Retina-inspired image and video coding

Effrosyni Doutsi

UNIVERSITÉ CÔTE D'AZUR

GRADUATE SCHOOL STIC

SCIENCE ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

PhD THESIS

A dissertation submitted in partial satisfaction of the requirements for the degree of

Doctor of Science

Specialized in Signal and Image Processing

presented by

Effrosyni Doutsi

Retina-Inspired Image and Video Coding

supervised by

Pr. Lionel Fillatre

and

Dr. Marc Antonini

prepared at

I3S Laboratory, SIS Team, Mediacoding Group

Funded by 4G-TECHNOLOGY and ANRT Grant

Defended on March 22, 2017

Committee:

Referees :	Pr. Janusz Konrad	-	Boston University
	Dr. Laurent Perrinet	-	CNRS, Aix-Marseille Université
Directors :	Pr. Lionel Fillatre	-	I3S, Université Côte d'Azur
	Dr. Marc Antonini	-	I3S, CNRS
Examiners :	Pr. Béatrice Pesquet	-	Télécom ParisTech
	Dr. Pierre Kornbropst	-	Inria
	Pr. Fernando Pereira	-	University of Lisbon
Invited :	Mr. Julien Gaulmin	-	4G-TECHNOLOGY

Contents

Abstract

The goal of this thesis is to propose a novel video coding architecture which is inspired by the visual system and the retina. If one sees the retina as a machine which processes the visual stimulus, it seems an intelligent and very efficient model to mimic. There are several reasons to claim that, first of all because it consumes low power, it also deals with high resolution inputs and the dynamic way it transforms and encodes the visual stimulus is beyond the current standards. We were motivated to study and release a retina-inspired video codec. The proposed algorithm was applied to a video stream in a very simple way according to the coding standards like MJPEG or MJPEG2000. However, this way allows the reader to study and explore all the advantages of the retina dynamic processing way in terms of compression and image processing. The current performance of the retina-inspired codec is very promising according to some final results which outperform MJPEG for bitrates lower than 100 kbps and MPEG-2 for bitrates higher than 70 kpbs. In addition, for lower bitrates the retina-inspired codec outlines better the content of the input scene. There are many perspectives which concern the improvement of the retina-inspired video codec which seem to lead to a groundbreaking compression architecture. Hopefully, this manuscript will be a useful tool for all the researchers who would like to study further than the perceptual capability of the visual system and understand how the structure and the functions of this efficient machine can in practice improve the coding algorithms.

Résumé

Cette thèse vise à proposer une nouvelle architecture de codage vidéo qui s'inspire du système visuel et de la rétine. La rétine peut être considérée comme une machine intelligente qui traite le stimulus visuel de façon très efficace. De ce fait, elle représente donc un bon candidat pour développer de nouveaux systèmes de traitement d'image. Il y a plusieurs raisons pour cela, tout d'abord parce qu'elle consomme peu d'énergie, elle traite également des entrées haute résolution et sa façon de transformer et d'encoder de manière dynamique le stimulus visuel dépasse les normes actuelles. Nous avons été motivés pour étudier et proposer un codec vidéo inspiré de la rétine. L'algorithme proposé a été appliqué à un flux vidéo d'une manière très simple, suivant le principe des standards de codage MJPEG ou MJPEG2000. Cette approche permet au lecteur d'étudier et d'explorer tous les avantages du traitement dynamique de la rétine en termes de compression et de traitement d'image. La performance actuelle du codec que nous avons développé est très prometteuse. Les résultats montrent des performances supérieures à MJPEG pour des débits inférieurs à 100 kbps et MPEG-2 pour des débits supérieurs à 70 kpbs. De plus, à faibles débits le codec proposé décrit mieux le contenu de la scène d'entrée. De nombreuses perspectives sont proposées afin d'améliorer ce codec inspiré de la rétine qui semblent conduire à un nouveau paradigme de compression vidéo. Nous espérons que ce manuscrit sera un outil utile pour tous les chercheurs qui voudraient, au delà de l'étude de la capacité perceptive du système visuel et, comprendre comment la structure et les fonctions de cette machine efficace peuvent en pratique améliorer les algorithmes de codage.

Acknowledgment

Firstly, I would like to express my sincere gratitude to my supervisors Pr. Lionel Fillatre and Dr. Marc Antonini for the continuous support of my Ph.D study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I feel honored and thankful for giving me the opportunity to explore the wonderful world of research that I hope I will keep serving it for the rest of my life. In addition, I would also like to express my appreciation to the industrial partner of my thesis, 4G-TECHNOLOGY, and especially Julien Gaulmin, who was the representative of the company during these three years, for our excellent collaboration.

Besides my professors and partners, I would like to thank the rest of my thesis committee mentioned in an alphabetical order: Pr. Janusz Konrad, Dr. Pierre Kornprobst, Pr. Fernando Pereira, Dr. Laurent Perrinet and Pr Batrice Pesquet-Popescu, for their insightful comments and encouragement, but also for the hard question which incented me to widen my research from various perspectives.

I thank my lab-mates first of all for their friendship. I will never forget all the precious moments we had together; the stimulating discussions, the coffee breaks, the long dinners, the holidays, the rugby games, all the amusing and difficult moments/days we spent together. They will always have a special place in my heart and mind. I also thank the big, fat, Greek, community. I feel so lucky and thankful I met all these people, they enlightened my days in Côte d'Azur. A big thank to all my friends in Greece who have been always supportive and encouraging.

Last but not least, I need to dedicate this PhD thesis to my family; my parents Elli and Giannis, my sister Mary and my beloved Lampros. Thank you for all your sacrifices you have made on my behalf, thank you for believing in me and heartening me when I was fading up. Thank you for asking me every single day about my progress. Thank you for sharing my anxieties. Thank you for being there for me anytime.

"You, your joys and sorrows, your memories and your ambitions, your sense of personal identity and your free will are in fact no more than the behavior of a vast assembly of nerve cells and their associated molecules. So all that we have here, basically, is a collection of, interacting elements, molecules, proteins, nerve cells, synapses. All of us, our emotions, all of us, our creativity, all of us, our sorrows, come from the physics of the brain."

Francis Crick, Nobel Laureate 1962, - "The father of DNA"

"Machine think? You bet! We are machines and we think, don't we"

Claude Shannon, - "The father of information age"

Chapter 1

Introduction

1.1 General Framework

During the last years the progress of video compression algorithms has become very challenging since the improvement of the already existing standards seems to be very difficult and time consuming. The latest standard HEVC was released in 2013, ten year later than its precedent H.264 was standardized. HEVC outperforms its precedent in terms of bitrates almost 50% which was an important achievement. However, this progress is supposed to be insufficient comparing to the compression needs which were raised during the last decade. The improvement of the technological equipment including 4K cameras, Facebook Surround 360 camera, UHDTV, smart-phones, the wide use of internet and all the social media, the extensive role of video surveillance cameras in security systems, etc. are some of the most important issues which require higher progress of the video compression algorithms.

There are many research teams all over around the world which aim to figure out which are the drawbacks of HEVC in order to improve them. One of the most important cons of HEVC is the encoding time. The encoding speed of this algorithm is much lower than the one of H.264 due to its high complexity. This is a common phenomenon which has been observed during the progress of the video compression algorithms. The complexity is the parameter which is usually sacrificed while one tries to achieve higher quality and/or bitrates. Google recently released VP9 which seems to be able to tackle some speed limitations and they recently announced the development of VP10 which is expected to reach the bitrate performance of HEVC. However, the encoding speed is not the only drawback of HEVC. Another important issue which rises is the power consumption of the latest compression algorithms. The reduction of the battery life which is caused due to the encoding and decoding of video streams is a serious problem for many devices with energy constraints, like cellphones, tablets, laptops, nomadic cameras, etc. This power consumption is related to the complexity of the algorithms which increases throughout the years.

Generally, the future of the compression algorithms seems to be foggy. Although numerous of techniques are represented in annual meetings and conferences each one of which seem to perform better or to be more efficient or to improve each of the quality, bitrate, complexity, encoding or decoding time of HEVC, finally the total gain is not adequate to replace HEVC. People think that the combination of all these new techniques could probably result in a new standard. To our point of view, while the encoding architecture remains the same it would be really tough to enhance the coding performance. Motion estimation, which is the heart of all the current coding architectures, seem to be responsible not only for the high complexity but also for the sedate progress of these systems. A video stream is considered to be as a sequence of pictures with high temporal redundancy. Motion estimation is the way to reduce this redundancy between sequential pictures. For higher reconstruction accuracy the motion estimation happens within small blocks of 8x8, 4x4 or even 2x2 pixels of an entire picture. However, the resolution of signals continues

increasing resulting in high computational cost while motion estimation is computed. We need to seek for groundbreaking solutions and novel architectures which may cause more efficient results.

