vec{d}}_0 - \hat{\vec{d}}_{1x} \right]^2 d\omega \quad . \tag{4.33}$$

One function for d_0 that minimizes equation 4.33 is the Gaussian function $D_0(x)$. The Gaussian has over many other functions the advantage that the impulse and the frequency responses are, up to a factor, the same[5]. Choosing a window such, that the amplitude of the coefficients outside the window are small then cutting the signal outside the window equals a cut of a small amount of energy and therefore leads to a reasonably small oscillation of the frequency response. Derivating $D_0(x)$ leads to $D_{1x}(x)$ which has a similar spread as $D_0(x)$. Also $D_{1x}(x)$ can be sampled with the same number of coefficients within the same interval. The resulting frequency response will show reasonably small oscillations of both, the frequency response of the Gaussian itself as well as of the frequency response of the approximated differentiator.

[5]Furthermore the Gaussian has the advantages of being isotropic and separable.

Sampling thus the Gaussian $\hat{D}_0(\omega)$ and, respectively its derivative $\hat{D}_{1x}(\omega)$ in frequency space and transforming the sampled functions with the inverse discrete Fourier transform into image space we achieve the coefficients for our frequency domain based (FDB) smoothing and differentiation filter (For a justification see section 4.4.1):

$$\begin{aligned} \vec{d_0} &= \mathcal{F}^{-1}\left\{\sum_{i=-M}^{M}\left(e^{-\frac{1}{2}\omega^2\sigma^2}\cdot\delta\left(\omega-\frac{\pi}{M}i\right)\right)\right\} \quad \text{and} \\ \vec{d_{1x}} &= \mathcal{F}^{-1}\left\{\sum_{i=-M}^{M}\left((j\omega)e^{-\frac{1}{2}\omega^2\sigma^2}\cdot\delta\left(\omega-\frac{\pi}{M}i\right)\right)\right\} \quad . \end{aligned}$$

M is the half-size of the filter, $\mathcal{F}$ in this case the *discrete* Fourier transformation and $\delta(\cdot)$ the Dirac function.

To examine the frequency responses $\hat{\vec{d_0}}$ and $\hat{\vec{d_{x1}}}$ of $\vec{d_0}$ and $\vec{d_{1x}}$, we back transformed the FDB-filters from the spatial samples with the Fourier transform. They are shown in Figure 4.9 for $M = 4$. The filter masks are symmetric for the Gaussian (table 4.1) and the Gaussian derivative (table 4.2). We can right away examine how well the discrete samples

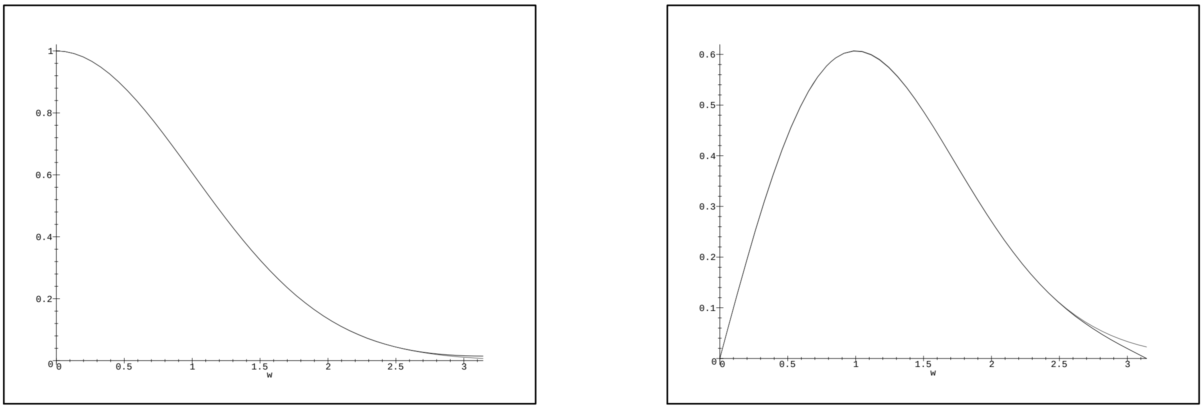


Figure 4.9: *The Gaussian and its first derivative are sampled in frequency space. The corresponding filter coefficients are obtained with the inverse DFT. The two graphs show the frequency responses of smoothing filter (left), the first derivative (right) (dotted curve). The continuous curves show the continuous functions.*

have preserved the function by dividing out either $(j\omega)$ or $\hat{g}$ (Figure 4.10). The resulting filters $\vec{d_0}$ and $\vec{d_{1x}}$ meet the requirements of section 4.3 (carrier-principle) that

1. the ideal derivative $(j\omega)$ has to be approximated sufficiently good and that

2. the frequency response of $\vec{d_0}$ is included as a factor in the frequency response of $\vec{d_{1x}}$ and, in a whole, $\hat{\vec{d_{1x}}} \approx (j\omega)\cdot\hat{\vec{d_0}}$.

