Aus dem Institut für Neuro- und Bioinformatik
der Universität zu Lübeck
Direktor: Univ.-Prof. Dr. rer. nat. Thomas Martinetz

Computational models and systems for gaze guidance

Inauguraldissertation
zur Erlangung der Doktorwürde
der Universität zu Lübeck
– Aus der Technisch-Naturwissenschaftlichen Fakultät –

vorgelegt von
Michael Dorr
aus Hamburg
Lübeck 2009

Erster Berichterstatter:	Prof. Dr.-Ing. Erhardt Barth
Zweiter Berichterstatter:	Prof. Dr. rer. nat. Heiko Neumann
Tag der mündlichen Prüfung:	23. April 2010

Zum Druck genehmigt: Lübeck, den 26. April 2010

Contents

Acknowledgements

Many friends and colleagues have contributed to the work presented in this thesis, both materially and by giving moral support, and expressing my gratitude here certainly does little to pay back my debts. After all these years, I can proudly say that I spent more than a third of my life under Erhardt Barth's supervision, and he has been a "Doktorvater" in the finest sense – thank you! At the Institute of Neuro- and Bioinformatics, Thomas Martinetz gave me the opportunity to pursue my PhD, and I am grateful for that. Martin Böhme taught me most of what I know about programming, and if it only was because I did not want to be humiliated anymore by his scathing yet correct reviews of my commits. Martin Haker, Sascha Klement, and Fabian Timm were the right persons to go to when weird computer problems had to be solved, metal needed to be bent, or I simply felt like recharging my scientific creativity at the foosball table – thanks, guys. Cicero Mota and Laura Pomarjanschi were collaborators on such diverse topics as the perception of multiple motions and playing computer games with gaze. Finally, to acknowledge Eleonóra Víg as my fellow PhD student in the GazeCom project does not even begin to do her justice. I am glad you came to Lübeck despite the horrible weather!

Over the years, many undergraduate students have worked with me, for me, and on me as research assistants or during their undergraduate theses. In alphabetical order, these are Judith Berger, Nils Borgmann, Stefan Gruhn, Andreas Gudian, Selin Kahya, Thomas Klähn, Sascha Klement, Martin Lechner, Sönke Ludwig, Irina Nemoianu, Max Pagel, Jan Rüther, Nicolas Schneider, and Henry Schütze.

In our ongoing collaboration with Karl Gegenfurtner's lab at the University of Giessen, Jan Drewes and Christoph Rasche ran the actual experiments. Halszka Jarodzka from the Knowledge Media Research Center Tübingen collaborated with me on the gaze visualization for perceptual learning project.

I received financial support from the German Ministry for Education and Research in the *ModKog* project (grant number 01IBC01B) and the European Commission within the project *GazeCom* (contract no. IST-C-033816) of the 6th Framework Programme, and travel grants from the EC Network of Excellence *COGAIN* (contract no. IST-2003-511598) and the Volkswagen Foundation.

Zusammenfassung

Die visuelle Aufmerksamkeit des Menschen ist auf wenige Ereignisse oder Objekteigenschaften gleichzeitig beschränkt, und nur ein Bruchteil der im Auge eintreffenden visuellen Information wird tatsächlich bewusst verarbeitet. So wird die optimale örtliche Auflösung des Sehens nur in der Fovea im Zentrum der Netzhaut erreicht; ungefähr die Hälfte aller Neurone im visuellen Kortex verarbeitet Information aus den zentralen zwei Prozent des Gesichtsfeldes. Als Konsequenz werden die Augen typischerweise zwei- bis dreimal pro Sekunde bewegt, um die visuelle Szene sukzessive mit der Fovea abzutasten. Die Blickmuster, mit denen verschiedene Beobachter eine Szene abtasten, unterscheiden sich dabei teilweise erheblich, und welche Nachricht einem Bild entnommen wird, hängt auch vom Blickmuster ab. In vielen Problembereichen haben z.B. Experten andere, effizientere Blickstrategien als Laien. Eine grundsätzliche Schwierigkeit beim Erlernen solcher Strategien ist dabei, dass Blickmuster nicht wie die klassischen Bildattribute Helligkeit und Farbe dargestellt werden können.

Ziel dieser Arbeit ist daher die Entwicklung von Systemen zur Aufmerksamkeitslenkung, die das Betrachten einer Szene bzw. eines Videos mit einem optimalen Blickmuster ermöglichen. Ein Gerät zur Messung der Blickrichtung wird mit einem schnellen Videoverarbeitungssystem gekoppelt und die Aufmerksamkeitslenkung erfolgt in Echtzeit und kontinuierlich in drei Schritten: i) basierend auf der derzeitigen Blickposition und den Bildeigenschaften des Videomaterials wird eine Liste von Kandidatenpunkten vorhergesagt, die mit der nächsten Augenbewegung angesprungen werden könnten; ii) die Wahrscheinlichkeit für den gewünschten Kandidatenpunkt wird durch eine Echtzeittransformation wie z.B. lokale Kontrasterhöhung des Videos vergrößert; iii) die Wahrscheinlichkeiten für die übrigen Kandidatenpunkte werden durch z.B. Filterung oder Kontrastabschwächung verringert.

Zuerst erarbeiten wir einige nötige, grundlegende Ergebnisse zum Verständnis des visuellen Systems. Während die meiste Forschung zu Augenbewegungen noch statische Bilder als Stimuli verwendet, sammeln wir einen großen Datensatz an Blickmustern auf natürlichen Videos und finden einen qualitativen Unterschied zu Blickmustern auf Bildern. Weiter untersuchen wir den Zusammenhang von Blickmustern und Bildeigenschaften wie Kontrast, Bewegung und Farbe mit Methoden des maschinellen Lernens. Anhand der geometrischen Invarianten, die die Zahl der lokal genutzten Freiheitsgrade eines Signals angeben, erzielen wir eine höhere Prädiktionsgenauigkeit als bisher in der Literatur berichtet.

Wir implementieren dann eine Reihe von blickwinkelabhängigen Displays, die Videos als Funktion der Blickrichtung modifizieren, und untersuchen ihren Einfluss auf die Aufmerksamkeit. Ein besonderes Augenmerk liegt dabei auf effizienten Bildverarbeitungsalgorithmen und Multiskalenmethoden. Bisherige Systeme waren beschränkt auf eine örtliche Tiefpassfilterung in retinalen Koordinaten, d.h. als Funktion der Blickrichtung. Wir erweitern diese Systeme in zwei Richtungen, die die Berechnungskomplexität erheblich erhöhen. Zum einen filtern wir Videos auch in der Zeitdomäne, da zeitliche Information wie z.B. Bewegung großen Einfluss auf die Aufmerksamkeit hat. Zum anderen erweitern wir die Flexibilität der Filterung; anstelle der Grenzfrequenz einer Tiefpassfilterung erlauben wir die Spezifikation individueller Gewichtskoeffizienten für alle orts-zeitlichen Frequenzbänder einer anisotropen Laplace-Pyramide. Durch verbesserte Algorithmen und eine Implementation auf Graphikhardware erreicht unser System eine Verarbeitungsrate von mehr als 60 Bildern pro Sekunde auf hochaufgelöstem Video. Kritischer als die Durchsatzleistung ist für blickwinkelabhängige Displays jedoch die Latenz zwischen einer Augenbewegung und der Bildschirmaktualisierung; wir erreichen eine Bildverarbeitungslatenz von 2 ms und eine Gesamtlatenz von 20–25 ms.

Erste Versuche mit unseren Systemen zur Aufmerksamkeitslenkung zeigen sowohl einen Effekt auf die Blickmuster als auch einen positiven Effekt auf die Verarbeitungsleistung in visuellen Aufgaben. Probanden, die einen Lehrfilm mit Aufmerksamkeitslenkung sehen, können relevante Bildbereiche in nachfolgenden Testfilmen schneller erkennen. Weitere Experimente zeigen dabei, dass blickwinkelabhängige Videomodifikationen aufgrund der geringen Latenzen unbemerkt bleiben können.

Systeme zur Lenkung der Aufmerksamkeit versprechen daher, ein Bestandteil optimierter Informations- und Kommunikationssysteme der Zukunft zu werden.

Part I

Introduction and basics

"Imagination is the beginning of creation."

George Bernard Shaw

1 Introduction

While you are reading this, countless photons enter your eyes every second that induce myriads of electrical discharges in the roughly two hundred million photosensitive cells of your retinae. Nevertheless, you are likely not overwhelmed by this stream of information, but simply take in one word at a time, while ignoring most of the black and white patterns that constitute the text on the rest of this page. The ability to focus attention on only a tiny selection of objects or features in the visual input is an important property of the human visual system that is also reflected in its anatomy. About half of all cells in the primary visual cortex are devoted to processing information from the central two per cent of visual field, where photoreceptor density and acuity are highest. The high-resolution centre of the retina, the fovea, is typically moved around several times per second to successively sample the visual scene, e.g. one word at a time while reading, whereas peripheral, low-resolution information is mainly used to determine where to look next. During reading, the decision where to look next is straightforward due to the sequential nature of text, and different readers differ in their eye movements mainly only in the number of fixations and the time required to process a single word. In general, however, eye movements are not restricted to reading, but are a ubiquitous phenomenon in virtually all activities. Because of the close relationship of attention and conscious processing with gaze direction, eye movements can be a critical factor in what we perceive in a scene and how well we can solve a visual task, but for complex scenes and tasks, deciding on an optimal eye movement pattern, or scanpath, is far from trivial.

Consequentially, experts in many problem domains exhibit different viewing behaviour from novices, for example in flying a helicopter, driving a car, analysing X-rays, or classifying fish locomotion patterns. To a certain extent, this difference can be explained by the better world model of the expert that

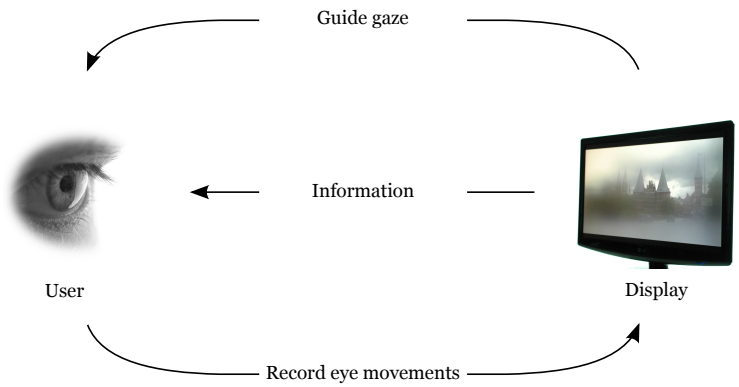


Figure 1.1: *Gaze-contingent interactive displays dynamically adapt to the user's eye movements and simultaneously guide gaze to improve communication.*

allows them to direct their eyes more efficiently. For example, a radiologist might know from experience where a tumour is most likely to be found in a radiogram, or a driver who has passed a complicated intersection before might not need to scan the scene for relevant street signs anymore. Yet, a fundamental problem of experts' eye movements is that they cannot be demonstrated and observed as easily as other movements.

In this thesis, we therefore propose the development of gaze-guidance systems that guide an observer's eye movements through a scene to follow a predetermined, optimal scanpath. Such an optimal scanpath could be the recorded scanpath of an expert or determined by computer vision algorithms; the latter might be useful in safety-critical applications such as in driving, because computers never get tired or distracted, but human drivers occasionally do.

Gaze guidance as it is envisaged in this thesis can be realized by gaze-contingent interactive displays that are connected to an eye tracker that constantly monitors where the user is looking (see Figure 1.1). High-speed eye trackers are commercially available already, and so our focus will be on the algorithms on the display side. If a mismatch between actual and optimal gaze position is detected, the display's contents are changed in real time to steer gaze towards the optimal location. More specifically, the gaze-guidance strategy consists of three parts: i) predict a set of candidate points where a subject will look next, based on the video input and current gaze position; ii) increase the probability for one candidate point to be attended next by increasing image-based saliency there; iii) decrease saliency everywhere else.

We should stress here that the development of full gaze-guidance systems is an interdisciplinary and complex research problem that we cannot expect to completely solve in the context of this thesis. Nevertheless, we will take

the first steps towards this goal and create technical systems that are capable of performing the necessary complex transformations of high-resolution video as a function of gaze in real time, and use these systems for first psychophysical experiments. The analysis of these and other experiments that we have performed will also further improve our theoretical understanding of human oculomotor behaviour.

As we shall see later, even our rudimentary gaze-guidance algorithms have a beneficial effect in training scenarios. Besides the aforementioned safety-critical applications, gaze guidance might also prove useful in the more general case of human-human and human-machine communication, where a visual message might be specified not only by its physical attributes such as colour and brightness anymore, but also by a prescription how to look at it. The gaze-contingent techniques developed in this thesis are also promising for patients with attention disorders such as visual neglect.

Historically, gaze guidance was first proposed in (Barth, 2001), and first experimental results were published in (Dorr, Martinetz, Gegenfurtner, and Barth, 2004; Dorr, Böhme, Martinetz, and Barth, 2005b). In the meantime, several other groups have also begun to investigate how eye movements can be guided by low-level changes to the stimulus, e.g. by locally removing fine spatial details (Su et al., 2005), changing contrast (Nyström and Holmqvist, 2010), or adding phase noise (Einhäuser et al., 2006). These approaches all failed to show a gaze-guidance effect for all but the most extreme local modifications, but suffered from two issues. First, they all modified static images only, but natural viewing behaviour has evolved to deal with dynamic content; motion often marks regions that require immediate visual attention, such as predators or prey, and is therefore a very strong attractor for eye movements. Second, image modifications were not gaze-contingent, i.e. they were static as well. Because image modifications thus could also be perceived foveally, subjects quickly became aware of them and could then consciously compensate for their presence.

For a well-defined visual search task in rendered scenes, McNamara et al. have successfully used very simple, but gaze-contingent stimuli to improve search performance (McNamara et al., 2008, 2009). Blinking Gabor-like stimuli were placed at target locations, but they were switched off before gaze position came close enough to enable their identification. Because stimuli were always presented in the periphery only, subjects did not report noticing them, but still managed to find the highlighted target locations faster.

Linţu and Carbonell (2009) recently described a gaze-contingent display that facilitates inhibition of return and thus faster exploration of the whole

stimulus by blurring image locations once they have been fixated, but have not reported experimental results yet.

1.1 Thesis organization

In more detail, this thesis will be structured as follows. We shall set out with a few basics on image processing and perception in Chapter 2; in particular, we shall review multiresolution methods that make possible efficient processing of frequency subbands, and a geometrical framework that will prove useful for understanding oculomotor control in the following. In the context of this geometrical framework, we can also explain some of our own results on the perception of transparent motions that we shall present in Appendix A.

After this first chapter on fundamentals, the following parts on "Models" and "Systems" will exclusively describe our own original research. Work on a novel and interdisciplinary research field such as gaze guidance cannot be performed by a lone researcher in their ivory tower; where results were obtained in collaboration, the respective contributions are listed at the beginning of each chapter.

The first step in the gaze-guidance strategy outlined above is to predict where subjects will look on dynamic natural scenes. Most research on eye movements so far, however, has dealt with static stimuli only. We shall therefore describe some basic properties of eye movements on several different stimulus types such as static images, natural movies, and Hollywood trailers in Chapter 3. One aspect we are particularly interested in is the variability of eye movements, that is, how similar the gaze patterns of different subjects are. Our hypothesis is that gaze guidance is possible only if variability is neither too low – when all subjects look at the same location – nor too high – when eye movements are essentially random. We shall see that both variability and other eye movement characteristics vary with stimulus type, and one important result thus is that the commonly used paradigm of collecting eye movement data on static images is not very representative of natural viewing behaviour.

In Chapter 4, we shall pursue the question of eye movement similarity further in the domain of image features, and investigate how predictive different low-level image features are for fixation behaviour. In one approach, we shall look at the relationship of features at the current centre of fixation with those at potential saccade targets, and develop novel methods to distinguish feature correlations that are induced by eye movements from those that are image-inherent. In a second approach, we shall use advanced machine learning algorithms to automatically classify movie patches as attended or non-attended, based on a large data set of examples. We compute the geometrical invariants

of the movies, which describe how many degrees of freedom are used locally, and achieve very high prediction rates with these features by using a trick to discard information and thus avoid the curse of dimensionality.

These results shed light on the human visual system, but to efficiently and robustly perform such analyses is also an interesting challenge from a technical viewpoint. More importantly, the implementation of gaze-contingent displays that will be described in the following chapters is impossible without efficient image processing algorithms. Therefore, Chapter 5 is devoted to technical details of the algorithms and the software infrastructure that was developed as a part of this thesis.

We shall then proceed to the core of our research in Chapter 6. Here, we shall present and analyse in detail a series of increasingly complex gaze-contingent displays and space-variant filtering algorithms. The technical goal was to develop algorithms and systems that are fast enough to react to changes in gaze position in very few milliseconds, and yet flexible and powerful enough to enable movie transformations that have a guiding effect on eye movements. As we shall see, this goal can be met by implementing all image processing operations on multiresolution pyramids. We shall show that the introduction of temporal blur in the periphery can alter eye movement statistics even though the blur is hardly noticeable to subjects. In another experiment, we shall increase transformation complexity and show that gaze guidance can indeed have a beneficial effect: spatio-temporal contrast modulations of training videos can facilitate perceptual learning and thus help novices to acquire experts' skills faster. However, the training videos in this experiment had to be precomputed because the algorithm is not suitable for real-time applications. We shall therefore end this chapter with the presentation of an improved algorithm for space-variant spatio-temporal filtering that was implemented on dedicated graphics hardware and has a very low image processing latency of 2 ms.

Finally, we shall conclude this thesis in Chapter 7.

1.2 Previous publications

The work presented throughout this thesis has been published in 17 full journal and conference papers and one book chapter; three papers are currently under submission and two more are in preparation. A poster on the estimation and perception of multiple motions (Dorr, Stuke, Mota, and Barth, 2001) won the "Best Student's Poster Prize" at the Tübingen Perception Conference 2001; the exhibit "Gaze-contingent displays and interaction" won a "Second Prize for Best Exhibit" at the Science Beyond Fiction Conference (Prague, 2009) of projects

funded by the European Commission's Future and Emerging Technologies programme, and was featured by the BBC. A similar exhibit was also presented at the CeBit trade fair 2006 and received press coverage, for example in *Computer Zeitung*, *Lübecker Nachrichten*, and *VDI nachrichten*.

Demo material This thesis deals with the real-time modification of natural videos, but paper does not lend itself easily to reproduce temporal content. Therefore, selected videos are available at
`http://www.gazecom.eu/demo-material`.

*"Given a signal with a 2 Hz bandwagon,
the Nyquil frequency must be 1 Hz to avoid
aliening of the furrier transformer."*

phdcomics.com

2 Basics

In this chapter, we will review some basics that will be relevant throughout the rest of this thesis. We shall start with fundamentals of signal and image processing; one important property of images is that they can be represented in terms of their frequency content. As it turns out, the human visual system processes information on several spatial scales, so a bandpass representation of images is highly useful for probing and understanding visual perception. To efficiently operate on such representations, we shall introduce the concept of multiresolution pyramids, which store information about an image sequence at multiple spatio-temporal scales. Because this information is stored at the optimal resolution, i.e. with no more pixels than necessary, these multiresolution pyramids are central to the work presented in this thesis: we extensively make use of them to efficiently analyse and modify high-resolution video content, and one of our contributions is the fast and flexible implementation of such pyramids for several hardware architectures (see Part III). The synthesis of novel stimuli is another application of multiresolution pyramids; they can be used to create noise that, in some characteristics, resembles natural movies (these stimuli will be used in Chapter 3), and to smoothly combine multiple movies into one blended movie (used in Section 4.4).

Then, we shall turn to a geometrical interpretation of image sequences and see that this interpretation leads to an "alphabet of changes", i.e. a categorization of how a spatio-temporal signal can change, which will prove to be very useful in understanding human vision in several places of this thesis. A further benefit of this framework is that it allows for a fast and robust algorithm for motion estimation, which we will describe briefly in this chapter and use in Section 4.2. The generalization of this framework to an arbitrary number of overlaid signals has obvious technical applications, but also allows further insight into human perception (see below).

Furthermore, we will cover some aspects of the human visual system that are relevant in the context of this interdisciplinary thesis. Obviously, the most important fact is that humans make several eye movements per second to successively sample the visual input with the high-resolution centre of the retina, and we will explore the anatomical basics for this property and some psychophysical data.

2.1 Image processing basics

A signal is a physical representation of a message from a sender to a receiver, such as acoustic waves transmitting voice or changes in electrical current to transmit bits in a computer network. In these examples, the signal usually is a function of time and is denoted as $s(t)$, but signals can also be a function of vectorial variables. In this thesis, we will concern ourselves mostly with images $s(x, y)$, which are a function of space, and image sequences $s(x, y, t)$, which are a function of space and time.

Fourier transform

Fourier analysis can tell us how much of a given frequency is present in a signal by means of a projection onto a set of orthogonal basis functions

$$\mathrm{e}^{-\mathrm{j}2\pi\vec{f}\vec{x}} = \cos(2\pi\vec{f}\vec{x}) + \mathrm{j}\cdot\sin(2\pi\vec{f}\vec{x}).$$

The Fourier transform $S(\vec{f})$ of a continuous image $s(\vec{x})$ is thus defined as

$$\begin{aligned} S(f_x, f_y) &= \int_{-\infty}^{+\infty} s(\vec{x})\cdot\mathrm{e}^{-\mathrm{j}2\pi\vec{f}\vec{x}}\mathrm{d}\vec{x} \\ &= \int_{-\infty}^{+\infty} s(x,y)\cdot\mathrm{e}^{-\mathrm{j}2\pi(f_x x + f_y y)}\mathrm{d}x\mathrm{d}y. \end{aligned}$$

In practice, we can only deal with images that comprise of a finite number of discrete pixel elements, or *pixels*. For discrete images with M by N pixels, the Fourier transform is defined as

$$F(k,l) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cdot\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{kx}{M}+\frac{ly}{N}\right)},$$

and there exist efficient algorithms for the fast computation of this transform (Cooley and Tukey, 1965; Frigo and Johnson, 2005); the extension to image sequences is straightforward.

The inverse transform from a continuous spectrum $S(\vec{f})$ to an image is defined as

$$s(\vec{x}) = \int_{-\infty}^{+\infty} S(\vec{f}) \cdot \mathrm{e}^{\mathrm{j}2\pi\vec{f}\vec{x}} \mathrm{d}\vec{f}$$

with its discrete counterpart

$$f(x,y) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k,l) \cdot \mathrm{e}^{\mathrm{j}2\pi\left(\frac{kx}{M}+\frac{ly}{N}\right)}.$$

The spectrum can also be represented in polar coordinates, so that

$$S(\vec{f}) = |S(\vec{f})| \cdot e^{j\varphi(\vec{f})},$$

with the *phase spectrum*

$$\varphi(\vec{f}) = \arctan\left[\frac{\mathrm{Im}\{S(\vec{f})\}}{\mathrm{Re}\{S(\vec{f})\}}\right]$$

and the *amplitude spectrum*

$$|S(\vec{f})| = \sqrt{\mathrm{Re}^2\{S(\vec{f})\} + \mathrm{Im}^2\{S(\vec{f})\}}$$

One important property of the Fourier transform that we will need in Section 2.4 is that of linearity, i.e.

$$a \cdot s(\vec{x}) + b \cdot g(\vec{x}) \circ\!\!-\!\!\bullet\, a \cdot S(\vec{f}) + b \cdot G(\vec{f}). \qquad (2.1)$$

For a more in-depth review, we refer to a textbook on signal processing, e.g. Lüke (1999) or Oppenheim et al. (1996).

2.2 Spectra of spatio-temporal natural scenes

The amplitude spectra of natural still images exhibit an abundance of lower frequencies and only little high-frequency content, roughly following a $1/f^{\beta}$ falloff with a β between 1.5-2.0 (Field, 1987; Balboa and Grzywacz, 2003; see also Figure 2.1).

For time-varying images, such an analysis is more difficult because the space of "natural" stimuli is much larger. One source of temporal variation in visual input is due to self-motion, and it is not necessarily clear what camera motion should be allowed to simulate this effect. Furthermore, both a still life as well as a busy suburban scene full of dynamic objects are valid natural

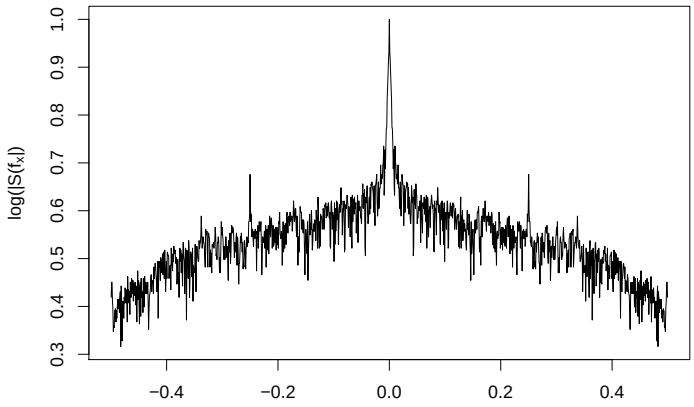


Figure 2.1: *Log-plot of the horizontal amplitude spectrum ($f_y = 0$) of natural images, averaged over all about 11000 frames of our data set of natural movies (see Chapter 3). To reduce the picket-fence effect, frames were masked with a Tukey window prior to the Fourier transform. Clearly, energy is concentrated in the lower frequencies.*

scenes; the average amount of temporal variation is difficult to estimate, in particular if one is interested in the ecological habitat in which the human visual system evolved (and which it presumably is optimized for), see e.g. Balboa and Grzywacz (2003). Nevertheless, at least approximately the temporal spectra of natural scenes exhibit a similar $1/f^\beta$ characteristic to that of spatial spectra of still images; in contrast to the two spatial dimensions in the case of still images, space and time are inseparable, however (Dong and Atick, 1995; Dong, 2001).

Much of vision research has focused on linking perception to the amplitude spectrum of a stimulus for at least two reasons. First, amplitude spectra are more similar across different natural images than phase spectra and therefore are an easier object of investigation; second, complex cells, which are among the most common neurons in the primary visual cortex, encode amplitude only but no phase. However, perception is ultimately dominated by phase information (Oppenheim and Lim, 1981), as demonstrated in Figure 2.2: here, images are created as a mixture of phase information from one and amplitude information from another image. Clearly, the results perceptually are closer to the image with the same phase.

(a) (b) (c) (d)

Figure 2.2: *Perception is dominated by the phase spectrum (Oppenheim and Lim, 1981): the image in (c) is computed from the phase information of (a) and the amplitude spectrum of (b); the image in (d) is computed from the phase information of (b) and the amplitude spectrum of (a). Clearly, (c) is perceptually more similar to (a) and (d) is more similar to (c). Nevertheless, complex cells in the visual cortex are known to encode only amplitude information.*

2.3 Gaussian multiresolution pyramids

A common problem in image processing is to access only certain parts of the frequency spectrum of an image; for example, adaptive thresholding algorithms differentiate between the local variation in image intensity, i.e. image structure, and the local intensity average, i.e. changes that are due to a variation in illumination. In terms of Fourier analysis, the relevant information in this example is contained in the high-frequency part of the spectrum, whereas the low-frequency content should be removed. A straightforward approach to separate low and high frequencies is to apply an appropriate filter; in the spatial domain, this can be achieved by convolution. However, as filter sizes grow larger, convolution quickly becomes very expensive in terms of computational costs. A more efficient solution was developed almost simultaneously in the context of image coding (Adelson and Burt, 1981; Burt and Adelson, 1983a) and of computer graphics (Williams, 1983): *Gaussian image pyramids* (or *mip maps*) store an image in several sizes and with different frequency content. To understand the benefits of this representation, however, we must first review how images are stored in image processing systems.

In theory, we can treat images as continuous functions $s(x, y)$; in practice, however, we can only deal with finite quantities and signals that comprise of discrete elements. An image thus is described as a – usually rectangular – grid of M by N pixels, where each pixel represents the average intensity of a finite neighbourhood around the pixel centre. A fundamental result on the relationship of continuous signals and their discrete approximations is the Nyquist theorem (Shannon, 1949; reprinted as Shannon, 1998), which specifies limits on the size of these neigbourhoods (i.e. the number of pixels):

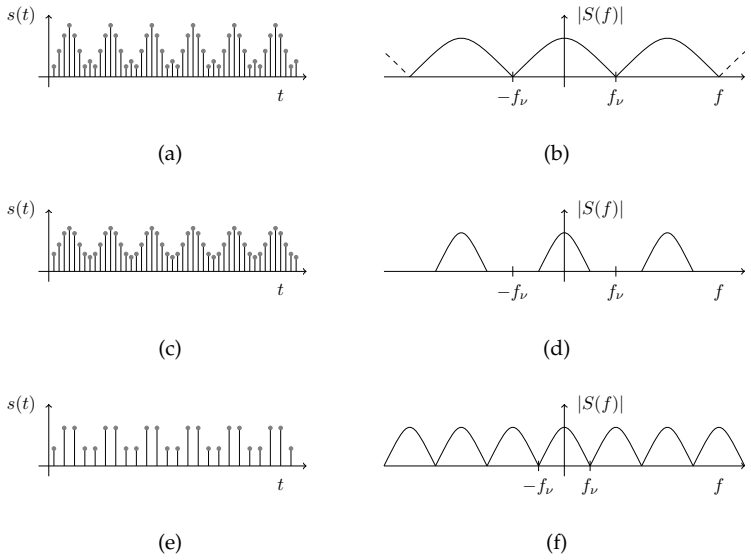


Figure 2.3: *Multiresolution pyramid for one-dimensional signals. (a-b) A signal and its schematic spectrum. For a given sampling rate, frequencies up to the Nyquist frequency f_ν can be represented faithfully. The spectra of discrete signals periodically repeat themselves in intervals of $2f_\nu$. (c-d) The signal in (a), filtered with a 5-tap binomial lowpass filter, and its spectrum. The high-frequency content has been filtered out, so that the spectrum has vanished beyond $f_\nu/2$ (in practice, some high-frequency content may remain, depending on the exact choice of filter kernel). (e-f) After lowpass filtering, the signal can be sampled at a lower rate. After downsampling, the whole spectrum is used up to the (new) f_ν.*

Theorem 1 *If a function $f(x)$ contains no frequencies higher than W cpd, it is completely determined by giving its ordinates at a series of points spaced $\frac{1}{2W}$ degrees apart.*

In other words, a signal must be sampled with at least twice the rate of the highest frequency that should be faithfully represented; it also directly follows that sampling a signal with a higher rate than this *Nyquist rate* does not add useful information. This fact can be exploited by reducing the resolution of an image, or *downsampling*, after lowpass-filtering the image has reduced its Nyquist rate (for a one-dimensional example, see Figure 2.3); usually, resolution is reduced by a factor of two per dimension, so that the number of pixels is reduced by a factor of four.

If the downsampled image is now convolved again with the same kernel, the effective filter width (relative to the original image) has doubled although the computational costs remain constant. Naturally, this operation can be iteratively repeated to obtain a series of successively smaller images, which, when displayed as stacked on top of each other, resemble a pyramid (hence the name). We note here that despite the original nomenclature of referring

to the individual levels according to their position in this pyramid (with low frequencies at the top), we shall identify pyramid levels by their frequency content, so that the "highest" level is the original image. If the size of this image is a power of two, the lowest possible level then consists of only one pixel, which represents the DC component, i.e. the average intensity of the whole image.

We will now briefly present some theoretical considerations; for a more in-depth coverage, we refer to textbooks on multiscale image processing such as Jähne and Haußecker (2000). Implementation details will be described in Part III.

We denote the original image by $I(x, y)$, which has a width of W and a height of H pixels. This original image is also the highest level $G_0(x, y)$ of the Gaussian pyramid. The other Gaussian levels G_k, which have a resolution of $W/2^k$ by $H/2^k$ pixels, can then be computed iteratively:

$$G_{k+1}(x, y) = \sum_{i=-c}^{c} w_i \sum_{j=-c}^{c} w_j \cdot G_k(2x + j, 2y + j).$$

Note that the filtering operation and the downsampling were combined into one step; only the pixels that remain after the downsampling are explicitly computed. The separable filtering kernel w has length $2c + 1$ and coefficients $w_{-c}, \ldots, w_c$; a non-separable kernel $w_{i,j}$ is also possible.

To avoid a phase shift during the filtering operation, the kernel w should be symmetric, i.e. $w_{-i} = w_i$. Also, in order to keep the energy per pixel constant, the filter coefficients should be normalized to unit sum,

$$\sum_{i=-c}^{c} w_i = 1.$$

A further constraint on the filter kernel is the so-called equal contribution principle. To avoid artefacts, each pixel in the high-resolution image has to contribute equally to the low-resolution version (because the number of pixels is reduced by a factor of four, this contribution factor must be $1/4$); for a five-tap filter kernel $w_0 = p, w_{-1} = w_1 = q, w_{-2} = w_2 = r$ thus follows $p+2r = 2q$, which is fulfilled by, for example, the commonly used binomial kernel $(1, 4, 6, 4, 1)/16$.

So far, we have only discussed a Gaussian pyramid of images, i.e. a successive reduction of spatial resolution. However, the same principle can also be applied to image sequences in time. Instead of disregarding every second pixel during the downsampling phase, every second frame of the image sequence can be thrown away. In principle, this is only a trivial modification to the spatial pyramid; in practice, however, the dimension of images is known beforehand

and typically small enough to fit into memory. Videos, on the other hand, are of potentially infinite length and cannot necessarily be stored in memory at once, so that an appropriate buffering scheme has to be developed (see Chapter 6).

Furthermore, image sequences can also be processed on a spatio-temporal pyramid, where the image sequence is filtered and subsampled in both space and time. We differentiate between an *isotropic pyramid*, where space and time are subsampled simultaneously, and an *anisotropic pyramid*, for which each level of a spatial pyramid is decomposed further into its temporal bands. This finer partition of the spectrum comes at a computational cost, however, which is directly linked to its memory requirements.

The Gaussian pyramid necessarily is an overcomplete representation of the original signal since G_0 alone has full resolution. Nevertheless, because of the exponential reduction in resolution on the lower levels, the overall number of pixels is only moderately increased; this is particularly true for an isotropic spatio-temporal pyramid where resolution is reduced on three dimensions simultaneously (and thus by a factor of eight per level). For an input image or image sequence with N pixels, a full spatial Gaussian pyramid will consume $\frac{4}{3}N$ and a temporal pyramid $2N$ pixels; an isotropic spatio-temporal pyramid requires only $\frac{8}{7}N$ pixels, whereas an anisotropic pyramid with S spatial and T temporal levels consumes $\frac{4}{3}T \cdot N$ pixels.

One limitation of the Gaussian pyramid is that it allows only for an octave-based decomposition into subbands. For a finer-grained partition, alternative methods exist, but are less efficient (Köthe, 2004).

2.4 Laplacian multiresolution pyramids

The Laplacian pyramid is based on the Gaussian pyramid and is one of the fundamental data structures in image processing; the seminal paper by Burt and Adelson (1983a) has been cited almost 3000 times. Its applications include image compression (Adelson and Burt, 1981), image mosaicing (Burt and Adelson, 1983b), texture synthesis (Heeger and Bergen, 1995), scene understanding (Jolion and Montanvert, 1992), template matching (Bonmassar and Schwartz, 1998), and medical image enhancement (Trifas et al., 2006).

The underlying idea of the Laplacian pyramid is based on the linearity of the Fourier transform (Equation 2.1). If two images are subtracted from each other, the spectrum of the result will also be equal to the subtraction of the two original spectra; for example, subtracting a lowpass-filtered version of an image from the original image will leave only the high-frequency content and thus is akin to a highpass filter. The different levels of the Gaussian pyramid all have different spectral content; however, because of their different resolution,

they cannot be subtracted from each other pixel-wise. Therefore, lower levels have to be brought to the resolution of the next higher level by upsampling first; this can be achieved by inserting zeros and a subsequent interpolation with a lowpass filter (in practice, this is often the same filter that was used for creating the Gaussian pyramid).[1] We denote the *expanded* version of G_k with $\uparrow G_k$ or G'_{k-1}:

$$G'_{k-1}(x,y) = \sum_{i=-c}^{c} w_i \sum_{j=-c}^{c} w_j \cdot G_k\left(\frac{x-i}{2}, \frac{y-j}{2}\right),$$

and only those pixels of G_k are included in the sums that exist for G_k, i.e. where $(x-i) \bmod 2 = 0$, $(y-j) \bmod 2 = 0$.

Now that adjacent Gaussian pyramid levels have matching resolution, the Laplacian levels can be formed by subtraction:

$$L_k(x,y) = G_k(x,y) - \uparrow G_{k+1}(x,y). \tag{2.2}$$

Because the lowest level G_N of a pyramid with $N+1$ levels has no lower neighbour anymore, we set the lowest level of the Laplacian to the same as that of the Gaussian, i.e. the DC component is

$$L_N = G_N.$$

A schematic overview of the Laplacian analysis phase is shown in Figure 2.4. Based on the rightmost column in that figure, we can see that the middle levels of the Laplacian pyramid represent subbands in octaves; it thus follows that the pyramid is an efficient bandpass representation.

In principle, the same result can be obtained by filtering the signal with a kernel that is the difference of two Gaussian kernels with varying width (DoG filter, see Figure 2.5). However, the pyramid scheme makes use of the reduced Nyquist frequency after filtering to reduce resolution of the lower levels; even large filter widths (relative to the original image) can then be computed efficiently.

A useful property of an octave-based decomposition into subbands is that this roughly matches the energy falloff in natural scenes (see above): lower bands represent only a smaller fraction of the spectrum, but have approximately the same overall amount of energy due to their higher energy concentration.

So far, we have only looked at the analysis phase of the pyramid. It is also possible, however, to reconstruct an image from its bandpass decomposition; this obviously is particularly useful if the frequency bands are modified between

[1] Note that depending on context, in this thesis we shall use the term "upsampling" both to describe only the insertion of zeros as well as the subsequent interpolation.

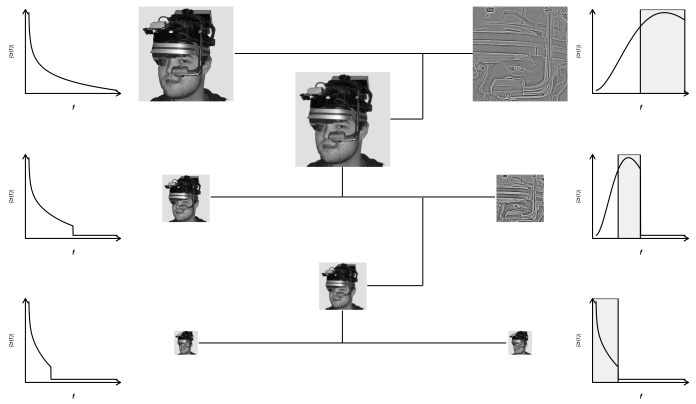


Figure 2.4: *Schematic overview of analysis phase of a spatial Laplacian pyramid. First, the Gaussian pyramid is computed by iteratively filtering and subsampling the input image; the smaller image versions lack high-frequency content and therefore contain only a (lowpass) part of the spectrum (left). Adjacent pyramid levels can be subtracted from each other after an upsampling operation that brings the lower level to matching resolution (middle). Finally, subtraction results yield images that contain information from specific subbands only (right).*

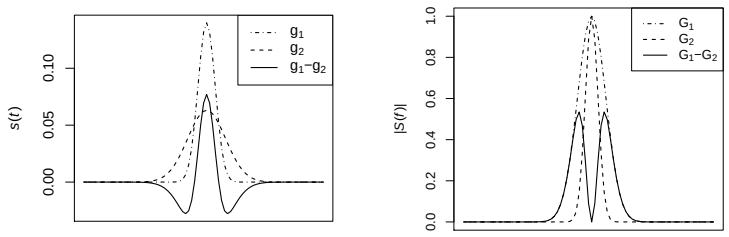
analysis and pyramid *synthesis*. Synthesis is straightforward and is achieved by iteratively upsampling and adding the Laplacian levels:

$$L'_k(x,y) = L_k(x,y) + \uparrow L'_{k+1}(x,y), \tag{2.3}$$

with $L'_0(x,y)$ a faithful reconstruction of the original image (if the L_k remained unmodified); because of the missing neighbour for the DC component, we set

$$L'_N = L_N.$$

From a theoretical standpoint, the upsampling operation in Equation 2.2 is redundant; the same information that is represented by $\uparrow G_{k+1}$ should be contained in the lowpass-filtered version of G_k prior to the downsampling step already. If speed is a major concern, it is therefore possible to subtract the filtered G_k instead of $\uparrow G_{k+1}$ from G_k (note, however, that this prevents us from rolling filtering and subsampling into one step as above). This comes at the expense of accuracy, though; practical filter kernels w do not set all frequencies above half the Nyquist frequency to zero, so that some alias is introduced in the subsampling step. Subtracting $\uparrow G_{k+1}$ in Equation 2.2 ensures that this alias



(a) (b)

Figure 2.5: *Example for DoG (Difference of Gaussians) filter. (a) In the spatial domain, the "Mexican hat" filter is the result of subtraction of two Gaussians with different bandwidths. (b) The bandpass characteristic can be seen clearly in the frequency domain.*

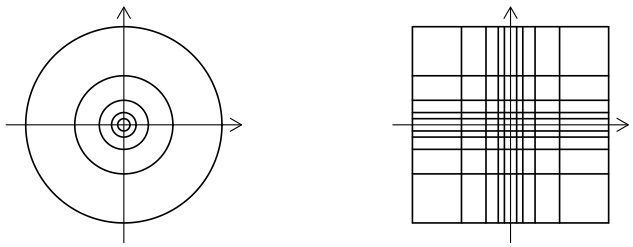
cancels out during pyramid synthesis, so that the pyramid reconstruction is identical to the input image.

In analogy to the Gaussian pyramid, the same principles as above on images can be applied in the temporal domain on image sequences. As in the spatial domain, the temporal Laplacian pyramid can make use of an underlying Gaussian pyramid; a further complexity arises, however, by the need to appropriately buffer intermediate results (temporal filtering requires recent and future items).

Spatio-temporal Laplacian pyramid

Once a temporal Laplacian pyramid is available, it is straightforward to extend the Laplacian pyramid also to the spatio-temporal domain. In analogy to the spatio-temporal Gaussian pyramid, such pyramid can either be isotropic or anisotropic (see Figure 2.6). On an isotropic pyramid, spatial and temporal frequencies vary together, e.g. one subband may encompass low spatial and low temporal frequencies and another subband may comprise of high spatial and high temporal frequencies. The anisotropic pyramid yields a finer-grained decomposition of the spectrum (similar to a steerable pyramid, see Simoncelli et al., 1992) and results in a higher number of subbands that also represent, for example, low spatial and high temporal frequencies. In principle, such an anisotropic decomposition could also be performed on the horizontal and vertical domain of a spatial pyramid; in practice, this is rarely done due to the computational cost.

In Chapter 6, we will present efficient implementations of an isotropic Laplacian pyramid for offline space-variant filtering and of an anisotropic Laplacian



(a) (b)

Figure 2.6: *Spectral decomposition of an isotropic Laplacian pyramid (a) and an anisotropic pyramid (b). Whereas e.g. low spatial and low temporal frequencies are represented together on an isotropic pyramid, the anisotropic pyramid also represents e.g. low spatial and high temporal frequencies.*

for real-time gaze-contingent rendering; in the following, we shall discuss further applications of a spatio-temporal pyramid for image processing.

2.5 Spatio-temporal "natural" noise

The typical input to the human visual system comprises of semantically meaningful objects. For research dealing with the influence of image features on perception, this poses a problem because it is difficult to disentangle the role of semantics (high-level information) from that of syntax (low-level image features). A common approach in vision science is therefore to generate random stimuli that convey no semantic information, but are similar to natural stimuli in their low-level content (Geisler et al., 2006; Jansen et al., 2009). One fundamental low-level property of natural images is that they are strongly correlated; in other words, as we have seen above, the amplitude spectra of natural scenes follow a $1/f^{\beta}$ falloff.

Random (white noise) images have a flat amplitude spectrum. Images without semantic content, but with natural spectrum, can therefore be produced by performing the inverse Fourier transform on a random spectrum with appropriate $1/f^{\beta}$ characteristic:

$$s(x,y) \circ\!\!-\!\!\bullet\, S(f_x,f_y) = \frac{1}{\sqrt{f_x^2+f_y^2}^{\beta}} \cdot X,$$

where X is a uniformly distributed random variable. Care has to be taken, however, how to treat the DC component ($f_x = 0, f_y = 0$); a common solution is to simply set the DC component to zero.

Alternatively, one can compute the Fourier transform of an image, randomly permutate its phase spectrum, and then perform the inverse transform to obtain a noise image with natural amplitude spectrum (Sadr and Sinha, 2004). This method is computationally more expensive than the first method, but models the falloff parameter β more accurately for a given input image.

In the context of this thesis, however, we are interested not only in images, but in image sequences. Because of the $1/f^{\beta}$ falloff of the temporal spectrum of natural movies and its spatio-temporal inseparability (see above), we cannot simply apply the Fourier-based technique to each frame of our videos; performing a Fourier analysis of long high-resolution videos is infeasible due to its memory and computational requirements.

Therefore, we use a spatio-temporal Laplacian pyramid to create spatio-temporal noise with a natural amplitude spectrum. On static images, the Laplacian has been used for texture synthesis before (Heeger and Bergen, 1995). The underlying idea is to measure the first-order statistics for each subband of an input movie and to synthesize an output movie from subbands that have matching statistics:

$$L'_{s,t}(x, y, n) = X_{s,t},$$

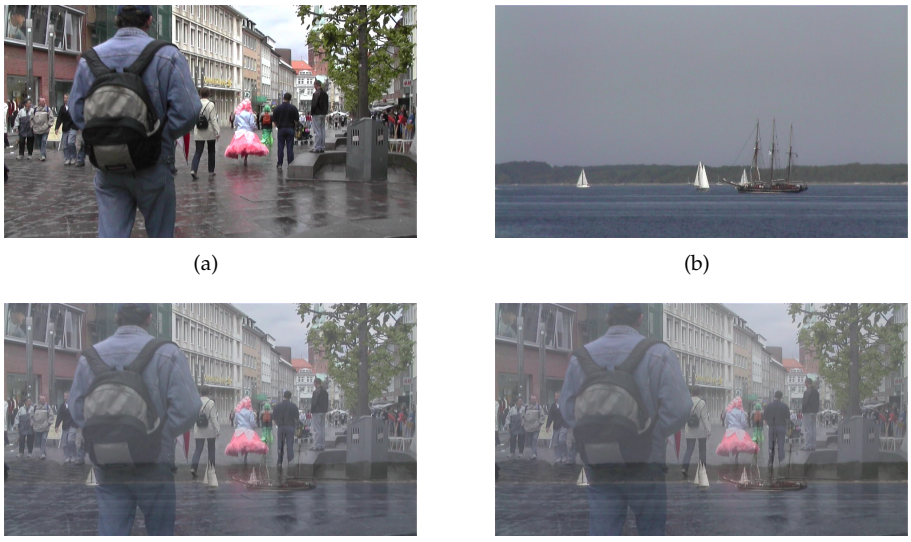
where $X_{s,t}$ is a random variable from a normal distribution with mean zero and a variance $\sigma^2_{s,t}$ that equals the average energy on the frequency band (s, t) of the input movie,

$$\sigma^2_{s,t} = \overline{L_{s,t}(\vec{x})^2}.$$

As a drawback compared to the Fourier-based methods above, this approach can only approximate the original spectrum because of alias during the down-sampling steps of the Laplacian pyramid. However, this technique is more efficient than a Fourier analysis and can be used simultaneously to seamlessly blend between natural and synthetic stimuli (see next section); we will use such noise stimuli in Chapter 3. An example stillshot is given in Figure 3.2.

2.6 Movie blending

The Laplacian pyramid can also be used to seamlessly stitch together two images (Burt and Adelson, 1983b). If a transition region between the images is defined by the same number of pixels on all pyramid levels, the effective size of the transition region will be larger on the lower levels; thus, the amount



(a) (b)

(c) (d)

Figure 2.7: *(a-b) Stillshots of two movies with very different spatio-temporal energy distributions. (c) Result of blending the movies by averaging frame-wise. (d) Result of blending on a spatio-temporal Laplacian pyramid, where the contribution of each movie to each frequency band was equalized. Note, for example, the reduced contrast of the oncoming pedestrians to the left or the man in pink in the middle. The effect is perceptually more striking when real movies instead of stillshots are displayed because temporal change is highly salient (and temporal contrast was also equalized).*

of the spectrum that is blended will vary smoothly, and sharp boundaries are avoided.

For our work on multiple overlaid motions (see Chapter 4), however, we are not interested in stitching two (or more) movies, but in blending them over their whole spatio-temporal extent. A trivial approach would be to simply take the average value of the input movies at each pixel to create the blended pixels. With N input movies $m_0(\vec{x}), \ldots, m_{N-1}(\vec{x})$ and output movie $b(\vec{x})$, this approach would be

$$b(\vec{x}) = \frac{1}{N} \sum_{i=0}^{N-1} m_i(\vec{x}).$$

An example is shown in Figure 2.7(c). However, when movies with very different spatio-temporal contrast distribution (such as the two movies in Figure 2.7(a) and 2.7(b)) are being blended, one movie might perceptually dominate the result. Especially if one movie contains significantly more motion or temporal change, this motion will grab the viewer's attention and render the second movie almost invisible. Therefore, we use a spatio-temporal Laplacian

pyramid to equalize the contribution of the individual movies to each frequency band.