In this thesis, we propose a novel architecture which aims to treat a video stream in a different way. A video is a dynamic signal which changes with respect to time. We search for a model which allows to process a video stream in a dynamic way. In other words, we aim to dynamically encode a video stream in order to avoid the exhaustive comparisons within key and predicted frames as it happens in motion compensation. This novel coding system is inspired by the visual system and the way the input visual system is encoded and transmitted to the brain. In more details, the visual stimulus, which is a continuous luminence of light that reaches the eyes, is captured by the innest tissue layer which is called retina. The retina is a multilayer structure which consists of different kind of cells. There is a high variety in the shape and the functionality of these cells. However, there are cells which contribute to the same processing step and for that reason they have been grouped to the same layer. There are three different processing layers: the Outer Plexiform Layer (OPL), the Inner Plexiform Layer (IPL) and the Ganglionic Layer (GL). Each one of the above layers is necessary in order to efficiently include all the necessary information concerning the input signal into a code which is going to be propagated to the visual cortex of the brain where it is analyzed.

To highlight the similarities between the retina processing and a compression algorithm we need to detail the role of each retina layer. The cells of OPL layer are responsible to dynamically transform the input luminence of light into current. This current under some non-linearities and feedback mechanisms which occur in the IPL cells are propagated to the GL. The GL is the processing layer where the current is dynamically transformed into continuous electrical impulses which are called spikes. The code of spikes is the only source of information which is sent to the brain through the optic nerve and the visual pathway. The visual pathway consists of approximately more than 10 processing layers which are able to enrich the efficiency of the information which is carried on the code of spikes. However, the first copy of the code is released at the retina level. Summing up, the retina first captures the visual stimulus which is dynamically transformed and encoded into a code of spikes which, in fact, is a kind of a binary code: absence or presence of a spike. This processing chain is very similar to the conventional coding principle which is used in compression algorithms. The first step of this principle is the transformation of the input signal into a more compressible domain using Discrete Wavelet Transforms (DWT), Discrete Cosine/Sine Transforms (DCT/DST), etc. The redundancy of these transforms is eliminated using quantization methods and then the entropy coding allows to represent the quantized signal into binary code which is a sequence of "0" and "1". Although these two architectures seem to follow the same principles, they have a major difference: the retina enables an on the fly spatiotemporal processing of the visual stimulus, while the compression algorithms treat their inputs first in space and then in time.

In this thesis, we propose a novel "Retina-inspired Video COder/DECoder (CODEC)". This codec consists of two basic models which have been derived by neuromathematical equations under given assumptions. The first model is a novel non-separable spatiotemporal OPL retina-inspired transform and it is simply termed retina-inspired filtering. The second model is a dynamic quantization which is based on the Leaky Integrate and Fire (LIF) spike generator mechanism and it is called LIF Quantizer (LIFQ). Both these models have been proposed under the strong assumption that the input signal is constant in time. This assumption serves an important purpose which is first of all, to interpret how the retina neurons work and why the neuromathematical models have been built in this way. Then, we need to realize what kind of information we are able to extract from the input signal and of course this assumption also simplifies the study of both these techniques in terms of signal processing.

1.2 Contributions and Outline

This manuscript is separated into four different parts. Each part has a complete and independent content that it could be also read separately by the reader. Part I of this thesis is a chapter dedicated to the state-of-the-art in compression. The perspective of this chapter is to introduce the important role of compression in technology. First, there is a description of different compression formats. Secondly, we present the basic coding principle which has been adapted by all the standards in image and video compression algorithms. This chapter also describes some metrics which are used in order to evaluate and compare the performance of compression algorithms. Last but not least, we conclude this state-of-the-art with an overview in the progress of video compression algorithms and a short discussion about the current and future expectations of video compression systems. Our contributions in video compression are introduced in parts II, III and IV.

1.2.1 Part II: Dynamic Filtering

This part aims to introduce the first processing step of the retina-inspired video codec which is the retina-inspired filtering. We decided to split this part in three different parts to be easier for the reader to understand our filter. Chapter 3 is an introduction to the Outer Plexiform Layer (OPL) retinal transform. We provide the background concerning the neuroscientific models which have been proposed in order to describe the way the input light is transformed into current through the first group of the retina cells which belong to the OPL. The behavior of these cells seems to be precisely described by a non-separable spatiotemporal Difference of Gaussian (DoG) filter. We introduce the special characteristics of this filter to the image processing community and we compare it to the already existed DoG based models.

Chapter 4 derives a novel non-separable spatiotemporal OPL retina-inspired filter from neuroscientific models under the assumption that the input signal is constant with respect to time. This filter which is simply termed as retina-inspired filter is a family of Weighted DoGs (WDoG). We have proven that this filter is simply a group of DoG which are weighted by two temporal functions. These temporal functions are responsible to change the shape of the DoG with respect to time. The impact of this temporal evolution is strong concerning the kind of information we are able to extract from the input signal. We have proven that this behavior is due to the bandwidth of the retina-inspired filter which also evolves in time. We first represent some numerical results of this transform applied to 1D signal which is straightforward. Then, we extend this approach to still-images and images retrieved from video streams, which, in this document, are called pictures. This is the first time, according to our knowledge, that people propose such a filter with a dynamic behavior which evolve in time according to the OPL retina cells.

The last but not least chapter of part II, chapter 5, is the key to our great novelty. As we discussed before, this is not only the first time a dynamic retina-inspired transform is released but we also mathematically prove that this transform is invertible which is necessary and of a high interest for signal processing community. We used the frame theory in order to prove that the retina-inspired decomposition is bounded. We calculated the analytic expression of the lower and the upper bounds and we illustrate the perfect reconstruction of an input signal when we use the full retina-inspired frame. In addition, we extended this study for some noisy cases. We used some Additive White Gaussian Noise (AWGS) to each decomposition layer to test the impact of the noise in terms of reconstruction. The results showed that the redundancy of the retina-inspired decomposition is efficient enough to guarantee high reconstruction quality even in the presence of noise. Finally, we have also illustrated some results of progressive reconstruction. The retina-inspired filter depends on time and we have proven that the perfect reconstruction exists when the retina-inspired decomposition is complete, meaning that the optimal result occurs at the time just before

the filter disappears. To our point of view, it was also interesting to show how the reconstruction is improved while time increases until we reach the perfect reconstruction. The results show that when the filter starts to evolve, the reconstruction is poor but it allows to outline the objects inside the input scene. However, while time increases, more texture information is carried on the retina-inspired decomposition which finally allows the perfect reconstruction.

1.2.2 Part III: Dynamic Quantization

This part consists of two chapters which aim to introduce the encoding process which takes place in the retina and the interpretation of this process in terms of quantization. The aim of the retina coding is to transmit enough information about the input stimulus on the retina to allow object and events to be identified. This information can be conveyed by analog electrical mechanisms locally, but over long distances it should be encoded into spatiotemporal spike trains which are generated by a population of neurons. Ganglion cells are the only retinal neurons which are able to produce spikes.

In this thesis, we are interested in adopting in the conventional coding principle a model which describes how the neurons spike. This model will reduce the spatiotemporal redundancy of the retina-inspired transformed input signal generating a code of spikes. This code will be used to reconstruct the input signal with the minimum distortion. Thus, we need a model which allows to interpret the code of spikes by providing a link between the input signal and the firing rate. In chapter 6, we describe that there have been proposed several non-linear, linear, stochastic, non-stochastic, rank order or time order coding systems which model the generation of spikes. However, it seems that one of the most efficient and easier to be adjusted to the conventional coding principle is the Leaky Integrate and Fire (LIF) model. The LIF model is based on the exact time each neuron emits its spike. This time carries all the necessary information about the intensity of the input. The higher the input intensity is, the sooner the spike will be emitted. If we assume that a neuron is inhibited just after the release of its spike, then for a given observation window a high intensity signal will generate large number of spikes comparing to lower intensities. The LIF model performs very well even under some time constraints. In other words, when the observation window is very small and some of the neurons will spike only ones, the produced code will be still efficient to interpret and assign each spike to an input intensity due to the delay. Last but not least, this chapter finishes with section 6.5 which is dedicated to some related works in compression where spikes are also used.