Furthermore, it has

1. zero phase and

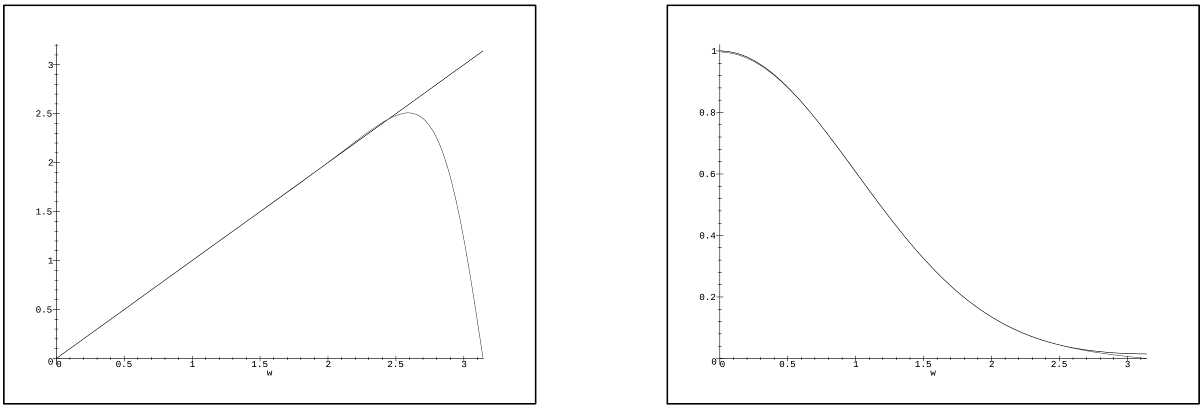


Figure 4.10: *The first derivative of a Gaussian in frequency space is the product of the Gaussian with the ideal differentiator* ($j\omega$). *The graph on the left shows the function* ($j\omega$) *(cont. curve) and its approximation (dotted curve), the graph on the right shows the FDB-Gaussian (dotted. curve) and the Gaussian that was approximated by the derivative (cont. curve) .*

coefficient	M=4	M=3	M=2
0	.3988264358	.3988264358	.3988264358
1	.2420498785	.2419978350	.2464040583
2	.05399473756	.0541507916	.1079894752
3	.0043541796	.0088124469	
4	.0003759724		

Table 4.1: *Coefficients for the FDB Gaussian filter for* $M = 4, 3, 2$.

2. it is separable which is preferable for the computation efficiency and for the filter design itself because it reduces multi dimensional convolution problems to one dimensional problems.

4.6 Experimental Comparison

In this section we present results that compare the frequency domain based filter design with the binomial filters. We first applied both filter types on different sinusoidal sequences that were subsampled by four octaves from 256×256 down to 64×64. We both predicted with Maple[6] and showed experimentally for the FDB-filter type a velocity u_{est} between 24.5 and 24.9 instead of $u = 0.25$ for nearly all frequencies up to the Nyquist frequency. Using the binomial filter type, the estimated velocity u_{est} was always below 0.20 for frequencies above 0.39 which corresponds to a period of 16 pixels in the original 256×256 image.

Next, we tested and compared the two filter types on the log-polar transforms of the synthetic sequence “Yosemite” [Barron *et al.*, 1992] and the real sequence “Marbled Block”

[6]Maple is a registered trademark of Waterloo Maple Software.

coefficient	M=4	M=3	M=2
0	0	0	0
1	$-.2422997645$	$-.2421522323$	$-.2287181028$
2	$-.1075424100$	$-.1072610375$	$-.8455351383e^{-10}$
3	$-.01358166153$	$-.2812741737e^{-10}$	
4	$-.2135657824e^{-10}$		

Table 4.2: *Coefficients for the FDB derivative filter for $M = 4, 3, 2$.*

[Otte and Nagel, 1994], both with a angular resolution of 128 samples/360 degrees and of 56 samples for the radial range [32..128] (Yosemite) and [32..256] (Marbled Block) with $\rho_0 = 32$ and $\rho_{\max} = 88$. An original image of the "Marbled Block" sequence and its log-polar transform are shown in Figure 4.11 left and middle. In Figure 4.11 (right)

Figure 4.11: *The "Marbled Block" sequence. From left to right: the original, the log-polar transform, the log-polar transformed flow field.*

we show the corresponding optical flow field. Figure 4.12 shows an original image of the Yosemite sequence, its log-polar transform and the transformed flow field. We smoothed the sequences and estimated their derivatives first with the binomial filters and estimated the optical flow. Then we used FDB-filters for the estimation of the derivatives and calculated, based on these derivatives, the optical flow. Then, we used the error-program from section 3.5 to compare the flow estimates with the known ground truth. The results are shown in table 4.3 and table 4.4. The notations are the same as in section 3.5. Subsampling effects increase with increasing distance to the fovea. In figure 4.14 and 4.13 we can see how the performance of the binomial filter (dashed line) decreases whereas the performance of the FDB-filter (continuous line) decreases comparatively less.

The averages of the relative errors in the graphs are the same as the ones given in tables 4.3 and 4.4. Notice that the local relative error for the FDB-filter is similar to the one for the binomial filters for small ρ. But for larger ρ the error is lower and much more constant for the FDB-filter that for the binomial filter: In table 4.4 for the Yosemite sequence e.g. it varies between 0.4 and 0.6 whereas the binomial filter varies between 0.4 and 0.9. for a $\rho_0 < \rho < 80$. This shows that in opposite to the binomial filter the

Figure 4.12: *The "Yosemite" sequence. From left to right: the original, the log-polar transform, the log-polar transformed flow field.*

FDB-filter is not effected as strong by the subsampling effects. Only at foveal distance $\rho > 80$ does the relative error increase significantly for the new filter. For the binomial filter this increase in error starts already at $\rho > 62$ which demonstrates once more its instability for high frequencies. [Kearney *et al.*, 1987] noted that gradient based optical flow algorithms often perform poorly on highly textured regions of an image and on noisy real sequences. The experiments reveal that this is due to the poor performance of the usual derivative filters for high frequencies. The frequency domain based filters performed noticeably better (table 4.4) on the synthetic sequence without noise in both the Cartesian and log-polar transformed cases. The performance of the FDB-filter on the real sequence was only insignificantly better (table 4.3). We think that this is due to the high frequency noise which even our filter cannot handle.