Let $L_{s,t}^{m_i}$ denote the s-th spatial and t-th temporal level of an anisotropic Laplacian pyramid (we here use a pyramid with five spatial and five temporal levels) of movie m_i. First, we compute the square root of the average energy for each frequency band,

$$E_{s,t}^{m_i} = \sqrt{\overline{L_{s,t}^{m_i}(\vec{x})^2}}.$$

Then,

$$L_{s,t}^{b}(\vec{x}) = \frac{1}{N} \cdot \sum_{i=0}^{N-1} E_{s,t}^{m_i} \cdot \sum_{i=0}^{N-1} \frac{L_{s,t}^{m_i}(\vec{x})}{E_{s,t}^{m_i}}.$$

The first two terms that average over E serve to normalize the mean energy of the output movie to the same range as that of the original movie; this is needed only for display purposes, where the dynamic range per pixel usually is fixed to $[0, 255]$. The result of this pyramid-based blending algorithm can be seen in Figure 2.7(d). Obviously, it is impossible to see the difference in temporal contrast to the simple averaging algorithm (Figure 2.7(c)) on these stillshots; nevertheless, spatial contrast of the street scene is clearly reduced to enhance visibility of the sailboat (e.g. at the oncoming pedestrians to the left or the pink-clad person in the centre).

It should be noted – and we shall elaborate on this in Part III – that the temporal filtering operations on the Laplacian pyramid lead to temporal border effects. In order to avoid artefacts, the spatio-temporally blended movies are thus slightly shorter than the original input movies.

2.7 Geometry of image sequences

We will now look at the geometry of image sequences and discuss how we can obtain an "alphabet of signal change" that will prove highly useful later in understanding several human vision phenomena.

Let a gray-scale image sequence be modelled by a function $f : \mathbb{R}^3 \to \mathbb{R}$. There are now four possibilities how f behaves in an (open) region Ω for all $(x, y, t) \in \Omega$: i) f is constant in all directions, $f(x, y, t) = c$; ii) f is constant in all directions but one; iii) f changes in two directions; iv) f varies in all directions. The number of locally used degrees of freedom of a signal is called the *intrinsic dimension* (Zetzsche and Barth, 1990). This concept is important for the representation of images and image sequences because in natural scenes, regions with high intrinsic dimension are less frequent than regions with low intrinsic dimension, and image regions of intrinsic dimension less than two are redundant (Mota and Barth, 2000). An example sketch for a synthetic image is

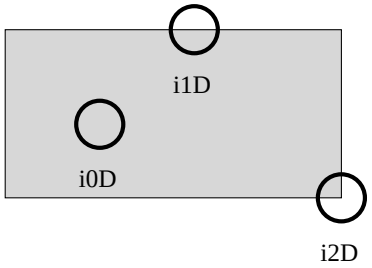


Figure 2.8: *Intrinsic dimension of an image. Homogeneous areas do not change in any direction and are i0D, edges change in a direction orthogonal to their orientation and are i1D, and corners use all degrees of freedom in an image and are i2D. A transient (i.e. appearing or disappearing) corner in an image sequence would be i3D.*

shown in Figure 2.8; for image sequences, homogeneous areas are of intrinsic dimension zero ($i0D$), stationary edges are $i1D$, corners and transient edges are $i2D$, and transient, i.e. appearing or disappearing corners are $i3D$ (for an example of $i2D$ regions on a natural movie, see Figure 4.2).

We will first formalize these observations and then introduce several techniques to estimate the intrinsic dimension, following Mota et al. (2006). For a given region Ω, we choose a linear subspace $E \subset \mathbb{R}^3$, of highest dimension, such that

$$f(\vec{x}+\vec{v}) = f(\vec{x}) \text{ for all } \vec{x}, \vec{v} \text{ such that } \vec{x}, \vec{x}+\vec{v} \in \Omega, \vec{v} \subset E. \tag{2.4}$$

The intrinsic dimension of f is then $3 - \dim(E)$.

Structure tensor

We note the equivalence of Ω above and the constraint

$$\frac{\partial f}{\partial \vec{v}} = 0 \text{ for all } \vec{v} \in E.$$

The subspace E can be estimated as the subspace spanned by the set of unity vectors that minimize the energy functional

$$\varepsilon(\vec{v}) = \int_\Omega \left| \frac{\partial f}{\partial \vec{v}} \right|^2 \mathrm{d}\Omega = \vec{v}^T J \vec{v}, \tag{2.5}$$

where the *structure tensor* J (Bigün et al., 1991) is given by

$$J = \int_\Omega \nabla f \otimes \nabla f \mathrm{d}\Omega$$

with the tensor product $\otimes$. Alternatively, we can then write

$$J = \omega * \begin{pmatrix} f_x f_x & f_x f_y & f_x f_t \\ f_x f_y & f_y f_y & f_y f_t \\ f_x f_t & f_y f_t & f_t f_t \end{pmatrix} \tag{2.6}$$

with a spatio-temporal lowpass filter kernel ω and partial derivatives f_x, i.e. $f_x = \partial f / \partial x$. Therefore, E is the eigenspace associated with the smallest eigenvalue of J, and the intrinsic dimension of f corresponds to the rank of J. Instead of performing an eigenvalue analysis, the intrinsic dimension can also be obtained from the symmetric invariants of J (see below).

We note here that the scale on which the intrinsic dimension is estimated depends on the bandwidth of the derivative operators and the kernel ω; this dependency holds also for the following methods. In practice, we therefore perform our computations on spatio-temporal multiresolution pyramids to analyse several scales simultaneously; we shall present efficient implementations in Part III.

Hessian matrix

From Equation 2.4, it follows that, in Ω,

$$\frac{\partial^2 f}{\partial \vec{w} \partial \vec{v}} = 0 \text{ for all } \vec{v} \in E \text{ and } \vec{w} \in \mathbb{R}^2,$$

which is equivalent to (Golland and Bruckstein, 1997)

$$H\vec{v} = 0 \text{ for all } \vec{v} \in E,$$

where H is the Hessian of f:

$$H = \begin{pmatrix} f_{xx} & f_{xy} & f_{xt} \\ f_{xy} & f_{yy} & f_{yt} \\ f_{xt} & f_{yt} & f_{tt} \end{pmatrix}$$

As in the case of the structure tensor, both the subspace E and the intrinsic dimension can be estimated by eigenvalue analysis of the Hessian of f (Zetzsche and Barth, 1990).

Energy tensor

The previous two methods both have drawbacks. The structure tensor fails at extreme points, whereas the method based on the Hessian matrix fails at

inflection points of the image. To overcome these drawbacks, the energy tensor (Felsberg and Granlund, 2004) is a combination of the structure tensor and the Hessian matrix:

$$E = \nabla f \otimes \nabla f - fH.$$

Felsberg and Granlund (2004) showed that the energy tensor is phase invariant, which is beneficial if an exact localization of features is desired; for example, one common supposition is that the human visual system has a dedicated "where" pathway for localization (Ungerleider and Mishkin, 1982). Furthermore, the statistics of natural images are separable only in terms of phase and amplitude, but not in terms of even and odd filters (Zetzsche et al., 1999), indicating a benefit of distinguishing phase and amplitude.

However, the energy tensor is well-defined only for continuous bandpass signals; since natural videos typically contain a strong DC component, f has to be appropriately filtered in practice.

2.8 Geometrical invariants

The numerically costly eigenvalue analysis for these tensor methods can be avoided by computing the geometrical invariants H, S, and K, which correspond to the minimum intrinsic dimension of a region:

$$\begin{array}{lll} H & = 1/3 \operatorname{trace}(J) & = \lambda_1 + \lambda_2 + \lambda_3 \\ S & = |M_{11}| + |M_{22}| + |M_{33}| & = \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_1\lambda_3 \\ K & = |J| & = \lambda_1\lambda_2\lambda_3. \end{array}$$

Regions where $H > 0$ are at least $i1D$, regions where $S > 0$ are at least $i2D$, and $K > 0$ in $i3D$ regions. The M_{ii} are the minors of J, i.e. the determinants of the matrices obtained by eliminating the row $4 - i$ and the column $4 - i$ from J, e.g.

$$M_{11} = \begin{vmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{vmatrix}.$$

The structure tensor is positive semidefinite and therefore $H, S, K \geq 0$; for the energy tensor, this holds only for appropriately filtered bandpass signals. On the Hessian, the invariants can also become negative.

2.9 Multispectral image sequences

Until now, we have discussed techniques to estimate the intrinsic dimension only for grayscale image sequences. However, colour does play an important

role both in perception (Wichmann et al., 2002) and in oculomotor control; for our work on eye movement prediction in Chapter 4, we therefore here extend the technique based on the structure tensor to multispectral, i.e. colour image sequences (Mota et al., 2006).

As above in Equation 2.4, we look for the subspace E of highest dimension such that, in Ω,

$$\frac{\partial \vec{f}}{\partial \vec{v}} = 0 \text{ for all } \vec{v} \in E.$$

Note that $\vec{f}$ is now a vector from $\mathbb{R}^q$ (for an image sequence with q colour channels), so we choose an appropriate scalar product for $\vec{y} = (y_1, \ldots, y_q)$ and $\vec{z} = (z_1, \ldots, z_q)$ such that $\vec{y} \cdot \vec{z} = \sum_{k=1}^{q} a_k y_k z_k$, with positive weights a_k that can be used to assign higher importance to certain colour channels.

In analogy to Equation 2.5, we may now estimate the intrinsic dimension of $\vec{f}$ by minimizing the energy functional

$$\varepsilon(\vec{v}) = \int_\Omega \left\| \frac{\partial \vec{f}}{\partial \vec{v}} \right\|^2 \mathrm{d}\Omega, \tag{2.7}$$

$$\frac{\partial \vec{f}}{\partial \vec{v}} = v_x \vec{f_x} + v_y \vec{f_y} + v_t \vec{f_t},$$

with $\vec{v} = (v_x, v_y, v_t)$. Rewriting Equation 2.7 as

$$\varepsilon(\vec{v}) = \vec{v}^T J \vec{v},$$

and we arrive at the multispectral structure tensor

$$J = \int_\Omega \begin{bmatrix} \|\vec{f_x}\|^2 & \vec{f_x} \cdot \vec{f_y} & \vec{f_x} \cdot \vec{f_t} \\ \vec{f_x} \cdot \vec{f_y} & \|\vec{f_y}\|^2 & \vec{f_y} \cdot \vec{f_t} \\ \vec{f_x} \cdot \vec{f_t} & \vec{f_y} \cdot \vec{f_t} & \|\vec{f_t}\|^2 \end{bmatrix} \mathrm{d}\Omega,$$

which, in the case of a grayscale video, is the same as in Equation 2.6.

2.10 Orientation estimation

The structure tensor can also be used to estimate local orientation. In an image, orientation simply refers to orientation of an edge; in image sequences, moving objects also produce oriented edges in space-time, so that similar techniques can be used to estimate motion (edge orientation and motion will be used in Section 4.2). Then, we shall briefly cover how the structure tensor can be extended to the *generalized structure tensor*, on which we can estimate the

superposition of multiple motions. In Chapter 4, we will then predict eye movements on overlaid movies (for their generation, see Section 2.6) by their intrinsic dimension computed on the generalized structure tensor.

Local orientation estimation on images based on an eigenvalue analysis of the two-dimensional structure tensor J^{2D} is a standard technique in image processing (for a textbook coverage, see e.g. Jähne, 1999). Similar to the three-dimensional structure tensor above,

$$J^{2D} = \omega * \begin{pmatrix} f_x f_x & f_x f_y \\ f_x f_y & f_y f_y \end{pmatrix},$$

where $f(x, y)$ is the image-intensity function, subscripts indicate partial derivatives, and ω is a spatial smoothing kernel applied to their products. If the rank of J^{2D} is zero (both eigenvalues $\lambda_1, \lambda_2 = 0$), the image patch is homogeneous. A rank of two ($\lambda_1 > 0, \lambda_2 > 0$) indicates a 2D feature, e.g. a corner. An ideal orientation corresponds to a rank of one ($\lambda_1 > 0, \lambda_2 = 0$), with a direction given by the eigenvector corresponding to the zero eigenvalue. To increase robustness in the presence of noise, however, eigenvalues typically are not checked against zero, but against a threshold θ_1 defined by $\lambda_{\max}$, the maximum eigenvalue over all image patches, and a second threshold θ_2 that controls the relative size of the eigenvalues:

$$\begin{aligned} \theta_1 &< \tfrac{\lambda_1+\lambda_2}{\lambda_{\max}} \\ \theta_2 &< \tfrac{\lambda_2-\lambda_1}{\lambda_1+\lambda_2}. \end{aligned} \qquad (2.8)$$

2.11 Motion estimation

To estimate the displacement vector (v_x, v_y) of a moving point in an image sequence $f(x, y, t)$, we assume that the intensity or colour of the point does not change; this is the well-known constant brightness equation (Horn and Schunck, 1981):

$$v_x f_x + v_y f_y + f_t = 0. \qquad (2.9)$$

However, this does not fully constrain $\vec{v}$ at a given position (x, y) and therefore $\vec{v}$ is estimated under the assumption of being constant in a spatio-temporal region Ω. Another way of looking at this is that the gradient of f lies in a plane whose normal is parallel to $(v_x, v_y, 1)$ in Ω. This normal $\vec{n} = (n_x, n_y, n_t)$ can be estimated by minimizing the energy functional

$$E_1(\vec{n}) = \int_\Omega [\nabla f \cdot \vec{n}]^2 \mathrm{d}\Omega.$$

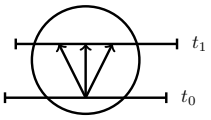


Figure 2.9: *The aperture problem is a fundamental problem for motion estimation of 1D patterns: the veridical motion of the line stimulus cannot be determined locally.*

We can compute the vector $\vec{n}$ up to a scaling factor by finding the eigenvector associated to the zero eigenvalue of the structure tensor J (Haußecker and Spies, 1999). Note that since E_1 is homogeneous, both $\vec{n}$ and $-\vec{n}$ are minimal points of E_1. Actually, $\lambda\vec{n}$ minimizes E_1 when the arguments of E_1 are vectors with norm λ. Therefore, we can think of $\vec{n}$ as homogeneous coordinates for $\vec{v}$ and simply write

$$\vec{v} = [v_x, v_y, 1] = [n_x, n_y, n_t].$$

We can see that $\vec{v}$ can only be estimated reliably if there is only one zero eigenvalue of J, i.e. rank $J = 2$. Even then, motion is only present if $n_t \neq 0$. In practice, due to noise, eigenvalues are rarely zero and thus a confidence measure is employed, e.g. with $\lambda_1 \geq \lambda_2 \geq \lambda_3$,

$$\theta < ((\lambda_1 - \lambda_3)/(\lambda_1 + \lambda_3))^2 - ((\lambda_1 - \lambda_2)/(\lambda_1 + \lambda_2))^2.$$

Based on the eigenvalues of J, we can now describe the local motion patterns in f. Under constant motion $\vec{v}$, the sequence f can be written as

$$f(\vec{x}, t) = g(\vec{x} - t\vec{v})$$

within Ω.

In regions with constant intensity ($\circ$), any motion vector is admissible and the rank of J is zero. A rank of one corresponds to the motion of a straight pattern ($|$); here the motion vectors are ambiguous due to the aperture problem (see Figure 2.9). Other moving patterns ($\bullet$) correspond to rank $J = 2$, and non-coherent motion types such as noise, popping up objects, etc. correspond to rank $J = 3$. These correspondences are summarized in Table 2.1.

A related, but more robust and more efficient algorithm to estimate motion than the above eigenvalue analysis is based on the minors of J (Barth, 2000); the matrix M has elements M_{ij}, $(i, j = 1, 2, 3)$, which are the determinants of the matrices obtained from J by eliminating the row $4 - i$ and the column $4 - j$, e.g. $M_{11} = (\omega * f_x^2)(\omega * f_y^2) - ((\omega * f_x f_y))^2$. Using the constant brightness equation

Moving patterns	rank J_1
∘	0
|	1
•	2
others	3

Table 2.1: *Different moving patterns and the ranks of the structure tensor (Mota et al., 2004a): (∘) constant intensity pattern; (|) 1D pattern; (•) 2D patterns.*

(Equation 2.9), we can obtain several expressions for the velocity vector $\vec{v}$:

$$\begin{aligned}(M_{31}, -M_{21})/M_{11} &= \vec{v}_1 \\ (M_{23}, -M_{22})/M_{12} &= \vec{v}_2 \\ (M_{33}, -M_{23})/M_{13} &= \vec{v}_3 \\ (M_{33}, -M_{22})/M_{11} &= (v_{4x}^2, v_{4y}^2),\end{aligned}$$

with $\vec{v}_4 = \left(\text{sign}(v_{1x})\sqrt{M_{33}}, \text{sign}(v_{1y})\sqrt{-M_{22}}\right)\Big/\sqrt{M_{11}}$. For a translation with constant velocity, the $\vec{v}_i$ are identical; for on- and offset (e.g. occlusions), however, the $\vec{v}_i$ differ, and it is thus straightforward to compare these estimates and use the results only if the maximum difference does not exceed a certain threshold. For a further confidence measure, see the section on estimation of multiple motions below.

The problem of motion estimation has often been studied in the Fourier domain and it is known that additive transparent moving patterns correspond to the additive superposition of Dirac planes through the origin. For an intuitive interpretation of multiple motions in the Fourier domain, we introduce a representation of f in the projective plane below.

Projective plane

A useful representation of different motion types is the projective plane (Mota et al., 2004a); intuitively, the projective plane can be obtained by adding an extra point, the so-called ideal point, to each line of the Euclidean plane under the constraint that all parallel lines share the same ideal point. The set of ideal points is called the ideal line. More precisely, we can use homogeneous coordinates to represent each point of the projective plane by a non-zero vector $\vec{P} = (X, Y, Z)$. Two vectors $\vec{P}$ and $\vec{Q}$ represent the same point if $\vec{P} = \lambda\vec{Q}$, $\lambda \in \mathbb{R}$. Points with a non-null Z-coordinate correspond to points of the Euclidean plane

by means of the projection

$$x = \frac{X}{Z}, y = \frac{Y}{Z}$$

and points with $Z = 0$ represent the ideal points of the projective plane. Using this projection, a point (x, y) of the Euclidean plane is identified to $(x, y, 1)$ in the projective plane; a line can be described by $(X, Y, Z) \mid AX + BY + CZ = 0$.

Turning now to the representation of moving patterns, a single motion layer with velocity $\vec{v}$ is described by

$$f(\vec{x}, t) = g(\vec{x} - t\vec{v}).$$

We can also write this in the Fourier domain as

$$F(\xi, \xi_t) = \delta(\vec{v} \cdot \xi + \xi_t) \cdot G(\xi)$$

and see that F is restricted to a plane through the origin of the Fourier domain, which corresponds to a line in the projective plane. The dual point to this line is then the velocity of the grating, which in the Fourier domain is encoded by the normal of the plane. For 1D spatial patterns, which correspond to lines in the Fourier domain (points in the projective plane), all the admissible velocities are represented by the dual line.

2.12 Generalized structure tensor

In the following, we will briefly discuss the extension of motion estimation to the case of two transparent motions. For the case of an arbitrary number of motions N, we refer to Mota et al. (2001).

An image sequence consisting of two transparent layers can be modeled as

$$f(\vec{x}, t) = g_1(\vec{x} - t\vec{u}) + g_2(\vec{x} - t\vec{v}),$$

where $\vec{u} = (u_x, u_y)$ and $\vec{v} = (v_x, v_y)$ are the velocities of the respective layers. In homogeneous coordinates, the basic constraint equation is

$$c_{xx}f_{xx} + c_{xy}f_{xy} + c_{yy}f_{yy} + c_{xt}f_{xt} + c_{yt}f_{yt} + c_{tt}f_{tt} = 0, \tag{2.10}$$

where $\vec{c} = (c_{ij})^T$ is given by

$$c_{ij} = \begin{cases} u_j v_j & \text{if } i = j \\ u_i v_j + u_j v_i & \text{otherwise} \end{cases}$$

Moving pattern	Projective representation	rank J_1	rank J_2
$\circ$	the empty set	0	0
$\|$	a point	1	1
$\| + \|$	2 points	2	2
$\bullet$	a line	2	3
$\bullet + \|$	a line + a point	3	4
$\bullet + \bullet$	2 lines	3	5
others	others	3	6

Table 2.2: *Different motion patterns (first column) and the ranks of the generalized structure tensors for 1 and 2 motions (table rows). This table shows the correspondence between the different motion patterns and the tensor ranks that can, in turn, be used to estimate the confidence for a particular pattern, i.e. a proper motion model. Note that the rank of J_2 induces a natural order of complexity for patterns consisting of two additive layers.*

with $u_t = v_t = 1$. As in the single motion case, Equation 2.10 implies that the Hessian of f lies in a hyperplane of a six-dimensional space (the space of 3×3 symmetric matrices) whose normal is the symmetric matrix $\boldsymbol{C}$ with entries c_{ij} if $i = j$ and $c_{ij}/2$ if $i \neq j$. In analogy to the case of a single motion, $\vec{c}$ is estimated as the eigenvector related to the smallest eigenvalue of the tensor

$$J_2 = \int_\Omega \begin{bmatrix} f_{xx}^2 & f_{xx}f_{xy} & \cdots & f_{xx}f_{tt} \\ f_{xx}f_{xy} & f_{xy}^2 & \cdots & f_{xy}f_{tt} \\ \vdots & \vdots & & \vdots \\ f_{xx}f_{tt} & f_{xy}f_{tt} & \cdots & f_{tt}^2 \end{bmatrix} \mathrm{d}\Omega. \qquad (2.11)$$

It follows that motion can be reliably estimated only if the rank of J_2 is $\mathrm{ord}(J_2) - 1 = 5$ and the last coordinate of the eigenvector that corresponds to the zero eigenvalue is different from zero (again as in the case of one motion). A summary of different motion types and the corresponding ranks of J_1 and J_2 is given in Table 2.2. Instead of comparing the eigenvalues against zero, a confidence measure based on the geometrical invariants is

$$K^{\frac{1}{m}} < \varepsilon S^{\frac{1}{m-1}} \leq H$$

with $m = \mathrm{ord}(J_2)$ (Mota et al., 2001). However, these conditions are only necessary but not sufficient; further constraints are given in (Mota et al., 2004a).

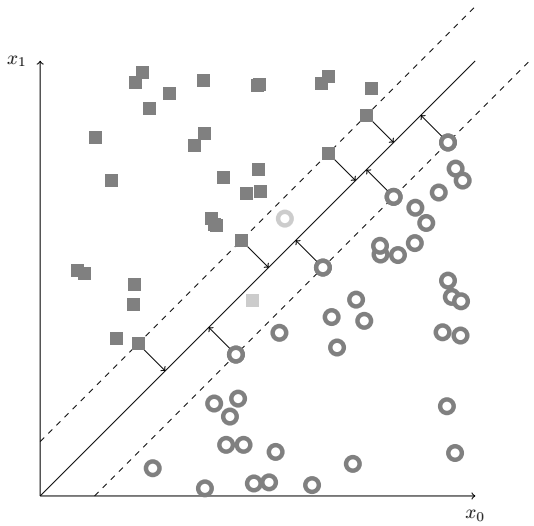


Figure 2.10: *Schematic representation of a soft-margin support vector machine (SVM). The two classes (dark circles and squares) are separated by the hyperplane that results in the largest margin between the two classes; only those data points that lie closest to that hyperplane are relevant for classification, and they are denoted as* support vectors*. The soft-margin SVM allows for some classification errors (light gray data points in the centre) in order to maintain a large margin, which often is beneficial for generalization performance.*

2.13 Support vector machines

For the prediction of eye movements in Chapter 4, we will use machine learning techniques, namely support vector machines. We shall here discuss the fundamental ideas of this classification algorithm very briefly; for a more detailed review, we refer to textbooks, e.g. Bishop (2006).

Given a set of (*training*) data points $\vec{x}_i = (x_{i0}, \ldots, x_{id})$ and a set of class labels $y_i \in \{-1, 1\}$, which assigns each $\vec{x}_i$ to either the negative or positive class, we want to find a hyperplane that separates these two classes; an additional constraint is that the distance of this hyperplane to both classes should be maximal. This constraint is based on the assumption that novel (or *test*) data points from a certain class might lie between the training points from this class and the hyperplane; the wider the margin, the higher the likelihood that these novel data points are also classified correctly. Ultimately, it is such *generalization* of correctly classifying novel data that determines the utility of a classification algorithm; good training performance, on the other hand, is of less importance because the training labels are known already (and could easily be recited by a trivial algorithm, the so-called "rote learner"). For a schematic overview, see Figure 2.10.

Formally, we are looking for the two hyperplanes with normal vector $\vec{w}$ that are defined by the set of points of $\vec{x}$ with

$$\begin{aligned} \vec{w} \cdot \vec{x} - b &= \quad 1 \\ \vec{w} \cdot \vec{x} - b &= -1 \quad , \end{aligned}$$

where $\frac{b}{\|\vec{w}\|}$ specifies the bias, i.e. the distance of the hyperplanes from the origin. The two classes should be separated by the hyperplanes, so

$$\begin{aligned} \vec{w} \cdot \vec{x_i} - b &\geq \quad 1 \quad \text{for } y_i = \quad 1 \\ \vec{w} \cdot \vec{x_i} - b &\leq -1 \quad \text{for } y_i = -1 \quad . \end{aligned}$$

This leads us to a minimization problem: minimize (in $\vec{w}, b$)

$$\frac{1}{2}\|\vec{w}\|^2 \text{ such that } y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1.$$

However, a solution can only be found if the two training classes are linearly separable; if this is not the case, the *soft margin support vector machine* introduces slack variables ξ_i that penalize, but allow wrongly classified data points $\vec{x_i}$:

$$y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 - \xi_i,$$

with the minimization problem

$$\frac{1}{2}\|\vec{w}\|^2 + C \sum_i \xi_i \text{ such that } y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 - \xi_i.$$

The penalty constant C – and other parameters if a kernel support vector machine is used, see e.g. Schölkopf and Smola (2002) – usually have to be subject to an optimization procedure themselves in order to obtain good generalization performance.

A standard implementation of various support vector machine types is the publicly available *libSVM* package (Chang and Lin, 2001), which we shall also use in Chapter 4.

2.14 Human vision basics

The human eye absorbs about 50 billion photons every second in moderate daylight, but sends only about 10 million bits per second to the visual processing areas of the brain (Koch et al., 2006). Ultimately, conscious perception is limited much more severely. Even though an exact number is hard to compute, human

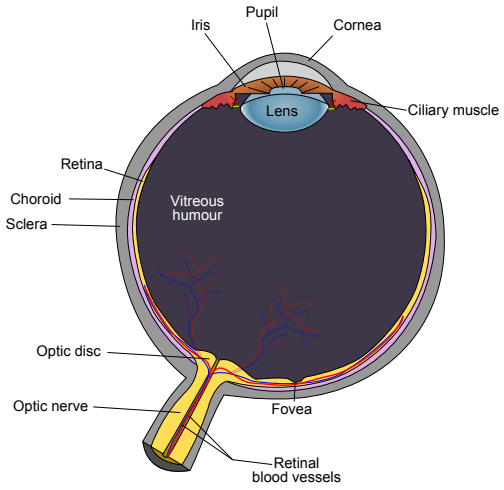


Figure 2.11: *Anatomy of the human eye. Light enters the eye through the cornea and is projected onto the retina through the lens. The fovea is the locus of highest acuity on the retina and consists almost exclusively of cones.*

cognitive capacity for simultaneous processing lies probably in the one-digit range – a popular number is about three bits or seven items (Miller, 1956).

In the following, we shall review some facts about the human visual system, with an emphasis on *convergence*, i.e. how the visual system distills relevant information from the continuous stream of light it receives. One particular trick is space-variant processing, which takes place on almost all levels of the visual system. Only a fraction of the incoming information in the centre of the retina is processed in full detail; to compensate for the peripheral resolution loss, the eyes are moved around several times per second. For a more detailed account of the human visual system, we refer to e.g. Kandel et al. (1995) and Findlay and Gilchrist (2003).

Our own experimental results on the perception of multiple overlaid motions are of interest in the context of mathematical algorithms for the estimation of multiple motions (see Section 2.12) and of prediction of eye movements on such stimuli (see Section 4.4). Because they do not strictly fit our work on gaze guidance, however, we shall present them in Appendix A.

Eye anatomy

The anatomy of the human eye is schematically shown in Figure 2.11. Light enters the eye through the *cornea* and is projected through the *lens* onto the *retina*, where it becomes transduced into electrical, i.e. neural signals. The shape of the

rods	cones
night vision, high light sensitivity	day vision, low sensitivity to light
single photon detection	detect only hundreds of photons
achromatic	three different types: S-, M-, L-cones (blue, green, red)
low spatial resolution	high spatial resolution
highly convergent neural pathways	less convergent pathways
low temporal resolution (12 Hz)	high temporal resolution (55 Hz)
fixed size across the visual field	larger towards the periphery

Table 2.3: *Differences between rods and cones.*

lens leads to spherical aberrations, which are stronger towards the periphery. Because of optical limits and facial features such as the nose, the two eyes have different fields of view. For example, the left visual hemifield is projected onto the temporal hemiretina of the right eye and the nasal hemiretina of the left eye. Even further to the left, the so-called *temporal crescent* is that part of the visual field that can only be seen with the left eye, due to the nose. Therefore, it is also called *left monocular zone*.

The eyeball can move in its socket with six degrees of freedom, three each for rotation and translation. The muscles responsible for these movements are the *superior* and *inferior recti* for up/down movement, the *medial* and *lateral recti* for movement to the left or right, and the *superior* and *inferior obliques* for rotational movement.

Retina

The retina consists of three different cell layers and, interspersed in-between, two synaptic layers. Farthest away from the incoming light is the *outer nuclear layer* that contains the photoreceptors that convert incoming light to electrical impulses. There are two types of photoreceptors, *rods* and *cones*. Rods are far more numerous than cones with about 90 million rods and about 4.6 million cones (Curcio et al., 1990). Rods also come in just one type that is responsible for achromatic low-light vision, whereas cones can be separated into S-, M-, and L-types which are sensitive to short, medium, and long wavelengths, respectively, and thus allow colour vision; cones also have much finer spatial resolution. The differences of rods and cones are summarized in Table 2.3.

As can be seen in Figure 2.12, the density of rods and cones varies greatly across the visual field. At the centre of the visual axis, but slightly offset (on average, about four to eight degrees) relative to the optical axis of the eye, lies the so-called *fovea*, an area of about two degrees diameter that contains only very few rods. Actually, the central one degree has no rods at all and is called

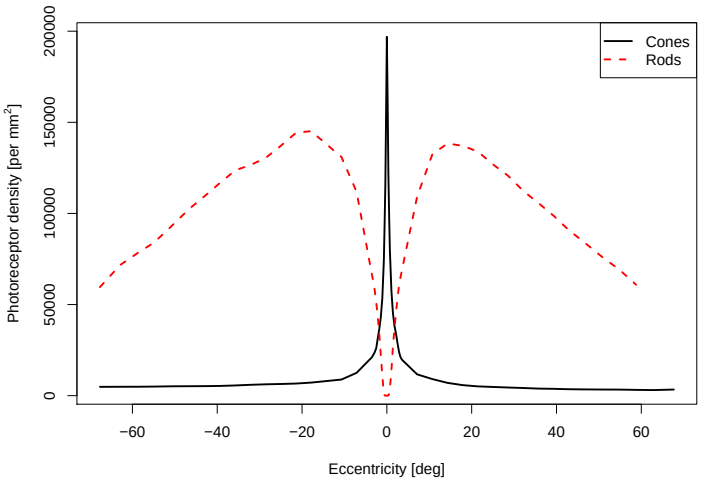


Figure 2.12: *Density of rod and cone photoreceptors as a function of visual eccentricity. Data replotted from Curcio et al. (1990).*

foveola. Because of the *macula*, a region of yellow pigmentation over the fovea, foveal vision is also often called macular vision. The term *parafoveal vision* is used for the visual field around the fovea spanning approximately 10 deg. Towards the periphery, the number of cones decreases sharply, and only very few cones can be found beyond 30 deg eccentricity.

Connected to the outer nuclear layer by synapses that form the *outer plexiform layer* is the *inner nuclear layer*. Here *horizontal*, *bipolar*, *amacrine*, and *interplexiform cells* can be found. At this stage, spatial aspects such as gradients of the scene illumination are processed. For example, there are two types of bipolar cells, centre-depolarizing and centre-hyperpolarizing ones, which respond strongest to dark spots in a bright surround and bright spots in dark surrounds, respectively. To extract such features, information has to be pooled over several photoreceptors, and the size of the pooling area increases towards the periphery. In the foveal region, one bipolar cell is connected to one cone directly, and indirectly to several cones by horizontal cells, which each connect to about six cones. In the periphery, horizontal cells connect to 30–40 cones and bipolar cells connect directly to several cones. A qualitatively similar pattern can be found for rods and rod bipolars, but these show a much higher connectivity with up to hundreds of rods connected to a single bipolar (Adelman, 1987).

Further towards the incoming light lie the inner plexiform layer and, connected by it, the *ganglion cell layer*, which serve to decorrelate the visual input (Atick and Redlich, 1992). In order to increase acuity, the ganglion cells in front

of the fovea are shifted sideways towards the periphery so that rays of light hitting the fovea do not get distorted. The approximately one million ganglion cells can be discriminated morphologically into two types and functionally into three types. Morphologically, about 80% of ganglion cells are of the β-type. They have relatively small cell bodies and dendrites, and their projections go to the *parvo-cellular layer* of the *lateral geniculate nucleus*, a brain area that relays signals to the visual cortex. They are well-suited to the discrimination of fine details, low contrast, and colour. The α-type cells make up about 10% of the ganglion cells. They have larger cell bodies and dendrites, are achromatic, and they respond better to moving stimuli. Their projections go to the *magno-cellular layer*.

Functionally, X-, Y-, and W-type ganglion cells can be distinguished. X-cells project to both the parvo- and magno-cellular layers and are sensitive to stationary stimuli with fine detail. Y-cells, on the other hand, project to the magno-cellular layer only and are sensitive to transient stimuli or motion. Of special interest are the W-type ganglion cells. They are sensitive to coarse features and motion and project to the *superior colliculus*, a brain area that is concerned with the control of involuntary eye movements.

We have here summarized retinal circuitry only very briefly, but should highlight two important characteristics of the retina in the context of this thesis. First, the retina does not simply "sense light" and passes this information on to higher visual areas; complex information processing, in particular the encoding of spatio-temporal change, takes place already at the retinal level. Second, the retina exhibits very space-variant behaviour, with an emphasis of processing on the central part of the visual field.

Beyond the retina

Visual signals from the retina are sent through the *optic nerve* towards the processing sites in the brain. On their way, the fibres from both right and left hemifields are brought together at the *optic chiasm*. The two optic tracts project to three subcortical targets. The *lateral geniculate nucleus* relays data to the *primary visual cortex* at the back of the head. Here, higher-order information processing such as form extraction or motion estimation takes place. Conscious perception is based on these processes. The other two targets are the *pretectum* that is responsible for pupillary reflexes, and the superior colliculus that uses its visual input to generate involuntary eye movements (Kandel et al., 1995).

The optic tracts can be discriminated into two pathways, the *parvo-* and *magno-cellular pathways*. As we have seen before, different types of ganglion cells project to these two pathways. Due to their cell characteristics and their

apparent functional distinction, the parvo-cellular pathway is also called the *what pathway* while the magno-cellular pathway is called the *where pathway*. The P-pathway is concerned with the details of an object and object recognition ("what"). It can actually be further discriminated into the *parvocellular-blob pathway* and the *parvocellular-interblob pathway*. The former deals with the perception of colour, the latter with the perception of shapes and depth. The M-pathway is concerned with the spatial relationship of objects and behaviour oriented towards them ("where"). This distinction can not only be made on anatomical grounds, but can also be established with patients who suffered a brain damage, usually due to a stroke, that left only one of the pathways intact. Despite this distinction, there are interactions between the pathways at many different levels (Kandel et al., 1995).

In analogy to these pathways, the human visual system also operates at a multitude of spatial scales or channels (Blakemore and Campbell, 1969).

Psychophysics

As we have seen in the previous sections, the visual system is optimized for the processing of fine details around the fovea, which is also reflected in psychophysical measurements.

The horizontal field of view is approximately 180 deg, the vertical field of view spans about 130 deg. Only the central 30 deg on both axes have a reasonable spatial resolution that can be used for object recognition and are thus called the "useful" visual field. At eccentricities exceeding 30 deg, only ambient motion can be perceived.

Whereas spatial resolution clearly is best in the centre of the visual field, the distribution of temporal resolution is more complex to establish. On the one hand, the threshold for the perception of translational motion increases with eccentricity, although not as strongly as that for spatial frequency (Leibowitz et al., 1972); multiple motion detection is also impaired in the periphery (de Bruyn, 1997). On the other hand, the perceptual sensitivity to flicker is much higher in the periphery (Baker and Braddick, 1985), even though the actual thresholds at which flicker can be detected are higher foveally because of the higher temporal resolution of cones.

As we have seen, many visual functions depend on eccentricity; the eyes move several times per second to successively sample a visual scene with the high-acuity fovea. In the following, we shall give a short introduction to some eye movement properties.

2.15 Eye movements

The first qualitative description of jump-like eye movements was made by the French ophthalmologist Javal (1878), who used a mirror to monitor the eye movements subjects made while reading. Quantitative studies of eye movements date back to Buswell (1935) and Yarbus (1967). They observed that the eyes move about two to three times per second in jump-like movements, the so-called *saccades*. Because saccades are rapid and have a duration of only 20–80 ms, about 90% of viewing time are spent with the eyes apparently stationary, in *fixations* with a typical duration of about 150–600 ms (Duchowski and Vertegaal, 2000). However, a closer look reveals that even during fixations, the eyes are not perfectly still and exhibit fixational eye movements, such as microsaccades. The role of these fixational eye movements is still under debate; one common explanation is that they prevent neural adaptation in the retina that would otherwise lead to fading of the perceived image (Rolfs, 2009).

Saccade dynamics

Because of mechanical constraints, the amplitude of (horizontal) eye-in-head movements is limited to about ± 55 deg away from the central position, and saccades with an amplitude of more than 30 deg usually are accompanied by a head movement.

For saccades with an amplitude of 5–50 deg, duration is linearly determined by amplitude (Becker, 1991):

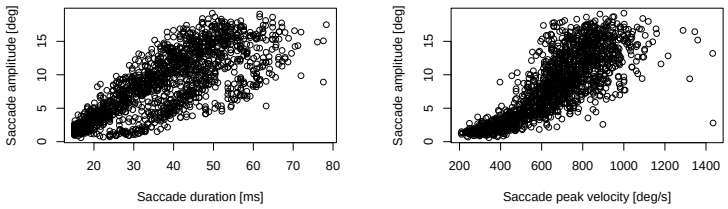
$$D = D_0 + dA$$

with $D_0 \approx$ 20–30 ms and $d \approx$ 2–3 ms/deg, e.g. a typical saccade of 15 deg has a duration of about 50–75 ms. Our own data (see Figure 2.13) shows approximately similar results, but with a lower D_0. For smaller saccades of up to five degrees, the above equation should be replaced by a power law,

$$D = D_1 A^p,$$

with $D_1 \approx D_0$ and $p \approx$ 0.15–0.2. For saccades larger than 50 deg, duration increases overproportionally because of the mechanical limits of the eye.

From Figure 2.13, we can see that peak velocity of a saccade increases with amplitude and saturates at about 700–1000 deg/s.



(a) (b)

Figure 2.13: *Relationship of saccade amplitude with saccade duration (a) and peak velocity (b). Eye movements were recorded at 1250 Hz from one subject fixating a marker that was randomly displaced on the screen every 900 ms. Overall, 1674 saccades were extracted.*

Saccade accuracy

The typical saccade duration of 20–80 ms is less than it takes an optic signal to be transduced to a neural signal and projected to the brain areas that are concerned with eye movements; thus, saccades are too fast to be guided by visual feedback and have to be ballistic. In complex scenes, saccades tend to undershoot, i.e. fall short of an intended target, by up to 18% of veridical amplitude to target (Rasche and Gegenfurtner, 2010).

Saccadic suppression

During a saccade, the visual input moves across the retina with a velocity of several hundred degrees per second. However, humans do not perceive this global motion due to *saccadic suppression*, which reduces visual sensitivity shortly before, during, and shortly after saccades. Sensitivity is reduced by several order of magnitude for image displacements and about two- to threefold for brief flashes; in the latter case, fine spatial details are less suppressed than low spatial frequencies (Deubel et al., 2004).

Other types of eye movements

In the following, we shall give a brief overview of other types of eye movements besides saccades. *Smooth pursuit* eye movements are used to fixate objects that move across the visual field. They have a very low latency of around 100 ms (Pack and Born, 2001), but their velocity is usually only up to 40 deg/s. A *nystagmus* is a periodic eye movement to track rotating targets, or to compensate for rotation of the body, respectively. It consists of two phases, the slow phase in which a visual feature is fixated, and the fast phase that brings the eye back to its initial position. The movement on the horizontal axis of both eyes

towards or away from each other is called *vergence*. It allows for fixation of the same object with both eyes, a prerequisite for stereopsis. The velocity of vergent eye movements is fairly slow at around 10 deg/s. *Vestibular* eye movements are made to maintain fixation of an object while the head or body moves. Head movements can have peak velocities of up 300 deg/s. The head-eye coordination is controlled by the *vestibulo-ocular reflex*. This reflex, which is controlled by information from the vestibular organs, is complemented by the *optokinetic reflex*, which is triggered by optical flow.

Attention and gaze

Everyday experience tells us that it is possible in principle to deploy attention without an accompanying fixation. However, deployment of covert attention is quite rare under natural viewing conditions. Indeed, eye movements are always preceded by a shift in attention to the subsequent saccade target (Currie et al., 1995; Deubel, 2008).

Part II

Models

The ultimate goal of this thesis is to develop systems that can guide the gaze of an observer. Before we can build such systems, however, we first have to understand more about how the human visual system selects saccade targets under natural viewing conditions. Thus, the following part of this thesis will present models of eye guidance on dynamic natural scenes.

The majority of research on eye movements so far has dealt with synthetic scenes, e.g. simple geometrical shapes popping up or being translated, or with static images. Only recently, more groups have begun to address eye movements on dynamic content. In Chapter 3, we will present a large data set of eye movements from more than 50 subjects who watched naturalistic high-resolution videos. We also collected data for several control conditions, such as noise movies that had the same frequency content as the natural movies but showed no discernible objects, "stop-motion" movies that lacked continuous motion, and professionally cut Hollywood action movie trailers. We compared eye movements across these conditions and found that gaze behaviour on natural movies differs both in first-order characteristics such as saccade amplitudes and fixation durations, and in second-order characteristics such as the similarity of eye movements of different observers.

This study required statistical methods and image processing techniques for the generation of stimuli and for data analysis. It is a product of a collaboration with Karl Gegenfurtner, University of Giessen; a manuscript is currently under submission (Dorr, Gegenfurtner, and Barth, 2010a).

Chapter 4 then will address the question of how well eye movements can be predicted based on low-level image features. We shall first test a recent hypothesis that neural adaptation might play a role in oculomotor control. To this end, we analysed the data set from Chapter 3 and looked at the correlations of a variety of image features along scanpaths that are induced by the subject's eye movements. However, natural scenes themselves are highly correlated in space and time, and we therefore developed novel algorithms to create artificial scanpaths that can serve as appropriate reference conditions. Once the image-inherent correlations were accounted for, we could not find evidence for a contribution of low-level features at the centre of gaze to saccade target selection.

An emphasis in this study was placed on solid image feature extraction on a spatio-temporal multiresolution pyramid, for example for orientation and motion estimation. This study's findings have been published in (Dorr, Gegenfurtner, and Barth, 2009a).

Then, we shall turn to the use of machine learning techniques for the prediction of eye movements. This work was performed in close collaboration with Eleonóra Víg, who signed responsible for the machine learning algorithms. A

major problem when dealing with the classification of movie patches is the curse of dimensionality. Because the dimensionality of the classification problem typically grows with the number of pixels (which grows cubically for space-time sub-volumes of a movie), classification quickly becomes intractable.

We therefore applied a trick to reduce the dimensionality of the problem. For image feature extraction, the tensor methods from Chapter 2 were applied on a spatio-temporal multiresolution pyramid. Then, only one scalar value per spatio-temporal scale, namely the average feature energy in a neighbourhood around fixation, was used to train a support vector machine with attended and non-attended locations.

The prediction results we obtain on this low-dimensional representation are very favourable. An additional finding is that movie regions that change in more spatio-temporal directions are also more predictive for eye movements. Because these regions are also less frequent, this finding is an indicator of efficient coding in the human visual system.

Some of these results that were obtained by state-of-the-art machine learning and image processing techniques have already been published (Vig, Dorr, and Barth, 2009); a more detailed analysis is currently in preparation.

Finally, we shall apply our machine learning framework to the case of multiple transparent movies. Gaze data from subjects watching two natural movies that had been blended on the spatio-temporal Laplacian pyramid was analysed in terms of the rank of the generalized structure tensor (see Chapter 2), and it turned out that the generalized structure tensor indeed represents higher-order motion types better than the classical structure tensor.

Experimental data collection and analysis were run by Laura Pomarjanschi; a manuscript that includes first results is currently under submission (Barth, Dorr, Vig, Pomarjanschi, and Mota, 2010).

"Gaze on them, till the tears shall dim thy sight,
But keep that earlier, wilder image bright."

William Cullen Bryant

3

Eye movements on natural videos

In this chapter, we shall investigate the variability of eye movements on dynamic natural scenes, and compare this variability with that on other stimulus types such as static images or semantic-free noise.

3.1 Previous work

Humans make several eye movements per second, and where they look ultimately determines what they perceive. Consequently, much research over several decades has been devoted to the study of eye movements, but for technical reasons, this research has mostly been limited to the use of static images as stimuli. More recently, however, an increasing body of research on eye movements on dynamic content has evolved. Blackmon et al. (1999) reported evidence for the "scanpath theory" (Noton and Stark, 1971) on very simple, synthetic dynamic scenes. Several studies were concerned with modelling saliency, i.e. the contribution of low-level features to gaze control (e.g. Itti, 2005; Meur et al., 2007), and found, not surprisingly, that motion and temporal change are strong predictors for eye movements. Tseng et al. (2009) quantified the bias of gaze towards the centre of the screen and linked this centre bias to the photographer's bias to place structured and interesting objects in the centre of the stimulus. Carmi and Itti (2006) investigated the role of scene cuts in "MTV-style" video clips and showed that perceptual memory has an effect of eye movements across scene cuts. 't Hart et al. (2009) used recordings of a head-mounted and gaze-controlled camera (Schneider et al., 2009) to replay the visual input during outdoor walking in either a continuous movie or in a random sequence of 1 s-stillshots. The distribution of gaze on the continuous stimuli was wider than for the static sequence and also a better predictor of gaze during the original natural behaviour. The variability of eye movements of different observers was

studied with an emphasis on how large the most-attended region must be to encompass the majority of fixations in the context of video compression (Stelmach et al., 1991; Stelmach and Tam, 1994) and enhancement for the visually impaired (Goldstein et al., 2007). Marat et al. (2009) evaluated eye movement variability on short TV clips using the Normalized Scanpath Saliency (Peters et al., 2005). Comparing the viewing behaviour of humans and monkeys, Berg et al. (2009) found that monkeys' eye movements were less consistent with each other than those of humans. Hasson et al. (2008a) presented clips from Hollywood movies and everyday street scenes to observers while simultaneously recording brain activation and eye movements; both measures showed more similarity across observers on the Hollywood movies (particularly by Alfred Hitchcock) than on the street scenes. However, when playing the movies backwards, eye movements remained coherent whereas brain activation did not.

With a few exceptions ('t Hart et al., 2009; Tseng et al., 2009; Carmi and Itti, 2006; Stelmach and Tam, 1994), these studies used professionally recorded and cut stimulus material such as TV shows or Hollywood movies. Arguably, such stimuli are not representative of the typical input to a primate visual system. Other authors therefore have also studied gaze behaviour in real-world tasks, such as driving (Land and Tatler, 2001; Land and Lee, 1994), food preparation (Land and Hayhoe, 2001), and walking around indoors (Munn et al., 2008) and outdoors (Schneider et al., 2009). We here set out to investigate how similar eye movements of a large group of observers are on natural, everyday outdoor videos and compare this similarity with the similarity of eye movements on several other stimulus categories.

3.2 Our work

For our work on gaze guidance, we aim to understand the common and the optimal degree of variability in eye movements that observers make on dynamic natural scenes. Intuitively, a very low variability, i.e. a scene on which all observers follow the same gaze pattern, offers little room to guide the observer's attention; at the same time, a very high variability might indicate a dominance of idiosyncratic viewing strategies that would also be hard to influence. We cannot expect to easily quantify such variability in absolute terms and therefore resort to a comparison of variability on different stimulus categories. As a baseline, we use high-resolution videos of everyday outdoor scenes without cuts because these are very close to natural viewing behaviour. For a lower limit of variability, we compare these natural videos to professionally-cut trailers for Hollywood action movies. On the other end of the variability range,

we investigate the role of semantic information by measuring variability of eye movements on synthetic noise movies with a natural spatio-temporal amplitude spectrum. As we have seen in Section 2.2, the amplitude spectra of natural still images exhibit an abundance of lower frequencies and only little high-frequency content, roughly following a $1/f^{\beta}$ falloff with a β between 1.5-2.0 (Field, 1987). Noise images with a similar amplitude spectrum have been used as semantic-free models of natural images (Geisler et al., 2006; Jansen et al., 2009) and can easily be synthesized by phase-scrambling a natural image or by appropriately filtering the Fourier transform of a random image. Because of the computational cost of performing such operations in space and time, we here use the more efficient approach to generate movies with a natural (spatio-temporal) amplitude, but random phase spectrum that was presented in Section 2.5.