In Chapter 7, we propose a retina-inspired quantizer being motivated by the LIF model which is a time encoder of spikes. This model allows to map the input intensity to the spike arrival time. Thus, the delay of each spike arrival is a clue to interpret a firing rate and reconstruct the neural code. In this thesis, we aim to link the neuroscientific LIF model with the conventional quantization. This connection results in the construction of a retina-inspired quantizer, which is termed as *LIF dead-zone quantizer* or *LIF-quantizer* or *LIFQ*. We also explain how the LIFQ is applied to the retina-inspired frame in order to reduce its redundancy. Depending on the value of the quantization step, we propose three different kind of LIFQ: the first one is called *perfect-LIF dead-zone quantizer* or *perfect-LIFQ* which is similar to the Integrate and Fire (IF) or Threshold And Fire (TAF) model. It is called perfect because a threshold θ is the only criterion to discard some intensities. Another, more advanced model is the *uniform-LIF dead-zone quantizer* or *uniform-LIFQ*. In this model, the intensities which exit the threshold θ are quantized using a given quantization step q which is unique for all the decomposition layers. In addition, we present the *adaptive-LIF dead-zone quantizer* or *adaptive-LIFQ*, which adapts the value of the quantization step with respect to the energy of each decomposition layer. We first propose some experimental evolution of the value of the quantization step for each subband. Then, we describe the methodology which is related to the bit-allocation optimization, tuning the quantization

step according to the energy of each subband. This is called *Optimized-LIF dead-zone quantizer* or *optimized-LIFQ*. We also present some numerical results to defend the efficiency of all the above LIFQ models. Last but not least, the comparison between the performance of the the retina-inspired codec when it is applied to a still-image and JPEG or JPEG2000 shows that our model performs much better for given bitrates.

1.2.3 Part IV: Applications

Chapter 8 introduces video surveillance systems as an application of the retina-inspired codec. We have chosen these systems due to the 4G-TECHNOLOGY, which is the industrial partner of this thesis. The beginning of this chapter is a state-of-the-art in video surveillance systems and some recent technologies which aim to maximize the received video quality under the resource limitations. These technologies provide power-efficiency solutions which is the major concern of nomadic video surveillance systems. In addition, this chapter represents EViBOX which is a system patented and provided by 4G-TECHNOLOGY. EViBOX can be advanced by the the retina-inspired systems and we propose several possible ways which concern this progress.

There are many perspectives which are important to be studied concerning the applications of the retina-inspired codec in video surveillance systems. We propose many different ways of how these systems could be advanced. The most important contributions would be the elimination of the static background and the enhancement of the Regions Of Interests (ROIs) due to the dynamic behavior of our codec. Last but not least, we illustrate some comparison results concerning well-known video streams which have been coded with MJPEG, MPEG-2 and the retina-inspired codec. Our coding system outperforms MJPEG and it provides much better visual results than MPEG-2.

Part I

STATE-OF-THE-ART

Chapter 2

State-of-the-art and Background in Compression

Contents

This chapter introduces the important role of compression in technology. First, there is a description of different compression formats. Secondly, we present the basic coding principle which has been adapted by all the standards in image and video compression algorithms. This chapter describes also some qualitative metrics which are used in the evaluation and comparison of the performance of compression algorithms. Last but not least, we conclude the state-of-the-art in video compression with an overview of video compression algorithms and a short discussion about the future in video compression.

2.1 Why we use compression?

During the last decades, technology has been advanced making people's life better and by far easier than it used to be in the past. High resolution photos and videos are within the aspects of this progress which have been adapted by most of the technological device. Some interesting examples are the following: analog television was replaced by digital television with a broadcasting not only through cables but also satellites and internet which offer high variety of channels and TV programs. Video tapes which were used in order to store TV programs or movies changed into CDs, DVDs, Blu-Ray Discs(BDs) and many alternative ways (hard-disks, keys) to store digital videos. Cellphones are used not only for calls and SMS but as pocket computers, cameras, web browsers, social network devices, navigation systems, etc. According to the trends of our time, people not only like to capture big amount of data but also to exchange them through the social media (Fig. 2.1). The home and cellular internet access and speed is also in progress allowing a widespread use of video-based web applications. In addition, most of the web applications, games, bank services and social networks change dynamically. Moreover, some of them, like Skype, Viber, iChat, Messenger, etc. allow live streaming communication. Thus, one should be able to compress these data in very efficient formats in order to store them or transmit them over given bandwidth channels.

Video compression or encoding is the process of reducing the amount of data required to represent a digital video signal, prior to transmission or storage. The complementary operation is the decoding which recovers a digital video stream from a compressed representation. An effective video coding is an essential component of all the technological devices that make a difference between the success or the failure of a business model. In general, compression is performed to remove the redundancy inherent in the input signal. There exist statistical, spatial and temporal redundancy which are eliminated by two different methods of compression: the lossless and the lossy.

2.1.1 Lossless compression

Many types of data contain statistical redundancy which can be effectively compressed using lossless compression. The principle of the lossless compression is to minimize the number of bits required to represent the original input signal without any loss of information. Lossless compression is also called reversible process. However, concerning the visual or audio human systems it is possible that a significant loss would not interfere with perception of

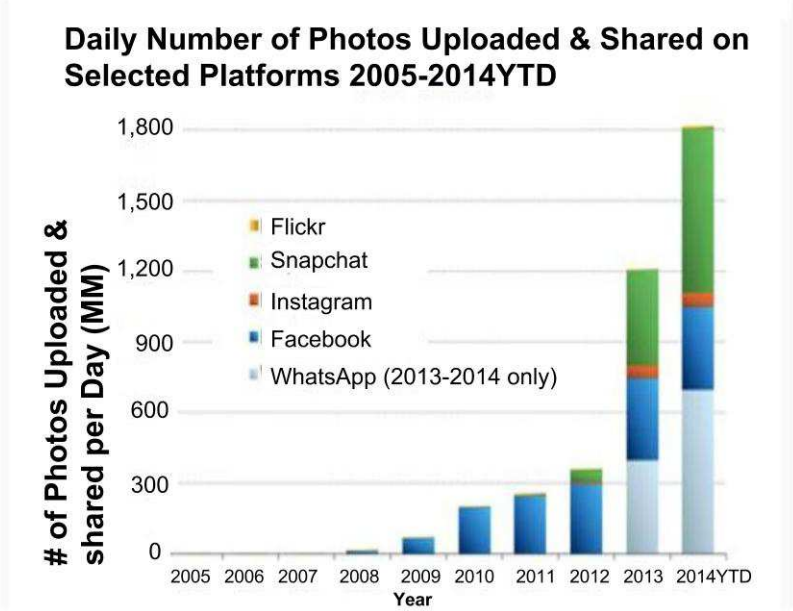


Figure 2.1: This graph illustrates the increase of the amount of images which are uploaded and shared every day through social media. Source: KPCB estimates based on publicly disclosed company data, 2014 YTD data per latest as of 5/14.

the output signal. In addition, one should always keep in mind the fact that the real world is converted into digital world which also results in a imperfect output signal. The lossless compression is required in applications related to medical data transmission, in which a loss of information may cause a wrong medical diagnosis. Lossless compression sets a trade-off between 3 different dimensions: the coding efficiency (entropy), the coding complexity (computational cost) and the coding delay (power consumption) (Remark: the trade-off is used to notice the balance between two or more values/items which are inverse with each other). The coding efficiency is measured by the entropy of the source. The entropy defines how easy is for a source to be compressed for a given randomness. For instance, a random noisy signal is very hard to be compressed. Sources with low entropy can be compressed easier. The coding complexity is also referred as the computational cost and it is related to the memory requirements or the number of arithmetic operations which are required in order to compress the input signal. The coding complexity most of the times increases the coding delay between the encoder and the decoder which is impractical for the power/energy constraints. For instance, the lower the coding complexity, the less the power consumption.

2.1.2 Lossy compression

The input signal can be represented with smaller number of bits by introducing some errors which cause some loss of information. The primary goal of lossy compression is to minimize the number of bits required to represent the input signal with the best possible quality. This compression, which is also called irreversible process, reduces the spatial and temporal redundancy of a signal. A lossy compression is used by applications or devices which do not require perfect reconstruction of the input signal. However, lossy compression seeks at first for subjective redundancy, selecting elements to be removed without significantly affecting the viewer's perception of visual quality. Lossy compression is more challenging because it sets a trade-off between 4 different dimensions: the *coding efficiency* (entropy), the *coding complexity* (computational cost), the *coding delay* (power consumption) and

the *reconstruction quality* (distortion). The distortion measures the difference in quality between the input f and the reconstructed signal $\tilde{f}$. Thus, the lower the distortion, the better the reconstruction.