	Transformed ground truth		
Filter	av. ang. err.	av. rel. err.	density
Binomial	4.95797	28.85278	0.56144
FDB	3.93289	24.72378	0.57276

Table 4.3: *Error statistics for the log-polar transform of the "Marbled Block" sequence (see text for explanation).*

	Transformed ground truth		
Filter	av. ang. err.	av. rel. err.	density
Binomial	11.60828	66.53323	0.61582
FDB	9.50164	48.19539	0.63958

Table 4.4: *Error statistics for the log-polar transform of the "Yosemite" sequence (see text for explanation).*

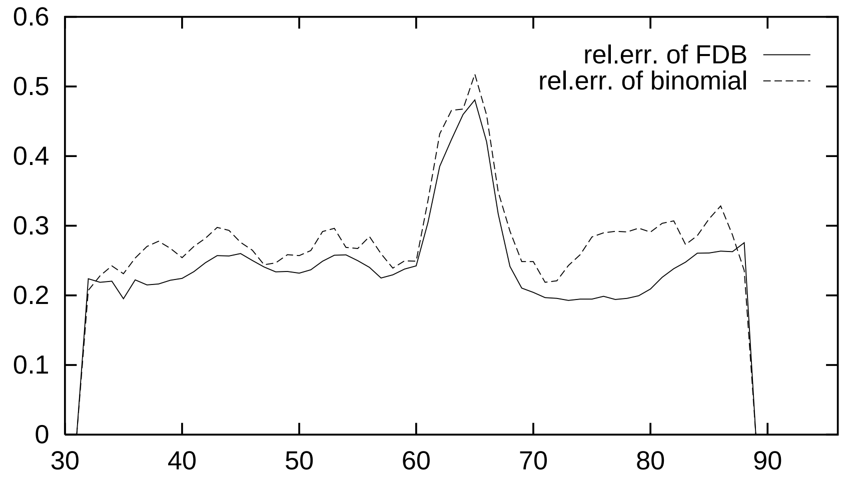


Figure 4.13: *The plot shows the rel. error as a function of the logarithm of eccentricity of the estimated flow for the binomial filters (dashed line) and for the FDB filters (continuous line) applied to the "Marbled Block" sequence.*

4.7 Summary

In this chapter we analyzed a 1-D logarithmically subsampled signal by first transforming it into the lossless signal. This intermediate step gave us then the possibility to use known Cartesian techniques for the analysis. It came out that the visual information in logarithmic subsampled images is coded with significantly higher frequencies than in the corresponding original Cartesian images. Many filters fail on high frequencies which explained the failing of gradient based optical flow techniques on these sequences. Applying a low-pass filter and thus limit the maximal frequency will result in a loss of relevant visual information. Different filters where thus needed that perform well on most or ideally on all frequencies. While dealing in this work with optical flow, we analyzed the computational steps that have to be carried out for the flow-estimation on a signal in space and time. We used the Gaussian filter and its derivative to calculate the derivatives of the signal and showed that the Gaussian has, in this context, a "carrier-function": It carries the derivative $(j\omega)$ which is, later, associated to the signal so that the product of the differentiated signal, multiplied with the Gaussian, remains. This Gaussian is, later on, canceled out. With this we reasoned that special care has to be taken while designing derivative and smoothing filters. This, we formulated in terms of the "carrier-principle". Using this criterion we proposed a new frequency domain based smoothing and gradient filter that shows in comparison with the binomial approach, which does not obey the formulated criterion, a much better performance. The latter we showed in various experiments.

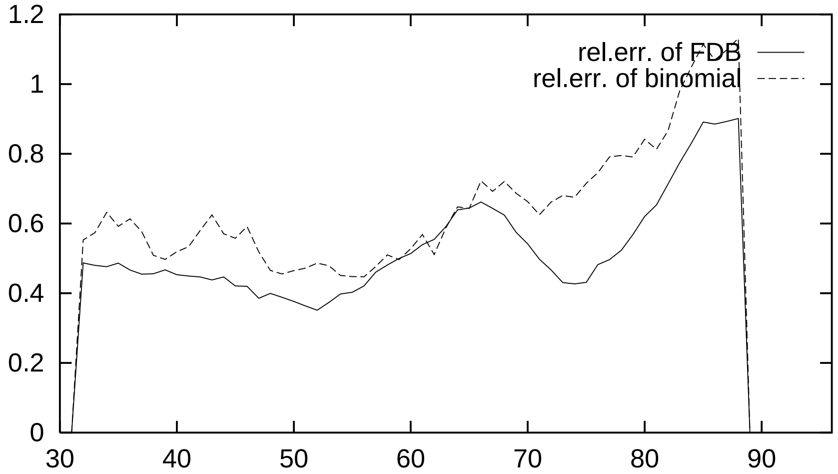


Figure 4.14: *The plot shows the rel. error as a function of the logarithm of eccentricity of the estimated flow for the binomial filters (dashed line) and for the FDB filters (continuous line) applied to the "Yosemite" sequence.*

Chapter 5

Summary and Conclusion

The paradigm of *active vision* can not be imagined without the complex logarithmic mapping as a space-variant resolution scheme because it provides both a very efficient coding and a useful representation for motion tasks. To compute the optical flow as an intermediate step for solving such tasks we have to overcome the problems of the polar deformation and the irregular subsampling of the image. As we are trying to exploit this flow for 3D-motion estimation we are interested in *what* amount of motion information is still preserved after the log-polar transform. In this thesis we studied the effects of the complex logarithmic mapping on the estimation of optical flow and showed that the problems for the optical flow estimation, that are introduced by the log-polar transform, are by no means trivial.

Namely, two problems are introduced by the complex logarithmic mapping: The deformation and the logarithmic subsampling. After having given an extensive introduction to the complex logarithmic mapping, we showed in the third chapter that the polar deformation introduces fictitious gray-value curvature that lead to an erroneous elimination of the *aperture problem*. Starting from this analysis, we introduced two new differential flow techniques that are superior to the conventional differential techniques as they reflect the harmonic variation of the optical flow in polar transformed images.