We also investigate the role of continuous temporal change in dynamic scenes as opposed to static images. A common psychophysical paradigm for the collection of eye movements on static images is to present a "random" series of images for several seconds each. If the presentation time is too short, obviously not much information can be extracted beyond the very first few fixations; if the presentation time is too long, on the other hand, observers might lose interest and resort to idiosyncratic top-down viewing strategies in absence of sufficient bottom-up stimulation. Indeed, some authors have argued that the direct contribution of low-level saliency to the choice of fixation targets decreases with viewing time (Parkhurst et al., 2002; Itti, 2006), while others, e.g. Tatler et al. (2005), argue that only the top-down strategy changes (that picks targets from a low-level defined set of candidate locations). Random series of images are typically used to avoid any potential bias introduced by prior knowledge of the stimulus, i.e. any upcoming stimulus image should be unpredictable by the observer. Contrary to this paradigm, we specifically designed the stop-motion movies to be fully predictable by the subjects; they were created so that the presented sequence of single frames (that were each shown for three seconds) was identical to one of the natural movies except for the absence of continuous motion. A similar study to compare static and continuous image presentation was recently undertaken by 't Hart et al. (2009), who took 1 s long stillshots from a set of natural videos and reassembled them into random sequences. However, in their experiment, depicted scenes were not predictable by the previous images, whereas in the work presented here, most of the scene (the static background, but not moving objects) stayed the same across image transitions.

Furthermore, we evaluate whether the scanpath theory (Noton and Stark, 1971) can also be applied to dynamic natural scenes; so far, it has only been tested empirically for very simple, synthetic videos (Blackmon et al., 1999).

3.3 Methods

In the following, we shall first describe the stimuli and experimental conditions and then review data analysis methods.

Natural movies

A JVC JY-HD10 HDTV video camera was used to record 18 high-resolution movies of a variety of real-world scenes in and around Lübeck. Eight movies depicted people in pedestrian areas, on the beach, playing mini golf in a park, etc.; three movies each mainly showed either cars passing by or animals; a further three movies showed relatively static scenes, e.g. a ship passing by in the distance; and one movie was taken from a church tower, giving a bird's-eye view of buildings and cars. All movie clips were cut to about 20 s duration; their temporal resolution was 29.97 frames per second and their spatial resolution was 1280 by 720 pixels (NTSC HDTV progressive scan). All videos were stored to disk in the MPEG-2 video format with a bitrate of 18.3 MBit/s. The camera was fixed on a tripod and most movies contained no camera or zooming movements; only four sequences (three of which depicted animals) contained minor pan and tilt camera motion. A representative sample of stillshots is given in Figure 3.1.

Trailers

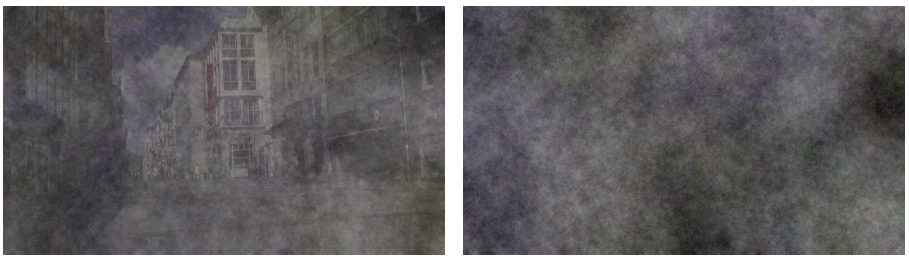
The official trailers for the Hollywood movies "Star Wars - Episode III" and "War of the Worlds" were used for this condition. Both had a duration of about 32 s each and a spatio-temporal resolution of 480 by 360 pixels, 15 fps, and 480 by 272 pixels, 24 fps, respectively. Some text on plain background is shown during the first and last few seconds, but in-between, these trailers are characterized by a large amount of object motion, explosions, etc., and many scene cuts (21 and 24, respectively). Camera work is deliberately aimed at guiding the viewer's attention, e.g. by zooming in on the face of a scared child.

The accompanying sound track was not played during stimulus presentation.

Figure 3.1: *Stillshots from all movies used in the natural condition.*

Natural noise

Six movies similar in content to the above natural movies (but of longer duration) were decomposed on a spatio-temporal anisotropic Laplacian pyramid with eight spatial and five temporal levels, and their frequency subbands were replaced with random noise with same statistics. After synthesis of the pyramid, the resulting output movies had approximately the same low-level content, i.e. the same spatio-temporal amplitude spectrum, but contained no discernible objects (see Section 2.5). In order to avoid that subjects lost interest too quickly watching such movies, the displayed movie during a trial would oscillate between the original and the noise movie, with intermediate periods where some noise and some natural structure were present. The transition function is shown as dashed line in Figure 3.8 and stillshots from a resulting movie are given in Figure 3.2. Overall, six such movies of 63 s duration each were shown to the subjects.



(a) (b)

Figure 3.2: *Example of movie that oscillates between a natural movie and one with similar amplitude, but random phase spectrum. See Figure 3.8 for the time course of oscillations. (a) 50% noise level; some natural movie structure is still visible. (b) 100% noise level.*

Stop-motion

Nine out of the 18 natural movies were also shown in a "stop-motion" condition. Instead of displaying all (around) 600 frames at 30 frames per second, only every 90th frame was displayed for a full three seconds. Thus, the sequence and timing of depicted events was the same as in the original movie, but was revealed only in steps similar to scene cuts (note that typically, the whole scene layout changes with a cut; here, only the position and appearance of moving objects changes, whereas the background stays the same).

Static images

Finally, stillshots from the nine movies not used in the "stop-motion" condition were used to record eye movements on static images. Similar to the "stop-motion" condition, every 90th frame of a movie was used, but the order was randomized over movies and the temporal sequence of stillshots so that subjects could not predict a stimulus from the previous one.

Data recording

All eye-movement recordings were made using the commercially available SR Research EyeLink II eye tracker running at 250 Hz. This tracker compensates for small head movements, but subjects' heads were still fixated in a chin rest. After an initial binocular calibration, only monocular data from the eye with the smaller validation error was used throughout the experiments (mean validation error 0.62 deg). Subjects were seated 45 cm away from an Iiyama MA203DT screen that had a width of 40 cm and a height of 30 cm. Since the videos (except for the Hollywood trailers) had an aspect ratio of 16:9 and would not natively fit on the monitor with an aspect ratio of 4:3, they were displayed

in the "letterbox" format with black borders below and above such that pixels had the same physical width as height. Videos covered about 48 by 27 degrees of visual field, and about 26.7 pixels on the screen corresponded to one degree of visual angle for the high-resolution movies (1280 by 720 pixels).

For a smooth playback of videos, two computers were used. The first computer ran the eye tracking software, the second was used for stimulus decoding and display. Therefore, gaze recordings and video timing had to be synchronized, for which two strategies were employed. In experiment one, the display computer sent a trigger signal to the tracking host via a dedicated ethernet link whenever a new frame was displayed (every 33 ms); these trigger signals and the gaze data were stored to disk using common timestamps by the manufacturer's software. In all other experiments, a three-computer setup was used. Gaze measurements were sent from the tracker across an ethernet link to a relay computer and from there on to the display computer, where independent threads wrote both gaze and video frame timestamps to disk using the same hardware clock. This seemingly complicated setup was necessary because the tracker manufacturer's API requires the network to be constantly monitored (polled) for new gaze samples to arrive, wasting CPU cycles and potentially disturbing the smooth playback of (high-resolution) video. The task of the relay computer thus was to constantly check whether a new gaze sample had arrived from the tracker, using the proprietary software; each sample was then converted to a custom clear-text format and sent on to the display computer, where the receiving thread (performing a "blocking wait" on its network socket) would only run very briefly every four milliseconds (at a sampling rate of 250 Hz). Because of the low system load and the low conversion rate, this relay step did not incur a significant delay; the latency of both synchronization approaches is in the single-digit millisecond range, and the latter approach has also been used successfully for latency-critical gaze-contingent paradigms, see Part III.

Fifty-four subjects (students at the Psychology Department of Giessen University; eight male, 46 female) participated in experiment one. After an initial nine point calibration and the selection of the preferred eye, all 18 movies were shown in one block. After every movie presentation, a drift correction was performed.

For the repetitive presentation of movies in experiment two, 11 subjects came to the lab for two days in a row. Each day, the trailers and six movies out of the 18 natural movies from experiment one (beach, breite_strasse, ducks_-children, koenigstrasse, roundabout, street) were shown five times each in randomized order. As in experiment one, the eye tracker was set up with a

nine point calibration procedure initially and drift corrections after each video clip; this scheme was also adhered to in the following experiments.

A further 11 subjects participated in experiment three and watched nine "stop-motion" movies, which were created from a subset of the 18 natural movies from experiment one (beach, breite_strasse, bridge_1, bumblebee, ducks_children, golf, koenigstrasse, st_petri_gate, st_petri_mcdonalds). Then subjects were shown, after another calibration, stillshots from the remaining nine movies (bridge_2, ducks_boat, doves, holsten_gate, roundabout, st_petri_market, street, sea, puppies) in randomized order. Stillshots were shown for two seconds each.

Finally, 12 subjects partipicated in experiment four and were shown the oscillating natural noise movies.

In all of the above experiments, subjects were not given any specific task other than to "watch the sequences attentively".

Data analysis

Gaze data preprocessing The eye tracker marks invalid samples and blinks, during which gaze position cannot be reliably estimated. Furthermore, blinks are often flanked by short periods of seemingly high gaze velocity because the pupil gets partially occluded by the eye lid during lid closure, which in turn leads to an erroneous gaze estimation by the tracker. These artefacts were removed and recordings that contained more than five per cent of such low-confidence samples were discarded. In experiment one, this left between 37 and 52 recordings per video sequence, and 844 recordings overall.

Saccades are typically extracted from raw gaze recordings based on the high velocity of saccadic samples. However, the choice of an optimal threshold for saccade velocity is difficult: a low threshold might lead to a high false positive rate, i.e. the detection of too many saccades due to microsaccades and impulse noise in the eye tracker measurements; a high threshold, on the other hand, might forfeit information from the beginning and end of saccades, where velocity is still accelerating or decelerating, respectively. Therefore, we labelled saccadic samples in a two-step procedure. To initialize search for a saccade onset, velocity had to exceed a relatively high threshold (138 deg/s) first. Then, going back in time, the first sample was searched where velocity exceeded a lower threshold θ_{off} (17 deg/s) that is biologically more plausible but less robust to noise (both parameters were determined by comparing detection results with a hand-labelled subset of our data). In a similar fashion, saccade offset was the first sample at which velocity fell below the lower threshold again. Finally, several tests of biological plausibility were carried out to ensure that

impulse noise was not identified as a saccade: minimal and maximal saccade duration (15 and 160 ms, respectively) and average and maximum velocity (17 and 1030 deg/s, respectively).

Determining fixation periods is particularly difficult for recordings made on dynamic stimuli (Munn et al., 2008). Smooth pursuit eye movements cannot occur on static images and are hard to distinguish from fixations because of their relatively low velocity of up to tens of degrees per second; but even a small, noise-induced displacement in the gaze measurement of just one pixel from one sample to the next already corresponds to about nine degrees per second. However, manual labelling of fixations is not feasible on such large data sets as that of experiment one (about 40000 fixations); we therefore used a hybrid velocity- and dispersion-based approach (Salvucci and Goldberg, 2000) and validated its parameters on a smaller data set of hand-labelled fixations. After saccade detection, the intrasaccadic samples were extracted. Here, a sliding window of at least 100 ms was moved across the samples until a fixation was detected. This minimum duration of 100 ms ensured that very brief stationary phases in the gaze data were not labelled as fixations. Then, this fixation window was extended until either one of two conditions was met: the maximum distance of any sample in the window to the centre of the fixation window exceeded 0.35 deg (this threshold was gradually increased to 0.55 deg with longer fixation duration); or the average velocity from beginning to end of the window exceeded five degrees per second. The latter condition served to distinguish pursuit-like motion from noise where sample-to-sample velocities might be high, but velocities integrated over longer time intervals are low because the direction of gaze displacements is random.

Eye movement similarity A variety of methods has been proposed in the literature to assess the consistency of eye movements across different observers. The fundamental problem is that there is no obvious metric for eye movement similarity since there is no direct (known) mapping from eye position to its perceptual consequences. In practice, there is only a small probability that two observers will fixate exactly the same location at exactly the same time; small spatio-temporal distances between eye positions, however, might have been introduced in the measurement only by fixational instability and the limited eye tracker accuracy, and are thus of little practical relevance. For larger distances of more than about one degree and a few tens of milliseconds, on the other hand, it is not clear how a similarity metric should scale: is a fixation twice as far also twice as different? How about two fixations to the same location, but of different duration? In the case of our (moving) stimuli, a further problem arises that looking at the same image region at different points in time, e.g.

in the background of the scene, might carry a different notion depending on what is (or is not) occurring elsewhere, e.g. in the foreground. As pointed out by Tatler et al. (2005), a good similarity metric should be robust to extreme outliers and sensitive not only to location differences, but also to differences in the probability of such locations; if all but one of the subjects looked at the same location A and the remaining subject looked at location B, this should be reflected as more coherent than an even distribution of fixations over A and B. Additionally, hard thresholds should be avoided in order to deal with the inherent spatio-temporal uncertainty in the eye tracker measurements. Finally, an ideal metric would yield an intuitively interpretable result and allow for fine-grained distinctions.

We will now discuss similarity metrics proposed in the literature according to the above criteria and then describe our modification of the Normalized Scanpath Saliency method that will be used in the remainder of this paper.

Several authors have used clustering algorithms to group fixations and then determined what percentage of fixations fell into the main cluster, or how large an image region must be to contain the gaze traces of a certain number of observers (Stelmach et al., 1991; Osberger and Rohaly, 2001; Goldstein et al., 2007). Obviously, these measures yield very intuitive values and are also robust to outliers. However, they might be sensitive to cluster initialization, and even if they were extended to regard the fixations in several clusters, they cannot capture differences in the distribution of fixations across several locations. Furthermore, a fixation can either be counted as inside the cluster or not, which means that a small spatial displacement can have a significant impact on the result. Some clustering algorithms introduce a certain smoothness to overcome this problem, e.g. mean-shift clustering (Santella and DeCarlo, 2004), but the scale of the resulting cluster becomes unpredictable, so that for densely distributed data, even two fixations that are very far apart might be classified as similar.

Another popular approach is to assign a set of letters to image regions and to create a string where the i-th letter corresponds to the location of fixation i. The resulting strings can then be compared by string editing algorithms, which sum penalties for every letter mismatch or other string dissimilarity such as letter insertions or transpositions. Drawbacks of this method are the need for an a priori definition of regions of interest for the string alphabet and of a penalty table; inherently, it cannot distinguish between fixations of different duration. Nevertheless, the string-editing approach has been used successfully on line drawings (Noton and Stark, 1971) and on semi-realistic dynamic natural scenes (Blackmon et al., 1999), and has been extended to handle the case where the order of fixated regions matters (Clauss et al., 2004).

Mannan et al. (1996) developed a measure to compare two sets of fixations by summing up the distances between the closest pairs of fixations from both sets. This is problematic because the result is dominated by outliers and probability distribution differences are not accounted for.

Hasson et al. (2008b) cross-correlated horizontal and vertical eye trace components of observers across two presentations of the same movie. The intuitive range of the measure is from minus one for highly dissimilar scanpaths to one for exactly the same scanpaths, with zero indicating no correlation between the traces. However, similarity here is defined relative to the mean position of the eye (which usually also is roughly the centre of the screen, see below); this means that two scanpaths oscillating between two fixations in counterphase, i.e. $ABAB\ldots$ and $BABA\ldots$ will always be classified as very dissimilar, regardless of the actual distance between A and B.

Another class of methods operates on so-called fixation maps or probability distributions created by the additive superposition of Gaussians, each centred at one fixation location $\vec{x} = (x, y)$ (to obtain a probability distribution function, a subsequent normalization step is required so that the sum of probabilities over the fixation map equals one). The inherent smoothness of the Gaussians offers the advantage that two fixations at exactly the same location will sum up to a higher value than two closely-spaced fixations, whereas very distant fixations will contribute only very little to their respective probabilities. This means that noise both in the visual system and the measurement has only a small impact on the final result; by definition, these methods also are sensitive to location distribution differences. There now are various possibilities to assess the similarity of two fixation maps, which includes both the comparison of two different groups of observers and the comparison of just one observer to another. Since in practice, fixation maps can only be created for a finite set of locations anyway, the most straightforward difference metric is the sum over a squared pointwise subtraction of two maps (Wooding, 2002a); Pomplun et al. (1996) have computed the angle between the vectors formed by a linearization of the two-dimensional fixation maps. In the latter study, fixations were also weighted with their duration, a modification that in principle could also be applied to the other fixation map-based measures as well.

An approach based in information theory, the Kullback-Leibler Divergence, was chosen by Rajashekar et al. (2004) and Tatler et al. (2005). This measure, which strictly speaking is not a distance metric and needs minor modifications to fulfill metric requirements (Rajashekar et al., 2004), specifies the information one distribution provides given knowledge of the second distribution. The KLD matches all of the above criteria for a good similarity measure with the possible exception of intuitiveness: identical distributions have a KLD of zero,

but the interpretation of the (theoretically unbounded) result for non-identical distributions is not straightforward.

For this reason, we use the Normalized Scanpath Saliency (NSS) measure as proposed by Peters et al. (2005). Originally, this measure has been developed to evaluate how closely artificial saliency models match human gaze data, but NSS directly can be applied to assess inter-subject variability as well. The underlying idea is to construct a fixation map by superposition of Gaussians as above, but with a different normalization scheme: mean intensity is subtracted and the resulting distribution is scaled to unit standard deviation. This has the effect that a random sampling of locations in the NSS map has an expected value of zero, with positive values resulting from fixated locations and negative values from non-fixated regions. To evaluate the similarity of eye movements of multiple observers, it is possible to use a standard method from machine learning, "leave one out". For each observer A, the scanpaths of all other observers are used to create the NSS map; the values of this NSS map are then summed up over all fixations made by A. If A tends to look at regions that were fixated by the other observers, the sum will be positive; for essentially uncorrelated gaze patterns, this value will be zero and it will be negative for very dissimilar eye movements. NSS has been used on videos before (Marat et al., 2009), but only on a frame-by-frame basis, similar to the analysis of static images by Peters et al. (2005). To achieve temporal smoothing, so that slightly shifted fixation onsets are not considered to be dissimilar by a hard cut-off, we extended NSS to the three-dimensional case.

Formally, for each movie and observers $i = 1, \ldots, N$, M_i gaze positions $\vec{x}_i^j = (x, y, t)$ were obtained, $j = 1, \ldots, M_i$. Then, for each $\vec{x}_i^j$ of the training set of observers $S = \{1, \ldots, k-1, k+1, \ldots, N\}$, a spatio-temporal Gaussian centred around $\vec{x}_i^j$ was placed in a spatio-temporal fixation map F,

$$F(\vec{x}) = \sum_{i \in S} \sum_{j=1}^{M_i} G_i^j(\vec{x}),$$

with

$$G_i^j(\vec{x}) = \mathrm{e}^{-\frac{\left(\vec{x}-\vec{x}_i^j\right)^2}{2\left(\sigma_x^2+\sigma_y^2+\sigma_t^2\right)}}.$$

This fixation map F was subsequently normalized to zero mean and unit standard deviation to compute an NSS map N,

$$N(\vec{x}) = \frac{F(\vec{x}) - \overline{F(\vec{x})}}{\mathrm{Std}(F)}.$$

Finally, the NSS score was evaluated as the sum of the NSS map values at the gaze samples of test observer k,

$$\mathrm{NSS} = \sum_{j=1}^{M_k} N(\vec{x}_k^j),$$

and this was repeated for all possible training sets (i.e. N times with N different test subjects).

The spatio-temporal Gaussian G had parameters $\sigma_x = \sigma_y = 1.2\,\mathrm{deg}$, $\sigma_t = 26.25\,\mathrm{ms}$. To evaluate gaze variability over the 20 s time course of the videos, NSS was not computed on the whole movie at once, but on temporal windows of 225 ms length that were moved forward by 25 ms every step. These parameters were varied systematically with qualitatively similar results. Because NSS is sensitive to the size of the Gaussian G, all results that are presented in the following were normalized with the inverse of the NSS of a single Gaussian.

Gaze positions $\vec{x}$ here refer to the raw gaze samples provided by the eye tracker except for those samples that were labelled as part of a saccade. Because visual processing is greatly reduced during saccades, these saccadic samples are of no practical relevance for the present analysis. In principle, the fixation spots could have been used instead of the raw samples as well, which would have significantly reduced the computational cost of this analysis; however, this might have biased results during episodes of pursuit, where automatic fixation detection algorithms still have problems and potentially ascribe fixations to random positions on the pursuit trajectory. Indeed, it was those movie parts in which many subjects made pursuit eye movements where we found eye movements to be particularly coherent. Furthermore, using the raw data allows for a distinction of different fixation durations; two fixations to the same location, but with varying duration will be classified as less similar than two fixations of identical length (given they take place at similar points in time).

In theory, this measure is independent of the number of training samples because it normalizes the training distribution to unit standard deviation. In practice, however, small training set sizes may lead to quantization artefacts; where applicable, we therefore matched the number of training samples when comparing two conditions. This was particularly important for the comparison of "local" and "repetitive", because in the latter condition each scanpath had to be evaluated in terms of a maximum of only four other scanpaths (the stimuli were repeated five times per day). A further consequence is that in the following, different absolute NSS values are occasionally reported for the same condition (but in the context of different comparisons).

Finally, we ran a comparison of the NSS measure with the Kullback-Leibler Divergence to exclude the possibility that our results might underlie some methodological bias. We did not want to compare two probability functions (that of the measured eye movement data and a saliency map), but we wanted to assess how systematic the subjects' eye movements were. We achieved this by calculating the divergence of the measured data D from a random, uniform model M, which is the difference of the cross entropy between D and M and the entropy of D:

$$\begin{aligned} KLD(D,M) &= H(D,M) - H(D) \\ &= -\sum^{\vec{x}} D(\vec{x}) \cdot \log(M(\vec{x})) - \sum^{\vec{x}} -D(\vec{x}) \cdot \log(D(\vec{x})) \\ &= -\log\left(\frac{1}{N}\right) + \sum^{\vec{x}} -D(\vec{x}) \cdot \log(D(\vec{x})) \end{aligned}$$

Even though the NSS analysis yields a more intuitive absolute score, NSS and KLD differ only slightly in their relative results. We computed both NSS and KLD scores over time for all movies in the "local" condition and found that they are highly correlated ($r = 0.87$, s.d. 0.05), i.e. both methods approximately mark eye movements on the same video parts as coherent or incoherent, respectively.

3.4 Results

Saccadic amplitudes and fixation durations The distribution of saccadic amplitudes for natural movies and for the other stimulus types is shown in Figures 3.3 and 3.4(a), respectively. On natural movies, saccadic amplitudes follow a skewed distribution with a mean of 7.4 deg and a median of 5.6 deg. Looking at the shape of the empirical cumulative distribution function (ECDF) in comparison to that of the other stimulus types, the ECDF for natural movies rises quickly, but saturates late. This means that observers tend to make both more small and more large saccades (with amplitudes of less than five and more than 10 degrees, respectively) on natural movies, whereas saccades of intermediate amplitudes are less frequent than in the other conditions. In contrast to this, the saccades on Hollywood trailers show the smallest fraction of large amplitudes, e.g. only 7.8% have an amplitude of 12 deg or more (natural movies: 18.2%). Spatio-temporal noise movies with a natural amplitude spectrum elicited fewer small saccades than all other stimuli types (median 6.5 deg; the mode of the am-

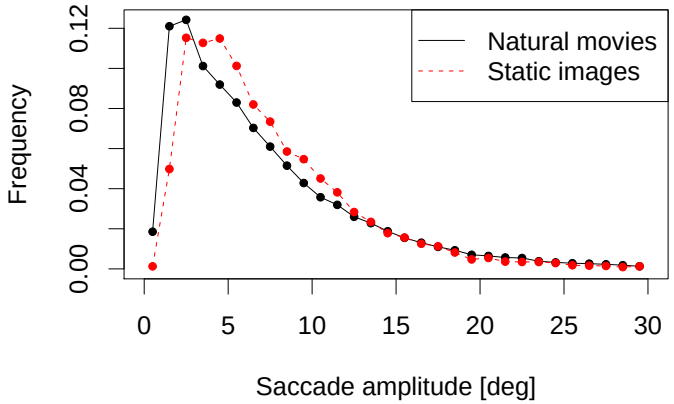


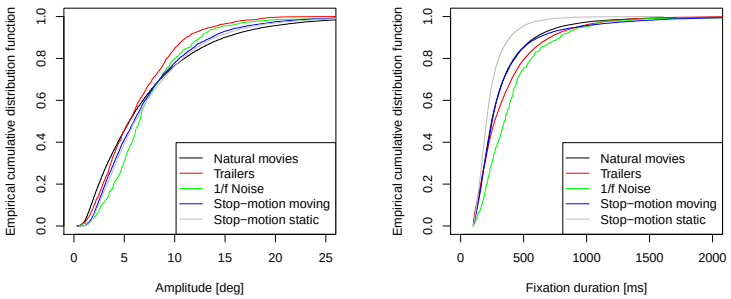
Figure 3.3: *Distribution of saccadic amplitudes on natural movies and static images. Saccades of medium amplitude (4–12 deg) are more frequent in the static images condition, whereas saccades on natural movies have small amplitude (up to four degrees) more often.*

plitude distribution lies at seven degrees compared to three degrees for natural movies) and more large saccades than the Hollywood trailers only. All the conditions differ from each other highly significantly (Kolmogorov-Smirnov test, $p < 10^{-10}$), with the only exception that the difference of saccadic amplitude distributions between stop-motion movies and static images is only weakly significant ($p < 0.027$).

Fixation durations are depicted in Figure 3.4(b). We here also find stimulus type-specific effects. Similar to saccadic amplitudes, the distributions are heavily skewed, which is reflected in a pronounced difference of mean and median values (for natural movies, 326 and 247 ms, respectively). The longest average fixation duration (mean 443, median 348 ms) can be found on the natural noise stimuli. Fixations on Hollywood trailers are much shorter, but still longer than on natural movies (mean 361.9, median 268.7 ms). The shortest fixations occur on static images (mean 239.8, median 205.8 ms).

All these differences are statistically significant (Kolmogorov-Smirnov test; natural and stop-motion movies $p < 0.021$, all other conditions $p < 10^{-10}$).

Centre bias of gaze and stimuli A well-documented property of human viewing behaviour is that observers preferentially look at the centre of the stimulus, the so-called centre bias (Tseng et al., 2009; Tatler, 2007; Parkhurst et al., 2002; Buswell, 1935). Especially on smaller displays, this stands to reason since the centre of the screen is the most informative location: because of the decline in



(a) (b)

Figure 3.4: *Empirical cumulative distribution functions (ECDF) of saccadic amplitudes (a) and fixation durations (b) for the different movie types. Natural movies elicit a higher number of either small or large (but not intermediate) saccades and relatively short fixations; natural noise movies and trailers elicit longer fixations.*

peripheral acuity of the retina, a fixation to one side of the screen will lead to an even lower resolution on the opposite side of the display. Because at least a coarse "snapshot" of the scene is particularly important during the first few, exploratory fixations, the central bias is strongest directly after stimulus onset (Tatler, 2007). In Figure 3.5, density estimates are shown for the different stimulus categories. Clearly, eye movements on the Hollywood trailers are the most centred; here, the densest 10% of screen area (15.2 by 8.5 deg) contain about 76% of all fixations, whereas for natural movies this number is only 30% (and 62% of fixations fall into the densest 30% of the screen). For natural noise and stop-motion movies, the centre bias is slightly stronger than for natural movies again (42 and 37%, respectively; 73% in the densest 30% for both conditions). In the latter case, fixations are redrawn to the centre at every new frame onset; in the former case, maintaining gaze close to the centre is a useful strategy because the central location will be most informative when the natural movie appears again in the oscillating noise movies.

A further common explanation for the centre bias of fixations is that there usually is a bias already in the stimuli because photographers (consciously or subconsciously) place objects of interest in the image centre. When recording the natural movies, no particular care was taken to avoid such central bias; on the contrary, the goal was to record image sequences "from a human standpoint" to fulfil a common definition of natural scenes (Henderson and Ferreira, 2004), which ruled out any truly random sampling. To assess the magnitude of this potential bias, the spatial distribution of image features was computed (see Figure 3.6). The feature used here is the geometrical invariant K, which denotes

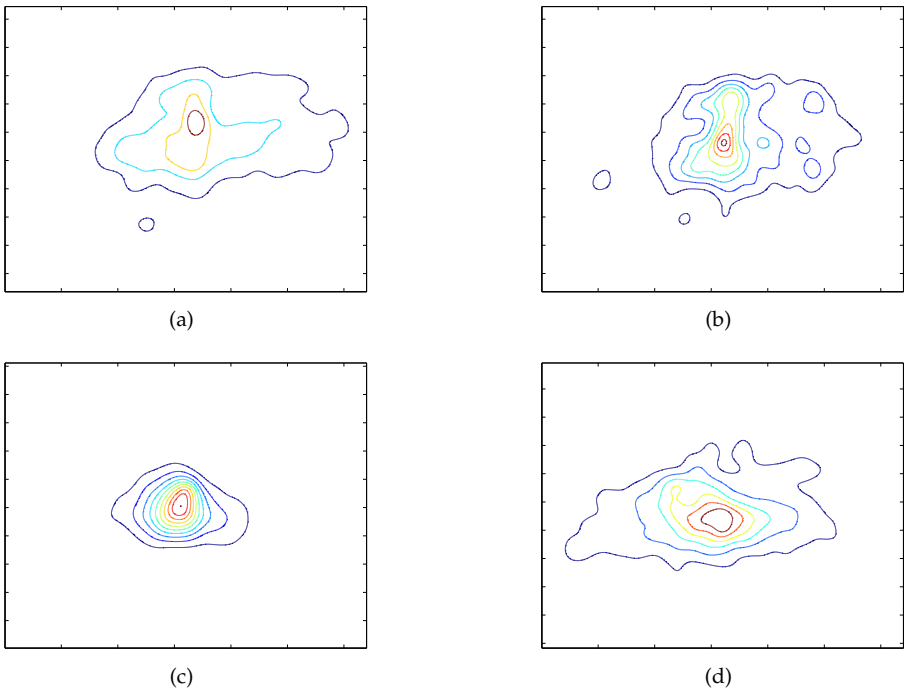
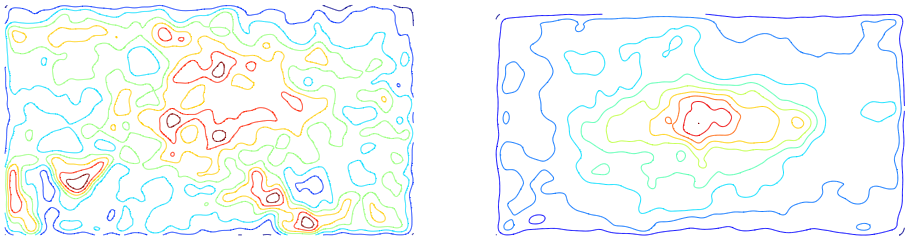


Figure 3.5: *Distribution of gaze in the different conditions, averaged over all movies and subjects. (a) Natural movies. (b) Stop-motion movies. (c) Hollywood trailers. (d) Noise movies. Probability maps were computed for each condition by the superposition of Gaussians ($\sigma = 0.96\,deg$) at each gaze sample and subsequent normalization; shown here are contour lines. The distribution of gaze on Hollywood trailers (c) is clearly more centred than in the other conditions. Gaze on natural movies (a) has the widest distribution; in the other conditions, frequent reorienting saccades to the centre are elicited by scene cuts (trailers), frame onsets (stop-motion), or "reappearance"/onset of structure from noise.*

those image regions that change in three spatio-temporal directions, i.e. transient corners; this feature has been shown to be predictive of eye movements (Vig et al., 2009). Even for the natural movies, there is a certain predominance of central features, but this effect is particularly strong for the Hollywood trailers (in fact, Figure 3.6 still underestimates the central bias because the frequent scene cuts introduce globally homogeneous temporal transients). It is worth pointing out that the fixation distribution for Hollywood trailers also reflects this central feature distribution; nevertheless, this does not necessarily imply a causal connection. Indeed, Tatler (2007) found that the centre bias of fixations on natural static images was independent of spatial shifts in the underlying feature distributions.

Variability of eye movements on natural videos After these general observations, we will now present results on the variability of eye movements. We



(a) (b)

Figure 3.6: *Distribution of spatio-temporal structure for natural movies (a) and Hollywood trailers (b). Shown here is the average spatial distribution of intrinsically three-dimensional regions as measured by the structure tensor, i.e. transient or non-rigidly moving corners, which have been shown to be highly predictive of eye movements (Vig et al., 2009). The trailers show a stronger bias for placing structure in the centre.*

start out with the variability across different observers watching the same natural movie for a single presentation of the stimulus (which Stark coined the "local" condition), see Figure 3.7. Shown here are one example where variability is very high, one example where most observers look at the same region at least temporarily, and data for one Hollywood movie trailer. Common to all movies is that variability is relatively low (coherence, as shown in the figures, is high) during the first one to two seconds due to the central bias of the first few saccades. After this initial phase, gaze patterns for the movie "roundabout" diverge and remain relatively incoherent until the end of the movie; this is not surprising since the scene is composed of a crowded roundabout seen from an elevated viewpoint, i.e. moving objects (cars, pedestrians, cyclists) are distributed almost uniformly across the screen. Nevertheless, gaze patterns are still more similar than the random baseline of different observers looking at different movies (what Stark coined the "global" condition, which models stimulus- and subject-independent effects such as the central bias; mean NSS for "roundabout" 0.45, for "global" 0.18, $p < 10^{-10}$). NSS for the movie "ducks_boat" is shown by the peaked curve in Figure 3.7. The overall scene is fairly static with two boats moored on a canal, but no humans or moving objects (see Figure 3.1). At about the 5 s mark, a bird flies by, followed by another bird at 10 s; both these events make virtually all observers look at the same location (max NSS 2.61, mean 0.84). For a comparison, NSS for the trailer "War of the Worlds" is also plotted and exhibits several such highly coherent peaks; on average, gaze on trailers is significantly more coherent than on natural movies (1.37 vs. 0.72, $p < 10^{-10}$).

A further prediction by the scanpath theory is that "idiosyncratic" viewing behaviour should be less variable than the "global" condition, i.e. the eye movements of one person watching different movies should be more coherent

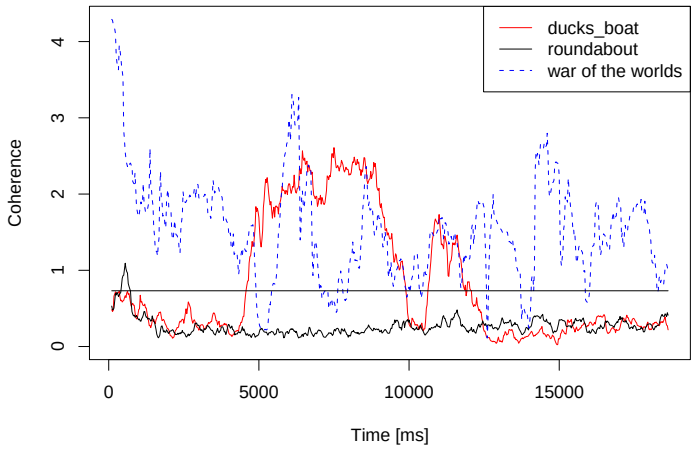


Figure 3.7: *Normalized Scanpath Saliency on natural movies: when a flock of birds flies by (from 5–10 s, some more birds follow 11–13 s), almost all observers orient their attention to the same spot (red line); in the "roundabout" video with small, moving objects evenly distributed across the scene, eye movements are highly variable and thus have a low coherence (black line). For comparison, the horizontal line denotes the average across all natural movies; the much higher coherence for one Hollywood trailer is also shown (dashed line).*

than those of different persons watching different movies. However, our data does not support this hypothesis; indeed, NSS for the idiosyncratic condition is even lower than for global (0.09 vs. 0.15).

Variability of eye movements on natural noise Figure 3.8 shows the NSS scores for the natural noise movies that oscillate between natural scenes and noise while the same spatio-temporal amplitude spectrum is maintained. One interesting observation is that even in those episodes where no discernible natural structure is visible (noise level $> 95\%$), eye movements are still significantly more coherent than in the global baseline condition (mean NSS for noise 0.49, for global 0.18; Kolmogorov-Smirnov test $p < 10^{-10}$). Yet, episodes with low noise level, i.e. more semantically meaningful content, have higher coherence; noise level and NSS are thus anti-correlated ($r = -0.57$). Also notable is the observers' behaviour when the natural scene re-appears, i.e. along the downward slopes of the noise level curve: during a slow change, eye movement coherence goes up only slightly (at around $t = 25$ s); faster transitions (at around $t_1 = 50$ s, $t_2 = 60$ s) seem to elicit a re-orientation response similar to abrupt frame onsets (since the underlying natural movie did not change during the noise oscillations, observers should have been able to predict the layout of the natural scene following a noise episode).

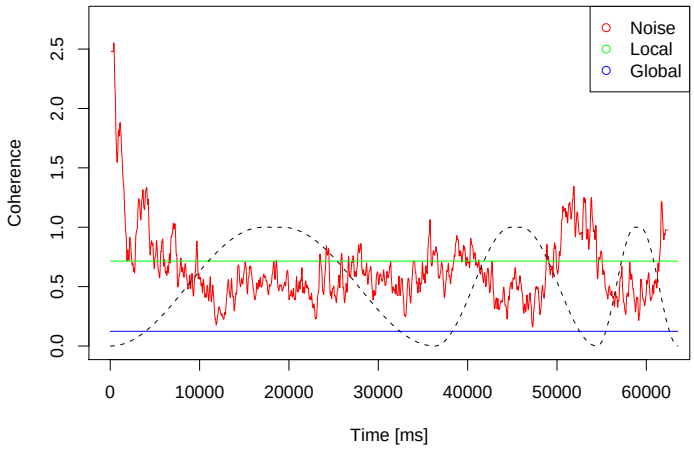


Figure 3.8: *NSS values averaged over the "natural noise" movies. The dashed line denotes the noise level, i.e. peaks in the sinusoidal curves correspond to phases where subjects only saw noise and minima correspond to phases where the original movie was fully visible. Eye movement coherence is inversely correlated with noise level ($r = -0.57$), i.e. when natural image structure is visible, eye movements are more coherent than on pure noise. However, even in phases of pure noises, the distribution of gaze is not random: NSS scores are higher than the "global" baseline of natural movies (lower horizontal line). The green horizontal line denotes the average coherence on natural movies ("local" condition).*

Variability of eye movements on stop-motion movies Figure 3.9 shows the average NSS for the stop-motion movies and for the matched set of natural movies (only nine out of the 18 natural movies were shown in a stop-motion version), with dashed vertical lines denoting the onset of new stop-motion frames. Inter-subject coherence spikes after every frame onset to above the NSS score on the continuous movies; after about one to two seconds, however, variability increases and the NSS score drops below that of the continuous case. This observation is statistically significant when pooling the first and second halves of the 3 s frame intervals: initially, stop-motion NSS is higher than local NSS (paired Wilcoxon's signed rank test, $p < 10{-}10$); in the second half, this relationship is reversed ($p < 0.007$).

Variability increases with repetitive viewing of the same stimulus Several studies have found that repetitive presentation of the same stimulus leads to similar scanpaths (on static images, Hasson et al., 2008b; Foulsham and Underwood, 2008; for simple artificial dynamic scenes, Blackmon et al., 1999). Results from experiment two confirm these earlier findings; indeed intra-subject variability is lower than inter-subject variability (mean NSS for repetitive 0.66,

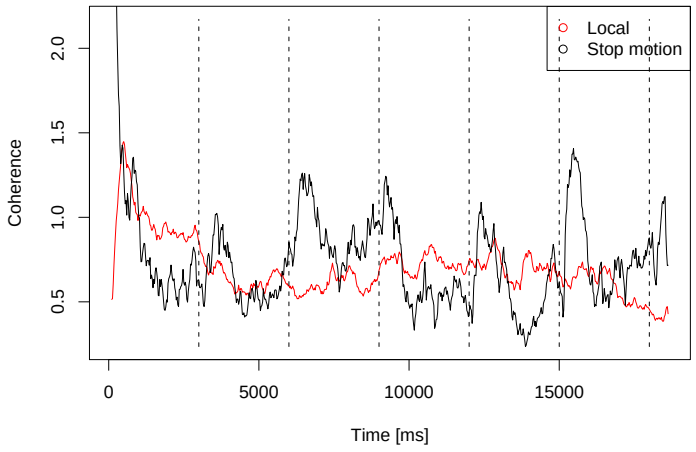
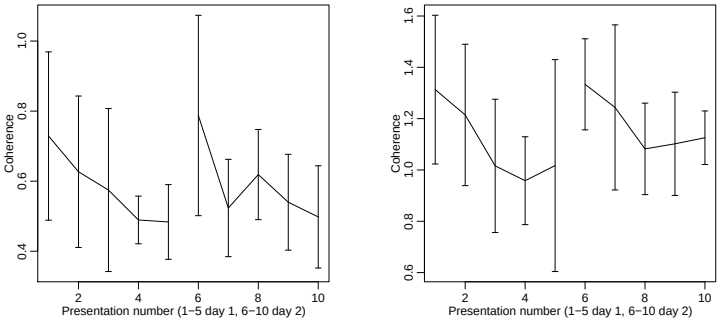


Figure 3.9: *Eye movement coherence on the same set of movies for continuous display (local condition) and for the stop-motion condition, where one frame is shown every three seconds. In the stop-motion condition, coherence spikes after each frame transition and then drops again steeply until the next frame onset. This demonstrates a systematic difference in gaze behaviour on static and dynamic stimuli.*

local 0.47, and 1.41/0.88 for trailers; Kolmogorov-Smirnov test $p < 10^{-10}$; the local score here is smaller than above because of the matched sample size, see Methods). One possible confound is that when recording eye movements from one subject in one session, calibration inaccuracies might not be independent across trials, i.e. eye movement coherence might be overestimated; we therefore compared one subject's scanpaths only with scanpaths from the other day of data collection (and indeed found that failure to do so resulted in an even higher increase in eye movement coherence than above). However, pooling together up to five repetitions of a movie also may underestimate how similar gaze patterns evoked by the same stimulus are: the variability of the individual presentations i.e. for the first, second, ... presentation is shown in Figure 3.10. With increasing number of repetitions, the variability of eye movements across subjects increased ($p < 10^{-4}$ for natural movies, $p < 0.002$ for trailers, paired Wilcoxon's test). Because the bottom-up stimulus properties were kept constant by definition, this means that individual viewing strategies had an increasing influence. Interestingly, though, this effect was reversed when the stimuli were presented again the following day. The first presentation on the second day (presentation 6 in Figure 3.10) led to a coherence across subjects comparable to that of the very first presentation (on day one); for subsequent presentations, coherence declined again.



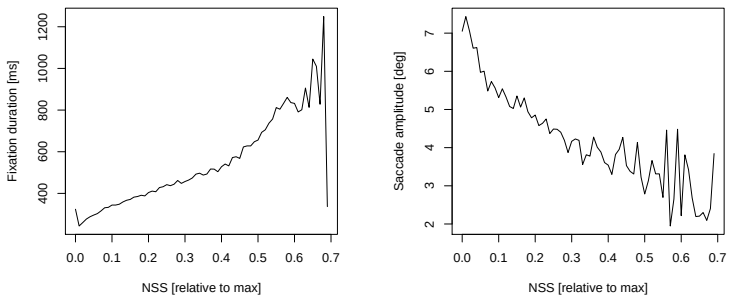
(a) (b)

Figure 3.10: *Evolution of coherence during repeated presentation of the same stimulus on natural movies (a) and Hollywood trailers (b). Each movie was presented five times in one session (trials 1–5) and another five times the following day (trials 6–10). Later presentations at the same day are significantly more variable (paired Wilcoxon's test, $p < 0.002$), but coherence is comparable between both days. Thus, it is not stimulus familiarity per se that drives variability, but an experimental artefact (subjects lost interest).*

Correlation of basic eye movement parameters with variability/hotspots Finally, we investigated whether the fixations at locations with high observer similarity, or hotspots, are different from random fixations. Figure 3.11 shows fixation duration and amplitude of the saccade preceding that fixation as a function of NSS at fixation (relative to the maximum NSS over all movies; because of the small sample size for larger values, the range of NSS is clipped at 70% of the maximum). Locations with high coherence, i.e. locations that were looked at by many observers simultaneously, were examined with fixations of longer duration compared to random locations; also, observers tended to make small saccades towards such highly coherent locations. In other words, the image regions that attract attention by a number of people also attract the attention of individual observers for longer and more small, object-investigating saccades.

3.5 Discussion

In this chapter, we have collected a large set of eye movements on natural movies and on several other stimulus types. To investigate the role of temporal change in dynamic stimuli, we used stop-motion stimuli that have the same semantic content as the natural movies, but lack continuous motion. We also probed the limits of the influence of semantics on eye movements: in the one extreme, eye movements were recorded on noise movies that had similar low-level features (spatio-temporal amplitude spectrum) as the natural movies, but lacked any semantic content; in the other extreme, we used trailers



(a) (b)

Figure 3.11: *(a) Correlation of NSS values on natural movies and fixation duration. "Hot spots", where many subjects look simultaneously and NSS is high, are fixated for longer periods of time (*$9.4\,\mathrm{ms}/\%$, $R^2 = 0.79$*). (b) Correlation of NSS values and saccadic amplitudes. Saccades towards hot spots are typically of small amplitude (*$-0.05\,\mathrm{deg}/\%$, $R^2 = 0.81$*).*

for Hollywood action movies where both low-level features and semantically meaningful objects were deliberately arranged in order to guide the viewer's attention. We found systematic differences throughout these different stimulus types, and will now discuss each of these findings in more detail.

General eye movement parameters Saccadic amplitude and fixation duration are two well-studied, basic eye movement parameters. In line with earlier findings, saccadic amplitudes on natural stimuli follow a heavily skewed distribution biased towards short amplitudes, with a long tail of relatively rare saccades of larger amplitude. In a review of several studies, von Wartburg et al. (2007) found that mean saccadic amplitude scales linearly with stimulus size; the largest natural stimuli reported had an extent of 34 by 26 deg and resulted in a mean saccadic amplitude of 6.3 deg (median 5.2 deg). In contrast to this, we measured only slightly larger saccades (mean 7.4, median 5.6 deg) on more than 30% larger stimuli (image extent of our videos 48 by 27 deg). However, this probably can be explained by the fact that there are obvious mechanical limits to the range of eye movements: under natural viewing conditions, saccades typically are accompanied by a head movement (Einhäuser et al., 2007), but in the present experiments, these were suppressed by a chin rest. When comparing the distributions of saccadic amplitudes across the different stimulus types, eye movements on natural movies comprised of more either small or large saccades, with less saccades of intermediate amplitude than in the other conditions. Apparently, viewing behaviour on natural movies can be characterized by occasional larger jumps between clusters of interesting

objects, which are then examined in detail by small, intra-object saccades. On Hollywood trailers, the smallest fraction of large amplitudes was observed; here, the producers deliberately capture the viewer's attention in the centre of the screen, using special effects such as explosions, tracking shots, etc., so that there is little incentive for large saccades towards the periphery. This was also reflected in the fact that saccades on this type of movie showed the highest centre bias. Spatio-temporal noise with a natural amplitude spectrum also elicited few large saccades, but also very few small saccades. This different viewing behaviour makes sense since the noise lacks semantic objects that might need to be examined in detail; because spatio-temporal structure is evenly distributed across the whole stimulus, there also is little stimulus-driven incentive to make (large) eye movements towards the periphery. Nevertheless, these results are at odds with a study by Jansen et al. (2009), who had found that – for static images – saccade lengths on natural and pink noise images were comparable. We see two differences in the used stimuli that could explain this difference: for once, the introduction of temporal changes obviously might have an impact on eye movements, and this impact might differ for semantically meaningful and meaningless stimuli, even if their amplitude spectra are matched; furthermore, different falloff parameters β during creation of the $1/f^{\beta}$ noise might lead to a different predominant scale in the stimuli, which in turn might lead to different saccadic amplitudes.

That fixation duration varies with task is a well-established fact (Tatler et al., 2006; Loschky et al., 2005; Canosa, 2009). We found the longest fixation durations on average on the natural noise movies, where subjects might not feel the need to quickly explore the scene since there are no discernible objects to examine. The shortest fixation durations were found on static images and possibly can be explained by an artefact of the experimental setup: the short presentation time of static images puts pressure on the subjects to quickly scan the image before it disappears again.

Summarizing, the exact stimulus type has a profound impact on saccadic amplitudes. Specifically, observers tend to make more saccades of intermediate amplitude and shorter fixations on static images than on image sequences.

Centre bias Our data replicated the well-studied phenomenon that subjects preferentially look at the centre of the screen (Tseng et al., 2009; Tatler, 2007; Parkhurst et al., 2002; Reinagel and Zador, 1999; Buswell, 1935); however, we could show that the effect varies with stimulus type. Not surprisingly, the strongest centre bias could be found on Hollywood trailers. Here, the photographer's bias to place interesting objects in the centre was deliberately employed, which is also reflected in the distribution of low-level spatio-temporal structure

that is skewed towards the centre much more than the natural movies. Furthermore, the frequent scene cuts also contributed to the centre bias; the abrupt frame transitions in the stop-motion condition led to a stronger centre bias than on natural movies.

Variability Not surprisingly, we found that eye movements of several observers on one natural movie are less variable than eye movements on different movies; in other words, that eye movements are at least partially determined by the visual input. This effect was even stronger for professionally cut Hollywood trailers.

It is a well-established fact that the consistency in fixation locations between observers decreases with prolonged viewing (Tatler et al., 2005). We here found a systematic difference in viewing behaviour on dynamic, i.e. more natural stimuli compared to static images predominantly used in eye movement studies. Whereas the first few fixations on the static images used in our stop-motion movies are heavily influenced by stimulus onset and drawn towards the centre of the stimulus, viewing after as little as 1.5 s becomes unnatural and idiosyncratic in the absence of continuous temporal change. 't Hart et al. (2009) found a similar result in a recent study. They presented their subjects with a random sequence of stillshots from a set of videos for one second each; these stillshots lacked continuous motion and elicited more centred eye movements than the original videos. The time course of such gaze behaviour led the authors to interpret this finding as a dominance of stimulus onset effects. We here confirm this finding for longer presentation times (3 s instead of 1 s) and extend it to a case where scenes are highly predictable; even though the stillshots are taken from the same movie and presented in their correct chronological sequence, frame transitions still elicit reorienting responses towards the centre.