Lossy compression algorithms have numerous of applications. Closed-circuit TeleVision (CCTV) systems, also known as video surveillance systems, is one of the applications among Hybrid Fiber Cable Networks (HFCs), Asymmetric Digital Subscriber Lines (ADSL), Digital Video/Versatile Discs (DVDs) and satellite TV, which ha been benefited by a high lossy compression performance.

2.2 Coding Principle

The coding principle is a common architecture which has been adapted by all the lossy compression algorithms and it describes the encoding and decoding process of an input signal (see Figure 2.2). This input signal f could be audio, image or video which is first transformed into a more compressible format. The Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), Discrete Sine Transform (DST), and Fourier are some of the transforms which have been used in compression algorithms. The result of this transform is identified by a quantizer in order to decide which information is redundant to be removed. Quantization is solely responsible to introduce distortion. As a result, in a lossless compression there is no quantization. The Entropy coding is a lossless function which translates the quantized intensity of the signal into codewords whose lengths vary inversely to the frequency of occurrence. Once the input signal has been coded, it is saved or sent through the communication channel to the receiver who needs to use this code in order to reconstruct the input signal. This is the decoding process which consists of the entropy decoding, the de-quantization and the inverse transform.

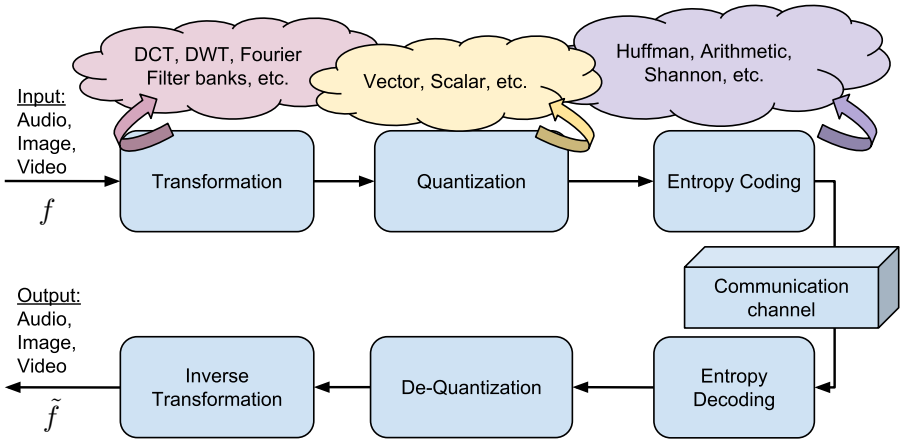


Figure 2.2: Coding Principle

As it is already mentioned, the goal of such an architecture is to find the best trade-off between the redundant information which is going to be eliminated during the quantization step while the reconstruction quality of the output signal $\tilde{f}$ needs to be as close as possible to the one of the input signal. Most of the times, people sacrifice the computational cost or the power consumption of the algorithm in order to achieve efficient entropy results and/or high reconstruction quality.

2.3 Rate-Distortion Theory

As explained above, in lossless compression, when the input source consists of real numbers it requires high number of bits (bitrate or rate) to be stored or transmitted. Generally, for

a computer this number of bits is approximately 64 bits/sample. However, no channel or storage unit enables an infinite rate. As a result, a finite representation of this source will never be perfect (lossy compression). The Rate-Distortion (RD) theory comes under the umbrella of source coding or compression and it is concerned with the task to find the best trade-off between the quality of the reconstruction (distortion D) and the loss of information (bitrate cost R). In other words, RD theory seeks for the fewer number of bits possible to achieve a given reproduction quality. The question is how to define the "goodness" of this representation? There are several metrics to measure the distortion between the input continuous signal and its representation. The issue of what kind of qualitative metric should be used to evaluate the quality for a source has been the objective of continuous study for many years now.

However, before we continue introducing the qualitative and the rate metrics we need first to define what a source is. A source could be one particular set of data (i.e. text file, audio signal, image, video, etc.). Alternatively, one could also consider a class of sources which are characterized by the same statistical properties. In such a case, the estimation of parameters which minimize the number of bits which corresponds to a given quality of reconstruction will be applied only for the class of sources of the same characteristics (i.e. a technique which works well for an audio signal may not be applied with the same success to video streams). On the other hand, parameters which are assigned to a class of sources will always result in variations among inputs. Thus, techniques which allow the " input-by-input" selection of parameters have been shown to be superior to those that work for a class of sources.

In early state-of-the-art subband image coding frameworks which were based on i.i.d. models for image subbands, the optimal bit allocation was necessary and very important to ensure that the bits were optimally distributed among the subbands in proportion to their importance. There have been proposed many models which optimize the bit allocation and are considered to be accurate for source classes of the same statistical distributions (i.e. Laplacian, Gaussian, Generalized Gaussians, etc.). This is going to be discussed in details in section 2.3.3 within the aspects of the optimization of the RD curve. Nevertheless, we need first to define some distortion and bitrate metrics. In section 2.3.1 we are going to represent the most commonly used qualitative metrics in compression. Apparently, the system which is going to in practice evaluate the quality of an input source is the human audio and/or visual system. Thus, the qualitative metrics are called to measure the audio/visual human perception. The most famous metrics among numerous of models, are the Mean Square Error (MSE), the Peak Signal to Noise Ratio (PSNR) and the Structure SIMilarities (SSIM). In section 2.3.2, we also describe the Shannon entropy as a necessary tool which allows to estimate the rate for a given distortion. Huffman, Stack-Run and Arithmetic entropy coders are also introduced in the same section as the lossless techniques which are used in practice to represent the quantized input signal as a binary stream.

2.3.1 Qualitative Metrics

There are several quality metrics which are responsible to measure the quality of the reconstructed signal with respect to the input one. It is desired to achieve the lower possible distortion D while a lot of the information has been discarded. The distortion of course should be assessed in an appropriate manner. Formally, it is defined as $D(f, \tilde{f})$, where f is the input signal of the coding principle which is described in Fig. 2.2 and $\tilde{f}$ is the output.

2.3.1.1 Mean Square Error (MSE)

The most commonly employed measure of distortion is MSE (Mean Squared Error), defined by:

$$\mathrm{MSE}(f, \tilde{f}) = \frac{1}{n}\sum_{i=1}^{n}(f_i - \tilde{f}_i)^2, \tag{2.1}$$

where n is the size of the input signal, $f = (f_1, \ldots, f_n)$ and $\tilde{f} = (\tilde{f}_1, \ldots, \tilde{f}_n)$. The distortion is minimized when the MSE approaches zero. It is worth noting that while it is typical to dismiss MSE as being poorly correlated to human perception, systems built on the above philosophy can be optimized for MSE performance with excellent results not only in MSE (as one would hope should be the case) but also in terms of perceptual quality.

2.3.1.2 Peak Signal to Noise Ratio (PSNR)

For image and video compression the MSE is most commonly quoted in terms of the equivalent reciprocal measure, PSNR (Peak Signal to Noise Ratio), defined by:

$$\mathrm{PSNR}(f, \tilde{f}) = 10\log_{10}\frac{(2^b - 1)^2}{\mathrm{MSE}(f, \tilde{f})}, \tag{2.2}$$

where b is the number of bpp (bits per pixel). The PSNR is expressed in dB (decibels) and it is based on the absolute error between the input and the output signals [Taubman and Marcellin, 2002]. The PSNR approaches infinity while MSE approaches zero, which means that a high PSNR value provides the high image quality. At the other end of the scale, a small value of the PSNR implies high numerical differences between images.

2.3.1.3 Structure SIMilarities (SSIM)

The SSIM used for measuring the similarity between two images is defined by:

$$\mathrm{SSIM}(f, \tilde{f}) = l(f, \tilde{f})c(f, \tilde{f})s(f, \tilde{f}), \tag{2.3}$$

where

$$l(f, \tilde{f}) = \frac{2\mu_f\mu_{\tilde{f}} + c_1}{\mu_f^2 + \mu_{\tilde{f}}^2 + c_1}, \tag{2.4}$$

$$c(f, \tilde{f}) = \frac{2\sigma_f\sigma_{\tilde{f}} + c_2}{\sigma_f^2 + \sigma_{\tilde{f}}^2 + c_2}, \tag{2.5}$$

$$s(f, \tilde{f}) = \frac{\sigma_{f,\tilde{f}} + c_3}{\sigma_f\sigma_{\tilde{f}} + c_3}, \tag{2.6}$$

where μ_f is the average of f, $\mu_{\tilde{f}}$ the average of $\tilde{f}$, σ_f^2 is the variance of f, $\sigma_{\tilde{f}}^2$ is the variance of $\tilde{f}$, $\sigma_{f,\tilde{f}}^2$ the covariance of f and $\tilde{f}$, $c_1 = k_1L^2$, $c_2 = k_2L^2$ and $c_3 = c_2/2$ are three positive variables to stabilize the division with weak denominator, L the dynamic range of the pixel-values (typically $2^{\mathrm{bits/pixel}} - 1$) and $k_1 = 0.01$, $k_2 = 0.03$ by default. The range of the SSIM output is $0 \leq \mathrm{SSIM} \leq 1$. This metric is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms [Wang et al., 2004].