In the fourth chapter we showed that the complex logarithmic mapping subsamples the visual data immensely, however, not evenly for all image pixels so that an application of linear shift invariant analysis techniques is not possible. We therefore introduced the lossless signal as a basis for an analysis of the logarithmic subsampling. Its introduction allowed the application of the traditional spectral analysis techniques. Then we analyzed the logarithmic subsampling effects on the image data and showed that this type of data contains unusual high frequencies on which common binomial approximations of the Gaussian derivatives fail. A possibility to ensure a good performance of binomial derivative filters is to band-limit the visual data. However, while needing the visual information in the periphery in tasks like motion detection, calculation of ego- or eco-motion, estimation of FOE etc., special care has to be taken for the higher frequencies. Therefore, we designed new smoothing and derivative filters with which our flow techniques from the third chapter performed significantly better.

Even though we have shown in this thesis that the problems for the estimation of optical flow, that are introduced by the complex logarithmic mapping, can be solved, it may not be forgotten, that the logarithmic subsampled visual data suffers from a loss of significant visual information. In fact, it is not clear, if the visual tasks of active vision systems, that shall rely on the optical flow, can successfully work with it. The question has to be raised on *how* the environment has to be sensed, *how* the camera system has to be moved and *how* the gaze has to be controlled in order to really exploit the flow information for the vision tasks.

Appendix A

Definitions and Conventions

Different sign conventions and constants are often a source of troubles. In this report we use the following definitions and conventions.

A.1 Definition of Gaussian

The Gaussian in its general form is given by

$$D_0^x(x) = N(a)e^{-ax^2} \tag{A.1}$$

where $N(a)$ is the normalization constant that only depends on the coefficient a in the exponent. We us $a = (2\sigma^2)^{-1}$ where σ is the standard deviation. For $N(a)$ we use $N = (\sqrt{2\pi}\sigma)^{-1}$.

The first two derivations of Gaussians are:

$$D_0^x(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma}} \tag{A.2}$$

$$D_{1x}^x(x) = -\frac{x}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2}{2\sigma}} \tag{A.3}$$

$$D_{2x}^x(x) = \frac{x^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{x^2}{2\sigma}} \tag{A.4}$$

$$D_{nx}^x(x) = \frac{d^n}{dx^n} D_0 = (-1)^n \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{2\sigma^2}^{\frac{n}{2}} H_n(\sqrt{\frac{1}{2\sigma^2}}x) e^{-\frac{x^2}{2\sigma^2}} \quad .$$

where H_n are the Hermite polynomials

$$H_n(x) = \frac{(-1)^n}{\exp(-x^2)} \frac{d^n}{dx^n} \exp(-x^2) \quad .$$

A.2 Fourier Transform

We use the following convention for the Fourier transform and its inverse and it is the same as in [Bronstein and Semendjajew, 1985] and Maple:

$$\mathcal{F}\{f\} = \hat{f}(\omega) = \int_{-\infty}^{\infty} f(x)e^{-j\omega x}dx \tag{A.5}$$

$$\mathcal{F}^{-1}\left\{\hat{f}\right\} = f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(x)e^{j\omega x}d\omega \ . \tag{A.6}$$

where $\hat{f}$ denotes the Fourier transform of function f and $j = \sqrt{-1}$. The symbol $\mathcal{F}$ is used for both the discrete and the continuous Fourier transform.

To derive the Fourier transform of the derivatives of Gaussian we use the derivative theorem:

$$\int_{-\infty}^{\infty} \frac{d^n f(x)}{dx^n} e^{-j\omega x}dx = (j\omega)^n \int_{-\infty}^{\infty} f(x)e^{-j\omega x}dx \ . \tag{A.7}$$

With this and (A.2)-(A.4) we get the following Fourier correspondencies:

Function	Fourier Transform
$\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{x^2}{2\sigma}}$	$e^{-\frac{1}{2}\omega\sigma^2}$
$-\frac{x}{\sqrt{2\pi\sigma^3}}e^{-\frac{x^2}{2\sigma}}$	$(j\omega)\, e^{-\frac{1}{2}\omega^2\sigma^2}$
$\frac{x^2}{\sqrt{2\pi\sigma^5}}e^{-\frac{x^2}{2\sigma}}$	$(j\omega)^2\, e^{-\frac{1}{2}\omega^2\sigma^2}$.

(A.8)

The Fourier transform of D_0^x is from [Bronstein and Semendjajew, 1985].

Appendix B

Proofs for Section 4.2.1

Lemma 1. *Given the signal g as defined above. Then the local logarithmic subsampling distance at radius ρ is*

$$\delta\rho = \rho(\sqrt{a} - \frac{1}{\sqrt{a}}) \ .$$

Proof. Each ρ is given by $\rho = \rho_0 a^{\xi}$ for some $\xi \in [0 \ldots N]$ (We can generalize ξ to be continuous). Thus $\delta\rho = \rho_0 a^{\xi+\frac{1}{2}} - \rho_0 a^{\xi-\frac{1}{2}} = \rho_0 a^{\xi}\left(\sqrt{a} - \frac{1}{\sqrt{a}}\right) = \rho\left(\sqrt{a} - \frac{1}{\sqrt{a}}\right)$.

Lemma 2. *The sampling distance of $\lambda(\hat{\xi})$ is $\frac{\ln a}{\ln b} = \ln_b a$.*

Proof. $\delta\hat{\xi} = \ln_b(\rho_{max}\sqrt{a}) - \ln_b(\rho_{max}\frac{1}{\sqrt{a}}) = \ln_b\left(\frac{\rho_{max}\sqrt{a}}{\rho_{max}\frac{1}{\sqrt{a}}}\right) = \ln_b(a) = \frac{\ln a}{\ln b}$.