In line with the scanpath theory, we could also confirm earlier findings on very simple, synthetic scenes (Blackmon et al., 1999) that a single observer watching the same movie repetitively exhibits more coherent eye movements than several observers watching this movie. This effect has also been shown for two consecutive presentations of static images (Foulsham and Underwood, 2008). However, we found that variability increased with a growing number of presentations, but dropped back to its original level when subjects watched the movie for the first time in the second half of the experiment on the following day. Considering that subjects presumably were still familiar with the stimuli (they had seen them five times the previous day), this implies that the increase in variability is not due to stimulus familiarity per se, but rather an artefact of the experimental conditions (subjects lost interest).

A further hypothesis of the scanpath theory (Noton and Stark, 1971) is that idiosyncratic viewing behaviour exists, so that eye movements of a single person on different stimuli are more coherent than those of different persons looking at these stimuli. Our experimental data does not support this notion on complex dynamic scenes; indeed, variability of eye movements was slightly higher in the idiosyncratic than in the global condition.

On natural noise movies, eye movements were more coherent than the global baseline (which incorporates the centre bias) even during the episodes where no discernible objects were visible. This finding pertains to the ongoing debate to what extent eye movements are driven by low-level features, such as contrast and motion, or whether the preference of the oculomotor system for highly structured image regions is merely correlative in nature because these regions coincide with meaningful objects (for this point of view, see e.g. Einhäuser et al. (2008a); Foulsham and Underwood (2008); Elazary and Itti (2008), on the other hand, contend that the low-level structure gives rise to the perception of objects). We here found evidence for a causal contribution of low-level features to oculomotor control, at least on movies with similar spatio-temporal amplitude spectrum as that of natural movies.

Finally, we investigated the characteristics of eye movements at hot spots, i.e. at regions that were fixated simultaneously by several observers. Results show that fixations on these regions on average lasted longer and were the result of smaller saccades than on less-fixated regions; this effect was linear in the amount of fixations (NSS value).

3.6 Chapter conclusion

We have extended the study of variability of eye movements to the temporal domain and natural videos and measured basic eye movement parameters on a range of different stimulus categories. We investigated the variability introduced by the temporal dynamics of a stimulus using novel "stop-motion" stimuli and found that briefly presented static images, as used in common psychophysical paradigms, are a special case and not very representative of human viewing behaviour. Noise movies with a natural amplitude spectrum elicited more coherent eye movements than predicted by the central bias alone; this indicates that the low-level features of the noise attracted attention. Less surprisingly, professionally-cut Hollywood trailers evoked very similar eye movement patterns among observers. We also put to test the "scanpath theory" on natural videos and found that repetitive viewing of the same stimulus of the same observer elicited more coherent eye movements than single stimulus

presentations from different observers. However, we did not find evidence for idiosyncratic viewing patterns of the same subject across different movies.

"It's tough to make predictions, especially about the future."

Yogi Berra

4

Prediction of eye movements

In the previous chapter, we have seen that eye movements are far from random, but show systematic tendencies. There are certain oculomotor constraints, such as a bias towards the centre of the display and a relative overabundance of saccades with small amplitude, but these constraints do not suffice to explain why fixation positions on dynamic natural scenes often are so similar between observers. In this chapter, we are now going to explore how this similarity is related to low-level image features and how these, in turn, can be used to predict where people look. The development of a successful prediction algorithm has not only consequences for the understanding of human vision, but might also have technical applications, for example in active vision.

We shall first give a brief overview of the literature on the prediction of eye movements. Over the last decade, numerous studies have dealt with the relationship of eye movements and low-level image features; most of these studies, however, analysed data recorded on static stimuli, which are not very representative for natural viewing behaviour, as we have discussed in the previous chapter.

In the main part of this chapter, we shall present our own work that studies the relationship of a wide range of low-level image features at the centre of gaze with such features at potential saccade targets. This work had been motivated by a study (Dragoi and Sur, 2006) that showed – for monkeys watching still grayscale images – that scanpaths were systematically biased towards alternating between iso-oriented and orthogonal edges, avoiding intermediate orientation differences. Together with electrophysiological recordings on monkeys and psychophysical discrimination experiments in humans, Dragoi and Sur explained this bias in terms of neural adaptation. We tested their hypothesis for humans and videos, using the data set from the previous chapter, and did not only look at orientation, but also a variety of other image features. We also

contributed a methodological improvement to the design of a proper baseline condition; we developed an algorithm that transforms gaze data from a set of subjects to random scanpaths while leaving certain statistical characteristics intact. Using this improved baseline condition and advanced image processing algorithms, we could show that the finding by Dragoi and Sur is likely due to a methodological bias; this study has been published in (Dorr, Gegenfurtner, and Barth, 2009a).

We will then turn to the prediction of eye movements using machine learning techniques. Some attempts at learning from data those image structures that draw gaze have been made before; however, these attempts usually suffered from the curse of dimensionality because even small image patches quickly become computationally intractable. We here use a – in hindsight – simple trick to reduce the dimensionality of the data and achieve prediction rates that outperform much more complex, state-of-the-art models. Instead of learning on the raw image intensities, our classifier operates on a set of image features such as the geometric invariants; these have been used to predict eye movements before, albeit without machine learning (Böhme, Dorr, Krause, Martinetz, and Barth, 2006a).

This work was performed in close collaboration with Eleonóra Víg, who signs responsible for the machine learning algorithms and modifications to the software framework that was used to efficiently compute geometrical invariants, process the videos, etc. (see next chapter). Some results obtained with this approach have been published already (Vig, Dorr, and Barth, 2009), a further manuscript with more detailed analyses is in preparation.

To conclude this chapter, we will present some early results that extend gaze prediction to transparently overlaid videos; to the best of our knowledge, this topic has not been addressed in the literature so far. These experiments were carried out by Laura Pomarjanschi, who collected the data and used the software for blending movies on a spatio-temporal pyramid (see Chapter 2) and for computing the eigenvalues of the generalized structure tensor written by Michael Dorr, and the machine learning framework by Eleonóra Víg. A manuscript that includes first results is currently under submission (Barth, Dorr, Vig, Pomarjanschi, and Mota, 2010).

4.1 Bottom-up and top-down eye guidance

An influence of the task at hand on gaze behaviour was already found by Yarbus (1967), a finding that was corroborated also for real-life activities (Land and Hayhoe, 2001; Ballard and Hayhoe, 2009). Because of the complexity of modelling cognitive factors, however, much research has focused on bottom-up,

low-level factors that can be computed from the stimuli alone. This was further facilitated by the finding that the distribution of image features at the centre of fixation differs significantly from that at random control locations (Mannan et al., 1997; Reinagel and Zador, 1999; Parkhurst et al., 2002; Tatler et al., 2005; Baddeley and Tatler, 2006; Tatler et al., 2006), which can be interpreted as a preference of the human visual system for highly structured image regions (but see below). Over the past decade, much research has been done to exploit this preference and to develop algorithms for the prediction of gaze based on bottom-up features.

A common approach to model such low-level factors is that of a *saliency map*, which was first formulated for static images (Itti et al., 1998; Privitera and Stark, 2000; Itti and Koch, 2001; Itti, 2005). Canonically, a set of biologically inspired feature detectors, such as for contrast, colour, or orientation is assembled that assigns a certain relevance value for each feature under consideration to every location in the image. In the case of contrast, for example, it is intuitively plausible that a high contrast value should also be assigned a high relevance; for orientation, on the other hand, all possible values have the same a priori saliency. Therefore, centre-surround detectors are used so that e.g. a single horizontally oriented edge amongst vertical edges is assigned a high relevance. These feature maps are combined by a weighting scheme to obtain one saliency value per image location; a simple model might then always pick the image location with the maximum saliency value as the next saccade target.

Several modifications and additions to the original saliency map model have been made over the years, including inhibition-of-return (Itti and Koch, 2001), extension to the temporal domain (Carmi and Itti, 2006), feature map fusion schemes (Meur et al., 2006, 2007), or features that are based on information-theoretic considerations rather than on having a direct (known) neural correlate (Böhme et al., 2006a; Bruce and Tsotsos, 2006; Guo et al., 2008; Bruce and Tsotsos, 2009; Seo and Milanfar, 2009).

Other studies have analysed fixation locations in a Bayesian framework (Zhang et al., 2008, 2009); Itti and Baldi (2006) coined the term "surprise" (measured in "wows") for videos that expresses how much a pixel or region deviates from its expected value.

Some authors have looked directly at the distribution of feature values at fixated and non-fixated locations and used decision-theoretic methods to classify novel locations (Tatler et al., 2006; Gao and Vasconcelos, 2009).

Another interesting approach is to use machine learning techniques. Instead of an a priori definition of important features and their appropriate scales, machine learning should be able to distill the relevant image structure from a data set of attended image or movie locations. Judd et al. (2009), in a first step,

learned optimal parameters for the Itti and Koch saliency model. The work by Kienzle et al. went further and learned interest points directly on the pixel intensity values of static scenes (Kienzle et al., 2006, 2009) and on Hollywood movies (Kienzle et al., 2007). These studies, however, were limited to only one spatial scale and also suffered from the curse of dimensionality, because their feature vectors had a separate dimension for each pixel of a neighbourhood around fixation; even with a relatively small neighbourhood of e.g. 32 by 32 pixels, the classification problem becomes more than 1000-dimensional. In comparison, even a very large gaze data set such as the one presented in the previous chapter consists of about 40000 saccades only. Nevertheless, Kienzle et al. achieved prediction rates that were comparable to previous approaches (ROC score of 0.63 on static images; 0.58 on videos).

Despite these successes in predicting eye movements based on low-level features alone during free viewing, it has also been shown that task demands can overrule image-based saliency (Henderson et al., 2007; Einhäuser et al., 2008b). Some authors argue that it is not low-level features per se, but semantically meaningful objects (the presence of which is correlated with image structure) that drive attention (Foulsham and Underwood, 2008; Einhäuser et al., 2008a); however, it is also still under debate whether low-level features are merely correlated with objects or give rise to their perception (Elazary and Itti, 2008).

Finally, it should be noted that the majority of work on gaze prediction has dealt with static stimuli; only very recently, several studies have been published that used eye movements on videos (Carmi and Itti, 2006; Böhme et al., 2006a; Meur et al., 2007; Kienzle et al., 2007; Zhang et al., 2009; Bruce and Tsotsos, 2009).

4.2 Saccade target selection based on low-level features at fixation

In the previous section, we have discussed that eye movements are guided by both image-driven, bottom-up properties as well as cognitive, top-down processes; the relative importance of these two mechanisms is still under debate. Recently, Dragoi and Sur (2006) introduced a further mechanism that does not fall neatly in either category and rests on the relationship of low-level features at the current centre of gaze and low-level features at potential saccade targets. Because information about the observer, the current gaze position, needs to be taken into account, a pure bottom-up model does not suffice to describe this mechanism; on the other hand, the mechanism seems to work at a pre-attentive stage, so a description as top-down would also be inadequate. Dragoi

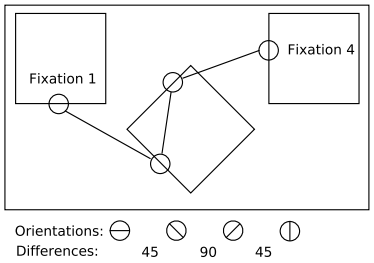


Figure 4.1: *Schematic illustration of the analysis for a synthetic scene; real data was measured on natural videos. Low-level features (here: orientation) are extracted from each fixated image patch and their differences along the scanpath are computed.*

and Sur based their work on measurements of rhesus monkeys watching still images. In this section, we will systematically investigate whether the proposed mechanism also can be found, for a variety of low-level features, in human observers watching videos, i.e. whether low-level features at the current centre of gaze contribute to saccade target selection under natural viewing conditions.

Correlations of low-level features at successive fixations

Based on psychophysical and electrophysiological evidence, a novel mechanism for the selection of saccade targets was put forward by Dragoi and Sur (2006). They showed that when V1 neurons were adapted to gratings of a certain orientation for 400 ms, subsequent discrimination performance improved for both iso-orientation and orthogonal gratings; discrimination of gratings with an intermediate orientation difference, on the other hand, did not change significantly. Dragoi and Sur (2006) related these findings to eye movement recordings from rhesus monkeys viewing still images that showed that fixations of an image patch were likely to be followed by either a small saccade to a patch with similar orientation or by a large saccade to a patch with largely dissimilar orientation. The proposed explanation was that eye movements exploit the improved discrimination performance and steer gaze towards either iso-oriented or orthogonally oriented image patches. A schematic illustration of this analysis can be found in Figure 4.1, which depicts a putative scanpath on a synthetic scene: from each fixation patch, dominant local orientation ϕ is extracted (e.g. $\phi_1 = 90\,\text{deg}$, $\phi_2 = 135\,\text{deg}$, etc.). The differences of orientation at successive fixations then can be computed (e.g. $\Delta\phi_1 = |\phi_2 - \phi_1| = 45\,\text{deg}$) and their distribution compared with a distribution of differences obtained on randomly generated control scanpaths. In the case of Dragoi and Sur, the distribution of differences in orientation was more U-shaped for measured than for random baseline scanpaths because both very small and very large differences occurred more often. Looking at these differences of low-level features can

also be interpreted as evaluating the correlation of such features introduced by the visual system's target selection process. At this point, however, it is important to note that natural scenes are highly correlated both in space and time (Zetzsche et al., 1993; Simoncelli, 1997); it is therefore crucial to carefully discriminate these image-inherent correlations from those that are due to eye movements.

If we found such eye movement–induced correlations indeed, we could also understand them as a contribution of low-level features at the current centre of fixation to the selection of the next saccade target. This is of particular interest to the prediction of eye movements: here, it were not sufficient anymore to look at a saliency map that is independent of current eye position. On the contrary, information from the current eye position would be required to determine where the eye will look next. Similar analyses of oculomotor tendencies such as saccadic amplitude and direction, fixation duration, and the bias towards the centre of the stimulus have shown that such factors can significantly improve feature-based models of eye guidance (Tatler and Vincent, 2008, 2009).

In the remainder of this section, we will apply the technique of looking at feature differences at successive fixations to our large set of eye movement data from human subjects watching high-resolution video clips that was presented in Chapter 3. To use video clips instead of still images has the advantage that viewing conditions are more natural; on still images, a few fixations might suffice to capture all relevant scene information, after which image sampling might become idiosyncratic.

To extend the analysis beyond that of Dragoi and Sur (2006), we did not only look at local orientation, but systematically investigated other low-level features as well. In particular, these were brightness, colour, and motion. Even though the choice of these features might be arbitrary to a certain extent, there seems to be a general consensus that these features are extracted at an early stage in visual information processing (Adelson and Bergen, 1991). Furthermore, we analysed the correlation of geometrical invariants (see Section 2.8), which are basic dynamic features from a computational perspective and have been shown to be useful in understanding various phenomena in biological vision (Zetzsche and Barth, 1990; Zetzsche et al., 1993; Barth and Watson, 2000). The invariant H can also be interpreted as spatio-temporal contrast.

Finally, our analysis was performed on a spatio-temporal multiresolution pyramid (see Section 2.3) in order to capture any effect that might be limited to a certain spatio-temporal scale.

Stimuli and gaze data

We used the data set of eye movements on high-resolution natural movies as presented in Chapter 3. Our algorithm for fixation detection is described in Section 3.3; however, the extraction of fixations made on dynamic scenes from raw eye movement data is not trivial due to the occurrence of smooth pursuit eye movements (Munn et al., 2008), and our investigation of successive fixations in this section obviously hinges crucially on a faithful detection of fixations. Therefore, we chose to implement a further fixation identification algorithm for comparison purposes, namely the GUIDe algorithm developed by Kumar (2007). Performance of both algorithms was validated against a randomly sampled set of 550 hand-labelled fixations; the GUIDe algorithm yielded a slightly better agreement and was therefore used for all results presented in this section. Nevertheless, we ran the same analyses using the second algorithm and obtained qualitatively similar results. For the GUIDe algorithm, we also computed the extent to which gaze samples remained unlabelled as either fixation or saccade, which might indicate a smooth pursuit movement. About 9% of gaze samples could not be labelled reliably; however, average duration of such unlabelled episodes was 37 ms, which would be fairly short for phases of smooth pursuit, so that it was possibly often rather the transitions between (high-velocity) saccades and (low-velocity) fixations that caused problems for the algorithm. Manual inspection further revealed that some clear episodes of smooth pursuit, e.g. when a flock of birds flies by in one of the videos, were broken into a series of fixations and 'undefined' samples. However, the depicted objects are not translated rigidly, change course, etc., so that even a manual labelling would be difficult. In the context of the present study, it is not clear at any rate how smooth pursuit should be treated, since e.g. catch-up saccades would keep fixation on the same object.

Low-level features

All low-level features were computed on a multiresolution pyramid constructed from the image sequence by successive blurring and sub-sampling in both the spatial and the temporal domain. In our implementation, we created five spatial (13.4, 6.7, 3.3, 1.7, and 0.8 cycles/deg) and three temporal (30, 15, 7.5 fps) scales. Except for colour, all features were determined on the luma channel (see below) of the video.

Timing of feature extraction with regard to fixation onset For each fixation, we extracted features from that video frame that was shown on the screen at the onset of fixation. The human visual system, however, has to base its

decision where to move the eyes next on information that was available earlier already because of its sensory-motor latency. Therefore, we additionally ran all analyses again with features that were extracted at up to 200 ms (in steps of 25 ms) before fixation onset, respectively; due to the temporal correlations in the videos, results were qualitatively similar (data not shown).

Orientation We extracted orientation as described in Section 2.10 by an eigenvalue analysis on the two-dimensional structure tensor, with an 11-tap binomial kernel for the lowpass filter ω. We already have discussed that for natural stimuli, eigenvalues of J are rarely exactly zero, and so confidence measures based on the relative size of the eigenvalues are needed to reliably detect oriented features, see Equation 2.8. For the present analysis, we systematically varied θ_1, θ_2 in the range 0.01–0.1 and 0.1–0.9, respectively.

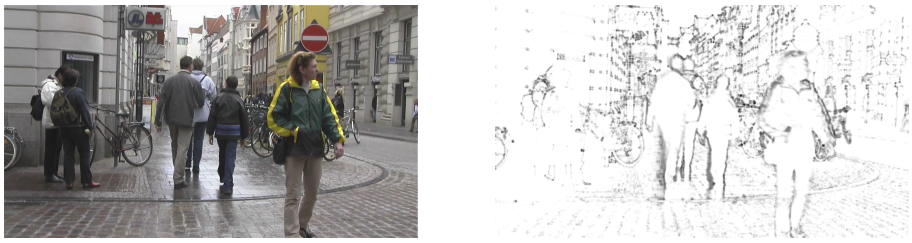
Colour MPEG-2 video as recorded by our camera stores colour in the $Y'C_rC_b$ format with one channel corresponding to brightness and two corresponding to colour-opponency information (Poynton, 2003). We directly used the intensity values from all channels.

Velocity Motion estimation followed the algorithm based on the minors of the structure tensor presented in Section 2.11. Here, ω was a spatio-temporal smoothing filter with five-tap binomial kernels in both space and time. Four estimates of local motion $\vec{v}_1, \ldots, \vec{v}_4$ were obtained and only processed further if they deviated from each other by not more than 45 deg. Velocity was then computed as $v = \sqrt{\overline{v_x}^2 + \overline{v_y}^2}$ and locations where v was less than 1% of the maximum velocity in that video frame were discarded. Finally, results were smoothed with a Gaussian kernel with length 15, $\sigma = 3$ pixels.

Geometrical invariants We also computed the geometrical invariants on the structure tensor (see Section 2.8) that have been shown to be useful in understanding biological vision (Barth and Watson, 2000); they have also been used to predict eye movements before (Böhme et al., 2006a; Vig et al., 2009). An example image for invariant S on a natural movie is shown in Figure 4.2.

Artificial scanpaths as baseline measure

To be able to compare our results against a baseline measure, we created random sequences of fixations, or scanpaths. However, real scanpaths have certain characteristics that need to be taken into account. For example, the distribution of saccadic amplitudes that subjects made on our stimuli (see Figure 3.3) is



(a) (b)

Figure 4.2: *(a) Stillshot from one of the movies used in our experiment. (b) Corresponding image of geometrical invariant S. Non-white locations change in at least two spatio-temporal directions (brightness thresholded and inverted for better legibility). For an illustration of intrinsic dimension on a synthetic scene, see Figure 2.8.*

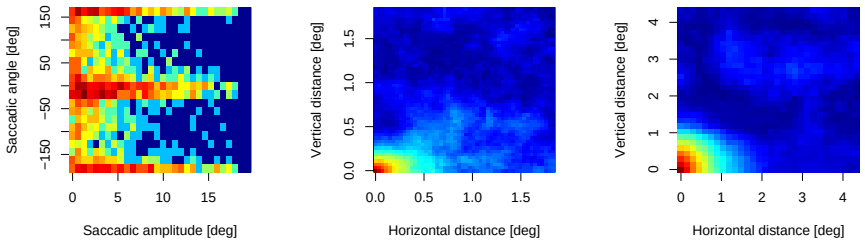
heavily skewed (mean amplitude is 7.4 deg, median is 5.6 deg). Because natural scenes show spatio-temporal correlations that vary with distance (Zetzsche et al., 1993; Simoncelli, 1997), see also Figure 4.3, any correlations found along the scanpath might be due to these image-inherent correlations alone. Furthermore, it is a well-known fact that human gaze prefers image patches with high local structure, such as edges, corners, or motion. This repulsion from homogeneous areas is of particular importance in the context of the orientation feature since orientation cannot be reasonably extracted from such areas.

In order to disambiguate these effects, we created four different sets of baseline scanpaths with a different similarity to the recorded scanpaths. A graphical illustration of these control conditions is given in Figure 4.4.

"Random" Fixation durations were copied from real scanpaths, but image coordinates of fixations were uniformly sampled across the whole scene, resulting in a mean saccadic amplitude of 19 deg. Thus, in this condition neither saccadic amplitude nor the set of fixated patches remained the same as in the real scanpaths.

"Same lengths" Saccade lengths were copied from real scanpaths, but direction was randomized; most correlations inherent in natural scenes were therefore conserved, but the image patches from which features were extracted were random.

"Scrambled" In this condition, the order in which a subject fixated a series of image patches was shuffled. This yielded a different distribution of saccadic amplitudes (mean 13 deg, almost twice as large as that of the original distribution), but the set of fixation coordinates (x, y) remained constant. Note that this does not imply that fixated image patches were exactly the same; because

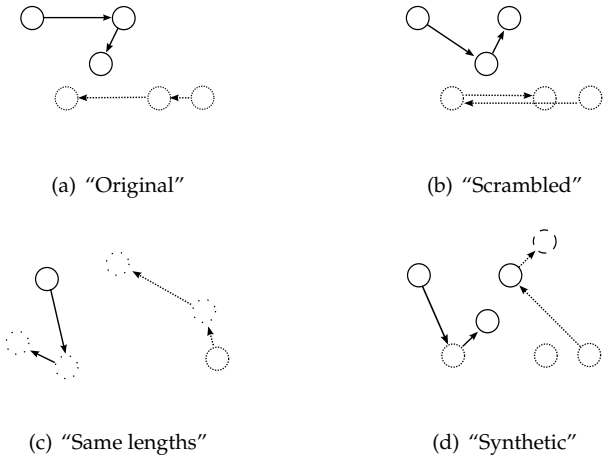


(a) (b) (c)

Figure 4.3: *(a) Log-plot of joint distribution of saccadic amplitudes and angles. There is a strong bias towards horizontal and vertical saccades. (b) Image-based correlation of local orientations on the highest spatial scale (13.4 cycles/deg). The bottom-left corner corresponds to the correlation of a pixel with itself, which is 1.0 by definition. At longer distances (above 0.5 to 1 deg), correlations drop to chance level; notable is the anisotropy that correlations decay more slowly along the horizontal axis. (c) Image-based correlation of local orientations for a middle spatial scale (3.3 cycles/deg). Again, correlations are anisotropic.*

of moving objects and illumination changes over time, features at (x, y, t_1) and (x, y, t_2) might differ for $t_1 \neq t_2$.

"Synthetic" All the above conditions are based on data from a single trial (combination of one subject and one movie) per output scanpath. Using data only from a single subject, it is impossible to change the scanpath (i.e. generate an artificial scanpath) while keeping constant both the set of fixated patches and the spatio-temporal distances between these fixations. However, by mixing scanpaths made by different observers on the same video, both these characteristics can be approximated simultaneously. Consider a sequence of two fixations made by subject A: $f_A(n) = (x_A(n), y_A(n))$, $f_A(n+1) = (x_A(n+1), y_A(n+1))$ with a distance $\Delta_A(n) = (x_A(n+1) - x_A(n), y_A(n+1) - y_A(n))$ (for simplicity, we ignore time in this example). In an artificial scanpath S, we would then want to model a pair of fixations with the same distance (since Δ is a vector-valued function, this also includes the angle between the two fixations), $f_S(n) = (x_S(n), y_S(n))$, $f_S(n+1) = f_S(n) + \Delta_A(n)$. Furthermore, $f_S(n)$ and $f_S(n+1)$ should not be random points, but real fixation points. Given a sufficient number of scanpaths from other subjects, it is not unlikely to find (at least approximately) such a pair of fixations, e.g. from subjects B and C: f_B, $f_C = f_B + \Delta_A(n) + \epsilon$, that we can use for our "synthetic" scanpath: $f_S(n) := f_B$, $f_S(n+1) := f_C$. Care has to be taken, however, that the artificial scanpath does not coincidentally become a mere copy of original scanpath segments, i.e. that there is no subject X with fixations $f_X(n) = f_B$, $f_X(n+1) = f_C$.



(a) "Original" (b) "Scrambled"

(c) "Same lengths" (d) "Synthetic"

Figure 4.4: *Illustration of the control conditions ("random" not shown). (a) Measured scanpaths from two subjects (solid line / dashed line). (b) "Scrambled": fixations are the same, but their order is randomized. (c) "Same lengths": the fixation locations (except for the start position) are random, but the connecting saccades have the same amplitudes as in the "original" condition. (d) "Synthetic": using scanpaths from several subjects, both the set of fixated locations and the joint distribution of saccadic angles and amplitudes are approximated (in this small sketch, only amplitudes are similar); no saccadic segment occurs in the "original" scanpaths. Note the fixation from a putative third subject in the top right corner.*

In practice, "synthetic" scanpaths were created as follows. An output scanpath was initialized with the first fixation of an original scanpath. Then, the same number of fixations as in the input scanpath was generated by sampling pairs of angles and amplitudes from the joint distribution over the original scanpaths (see Figure 4.3(a)); for each sample, we searched among all observers' fixations for one with a similar distance at a similar angle to the current fixation (tolerances were 0.2 deg of amplitude and 10 deg of angle). Because of moving objects, the image patch around one fixation point might look different over time, and therefore we initially searched only among those fixations that had been made at a similar point in time (tolerance 0.5 s). As mentioned above, theoretically it would be possible to end up with an exact copy of the input scanpath, since that copy trivially mimics both saccadic amplitudes and angles and the set of fixation points. Therefore, a further constraint was that no sampled pair of saccade onset and offset was also part of any of all subjects' original scanpaths (again with a tolerance of 0.2 deg). Obviously, these conditions could not always be fulfilled: even a large data set of fixation points is relatively sparse on the screen (the screen measures about 1300 deg^2; at a spatial tolerance of 0.2 deg, a single fixation point covers only 0.01% of this area), and certain combinations of angles and amplitudes might take a scanpath outside

the borders of the video, which is clearly nonsensical. In these cases, sampling from the joint distribution was repeated up to 10 times and the tolerance for "similar" time points was gradually relaxed until a matching fixation patch could be found.

To assess how closely the original distribution of saccade length and direction was approximated, we computed the Kullback-Leibler divergence between the original distribution and those generated by the baseline conditions. For a reference point, we also computed the KLD of one half of the original data set to the other half. Results were 1.19, 0.4, 0.08, 0.07, and 0.05, respectively (for "random", "scrambled", "lengths", "synthetic", and "original"). These results show that the "synthetic" scanpaths are only an approximation to the original scanpaths, but model the saccade characteristics of original scanpaths more closely than those in the "lengths" condition, even though they consist only of real fixation points (in the "lengths" condition, fixation points are random).

In summary, by introducing the concept of synthetic scanpaths, we can avoid the shortcomings of random and scrambled scanpaths and, in addition, match the natural distribution of saccade length and direction.

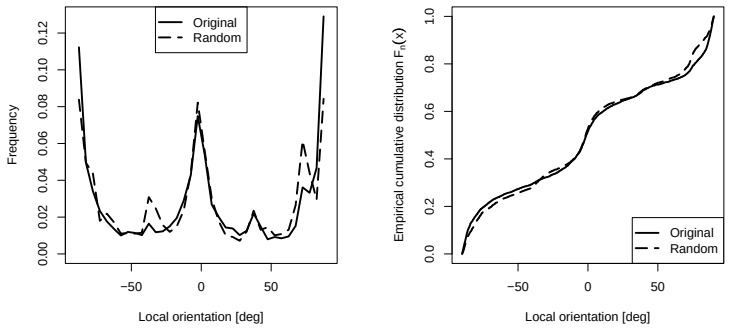
4.3 Results

To see whether the features along scanpaths made by human observers are correlated beyond the level that is to be expected from image-inherent spatio-temporal correlations alone, we have to compare the distributions of feature differences along the "original" scanpaths with distributions based on the control scanpaths. Because of random fluctuations, finding subtle differences in raw distributions is quite hard; we therefore look at the empirical cumulative distribution functions

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{X_i \le x},$$

where $I_{X_i \le x} = \begin{cases} 1 & \text{if } X_i \le x \\ 0 & \text{otherwise} \end{cases}$, which integrate over difference magnitude.

As an example, consider the two distributions of orientation values in Figure 4.5. The solid line depicts the distribution of orientations at human fixation points and the dashed line those at random control points; the dominance of the horizontal ($\phi = 0\,\text{deg}$) and vertical ($\phi = \pm 90\,\text{deg}$) axes is a well-known property of natural scenes and can therefore be found both in human and random data. The ECDF (shown in the right panel) at x tells us what proportion of samples have a value of less than or equal to x, e.g. about 50% of samples have an orientation between -90 and 0 deg. Peaks in the probability distribution (left



(a) (b)

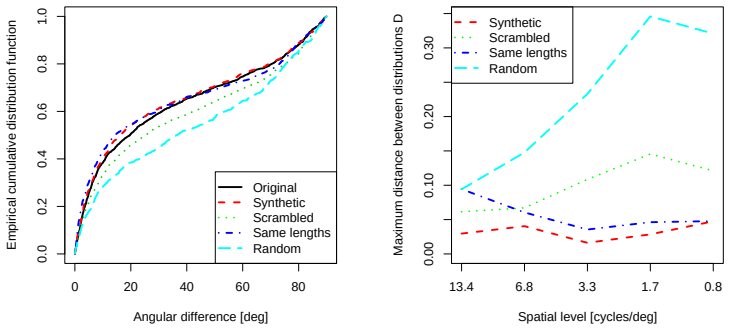
Figure 4.5: *Example of probability and empirical cumulative distribution functions (ECDF); here, the distribution of orientations is plotted. $F_n(x)$ denotes the proportion of samples having a value of less than or equal to x, e.g. 50% of samples have an orientation between -90 and 0 deg. Peaks in the probability distribution (a) correspond to a steep slope in the ECDF (b), e.g. at -90, 0, 90 deg.*

panel) correspond to a steep slope in the ECDF (e.g. for the cardinal axes); low $p(x)$ values correspond to plateaus (e.g. for oblique orientations). Based on the ECDF, the Kolmogorov-Smirnov test statistic $D_{ij} = \sup_x |F_i(x) - F_j(x)|$ denotes the maximum distance of two cumulative distributions on the y-axis. In our example in Figure 4.5, this maximum distance is 6.3%: around 82% of samples in the "original" distribution have an orientation of less than $x = 80$ deg, but the dashed "random" curve has reached more than 88% at this point already.

Depending on the number of samples in the distributions, every such distance D_{ij} is then assigned a probability p to test for statistical significance. Since the Kolmogorov-Smirnov test is valid only for continuous distributions, but the low-level features colour and invariants are represented by discrete values, we performed a 1000-fold bootstrap test and report 95% confidence interval values.

We should take statistical tests with a grain of salt, though. Overall, we have almost 500 conditions (5·3 spatio-temporal levels, eight different features with varying parameters, four types of control scanpaths). Even at a significance level of $p = 0.01$, this implies that we have to expect around five conditions with presumably significant results, even if there was no underlying effect. Therefore, we carefully have to look out for systematic effects, i.e. those that are robust against scale or parameter changes. Also, because of the high number of samples, even miniscule effects can show up as highly significant.

In the following, we will present and discuss some representative findings. We will start out with orientation and colour because here the analysis is



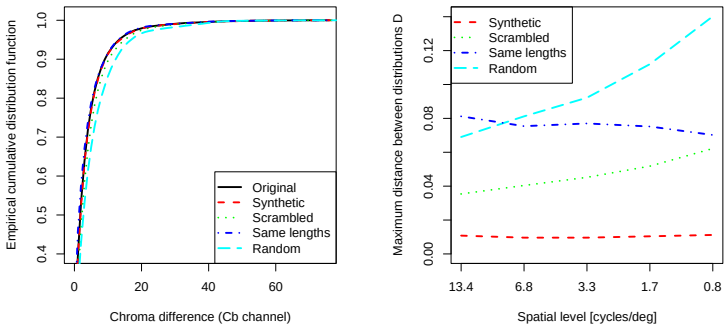
(a) (b)

Figure 4.6: *Results for local orientation. (a) ECDF for differences of orientations on the second spatial, first temporal level. The "random" and "scrambled" conditions strongly differ from the original data in their saccadic amplitudes and therefore also differ in their orientation differences. The "same lengths" condition is closer, but still different at around significance level ($D = 6.0\%$, $p < 0.017$); "synthetic" scanpaths show no such difference ($D = 4.0\%$, $p > 0.18$). (b) Maximum distance between original data distribution and control conditions for different spatial scales (first temporal scale; results are similar for other temporal scales). The "synthetic" condition is always closest; "random" is particularly different on the lower-frequency scales.*

straightforward; results for motion and the geometrical invariants need more consideration.

The evaluation of local orientation poses the problem that the thresholds θ_1, θ_2, which separate oriented from homogeneous patches, have to be defined. We systematically varied these parameters and found, not surprisingly, that for low orientation specificity ($\theta_1 < 0.02, \theta_2 < 0.4$), random noise dominates the measurements and the control conditions cannot be distinguished from the "original" condition. At e.g. $\theta_1 = 0.05, \theta_2 = 0.8$, however, reliability of orientation estimation is high; at only about 12% of image patches can orientation be extracted then (nevertheless, the following also holds true for moderate parameter variation). Because of the well-known fact that human fixations are drawn to structured image regions, the number of strongly oriented patches decreases slightly for the "same lengths" and the "random" conditions (to about 9%).

In Figure 4.6(a), the distributions of orientation differences along the scanpaths are plotted for one exemplary spatio-temporal scale. Clearly, the "scrambled" and the "random" conditions are very different from the original data. In these conditions, the saccadic amplitudes changed drastically and hence, also the distance-determined correlations of the image patches changed. The "same lengths" condition mimics the original data more closely, but is still different almost at significance level ($D = 6.0\%$, $p < 0.017$); however, only when the image-based correlations are fully modelled in the "synthetic" condition and



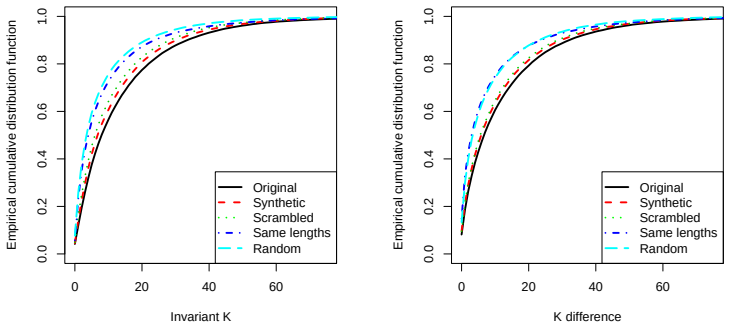
(a) (b)

Figure 4.7: *(a) ECDFs of colour differences on the fourth spatial, first temporal level. The "synthetic" condition shows no significant difference to the original data ($D = 0.7\%$, $p > 0.07$). (b) Maximum distance between original data distribution and controls for different spatial scales.*

even the angular distribution of saccades is taken into account, the difference to the human data vanishes. Compared to the "synthetic" artificial scanpaths, human subjects did not show a preference for certain orientation differences from one fixation to the next ($D = 4.0\%$, $p > 0.18$). The same pattern can be seen in Figure 4.6(b), where the test statistic D is plotted for all spatial scales. The "synthetic" condition is always closest to "original", and "random" is particularly bad on the lower spatial scales.

Because Dragoi and Sur (2006) found different effects for saccades of different sizes, we also evaluated subsets of our data based on saccadic amplitude: following Dragoi and Sur, we binned saccades into small (<1 deg), medium (1–3 deg), and large (>3 deg); since the stimuli in our data set were much larger, we also partitioned the saccades along the median of roughly 6 deg. No significant differences between "synthetic" and "original" could be found in any of these subsets (data not shown).

Proceeding to the next low-level feature, colour, Figure 4.7(a) shows exemplary data for the blue-difference chroma channel C_b on the fourth spatial level, but the following applies also to luma (Y) and red-difference chroma (C_r). Here, all those artificial scanpath models with different saccadic amplitudes ("scrambled" and "random") or different fixation locations ("same lengths") lead to very different colour differences along the scanpath ($p < 10^{-5}$ on almost all spatio-temporal levels). Only the "synthetic" condition shows no significant difference to the original data ($D = 0.7\%$, $p > 0.07$); for this condition, no such difference can be found for any spatio-temporal level (see Figure 4.7(b)) or colour or brightness channel.



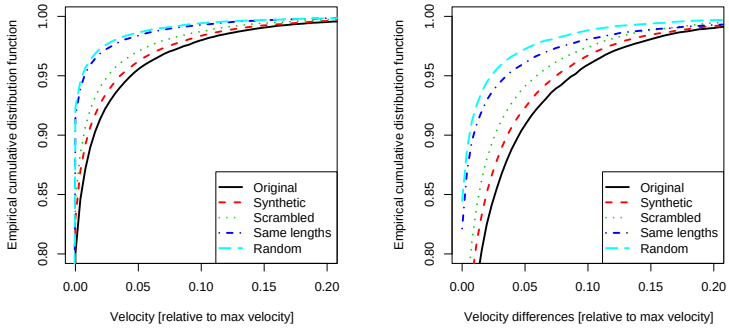
(a) (b)

Figure 4.8: *(a) Cumulative distribution of K values at fixated image patches. The "original" condition shows a small bias towards larger K values compared to "synthetic" and a large bias compared to "same lengths". (b) ECDF of differences of geometrical invariant K on the third spatial and the first temporal level. There is a statistically significant difference ($D = 2.5\%$, $p < 10^{-5}$) between the "original" and the "synthetic" condition, but this difference can be explained by the difference in the underlying feature distributions, see (a).*

In Figure 4.8(a), results are plotted for the geometrical invariant K, which describes the intrinsically three-dimensional video patches such as transient corners. Similar effects could be found on several spatio-temporal levels, and we will here describe one exemplary case (third spatial, first temporal level). Statistically significant differences could not be found for invariants H and S, which correspond to intrinsically one- and two-dimensional features; these features are less sparsely distributed than K and the following discussion therefore does apply only loosely to them.

The black solid curve in Figure 4.8(b), which represents the "original" data, saturates later than the other curves; they, in turn, have a steeper slope near $\Delta K = 0$. This means that in the original scanpaths, K values showed larger absolute differences. This effect is particularly strong when comparing the original scanpaths with the conditions "same lengths" and "random", which are those conditions where image patches were drawn (quasi-)randomly. The difference for the "synthetic" and "scrambled" conditions is less pronounced, but still is statistically significant ($D = 0.9\%, p < 0.017$).

Let us now turn to Figure 4.8(b) for an explanation. Shown here are the cumulative distributions of raw K values at fixated image patches. When comparing the "original" condition with "same lengths", we can see that there is a strong bias towards higher K values, which is in line with the observation that humans prefer to look at highly structured image regions. The image patch selection in the "same lengths" condition, on the other hand, was random and therefore showed no such bias. Although the set of image patches in the



(a) (b)

Figure 4.9: *(a) Cumulative distribution of velocities. Subjects exhibit a clear bias towards image patches with high velocities. (b) ECDF of differences of velocity on the third spatial and the first temporal level. There is a statistically significant ($D = 2.9\%$, $p < 10^{-5}$) difference between the "original" and the "synthetic" condition.*

"synthetic" condition approximates the measured set of fixated image patches, some spatio-temporal uncertainty is introduced (see above), so the raw K values for this condition are slightly smaller than for the recorded data ($D = 3.6\%$; although these numbers cannot be compared directly, this is at least in the same order of magnitude as the distance of the distributions of ΔK, $D_{\Delta K} = 2.5\%$).

Thus, we can state that the distribution of K at the centre of gaze is wider for human data than for artificial scanpaths; therefore, the distribution of differences along the scanpath also becomes wider. This bias of the human visual system towards image regions with higher K values, i.e. regions of changes in all spatio-temporal dimensions, can be used to reliably predict eye movements (Vig et al., 2009), regardless of the question whether this observed bias is merely a correlate of other, top-down factors such as a preference for (moving) objects. However, there is no strong evidence for a particular bias in selecting the next saccade target based on the K value at the current centre of fixation.

A similar effect could be found for the motion feature. On almost all spatio-temporal levels, there are significant differences between the original scanpaths and all control conditions. For an example, the distribution of velocity differences on the third spatial and first temporal level is shown in Figure 4.9(b). Again, human subjects show a bias towards larger absolute feature differences compared to random processes, but as in the case of K, the underlying distribution is also different. As can be seen in Figure 4.9(a), humans tend to fixate moving objects more often (in practice, moving objects are often followed with a smooth pursuit eye movement; see the Methods section for a discussion). Note that the difference between the "original" and the "scrambled" condition

is fairly large here even though the spatial locations of the image patches stay the same. Their temporal order changes, and by definition, a moving object will be at a different place at a different time.

Summarizing our results, we can conclude that for orientation, as well as for the other low-level features, there is no significant contribution of the feature at the current centre of gaze to saccade target selection.

4.4 Gaze prediction with machine learning

So far, we have attempted to predict where a person will look next based on what they are looking at right now. We did not find strong evidence for a mechanism in the human visual system that would pick fixation targets based on low-level features at the current fixation; however, we found – in line with earlier findings – a very pronounced preference for fixating image regions with local structure. As we have discussed at the beginning of this chapter, much research has been devoted to the question how this preference can be used to predict where human subjects will look in visual scenes. A common approach is that of a saliency map that is typically computed from a set of biologically inspired feature detectors. An alternative approach is to employ techniques from machine learning to differentiate between attended and non-attended movie patches, and thus to obtain information about the image structures that are relevant for this distinction. The first to follow such an approach were Kienzle et al. (Kienzle et al., 2007, 2006, 2009), who learned on all raw pixel intensities in a neighbourhood around fixation on one spatial scale. Their trained classifier had a centre-surround receptive field structure, which often also is a part of biologically plausible saliency map formulations, and prediction performance was in line with other results in the literature. A major problem with learning on all pixels in a neighbourhood, however, is the curse of dimensionality. Even small (spatio-temporal) neighbourhood sizes quickly become intractable, and so learning has to be constrained to one scale with relatively small resolution. In the following, we shall therefore present a novel algorithm to predict eye movements on natural movies based on machine learning that employs a trick to reduce dimensionality, and thus makes the problem tractable. We explicitly discard information by averaging image feature energy in the spatial neighbourhood around fixation, so that we obtain only one scalar value instead of up to thousands (e.g. 4096 for an image patch of 64 by 64 pixels). This dimensionality reduction, on the other hand, allows us to use more information elsewhere, and we obtain such averaged energy on every scale of a spatio-temporal pyramid, so that the feature vector

for the classifier is multidimensional again; the use of temporal scales ensures that information is taken from more than one point in time.

Obviously, the choice of image feature is crucial for this algorithm. Simply using the raw pixel intensities would result in a distinction of bright and dark patches only after averaging, which probably is not enough to capture the complexity of eye guidance processes. We therefore use the geometrical invariants on tensor-based image representations (see Chapter 2); without machine learning, the geometrical invariants on the structure tensor have been used to predict gaze before (Böhme, Dorr, Krause, Martinetz, and Barth, 2006a). Besides the prediction of eye movements per se, our algorithm can also be used to compare different image features and their predictive power. We shall therefore also compute the invariants on the multispectral structure tensor (see Section 2.9) and investigate whether the inclusion of colour information improves predictability.

To evaluate the performance of our algorithm, we use the receiver operating characteristic (ROC) curve, which gives us insight on the relationship of specificity (how many of the patches that were classified as fixated indeed are from the fixated class) and sensitivity (how many of the fixated patches were classified as such) of the algorithm. The area under this curve (AUC) provides an intuitive number for performance; a perfect classifier would achieve an AUC of 100% and a random classification would result in an AUC of 50%. The first to use ROC scores for the analysis of gaze prediction schemes were Tatler et al. (2006); in the meantime, several groups have now reported such scores, typically in the range 0.58–0.68, for their algorithms.

Prediction of eye movements on natural movies

We computed image features at about 40000 saccadic landing positions of the data set presented in the previous chapter, i.e. of 54 subjects watching 18 high-resolution movies of natural scenes. Because of the latency of the oculomotor system, we extracted image features not at the end of the saccade (i.e. at fixation onset), but 70 ms earlier. We derived this number by choosing the time offset that yielded maximum cross-correlation of our dynamic saliency measures with the so-called empirical saliency, i.e. a saliency map obtained from real gaze data. Obviously, these 70 ms are much shorter than the well-established latency of about 150–250 ms reported from laboratory experiments with synthetic stimuli, such as the sudden onset of saccade targets. Apparently, natural stimuli are highly predictive, and the human visual system can thus partially compensate for its physiological and mechanical latency.

Generating a set of negative examples, i.e. non-attended movie locations, is a non-trivial challenge. A commonly used, straightforward approach is to randomly sample locations in (x, y, t); a slightly better way is to pick locations only if they were in fact not attended, i.e. if their spatio-temporal distance to the nearest fixation exceeds a threshold. This, however, poses the problem that a spatial separation of several fovea diameters might arguably be considered to be enough to make two fixations dissimilar; the role of temporal distance, on the other hand, is not that clear.

Both these approaches have the disadvantage that they ignore two spatial biases of stimulus design and subjects. First, the central bias (see Chapter 3) makes subjects look preferentially at the stimulus centre. The central bias can be used by itself to predict eye movements, but here we want to delineate the predictability based on image features alone from other factors. Second, the so-called photographer's bias leads to highly structured and semantically interesting objects often being placed in the centre of the stimulus, which might lead to a structural difference of central and peripheral image patches.

Therefore, we chose to generate negative examples in a way that kept the spatial distribution of fixations intact. Scanpaths of subjects and movies were shuffled, so that the set of fixations on movie A served as positive training examples on movie A; the same number of fixations was drawn randomly from all other movies to create the set of negative examples.

As stated above, the feature vector contained the average image feature energy in a neighbourhood around the saccadic landing point (x, y) on each scale of a multiresolution pyramid; for an anisotropic pyramid with S spatial and T temporal levels, the feature vector formally was thus

$$\vec{x} = (e_{0,0}, e_{0,1}, \ldots, e_{S,T}),$$

$$e_{s,t} = \sqrt{\frac{1}{W_s H_s} \sum_{i=-W_s/2}^{W_s/2} \sum_{j=-W_s/2}^{H_s/2} I_{s,t}^2(x_s - i, y_s - j)}. \qquad (4.1)$$

with a neighbourhood width and height on spatial scale s of W_s and H_s, respectively, and a gaze position of $(x_s, y_s) = (x/2^s, y/2^s)$ because of the reduced resolution on spatial scale s (see Chapter 2).

For the results reported below, we used a neighbourhood size of 128 by 128 pixels on the highest scale and accordingly smaller sizes on lower scales, so that the effective window size was about 4.8 deg on all scales. Larger window sizes tend to average over too much of the stimulus area to still allow for fine distinctions; smaller window sizes suffer from the spatial uncertainty both in saccade programming (saccades tend to undershoot, see Chapter 2) and the eye

tracker measurement. In time, only the information from the current frame was used; on the lower scales, this implicitly averaged over a longer time course (up to half a second for five scales).

Spatial uncertainty also affects the choice of filter kernels to compute the structure tensor, on which the geometrical invariants are based. We here used spatio-temporal binomial kernels of length five, $(1, 4, 6, 4, 1)/16$ both for the lowpass filter ω and for smoothing before computing the partial derivatives; for derivation, a standard highpass $(-1, 0, 1)$ was used.

Ideally, we would have computed the geometrical invariants "on the fly", because this would have left intact the dynamic range of the floating-point computations. However, in order to achieve statistical confidence that our results are not due to chance alone, we had to analyse the features multiple times; despite our efficient implementation, this was not feasible computationally. Therefore, we stored the geometrical invariants to disk as video streams, which could then be easily read repeatedly. The dynamic range of videos is limited to $[0, 255]$ and the geometrical invariants thus had to be normalized to that range. To this end, H, S, and K were first raised to the power of six, three, and two, respectively, because they comprise of products of one, two, and three eigenvalues, respectively. The dynamic range then had to be reduced again by taking the eighth root and was finally mapped linearly to $[0, 255]$.