2.3.1.4 PSNR vs SSIM

There are no precise rules of how to select a quality metric when the evaluation of an image quality is required. The numerical values which are obtained during evaluation have been interpreted in different ways. Some studies have shown that MSE and consequently PSNR perform badly comparing to SSIM due to a various type of degradations which can be assigned to the same value of MSE [Teo and Heeger, 1994, der Weken et al., 2002, Eskicioglu and Fisher, 1995]. On the contrary, there are studies which support that MSE and PSNR perform better in assessing the quality of noisy images. Figure 2.3 shows the performance of PSNR and SSIM metrics when the well-known image "lena" is compressed with JPEG standard for different bitrates. The authors in [Horé and Ziou, 2010] proposed

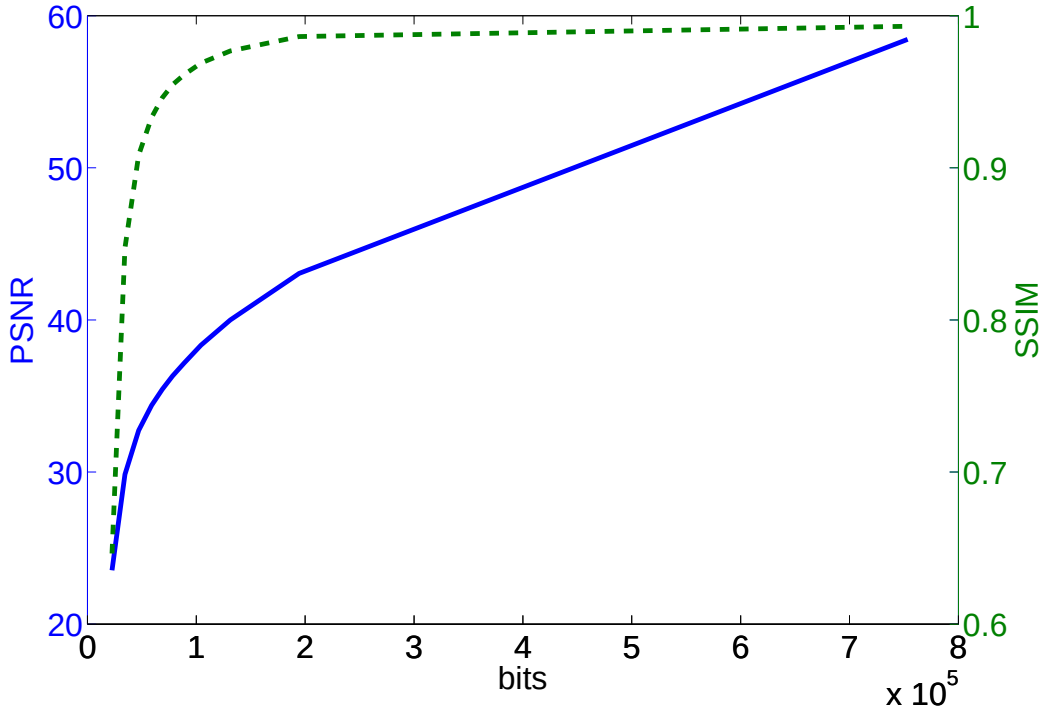


Figure 2.3: The figure illustrates the PSNR and the SSIM values which correspond to the same bitrates when lena image is compressed with JPEG standard obtaining different bitrates.

a relation between PSNR and SSIM which is described as:

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{2\sigma_{f,\tilde{f}}(l(f,\tilde{f}) - \mathrm{SSIM})}{(2^b-1)^2\mathrm{SSIM}} + \left(\frac{\mu_f - \mu_{\tilde{f}}}{2^n-1}\right)^2\right], \tag{2.7}$$

where

$$l(f,\tilde{f}) = \frac{2\mu_f\mu_{\tilde{f}} + c_1}{(\mu_f^2 + \mu_{\tilde{f}}^2 + c_1)}. \tag{2.8}$$

This relation concludes that the values of PSNR is possible to be predicted by the value of SSIM and vise-versa. Actually, it suggests that the values of the SSIM and those of the PSNR are not independent (see Fig. 2.4). There is a general relation which can be used for any kind of image degradation. In addition, if we use $l(f,\tilde{f}) = 1$ and $\mu_f = \mu_{\tilde{f}}$, equation (2.7) can be simplified and rewritten as:

$$\mathrm{PSNR} = -\left(10\log_{10}\left[\frac{(2^b-1)^2}{2\sigma_{f,\tilde{f}}}\right] + 10\log_{10}\left[\frac{\mathrm{SSIM}}{1-\mathrm{SSIM}}\right]\right) \tag{2.9}$$

However, estimating a relation between the two qualitative metrics does not indicate which one is more accurate and efficient to better evaluate the quality of an image. An easy way to compare the two metrics is proposed in [Horé and Ziou, 2010]. This method uses F-scores for different degradation parameters related to the JPEG and JPEG2000 compression qualities, Gaussian blur, Additive Gaussian Noise, etc. (see Fig. 2.5). Let suppose, we are interested in testing the quality of the JPEG compression for 4 different parameters: 30%, 50%, 70% 90%. For each parameter, we need to test a group of images in order to compute a group of PSNR and SSIM values (one PSNR (SSIM) value per image). The F-score associated to the PSNR corresponds to the ratio of the variance of the mean values which has been computed for each group of PSNR for each parameter, over the mean value of the within-group variances. In exactly the same way are computed the F-scores of the SSIM. The F-score varies in $[0, \infty[$ where low values indicate that the parameters have low impact on the values of the quality measure while high F-score values stand for high impact. Figure 2.5 shows that the highest sensitivity for both the metrics is given for the Additive Gaussian Noise. The PSNR performs better than SSIM in discriminating the Gaussian blur and it is also considered to be by far the most efficient metric for Additive Gaussian Noise. On the contrary, SSIM seems to be the more sensitive than PSNR for JPEG and JPEG2000 compression quality.

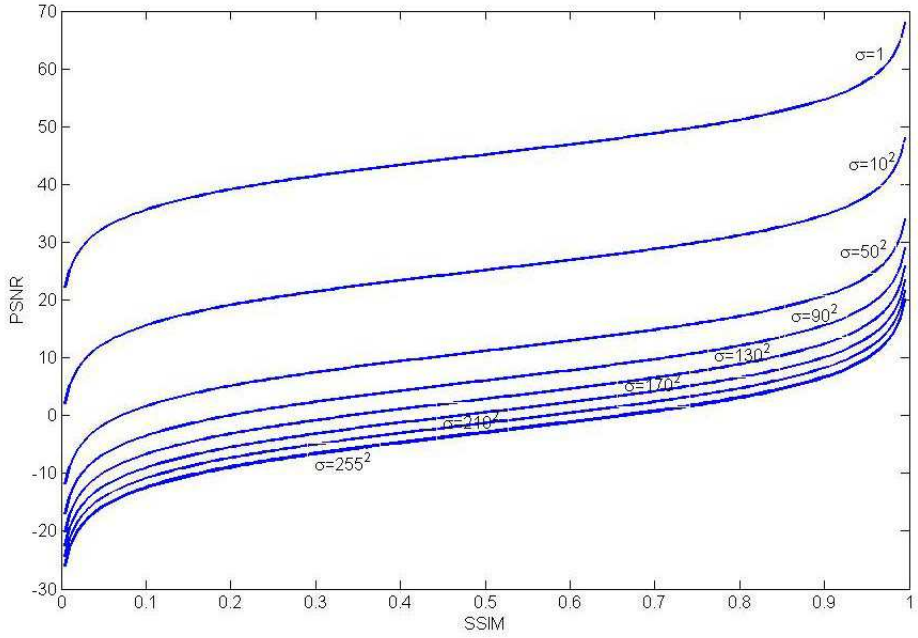


Figure 2.4: Variation of the PSNR as function of the SSIM for different fixed values of $\sigma_{f,\tilde{f}}$ (extracted from [Horé and Ziou, 2010]).