Lemma 3. *The sampling distance of the intermediate function λ and the most peripheral pixel of signal g is the same.*

Proof. It is $\rho_{max} = \rho_0 b^M$. Since the Sampling-distance of λ is $\delta\hat{\xi} = \ln_b a$ (Lemma 2), we get $\rho_0 b^{M-1} = \frac{\rho_0 b^M}{b} = \frac{\rho_{max}}{b} = \rho_{max} - 1$.

Appendix C

Proofs for Section 3.3

Lemma 4. *Let $E(\rho,\eta)$ denote the polar mapped image of the Cartesian image $I(x,y)$. Then the following equation holds:*

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} \\ E_{\rho\eta} & E_{\eta\eta} \end{pmatrix} = \frac{\partial(x,y)}{\partial(\rho,\eta)}^T \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \frac{\partial(x,y)}{\partial(\rho,\eta)} + I_x \begin{pmatrix} \frac{\partial^2 x}{\partial\rho^2} & \frac{\partial^2 x}{\partial\rho\partial\eta} \\ \frac{\partial^2 x}{\partial\rho\partial\eta} & \frac{\partial^2 x}{\partial\eta^2} \end{pmatrix} + I_y \begin{pmatrix} \frac{\partial^2 y}{\partial\rho^2} & \frac{\partial^2 y}{\partial\rho\partial\eta} \\ \frac{\partial^2 y}{\partial\rho\partial\eta} & \frac{\partial^2 y}{\partial\eta^2} \end{pmatrix}$$

with

$$\frac{\partial(x,y)}{\partial(\rho,\eta)} = \begin{pmatrix} \frac{\partial x}{\partial\rho} & \frac{\partial x}{\partial\eta} \\ \frac{\partial y}{\partial\rho} & \frac{\partial y}{\partial\eta} \end{pmatrix} \quad .$$

Proof. From

$$E(\rho,\eta) = I(\rho\cos\eta, \rho\sin\eta)$$

we obtain

$$\begin{aligned} \frac{\partial}{\partial\rho}E(\rho,\eta) &= \frac{\partial x}{\partial\rho}\frac{\partial}{\partial x}I(x,y) + \frac{\partial y}{\partial\rho}\frac{\partial}{\partial y}I(x,y) \\ &= \frac{\partial x}{\partial\rho}I_x(x,y) + \frac{\partial y}{\partial\rho}I_y(x,y) \quad . \end{aligned}$$

The subscripts denote the partial derivatives. Further holds

$$\frac{\partial}{\partial\rho}\left[\frac{\partial x}{\partial\rho}I_x(x,y)\right] = \frac{\partial x^2}{\partial\rho^2}I_x(x,y) + \frac{\partial x}{\partial\rho}\left[\frac{\partial x}{\partial\rho}I_{xx}(x,y) + \frac{\partial y}{\partial\rho}I_{xy}(x,y)\right]$$

and

$$\frac{\partial}{\partial\eta}\left[\frac{\partial y}{\partial\rho}I_y\right] = \frac{\partial y^2}{\partial\rho^2}I_y + \frac{\partial y}{\partial\rho}\left[\frac{\partial x}{\partial\rho}I_{yx} + \frac{\partial y}{\partial\rho}I_{yy}\right] \quad .$$

All together we obtain

$$E_{\rho\rho} = \begin{pmatrix} \frac{\partial x}{\partial\rho} \\ \frac{\partial y}{\partial\rho} \end{pmatrix} \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \begin{pmatrix} \frac{\partial x}{\partial\rho} \\ \frac{\partial y}{\partial\rho} \end{pmatrix} + \frac{\partial x^2}{\partial\rho^2}I_x + \frac{\partial y^2}{\partial\rho^2}I_y \quad .$$

Repeating this for all component and for the derivatives in $\eta\rho$ and η^2 we obtain the lemma.

Lemma 5. *The coefficient matrix of the linear system after differentation of the BCCE with respect to (ρ, η) and the assuption of spatial constancy of (u, v) reads*

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} - \frac{E_\eta}{\rho} \\ E_{\rho\eta} - \frac{E_\eta}{\rho} & E_{\eta\eta} + \rho E_\rho \end{pmatrix} .$$

Proof. It holds

$$E_\rho u^p + E_\eta v^p = -E_t .$$

Differentiating for ρ leads

$$E_{\rho\rho} u^p + E_\rho \frac{\partial u^p}{\partial \rho} + E_{\eta\rho} v^p + E_\eta \frac{\partial v^p}{\partial \rho} = -E_{t\rho} .$$

For $\frac{\partial u^p}{\partial \rho}$ and $\frac{\partial v^p}{\partial \rho}$ holds

$$\begin{aligned} \frac{\partial u^p}{\partial \rho} &= \underbrace{\frac{\partial u}{\partial \rho}}_{=0} \cos(\eta) + \underbrace{\frac{\partial v}{\partial \rho}}_{=0} \sin(\eta) = 0 \\ \frac{\partial v^p}{\partial \rho} &= \underbrace{-\frac{1}{\rho^2} \left(-u \sin(\eta) + v \cos(\eta)\right)}_{-\frac{v^p}{\rho}} + \frac{1}{\rho} \left(-\frac{\partial u}{\partial \rho} \sin(\eta) + \frac{\partial v}{\partial \rho} \cos(\eta) \right) . \end{aligned}$$

Therefore

$$\begin{aligned} E_{\rho\rho} u^p + E_{\eta\rho} v^p - E_\eta \frac{v^p}{\rho} &= -E_{t\rho} \\ \begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} - \frac{E_\eta}{\rho} \end{pmatrix} \begin{pmatrix} u^p \\ v^p \end{pmatrix} &= -E_{t\rho} . \end{aligned}$$

The proof for η is equivalent.