Once features were extracted at attended and non-attended locations, data was partitioned into a training set of two thirds of the available data and a test set of one third. A soft-margin support vector machine with Gaussian kernel (Schölkopf and Smola, 2002) was trained using the training set; the optimal parameters for the width of the Gaussian γ and the penalty constant C were found by five-fold cross-validation. To obtain statistical confidence, prediction performance was evaluated on 10 realizations of the data into training and test set.

Results for the geometrical invariants both on the luminance channel alone and a multispectral representation are shown in Figure 4.10. Predictability reaches an ROC score of up to 0.74, which is favourable in comparison to the numbers reported in the literature so far that have been in the range between 0.58 and 0.71 (note, though, that numbers cannot be directly compared across different data sets; many studies have also reported other measures than ROC scores). The qualitatively most relevant result, however, is that prediction performance increases with the intrinsic dimension ($K > S > H$, $p << 0.0003$, paired non-parametric Wilcoxon's signed rank test); movie regions that change in more spatio-temporal directions (and are thus more informative) are also more predictive for eye movements. This relates directly to the fact that $i0D$ and $i1D$ are redundant; a reasonable supposition then is that the human visual

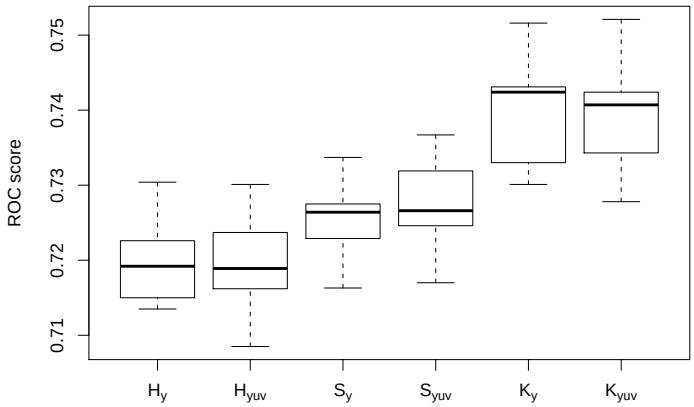


Figure 4.10: *ROC scores for prediction of eye movements on natural movies using the geometrical invariants of the structure tensor on the luminance channel (Y) and of the multispectral structure tensor (YUV). Invariants were computed on an anisotropic pyramid with five spatial and five temporal levels, and feature energy was averaged in a window of about five degrees. Regions with higher intrinsic dimension are significantly more predictive for eye movements, $K > S > H$ ($p < 0.0003$, paired Wilcoxon's signed rank test) for both conditions. However, there is no significant difference between the single-channel and the colour condition except for the invariant S, which is weakly significantly better on colour ($p < 0.043$).*

system has learned to preferentially fixate those regions that are most informative.

No strong difference could be found between the invariants on luminance and those on a multispectral representation (S is better on colour with weak significance, $p < 0.043$). We also found no significant difference between the invariants computed on the structure tensor and those computed on the energy tensor (data not shown). A possible explanation is that the main advantage of the energy tensor over the structure tensor is that is does not require strong regularization (with the lowpass kernel ω, see Equation 2.6). In principle, the energy tensor thus yields a sparser representation; this advantage, however, is lost during our computation of average energy.

Prediction of eye movements on overlaid movies

An interesting question now is whether the prediction of eye movements on natural movies also generalizes to the case of multiple overlaid movies. With the generalized structure tensor (see Section 2.12), we already have a tool available to describe multidimensional signal variation, and we shall present results for gaze prediction on transparently overlaid movies using this generalized tensor

in the following. An example stimulus and a description how stimuli were created can be found in Section 2.6. Obviously, such visual input with global transparency is not fully realistic anymore; locally, however, multiple motions are common in natural scenes due to occlusions, reflections, etc.

The generalized structure tensor J_2 is more powerful than the previously used J_1. For example, based on the rank of J_2, a distinction becomes possible between the superposition of a moving 1D and a moving 2D pattern (rank $J_2 = 4$), two moving 2D patterns (rank $J_2 = 5$), and higher-order motion types; the rank of J_1, however, is three in all these cases. Therefore, we tested the hypothesis that a prediction based on J_2 might outperform prediction based on J_1. If this is indeed the case, this will be even more remarkable since the tensor products in J_2 consist of second-order derivatives, which are more sensitive to noise than the first-order derivatives in J_1.

We first created a set of 19 movies based on two randomly chosen movies from our set of natural movies (see above). They were blended following the algorithm in Section 2.6 on an anisotropic spatio-temporal Laplacian pyramid with five spatial and five temporal levels; the contribution of the two movies to each frequency band was equalized to obtain similar visibility. Because of temporal border effects of the pyramid, resulting movies were slightly shorter than their individual parts (17 s instead of 20 s).

Ten subjects watched the 19 transparent movies while their eye movements were recorded by an SMI Hi-Speed eye tracker running at 1250 Hz. Initially, the eye-tracking equipment was calibrated using a five-point procedure, and a drift correction was applied before each movie screening. Stimuli were displayed on an Iiyama MA204DT screen at a distance of 55 cm from subjects, so that they covered a visual field of 40 by 22.5 deg. The subjects' task was simply to "watch the movies attentively".

Overall, about 17000 saccades were extracted. Negative training examples were obtained by shuffling movies and scanpaths, but with a slightly different algorithm than on single natural movies. The set of fixations on movie A served as positive examples for movie A and as negative examples for movie B, and vice versa; the pairs of movies A, B were drawn randomly for each realization. Because of this difference, numbers cannot be compared directly between single and overlaid movies; here, however, we only compared prediction performance of features computed on J_1 and J_2. Both tensors were computed on an anisotropic Gaussian pyramid with 25 levels (five spatial, five temporal) as above. In principle, it is also possible to estimate the rank of J_2 based on its minors; the necessary terms become rather complex, however, and so we chose to use a publicly available eigenvalue solver (Galassi et al., 2009).

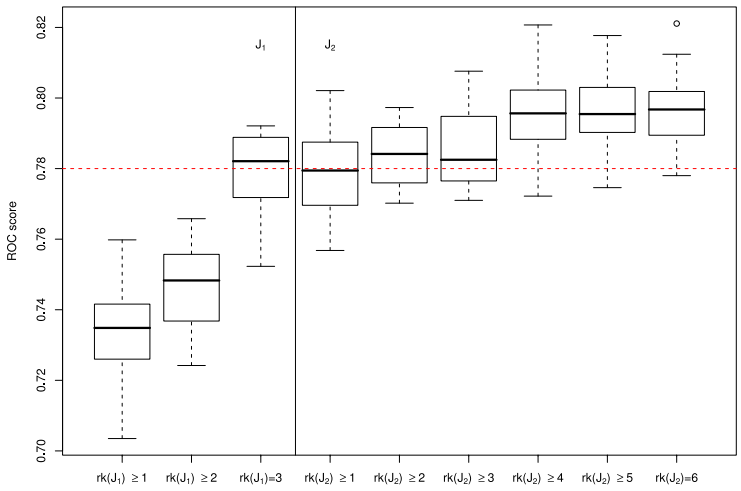


Figure 4.11: *ROC scores for prediction of eye movements on overlaid movies. A support vector machine was trained with 25-dimensional feature vectors; these contained the average feature energy in a neighbourhood around fixation (about five degrees diameter) on each spatio-temporal scale of an anisotropic multiresolution pyramid (five spatial, five temporal levels). Features were computed at those movies regions where the structure tensor J_1 (left) or the generalized structure tensor J_2 (right) had at least a certain rank. As in Figure 4.10, predictability is better with a higher rank. J_2 is also significantly better than J_1 ($p \approx 10^{-4}$, paired Wilcoxon's test, 25 cross-validations).*

Results for 25 realizations of the data into training and test set are shown in Figure 4.11, and we can draw three interesting conclusions. First, we can confirm the result obtained on single natural movies that eye movements are highly predictable based on a simple model of how the spatio-temporal signal changes locally. Second, regions that change in more directions (where the rank of J_2 is higher) are more predictive than more uniform regions. Finally, the higher-order representation of J_2 that allows for a finer distinction between motion types yields significantly better results than J_1 ($p < 1.1 \cdot 10^{-4}$, paired Wilcoxon's test).

4.5 Discussion

The present analyses were motivated by our research on gaze prediction and gaze guidance. As we saw at the beginning of this chapter, low-level features such as contrast and motion can be successfully used to predict where observers will direct their gaze in natural movies. In order to potentially improve such prediction algorithms, we investigated the correlation of a variety of low-level features across consecutive fixations. In line with earlier findings by Dragoi and Sur (2006), we found that such correlations are not random and feature differences along the scanpath exhibit systematic characteristics. However, our

data does not support the hypothesis by Dragoi and Sur that neural adaptation plays a crucial role in forming these characteristics; in other words, that low-level features at the centre of gaze contribute to saccade target selection.

On the contrary, we found that the correlations of features along the scanpath can be explained by two factors. First, natural scenes themselves show strong spatio-temporal correlations, and any distribution of saccadic amplitudes and angles will reproduce these correlations to a varying degree. Second, there exists a general bias in saccade target selection, e.g. the preference of human observers to look at image regions with spatio-temporal structure, which in natural scenes often corresponds to object locations.

For geometrical invariants, which describe the number of spatio-temporal dimensions that change locally, and motion, this preference resulted in a wider distribution of raw feature values at fixated patches; therefore, differences of those features at successive fixations also differed from those in control conditions. For colour and local orientation, we were able to find an effect only for some of the control scanpath models; when we matched saccade statistics and therefore matched the scene-inherent spatio-temporal correlations, the effect vanished.

Nevertheless, we should stress that our findings do not rule out that low-level features at fixation contribute to saccade target selection at all; it is possible that the human visual system might have learned to make use of a specific distribution of saccadic amplitudes and angles, which induces correlations in the sequence of fixated low-level features that may be beneficial in terms of neural adaptation. However, such a putative mechanism would require no direct knowledge of the relationship of features at fixation and potential saccade targets.

If we had found strong evidence that the visual system does indeed evaluate and compare low-level features at fixation and in the periphery, this would have been a strong argument in the ongoing debate whether top-down or bottom-up factors are more important in the control of eye movements on natural scenes. We here found no indicator that low-level features are explicitly represented and used in oculomotor control. Nonetheless, the opposite conclusion that low-level features are irrelevant is also not supported by our data, since here we investigated exclusively the role of features along the scanpath, not at single fixations.

In order to be able to distinguish between the different sources of feature correlations, we developed and compared several methods to generate artificial scanpaths. The “scrambled” and the “lengths” condition focus on the characteristics of saccade target selection and of oculomotor tendencies, respectively; the “synthetic” condition accurately models both these processes and should

thus be preferred, but requires a larger data set to sample from. The highly different results we obtained for these different control conditions emphasize the importance of precisely modelling saccade statistics when comparing human subjects with random processes. In general, this helps to disentangle the properties of the visual input and those of the human visual system.

In the second part of this chapter, we used a novel machine learning algorithm to classify movie patches as attended or non-attended and obtained very favourable prediction results. Image patches and movie sub-volumes have many pixels and therefore machine learning methods on such patches suffer from the curse of dimensionality. We explicitly forfeited information and reduced dimensionality by computing only the average of feature energy in a neighbourhood around fixation. This step then enabled us to compute features on multiple scales of a multiresolution pyramid, thereby modestly increasing dimensionality again.

A critical choice in this approach is that of the correct image feature. We first computed the intrinsic dimension of the movie patches based on the geometric invariants of the structure tensor. Results showed that those patches with higher intrinsic dimension were significantly more predictive for eye movements. This is an interesting finding for two reasons. First, regions with higher intrinsic dimension change in more spatio-temporal directions, and therefore are more informative in an information-theoretic sense; the human visual system apparently prefers these informative patches. Second, only patches with an intrinsic dimension of at least two are needed to fully determine a movie, and the fact that patches with lower dimension, that is redundant patches, are less often fixated indicates an efficient coding strategy of the brain. Estimating the intrinsic dimension on different tensor representations, such as the multispectral structure tensor and the energy tensor, did not significantly change prediction results.

Finally, we extended our approach also to the case of multiple overlaid movies. Typical natural input obviously does not comprise of two very different superimposed movies, but locally, multiple motions abound because of occlusions. Not only could we replicate findings that movie patches with a higher intrinsic dimension are more predictive of eye movements, but we could also show that the generalized structure tensor is able to capture the effects of multiple signals better than the classical structure tensor. The geometric interpretation of multidimensional signal variation thus is a useful tool in understanding human vision.

4.6 Chapter conclusion

In this chapter, we have investigated how eye movements on dynamic natural scenes can be predicted based on a low-level stimulus description. With a novel application of machine learning techniques, we were able to show that the human oculomotor system preferentially fixates more informative image regions. In computer vision, such a strategy is known as an interest point detector, and there are technical applications such as image understanding or robotics. Here, however, we were interested in gaze-guidance systems and the understanding of human vision required to build them. As a benefit across interdisciplinary borders, we could show that a neural adaptation mechanism that supposedly was found in monkeys does not have a human homologue.

Part III

Systems

Now that we have covered some theoretical groundwork on how the human visual system controls eye movements on dynamic natural scenes, we are ready to move forward to the design and implementation of systems that can react to and ultimately guide eye movements.

Even with the continuous growth in hardware speed, high-resolution video processing still is computationally challenging, especially if low latencies in the single-digit range of milliseconds are required. In Chapter 5, we shall therefore discuss the software infrastructure that was created together with Martin Böhme to enable easy and efficient video handling and the synchronization of data streams (such as gaze data and video). A particular emphasis was put on the implementation of spatio-temporal multiresolution data structures.

The multiresolution pyramids will then be analysed in detail in Chapter 6. We shall start with the description of a gaze-contingent display that is based on a spatial Laplacian pyramid and capable of locally weighting individual frequency subbands. We shall then successively increase the complexity of the underlying pyramid and finally arrive at a system that is based on an anisotropic spatio-temporal Laplacian, where each spatio-temporal subband can be weighted locally. Because of the vastly increased computational cost of this pyramid (compared to the spatial Laplacian, complexity increases by a factor of about 40), this system was implemented on dedicated graphics hardware to meet real-time constraints. Due to further algorithmic improvements, the time required for gaze-contingent pyramid synthesis, which is critical for overall system latency (see Equation 5.1), was reduced to as little as 2 ms. Throughout Chapter 6, we shall also discuss results from experiments that evaluate the gaze-guiding effect of the respective gaze-contingent displays.

The work presented in Chapter 6 has been published in numerous places (Dorr, Böhme, Martinetz, and Barth, 2005a; Barth, Dorr, Böhme, Gegenfurtner, and Martinetz, 2006; Böhme, Dorr, Martinetz, and Barth, 2006b; Dorr, Vig, Gegenfurtner, Martinetz, and Barth, 2008; Jarodzka, Scheiter, Gerjets, van Gog, and Dorr, 2009; Dorr, Jarodzka, and Barth, 2010b; currently under submission is Jarodzka, van Gog, Dorr, Scheiter, and Gerjets, 2010b). The most complex gaze-contingent display that is based on a spatio-temporal Laplacian pyramid has not been described in a publication yet; a manuscript is in preparation.

"Make it work, make it right, make it fast."

Kent Beck

5 Software

Gaze guidance might lead to a deeper understanding of human vision and improved human-machine communication. Before we can aspire to reach these goals, however, we need to develop the software infrastructure that is necessary for low-latency processing of high-resolution video, and a major contribution of the work presented in this thesis is such development. For real-time gaze-contingent displays, this need is obvious; but even for the kind of offline analyses of gaze data on videos that we have presented in Part II, efficient algorithms are paramount. In Chapter 3, for example, smooth three-dimensional probability density functions that consume more than 2 GB of memory have to be computed for each realization of a leave-one-out validation scheme on more than 800 samples, and a full run of all analyses on all conditions and parameter sets takes almost 48 hours on a grid system with more than 50 nodes. In Chapter 4, image features are computed on all scales of a spatio-temporal multiresolution pyramid. The extension of multiscale methods into the temporal domain brings two problems. First, the complexity of neighbourhood-based algorithms grows from at least $O(n^2)$ to at least $O(n^3)$ due to the addition of an extra dimension. Second, temporal filtering of data with non-causal filters requires buffering of a suitable number of video frames and introduces a latency; to ensure temporal coherence of data buffers is not a trivial task. For example, to compute the structure tensor (see Chapter 2) on several temporal scales, a temporal shift between the original video and its multiresolution representation is first incurred during computation of the underlying Gaussian temporal pyramid (see also Chapter 6); then, computation of the first-order derivatives introduces a further latency (note that because of the different filter bandwidths relative to the original video, these latencies differ from scale to scale), and finally, another spatio-temporal filtering step, smoothing with ω, adds another source of latency that varies with scale. Therefore, a major goal

for the development of the `Data Source Framework` was to hide such intricacies of temporal synchronization from the user; a further, related problem that is addressed by the `Data Source Framework` is the simple access to data streams that concurrently produce data at different temporal resolution, such as (multiple) sources of gaze and video data.

"Ease of use" certainly is a design goal of many software frameworks, but the proof of the pudding is in the eating. Despite the difficulty of demonstrating this feature, we shall first give a quick tour of the `Data Source Framework`, and then conclude this chapter with some further considerations of hard- and software implementations.

It is important to note that the work presented here is the fruit of a very collaborative effort. Martin Böhme initiated the development of the software framework in 2003 and for the early years was the lead maintainer in the sense that every line the author of this thesis committed to the central repository was reviewed in painstaking detail, which certainly had a profound impact on overall code quality. Over the years, numerous students also have contributed code; of particular note, Sönke Ludwig laid the foundations for processing videos on Graphics Processing Units during work on his Bachelor's thesis.

5.1 Real-time video processing framework

We shall start this exposition of the `Data Source Framework` with the fundamental concept that gave rise to its name, namely that of `DataSources` and their corresponding `DataReaders`. A `DataSource<T>` is a templatized object that runs in a separate thread and continuously produces items of type `T`. The data types most often used in practice are `GazeCoord`, that is (x, y) pairs that represent gaze position with an associated confidence value (which is low, for example, when the eye tracker lost the eye due to a blink, or a binocular tracker could reliably track only one eye), and `ImageYUV420`, which represents a video frame in the most common colour space format with one brightness channel and two colour opponency channels at reduced resolution. Both these types can be decoded from files on hard disk or produced online (e.g. from a network stream of gaze data or a camera recording video frames), and many other data types, such as information on image features, can also be produced by a `DataSource`. An `Adapter` is derived from a `DataSource` and operates on its items; for example, this makes it possible to decompose an image into its frequency bands on a Laplacian pyramid. Because `DataSources` run in separate threads, this is a straightforward tool to distribute computational load on multicore systems.

Access to the items produced by a `DataSource<T>` is realized by means of a `DataReader<T>`. During initialization, a number of history and lookahead items that need to be available can be specified; a current item is always available. This makes temporal filtering trivial:

```
DataSource<GazeCoord> source;
// ...
DataReader<GazeCoord> reader(&source, history, lookahead);
// ...

// Five-tap box filter of horizontal gaze coordinate
double sum=0.0;
for(i=-history; i<=lookahead; ++i)
    sum+=reader.PData(i)->x/(history+lookahead+1);

fprintf(stdout, "x coordinate at %zd: %g\n",
    reader.GetTimeStamp(0).GetMicroseconds(), sum);
```

From this example, we can also see that each item that can be accessed by a `DataReader` has a time stamp associated with it. So far, the benefit of the `DataSource`–`DataReader` concept might not be intuitively clear, so let us look at the above piece of code using some example numbers. The eye tracker `source` might provide one gaze sample every millisecond, starting with the first sample at t=0 ms. Now, the first time `reader.PData(0)` is accessed, the gaze sample for t=2 ms will be returned (because before that, the two required history items cannot be provided), but only at t=4 ms (because the two lookahead items `reader.PData(1)` and `reader.PData(2)` need to be valid already). Moving forward the current item is achieved by a call to

```
reader.Advance();
```

and this will, transparently to the user, block the calling thread until at least t=5 ms. There are two modes to compute the current time t. In the default mode that is suited for real-time, online applications, t is simply linked to the hardware clock (for e.g. slow-motion playback of video, the hardware clock can also be multiplied with a constant). In the second mode, t is spun forward as fast as items can be produced; this is useful, for example, for offline analysis of gaze that can often happen at a much faster rate than the rate of the eye tracker during data collection. In both modes, however, synchronization of different `DataSources` is guaranteed because calls to `Advance()` block until all necessary items are produced; at the same time, only the calling thread blocks, so that all `DataSource` threads can continue to produce items independently.

Different sources can not only be synchronized by a balanced number of calls to `Advance()`, but also synchronized explicitly; in the following example, the eye tracker produces items much faster than the frame rate of the video and is always moved forward to the first gaze sample that has a time stamp right at or after the beginning of the current video frame:

```
DataReader<ImageYUV420> videoReader;
DataReader<GazeCoord>   gazeReader;

// ...

while(videoReader.Advance())
{
    gazeReader.AdvanceTo(videoReader.GetTimeStamp(0));

    // Print first gaze sample per frame
    fprintf(stdout, "%g %g\n",
        gazeReader.PData(0)->x, gazeReader.PData(0)->y);
}
```

We have successfully used these mechanisms to synchronize more than 1000 concurrent threads and to operate on 75 video streams (quasi-) simultaneously on hardware with up to 16 processor cores.

Another important feature of the `Data Source Framework` is the ability to encode and write videos to disk. Again, the user can write an arbitrary number of videos concurrently because each `VideoFileWriter` runs in a dedicated thread. Because video en- and decoding are based on the `FFMPEG` libraries (FFM, 2009), lossless compression of videos is supported, which is particularly useful for storage of sparse image features such as the geometrical invariants. Lossy video codecs, in contrast to this, are typically optimized towards natural scenes and therefore can introduce artefacts on sparse videos that may distort results.

As we have noted in the introduction to this chapter, operating on several scales of temporal multiresolution pyramids raises the problem of synchronized access. To ease the burden on the user, the `Data Source Framework` contains classes that encapsulate these problems. For example, the interface for `InvariantsStructureTensorPyramid`, which computes the structure tensor J on a spatio-temporal pyramid, looks as follows:

```
void Init(const Parameters &params);

int AddImage(const ImageYUV420 &img, const TimeStamp &t);

int Latency() const;

int EquilibriumLatency() const;

TimeStamp        CurrentTimeStamp() const;

const Image32F &Invariant(Type inv, int s, int t) const;
```

During initialization, the user can specify both the spatio-temporal smoothing kernel ω and the noise-reduction smoothing kernel that is applied before taking the derivatives as well as the desired number of spatial and temporal levels. Videos are fed to this class image by image with their corresponding time stamps using `AddImage()`, and the invariants H, S, and K can be retrieved at spatial scale s and temporal scale t using `Invariant()`. However, because of temporal filtering operations with non-causal kernels, there is a delay between the time stamp of the most recently added image and the point in time for which `Invariant()` returns image features, which can be obtained by `CurrentTimeStamp()`. This delay corresponds to `Latency()` many video frames; this means that only `Latency()` many calls to `AddImage()` after the video started, any (non-black) response can be obtained (for a detailed account of this latency, see the following chapter). However, it takes even more time to fill the underlying temporal pyramid with valid history items, and thus it takes `EquilibriumLatency()` many frames before temporal border effects cannot be observed anymore.

Implementation details

In the `Data Source Framework` images can either be held in main memory and be operated upon by the Central Processing Unit (CPU), or they can be represented as textures in graphics memory and be operated upon by the Graphics Processing Unit (GPU). In the former case, `ImageOf<T>` is an image structure that is based on the `Intel OpenCV Computer Vision Library` (Bradski et al., 2008), templatized with the bit depth of each pixel, and a collection of stateless functions in a class called `ImageOps` that operate on these images. `ImageOps` started out as a simple wrapper for basic image processing functions from `OpenCV`, but over time, many functions were replaced with more efficient implementations from the `Intel Performance Primitives` and

the AMD counterpart `FrameWave`. Functions that turned out to be critical for overall performance were implemented in hand-written assembly using the SSE vector extensions that can be used to perform operations on several pixels at once. For example, computation time for the geometrical invariants was reduced by more than 50% by replacing a high-level implementation of the normalization function that raised pixel values to the 1/8th power and mapped them to $[0, 255]$ (see Chapter 4). The normalization function itself was sped up by a factor of 125 by replacing C++ with SSE code, replacing the power function by three square roots, and the square roots with reciprocal square roots (which are faster because of their reduced accuracy of 12 bits; this is sufficient for normalization because the output video has 8 bits accuracy only).

Videos are commonly stored in the $Y'C_rC_b$ colour space (Poynton, 2003), where only the luminance channel (Y') is encoded at full spatial resolution and two colour opponency channels are encoded at half resolution. Compared to an RGB representation, frame size is reduced by 50%, and this also reduces computational cost by 50%. Therefore, the `Data Source Framework` operates on $Y'C_rC_b$ images; the conversion to the RGB colour space is performed in hardware using `XVideo` or `DirectDraw` functions right before display on the screen only.

As a recent, more powerful alternative to image processing on the CPU, we have also implemented our algorithms on the GPU. GPUs are geared towards highly parallel, throughput-oriented processing and have limited flexibility compared to CPUs (recently, however, the distinction has become blurrier with the advent of GPGPU, General-Purpose computation on Graphics Processing Units). They consist of an array of up to hundreds of so-called shader units, which all execute the same small *kernel* in parallel on one pixel each with the position of the pixel in the image as a parameter. Obviously, this is suited ideally for gaze-contingent displays, where each output pixel is a function of its position relative to gaze. Several high-level programming languages for GPUs have recently become available; the `Data Source Framework` uses Cg (C for graphics, Mark et al., 2003). A toy example that illustrates a shader kernel in Cg is as follows:

```
struct foutput { float4 color : COLOR; };

float2 random(float distance) { /* ... */ }

foutput main(float2 texCoord : TEXCOORD0,
        uniform sampler2D input : TEX0,
        uniform float2 gazePos)
{
    foutput OUT;

    float       d = distance(texCoord, gazePos);
    float2 offset = random(d);

    OUT.color     = tex2D(input, texCoord+offset);

    return OUT;
}
```

The Cg built-in function `distance` computes the length of its input vector in just one shader clock cycle. `random(d)` is a putative function that produces a random perturbation vector `offset` whose length increases with `d`. `tex2D` is a texture lookup function and here for the output pixel at position `texCoord` does not look up the input texel at the same position, but at a position perturbed by `offset`. In effect, this small example program scrambles pixels of an image locally, with a small scrambling distance at fixation and a larger distance in the periphery.

To make Cg kernels as the above example usable from C++ code, we implemented a `TextureOperator` class to which a kernel can be associated. Textures are images that were uploaded to the graphics memory; to the CPU, they are only references afterwards. Any texture reference can then be passed to the `TextureOperator` to have the kernel executed on each of its pixels. In principle, this emulates the state-less functions in `ImageOps` for CPU-based image processing; in practice, however, the very different programming model for the GPU does require some algorithmic changes, especially if the higher theoretical throughput of the GPU should be fully exploited.

One major difference is that GPU shader units are optimized to process tuples of red, green, blue, and alpha channels simultaneously. As we have noted above, however, videos are stored in the memory-saving $Y'C_bC_r$ format, which is not natively supported by GPUs. Especially for temporal pyramids with high memory requirements, this is a drawback of GPUs. If memory is not

the limiting issue, one can simply convert $Y'C_bC_r$ images to RGB right before or after texture upload to graphics memory; if, on the other hand, speed is not critical, each colour channel can be stored as a separate texture. Even though this reduces the overall memory bandwidth that is required, the three-fold increase in shader passes can significantly affect system performance because of the communication overhead between CPU and GPU (see measurements in Section 6.7).

However, the fact that GPUs are designed specifically for image processing and for rendering textures with varying resolution obviously also offers benefits. For example, texture lookups can be interpolated in hardware, so that an n-tap binomial filter kernel can be realized with only $n - 1$ texture accesses. Furthermore, in the next chapter we shall see that gaze-contingent displays often use so-called resolution maps that specify filter parameters for each pixel. On the GPU, these maps can be stored at the minimum resolution required (for example, one coefficient per degree of visual angle), and hardware interpolation ensures that the values from these maps still vary smoothly for each pixel of the output image. This saves memory bandwidth and the cost of computing the resolution map at full resolution; on the CPU, a coordinate conversion for each pixel access would be prohibitively expensive.

5.2 Latency

In this section, we shall briefly estimate the system latency of a gaze-contingent display, that is the time between an eye movement and the appearance of the corresponding display change on the screen. Obviously, we would like to reduce latency as much as possible; from work on saccadic suppression (see Section 2.15), we know that visual sensitivity is reduced during saccades and about 20–50 ms after the end of a saccade, so a maximum latency of 20 ms is desirable. We here assume a setup where eye tracker and display workstation are two independent computers connected via ethernet.

The end-to-end latency of the whole system can be estimated as

$$\tau = \tau_{\text{tracker}} + \tau_{\text{net}} + \tau_{\text{collect}} + \tau_{\text{imgproc}} + \tau_{\text{display}}.$$

The latency of the SMI iViewX Hi-Speed eye tracker running at 1250 Hz is specified with 1 ms. The network latency τ_{net} can be assumed to be well below 1 ms for a dedicated Gigabit Ethernet link, where average "ping" round-trip times are around 0.2 ms. On the display workstation, gaze information is immediately collected by a separate thread (see discussion of `DataSources` above), but the display thread might use this information with delay τ_{collect}

only. Because the display thread is synchronized with the vertical retrace signal of the display, $\tau_{\text{collect}} < 1/f_{\text{dsp}}$, with a display refresh rate f_{dsp} that is 120 Hz in our setup. Image processing latency τ_{imgproc} will be discussed in Chapter 6 and is on the order of 2–10 ms. The display uses double buffering to avoid tearing effects. Therefore, for every vertical refresh, fore- and background buffers are swapped. After a buffer swap, the image is finally drawn to the screen by an electron beam (on a CRT) that traverses the frontal glass pane of the screen from the top left to the bottom right corner. The contents of the bottom right corner are therefore updated only at the very end of a vertical screen refresh cycle, and $\tau_{\text{display}} \in [0, 2/f_{\text{dsp}}]$. At f_{dsp}=120 Hz, the overall system latency can now be estimated to $\tau = 27\,\text{ms} + \tau_{\text{imgproc}}$. In practice, concurrent system activities might make this estimate slightly too optimistic. For example, during image processing on the GPU, commands may be buffered and thus not executed immediately. On the other hand, we can impose an upper bound on τ_{collect} by detecting saccades online. Because it is particularly critical to react after a saccade has ended (and a new fixation has begun), the display thread can spin-wait for saccade offset, thus reducing τ_{collect} to about 1 ms, and

$$\tau = 20\,\text{ms} + \tau_{\text{imgproc}}, \tag{5.1}$$

which just meets our goal of 20 ms overall latency.

From these observations, we can see that a large part of system latency is determined by the refresh rate of the display hardware, which currently is still limited to 120 Hz both for TFTs and for CRTs (at high resolution). Because the production of display panels is very expensive and the number of vision science laboratories is comparatively small, we here can only hope for faster consumer electronics devices eventually becoming available. TV sets with up to 200 Hz panels are on the market already, but they accept input signals only at 50 Hz and interpolate to generate intermediate frames, which in fact introduces even further delays. Therefore, our only means of substantially lowering system latency is a reduction in image processing latency.

5.3 Chapter conclusion

In this chapter, we have presented some key concepts of the software infrastructure that was built to enable the development of gaze-contingent displays that process high-resolution videos in real time. Despite its powerful and flexible abstraction, efficiency was one of the major implementation criteria. In particular, we exploited parallelism on all levels: instruction level parallelism is a feature of all modern CPUs. Parallelism on the data level was exploited by

hand-writing assembler routines for CPU vector extensions and by implementing image processing routines on highly parallel graphics hardware. At the thread level, the `DataSource` concept ensured that expensive operations such as lossless video de- and encoding scale easily with the number of CPU cores in a system; we have successfully run programs with more than 1000 threads on 16-core hardware. Not described in detail in this chapter was the use of parallelism at the cluster level. Up to almost a hundred computer nodes at the Institute for Neuro- and Bioinformatics were employed during data analysis and machine learning using the Portable Batch System (PBS, 2009).

In summary, building systems for gaze guidance is a technical challenge because of the strict real-time constraints. We have invested a major effort into the development of efficient image processing algorithms to face this challenge.

"To find out what happens to a system when you interfere with it you have to interfere with it (not just passively observe it)."

George Edward Pelham Box

6

Space-variant filtering and gaze-contingent displays

In this chapter, we shall present a series of successively more complex approaches to efficient space-variant filtering. Gaze-contingent displays perform space-variant filtering in real time, and we need this for steps ii and iii of our gaze-guidance strategy that we outlined in Chapter 1: i) predict a set of candidate points where a subject will look next, based on the video input and current gaze position; ii) increase the probability for one candidate point to be attended next by increasing image-based saliency there; iii) decrease saliency everywhere else.

The algorithms that we shall present in the following were inspired by the work of Geisler and Perry (2002), who were the first to implement a gaze-contingent display to simulate smooth visual fields. They created a Gaussian multiresolution pyramid for each image of an input video in real time; instead of individually filtering each pixel of the output image in retinal coordinates, i.e. relative to gaze position, they computed output pixels by interpolating between two adjacent pyramid levels to efficiently approximate any desired filter bandwidth. The *resolution map* that assigned a filter bandwidth to each retinal location was precomputed; for example, the input image was strongly lowpass filtered in a certain retinal region to simulate a scotoma.

We will now extend this concept of assembling an output image from the levels of a multiresolution pyramid in real time to various pyramid types, and perform first gaze-guidance experiments with the thus obtained gaze-contingent displays.

Following a brief overview of gaze-contingent displays in general, we shall discuss a gaze-contingent display based on a spatial Laplacian pyramid. The extension from an underlying Gaussian to a Laplacian pyramid significantly

increases the computational cost of the system, but instead of mere lowpass filtering allows to individually weight frequency bands. As a further modification, the resolution map for this gaze-contingent display was not precomputed, but updated after each saccade. This modification makes a very low latency crucial, because the human visual system is highly sensitive to the temporal transients induced by the sudden on- and offsets of filtering results, i.e. contrast changes; these changes remain invisible only if they still take place during saccadic suppression. Results obtained with this gaze-contingent display were published in (Dorr, Vig, Gegenfurtner, Martinetz, and Barth, 2008).

Then, we shall review the adaptation of Geisler and Perry's algorithm to the temporal domain that was developed and implemented by Martin Böhme (Böhme, Dorr, Martinetz, and Barth, 2006b) in detail to facilitate a later comparison with improved algorithms. In analogy to the simulation of spatial visual fields, i.e. foveation, this gaze-contingent display was termed "temporal foveation". Using this display, we could show that peripheral temporal blur is hardly noticeable (Dorr, Böhme, Martinetz, and Barth, 2005a), and that eye movement characteristics change when a movie is displayed with temporal filtering in the periphery (Barth, Dorr, Böhme, Gegenfurtner, and Martinetz, 2006).

Taking a step back from gaze-contingent real-time applications, we shall then discuss how (offline) space-variant filtering based on an isotropic spatio-temporal Laplacian pyramid can be used for the visualization of expert's eye movements during training of novices. Those regions that were not attended by the expert are reduced both in their contrast and their colour saturation, so that the novices' gaze is drawn towards the relevant movie regions. In subsequent tests, novices who received such gaze guidance during training perform better than novices without gaze guidance. The perceptual learning experiments were designed and carried out by Halszka Jarodzka, Knowledge Media Research Center, Tübingen. Initial results have been outlined in (Jarodzka, Scheiter, Gerjets, van Gog, and Dorr, 2009), the video processing algorithm is described in detail in (Dorr, Jarodzka, and Barth, 2010b), and a manuscript on the experimental data is currently under submission (Jarodzka, van Gog, Dorr, Scheiter, and Gerjets, 2010b).

Finally, we shall present a gaze-contingent display that is based on an anisotropic spatio-temporal Laplacian pyramid. The spatio-temporal Laplacian pyramid used for gaze visualization above is too computationally expensive to be used in a gaze-contingent fashion; we therefore developed a modified approach to efficiently compute spatio-temporal subbands. We also improved upon the temporal upsampling algorithm first used for temporal foveation above. Despite these improvements, the computational complexity

of this display is still too great for conventional computer workstations; we therefore implemented the system on dedicated graphics hardware (the GPU). The first such implementation was undertaken by Sönke Ludwig as part of his undergraduate thesis.

6.1 Gaze-contingent displays

The properties of the human visual system vary significantly across the visual field. The density of photoreceptors is much higher in the centre of the retina than in the periphery, and about 50% of visual cortex are devoted to the processing of input from the central 2% of visual field (Wandell, 1995). Consequently, humans move their eyes about two to three times per second to successively sample a visual scene with the region of highest acuity, the so-called fovea. Gaze-contingent displays render their contents as a function of where the user is looking, using real-time information about eye position gained by an eye tracker. Some systems simply mask parts of the visual field, e.g. the first gaze-contingent displays used in reading research (McConkie and Rayner, 1975; Rayner, 1975, 1998) to investigate the perceptual span, or to simulate scotomata to study visual search strategies (Cornelissen et al., 2005). Another class of displays takes advantage of the reduced visual sensitivity during saccades and modifies some property of the scene whenever the subject moves their eyes; these displays have been used for research on change blindness (Henderson and Hollingworth, 1999), transsaccadic integration (Germeys et al., 2004), and saccadic adaptation (Garaas et al., 2008). Rucci et al. (2007) developed a gaze-contingent system with very low latency to artificially stabilize an image on the retina and studied the role of fixational eye movements. Furthermore, gaze-contingent displays can be used to aid the user in making the right eye movements to improve visual communication (McNamara et al., 2008; Barth et al., 2006). In a broader sense, gaze-controlled games can also be understood as gaze-contingent displays (Dorr, Böhme, Martinetz, and Barth, 2007; Dorr, Pomarjanschi, and Barth, 2009b; Smith and Graham, 2006; Isokoski and Martin, 2006); in the human-computer interaction field, several gaze-contingent techniques have been developed to use gaze as an input modality or for improved visibility of user interface elements (Dorr, Rasche, and Barth, 2009c, Jacob, 1993; Istance et al., 2008). For a more detailed review, we refer to e.g. Duchowski et al. (2004) and Reingold et al. (2003).

The category of gaze-contingent displays that is most relevant to the present work exploits the space-variant spatio-temporal properties of the visual system. For computer-generated content, information about gaze can be used to reduce the level of detail at which the periphery is rendered to achieve higher rendering

throughput (Parkhurst and Niebur, 2004; Duchowski et al., 2009); in a similar fashion, video transmission systems can reduce bandwidth requirements for given, non-rendered images by streaming only the fixated image region at full resolution (Geisler and Perry, 1998; Perry and Geisler, 2002; Sheikh et al., 2003). Such space-variant filtering in retinal coordinates has been termed "foveation" and its psychophysical effects have been studied empirically (Loschky and McConkie, 2000; Loschky and Wolverton, 2007; Loschky et al., 2005; Geisler et al., 2006; Dorr et al., 2005a); without the real-time constraints, similar algorithms have also been used in order to accurately model the input to the oculomotor system (Itti, 2006; Rajashekar et al., 2007). For alternative strategies towards space-variant filtering, we refer to e.g. Hua and Liu (2008), who implemented foveation in an optical system, or Tan et al. (2003).

6.2 Real-time spatial Laplacian pyramid

In this section, we shall present a modification of the gaze-contingent display by Geisler and Perry (2002). Our modification uses a spatial Laplacian instead of a spatial Gaussian pyramid to be able to specify weights for individual frequency bands instead of the specification of a cutoff frequency for a lowpass filter only. In principle, the possible use of a Laplacian has been mentioned already in the original paper; in practice, this modification significantly increases computational complexity for two reasons. First, computing the Laplacian is obviously more costly because computation of the Gaussian is one part of Laplacian pyramid analysis; for each downsampling operation, one additional upsampling operation and one subtraction at the higher resolution are necessary. A second performance issue arises because of the need to store signed intermediate results, i.e. the need of a sign bit without losing precision; in practice, this leads to a doubling of the data type width from 8 bits to 16 bits, so that required memory bandwidth is also doubled (plus an additional overhead because of the limited support of signed words in the SIMD instruction set).

The main idea of this algorithm is to analyse and synthesize a Laplacian pyramid in real time; the individual levels are weighted as a function of gaze position during the synthesis phase. We thus extend Equation 2.3, which describes the pyramid synthesis phase, and introduce a space-variant weighting function α for each level l:

$$L'_l(x,y) = \alpha_l(g_x, g_y, x, y) \cdot L_l(x,y) + \uparrow L_{l+1}(x,y),$$

with gaze position (g_x, g_y).

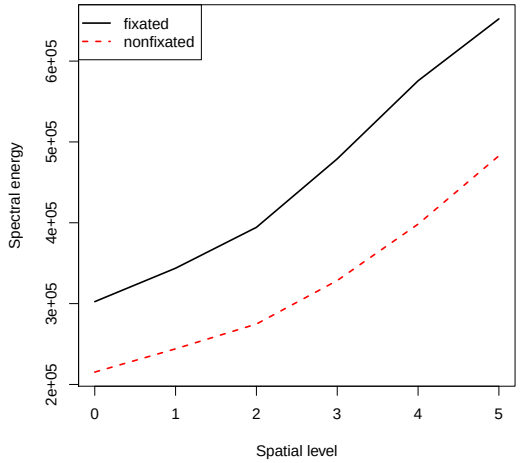


Figure 6.1: *Distribution of spectral energy across spatial frequency bands for fixated and non-fixated locations. Clearly, spectral energy is higher in image regions that drew attention. Energy per stimulus area here was computed in a neighbourhood of two degrees diameter around each fixation point; the non-fixated class comprised of all fixations that were made on different movies and thus represents the central bias.*

An analysis of the average local spectral energy at fixated and non-fixated image regions showed that fixated locations have a much higher spectral energy (see Figure 6.1). The rationale for the experiment that shall be described in the following was therefore to reduce spectral energy at a subset of likely fixation locations, and thus change the saliency distribution of the scene in real time.

We used six out of the 18 high-resolution natural videos from our data set presented in Chapter 3. Because of real-time constraints, we reduced their resolution from 1280 by 720 pixels to 1024 by 576 pixels. For each frame of the input movies, we determined up to 20 candidate locations that were likely to be fixated. In principle, we could have chosen to use our gaze prediction algorithm for this purpose (see Chapter 4); because we were here interested in testing our gaze-contingent display under optimal conditions, we used the fixation data from our 54 subjects described in Chapter 3. We computed a probability density for each frame by placing a two-dimensional Gaussian with standard deviation 0.75 deg at each gaze sample and normalizing the superposition of all these Gaussians to unit sum; then, we iteratively extracted the maximum and suppressed the location of that maximum by lateral inhibition with an inverted Gaussian of standard deviation 2.35 deg.

Under the assumption that these eye movements had been driven by image features, we had to account for the oculomotor latency between a gaze-

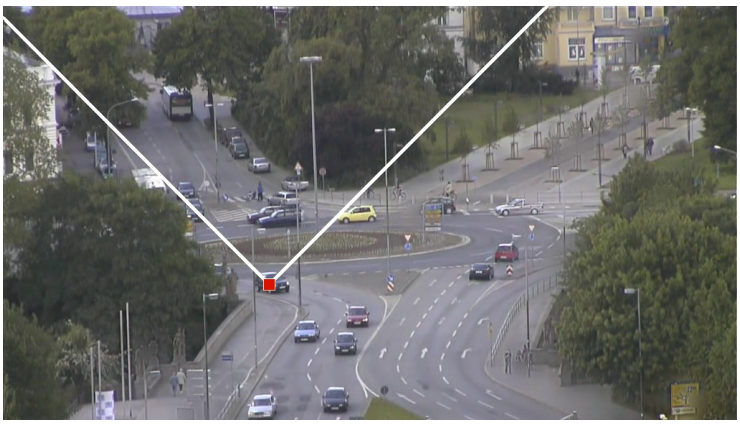


Figure 6.2: *Example stillshot of gaze-contingent display based on a real-time spatial Laplacian pyramid. The red marker (gaze position) and the white lines were not shown during the experiment and serve illustrative purposes only. In one randomly chosen quadrant of the visual field (indicated by the white lines), contrast remained as in the original video; in the remaining three quadrants, spatial contrast was reduced at up to 20 candidate locations that were likely fixation targets (for example, note the street sign or the pedestrians bottom right).*

capturing event and a saccade landing at its location. Therefore, the candidate points were shifted backwards in time by 100 ms. This number was a compromise between the about 200 ms that are the typically recorded oculomotor latency with synthetic stimuli, and the 70 ms that we obtained as the average time shift between dynamic features and eye movements (see Chapter 4).

We implemented a `Data Source Framework` adapter for the Laplacian analysis phase (see previous chapter), so that video decoding and pyramid analysis were computed in dedicated threads and decoupled from the main display thread; therefore, image processing latency between an eye movement and a change in the display was only affected by the gaze-contingent synthesis (if it was ensured that the display thread was never preempted; we achieved this by assigning appropriate priorities to all threads). Performance measurements on a 3 GHz Pentium 4 showed that the space-variant pyramid synthesis of a pyramid with five levels took about 10 ms on the videos with 1024 by 576 pixels.

Instead of shifting a precomputed resolution map around with gaze position as in Geisler and Perry (2002), we updated the resolution map after each saccade. Once a saccade offset was detected (see Section 3.3), one quadrant of the subject's visual field was chosen randomly (see Figure 6.2). In the remaining three quadrants, all candidate points outside a radius of five degrees around centre of gaze were modified in their saliency as follows.

The spectral energy $E_{i,k}$ in a neighbourhood around each candidate point i was computed for each Laplacian scale k,

$$E_{i,k} = \sqrt{\sum_{m=-c}^{c} \sum_{n=-c}^{c} L_k^2(x_{i,k} - m, y_{i,k} - n)}$$

and if this energy was higher than the average energy of non-fixated locations E_k^{nonfix}, spectral energy was reduced by a multiplication of that neighbourhood with a factor that was smaller than one,

$$\alpha'_k(x_{i,k}, y_{i,k}) = \begin{cases} 1 & E_{i,k} \leq \theta E_k^{\text{nonfix}} \\ \theta \cdot E_k^{\text{nonfix}} / E_{i,k} & \text{otherwise} \end{cases}$$

We here used a neighbourhood size of $2c + 1 = 9$ pixels that was held constant over spatial scales; thus, this corresponded to about 3.4 deg on the fourth scale, and due to the multiresolution pyramid, the transition between unmodified and modified areas was a smooth gradient instead of a sharp edge. The threshold θ that determined how far energy should be decreased was set to 1.2 after the informal observation that a lower threshold led to a peripheral visibility of the modified locations. On average, this meant a reduction factor of 1.6 for the spectral energy of candidate locations. In order to avoid a highly noticeable change in local brightness, the lowest level of the Laplacian, which represents the DC component, was not modified.

Twelve subjects took part in the experiment; the physical setup was the same as in Chapter 3. The hypothesis that drove this experiment was that the reduced spectral energy at some candidate points would render these candidate points less likely to become fixated. If this had been the case, we would have registered less saccades that went into the modified quadrants, and thus more than 25% of eye movements into the unmodified quadrant. Unfortunately, subjects reported having seen occasional flicker in the periphery; this indicates that the graphics update after a saccade was at least sometimes too slow and a temporal transient became visible to the subjects at exactly those locations that we wanted to reduce in their conspicuity. We believe it is due to this unintended increase rather than decrease in saliency that the distribution of fixations over the quadrants did not change overall, i.e. only 25% of eye movements landed in the unmodified quadrant.

Even though no gaze-guiding effect could be found, one interesting effect of the gaze-contingent stimulation is shown in Figure 6.3. The number of saccades per second is significantly reduced on the gaze-contingent display. Two possible explanations can be given for this phenomenon. One possibility

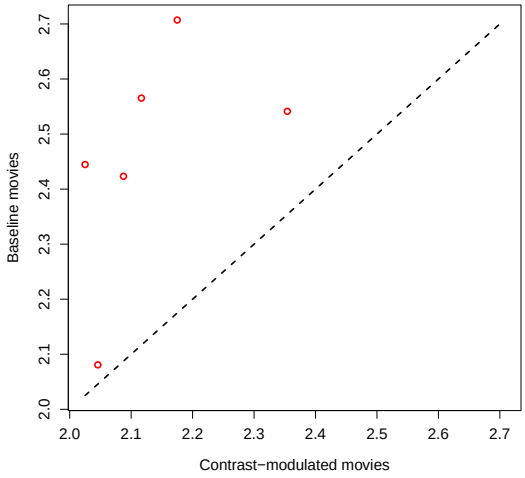


Figure 6.3: *Saccade rates per second for movies that were shown with gaze-contingent spatial contrast modification and for original movies. Saccade rate is significantly reduced by the contrast modification ($p < 0.032$, paired Wilcoxon's test).*

is that temporal transients early during fixation increased fixation duration, an effect that is known from reading research (Yang, 2009); in those experiments, however, temporal transients were introduced much later during fixation. The other possibility is that the reduced amount of contrast in the overall scene led to a lower saccade rate.

In conclusion, we can state that the gaze-contingent display that modified spatial spectral energy suffered from the restriction to the spatial domain and, despite our efforts to implement a low-latency system, a failure to make saliency changes happen invisibly, i.e. directly after fixation onset. In the following, we shall therefore present algorithms that also modify the temporal domain and faster implementations.

6.3 Temporal filtering on a Gaussian pyramid

In this section, we shall extend the original pyramid-based gaze-contingent display by Geisler and Perry (2002) to the temporal domain. Two major challenges have to be met for this extension. First, computational costs increase by about one order of magnitude, and second, a suitable buffering scheme has to be developed because videos, in contrast to static images, usually cannot be held in memory completely.

We shall first look at the creation of a canonical temporal Gaussian pyramid where resolution is successively reduced on the lower levels. Ultimately, we

want to blend different levels of such multiresolution pyramid to create an output image with space-variant temporal resolution. However, because we want to perform such blending in every time step, even the lower pyramid levels need to be updated in every time step and cannot be kept at reduced resolution anymore. We shall therefore describe a scheme to iteratively interpolate the lower pyramid levels to full resolution; because interpolation cannot add information, these upsampled levels still have lower cutoff frequencies than the original video.