2.3.2 Shannon Entropy

A quantity known as "entropy" is defined in terms of the statistical properties of the information source. In other words, the entropy is a measure of randomness. The entropy represents a lower bound on the average number of bits required to represent the source output without loss of information. Given an input source S there are random symbols $s_1, s_2, \ldots, s_n$. Each one of these symbols i has a probability to be occurred, p_i. According to Shannon, the entropy of a source S is given by :

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i. \tag{2.10}$$

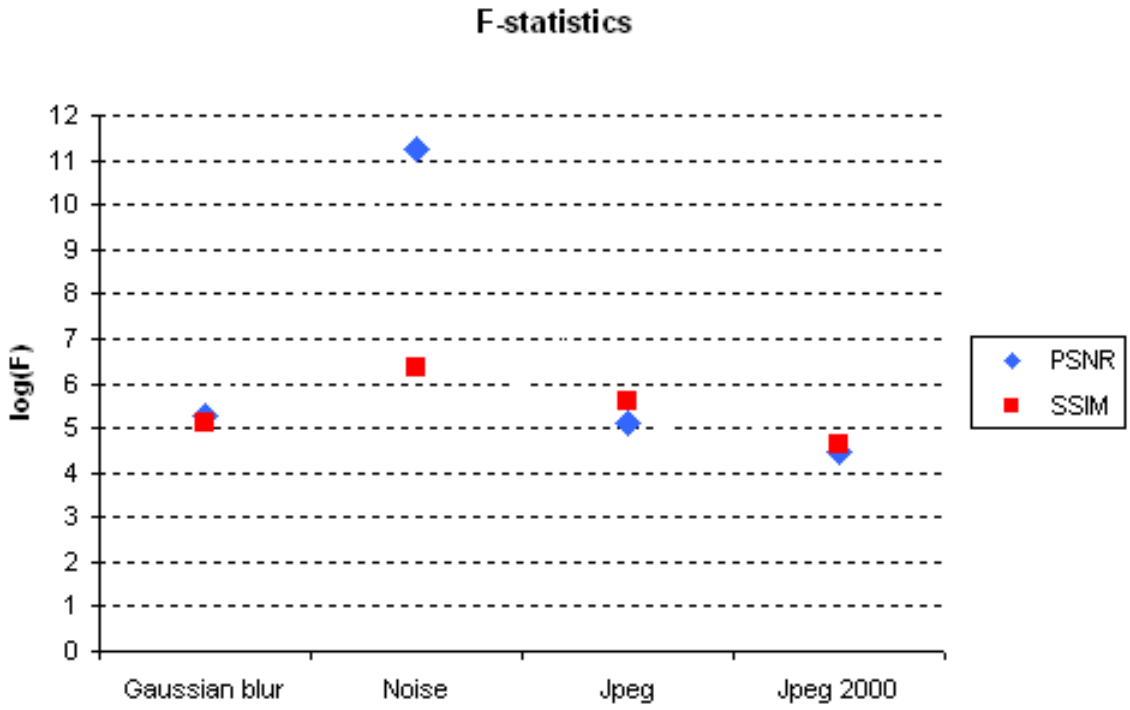


Figure 2.5: Comparison of the sensitivity of the PSNR and the SSIM using the F-scores (extracted from [Horé and Ziou, 2010]).

The Shannon entropy measures information in unit of bits/sample. If one changes the base of the logarithm, then the unit of the entropy changes i.e. nat is for $\log_e$, where e is the Euler's number and hartley for $\log_{10}$. The Shannon entropy coding gives us the opportunity to map each symbol s_i into a codeword c_i. To define the length l_i of this codeword there are numerous techniques like Huffman Coding [Huffman, 1952, Bhaskaran and Konstantinides, 1997, Mishra and Singh, 2015], Arithmetic coding [Rissanen and Langdon, 1979, Witten et al., 1987, Bhaskaran and Konstantinides, 1997, Howard and Vitter, 1992], Stack-Run Coding [Tsai et al., 1996, Tsai, 1998], LZW coding [Mishra and Singh, 2015], Run-length Coding [Abdelgattah and Mohiuddin, 2010], DPCM coding [Tomar and Jain, 2016], etc. All these techniques are used to perform lossless compression. In this document, we are going to introduce only Huffman and Arithmetic coding because these two methods have been widely used in image and video compression algorithms.

2.3.2.1 Huffman Coding

Huffman Coding [Huffman, 1952] was used in JPEG and H.261 while 2D and 3D Huffman coding were also part of MPEG and H.263. The benefit of Huffman coding is that it takes advantage of the disparity between the frequencies. It uses less storage for the frequently occurring characters at the expense of having to use more storage for each of the more rare characters [Bhaskaran and Konstantinides, 1997, Mishra and Singh, 2015]. The coding processing evolves along the following steps:

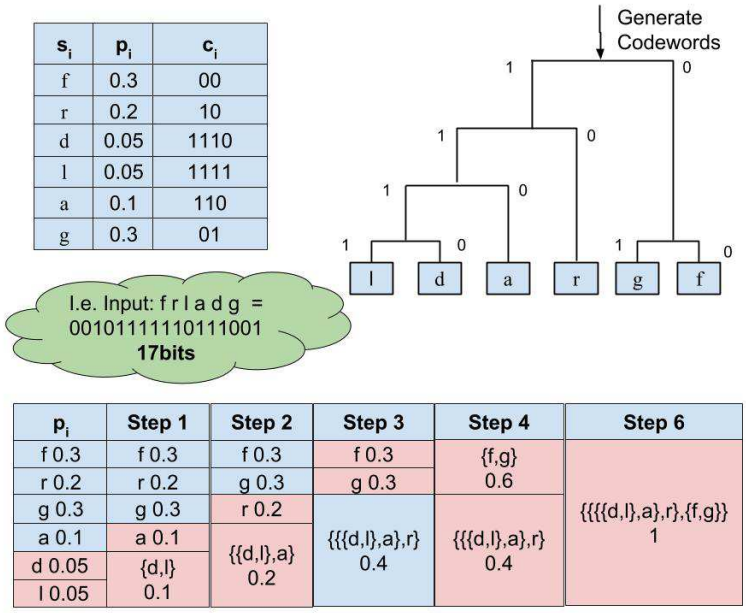
1. Order the symbols according to their probabilities. The frequency of occurrence of each symbol must be known as a prior in order to build the Huffman code. In practice, the frequency of occurrence can be estimated from a training set of data that is representative of the data to be compressed in a lossless manner. For instance, if an alphabet is composed of n distinct symbols $s_1, s_2, \ldots, s_n$ and the probabilities

of occurrence are $p_1, p_2, \ldots, p_n$, then the symbols are rearranged so that $p_1 > p_2 > \ldots > p_n$.

2. Apply a contraction process to the two symbols with the smallest probabilities. Suppose the two symbols are s_{n-1} and s_n. We replace these two symbols by a hypothetical symbol. say, H_{n-1} , that has a probability of occurrence $p_{n-1} + p_n$. Thus the new set of symbols has $n - 1$ members: $s_1, s_2, \ldots, s_{n-2}, H_{n-1}$.

3. We repeat the previous step until the final set has only one member.

The second step is responsible to build a binary tree, since at each step two symbols are merged. At the end of this process all the symbols will stand as the leaf nodes of the tree. The codeword for each symbol s_i is obtained by traversing the binary tree from its root to the leaf node corresponding to s_i. An interesting example of Huffman coding is given in Fig. 2.6.



s_i	p_i	c_i
f	0.3	00
r	0.2	10
d	0.05	1110
l	0.05	1111
a	0.1	110
g	0.3	01

p_i	Step 1	Step 2	Step 3	Step 4	Step 6
f 0.3	f 0.3	f 0.3	f 0.3	{f,g} 0.6	{{{{d,l},a},r},{f,g}} 1
r 0.2	r 0.2	g 0.3	g 0.3		
g 0.3	g 0.3	r 0.2	{{{d,l},a},r} 0.4	{{{d,l},a},r} 0.4	
a 0.1	a 0.1	{{d,l},a} 0.2			
d 0.05	{d,l} 0.1				
l 0.05					

Figure 2.6: Huffman coding. This figure show the step-by-step Huffman coding approach and the resulting coding tree for an input sequence "frladg", given the probabilities of each symbol. Huffman encoding requires 17bits for this sequence.

The Huffman code is perfectly decodable using a Look-Up-Table (LUT). The LUT is constructed at the decoder from the symbol-to-codeword mapping table. So, if the longest codeword is of a length L, there are 2^L entries to the LUT. In order to decode symbol s_i using only its codeword c_i of a length l_i, we retrieve the value of the 2^{L-l_i} address.