Lemma 6. *Let $E(\rho, \eta)$ denote the polar mapped image of the Cartesian image $I(x, y)$. Then the following equation holds:*

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} - \frac{E_\eta}{\rho} \\ E_{\rho\eta} - \frac{E_\eta}{\rho} & E_{\eta\eta} + \rho E_\rho \end{pmatrix} = \frac{\partial(x, y)}{\partial(\rho, \eta)}^T \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \frac{\partial(x, y)}{\partial(\rho, \eta)}$$

Proof. Rewriting the coefficient matrix as

$$\begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} - \frac{E_\eta}{\rho} \\ E_{\rho\eta} - \frac{E_\eta}{\rho} & E_{\eta\eta} + \rho E_\rho \end{pmatrix} = \begin{pmatrix} E_{\rho\rho} & E_{\eta\rho} \\ E_{\rho\eta} & E_{\eta\eta} \end{pmatrix} + \begin{pmatrix} 0 & -\frac{E_\eta}{\rho} \\ -\frac{E_\eta}{\rho} & \rho E_\rho \end{pmatrix}$$

and applying (Lemma 4) leads to the lemma.

Appendix D

The Simulation Software

D.1 The "vcrtsens" Program

The program "vcrtsens" is used to define the log-polar grid. It takes ρ_0 (-r_min), N_r (-necc), N_a (-nang) and the image dimension (-dimim) as input. The output of the program (-o) is a look-up-table which contains the sensor definitions and which is used by the programs "vcort" and "vremap". The fovea is defined as a polar grid with $\delta\rho = 1$. Defining $\rho_0 = \rho_m ax$ allows to position the entire image within the fovea and to derive therefore the polar transform of the image. This trick was used for the first experiment in section 3.5.

D.2 The "vcort" Program

The program "vcort" is used to map a Cartesian image into its log-polar mapped image. The input (-i) is a Cartesian image and the sensor date (-s), which is given by the output of "vcrtsens". Further input is the position (-x, -y) of the fovea of the input sensor. The origin of the input image is assumed in the top left corner. (x,y)==(-1,-1) positions the sensor automatically in the center of the image.

D.3 The "vremap" Program

The program "vremap" performs a re-mapping from a log-polar mapped image into a Cartesian image. Its input (-i) is a log-polar mapped image and and the sensor date (-s), which is given by the output of "vcrtsens". The output (-o) is a Cartesian image.

D.4 The Optical Flow Program

Even though we used four different optical flow methods (LCC, LAC, LCT, LAT), we show only the sources of the LAC method here. All others programs can be derived from this source.

The program "LAC" takes as input the spatio-temporal derivatives (-i1, -i2, -i3) of an image sequence. Further, the retinal sensor is defined by ρ_0 (-fovea), N_r (-peri), N_a (-ang), the resolution of the input image (-res), a logical variable (-logical) which is set to true, if the flow should also be estimated in the fovea, the half-size of the neighborhood window (-m) and the threshold for the lowest singular value (-f1). The output (-o) is a multiband image: image three is boolian and contains a one at each position, where the flow was estimable, a zero elsewhere. Image one and two contain the values for u and v, respectively.