Notation

The input video is given as an image sequence $I(t)$. Images have a size of W by H pixels and represent a single colour channel; for videos with several colour channels, each channel can be filtered separately. Operations on entire images, such as addition, are to be applied pixelwise to all pixels in the image. The individual levels of the multiresolution pyramid are referred to as G_0 to G_N (for a pyramid with $N+1$ levels). $G_k(n)$ refers to the n-th image at level k. Because of the temporal downsampling, lower levels have fewer frames, so that $G_{k+1}(n)$ corresponds to the same point in time as $G_k(2n)$. For time steps t that are not a multiple of 2^N, not all pyramid levels have a corresponding image $G_k(t/2^k)$; we use C_t to denote the highest index of levels with valid images at time t (in the implementation, these are the levels that have changed at time t and need to be updated), i.e. C_t is the largest integer with $C_t \leq N$ and $t \bmod 2^{C_t} = 0$.

To interpolate lower levels G_k, $k > 0$, back to full temporal resolution, we introduce upsampled levels U_k^l with intermediate resolutions, so that U_k^k has the same frame rate as G_k and U_k^0 has the same frame rate as G_0. However, all U_k^l have the same spectral content as G_k.

Downsampling

$G_{k+1}(n)$ is obtained by low-pass filtering images in G_k and discarding every second frame. In practice, only every second frame has to be computed:

$$G_{k+1}(n) = \sum_{i=-c}^{c} w_i \cdot G_k(2n-i) \Big/ \sum_{i=-c}^{c} w_i \; .$$

The $w_{-c}, \ldots, w_c$ are the kernel coefficients. We use a binomial filter with $c = 2$, $w = (1, 4, 6, 4, 1)$. As we shall see later, the use of a symmetric, i.e. non-causal filter kernel requires the use of video frames with future time stamps, or lookahead frames. This makes this algorithm usable only for pre-recorded

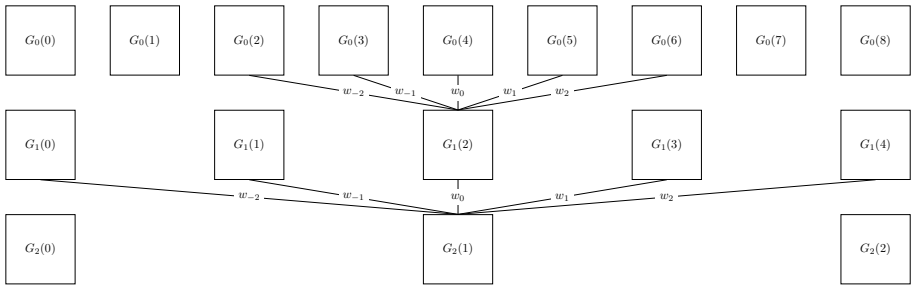


Figure 6.4: *Temporal Gaussian pyramid with three levels and $c = 2$. Lower levels have a reduced frame rate. Note that history and lookahead video frames are required because of the temporal filter with symmetric kernel, e.g. computation of $G_2(1)$ depends on $G_1(4)$, which in turn depends on $G_0(6)$.*

video sequences, but the use of non-symmetric filter kernels, which do not require future items, would lead to phase shifts and thus visible artefacts.

A schematic illustration of the downsampling phase is shown in Figure 6.4.

Upsampling

Interpolation to full temporal resolution is achieved by iteratively upsampling by a factor of two until full resolution is reached and storing the intermediate results in U_k^i. For level G_k, we start with

$$U_k^k(n) = G_k(n)$$

and then upsample by inserting zeros and a subsequent lowpass filtering:

$$U_k^l(n) = \sum_{i \in P(n)} w_i \cdot U_k^{l+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i \in P(n)} w_i \,, \tag{6.1}$$

with an index function $P(n) = \{j = -c, \ldots, c \,|\, (n-j) \bmod 2 = 0\}$ that lists the valid images that are available on the lower level. A schematic overview is shown in Figure 6.5.

Number of lookahead frames From Figures 6.4 and 6.5, it is clear that the temporal filtering uses video frames both from past and future time steps. To estimate the number of video frames that need to be held in memory, we first analyse the upsampling phase; the required number of history and lookahead items during the downsampling phase follows from that. We first note that to produce $U_k^0(t)$, we need to already have produced frames $U_k^1\left(\frac{t}{2} - \frac{c}{2}\right)$ to $U_k^1\left(\frac{t}{2} + \frac{c}{2}\right)$, i.e. we have $\frac{c}{2}$ history and $\frac{c}{2}$ lookahead items. For simplicity, we can achieve the same result by using zero history and c lookahead, and then

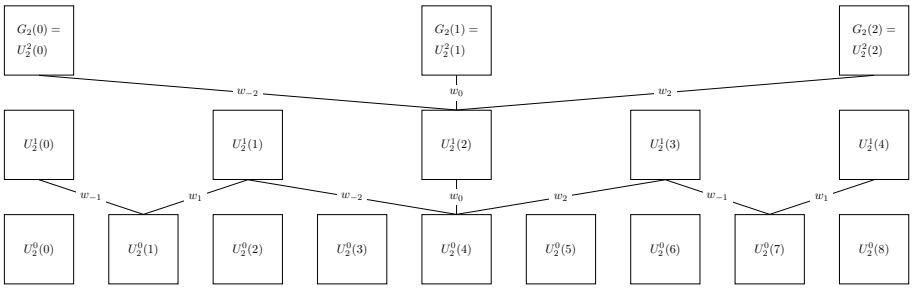


Figure 6.5: *Schematic view of interpolating the third level G_2 of a temporal Gaussian pyramid back to full temporal resolution. In each interpolation step, resolution is doubled by inserting zeros and a subsequent lowpass filtering. Note that similar to the downsampling step, upsampling requires history and lookahead items to be available; for example, computation of $U_2^0(4)$ requires $U_2^1(3)$ to be known. This, in turn, requires $U_2^2(2)$, which corresponds to the same point in time as $U_2^0(8)$.*

repeat the same argument as above: e.g. to compute $U_k^1\left(\frac{t}{2}+c\right)$, we need to know $U_k^2\left(\frac{t}{4}+\frac{c}{2}-\frac{c}{2}\right),\ldots,U_k^2\left(\frac{t}{4}+\frac{c}{2}+\frac{c}{2}\right)$. We can therefore conclude that zero history and c lookahead items are sufficient for all levels.

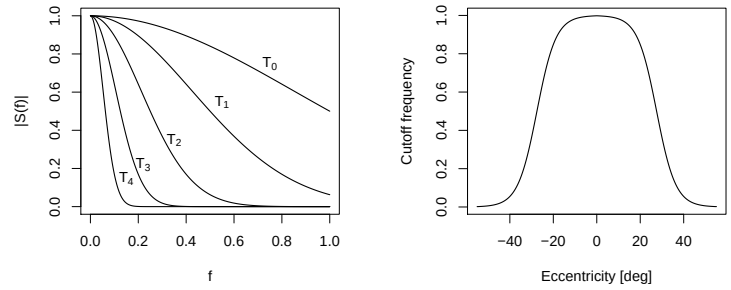
With these constraints imposed by the upsampling phase, we can now address the history and lookahead requirements of the downsampling phase. On the lowest level, we need to have available frames $U_N^N\left(\frac{t}{2^N}\right)=G_N\left(\frac{t}{2^N}\right)$ to $U_N^N\left(\frac{t}{2^N}+c\right)=G_N\left(\frac{t}{2^N}+c\right)$. We note that in time step t, we only need to update $G_N\left(\frac{t}{2^N}+c\right)$, so that we need frames $G_{N-1}\left(\frac{t}{2^{N-1}}+2c-c\right)$ to $G_{N-1}\left(\frac{t}{2^{N-1}}+2c+c\right)$, i.e. a history of zero and a lookahead of $3c$ items. We can repeat this argument and obtain a lookahead λ_k for level k of

$$\lambda_k=(2^{N-k+1}-1)\cdot c,$$

and the lookahead λ_0 on the highest level, i.e. the overall latency of the pyramid, is thus $(2^{N+1}-1)\cdot c$.

Space-variant temporal filtering

Now that we have interpolated all pyramid levels to full temporal resolution, we can blend them in a gaze-contingent fashion. We first define a *resolution map* $\alpha(x,y)$ that specifies the desired cutoff frequency for each pixel in retinal coordinates. Because gaze position usually does not coincide with the centre of the screen, but can also be in any corner of the display, the size of the resolution map must be almost twice that of the video, with $-(W-1)\leq x\leq(W-1)$ and $-(H-1)\leq y\leq(H-1)$. The values that are stored in α range from 0 to 1 and represent the cutoff frequency for each pixel relative to the highest resolution, so that a pixel (x,y) with $\alpha(x,y)=1$ should be taken directly from



(a) (b)

Figure 6.6: *(a) Transfer functions of pyramid levels G_l. Lower pyramid levels have successively lower cutoff frequencies. (b) Example resolution map. Full temporal resolution is retained at the centre of fixation (0 deg eccentricity) and falls off steeply towards the periphery.*

the highest level, $\alpha = 0.5$ corresponds to the second level, etc. Values that are not a power of two are interpolated between the two levels that bracket this resolution; even though the lowest level N typically is not the true DC component, but still contains a considerable portion of the frequency spectrum, resolution values smaller than 2^{-N} correspond to that lowest level, because no further downsampled level is available.

For the interpolation of adjacent pyramid levels, the transfer function T_l of each pyramid level G_l can be approximated by

$$T_l(r) = \mathrm{e}^{-r^2/\left(2\sigma_l^2\right)},$$

with r as relative resolution and $\sigma_l^2 = 1/(2^{2l+1}\ln 2)$. The first few T_l are plotted in Figure 6.6(a); for further details, we refer to Perry and Geisler (2002).

Based on these transfer functions, we obtain the blending function B_l that specifies the weight for level U_l^0 at each pixel; note that the contribution of U_l^0 is non-zero for at most two (adjacent) frequency bands:

$$B_l(x,y) = \begin{cases} \frac{\frac{1}{2} - T_{l+1}(\alpha(x,y))}{T_l(\alpha(x,y)) - T_{l+1}(\alpha(x,y))} & 2^{-(l+1)} < \alpha(x,y) < 2^{-l} \\ 1 - B_{l-1} & 2^{-l} \le \alpha(x,y) \le 2^{-(l-1)} \\ 0 & \text{otherwise} \end{cases}$$

with an additional constraint that $B_N = 1$ if $\alpha < 2^{-N}$. The output image $O(t)$ can now be computed simply by adding up the interpolated levels weighted

Algorithm 1 Pseudocode for gaze-contingent temporal filtering and rendering.

Input: t Time step

Downsampling step: update pyramid levels $G_0, \ldots, G_{C_t}$
Upsampling step: update $U_k^{l(k)}$, with $k \leq C_t, l(k) = \{0, \ldots, k\}$
Get current gaze position $(g_x(t), g_y(t))$
Compute output image $O(t)$
Display image $O(t)$

with the blending function,

$$O(t)(x,y) = \sum_{l=0}^{N} B_l(x - g_x(t), y - g_y(t)) \cdot U_l^0(t)(x,y).$$

To summarize, the complete algorithm for gaze-contingent temporal filtering is listed in Algorithm 1. From this algorithm, it can be seen that the current gaze position needs to be obtained only right before blending the pyramid levels to create the output image. The critical system latency from a change in gaze position to a display update is therefore determined only by the weighted addition of a small number of video frames; the downsampling and in particular the upsampling phase, which require significantly more computation, can be performed independently in the background. It can also be seen that the number of levels G and U that need to be updated in each time step t varies with t; with a suitable buffering scheme, the computational load can be distributed over more time steps.

An example resolution map, where $\alpha(r)$ is a sigmoidal function of radius r relative to the maximum eccentricity is

$$\alpha(r) = \frac{1}{2} - \frac{1}{2}\tanh(2\pi r - \pi),$$

which is also plotted in Figure 6.6(b). In a loose analogy to the well-established falloff in spatial acuity across the retina, this example resolution map retains full temporal resolution in the centre, i.e. at the fovea, and introduces temporal blur steeply with eccentricity. An example stillshot of a video together with its temporally filtered counterpart is rendered in Figure 6.7.

Downsampling memory requirements On each level k, we need to store λ_k lookahead items and the current image, so that the overall number $\text{frames}_{\text{down}}$ of full-size video frames to be stored for the creation of a temporal Gaussian

(a) (b)

Figure 6.7: *Example stillshot from "temporal foveation" experiment. (a) Original video frame. (b) Video frame after gaze-contingent temporal filtering; gaze position is indicated by the white marker at the left image border below the sail. The static background (low temporal frequencies) is unmodified, but moving objects are progressively filtered out with increasing distance from gaze (note that two walkers in the original frame have disappeared). The resolution falloff is described by the curve in Figure 6.6(b).*

pyramid with c lookahead items on the lowest level is

$$\begin{aligned} \text{frames}_{\text{down}} = \sum_{k=0}^{N} (\lambda_k + 1) \quad &= \sum_{k=0}^{N} (2^{N-k+1} - 1) \cdot c + 1 \\ &= c \cdot \sum_{k=1}^{N+1} \left[2^k\right] - (N+1) \cdot c + N + 1 \\ &= c \cdot (2^{N+2} - N - 3) + N + 1. \end{aligned} \tag{6.2}$$

Downsampling computational costs A downsampling operation is required for every frame $G_k(n), k > 0$; on average, this means that $\sum_{k=1}^{N} 2^{-k}$ downsampling operations are required in each time step. For large N, this approaches 1. However, the number of downsampling operations varies for each frame; because level l has to be updated in every 2^l-th time step only, every second input frame sees no downsampling performed, whereas in some frames, all N lower pyramid levels have to be updated. Nevertheless, this varying computational load (and therefore, varying latency for the finally rendered output image) can be overcome by appropriately buffering the output of the downsampling pyramid in a variable-length buffer.

Because our experimental results indicate that temporal filtering is mainly limited by memory bandwidth, we ignore the exact number of arithmetic CPU instructions required and approximate computational costs by memory accesses: for each downsampling operation, the number of images to read is $\text{reads}_{\text{down}} = 2c + 1$ and that to be written is $\text{writes}_{\text{down}} = 1$.

6.4 Experiments with peripheral motion blur

We shall now use our gaze-contingent display that can lowpass filter in the temporal domain as a function of gaze for psychophysical experiments.

Effect on eye movements

It is a well-established fact that motion and temporal transients in the periphery attract attention; the hypothesis we tested therefore was that the removal of high temporal frequencies in the periphery should reduce the number of saccades towards the periphery, i.e. saccades with a large amplitude. Ten subjects participated in the experiment and were presented with a set of seven movies out of our set of 18 high-resolution natural movies (see Chapter 3). Due to real-time constraints, the videos were reduced in resolution to 1024 by 576 pixels; on a temporal pyramid with six levels, the initial latency is 126 video frames (see above), so that video duration also was reduced from about 20 s to about 16 s. Eye movements were recorded at 250 Hz using an SR Research EyeLink II eye tracker; overall, about 2800 saccades were collected. The resolution map was as above in Figure 6.6(b), so that full temporal resolution of 29.97 frames per second was retained foveally, and dropped to 0.94 frames per second in the far periphery (the lowest resolution possible on a pyramid with six temporal levels).

Results are shown in Figure 6.8. The baseline condition is described by the dashed line and corresponds to the saccades in our data set of 54 subjects from the same seven movies (see Chapter 3). Beyond an eccentricity of about 18 deg, the temporal filtering sets in (see the resolution map in Figure 6.6(b)), and large-amplitude saccades are less frequent; the difference in saccade amplitude distributions is statistically significant ($p < 0.013$, Kolmogorov-Smirnov test).

Visibility

We will now describe an experiment where we locally suppress higher temporal frequencies. Locus and size of the suppressed region are varied in order to investigate the visibility of such changes as a function of eccentricity. Note that we do not intend to measure the threshold for the maximum temporal frequency that can be detected at a given eccentricity; it is the absence of higher temporal frequencies above a certain threshold that should go unnoticed. For typical natural scenes with their multitude of different moving objects, these two thresholds may differ considerably. Because visual attention is limited to only a small number of objects or events at any one time (O'Regan et al., 1999),

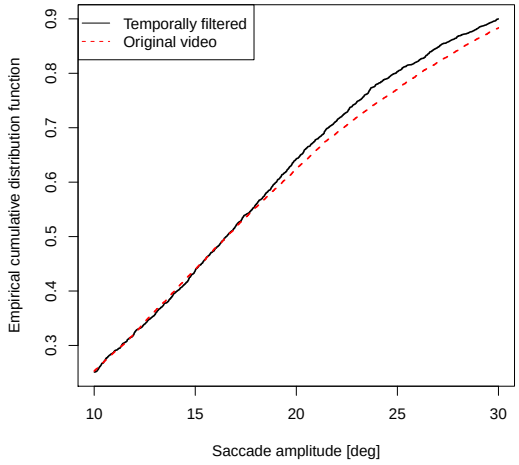


Figure 6.8: *Effect of temporal foveation on saccade amplitudes. The rate of saccades with large amplitude (≥ 18 deg) is reduced compared to the unfiltered display. Temporal filtering increases with eccentricity and is very weak close to the fovea, so small-amplitude saccades are not affected. The difference in saccade amplitude distributions is statistically significant ($p < 0.013$, Kolmogorov-Smirnov test).*

it is essential only to preserve a "natural" percept of a scene rather than its full spatio-temporal content.

We used a temporal pyramid as above with five levels, so that the lowest temporal resolution on the display was about 1.9 frames per second. The resolution map preserved full temporal resolution everywhere in the visual field except for a ring-shaped region at a given eccentricity around centre of fixation; to prevent the rise of sharp ring boundaries, the transition between filtered and non-filtered region was smoothened by a Gaussian. Formally, the resolution map α was defined as

$$\alpha(\phi) = \begin{cases} \alpha_r & \phi_i \leq \phi \leq \phi_o \\ 1 & \phi < \phi_i - 2\sigma \vee \phi > \phi_o + 2\sigma \\ \alpha_r + (1 - \alpha_r) \cdot G'_\sigma \left(\min_{k=i,o} \{|\phi - \phi_k|\} \right) & \text{otherwise,} \end{cases}$$

where ϕ_i is the eccentricity of the inner border of the ring, w is the width of the ring, $\phi_o = \phi_i + w$ is the eccentricity of the outer border of the ring, and G'_σ is an inverted Gaussian with standard deviation σ.

We measured thresholds for three different scotoma widths (1.25, 2.5, and 5 deg), a flank width of $2\sigma = 0.5$ deg, and five different eccentricities ϕ_i, namely 0 deg (a foveally presented circular disk), 10, 20, 30, and 40 deg. Thresholds

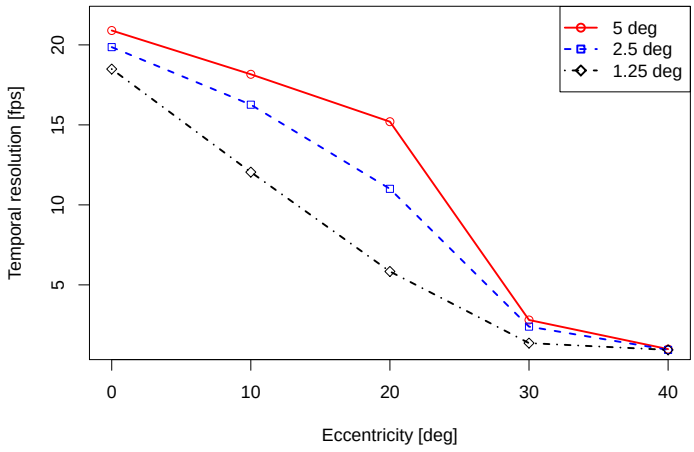


Figure 6.9: *Visibility of temporal blur on a gaze-contingent display. Temporal blur was introduced along a ring-shaped mask of varying width (1.25, 2.5, and 5 deg) and eccentricity. At far eccentricities, even large parts of the temporal frequency spectrum can be filtered out without subjects being able to notice; if these modifications take place in a larger area, they are also more visible.*

were determined in a staircase procedure where blur was increased until subjects reported noticing it; then, blur was decreased again until subjects did not report perception anymore. The average intensity of 10 reversal points was the threshold estimate. Before each trial, one parameter set was chosen randomly so that the subject could not know beforehand where in the visual field the "temporal scotoma" would occur.

Figure 6.9 shows results for three subjects. Sensitivity to high temporal frequencies drops towards the periphery, and thresholds are lower (detection frequencies are higher) when a larger area is temporally blurred, i.e. for larger scotoma widths. These results demonstrate that it is possible to introduce strong modifications as long as these preserve the "naturalness" of a scene (many scenes do not contain moving objects; thus, removing them does not change naturalness) and as long as they happen in a gaze-contingent fashion, i.e. the scene remains unmodified at centre of fixation. The results in Figure 6.9 may seem counterintuitive at first, considering that laboratory experiments with synthetic stimuli such as drifting gratings have shown much higher sensitivity to high temporal frequencies in the periphery. We here have not tested the threshold for the presence of certain frequencies, but for their absence; furthermore, we do not claim that the curve in Figure 6.9 is universally applicable.

Ultimately, however, visual performance in natural scenes with their specific spatio-temporal energy distribution is more relevant than on artificial stimuli.

6.5 Gaze visualization

As we have seen already, humans move their eyes around several times per second to successively sample visual scenes with the high-resolution centre of the retina. The direction of gaze is tightly linked to attention, and what people perceive ultimately depends on where they look (Stone et al., 2003). Naturally, the ability to record eye movement data led to the need for meaningful visualizations. One-dimensional plots of the horizontal and vertical components of eye position over time have been in use since the very first gaze recording experiments (Delabarre (1898) affixed a small cap on the cornea to transduce eye movements onto a rotating drum, using plaster of Paris as glue). Such plots are useful for detailed quantitative analyses, but not very intuitively interpreted. Other tools supporting interpretation of the data include the visualization of gaze density by means of clustered gaze samples (Heminghous and Duchowski, 2006) or the visualization of other features such as fixation duration (Ramloll et al., 2004). Better suited for visual inspection are approaches that use the stimulus and enrich it with eye movement data; in the classical paper of Yarbus (1967), gaze traces overlaid on the original images immediately show the regions that were preferentially looked at by the subjects. Because of the noisy nature of both eye movements and their measurements, there is also an indirect indication of fixation duration (traces are denser in areas of longer fixation). However, such abstract information can also be extracted from the raw data and presented in condensed form: for example, bars of different size are placed in a three-dimensional view of the original stimulus to denote fixation duration in (Lankford, 2000); in a more application-specific manner, Špakov and Räihä (2008) annotate text with abstract information on gaze behaviour for the analysis of translation processes.

Another common method is the use of so-called fixation maps (Velichkovsky et al., 1996; Wooding, 2002b). Here, a probability density map is computed by the superposition of Gaussians, each centred at a single fixation (or raw gaze sample), with a subsequent normalization step. Areas that were fixated more often are thus assigned higher probabilities; by varying the width of the underlying Gaussians, it is possible to vary the distance up to which two fixations are considered similar. Based on this probability map, the stimulus images are processed so that for example luminance is gradually reduced in areas that received little attention; so-called heat maps mark regions of interest

with transparently overlaid colours. Špakov and Miniotas (2007) add "fog" to render visible only the attended parts of the stimulus.

For dynamic stimuli, such as movies, all the above techniques can be applied as well; one straightforward extension from images to image sequences would be to apply the fixation map technique to every video frame individually. Care has to be taken, however, to appropriately filter the gaze input in order to ensure a smooth transition between video frames.

In this section, we shall present an algorithm to visualize dynamic gaze density maps by locally modifying spatio-temporal contrast on a spatio-temporal Laplacian pyramid. In regions of low interest, spectral energy is reduced, i.e. edge and motion intensity are dampened, whereas regions of high interest remain as in the original stimulus. Conceptually, this algorithm is related to gaze-contingent displays simulating visual fields as presented in the previous sections; in these approaches, however, fine spatial or temporal details are blurred selectively. Instead of blurring, the work presented here leaves details intact but reduces spectral amplitude equally across all frequency bands (note, however, that an individual weighting of separate frequency bands is a trivial extension; also see Section 6.7). Furthermore, while the algorithm presented here is based on work by Geisler and Perry (2002) and Böhme et al. (2006b), it cannot be used for gaze-contingent applications where all levels of the underlying pyramid need to be upsampled to full temporal resolution for every video frame. Its purpose is the off-line visualization of pre-recorded gaze patterns. A gaze-contingent version of a spatio-temporal Laplacian pyramid, which has much higher computational costs, shall be introduced at the end of this chapter.

Pyramid-based rendering as a function of gaze has been shown to have a guiding effect on eye movements (see previous sections). To further demonstrate the usefulness of our algorithm, we will present some results from a validation experiment in which students received instructional videos either with or without a visualization of the eye movements of an expert watching the same stimulus. Results show that the visualization technique presented here indeed facilitates perceptual learning and improves students' later visual search performance on novel stimuli.

In the previous sections, we have discussed a spatial Laplacian and a temporal Gaussian pyramid. For the temporal pyramid, we analysed in detail which video frames need to be held in memory for the temporal filtering operations, which require history and lookahead frames. We also looked at the interpolation or upsampling of lower pyramid levels to be able to use them in every time step of a gaze-contingent display. We shall now combine these techniques to develop an isotropic spatio-temporal Laplacian pyramid.

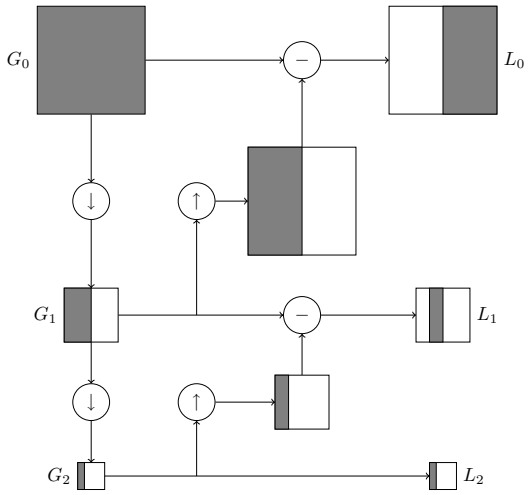


Figure 6.10: *Analysis phase of a Laplacian pyramid in space. Based on the Gaussian pyramid on the left side, which stores successively smaller image versions (with higher-frequency content successively removed), differences of Gaussian pyramid levels are formed to obtain individual frequency bands (right side). To be able to form these differences, lower levels have to be upsampled before subtraction (middle). The gray bars indicate – relative to the original spectrum – what frequency band is stored in each image. The extension into the temporal domain results in lower frame rates for the smaller video versions (not shown).*

From Chapter 2, we recapitulate that the analysis phase of a Laplacian pyramid is performed by subtracting from each other adjacent levels of a Gaussian pyramid; the resulting frequency subbands are then (possibly after a modification) added up to synthesize the original signal again. A schematic overview of pyramid analysis is depicted in Figure 6.10, and pyramid synthesis is shown in Figure 6.11.

Both for the subtraction of adjacent Gaussian pyramid levels (to create Laplacian levels) and for the reconstruction step (in which the Laplacian levels are recombined), lower levels first have to be upsampled to match the resolution of the higher level. Following these upsampling steps, the results have to be filtered to interpolate at the inserted pixels and frames; again, history and lookahead video frames are required. We shall now describe these operations in more detail and analyse the number of video frames to be buffered.

Notation

The sequence of input images is denoted by $I(t)$; input images have a size of W by H pixels and an arbitrary number of colour channels (individual channels are treated separately). A single pixel at location (x, y) and time t is referred to as $I(t)(x, y)$; in the following, operations on whole images, such as addition, are to be applied pixelwise to all pixels.

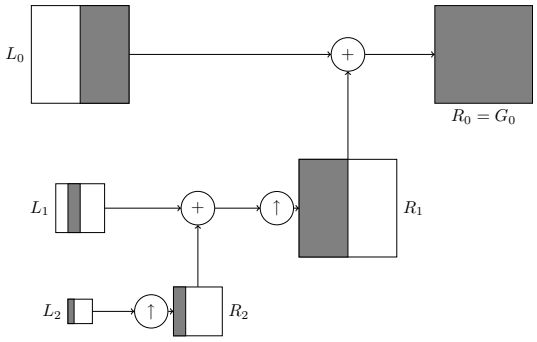


Figure 6.11: *Synthesis phase of a Laplacian pyramid in space. The Laplacian levels are iteratively upsampled and added up to obtain a series of reconstructed images $R_N, R_{N-1}, \ldots, R_0$ with increasing cutoff frequencies. If the L_n remain unchanged, R_0 is an exact reproduction of the original input image G_0.*

The individual levels of a Gaussian multiresolution pyramid with $N + 1$ levels are referred to as $G_k(t)$, $0 \leq k \leq N$. The highest level G_0 is the same as the input sequence; because of the spatio-temporal downsampling, lower levels have fewer pixels and a lower frame rate, so that $G_k(n)$ has a spatial resolution of $W/2^k$ by $H/2^k$ pixels and corresponds to the same point in time as $G_0(2^k n)$. Spatial up- and downsampling operations on an image I are denoted as $\uparrow [I]$ and $\downarrow [I]$, respectively. As was the case for the temporal Gaussian pyramid in the previous section, not all pyramid levels have a valid image $G_k(t/2^k)$ for time steps t that are not a multiple of 2^N. Therefore, C_t denotes the highest index of levels with valid images at time t, i.e. C_t is the largest integer with $C_t \leq N$ and $t \bmod 2^{C_t} = 0$. Similar to the Gaussian levels G_k, we refer to the levels of the Laplacian pyramid as $L_k(t)$, $0 \leq k \leq N$ (again, resolution is reduced by a factor of two in all dimensions with increasing k); the intermediate steps during the iterative reconstruction of the original signal are denoted as $R_k(t)$.

The temporal filtering which is required for temporal down- and upsampling introduces a latency. The number of lookahead items required on level k is denoted by λ_k for the analysis phase and by Λ_k for the synthesis phase.

Analysis phase

To compute the Laplacian levels, the Gaussian pyramid has to be created first (see Figure 6.10). The relationship of different Gaussian levels is shown in Figure 6.12; lower levels are obtained by lowpass filtering and spatially downsampling higher levels:

$$G_{k+1}(n) = \sum_{i=-c}^{c} w_i \cdot \downarrow [G_k(2n - i)] \Big/ \sum_{i=-c}^{c} w_i \; .$$

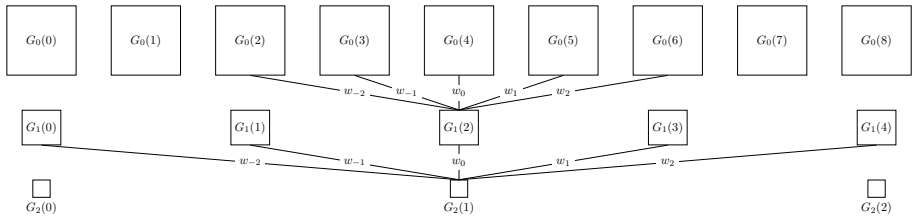


Figure 6.12: *Spatio-temporal Gaussian pyramid with three levels and $c = 2$. Lower levels have reduced resolution both in space and in time. Note that history and lookahead video frames are required because of the temporal filter with symmetric kernel, e.g. computation of $G_2(1)$ depends on $G_1(4)$, which in turn depends on $G_0(6)$.*

We here use a binomial filter kernel $(1, 4, 6, 4, 1)$ with $c = 2$.

The Laplacian levels are then computed as differences of adjacent Gaussian levels (the lowest level L_N is the same as the lowest Gaussian level G_N); before performing the subtraction, the lower level has to be brought back to a matching resolution again by inserting zeros (blank frames) to upsample and a subsequent lowpass filtering. In practice, the inserted frames can be ignored and their corresponding filter coefficients are set to zero:

$$L_k(n) = G_k(n) - \uparrow\left[\sum_{i \in P(n)} w_i \cdot G_{k+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i \in P(n)} w_i\right],$$

with $P(n) = \{j = -c, \ldots, c \mid (n - j) \bmod 2 = 0\}$ giving the set of valid images on the lower level.

Based on these equations, we can now derive the number of lookahead items required for the generation of the Laplacian. For the upsampling of lower Gaussian levels, we need a lookahead of $\beta = \lfloor \frac{c+1}{2} \rfloor$ images on each level, with $\lfloor \rfloor$ denoting floating-point truncation. Starting on the lowest level G_N, this implies that $2\beta + c$ images must be available on level G_{N-1} during the downsampling phase; we can repeatedly follow this argument and obtain $\lambda_k = 2^{N-k} \cdot (\beta + c) - c$ as the number of required lookahead images for level k.

Synthesis phase

Turning now to the synthesis phase of the Laplacian pyramid, we note from Figure 6.11 that the Laplacian levels are successively upsampled and added up to reconstruct the original image; this simply is the inverse of the "upsample–and–subtract" operation during the analysis phase. On the lowest level, $R_N(n) = L_N(n)$; for higher levels, the intermediate reconstructed images are computed

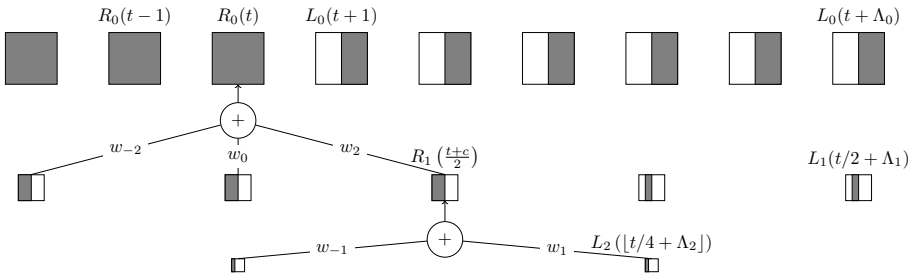


Figure 6.13: *Synthesis step of spatio-temporal Laplacian pyramid. Here shown are both the Laplacian levels L_i and the (partially) reconstructed images R_i, which are based on lower levels with indices $\geq i$; in practice, the same buffers can be used for both R and L. For example, to compute the reconstruction $R_0(t)$ of the original image, we have to add $L_0(t)$ to a spatio-temporally upsampled version of R_1. The second level L_1 is combined with the upsampling result of L_2 in $R_1(\frac{t+c}{2}) = R_1(\frac{t}{2}+\beta_1)$ (see pseudocode). In this schematic overview, new frames are added on the right side and shifted leftwards with time.*

as

$$R_k(n) = L_k(n) + \sum_{i \in P(n)} w_i \uparrow\left[R_{k+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i \in P(n)} w_i \right]. \tag{6.3}$$

Clearly, a further latency is incurred between the point in time for which band-pass information and the reconstructed or filtered image are available. Similar to the study of the analysis phase in the preceding paragraphs, we can compute the number Λ_k of required lookahead items on each level by induction. On the lowest level L_N, again $\beta = \lfloor \frac{c+1}{2} \rfloor$ images are required for the upsampling operation, which corresponds to 2β images on level $N-1$. As can be seen in Figure 6.13, the result of the upsampling operation is added to the β-th lookahead item on level $N-1$, so that $\Lambda_{N-1} = 3\beta$. Repeating this computation, we obtain $\Lambda_k = (2^{N+1-k} - 1) \cdot \beta$; for L_0, however, no further upsampling is required, so it is possible to reduce the lookahead on the highest level to $\Lambda_0 = (2^{N+1} - 2) \cdot \beta$.

In practice, we do not need to differentiate explicitly between L and R; the same buffers can be used for both L and R images. Care has to be taken then not to perform a desired modification of a given Laplacian level on a buffer that already contains information from lower levels as well (i.e. an R image).

Pseudocode and implementation

We are now ready to bring together the above observations and put them into pseudocode, see Algorithms 2 and 3. Based on $P(n)$ above, the index function that determines which images are available on lower levels in the following is $P_k(t) = \{j = -c, \ldots, c \mid \left(\frac{t}{2^k} + \beta_k - j\right) \bmod 2 = 0\}$. In the synthesis phase, the image offset β_k at which the recombination L and R takes place can be set to zero on the highest level; we therefore use $\beta_k = \beta$ for $k > 0$, $\beta_0 = 0$.

Algorithm 2 Pseudocode for one time step of the pyramid analysis phase.

Input: t Time step to update the pyramid for
$G_0, \ldots, G_N$ Levels of the Gaussian pyramid
$L_0, \ldots, L_N$ Levels of the Laplacian pyramid

$C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$

▷ Gaussian pyramid creation

$G_0(t + \lambda_0) = I(t + \lambda_0)$
for $k = 1, \ldots, C_t$ **do**

$$G_k\left(\frac{t}{2^k} + \lambda_k\right) =$$

$$\sum_{i=-c}^{c} w_i \cdot \downarrow\left[G_{k-1}\left(\frac{t}{2^{k-1}} + 2\lambda_k - i\right)\right] \Big/ \sum_{i=-c}^{c} w_i$$

end for

▷ Laplacian pyramid creation

for $k = 0, \ldots, C_t$ **do**
if $k = N$ **then**

$$L_N\left(\frac{t}{2^N} + \Lambda_N\right) = G_N\left(\frac{t}{2^N} + \Lambda_N\right)$$

else

$$L_k\left(\frac{t}{2^k} + \Lambda_k\right) = G_k\left(\frac{t}{2^k} + \Lambda_k\right) -$$

$$\uparrow\left[\sum_{i \in P_k(t)} w_i \cdot G_{k+1}\left(\frac{t}{2^{k+1}} + \frac{\Lambda_k - i}{2}\right) \Big/ \sum_{i \in P_k(t)} w_i\right]$$

end if
end for

From the pseudocode, a buffering scheme for the implementation directly follows. First, images from the Gaussian pyramid have to be stored; each level k needs at least λ_k lookahead images, one current image, and β_k history. Trading memory requirements for computational costs, it is also possible to keep all images of the Gaussian pyramid in memory twice, once in the "correct" size and once in the downsampled version; for each frame of the input video, only one downsampling operation has to be executed then. In analogy to the Gaussian levels, both the Laplacian and the (partially) reconstructed levels L and R can be held together in one buffer per level k with Λ_k lookahead, one current image, and the β_k history.

In practice, the output of the pyramid can be accessed only with a certain latency because of the symmetric temporal filters that require video frames from the future. Input images are fed into lookahead position λ_0 of buffer G_0, and images are shifted towards the "current" position by one position for every new video frame. This means that only λ_0 many time steps after video

Algorithm 3 Pseudocode for one time step of the pyramid synthesis phase.

Input: t Time step to update the pyramid for
$G_0, \ldots, G_N$ Levels of the Gaussian pyramid
$L_0, \ldots, L_N$ Levels of the Laplacian pyramid

$C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$
for $k = C_t, \ldots, 0$ **do**
  if $k = N$ **then**

$$R_N\left(\frac{t}{2^N} + \beta_N\right) = L_N\left(\frac{t}{2^N} + \beta_N\right)$$

  else

$$R_k\left(\frac{t}{2^k} + \beta_k\right) = L_k\left(\frac{t}{2^k} + \beta_k\right) +$$

$$\uparrow\left[\sum_{i \in P_k(t)} w_i \cdot R_{k+1}\left(\frac{t}{2^{k+1}} + \frac{\beta_k - i}{2}\right) \Big/ \sum_{i \in P_k(t)} w_i\right]$$

  end if
end for

frame $I(t_0)$ has been added, the Gaussian images G_0 to G_N that represent $I(t_0)$ at various spatio-temporal resolutions are available in the "current" positions of the Gaussian buffers. The resulting differences L_0 to L_N then are stored at the lookahead positions Λ_0 to Λ_N of the Laplacian buffers, respectively; here, different frequency bands can be accessed both for analysis and modification. Only Λ_0 time steps later does the input image I re-appear after pyramid synthesis; overall, this leads to a pyramid latency between input and output of $\lambda_0 + \Lambda_0$ time steps.

The necessary buffering and the handling of lookahead frames could be reduced and simplified if causal filters were used; a further possibility to efficiently filter in time without lookahead is to use temporally recursive filters. However, any non-symmetry in the filters will introduce phase shifts. Particularly in the case of space-variant filtering (see below), this would produce image artefacts (such as a pedestrian with disconnected – fast – legs and – relatively slow – upper body).

Space-variant pyramid synthesis

In the previous section, we described the analysis and synthesis phase of a spatio-temporal Laplacian pyramid. However, the result of the synthesis phase is a mere reconstruction of the original image sequence; we want to filter the image sequence based on a list of gaze positions instead.

Algorithm 4 Pseudocode for one time step of the space-variant synthesis phase.

Input: t Time step to update the pyramid for
$G_0, \ldots, G_N$ Levels of the Gaussian pyramid
$L_0, \ldots, L_N$ Levels of the Laplacian pyramid
$W_0, \ldots, W_N$ Coefficient maps

$C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$
for $k = C_t, \ldots, 0$ **do**
  if $k = N$ **then**

$$R_N\left(\frac{t}{2^N} + \beta_N\right) = L_N\left(\frac{t}{2^N} + \beta_N\right)$$

  else

$$R_k\left(x, y, \frac{t}{2^k} + \beta_k\right) = W_k\left(x, y, \frac{t}{2^k} + \beta_k\right) \cdot L_k\left(x, y, \frac{t}{2^k} + \beta_k\right) +$$

$$\uparrow\left[\sum_{i \in P_k(t)} w_i \cdot R_{k+1}\left(x, y, \frac{t}{2^{k+1}} + \frac{\beta_k - i}{2}\right) \Big/ \sum_{i \in P_k(t)} w_i\right]$$

  end if
end for

For the gaze-contingent displays discussed earlier, we had introduced the concept of a resolution map. Because we here have not only a single cutoff frequency anymore, but a set of coefficients that indicates how spectral energy should be modified in each frequency band at each pixel of the output image sequence, we shall denote the *coefficient map* for level k at time t with $W_k(t)$; the W_k have the same spatial resolution as the corresponding L_k, i.e. $W/2^k$ by $H/2^k$ pixels.

To bandpass-filter the image sequence, the Laplacian levels L_k are simply multiplied pixel-wise with the W_k prior to the recombination into R_k.

Based on the pseudocode (Algorithm 4), we can see that coefficient maps for different points in time are applied to the different levels in each synthesis step of the pyramid; this follows from the iterative recombination of L into the reconstructed levels. In practice, a more straightforward solution is to apply coefficient maps corresponding to one time t to the farthest lookahead item Λ_k of each level L_k (i.e. right after subtraction of adjacent Gaussian levels).

As noted before, in the following validation experiment we will use the same coefficient map for all levels (for computational efficiency, however, coefficient maps for lower levels can be stored with fewer pixels). In principle, this means that a similar effect could be achieved by computing the mean pixel intensity of the whole image sequence and then, depending on gaze position, smoothly blending between this mean value and each video pixel. However, for practical

reasons, the lowest level of the pyramid does not represent the "true" DC (the mean of the image sequence), but merely a very strongly lowpass-filtered video version; this means that some coarse spatio-temporal structure remains even in regions where all contrast in higher levels is removed by setting the coefficient map to zero. The temporal multiresolution character of the pyramid also adds smoothness to changes in the coefficient maps over time; because temporal levels are updated at varying rates, such changes are introduced gradually. Finally, by using different coefficient maps for each level, it is trivially possible to highlight certain frequency bands, which is impossible based on a computation of the mean alone.

6.6 Perceptual learning experiment

Pyramid-based rendering of video as a function of gaze has been shown to have a guiding effect on eye movements. For example, we have seen in the previous section that the introduction of peripheral temporal blur on a gaze-contingent display reduces the number of large-amplitude saccades, even though the visibility of such blur is low. Using a real-time gaze-contingent version of a spatial Laplacian pyramid, locally reducing (spatial) spectral energy at likely fixation points also changed eye movement characteristics.

In the following, we will therefore briefly summarize how the gaze visualization algorithm from the previous section can be applied in a learning task to guide the student's gaze. For further details of this experiment, we refer to Jarodzka et al. (2010b).

Perceptual learning

In many problem domains, experts develop efficient eye movement strategies because the underlying problem requires substantial visual search. Examples include the analysis of radiograms (Lesgold et al., 1988), driving (Underwood et al., 2003), and the classification of fish locomotion (Jarodzka et al., 2010a). In order to aid novices in acquiring the efficient eye movement strategies of an expert, it is possible to use cueing to guide their attention towards relevant stimulus locations; however, it often remains unclear where and how to cue the user. Van Gog et al. (2009) guided attention during problem-solving tasks by directly displaying the eye movements of an expert made during performing the same task on modeling examples, but found that the attentional guidance actually decreased novices' subsequent test performance instead of facilitating the learning process. One possible explanation of this effect could be that the chosen method of guidance (a red dot at the experts' gaze position that grew in

size with fixation duration) was not optimal because the gaze marker covered exactly those visual features it was supposed to highlight, and its dynamical nature might have distracted the observers. To avoid this problem, we here use the space-variant filtering algorithm presented in the previous sections to render instructional videos such that the viewer's attention is guided to those areas that were attended by the expert. However, instead of altering these attended areas, we decrease spatio-temporal contrast (i.e. edge and motion intensity) elsewhere, in order to increase the relative visual saliency of the problem-relevant areas without covering them or introducing artefacts.

Stimulus material and experimental setup

Eight videos of different fish species with a duration of 4 s each were recorded, depicting different locomotion patterns. They had a spatial resolution of 720 by 576 pixels and a frame rate of 25 frames per second. Four of these videos were shown in a continuous loop to an expert on fish locomotion (a professor of marine zoology) and his eye movements were collected using a Tobii 1750 remote eye tracker running at 50 Hz. Simultaneously, a spoken didactical explanation of the locomotion pattern (i.e. how different body parts moved) was recorded. These four videos were shown to 72 subjects (university students without prior task experience) in a training phase either as-is, with the expert's eye movements marked by a simple yellow disk at gaze position, or with attentional guidance by the pyramid-based contrast reduction. In the subsequent test or recall phase, the remaining four videos were shown to the subjects without any modification. After presentation, subjects had to apply the knowledge acquired during the training phase and had to name and describe the locomotion pattern displayed in each test video; the number of correct answers yielded a performance score.

Gaze filtering

Functionally, a sequence of eye movements consists of a series of fixations, where eye position remains constant, and saccades, during which eye position changes rapidly (smooth pursuit movements here can be understood as fixations where gaze position remains constant on a moving object). In practice, however, the eye position as measured by the eye tracker hardly ever stays constant from one sample to the next; the fixational instability of the oculomotor system, minor head movements, and noise in the camera system of the eye tracker all contribute to the effect that the measured eye position exhibits a substantial jitter. If this jitter were to be replayed to the novice, such constant erratic motion might distract the observer from the very scene that gaze guid-

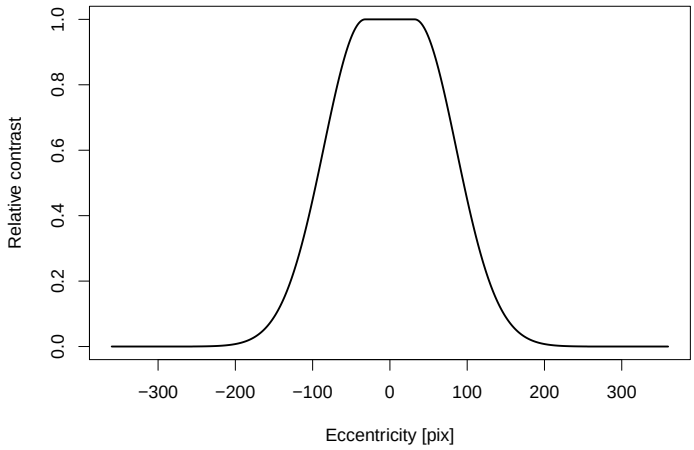


Figure 6.14: *Eccentricity-dependent coefficient map: at centre of fixation, spectral energy remains the same; energy is increasingly reduced with increasing distance from gaze.*

ance is supposed to highlight. In order to reduce the jitter, raw gaze data was filtered with a temporal Gaussian lowpass filter with a support of 200 ms and a standard deviation of 42 ms.

Space-variant filtering and colour removal

A Laplacian pyramid with five levels was used; coefficient maps were created in such a way that the original image sequence was reconstructed faithfully in the fixated area (the weight of all levels during pyramid synthesis was set to 1.0) and spatio-temporal changes were diminished (all level weights set to 0.0) in those areas that the expert had only seen peripherally. On the highest level, the first zone was defined by a radius of 32 pixels around gaze position and weights were set to 0.0 outside a radius of 256 pixels; these radii approximately corresponded to 1.15 and 9.2 degrees of visual angle, respectively. In parafoveal vision, weights were gradually decreased from 1.0 to 0.0 for a smooth transition, following a Gaussian falloff with a standard deviation of 40 pixels (see Figure 6.14). Furthermore, these maps were produced not only by placing a mask at the current gaze position in each video frame; instead, masks for all gaze positions of the preceding and following 300 ms were superimposed and the coefficient map was then normalized to a maximum of 1.0. During periods of fixation, this superposition had little or no effect; during saccades, however, this procedure elongated the radially symmetric coefficient map along the direction of the saccade. Thus, the observer was able to follow the expert's

(a) (b)

Figure 6.15: *(a) Stillshot from an instructional video on classifying fish locomotion patterns. (b) The eye movements of an expert giving voiceover explanations are visualized by space-variant filtering on a spatio-temporal Laplacian pyramid: spatio-temporal contrast and colour saturation are reduced in unattended areas. This visualization technique aids novices in acquiring the expert's perceptual skills.*

saccades and unpredictable large displacements of the unmodified area were prevented. Finally, colour saturation was also removed from non-attended areas similar to the reduction of spectral energy; here, complete removal of colour started outside a radius of 384 pixels around gaze, and the Gaussian falloff in the transition area had a standard deviation of 67 pixels. Note that these parameters were determined rather informally to find a reasonable trade-off between a focus that would be too restricted (if the focus were only a few pixels wide, discrimination of relevant features would be impossible) and a wide focus that would be without guidance effect (if the unmodified area encompassed the whole stimulus). As such, these parameters are likely to be specific to the stimulus material used here. For a thorough investigation of the visibility of peripherally removed colour saturation using a gaze-contingent display, we refer to Duchowski et al. (2009). An example frame is shown in Figure 6.15.