The basic drawbacks of Huffman coding are first of all, its sensitivity to changes in signal statistics. If the probability of occurrence of the input symbols changes, one should redesign the Huffman code from the beginning. The adaptive Huffman coding [Vitter, 1987] which is an extension of Huffman coding deals with non-stationary signals. A second disadvantage of Huffman coding is that it finds a codeword for each symbol. However, it has been proven that the most efficient coding which allow high compression ratios, can be achieved if many symbols are assigned to the same unit.

2.3.2.2 Arithmetic Coding

Arithmetic coding is an alternative to Huffman coding. It has been used both in image and video coding standards like JBIG, JPEG, JPEG2000 and H.263, H.264/MPEG-4/AVC,

HEVC respectively. The advantage of this method is that multiple symbols are treated as single data unit but at the same time it retains the symbol-by-symbol coding approach like Huffman coding [Bhaskaran and Konstantinides, 1997, Howard and Vitter, 1992]. As a result, Arithmetic coding is adaptable to the frequency of occurrence which is unrelated to the design of the coder. The coding process evolves along the following steps:

1. There is the current interval $[Current_{low}, Current_{high})$ which is initialized to [0,1). The current interval is subdivided into several half-open subintervals each one of which corresponds to a symbol s_i of the input source S. Each subinterval is considered to be a codeword c_i. The upper limit of each subinterval is the cumulative probability up to and including the corresponding symbol. The lower limit is the cumulative probability up to but not including the symbol.

2. When the first symbols of the s_i appears, its correspondent subinterval is selected to become the current interval. For instance, let $[b^i_{low}, b^i_{high})$ the correspondent subinterval of the symbol s_i with lower bound b^i_{low} and upper bound b^i_{high}. This subinterval should replace [0,1) which is the initial current one. Let $Previous_{low}$ and $Previous_{high}$ be the lower and the upper limits of the old interval (in this case 0 and 1 respectively) and $Range = Previous_{high} - Previous_{low}$. One needs to compute the bounds of the new current interval according to the following rules: $Current_{low} = Previous_{low} + Range \times b^i_{low}$ and $Current_{high} = Previous_{low} + Range \times b^i_{high}$.

3. Step 2 is repeated each time a new symbol appears. The current interval becomes the previous one and using the rules described above we compute the current interval with respect to the correspondent subinterval of the new symbol.

4. There is no need to transmit both the lower and the upper bound of the last new interval. Usually, a value of a fractional number which is within the final range is an efficient output.

Arithmetic coding yields better compression because it encodes a message as a whole new symbol instead of as separate symbols. One, should use enough bits to distinguish the final current interval from all the other possible final intervals. An example of arithmetic coding is given in Fig. 2.7.

The decoding process of the Arithmetic coding is based on a unique number j and the cumulative probabilities $cumprob_j$ which are both assigned to each symbol of the sequence. Starting from the output value and the initial interval [0,1), one should search for which j the following inequality is true: $cumprob_j \leq \dfrac{value - Previous_{low}}{Range} < cumprob_{j-1}$. Then, the values are updated and the decoding process continuous until all the values will be perfectly reconstructed: $Previous_{high} = Previous_{low} + Range \times cumprob_{j-1}$ and $Previous_{low} = Previous_{low} + Range \times cumprob_j$.

2.3.2.3 Stack-Run Coding

The stack-run coder is an algorithm which is applied to a signal after it has been first transformed and quantized to represent every meaningful coefficient which is necessary for the reconstruction of the signal [Tsai et al., 1996]. As meaningful coefficients are considered the non-zero positive or negative values which are called"stack" or significant coefficients, while the zeros between them are meaningless coefficients which are called "run". An adaptive arithmetic coding is then used to compress the sequence in higher efficiency.

Opposed to other similar techniques which have been used in compression, like the zerotree [Shapiro, 1993] or the run-length coder which take advantage of the relationship within the subbands, the Stack-Run coder is applied to each subband independently. This is one of the great advantages of the algorithms which gains in simplicity. Another benefit

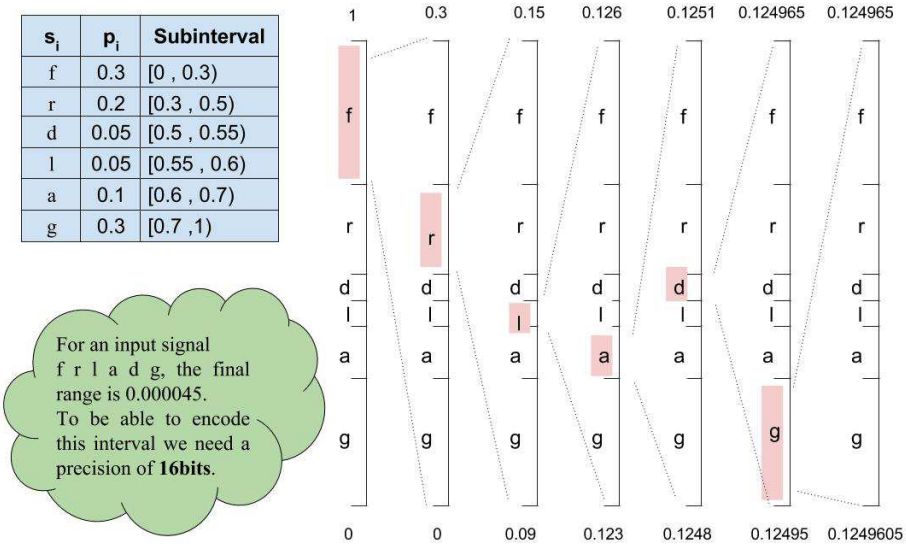


Figure 2.7: Arithmetic coding. This figure illustrates the sub-intervals and the encoding process for the same input sequence we used in Fig. 2.6. Arithmetic coding needs 16 bits to encode this signal which means that it is more efficient than Huffman coding (17 bits).

of this algorithm is that it uses only 4 different symbols ("+", "-", "0" and "1") to represent all the values in a subband allowing the use of the arithmetic coding . Let each significant coefficient be represented by a binary stack starting from the Least Significant Bit (LSB) to the Most Significant Bit (MSB). The MSB of this stack is always the sign of the coefficient which is represented right after the binary stream with a "+" if the value is positive and "-" if the value is negative. Another issue of the Stack-Run coder is that no binary stream encodes the zero value, as in ASCII table. For example, the value +4 is going to be represented by 10+ instead of 00+. A complete description of a subband can be provided by a group of pairs (a, b), where a defines the length of zero-runs before a significant coefficient value b arrives.

Figure 2.8 shows an example of a Stack-Run coder when it is applied to a small subband of a size 4×4 (in green). The values of this subband are the input of the algorithm (in blue). The symbols which are used to represent each significant coefficient are introduced in the red table followed by an example of how the stack of few values is formed from LSB to MSB. The gray table stands for the symbols which are used to encode the zero-runs followed by an example of how different lengths of zero-runs will be encoded.

An extension of the Stack-Run coder which is called Stack-Run-End coder introduces two more symbols the "EOB" for the zero-runs which lead to the end of a subbands and the "EOI" for the zero-runs in a sequence of subbands which lead to the end of the image decomposition [Tsai, 1998]. We introduce also the encoded chain of the input signal (in yellow) with both Stack-Run and Stack-Run-End coders.