Appendix E

Maple Simulation for the Frequncy Based Filters

```
readlib(fourier);

M:=2;

#####################################################################

gauss := 1/2**(1/2)/Pi**(1/2)/s*exp(-1/2*x**2/s**2);
fougauss_cont := exp(-w^2*s^2/2);

gauss_discr := sum(gauss * Dirac(x-i),i=-M..M);
fougauss_discr := fourier(gauss_discr,x,w);

binom_coeff := x -> subs(N=s^2*4,1/2^N*binomial(N,x));
gauss_binom := sum(binom_coeff(iter+M)*Dirac(x-iter),iter=-M..M);

fougauss_binom := fourier(gauss_binom,x,w);

#####################################################################

gauss1 := -1/2**(1/2)/Pi**(1/2)/s**3*x*exp(-1/2*x**2/s**2);
fougauss1_cont := I*w*exp(-w^2*s^2/2);

gauss1_discr := sum(gauss1 * Dirac(x-i),i=-M..M);
fougauss1_discr := fourier(gauss1_discr,x,w);

hash_coeff := x -> subs(N=2*M,1/2^(N-1) * sum((-1)^i *
                   binomial(N-1,x-i) * binomial(1,i),i=0..1));
gauss1_binom := sum(hash_coeff(iter+M)*Dirac(x-iter),iter=-M..M);
```

```
fougauss1_binom := fourier(gauss1_binom,x,w);

####################################################################
Chapter 4.3 The Effects on Spatio-temporal Filtering
####################################################################

plot({w,evalc(Im(subs(s=sqrt(M/2),fougauss1_binom)))/
        evalc(Re(subs(s=sqrt(M/2),fougauss_binom)))},w=0..1.8);

plot({evalc(Re(subs(s=sqrt(M/2),fougauss_cont))),
        evalc(Im(subs(s=sqrt(M/2),fougauss1_binom)))/w},w=0..Pi);

####################################################################
Chapter 4.4.2 Differentation of the Interpolation Template
####################################################################
M:=5;

sinc := sin(x*Pi)/(x*Pi);

dsinc := diff(sinc,x);

dsinc_s := sum(dsinc*Dirac(x-i),i=-M..M);

#fdsinc_s := fourier(dsinc_s, x, w);

fdsinc_s_2 :=- 1/2*exp(2*I*w) + exp(I*w)
               - exp(- I*w) + 1/2*exp(- 2*I*w);

fdsinc_s_5 := 1/5*exp(5*I*w) - 1/4*exp(4*I*w) + 1/3*exp(3*I*w)
              - 1/2*exp(2*I*w)+ exp(I*w) - exp(- I*w) + 1/2
              * exp(- 2*I*w)- 1/3 *exp(- 3*I*w) +
              1/4*exp(- 4*I*w) - 1/5*exp(- 5*I*w);

plot({w,Im(evalc(fdsinc_s_2))}, w=0..Pi);
plot({w,Im(evalc(fdsinc_s_5))}, w=0..Pi);

M := 5;
readlib(fourier);

####################################################################
Chapter 4.4.4 Design of Derivative Filters using Taylor Coefficients
####################################################################
```

```
A := array(1..M,1..M,[
seq([seq(j^(2*i-1),j=1..M)],i=1..M)]);


b := array(1..M,[1,seq(0,i=1..M-1)]);

half_mask := evalm(1/2 *inverse(A) &* b);
jaehne := sum(-half_mask[l]*Dirac(x-l),l=1..M)
+ sum(+half_mask[l]*Dirac(x+l),l=1..M);

jaehne_trans := fourier(jaehne,x,w);
plot({w,evalc(Im(jaehne_trans))},w=0..Pi);

####################################################################
Chapter 4.4.5 Frequency Domain Based Derivative Filters
####################################################################
M:=4;

fougauss_discr := Pi/M * sum(fougauss_cont *
                        Dirac(w-i*Pi/M),i=-M..M);
gauss_discr := (invfourier(fougauss_discr,w,x));
re_gauss_discr := sum(gauss_discr * Dirac(x-i),i=-M..M);
re_fougauss_discr := fourier(re_gauss_discr,x,w);

fougauss1_discr := Pi/M * sum(fougauss1_cont *
                         Dirac(w-i*Pi/M),i=-M..M);
gauss1_discr := invfourier(fougauss1_discr,w,x);
re_gauss1_discr := sum(gauss1_discr * Dirac(x-i),i=-M..M);
re_fougauss1_discr := fourier(re_gauss1_discr,x,w);


plot({evalc(Re(subs(s=1,fougauss_cont))),
      evalc(Re(subs(s=1,re_fougauss_discr)))},w=0..Pi);

plot({evalc(Im(subs(s=1,fougauss1_cont))),
      evalc(Im(subs(s=1,re_fougauss1_discr)))},w=0..Pi);

plot({w,evalc(Im(subs(s=1,re_fougauss1_discr)))/
        evalc(Re(subs(s=1,re_fougauss_discr)))},w=0..Pi);

plot({evalc(Re(subs(s=1,fougauss_cont))),
        evalc(Im(subs(s=1,re_fougauss1_discr)))/w},w=0..Pi);
```

```
######################################################################
Chapter 4.5
######################################################################
M:=2;

GA_t := evalf(subs(w=2*Pi/16,s=sqrt(M/2),
                             evalc(Im(fougauss1_cont))));
GA_x := evalf(subs(w=2*Pi/4,s=sqrt(M/2),
                            evalc(Im(fougauss1_cont))));
G_t := evalf(subs(w=2*Pi/16,s=sqrt(M/2),
                            evalc(Re(fougauss_cont))));
G_x := evalf(subs(w=2*Pi/4,s=sqrt(M/2),
                           evalc(Re(fougauss_cont))));
evalf(GA_t * G_x / (GA_x * G_t));

GA_t := evalf(subs(w=2*Pi/16,s=1,
                             evalc(Im(re_fougauss1_discr))));
GA_x := evalf(subs(w=2*Pi/4,s=1,
                            evalc(Im(re_fougauss1_discr))));
G_t := evalf(subs(w=2*Pi/16,s=1,
                            evalc(Re(re_fougauss_discr))));
G_x := evalf(subs(w=2*Pi/4,s=1,
                           evalc(Re(re_fougauss_discr))));
evalf(GA_t * G_x / (GA_x * G_t));

GA_t := evalf(subs(w=2*Pi/16,s=sqrt(M/2),
                             evalc(Im(fougauss1_binom))));
GA_x := evalf(subs(w=2*Pi/4,s=sqrt(M/2),
                            evalc(Im(fougauss1_binom))));
G_t := evalf(subs(w=2*Pi/16,s=sqrt(M/2),
                            evalc(Re(fougauss_binom))));
G_x := evalf(subs(w=2*Pi/4,s=sqrt(M/2),
                           evalc(Re(fougauss_binom))));
evalf(GA_t * G_x / (GA_x * G_t));
```

Bibliography

[Barron *et al.*, 1992] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 236–242, CVPR92, 1992.

[Bolduc and Levine, 1994] M. Bolduc and M.D. Levine. A foveated retina system for robotic vision. In *ECCV-94 Workshop on Natural and Artificial Visual Sensors*, 1994.

[Bronstein and Semendjajew, 1985] I.N. Bronstein and K.A. Semendjajew. *Taschenbuch der Mathematik*. Harri Deutsch, Thun, 1985.

[Burt, 1984] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution image processing and analysis*, volume 12 of *Springer Series in Information Sciences*. Springer, Berlin, 1984.

[Campani and Verri, 1992] M. Campani and A. Verri. Motion analysis from first-order properties of optical flow. *CVGIP: Image Understanding*, 56:90–107, 1992. Special Issue on Purposive, Qualitative, Active Viison, Y. Aloimonos (ed.).

[Daniilidis, 1994] K. Daniilidis. Attentive visual motion processing: computations in the log-polar plane. Accepted in "Computing, Archives for Informatic and Numerical Computation", 1994.

[Ferrari *et al.*, 1995] F. Ferrari, J. Nielsen, P. Questa, and G.Sandini. Space variant imaging. *Sensor Review*, 15(2):17–20, 1995.

[Gonzalez and Woods, 1992] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, 1992.

[Hashimoto and Sklansky, 1987] M. Hashimoto and J. Sklansky. Multiple-Order Derivatives for Detecting Local Image Characteristics. *Computer Vision, Graphics, and Image Processing*, 39:28–55, 1987.

[Hawken and Parker, 1994] M.J. Hawken and A.J. Parker. Spatial Receptive Field Organization in Monkey V1 and its Relationship to the Cone Mosaic. In *Models of Neural Functions*. Landy and Monshon, 1994.