Results

Previous research has already shown that providing a gaze marker in the highly perceptual task of classifying fish locomotion facilitates perceptual learning: subjects look at relevant movie regions for a longer time and take less time to find relevant locations after stimulus onset, which in turn results in higher performance scores in subsequent tests on novel stimuli (Jarodzka et al., 2009). The gaze visualization technique presented here does not cover these relevant locations; subjects' visual search performance is improved even beyond that obtained with the simple gaze marker. Results are summarized in Table 6.1: time needed to find relevant locations after stimulus onset decreases by 21.26%

	Control group	Gaze marker	Gaze guidance
Time until looked at relevant areas (s)	1662.87 (697.05)	1530.36 (509.55)	1205.00 (391.75)
Time spent on relevant areas (s)	505.19 (353.68)	701.00 (332.07)	751.82 (300.57)
Multiple choice test performance	0.97 (0.32)	1.14 (0.47)	1.02 (0.44)

Table 6.1: *Results for test phase of perceptual learning experiment on fish locomotion patterns (first row shows mean performance, standard deviation in parentheses). Students had received instructional videos during the training phase either as-is (control group), with the expert's gaze position indicated by a simple yellow marker (gaze marker), or with the expert's gaze position highlighted by a contrast reduction of non-relevant locations (gaze guidance). On novel stimuli during test (without guidance), the gaze-guidance group finds task-relevant stimulus areas faster and spends more time fixating them. Recognition performance, i.e. correctly naming the depicted locomotion pattern, is also slightly improved compared to the control group.*

compared to the gaze marker condition and by 27.53% compared to the condition without any guidance. Moreover, dwell time on the relevant locations increases by 7.25% compared to the gaze marker condition and by 48.82% compared to the condition without any guidance. The recognition performance on novel test stimuli is also slightly improved in those students that received attentional guidance, but this effect is not statistically significant. For a more in-depth analysis see Jarodzka, van Gog, Dorr, Scheiter, and Gerjets (2010b).

Discussion

We have presented a novel algorithm to perform space-variant filtering of a movie based on a spatio-temporal Laplacian pyramid. One application is the visualization of eye movements on videos; spatio-temporal contrast is modified as a function of gaze density, i.e. spectral energy is reduced in regions of low interest. In a validation experiment, subjects watched instructional videos on fish locomotion either with or without visualization of the eye movements of an expert. We were able to show that on novel test stimuli, subjects who had received such information performed better than subjects who had not benefited from the expert's eye movements during training, and that the gaze visualization technique presented here facilitated learning better than a simple gaze display (yellow gaze marker). In principle, any visualization technique that reduces the relative visibility of those regions not attended by the expert might have a similar effect; our choice for this particular technique was motivated by our work on eye movement prediction, which shows that spectral energy is a good predictor for eye movements (see Chapter 4).

6.7 Real-time spatio-temporal Laplacian pyramid

In this section, we will now present an algorithm to perform efficient space-variant spatio-temporal bandpass filtering of video as a function of gaze. We have implemented this algorithm on the Graphics Processing Unit of commodity graphics hardware and achieve frame rates of more than 60 frames per second on HDTV video (1280 by 720 pixels). Using an eye tracker to perform filtering in retinal coordinates, we can thus simulate spatio-temporal visual fields in real time.

Overview

We presented an isotropic spatio-temporal Laplacian pyramid for space-variant filtering and gaze visualization in the previous section. As we already have pointed out, that algorithm is not suitable for gaze-contingent filtering because the lower levels are stored at a lower frame rate than the original video. According to the Nyquist theorem (Theorem 1), this lower frame rate suffices to eventually reconstruct the original video faithfully; however, for a gaze-contingent application we want to be able to (locally) modify the weight assigned to each band at any time in response to an eye movement. Therefore, lower temporal levels have to be upsampled and interpolated at every frame. In principle, this could be achieved by synthesizing the full pyramid in every time step; as we shall see in the following, however, this approach is prohibitively expensive so that low latencies cannot be realized.

We shall therefore present a more efficient algorithm that computes the frequency subbands in each time step, based on a Gaussian pyramid where all levels have been upsampled to full temporal resolution. Using this algorithm, even the more costly anisotropic decomposition of the spectrum into individual spatio-temporal subbands becomes feasible, so that e.g. content with high spatial and low temporal frequencies can also be individually weighted.

An iterative upsampling scheme for a temporal Gaussian pyramid was developed by Böhme, Dorr, Martinetz, and Barth (2006b) and reviewed earlier in this chapter; later in this section, we shall present a more efficient scheme that directly upsamples lower levels to full temporal resolution.

Finally, we have implemented the spatio-temporal gaze-contingent display on commodity graphics hardware using the Cg shading language (Mark et al., 2003). Today's Graphics Processing Units (GPUs) can operate on several hundred pixels simultaneously and have much higher throughput than CPUs; because the pixel position in an image is an explicit parameter in GPU programs, they are ideally suited for gaze-contingent displays (Duchowski and

Algorithm 5 Algorithm for gaze-contingent spatio-temporal filtering.

Input: n Time step

Update spatio-temporal Laplacian pyramid with new image $I(n)$
Locally weight each pyramid level $L_{s,t}(n)$ as a function of gaze $(g_x(n), g_y(n))$
Synthesize pyramid to create reconstructed image $R_{0,0}(n)$

Çöltekin, 2007; Nikolov et al., 2004). We shall therefore conclude this chapter with data from an experiment that was performed with this setup, and present benchmark results.

Upsampling all Laplacian levels to full resolution

We shall now look at the implementation of gaze-contingent filtering on an anisotropic spatio-temporal Laplacian pyramid in detail. In contrast to the isotropic pyramid described previously, such a pyramid does not require to mix spatial and temporal up- and downsampling steps. Instead, it is straightforward to first create a spatial Laplacian pyramid from each frame of the input image sequence, and then further decompose each of these spatial subbands into a temporal Laplacian pyramid. Pyramid synthesis in this case is equally simple and can be achieved by synthesis of each temporal pyramid first, followed by synthesis of a spatial pyramid where each level corresponds to a temporal synthesis result. Because computation of a spatial Laplacian has been covered before, we shall here put an emphasis on the details of the temporal Laplacian. The same notation applies as before, but to avoid confusion with the indexing of temporal levels later, we shall denote video frames with time step n instead of t, so that e.g. the input image sequence is denoted by $I(n)$.

To understand the basic steps required for gaze-contingent filtering, a very high-level overview of the gaze-contingent algorithm is listed in Algorithm 5. We can see that the current gaze position only needs to be known before the (locally weighted) pyramid synthesis; it is thus only the synthesis phase that is critical for display latency, which is a major performance goal for gaze-contingent applications.

To compute the Laplacian levels L_k from the Gaussian levels G_k, we refer back to Equation 6.4 and adapt it to the case of an anisotropic pyramid that subsamples in the temporal domain only:

$$L_k(n) = G_k(n) - \sum_{i \in P(n)} w_i \cdot G_{k+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i \in P(n)} w_i \ .$$

Computation of $L_k(n)$ on average requires $1 + \frac{2c+1}{2}$ reads and 1 write. Since L_k needs to be updated only every 2^k-th frame, the cost of computing all Laplacian levels from a Gaussian pyramid with $N+1$ levels is (note that $L_N = G_N$ and thefore no explicit additional computation is required for L_N)

$$\begin{aligned} \text{reads}_{\text{L}}(N) &= \sum_{k=0}^{N-1} 1 + \left(\frac{2c+1}{2^{k+1}}\right) = N + (2c+1)\left(1 - \frac{1}{2^N}\right) \\ \text{writes}_{\text{L}}(N) &= N. \end{aligned} \tag{6.4}$$

We now move on to temporal pyramid synthesis; a graphical illustration of the canonical algorithm is shown in Figure 6.16 (similar to synthesis of an isotropic pyramid, which was shown in Figure 6.13). We can here see that a problem arises with the gaze-contingent algorithm (Algorithm 5): $R_0(n)$ does not only depend on the $L_k(n)$, but also on $L_1(n-1)$, $L_1(n+1)$, etc. If the local weighting of the frequency bands changes due to an eye movement, all R_k have to be recomputed, which leads to an exponential increase in run time with the number of levels. For simplicity, we shall ignore the computational cost of applying the local weighting in the following and give an approximation of the number of memory accesses required. In our experience, the number of memory accesses is a reasonable indicator for computational complexity because algorithms on temporal pyramids typically are memory bandwidth-bound; operations on several high-resolution video frames at once usually exceed cache sizes. Furthermore, the implementation target are Graphics Processing Units with hundreds of shader units that can perform multiply–and–add operations in one clock cycle, so that it is possible to e.g. add a local weighting scheme at virtually no cost.

The adaptation of Equation 6.3 to the case of an anisotropic pyramid is also straightforward:

$$R_k(n) = L_k(n) + \sum_{i \in P(n)} w_i \, R_{k+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i \in P(n)} w_i \, . \tag{6.5}$$

From this, we can see that computation of $R_0(n)$ from R_1 on average requires $1 + \frac{2c+1}{2}$ reads and 1 write. The $\frac{2c+1}{2}$ reads are from R_1 (the remaining read is from $L_0(n)$), which again each require the same number of reads and writes as $R_0(n)$; this reasoning can be repeated until level R_{N-1} is reached, where for each level image $\frac{2c+1}{2}$ images from L_N are read. Putting these thoughts together, we can estimate the number of reads and writes for synthesis of a pyramid with $N+1$ levels as

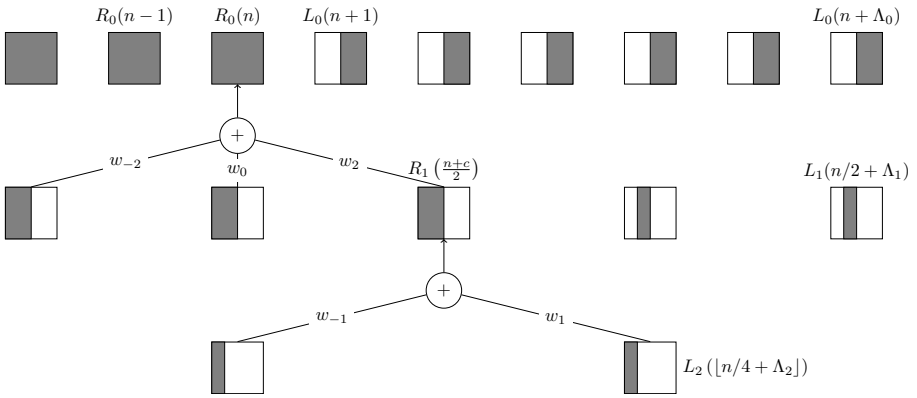


Figure 6.16: *Schematic overview of temporal pyramid synthesis. For example, $R_0(n)$ is computed by adding to $L_0(n)$, which represents the high-frequency information, a temporally upsampled and filtered version of R_1; R_1, in turn, is the result of adding the mid-frequency band L_1 to the DC component L_2. This overview shows the anisotropic case of Figure 6.13.*

$$\begin{aligned} \text{reads}_{\text{canonical}}(N) &= \sum_{k=0}^{N-1} \left(\frac{2c+1}{2}\right)^k + \sum_{k=1}^{N} \left(\frac{2c+1}{2}\right)^k \\ \text{writes}_{\text{canonical}}(N) &= \sum_{k=0}^{N-1} \left(\frac{2c+1}{2}\right)^k . \end{aligned} \tag{6.6}$$

Even for a relatively small number of temporal levels such as four ($N = 3$), this results in an average of about 44 accesses to high-resolution video frames that need to be processed before the gaze-contingent display can be updated. We have seen in Section 6.2 that an image processing latency even as low as 10 ms can lead to noticeable artefacts on gaze-contingent displays; we therefore need a more efficient algorithm for locally weighted pyramid synthesis.

Such a more efficient algorithm can be found if we do not compute the Laplacian levels L_k as differences of Gaussian levels G_k, G_{k+1} at reduced frame rate $1/2^k$ anymore, but at full temporal resolution. We remember from Section 6.3 that we already have an algorithm that provides levels U_k^0 with the same spectral content as G_k, but at full frame rate. Ignoring the cost of computing the U_k^k for the moment, we note that we can obtain bandpass levels $L_k^u(n)$ with full temporal resolution as

$$L_k^u(n) = U_k^0(n) - U_{k+1}^0(n),$$

so that the set of L_k^u depends only on $N+1$ frames with time stamp n, and notably does not depend on gaze position. The number of necessary memory accesses for this step is (assuming an implementation that can read all U_k^0

simultaneously)

$$\begin{aligned} \text{reads}_{\text{new}} &= N+1 \\ \text{writes}_{\text{new}} &= N. \end{aligned} \tag{6.7}$$

A weighted, gaze-contingent synthesis can then be obtained simply as

$$R(n) = \sum_{k=0}^{N} \alpha_k(n, g_x(n), g_y(n)) \cdot L_k^u(n), \tag{6.8}$$

with $\alpha_k(x, y)$ a coefficient map that directly specifies the desired weighting coefficient for frequency band k at pixel (x, y). It directly follows that for the latency-critical part of the algorithm, namely the operations required in response to an eye movement and thus a change in the local weighting, computational cost is greatly reduced:

$$\begin{aligned} \text{reads}_{\text{synth}} &= N+1 \\ \text{writes}_{\text{synth}} &= 1. \end{aligned} \tag{6.9}$$

Of course, this reduction in latency-critical operations comes at the expense that we now have to compute the U_k^0 and the L_k^u for every time step. For three reasons, however, this trade-off is justified. First, this information can be computed in the background and does not affect the primary acceptance criterion of gaze-contingent displays, namely latency. Second, the U_k^0 and L_k^u have to be computed with the frame rate of the input video only; if higher display update rates are desired, additional computation is limited to Equation 6.8. Finally, as we shall see in the following, upsampling all levels of a Gaussian pyramid (Equation 6.1) to full temporal resolution can be achieved with significantly fewer operations than in Equation 6.6; in particular, we shall present an improved upsampling algorithm that saves up to a further 20% of operations compared to Equation 6.1.

Upsampling of a temporal Gaussian pyramid

We shall first analyse the iterative upsampling scheme from Equation 6.1. To recapitulate, U_k^0 is obtained from G_k by performing k iterative upsampling steps. The intermediate results of these operations are denoted by U_k^k to U_k^0, where $U_k^k = G_k$, and U_k^l is the result of upsampling U_k^{l+1}.

Iterative upsampling memory requirements As we have seen in Section 6.3, c lookahead, one current, and zero history items have to be buffered on each level U_k^i. Since the images U_k^k do not need to be explicitly represented, but can be taken from G_k, overall we have to buffer $M_{\text{iter}} = (c+1) \cdot \frac{N(N+1)}{2}$ video frames for the iterative upsampling scheme.

Iterative upsampling computational costs The length of the kernel w used for interpolation during the upsampling step is the same as for downsampling, $K = 2 \cdot c + 1$. However, on average, half of the coefficients would apply only to empty frames on the lower level, so do not need to be taken into account. On average, one upsampling operation therefore requires $K/2$ reads and one write. The number of reads required to compute U_k^l from U_k^{l+1} is thus $K/2 \cdot \frac{1}{2^l}$, and to compute U_k^0 from U_k^k,

$$\text{reads}_\text{k} = K/2 \cdot \sum_{i=0}^{k-1} \frac{1}{2^i} \, .$$

To upsample all levels to full temporal resolution, we then need

$$\begin{aligned} \text{reads}_\text{iter} &= K/2 \cdot \sum_{i=0}^{N-1} \frac{N-i}{2^i} \\ \text{writes}_\text{iter} &= 2 \cdot \text{reads}_\text{iter}/K = \sum_{i=0}^{N-1} \frac{N-i}{2^i} \, . \end{aligned}$$

Direct upsampling We now improve upon the temporal upsampling algorithm by "direct upsampling" as opposed to the previously implemented "iterative upsampling". The underlying idea is that for each pyramid level k, we precompute the filter kernel $w^k = w^k_{-c_k}, \ldots, w^k_0, \ldots, w^k_{c_k}$ that effectively assigns the same weight to each frame on level k as would do the iterative convolution with the standard kernel w. Intuitively, one might assume that this is inefficient because the length of such direct upsampling kernels grows exponentially in the number of levels; however, at the same time, the frame rate and thus the rate of kernel coefficients that are used at all shrinks exponentially on the lower levels as well. As we will see later, the number of memory accesses is indeed reduced. The main benefit of this scheme is that no intermediate results need to be stored because the upsampling from G_k to $U_k := U_k^0$ can be performed in one step,

$$U_k(n) = \sum_{i \in P_k(n)} w_i^k \cdot G_k\left(\frac{n-i}{2^k}\right) \Bigg/ \sum_{i \in P_k(n)} w_i^k$$

with $P_k(n) = \{j = -\frac{|w^k|-1}{2}, \ldots, \frac{|w^k|-1}{2} \mid (n-j) \bmod 2^k = 0\}$. We can now iteratively derive the upsampling kernel w^k that upsamples G_k to U_k. Since level G_0 is the same as the input sequence, no upsampling is required, and $w^0 = (1)$ therefore is the identity. To obtain w^{k+1}, we upsample w^k by filling in zeros and then convolve the result w^k_up with $w^1 = w$; this is the same operation as the one performed in each iteration of the iterative upsampling algorithm. To es-

tablish a straightforward index scheme, we assign indices $i = -\frac{|w^k|-1}{2}, \ldots, \frac{|w^k|-1}{2}$ to the kernel coefficients w_i^k; because the original kernel w should be chosen symmetric, $w_i^k = w_{-i}^k$. Formally,

$$w^{k+1} = w_{\text{up}}^k * w,$$

and we can prove that

$$|w^k| = (2^k - 1) \cdot |w| - 2^k + 2.$$

We begin by noting that $|w_{\text{up}}^k| = 2 \cdot |w^k| - 1$ because of the insertion of zeros, and $|w^{k+1}| = |w_{\text{up}}^k| + |w| - 1$ due to the convolution. $|w^k| = (2^k - 1) \cdot |w| - 2^k + 2$ holds for $|w^0| = 1$, and

$$\begin{aligned} |w^{k+1}| &= 2 \cdot |w^k| - 1 + |w| - 1 \\ &= 2[(2^k - 1) \cdot |w| - 2^k + 2] - 1 + |w| - 1 \\ &= (2^{k+1} - 1) \cdot |w| - 2^{k+1} + 4 - 2. \end{aligned}$$

Direct upsampling memory requirements Since there are no intermediate results that have to be stored in memory and all operands can be taken from the downsampling pyramid, direct upsampling only uses $M_{\text{direct}} = N$ frames extra memory (for $U_0 = G_0$, the original image can be used).

Direct upsampling computational costs As noted above, the size of the upsampling kernel on level k is $|w^k| = (2^k - 1) \cdot |w| - 2^k + 2$; however, since this size refers to the frame rate on the highest level, only a fraction of kernel coefficients need to be used – the frame rate on level k is $1/2^k$. On average, the number of reads per frame for upsampling one level k to full resolution thus is

$$\text{reads}_{\text{direct}}^k = \frac{|w^k|}{2^k} = (1 - 2^{-k}) \cdot |w| - 1 + \frac{2}{2^k},$$

and the maximum number of reads for upsampling one level is

$$\lim_{k \to \infty} \text{reads}_{\text{direct}}^k = |w| - 1. \tag{6.10}$$

The number of reads for all levels combined is

$$\begin{aligned} \text{reads}_{\text{direct}} &= \sum_{k=1}^{N} (1 - 2^{-k})|w| - 1 + \frac{2}{2^k} \\ &= \left(N - \sum_{k=1}^{N} 2^{-k} \right) (|w| - 1) + \sum_{k=1}^{N} 2^{-k} \\ &= \mathcal{O}\Big((N-1)\,(|w| - 1) \Big). \end{aligned} \tag{6.11}$$

Since no intermediate frames are generated and only the output frame has to be written for every upsampled level, the number of writes is

$$\text{writes}_{\text{direct}} = N. \tag{6.12}$$

Comparison canonical and improved algorithm From Figure 6.17, we can see that the number of both reads and writes is reduced by direct upsampling compared to iterative upsampling. For eight temporal levels, about 21% of memory accesses can be saved. We can now also tally the number of memory accesses required to compute the Laplacian levels on a Gaussian that is fully upsampled in each time step. We have to add the cost of upsampling the Gaussian (Equations 6.11,6.12) and that of computing the Laplacian levels (Equation 6.7) to the cost of pyramid synthesis (Equation 6.9) and can estimate the number of memory accesses in each time step for a pyramid with five temporal levels and a five-tap filter kernel ($c = 2$) as 32.19. Compared to the canonical generation and synthesis of the Laplacian pyramid (Equations 6.4,6.6) with 126.88 memory accesses, this is a reduction by almost 75%; taking into account the cost of downsampling the underlying Gaussian pyramid first with 5.63 memory accesses, which is the same for both algorithms, the reduction is still more than 71%. As a further benefit, the levels of a Gaussian pyramid can be stored with a shorter data type than the Laplacian levels, which require a sign bit. To avoid quantization artefacts, the canonical algorithm needs 16 bits per pixel instead of eight for the improved algorithm, so that the effective speedup of the new algorithm (required memory bandwidth) is about eight-fold. Additionally, the number of frames that need to be stored is also reduced. From Section 6.5, we know that two sets of buffers need to be retained for the canonical algorithm, and we have computed the necessary number of lookahead items on each level λ_k (for the buffer that stores the Gaussian pyramid) and Λ_k (for the bandpass information); furthermore, each buffer level requires one current item and β_k history items. The computation here is simplified because for one spatial level

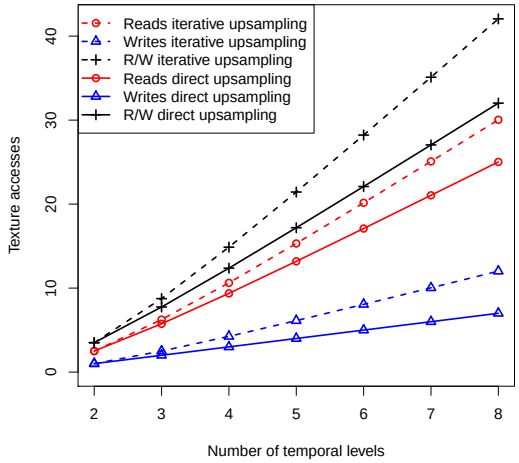


Figure 6.17: *Comparison of read and write memory accesses for direct and iterative upsampling schemes. Direct upsampling saves about 21% of accesses for eight temporal levels.*

of an anisotropic pyramid, video frames have the same size on all temporal levels; memory consumption of the canonical algorithm therefore is

$$\begin{aligned}
\text{frames}_{\text{gauss}} &= \left\lceil \sum_{k=0}^{N} \lambda_k \right\rceil + N(\beta+1) + 1 \\
&= \left\lceil \sum_{k=0}^{N} 2^{N-k} \cdot (\beta + c) - c \right\rceil + N(\beta+1) + 1 \\
&= (2^{N+1} - 1) \cdot (\beta + c) - (N+1) \cdot (c-1) + N\beta
\end{aligned}$$

$$\begin{aligned}
\text{frames}_{\text{laplace}} &= \left\lceil \sum_{k=0}^{N} \Lambda_k \right\rceil + N(\beta+1) + 1 \\
&= \beta \cdot \left\lceil \sum_{k=0}^{N} 2^{N+1-k} - 1 \right\rceil - \beta + N(\beta+1) + 1 \\
&= (2^{N+2} - 4) \cdot \beta + N + 1,
\end{aligned}$$

and overall $\text{frames}_{\text{canonical}} = \text{frames}_{\text{gauss}} + \text{frames}_{\text{laplace}}$. The number of frames required by the improved algorithm can be estimated as follows. The maximum number of frames needed for one direct upsampling step is $|w| - 1$ (Equation 6.10), and we thus need at least c history and c lookahead items on

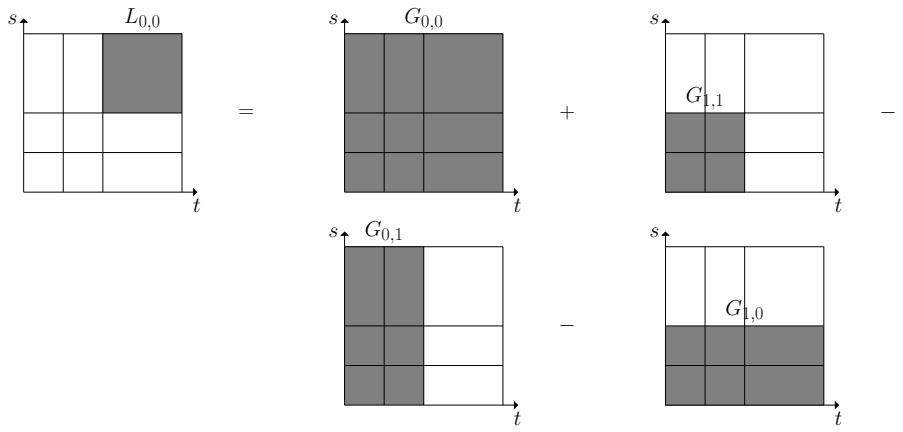


Figure 6.18: *Schematic overview of generation of anisotropic Laplacian levels based on a Gaussian pyramid where all levels are upsampled to full temporal resolution,* $L_{0,0} = G_{0,0} - G_{0,1} - G1,0 + G_{1,1}$. *The two levels* $G_{1,0}$ *and* $G_{1,1}$ *need to be spatially upsampled; for efficiency, their difference should be computed and then upsampled.*

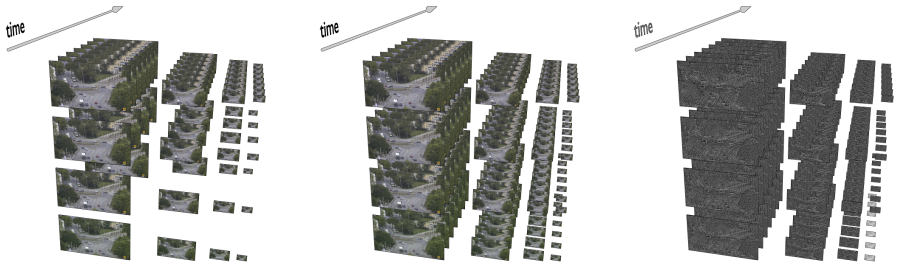
each level of the Gaussian pyramid (except for the highest level). The memory requirements for a Gaussian with zero history and c lookahead were computed in Equation 6.2; adding the $N \cdot c$ history items and the N images to store the differences of adjacent Gaussian levels, we obtain

$$\text{frames}_{\text{new}} = c \cdot \left(2^{N+2} - 3\right) + 2N + 1.$$

It follows that the new algorithm is not only more efficient computationally, but also has a lower memory consumption. For a pyramid with five temporal levels and $c = 2$, the number of frames to buffer is reduced from 157 to 131 (a reduction of 16%); in the limit, memory requirements can be reduced by up to 20%. Again, the use of narrower data types can also yield a further 50% reduction in memory footprint, which is of particular importance for an implementation on the GPU because the amount of memory available on graphics cards is currently still much smaller than that of main memory.

Gaze-contingent spatio-temporal Laplacian

So far, we have only addressed a temporal Laplacian pyramid. Using the observation that it is more efficient to first upsample all levels of the underlying Gaussian pyramid to full temporal resolution and only then compute the differences of adjacent levels in each time step, we can now extend our algorithm to the spatio-temporal domain. We begin with notation and remark that because we are here dealing with anisotropic pyramids, we need two-dimensional indices. The individual levels of an anisotropic spatio-temporal Gaussian pyra-



(a) (b) (c)

Figure 6.19: *Schematic overview of the spatio-temporal Laplacian pyramid underlying the gaze-contingent display. First, an anisotropic spatio-temporal Gaussian pyramid is created where lower levels are stored at lower resolution (a). Then, the lower temporal levels are upsampled to full temporal resolution again (b). Finally, adjacent pyramid levels are subtracted from each other to obtain individual frequency bands (c). These frequency bands can easily be modified locally; their sum yields the (modified) input image.*

mid with $S+1$ spatial and $T+1$ temporal levels therefore are referred to as $G_{s,t}(n)$, $0 \leq s \leq S$ and $0 \leq t \leq T$; images on level (s,t) have a spatial resolution of $W/2^s$ by $H/2^s$ pixels and are updated with a frame rate of $1/2^t$. With $U_{s,t}(n)$, we denote the temporally upsampled $G_{s,t}(n)$ that has full temporal resolution (but still reduced spatial resolution). With $\uparrow [I]$ a spatial upsampling operation, we can obtain a bandpass representation of the image sequence in each time step as

$$L^u_{s,t} = U_{s,t} - U_{s,t+1} - \uparrow [U_{s+1,t} - U_{s+1,t+1}] \, .$$

This procedure is shown schematically in Figures 6.18 and 6.19. The pseudocode that now describes the direct upsampling to full temporal resolution of a spatio-temporal Gaussian pyramid and the generation of anisotropic Laplacian levels based on this pyramid is listed in Algorithm 6.

Following generation of the Laplacian levels, we want to synthesize the pyramid again with a gaze-contingent, space-variant filtering mask. In order to specify a pixel-wise weighting coefficient for each spatio-temporal frequency band, we can simply extend Equation 6.8 to

$$R(n) = \sum_{s=0}^{S} \uparrow^s \left[\sum_{t=0}^{T} \alpha_{s,t}\Big(n, g_x(n), g_y(n)\Big) L^u_{s,t}(n) \right] ,$$

where $\uparrow^k [I]$ denotes k iterative spatial upsampling operations and $\alpha_{s,t}$ denotes the coefficient map for spatial level s and temporal level t. In practice, the lower-resolution levels are explicitly upsampled in space only once and then added to the next level, so that multiple upsampling steps are not necessary. This procedure is also described as pseudocode in Algorithm 7.

Algorithm 6 Pseudocode for direct upsampling of spatio-temporal Gaussian pyramid with efficient computation of anisotropic Laplacian levels $L^u_{s,t}$.

Input:	n	Time step to update the pyramid for
	$G_{s,t}(n)$	Gaussian levels, $0 \le s \le S$, $0 \le t \le T$
	w^t	Temporal upsampling kernel for level t
Output:	$L^u_{s,t}(n)$	Laplacian levels

for $s = S, \ldots, 0$ **do**
  for $t = T, \ldots, 0$ **do**
    $U_{s,t} = \lambda$ ▷ Set $U_{s,t}$ to empty image
    $i = \lfloor \frac{n}{2^t} \rfloor$ ▷ Index of current image on Gaussian level t
    $p = n - 2^t \cdot i$ ▷ Sampling position in upsampling filter kernel
    $P_t = \{j = -c, \ldots, c \mid 2^t \cdot j - p \in [-c_t, c_t]\}$
    $U_{s,t}(n) = \sum_{k \in P_t} w^t_k \cdot G_{s,t}(i+k) \Big/ \sum_{k \in P_t} w^t_k$
    if $s = S$ AND $t = T$ **then**
      $L^u_{s,t}(n) = U_{s,t}(n)$ ▷ Spatio-temporal DC
    else if $s = S$ **then**
      $L^u_{s,t}(n) = U_{s,t}(n) - U_{s,t+1}(n)$ ▷ Spatial DC
    else if $t = T$ **then**
      $L^u_{s,t}(n) = U_{s,t}(n) - \uparrow[U_{s+1,t}(n)]$ ▷ Temporal DC
    else
      $L^u_{s,t}(n) = U_{s,t}(n) - U_{s,t+1}(n) - \uparrow[U_{s+1,t}(n) - U_{s+1,t+1}(n)]$
    end if
  end for
end for

Performance

Now that we have theoretically analysed our gaze-contingent display and developed a pseudocode description, we can turn to an implementation and its performance. We shall first give some examples of the effects we can achieve with our space-variant filtering algorithm, and then provide some benchmark numbers for throughput and latency.

Two example stillshots are shown in Figures 6.20 and 6.21. In Figure 6.20, the coefficient maps for the mid-spatial frequency bands gradually change from 1.0 at the top of the image to 0.0 at the bottom; in other words, only very low (<0.8 cycles/deg) and very high (>6.7 cycles/deg) spatial frequencies constitute the bottom part of the image. A similar gradient was introduced for the mid-range of temporal frequencies, but here filter strength increases from right to left. The contours of the two walking men coming into the image from the left, for example, are still clearly visible, but overall contrast of the walkers is reduced; compare that with the effect of temporal blur shown in Figure 6.7.

In Figure 6.21, it is shown that the range of subband coefficients is not limited to $[0, 1]$. High spatial and high temporal frequencies are amplitude-enhanced

Algorithm 7 Pseudocode for gaze-contingent synthesis step.

Input:	n	Time step
	$L^u_{s,t}(n)$	Laplacian levels
	$\alpha_{s,t}$	Coefficient map for each level
	$g_x(n), g_y(n)$	Gaze position
Output:	R	Spatio-temporally modified pyramid reconstruction

$R(n) = L^u_{S,T}(n)$ ▷ Initialize $R(n)$ with DC component
for $t = T-1, \ldots, 0$ **do**
  $R(n,x,y) = R(n,x,y) + \alpha_{S,t}(x - g_x(n), y - g_y(n)) \cdot L^u_{S,t}(n)$
end for
for $s = S-1, \ldots, 0$ **do**
  $R(n) = \uparrow [R(n)]$
  for $t = T, \ldots, 0$ **do**
    $R(n,x,y) = R(n,x,y) + \alpha_{s,t}(x - g_x(n), y - g_y(n)) \cdot L^u_{s,t}(n)$
  end for
end for

by a factor of three, whereas low- and mid-frequencies are removed. Only the "DC" component remained constant, but on a pyramid with five spatial and five temporal levels, the DC component still contains a considerable range of frequencies (up to about one cycle per degree or per second). Whereas the overall scene appears slightly blurred, fine details especially of moving objects have strongly increased contrast, such as the cars. Obviously, it is a trivial step to also change the coefficients for the DC component. This, however, requires different normalization schemes for display purposes, because output images have to be mapped to pixel intensity values in $[0, 255]$, but about 50% of pixels would have negative values without the DC component. For extreme enhancements of non-DC levels, pixel saturation can also occur; note the black and white areas around the cars in Figure 6.21. We shall discuss a solution to this problem below.

Figures 6.22 and 6.23 show some benchmark results for the gaze-contingent display as implemented on the GPU. Measurements were obtained on a system with an NVIDIA GeForce GTX280 GPU with 1 GB of RAM and an Intel Core 2 Duo CPU running at 3 GHz. In Figure 6.22, we plot the median of image processing time on high-resolution movies (1280 by 720 pixels) for the downsampling phase, upsampling the pyramid to full temporal resolution on all levels, and space-variant synthesis for a pyramid with two to six temporal (on the y-axis) and two to six spatial levels (quasi-parallel lines). Even for a pyramid with six spatial and four temporal levels, all image processing combined takes less than 15 ms, which means that frame rates of more than 60 frames per second are possible. Even more importantly, the latency-critical synthesis step

(a) (b)

Figure 6.20: *Example of space-variant bandpass filtering. (a) Original image. (b) Mid-spatial frequencies (bands 1–3, corresponding to 0.8–6.7 cycles/deg) are filtered out progressively from top to bottom, and mid-temporal frequencies (1.9–7.5 cycles/s) are filtered out progressively from right to left. Note, for example, the temporal ringing effect around the pedestrian in red to the left: subtle traces of the pedestrian can also be found ahead and behind of him.*

takes only 2 ms, and we can estimate overall system latency thus to 20–25 ms (see Equation 5.1). For five and six temporal levels, computation time increases superlinearly, which is due to the fact that current GPUs have limited amounts of memory. For five and more temporal levels, textures have to be stored in the computer's main memory and the transfer via the system bus incurs a performance penalty. This, however, is not a fundamental problem because graphics boards with larger memory sizes have recently become available. Also notable is the increase in computation time with the number of spatial levels. From a theoretical standpoint, going from two to six spatial levels increases the number of pixels by less than seven per cent, yet processing time more than doubles for a pyramid with two temporal levels. This can be explained by two factors. First, the communication costs between CPU and GPU per shader run are constant, and the number of shader runs increases linearly with the number of spatial levels. Second, the number of pixels on lower spatial levels is so small that the cost of actual image processing on these levels vanishes against setting up shader units on the GPU.

For comparison purposes, similar measurements are plotted for smaller videos (640 by 360 pixels) in Figure 6.23 (only two to four spatial levels are plotted). Because frame size is reduced by a factor of four, even a pyramid with six temporal levels fits into the GPU's memory completely, so there is no sharp performance drop as on high-resolution video. Nevertheless, image processing time is only slightly lower than for high-resolution videos, and it follows that the GPU is fully utilized only with high-resolution material.

(a) (b)

Figure 6.21: *Example of contrast enhancement in selected subbands. (a) Original image. (b) High spatio-temporal frequencies ($f_s > 6.7$ cycles/deg, $f_t > 7.5$ cycles/s) are enhanced three-fold; all other subbands except for the lowest-frequency component are set to zero (computed on a pyramid with five temporal and five spatial levels, even the "DC" component still contains a considerable range of frequencies, so that the overall impression of the scene does not change dramatically).*

6.8 Applications

In order to finally put our gaze-contingent display to experimental use, we shall now present an experiment where the coefficient map is derived from information on salient and non-salient image structures that was obtained using the machine learning techniques that were discussed in Chapter 4. First, a kernel support vector machine was trained with the spectral energy of fixated and non-fixated patches. This feature was computed as the mean energy of pixel intensity in a neighbourhood around fixation on each level of a spatio-temporal Laplacian. Under the assumption that the correctly classified patches approximate the manifolds of their respective classes, a second, linear support vector machine was trained with only these patches. Using spectral energy and a linear SVM has the advantage that the feature space is now approximately invertible, i.e. any feature vector can be mapped back to an image patch. This mapping is only approximate because there are various possibilities to increase or decrease spectral energy of an image patch; we here chose the straightforward approach of multiplying every pixel in the patch with the ratio of desired and actual energy. For the geometrical invariants on the structure tensor, which showed slightly better prediction performance than spectral energy and therefore would be a natural choice of image features, such "inversion" strategy does not exist. We used this property of spectral energy to modify candidate points (see Section 6.2) and move their feature space representation perpendicular to the separating hyperplane, either to make a patch less salient (towards the non-attended class, which for salient points means towards the hyperplane) or to increase its saliency (away from the hyperplane).

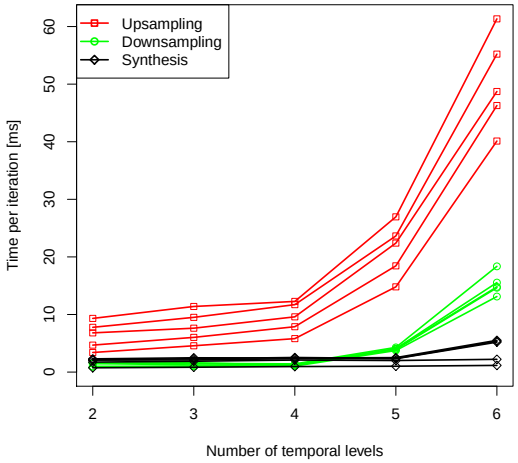


Figure 6.22: *Performance of space-variant spatio-temporal filtering algorithm implemented on the GPU and on high-resolution video. Quasi-parallel lines show image processing times for different numbers of spatial levels (two to six). A pyramid with four temporal and six spatial levels can be downsampled and resynthesized as a function of gaze with more than 60 frames per second. For a higher number of temporal levels, memory on the graphics board becomes a bottleneck and textures have to be transferred back and forth over the system bus, which reduces performance.*

This procedure is a more elaborate version of the heuristic used in Section 6.2 that simply set spectral energy to a fixed constant relative to the mean spectral energy of the non-attended patches. Now, local structure of the manifold of natural movie patches is also taken into account and relative weightings of the individual frequency bands become possible. As we have mentioned above, contrast enhancements can lead to artefacts because of the limited dynamic range of the display; an additional normalization step is too costly to perform in real time. We therefore reduced all movies to 95% overall contrast and adjusted the learned weights such that no overflows would occur. A reduction of saliency (moving a point towards the separating hyperplane) usually led to a decrease in coefficients for most frequency bands; in principle, even negative weightings are possible. To avoid this situation, the strength of the saliency-reducing transformation was chosen such that for each modified patch, only three individual frequency band coefficients would be set to zero, and negative coefficients were clipped.

Six subjects participated in the experiment and watched our 18 natural movies on the gaze-contingent display (see Chapter 3 for the physical setup). Their task was to press the space bar whenever they detected contrast modifications similar to a set of modifications that was shown prior to the experiment,

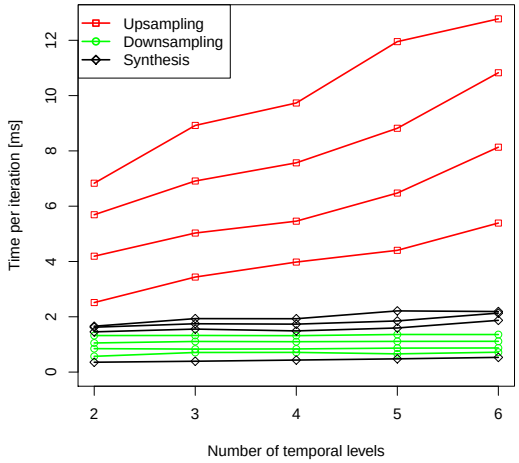


Figure 6.23: *Performance of space-variant spatio-temporal filtering algorithm implemented on the GPU as in Figure 6.22, but on smaller videos. Because of the reduced memory requirements, performance does not drop as sharply for a large number of temporal levels as in the case of high-resolution video. However, the latency-critical synthesis step is only marginally faster because of the fixed cost of communication between CPU and GPU.*

where modifications were not contingent upon gaze and therefore easily visible. Six different parameter strengths were tested so that each subject watched a different set of three movies with a particular set of parameters; overall, we thus obtained data for each combination of movie and parameter strength. For simplicity, we will here report data only from one condition; by pooling together data, stronger statistical significance could easily be achieved. During the experiment, up to 20 candidate points were determined after each saccade. The saliency of the most likely candidate point was increased further, and all other candidate points were decreased in their saliency. To reliably associate responses with modifications, the rate of modifications was limited to one every three seconds.

The effect of the gaze-contingent contrast modifications that were derived by machine learning algorithms on eye movements is shown in Figure 6.24. Here, distance is measured between saccade landing points and the nearest modified candidate location. The distribution is shifted to the left (towards smaller distances) compared to the control condition (where modifications were not shown, so could possibly have no effect). This means that gaze was drawn towards the modified regions significantly more often ($p < 0.05$, even on the relatively limited sample size). Notable is the fact that subjects reported modification detection in only 4.5% of trials; this means that the oculomotor system

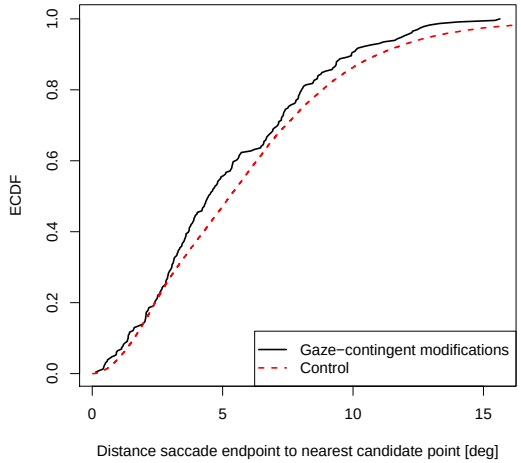


Figure 6.24: *Effect of gaze-contingent spatio-temporal contrast modifications on eye movements. Gaze is drawn towards modified locations (the distribution of distances to the nearest modified location is shifted to the left).*

could detect contrast modifications and steer gaze towards them without subjects becoming aware of this. The existence of such a gap between unconscious peripheral processing and consciousness is a good indicator that gaze guidance is possible and useful.

6.9 Chapter conclusion

In this chapter, we have presented a set of increasingly more complex and powerful space-variant filtering algorithms, and performed first gaze-guidance experiments. We started out with a gaze-contingent display based on a spatial Laplacian pyramid. For the first time, this display allowed to specify a full set of weights for each frequency band in retinal coordinates, whereas previous gaze-contingent displays only allowed for lowpass filtering with a specified cutoff frequency. Implemented on the CPU, however, this spatial bandpass filtering still suffered from latency problems due to its increased computational complexity compared to gaze-contingent displays based on a Gaussian pyramid.

We then analysed in detail an algorithm to upsample all levels of a temporal Gaussian pyramid to full temporal resolution. For gaze-contingent applications, this is a necessary step because the contribution of each level to a specific pixel of the output image can change at any point in time due to eye movements. Again, this algorithm can only be used for a lowpass-filtering, but

operates on the computationally much more costly temporal domain. The temporal domain offers two advantages over the modification of spatial content, as we have shown in two validation experiments. First, the natural appearance of a visual scene is less impaired by the introduction of temporal blur in the periphery than by spatial blur. A scene may contain no moving objects at all and therefore little high temporal frequencies, but a scene with no spatial details seems unnatural even when viewed only peripherally. Second, it is a well-known fact that high temporal frequencies in the periphery attract attention; conversely, the removal of high frequencies in the periphery reduces the number of saccades with large amplitude, that is saccades towards the periphery. One conclusion from these experiments is that it is possible to alter eye movement characteristics even with comparatively simple gaze-contingent modifications such as peripheral blur.

The logical next step after a spatial Laplacian and a temporal Gaussian is the extension to a spatio-temporal Laplacian pyramid. Due to the immensely increased computational complexity, however, we first implemented an algorithm that was not suitable for real-time gaze-contingent applications, but for offline rendering of video only. Nevertheless, we could show that the visualization of an expert's eye movements on instructional videos on fish locomotion patterns had a positive effect on students' performance. Their gaze was guided towards the relevant stimulus locations, which were difficult to determine for novices in a visually very rich scenery. Interestingly, the removal of peripheral distractors did not help students to acquire conceptual skills (after training, they could not name different locomotion patterns better than controls), but they had acquired significantly better perceptual skills (they could find task-relevant locations on novel stimuli faster than controls), which obviously is a necessary, but not sufficient condition for subsequent conceptual knowledge.

Then, we proceeded with a thorough analysis of a novel algorithm to compute a spatio-temporal Laplacian pyramid that is suitable for gaze-contingent applications. One critical element of this algorithm is that the image processing latency of the system depends only on the space-variant addition of a few video frames, whereas the canonical algorithm to compute the Laplacian would require processing of dozens of images. We also developed a more efficient algorithm to upsample a temporal Gaussian pyramid. In combination with an implementation on dedicated graphics hardware, these algorithmic improvements led to a system capable of space-variant spatio-temporal filtering of high-resolution video with 60 frames per second. Pyramid synthesis latency, which is critical to react to eye movements, is as low as 2 ms.

Finally, we ended this chapter with a validation experiment of our novel gaze-contingent system. This experiment showed that movie transformations

that were derived by machine learning techniques to make movie patches more or less salient could attract attention without becoming visible to the observer.

"Whatever good things we build end up building us."

Jim Rohn

7 Conclusion

The development of gaze-guidance systems is an interdisciplinary challenge. On the technical side, gaze-contingent displays are needed that can react to eye movements and modify videos accordingly with very low latencies. Eye trackers with high temporal resolution are already commercially available, and thus the focus must be put on fast video transformations. However, sophisticated real-time image processing routines alone are not sufficient, because a deep understanding of the human visual system is also necessary to know where and how to apply video transformations in an optimal fashion. Once these requirements are fully met, gaze guidance promises considerable benefits in many areas of human-human and human-machine communication. In safety-critical applications, such as driving, users might be alerted of potential hazards unobtrusively, which is a major acceptance criterion for driver assistance systems. In training scenarios, the demonstration of an expert's more efficient viewing strategy might help novices to acquire task-specific skills faster. Also, patients with attentional deficits such as neglect might benefit from gaze guidance.

In this thesis, we have approached the development of gaze-guidance systems from an engineering viewpoint. We first created a software framework that allows flexible, yet highly efficient image processing on multiresolution pyramids both on commodity hardware as well as on dedicated graphics boards (Chapter 5). Using this software framework, we significantly advanced the state of the art in gaze-contingent displays. Prior to the work presented in this thesis, real-time space-variant filtering algorithms were limited to the introduction of spatial blur as a function of gaze. We pushed the envelope on these algorithms along two lines. First, we extended gaze-contingent displays from a modification of spatial information only to the spatio-temporal domain, which greatly increases computational complexity, but is very important because temporal information is a strong factor in oculomotor control. Second, we increased

flexibility of the filtering by moving from the specification of a single cutoff frequency per output pixel to the specification of a weighting coefficient for each individual spatio-temporal frequency band, i.e. from lowpass filtering to a fully specified frequency response, which led to a further increase in computational costs and memory requirements (Chapter 6).

With these extensions, a gaze-contingent display based on a spatio-temporal Laplacian pyramid with five spatial and four temporal scales allows the space-variant specification of 20 different coefficients for subbands of the spatio-temporal frequency spectrum in retinal coordinates. Despite this computational complexity, we achieved frame rates of 60 frames per second even on high-resolution videos with an implementation on the Graphics Processing Unit. An even more important performance measure than system throughput, however, is the system latency between an eye movement and the corresponding update on the screen. We achieved an image processing latency of only 2 ms and an overall system latency including eye tracking and screen refresh of 20–25 ms, which is even faster than for previous gaze-contingent displays of lower complexity. Such a performance improvement would not have been possible using faster hardware alone, but also required improved algorithms. In particular, we contributed novel, more efficient algorithms for temporal upsampling and the creation of a temporal Laplacian pyramid.

In summary, we have successfully established the necessary technical foundations for gaze-guidance systems. Beyond this technical achievement, further contributions were also made to the understanding of human vision. We collected a large data set of eye movements on dynamic natural scenes and investigated what factors drive oculomotor behaviour under naturalistic conditions. This was particularly important because much prior research on eye movements has focused mainly on synthetic scenes or static images as stimuli, and we could show that viewing behaviour under such circumstances is qualitatively different (Chapter 3). Using our image processing framework and advanced machine learning methods, we could also improve the state of the art in eye movement prediction based on low-level features (Chapter 4).

Finally, we used these perceptual insights and our gaze-contingent systems to perform several validation experiments. We found that certain video modifications can go unnoticed in the visual periphery, and that these modifications can change eye movement characteristics. In a first test of gaze guidance in a real-world training scenario, we were able to show that students who had received gaze guidance during training could recognize relevant image locations during test significantly faster than controls. This result shows that gaze guidance does not only affect eye movements, but is also beneficial for real-world task performance.

Now that the technical foundations for gaze guidance and a proof of concept are at hand, it remains for future work to generalize gaze guidance to a broader set of applications. Some work on putting to use both gaze-contingent displays and gaze guidance has already been started, but has been omitted from this thesis for brevity. For example, we were able to show that the introduction of peripheral temporal blur in a head-mounted display can reduce simulator sickness, which so far has been a major obstacle in the wide-spread adoption of head-mounted displays. A gaze-contingent mouse cursor that changes its size as a function of gaze to remain visible even in the periphery can be useful on large screen setups with a wide field of view and was rated positively by a group of test users. Gaze can also be used as an alternative input modality for people with motor impairments, and we could demonstrate that gaze control can beat mouse input in an open-source game that we adapted. For children with dyslexia, a display that rendered text such that only the fixated syllable or word was visible and distractors were suppressed proved motivating; in a similar fashion, we have worked on displays for patients with visual neglect that encourage them to explore the impaired hemifield more.

Ultimately, the incorporation of gaze information in general and gaze guidance in particular promise to optimize future information and communication systems.

A

Perception of multiple motions

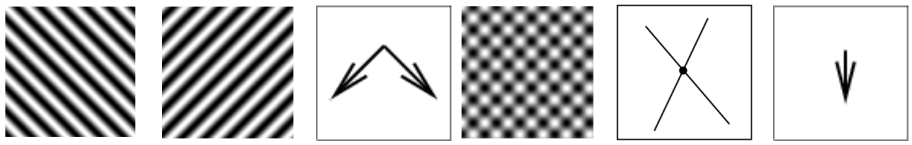
In this appendix, we shall present some of our results on the perception of multiple motions. A common class of stimuli used in experiments on motion perception is that of line gratings; on these stimuli, however, any local motion detection scheme (such as the receptive field of a motion-sensitive cell in the visual cortex) suffers from the so-called "aperture problem", which states that the veridical motion of the grating cannot be detected reliably. The superposition of several gratings (and other stimuli) gives rise to higher-order aperture problems, and we shall describe some perceptual phenomena that occur with these stimuli, and how these phenomena can be understood by the the intersection of lines (for 1D patterns such as gratings) and points (for 2D patterns) in the projective plane (see Section 2.11).

The framework that can explain these phenomena has been published in several places (Mota, Dorr, Stuke, and Barth, 2004a,b, 2003; Barth, Dorr, Vig, Pomarjanschi, and Mota, 2010 currently under submission); a conference contribution detailing the perceptual results has won a poster prize (Dorr, Stuke, Mota, and Barth, 2001).

Perception

The aperture problem has a high significance for the visual perception of motion and has been well-studied for single motions (Wuerger et al., 1996). The motion of a 1D pattern such as a line grating is inherently ambiguous (see Figure 2.9); an initial neural response simply assumes motion orthogonal to the grating, but both neural response and perception are quickly determined by the motion of the so-called terminators, i.e. the ends of the 1D patterns (Pack and Born, 2001).

For superimposed gratings, similar effects can occur, and motion percepts may be different from the directions orthogonal to the individual gratings. For



(a) (b) (c) (d) (e) (f)

Figure A.1: *If two gratings of different orientations - as shown in (a) and (b) - are moved in the directions shown in (c), the plaid pattern shown in (d) is seen as moving in the direction indicated in (f) which corresponds to the only coherent velocity that is defined by the intersection of the projective lines as shown in (e).*

example, two gratings, one moving down and to the left, the other one moving down and to the right, are perceived as a single pattern moving downwards under most experimental conditions, see Figure A.1. Three moving gratings, on the other hand, can be perceived in several different ways (Adelson and Movshon, 1982).

Two 1D transparent moving gratings

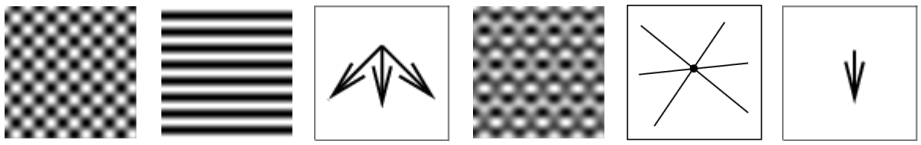
In the projective plane, two moving gratings correspond to the {line, line} case – see Table 2.2. According to the theory, the perceived motion should correspond to the intersection point the two lines and indeed it does – see Figure A.1. We shall show further examples in the following; nevertheless, for a more intuitive visualization of perceptual effects, we also recommend to use the interactive tool for multiple motion synthesis at
`http://www.inb.uni-luebeck.de/~barth/demos/ppmotion`.

Three 1D transparent moving gratings

In the case of three moving gratings, a percept of one coherent pattern only arises when all three lines intersect in the same point. This is, for example, the case for the configuration shown in Figure A.2. On the other hand, a configuration as shown in Figure A.3 has no unique percept: human observers see the three 1D patterns as moving individually or see combinations of one 1D pattern and a 2D plaid pattern.

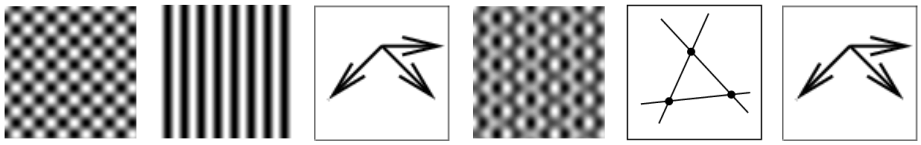
Entrainment effect for 2D patterns over 1D patterns

A spatial field of dots superimposed on a grating (see Figure A.4) corresponds to the {line, point} case. From Table 2.2, we can see that this motion configuration corresponds to a rank of J_2 of four and is thus a higher-order aperture problem. The direction of the grating by itself is not uniquely determined; if the point representing the spatial dot field falls on or close to the line representing the



(a) (b) (c) (d) (e) (f)

Figure A.2: *Coherent motion of three superimposed gratings. To the superposition of two gratings (a) a third grating shown in (b) is added. The physical motions of the three gratings are as shown in (c) and the lines of admissible velocities for each grating in (e). The percept is that of a coherent pattern as shown in (d) moving in the direction indicated by the arrow in (f). The coherent percept of one motion corresponds to the intersection of the lines in only one point.*



(a) (b) (c) (d) (e) (f)

Figure A.3: *Incoherent motion of three superimposed gratings. The sub-figures are according to those in Figure A.2. However, the directions of motions are now changed such that the lines of motion in the projective plane do not intersect in a single point (e). This makes the motions undefined and causes the percept to change dramatically such that a coherent motion is not perceived. Observers can see either of the single motions indicated in (f) (the other two motions are seen either individually or grouped to a plaid motion).*

grating in the projective plane, the grating should seem to move in coherence with the random dots. To test this hypothesis, we generated sinusoidal gratings of frequency $\xi = 1/8$, orientation $\psi = k\pi/4, k = 1, .., 8$, and a size of 10 by 10 deg visual angle. These were translated perpendicular to their orientation ($\phi_g = \psi \pm \pi/2$) with a velocity of $v_g = 1.6$ deg/s. Mean brightness of the screen was 10 cd/m^2. Then, a 2D dot pattern with same brightness distribution was overlaid to the grating and translated with direction $\phi_r = \phi_g \pm \pi/4$ and velocity $v_r = v_g/\sqrt{2}$, so that one component of the motion vector always coincided in the grating and the moving dot pattern. Fifteen of these stimuli were presented to seven human subjects for 1.6 seconds. After presentation of each stimulus, subjects had to rotate an arrow to indicate the direction of the grating they had perceived. The deviation of subjects' responses from the true direction of the grating is given in Figure A.6(a). If the dot pattern had exerted no influence on the percept for the grating at all, a single peak at 0 deg could be expected. Analogously, a single peak at 45 deg would indicate that subjects always perceived a single coherent pattern. Note that the small peak at 135 deg actually corresponds to cases of 45 deg deviation and can be attributed to the phenomenon of induced motion (the same effect that makes the platform appear moving while sitting in a moving train).

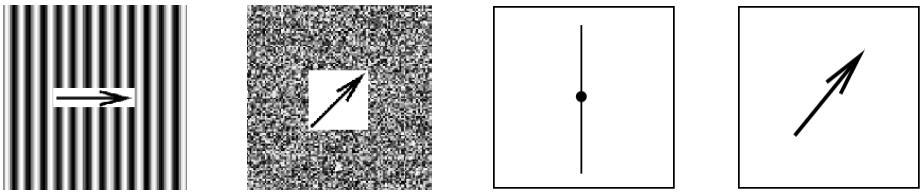
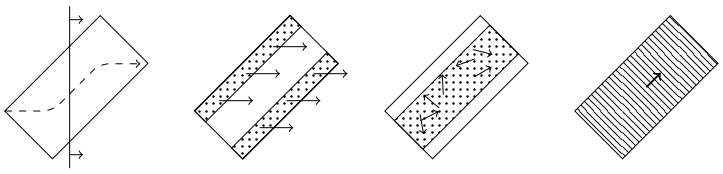


Figure A.4: *Schematic illustration of the 2D-over-1D entrainment effect. The admissible velocities for the grating (a) all lie on a line (c), the only admissible velocity for the 2D stimulus (b) is a point (c). The percept of the superposition of 1D and 2D pattern is that of a single motion (d).*

Entrainment effect and the barberpole illusion

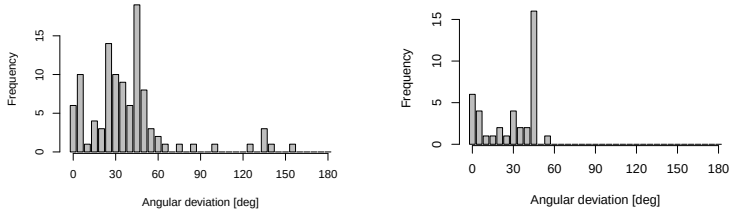
The shape of an aperture through which a grating is seen can strongly influence motion perception. This phenomenon is called the barberpole illusion. For example, the straight lines in Figure A.5 seem to change their direction along their path behind the aperture (Wuerger et al., 1996): the bar moves as indicated by the arrows and the perceived motion is indicated by the dashed line.

To show that the entrainment effect is able to override the barberpole illusion, we designed the stimuli illustrated in Figure A.5. We masked the moving grating by an aperture perpendicular to the orientation of the grating. This should strengthen the percept of motion in a direction orthogonal to the grating. As an additional modification, only the terminators of the grating were overlaid with a random dot field that moved in one coherent direction. Because this led to the rise of new terminators at the boundary of the coherent random dot field, the remaining middle of the stimulus was overlaid with a white-noise pattern, which had the same density and brightness as the coherent noise pattern. Nevertheless, the entrainment effect seen in Figure A.6(b) is still qualitatively similar to that in Figure A.6(a) which shows that the effect dominates over the influence of the aperture.



(a) (b) (c) (d)

Figure A.5: *Barberpole illusion. The veridical motion of the vertical line that is seen through the aperture is constant from left to right, but the perceived direction changes with the perceived direction of the line terminators at the aperture boundaries (a). Stimulus configuration for our entrainment experiment. Overlaid are a random dot field moving coherently from left to right that is shown only at the aperture boundaries (b), random noise in the centre of the aperture (c), and a 1D grating with an orientation perpendicular to that of the aperture (d). Both the orientation of the grating and that of the aperture should facilitate a percept of the grating as moving from bottom left to top right; experimental results in Figure A.6 show that the entrainment effect at the line terminators still dominates the percept.*



(a) (b)

Figure A.6: *Data illustrating the entrainment effect of a 2D pattern over a 1D grating. No aperture (a). Aperture orientation perpendicular to that of the 1D grating (b). Both plots have peaks at 45 deg, indicating that the 2D motion pattern (coherent random dot field) entrains the grating, even if it is superimposed only at the line terminators and if the orientation of the aperture facilitates perception of the veridical direction of the grating (0 deg deviation).*

Bibliography

Ffmpeg, 2009. URL `http://www.ffmpeg.org`.

Torque, 2009. URL `http://www.clusterresources.com`.

George Adelman, editor. *Encyclopedia of neuroscience*. Birkhäuser Boston, 1987.

Edward H Adelson and James R Bergen. The plenoptic function and the elements of early vision. In M S Landy and J A Movshon, editors, *Computational Models of Visual Processing*, pages 3–20. MIT Press, Cambridge, MA, 1991.

Edward H Adelson and Peter J Burt. Image data compression with the Laplacian pyramid. In *Proceeding of the Conference on Pattern Recognition and Image Processing*, pages 218–223. Los Angeles, CA: IEEE Computer Society Press, 1981.

Edward H Adelson and J Anthony Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300(5892):523–5, 1982.

Joseph J Atick and A Norman Redlich. What does the retina know about natural scenes? *Neural Computation*, 4:196–210, 1992.

Roland J Baddeley and Benjamin W Tatler. High frequency edges (but not contrast) predict where we fixate: A Bayesian system identification analysis. *Vision Research*, 46:2824–33, 2006.

C L Baker and O J Braddick. Eccentricity-dependent scaling of the limits of short-range motion perception. *Vision Research*, 25:803–12, 1985.

Rosario Balboa and Norberto M Grzywacz. Power spectra and distribution of contrasts of natural images from different habitats. *Vision Research*, 43: 2527–37, 2003.

Dana H Ballard and Mary M Hayhoe. Modelling the role of task in the control of gaze. *Visual Cognition*, 17(6-7):1185–204, 2009.

E Barth. The minors of the structure tensor. In G Sommer, editor, *Mustererkennung 2000*, pages 221–228, Berlin, 2000. Springer.

E. Barth. Information technology for active perception: Itap. In *First GRP-Symposium, Sehen und Aufmerksamkeit im Alter, Benediktbeuren, December 2001*, 2001.

E. Barth and A. B. Watson. A geometric framework for nonlinear visual coding. *Optics Express*, 7(4):155–165, 2000.

Erhardt Barth, Michael Dorr, Martin Böhme, Karl R. Gegenfurtner, and Thomas Martinetz. Guiding the mind's eye: improving communication and vision by external control of the scanpath. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly, editors, *Human Vision and Electronic Imaging*, volume 6057 of *Proc. SPIE*, 2006. Invited contribution for a special session on Eye Movements, Visual Search, and Attention: a Tribute to Larry Stark.

Erhardt Barth, Michael Dorr, Eleonora Vig, Laura Pomarjanschi, and Cicero Mota. Efficient coding and multiple motions. *Vision Research*, 2010. (submitted).

W Becker. Saccades. In R H S Carpenter, editor, *Vision & Visual Dysfunction Vol 8: Eye Movements*, pages 95–137. CRC Press, 1991.

David J. Berg, Susan E. Boehnke, Robert A. Marino, Douglas P. Munoz, and Laurent Itti. Free viewing of dynamic stimuli by humans and monkeys. *Journal of Vision*, 9(5):1–15, 5 2009.

J Bigün, G H Granlund, and J Wiklund. Multidimensional orientation estimation with application to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–90, 1991.

Chistopher Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

Theodore T Blackmon, Yeuk Fai Ho, Dimitri A Chernyak, Michela Azzariti, and Lawrence W Stark. Dynamic scanpaths: eye movement analysis methods. In *Human Vision and Electronic Imaging IV: SPIE Proceedings*, volume 3644, pages 511–9, 1999.

C Blakemore and F W Campbell. On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images. *Journal of Physiology*, 203(1):237–260, 1969.

Martin Böhme, Michael Dorr, Christopher Krause, Thomas Martinetz, and Erhardt Barth. Eye movement predictions on natural videos. *Neurocomputing*, 69(16–18):1996–2004, 2006a.

Martin Böhme, Michael Dorr, Thomas Martinetz, and Erhardt Barth. Gaze-contingent temporal filtering of video. In *Proceedings of Eye Tracking Research & Applications (ETRA)*, pages 109–115, 2006b.

Giorgio Bonmassar and Eric L. Schwartz. Improved cross-correlation for template matching on the Laplacian pyramid. *Pattern Recognition Letters*, 19(8): 765–770, 1998.

Gary Bradski, Adrian Kaehler, Mike Loukides, and Robert Romano. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

Neil Bruce and John Tsotsos. Saliency based on information maximization. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 155–162. MIT Press, Cambridge, MA, 2006.

Neil Bruce and John K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9(3):1–24, 2009.

Peter J Burt and Edward H Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983a.

Peter J Burt and Edward H Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983b.

Guy Thomas Buswell. *How people look at pictures: A study of the psychology of perception in art*. Chicago:University of Chicago Press, 1935.

Roxanne L. Canosa. Real-world vision: Selective perception and task. *ACM Transactions on Applied Perception*, 6(2):1–34, 2009.

Ran Carmi and Laurent Itti. The role of memory in guiding attention during natural vision. *Journal of Vision*, 6(9):898–914, 2006.

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

Martin Clauss, Pierre Bayerl, and Heiko Neumann. A statistical measure for evaluating regions-of-interest based attention algorithms. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Bernhard Schölkopf, and Martin A. Giese, editors, *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 383–390. Springer, 2004.

J. W. Cooley and J. W. Tukey. An algorithm for machine calculation of complex Fourier series. *Mathematics of Computation*, 19:291–301, 1965.

Frans W Cornelissen, Klaas J Bruin, and Aart C Kooijman. The influence of artificial scotomas on eye movements during visual search. *Optometry and Vision Science*, 82(1):27–35, 2005.

Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. Human photoreceptor topography. *The Journal of Comparative Neurology*, 292:497–523, 1990.

C Currie, G W McConkie, L A Carlson-Radvansky, and D E Irwin. Maintaining visual stability across saccades: Role of the saccade target object. Technical Report UIUC-BI-HPP-95-01, The Beckman Institute, University of Illinois, Champaign, IL, 1995.

Bart de Bruyn. Blending Transparent Motion Patterns in Peripheral Vision. *Vision Research*, 37(5):645–8, 1997.

Edmund Burke Delabarre. A method of recording eye-movements. *American Journal of Psychology*, 9(4):572–574, 1898.

Heiner Deubel. The time course of presaccadic attention shifts. *Psychological Research*, 72:630–640, 2008.

Heiner Deubel, Bruce Bridgeman, and Werner X Schneider. Different effects of eyelid blinks and target blanking on saccadic suppression of displacement. *Perception & Psychophysics*, 66(5):772–778, 2004.

D. W. Dong and J. J. Atick. Statistics of natural time-varying images. *Network: Computation in Neural Systems*, 6:345–358, 1995.

Dawei W Dong. Spatiotemporal Inseparability of Natural Images and Visual Sensitivities. In J M Zanker and J Zeil, editors, *Computational, neural & ecological constraints of visual motion processing*, pages 371–80. Springer, Berlin, 2001.

Michael Dorr, Ingo Stuke, Cicero Mota, and Erhardt Barth. Mathematical and perceptual analysis of multiple motions. In Heinrich H Bülthoff, Karl R Gegenfurtner, Hanspeter A Mallot, and Rolf Ulrich, editors, *TWK 2001 Beiträge zur 4. Tübinger Wahrnehmungskonferenz*, page 174, 2001.

Michael Dorr, Thomas Martinetz, Karl Gegenfurtner, and Erhardt Barth. Guidance of eye movements on a gaze-contingent display. In Uwe J. Ilg, Heinrich H. Bülthoff, and Hanspeter A. Mallot, editors, *Dynamic Perception Workshop of the GI Section "Computer Vision"*, pages 89–94, 2004.

Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth. Visibility of temporal blur on a gaze-contingent display. In *APGV 2005 ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, pages 33–36, 2005a.

Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth. Predicting, analysing, and guiding eye movements. In *Neural Information Processing Systems Conference (NIPS 2005), Workshop on Machine Learning for Implicit Feedback and User Modeling*, 2005b.

Michael Dorr, Martin Böhme, Thomas Martinetz, and Erhardt Barth. Gaze beats mouse: a case study. In *The 3rd Conference on Communication by Gaze Interaction - COGAIN 2007, Leicester, UK*, pages 16–19, 2007.

Michael Dorr, Eleonora Vig, Karl R Gegenfurtner, Thomas Martinetz, and Erhardt Barth. Eye movement modelling and gaze guidance. In *Fourth International Workshop on Human-Computer Conversation*, 2008.

Michael Dorr, Karl Gegenfurtner, and Erhardt Barth. The contribution of low-level features at the centre of gaze to saccade target selection. *Vision Research*, 49(24):2918–2926, 2009a.

Michael Dorr, Laura Pomarjanschi, and Erhardt Barth. Gaze beats mouse: A case study on a gaze-controlled Breakout. *PsychNology*, 7(2):197–211, 2009b.

Michael Dorr, Christoph Rasche, and Erhardt Barth. A gaze-contingent, acuity-adjusted mouse cursor. In Arantxa Villanueva, John Paulin Hansen, and Bjarne Kjær Ersbøll, editors, *5th Conference on Communication by Gaze Interaction (COGAIN)*, pages 39–41, 2009c.

Michael Dorr, Karl Gegenfurtner, and Erhardt Barth. Variability of eye movements on dynamic natural scenes. *Journal of Vision*, 2010a. (submitted).

Michael Dorr, Halszka Jarodzka, and Erhardt Barth. Space-variant spatio-temporal filtering of video for gaze visualization and perceptual learning. In *Eye-tracking research & applications*, pages 307–314, 2010b.

Valentin Dragoi and Mriganka Sur. Image structure at the center of gaze during free viewing. *Journal of Cognitive Neuroscience*, 18(5):737–48, 2006.

Andrew Duchowski and Roel Vertegaal. Eye-Based Interaction in Graphical Systems: Theory and Practice. SIGGRAPH 2000 Course Notes. http://www.vr.clemson.edu/eyetracking/sigcourse, 2000.

Andrew T Duchowski and Arzu Çöltekin. Foveated gaze-contingent displays for peripheral LOD management, 3D visualization, and stereo imaging. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(4): 1–21, 2007.

Andrew T Duchowski, Nathan Cournia, and Hunter Murphy. Gaze-contingent displays: A review. *CyberPsychology & Behavior*, 7(6):621–634, 2004.

Andrew T. Duchowski, David Bate, Paris Stringfellow, Kaveri Thakur, Brian J. Melloy, and Anand K. Gramopadhye. On spatiochromatic visual sensitivity and peripheral color LOD management. *ACM Transactions on Applied Perception*, 6(2):1–18, 2009.

W Einhäuser, M Spain, and P Perona. Objects predict fixations better than early saliency. *Journal of Vision*, 8(14):11–26, 2008a.

Wolfgang Einhäuser, Ueli Rutishauser, E. Paxon Frady, Swantje Nadler, Peter König, and Christof Koch. The relation of phase noise and luminance contrast to overt attention in complex visual stimuli. *Journal of Vision*, 6(11):1148–1158, 2006.

Wolfgang Einhäuser, Frank Schumann, Stanislavs Bardins, Klaus Bartl, Guido Böning, Erich Schneider, and Peter König. Human eye-head co-ordination in natural exploration. *Network: Computation in Neural Systems*, 18(3):267–297, 2007.

Wolfgang Einhäuser, Ueli Rutishauser, and Christof Koch. Task-demands can immediately reverse the effects of sensory-driven saliency in complex visual stimuli. *Journal of Vision*, 8(2):1–19, 2008b.

Lior Elazary and Laurent Itti. Interesting objects are visually salient. *Journal of Vision*, 8(3):1–15, 2008.

Michael Felsberg and Gösta Granlund. POI detection using channel clustering and the 2D energy tensor. In *Pattern Recognition: 26th DAGM Symposium*, volume 3175 of *LNCS*, pages 102–110, 2004.

David J Field. Relations between the statistics of natural images and the response profiles of cortical cells. *Journal of the Optical Society of America A*, 4: 2379–2394, 1987.

John M Findlay and Iain D Gilchrist, editors. *Active Vision: The Psychology of Looking and Seeing*, volume 37 of *Oxford Psychology Series*. Oxford University Press, 2003.

Tom Foulsham and Geoffrey Underwood. What can saliency models predict about eye movements? Spatial and sequential aspects of fixations during encoding and recognition. *Journal of Vision*, 8(2):1–17, 2 2008.

Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW. *Proceedings of the IEEE*, 93(2):216–231, 2005.

M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual*. Network Theory Ltd, 2009.

Dashan Gao and Nuno Vasconcelos. Decision-theoretic saliency: Computational principles, biological plausibility, and implications for neurophysiology and psychophysics. *Neural Computation*, 21(1):239–271, 2009. ISSN 0899-7667.

T W Garaas, T Nieuwenhuis, and M Pomplun. A gaze-contingent paradigm for studying continuous saccadic adaptation. *Journal of Neuroscience Methods*, 168:334–340, 2008.

Wilson S. Geisler and Jeffrey S. Perry. A real-time foveated multiresolution system for low-bandwidth video communication. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging: SPIE Proceedings*, volume 3299 of *Proc. SPIE*, pages 294–305, 1998.

Wilson S Geisler and Jeffrey S Perry. Real-time simulation of arbitrary visual fields. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 83–87. ACM, New York, NY, USA, 2002.

Wilson S Geisler, Jeffrey S Perry, and Jiri Najemnik. Visual search: The role of peripheral information measured using gaze-contingent displays. *Journal of Vision*, 6:858–73, 2006.

Filip Germeys, Peter de Graef, Sven Panis, Caroline van Eccelpoel, and Karl Verfaillie. Transsaccadic integration of bystander locations. *Visual Cognition*, 11(2-3):203–34, 2004.

Robert B Goldstein, Russell L Woods, and Eli Peli. Where people look when watching movies: Do all viewers look at the same place? *Computers in Biology and Medicine*, 3(7):957–64, 2007.

P Golland and A M Bruckstein. Motion from color. *Computer Vision and Image Understanding*, 68(3):346–62, 1997.

Chenlei Guo, Qi Ma, and Liming Zhang. Spatio-temporal Saliency Detection Using Phase Spectrum of Quaternion Fourier Transform. In *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

Uri Hasson, Ohad Landesman, Barbara Knappmeyer, Ignacio Vallines, Nava Rubin, and David J Heeger. Neurocinematics: The neuroscience of film. *Projections*, 2(1):1–26, 2008a.

Uri Hasson, Eunice Yang, Ignacio Vallines, David J. Heeger, and Nava Rubin. A hierarchy of temporal receptive windows in human cortex. *The Journal of Neuroscience*, 28(10):2539–2550, March 2008b.

Horst Haußecker and Hagen Spies. Motion. In Bernd Jähne, Horst Haußecker, and Peter Geißler, editors, *Handbook of Computer Vision and Applications*, volume 2, chapter 13, pages 309–96. Academic Press, 1999.

D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. *Computer Graphics Proceedings*, pages 229–238, 1995.

John Heminghous and Andrew T. Duchowski. iComp: a tool for scanpath visualization and comparison. In *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, page 152, New York, NY, USA, 2006. ACM.

John M Henderson and Fernanda Ferreira, editors. *The interface of language, vision, and action: Eye movements and the visual world*. Psychology Press, New York, 2004.

John M Henderson and Andrew Hollingworth. The role of fixation position in detecting scene changes across saccades. *Psychological Science*, 10(5):438–43, 1999.

John M Henderson, James R Brockmole, Monica S Castelhano, and Michael Mack. Visual saliency does not account for eye movements during visual search in real-world scenes. In R van Gompel, M Fischer, W Murray, and R Hill, editors, *Eye Movement Research: Insights into Mind and Brain*. Elsevier, 2007.

Berthold Horn and Brian Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, Aug 1981.

Hong Hua and Sheng Liu. Dual-sensor foveated imaging system. *Applied Optics*, 47(3):317–27, 2008.

Poika Isokoski and Benoît Martin. Eye tracker input in first person shooter games. In *Proceedings of The 2nd Conference on Communication by Gaze Interaction - COGAIN 2006*, 2006.

H Istance, R Bates, A Hyrskykari, and S Vickers. Snap clutch, a moded approach to solving the midas touch problem. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 221–228. New York: ACM Press, 2008.

L Itti. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 12(6):1093–1123, 2005.

Laurent Itti. Quantitative modeling of perceptual salience at human eye position. *Visual Cognition*, 14(4):959–984, 2006.

Laurent Itti and Pierre Baldi. Bayesian Surprise Attracts Human Attention. In *Advances in Neural Information Processing Systems, Vol. 19 (NIPS 2005)*, pages 547–554, Cambridge, MA, 2006. MIT Press.

Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001.

Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

Robert J K Jacob. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in Human-Computer Interaction*, 4:151–80, 1993.

Bernd Jähne. Local structure. In Bernd Jähne and Horst Haußecker, editors, *Handbook of Computer Vision and Applications*, volume 2, chapter 10, pages 209–38. Academic Press, 1999.

Bernd Jähne and Horst Haußecker, editors. *Computer Vision and Applications*. Academic Press, 2000.

Lina Jansen, Selim Onat, and Peter König. Influence of disparity on fixation and saccades in free viewing of natural scenes. *Journal of Vision*, 9(1):1–19, 1 2009.

Halszka Jarodzka, Katharina Scheiter, Peter Gerjets, Tamara van Gog, and Michael Dorr. How to convey perceptual skills by displaying experts' gaze data. In N A Taatgen and H van Rijn, editors, *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, pages 2920–2925. Austin, TX: Cognitive Science Society, 2009.

Halszka Jarodzka, Katharina Scheiter, Peter Gerjets, and Tamara van Gog. In the eyes of the beholder: How experts and novices interpret dynamic stimuli. *Journal of Learning and Instruction*, 20(2):146–154, 2010a.

Halszka Jarodzka, Tamara van Gog, Michael Dorr, Katharina Scheiter, and Peter Gerjets. Guiding attention guides thought, but what about learning? Eye movements in modeling examples. 2010b. (submitted).

E Javal. Essai sur la physiologie de la lecture. *Annales d'Oculistique*, 79:97, 1878.

J. M. Jolion and A. Montanvert. The adaptive pyramid: a framework for 2D image analysis. *CVGIP: Image Underst.*, 55(3):339–348, 1992.

Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to Predict Where Humans Look. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.

Eric R Kandel, James H Schwartz, and Thomas M Jessell, editors. *Essentials of neural science and behavior*. Prentice Hall International, 1995.

Wolf Kienzle, Felix A Wichmann, Bernhard Schölkopf, and Matthias O Franz. A Nonparametric Approach to Bottom-Up Visual Saliency. In *Advances in Neural Information Processing Systems (NIPS 2006)*, pages 689–696, Cambridge, Mass. USA, 2006. MIT Press.

Wolf Kienzle, Bernhard Schölkopf, Felix A. Wichmann, and Matthias O. Franz. How to find interesting locations in video: a spatiotemporal interest point detector learned from human eye movements. In *Proceedings of the 29th Annual Symposium of the German Association for Pattern Recognition (DAGM 2007)*, pages 405–414, Berlin, Germany, 2007. Springer Verlag.

Wolf Kienzle, Matthias O. Franz, Bernhard Schölkopf, and Felix A. Wichmann. Center-surround Patterns Emerge as Optimal Predictors for Human Saccade Targets. *Journal of Vision*, 9(5):1–15, 2009.

Kristin Koch, Judith McLean, Ronen Segev, Michael A Freed, Michael J Berry II, Vijay Balasubramanian, and Peter Sterling. How much the eye tells the brain. *Current Biology*, 16:1428–34, 2006.

Ullrich Köthe. Accurate and efficient approximation of the continuous Gaussian scale-space. In Carl E Rasmussen, Heinrich H Bülthoff, Martin Giese, and Bernhard Schölkopf, editors, *26th DAGM-Symposium*, volume 3175 of *LNCS*, pages 350–358, Heidelberg, 2004. Springer.

Manu Kumar. GUIDe saccade detection and smoothing algorithm. Technical Report CSTR 2007-03, Stanford, 2007.

Michael F Land and Mary Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41:3559–65, 2001.

Michael F Land and D N Lee. Where we look when we steer. *Nature*, 369: 742–744, 1994.

Michael F Land and Benjamin W Tatler. Steering with the head: The visual strategy of a racing driver. *Current Biology*, 11:1215–1220, 2001.

Chris Lankford. Gazetracker: software designed to facilitate eye movement analysis. In *ETRA '00: Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 51–55, New York, NY, USA, 2000. ACM.

H W Leibowitz, C A Johnson, and E Isabelle. Peripheral motion detection and refractive error. *Science*, 177:1207–8, 1972.

A Lesgold, H Rubinson, P Feltovich, R Glaser, D Klopfer, and Y Wang. Expertise in a complex skill: diagnosing X-ray pictures. In M. T. H. Chi, R. Glaser, and

M. Farr, editors, *The nature of expertise*, pages 311–342. Hillsdale, NJ: Erlbaum, 1988.

Andrei Linţu and Noëlle Carbonell. Gaze guidance through peripheral stimuli. Rapport de Recherche inria-00421151, version 30 Sep 2009, INRIA Lorraine, 2009.

Lester C Loschky and George W McConkie. User performance with gaze-contingent multiresolutional displays. In *Proceedings of Eye Tracking Research & Applications*, pages 97–103, 2000.

Lester C Loschky and G S Wolverton. How late can you update gaze-contingent multi-resolutional displays without detection? *ACM Transactions on Multimedia Computing, Communications and Applications*, 3(4):1–10, 2007.

Lester C Loschky, George W McConkie, Jian Yang, and Michael E Miller. The limits of visual resolution in natural scene viewing. *Visual Cognition*, 12(6): 1057–1092, 2005.

Hans Dieter Lüke. *Signalübertragung*. Springer-Verlag, 1999.

S. K. Mannan, K. H. Ruddock, and D. S. Wooding. The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images. *Spatial Vision*, 10:165–188, 1996.

S K Mannan, K H Ruddock, and D S Wooding. Fixation sequences made during visual examination of briefly presented 2D images. *Spatial Vision*, 11 (2):157–78, 1997.

Sophie Marat, Tien Ho Phuoc, Lionel Granjon, Nathalie Guyader, Denis Pellerin, and Anne Guérin-Dugué. Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, 82:231–43, 2009.

William R. Mark, R. Steven Glanville, Kurt Akeley, and Mark J. Kilgard. Cg: a system for programming graphics hardware in a C-like language. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 896–907, New York, NY, USA, 2003. ACM.

George W. McConkie and Keith Rayner. The span of the effective stimulus during a fixation in reading. *Perception & Psychophysics*, 17:578–586, 1975.

Ann McNamara, Reynold Bailey, and Cindy Grimm. Improving search task performance using subtle gaze direction. In *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pages 51–56, New York, NY, USA, 2008. ACM.

Ann McNamara, Reynold Bailey, and Cindy Grimm. Search task performance using subtle gaze direction with the presence of distractions. *ACM Transactions on Applied Perception*, 6(3):1–19, 2009.

Olivier Le Meur, Patrick Le Callet, Dominique Barba, and Dominique Thoreau. A coherent computational approach to model bottom-up visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):802–817, 2006. ISSN 0162-8828.

Olivier Le Meur, Patrick Le Callet, and Dominique Barba. Predicting visual fixations on video based on low-level visual features. *Vision Research*, 47(19): 2483–2498, Sept 2007.

George A Miller. The magical number seven, plus or minus two. *The Psychological Review*, 63(2):81–97, 1956.

Cicero Mota and Erhardt Barth. On the uniqueness of curvature features. In G. Baratoff and H. Neumann, editors, *Dynamische Perzeption*, volume 9 of *Proceedings in Artificial Intelligence*, pages 175–178, Köln, 2000. Infix Verlag.

Cicero Mota, Ingo Stuke, and Erhardt Barth. Analytic solutions for multiple motions. In *Proc. IEEE Int. Conf. Image Processing*, volume II, pages 917–20, Thessaloniki, Greece, October 7-10, 2001. IEEE Signal Processing Soc.

Cicero Mota, Michael Dorr, Ingo Stuke, and Erhardt Barth. Categorization of Transparent-Motion Patterns Using the Projective Plane. In Walter Dosch and Roger Y Lee, editors, *Proceedings of the ACIS Fourth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD'03)*, pages 363–369. International Association for Computer and Information Science, 2003.

Cicero Mota, Michael Dorr, Ingo Stuke, and Erhardt Barth. Categorization of Transparent-Motion Patterns Using the Projective Plane. *International Journal of Computer and Information Science*, 5(2):129–40, 2004a.

Cicero Mota, Michael Dorr, Ingo Stuke, and Erhardt Barth. Analysis and synthesis of motion patterns using the projective plane. In Bernice E Rogowitz and Thrasyvoulos N Pappas, editors, *Human Vision and Electronic Imaging Conference IX*, volume 5292 of *Proceedings of SPIE*, pages 174–81, 2004b.

Cicero Mota, Ingo Stuke, and Erhardt Barth. The Intrinsic Dimension of Multispectral Images. In *MICCAI Workshop on Biophotonics Imaging for Diagnostics and Treatment*, pages 93–100, 2006.

Susan M Munn, Leanne Stefano, and Jeff B Pelz. Fixation-identification in dynamic scenes: Comparing an automated algorithm to manual coding. In *APGV '08: Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pages 33–42, New York, NY, USA, 2008. ACM.

Stavri G Nikolov, Timothy D Newman, David R Bull, Cedric Nishan Canagarajah, Michael G Jones, and Iain D Gilchrist. Gaze-contingent display using texture mapping and OpenGL: system and applications. In *Eye Tracking Research & Applications (ETRA)*, pages 11–18, 2004.

David Noton and Lawrence Stark. Eye movements and visual perception. *Scientific American*, 224(6):34–43, 1971.

Marcus Nyström and Kenneth Holmqvist. Effect of compressed off-line foveated video on viewing behavior and subjective quality. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(1):1–14, 2010.

A. V. Oppenheim and J. S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69:529–541, 1981.

Alan V Oppenheim, Alan S Willsky, and S Hamid. *Signals and Systems*. Prentice Hall, Englewood Cliffs, NJ 07632, 2 edition, 1996.

J Kevin O'Regan, Ronald A Rensink, and James J Clark. Change-blindness as a result of 'mudsplashes'. *Nature*, 398:34, 1999.

Wilfried Osberger and Ann Marie Rohaly. Automatic detection of regions of interest in complex video sequences. In Bernice E Rogowitz and Thrasyvoulos N Pappas, editors, *Human Vision and Electronic Imaging VI*, volume 4299, 2001.

Christopher C Pack and Richard T Born. Temporal dynamics of a neural solution to the aperture problem in visual area MT of macaque brain. *Nature*, 409: 1040–2, 2001.

Derrick Parkhurst and Ernst Niebur. A feasibility test for perceptually adaptive level of detail rendering on desktop systems. In *APGV '04: Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, pages 49–56, New York, NY, USA, 2004. ACM.

Derrick Parkhurst, Klinton Law, and Ernst Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42:107–23, 2002.

Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In Bernice E Rogowitz and Thrasyvoulos N Pappas, editors, *Human Vision and Electronic Imaging: Proceedings of SPIE, San Jose, CA*, volume 4662, pages 57–69, 2002.

Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(8):2397–416, 2005.

Marc Pomplun, Helge Ritter, and Boris Velichkovsky. Disambiguating complex visual information: Towards communication of personal views of a scene. *Perception*, 25:931–48, 1996.

Charles Poynton. *Digital Video and HDTV*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2003.

Claudio M Privitera and Lawrence W Stark. Algorithms for defining visual regions-of-interest: Comparison with eye fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):970–982, 2000.

Umesh Rajashekar, Lawrence K Cormack, and Alan C Bovik. Point of gaze analysis reveals visual search strategies. In Bernice E Rogowitz and Thrasyvoulos N Pappas, editors, *Proc. SPIE Human Vision and Electronic Imaging IX, San Jose, CA*, volume 5292, pages 296–306, 2004.

Umesh Rajashekar, Ian van der Linde, Alan C Bovik, and Lawrence K Cormack. Foveated analysis of image features at fixations. *Vision Research*, 47(25):3160–72, 2007.

Rameshsharma Ramloll, Cheryl Trepagnier, Marc Sebrechts, and Jaishree Beedasy. Gaze data visualization tools: Opportunities and challenges. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference*, pages 173–180, Washington, DC, USA, 2004. IEEE Computer Society.

Christoph Rasche and Karl R Gegenfurtner. Visual orienting in dynamic broadband (1/f) noise sequences. *Attention, Perception & Psychophysics*, 71(1):100–113, 2010.

Keith Rayner. The perceptual span and peripheral cues in reading. *Cognitive Psychology*, 7:65–81, 1975.

Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological Bulletin*, 124(3):372–422, 1998.

Pamela Reinagel and Anthony M Zador. Natural scene statistics at the centre of gaze. *Network: Computation in Neural Systems*, 10:341–350, 1999.

Eyal M Reingold, Lester C Loschky, George W McConkie, and David M Stampe. Gaze-contingent multiresolutional displays: An integrative review. *Human Factors*, 45(2):307–28, 2003.

Martin Rolfs. Microsaccades: Small steps on a long way. *Vision Research*, 49: 2415–2441, 2009.

Michele Rucci, Ramon Iovin, Martina Poletti, and Fabrizio Santini. Miniature eye movements enhance fine spatial detail. *Nature*, 447(7146):851–4, 2007.

Javid Sadr and Pawan Sinha. Object recognition and Random Image Structure Evolution. *Cognitive Science*, 28:259–87, 2004.

Dario D. Salvucci and Joseph H. Goldberg. Identifying fixations and saccades in eye-tracking protocols. In *ETRA '00: Proceedings of the 2000 symposium on Eye tracking research & applications*, pages 71–78, New York, NY, USA, 2000. ACM.

Anthony Santella and Doug DeCarlo. Robust clustering of eye movement recordings for quantification of visual interest. In *ETRA '04: Proceedings of the 2004 symposium on Eye tracking research & applications*, pages 27–34, New York, NY, USA, 2004. ACM.

E. Schneider, T. Villgrattner, J. Vockeroth, K. Bartl, S. Kohlbecher, S. Bardins, H. Ulbrich, and T. Brandt. EyeSeeCam: An eye movement-driven head camera for the examination of natural visual exploration. *Annals of the New York Academy of Sciences*, 1164:461–467, 2009.

B. Schölkopf and A. Smola. *Learning with kernels*. MIT University Press, Cambridge, 2002.

Hae Jong Seo and Peyman Milanfar. Nonparametric bottom-up saliency detection by self-resemblance. In *Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1st International Workshop on Visual Scene Understanding, Miami, 2009*, pages 45–52, 2009.

Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

Claude E Shannon. Communication in the presence of noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.

Hamid R Sheikh, Brian L Evans, and Alan C Bovik. Real-time foveation techniques for low bit rate video coding. *Real-Time Imaging*, 9(1):27–40, 2003.

Eero P Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *Proc 31st Asilomar Conf on Signals, Systems and Computers*, volume 1, pages 673–8. IEEE Computer Press, 1997.

Eero P Simoncelli, William T Freeman, Edward H Adelson, and David J Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, 38 (2):587–607, 1992.

J D Smith and T C Graham. Use of eye movements for video game control. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, page 20. New York: ACM Press, 2006.

Lew B Stelmach and Wa James Tam. Processing image sequences based on eye movements. In *Human Vision, Visual Processing and Digital Display*, volume 2179 of *Proceedings of the SPIE*, pages 90–8. IEEE Computer Press, 1994.

Lew B Stelmach, W James Tam, and Paul J Hearty. Static and dynamic spatial resolution in image coding: An investigation of eye movements. In *Human Vision, Visual Processing and Digital Display II*, volume 1453 of *Proceedings of the SPIE*, pages 147–52. IEEE Computer Press, 1991.

Leland S. Stone, Frederick A. Miles, and Martin S. Banks. Linking eye movements and perception. *Journal of Vision*, 3(11):i–iii, 2003.

Sara L. Su, Frédo Durand, and Maneesh Agrawala. De-emphasis of distracting image regions using texture power maps. In *Texture 2005: Proceedings of the 4th IEEE International Workshop on Texture Analysis and Synthesis in conjunction with ICCV'05*, pages 119–124, October 2005.

Bernard Marius 't Hart, Johannes Vockeroth, Frank Schumann, Klaus Bartl, Erich Schneider, Peter König, and Wolfgang Einhäuser. Gaze allocation in natural stimuli: Comparing free exploration to head-fixed viewing conditions. *Visual Cognition*, 17(6/7):1132–1158, 2009.

Sovira Tan, Jason L Dale, and Alan Johnston. Performance of three recursive algorithms for fast space-variant Gaussian filtering. *Real-Time Imaging*, 9(3): 215–228, 2003.

Benjamin W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):1–17, 11 2007.

Benjamin W Tatler and Benjamin T Vincent. Systematic tendencies in scene viewing. *Journal of Eye Movement Research*, 2(2):1–18, 2008.

Benjamin W Tatler and Benjamin T Vincent. The prominence of behavioural biases in eye guidance. *Visual Cognition*, 17(6-7):1029–54, 2009.

Benjamin W Tatler, Roland J Baddeley, and Iain D Gilchrist. Visual correlates of fixation selection: effects of scale and time. *Vision Research*, 45:643–59, 2005.

Benjamin W Tatler, Roland J Baddeley, and Benjamin T Vincent. The long and the short of it: Spatial statistics at fixation vary with saccade amplitude and task. *Vision Research*, 46:1857–62, 2006.

Monica A. Trifas, John M. Tyler, and Oleg S. Pianykh. Applying multiresolution methods to medical image enhancement. In *ACM-SE 44: Proceedings of the 44th annual Southeast regional conference*, pages 254–259, New York, NY, USA, 2006. ACM.

Po-He Tseng, Ran Carmi, Ian G. M. Cameron, Douglas P. Munoz, and Laurent Itti. Quantifying center bias of observers in free viewing of dynamic natural scenes. *Journal of Vision*, 9(7):1–16, 7 2009.

G Underwood, P Chapman, N Brocklehurst, J Underwood, and D Crundall. Visual attention while driving: Sequences of eye fixations made by experienced and novice drivers. *Ergonomics*, 46:629–646, 2003.

L. G. Ungerleider and M. Mishkin. Two cortical visual systems. In D. J. Ingle, M. A. Goodale, and R. J. W. Mansfield, editors, *Analysis of Visual Behavior*, page 549586. MIT Press: Cambridge, MA, 1982.

Tamara van Gog, Halszka Jarodzka, Katharina Scheiter, Peter Gerjets, and F Paas. Effects of attention guidance during example study by showing students the models' eye movements. *Computers in Human Behavior*, 25:785–791, 2009.

B Velichkovsky, M Pomplun, and J Rieser. Attention and communication: Eye-movement-based research paradigms. In W H Zangemeister, H S Stiehl, and C Freksa, editors, *Visual Attention and Cognition*, pages 125–54. Amsterdam, Netherlands: Elsevier Science, 1996.

Eleonora Vig, Michael Dorr, and Erhardt Barth. Efficient visual coding and the predictability of eye movements on natural movies. *Spatial Vision*, 22(5): 397–408, 2009.

Roman von Wartburg, Pascal Wurtz, Tobias Pflugshaupt, Thomas Nyffeler, Mathias Lüthi, and René Müri. Size matters: Saccades during scene perception. *Perception*, 36:355–65, 2007.

O Špakov and D Miniotas. Visualization of eye gaze data using heat maps. *Electronica & Electrical Engineering*, 2:55–58, 2007.

Oleg Špakov and Kari-Jouko Räihä. KiEV: A tool for visualization of reading and writing processes in translation of text. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 107–110, New York, NY, USA, 2008. ACM.

Brian A Wandell. *Foundations of Vision*. Sinauer Associates, 1995.

Felix A Wichmann, L T Sharpe, and Karl R Gegenfurtner. The contributions of color to recognition memory for natural scenes. *Journal of Experimental Psychology: Learning, Memory & Cognition*, 28:509–520, 2002.

Lance Williams. Pyramidal parametrics. *Computer Graphics*, 17(3):1–11, 1983.

David S Wooding. Eye movements of large populations: II. Deriving regions of interest, coverage, and similarity using fixation maps. *Behavior Research Methods, Instruments, & Computers*, 34(4):518–28, 2002a.

David S Wooding. Fixation maps: Quantifying eye-movement traces. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, pages 31–36, New York, NY, USA, 2002b. ACM.

Sophie Wuerger, Robert Shapley, and Nava Rubin. On the visually perceived direction of motion by Hans Wallach: 60 years later. *Perception*, 25:1317–67, 1996.

Shun-Nan Yang. Effects of gaze-contingent text changes on fixation duration in reading. *Vision Research*, 49(23):2843–2855, 2009.

Alfred L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, 1967.

Christoph Zetzsche and Erhardt Barth. Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research*, 30:1111–1117, 1990.

Christoph Zetzsche, Erhardt Barth, and Bernhard Wegmann. The importance of intrinsically two-dimensional image features in biological vision and picture coding. In Andrew B. Watson, editor, *Digital Images and Human Vision*, pages 109–38. MIT Press, October 1993.

Christoph Zetzsche, Gerhard Krieger, and Bernhard Wegmann. The atoms of vision: Cartesian or polar? *Journal of the Optical Society of America A*, 16(7): 1554–1565, 1999.

Lingyun Zhang, Matthew H. Tong, Tim K. Marks, Honghao Shan, and Garrison W. Cottrell. SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, 8(7):1–20, 2008.

Lingyun Zhang, Matthew H. Tong, and Garrison W. Cottrell. SUNDAy: Saliency Using Natural Statistics for Dynamic Analysis of Scenes. In *Proceedings of the 31st Annual Cognitive Science Conference, Amsterdam, Netherlands*, 2009.