[Horridge, 1986] G.A. Horridge. A theory of insect vision: velocity parallax. *Proceedings of the Royal Society of London, B*, 229:13–27, 1986.

[Hubel and Wiesel, 1962] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiology*, 160:106–154, 1962.

[Hubel, 1989] D.H. Hubel. *Auge und Gehirn: Neurobiologie des Sehens.* Spek-Wis-Vg, Heidelberg, 1989.

[Jähne, 1993] B. Jähne. *Digital Image Prcessing.* Springer Verlag, Berlin et al., 1993.

[Karabassis and Spetsakis, 1995] E. Karabassis and M. Spetsakis. An Analysis of Image Interpolation, Differentation, and Reduction Using Local Polynomial Fits. *CVGIP: Graphical Models and Image Processing*, 57(3):183–196, May 1995.

[Kearney *et al.*, 1987] J.K. Kearney, W.B. Thompson, and D.L. Boley. Optical flow estimation: An error analysis of gradient-based methods with local optimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9:229–244, 1987.

[Kreider *et al.*, 1990] G. Kreider, J. Van der Spiegel, I. Born, C. Claeys, I. Debusschere, G. Sandini, P. Dario, and F. Fantini. A Retina Like Space Variant CCD Sensor. In *Proc. SPIE Conference on Charged-Coupled Devices and Solid State Optical Sensors*, Santa Clara, 1990.

[Kronauer and Zeevi, 1985] R. E. Kronauer and Y. Y. Zeevi. Reorganization and Diversification of Signals in Vision. *IEEE Trans. Systems, Man, and Cybernetics*, 15(1):91–101, February 1985.

[Lindeberg and Florack, 1994] T. Lindeberg and L. Florack. Foveal scale-space and the linear increase of receptive field size as a function of eccentricity. private correspondency, 1994.

[Lucas and Kanade, 1981] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understandig Workshop*, pages 121–130, 1981.

[Meer *et al.*, 1987] P. Meer, E. Baugher, and A. Rosenfeld. Frequency Domain Analysis and Synthesis of Image Pyramid Generating Kernels. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(4):512–522, July 1987.

[Otte and Nagel, 1994] M. Otte and H.-H. Nagel. Optical flow estimation: advances and comparisons. In *Proc. Third European Conference on Computer Vision*, pages 51–60. Stockholm, Sweden, May 2-6, 1994.

[Pahlavan *et al.*, 1993] K. Pahlavan, T. Uhlin, and J.-O. Eklundh. Active Vision as a Methodology. In Yannis Aloimonos, editor, *Active Perception*, pages 19–46. Lawrence Erlbaum, Hillsdale, 1993.

[Poggio and Reichardt, 1973] T. Poggio and W. Reichardt. Theory of the Pattern Induced Flight Orientation of the Fly *Musca Domestica*. *Kybernetik*, 12(4):185–203, 1973.

[Porat and Zeevi, 1988] M. Porat and Y. Zeevi. The Generalized Garbor Scheme of Image Representation in Biological and Machine Vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(4):452–468, July 1988.

[Press *et al.*, 1986] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes, The Art of Scientific Computing.* Cambridge University Press, Cambridge, UK, 1986.

[Schwartz, 1977] E.L. Schwartz. Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevance to Perception. *Biological Cybernetics*, 25:181–194, 1977.

[Simoncelli, 1994] E.P. Simoncelli. Design of Multi-Dimensional Derivative Filters. IEEE Intern. Conference on Image Processing, Austin TX, Nov 1994.

[Sommer, 1995] G. Sommer. Verhaltensbasierter Entwurf technischer visueller Systeme. *Künstliche Intelligenz*, 3(5):42–45, 1995.

[Spillmann and Werner, 1990] L. Spillmann and J. S. Werner, editors. *Visual Perception.* Academic Press, INC, 1990.

[Stoer, 1985] J. Stoer. *Numerische Mathematik, Bd. 1.* Springer, 1985.

[Tistarelli and Sandini, 1992] M. Tistarelli and G. Sandini. Dynamic Aspects in Active Vision. *CVGIP Image Understanding*, 56(1):108–129, July 1992.

[Tistarelli and Sandini, 1993] M. Tistarelli and G. Sandini. On the Advantage of Polar and Log-Polar Mapping for Direct Estimation of Time-to-Impact from Optical Flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):401–410, April 1993.

[Tretiak and Pastor, 1984] O. Tretiak and L. Pastor. Velocity Estimation From Image Sequences with Second Order Differential Operators. In *Proc. Int. Conf. on Pattern Recognition*, pages 16–19, 1984.

[Tunley and Young, 1994] H. Tunley and D. Young. First Order Optic Flow from Log-Polar Sampled Images. *Lecture Notes in Computer Science*, 800, 1994.

[van Doorn *et al.*, 1972] A.J. van Doorn, J.J. Koenderink, and M.A. Bouman. The Influence of the Retinal Imhomogeneity on the Perception of Spatial Patterns. *Kybernetik*, 10(4):223–230, 1972.

[Vieville and Faugeras, 1994] T. Vieville and O. Faugeras. Robust and fast computation of edge characteristics in image sequences. *International Journal of Computer Vision*, 13:153–179, 1994.

[Wässle *et al.*, 1990] H. Wässle, U. Grünert, J. Röhrenbeck, and B. Boycott. Retinal Ganglion Cell Density and Cortical Magnification Factor in the Primate. *Vision Research*, 11:1879–1911, 1990.

[Weiman and Chaikin, 1979] C.F.R. Weiman and G. Chaikin. Logarithmic spiral grids for image processing and display. *Computer Graphics and Image Processing*, 11:197–226, 1979.

[Weiman, 1994] C. Weiman. Log-polar Binocular Vision System – final report. Technical report, NASA Phase II SBRI, December 1994.

Erklärung:

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.
Kiel, den