2.3.3 Rate-Distortion Optimality

We are now familiar with the most famous metrics which are used to evaluate the difference between the quality of the input and the reconstructed signal, and the number of bits which are required to store and/or transmit this quality. For a given source, if we consider all the possible quantization choices, we are able to define a point cloud of all the possible RD couples. The RD optimization requires to minimize the distortion D under the constraint of $R < R_{\max}$, where $R_{\max}$ is a maximum bitrate bound. The solution of the RD optimization defines the *operational rate-distortion curve* (see Fig. 2.9). This curve is

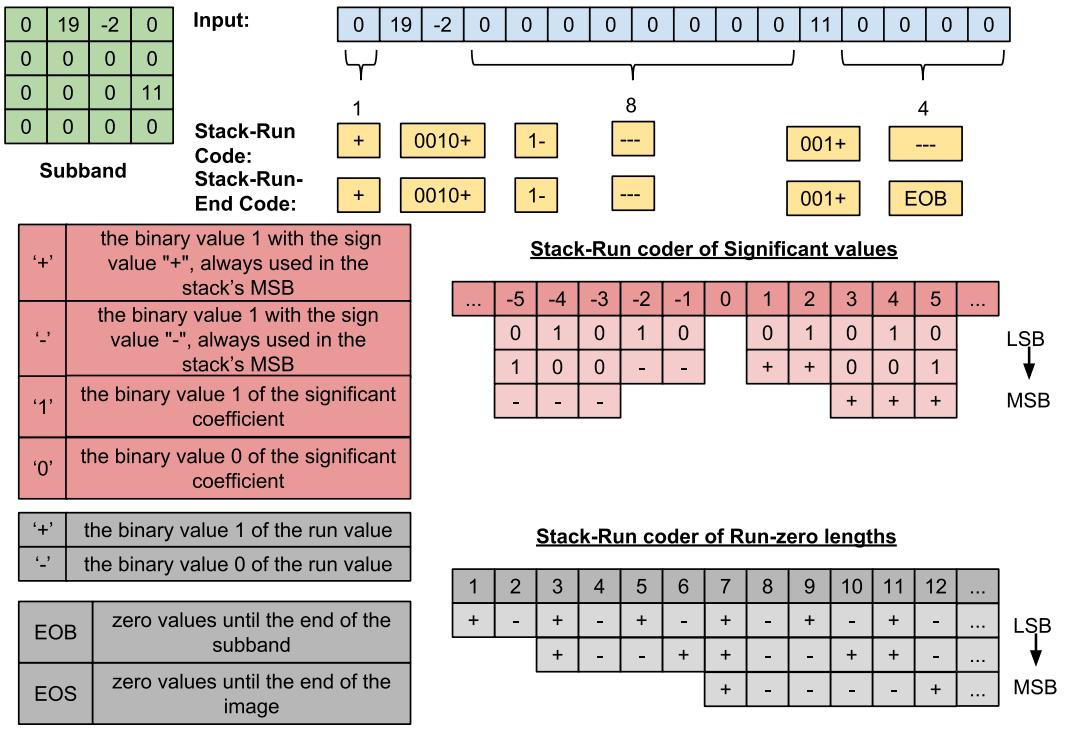


Figure 2.8: Stack-Run Coder. This figure represents output of the Stack-Run and the Stack-Run-End coders (yellow) which are applied to an input vector (blue). There are two dictionaries, one related to the symbols (red) and another one for the zero values (gray).

obtained by selecting the best rate-distortion pairs, within all the possible rate points and their corresponding distortions. Each point which lies on the operational curve is achievable depending on the system and the test data. However, for each system there should be some bounds to distinguish the best points to those ones whose performance is considered to be sub-optimal or unachievable.

2.3.4 Coding Unit and Complexity

The optimal trade-off between the rate and the distortion is computed for different *coding units*, which could be a sample, an image block, or a full image which are encoded given a distortion value for each different selected rate. The selection of the size of the coding units is directly linked to the complexity of the system. Undoubtedly, if the size of the coding unit is very small (i.e. an 8×8 block of a picture of a high definition video stream) and the number of operations (i.e. different quantization values) for each unit is large, the complexity implications will be dramatic for the whole system. One should keep in mind that the complexity also depends on the delay in computing the optimal solution. Especially in online encoding systems this delay should be the minimum one, thus the complexity should be also low. On the other hand, off-line encoding applications maybe supported by more complex algorithms. In order to reduce the computational cost and achieve the best trade-off between the rate and the distortion, instead of trying every possible value, one may use an optimization algorithm.

2.3.5 Lagrangian Optimization

A very well known method which has been used to seek for the operational rate-distortion curve is the Lagrangian optimization algorithm [Everett, 1963]. The optimization algorithm of the RD curve for biorthogonal sources could be described by a cost function J depending

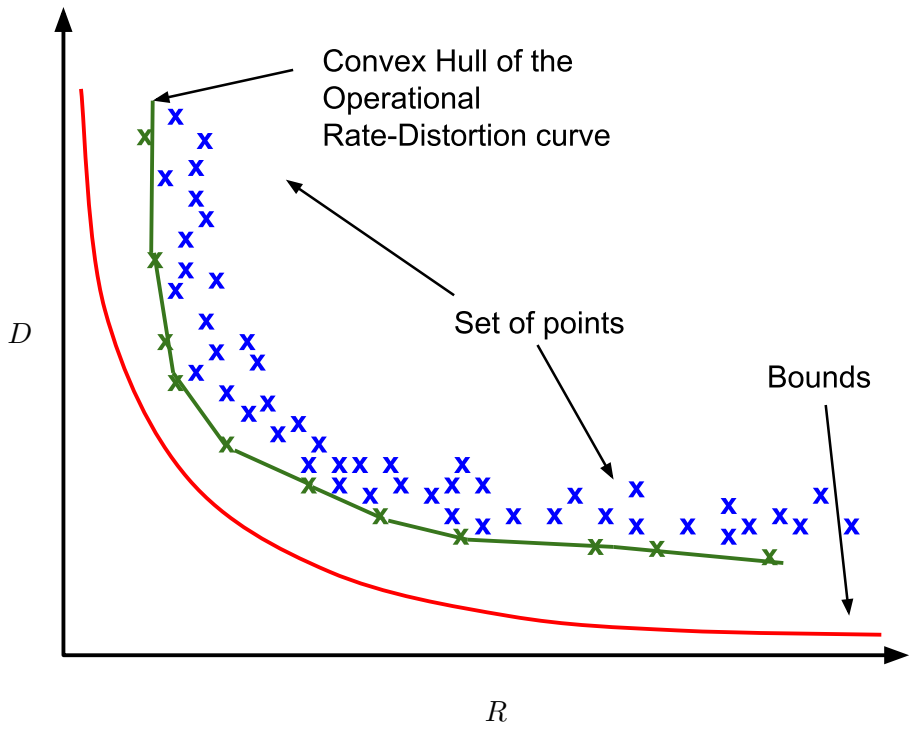


Figure 2.9: Operational RD curve. The cloud of points (in blue) correspond to the some possible quantization choices which result in a pair of a rate and distortion. Some of these points define the convex hull of this cloud which stands for the optimal rate-distortion pairs. The red curve shows the achievable performance of the system for a given input of known distribution.

on the distortion D and the rate R which needs to be optimized:

$$J_\mu = D + \mu R, \tag{2.11}$$

where $\mu \in \mathbb{R}^+$ is the Lagrange multiplier. When $\mu \approx 0$ there is more emphasis to the minimization of the distortion D enabling higher bitrate. On the other hand, if μ is large tends to minimize R and increase the quality of the reconstruction. The estimation of the Lagrange parameter is a highly complex problem [Ortega and Ramchandran, 1998]. Fortunately, there have been proposed empirical approximations to effectively choose μ in a practical mode selection scenario [Tseng et al., 2006]. If the source is a set of n decomposition layers after a given transform and if each subband i is orthogonal, the global distortion D is the sum of subband distortions D_i [Gersho and Gray, 1992] which is a function of rate:

$$D = \sum_{i=1}^{n} D_i = \sum_{i=1}^{n} D_i(R_i). \tag{2.12}$$

When the filter is not biorthogonal then there should also be considered suitable weights which account for non-orthogonality (see eq. 2.13) [Usevitch, 1996]. The goal of the Lagrangian optimization algorithm is to minimize J with respect to the distortion D and the rate R under given constraints for both of these magnitudes. Although, sometimes it is easier to describe two magnitudes at the same time the dependency effects are often ignored

to speed up the computation. The independent allocation strategies are introduced in the following sections.

2.3.5.1 Rate Allocation Problem

Considering the distortion as a function of rate, the cost function J can be described as a function of rate $J(R)$. The constraint of such a case is imposed to the total subband bitrate which should not be larger than a given bound $R_{\max}$:

$$R = \sum_{i=1}^{n} a_i R_i \leq R_{\max}, \tag{2.13}$$

where the coefficient a_i depends on the size of the subbands. To integrate the cost function we also need to define the distortion as a function of rate (see 2.12). In [Usevitch, 1996], the authors developed an expression of distortion for different wavelet transforms, showing that there is a weight w_i linked to each subband. This approach was proposed for Daubechies 9/7 or 5/3 filters.

$$D(R) = \sum_{i=1}^{n} w_i D_i(R_i). \tag{2.14}$$

Now, we have all the tools to define the cost function in terms of rate allocation:

$$J(R) \quad = \quad \sum_{i=1}^{n} w_i D_i(R_i) - \mu \left(\sum_{i=1}^{n} a_i R_i - R_{\max} \right), \quad \text{with} \quad \mu \leq 0. \tag{2.15}$$

The Lagrange multiplier could be seen as the slop of the RD curve which is defined according to equation (2.20). The optimal rate allocation corresponds to the points having the same slop on the "weighted" curve.

$$\frac{w_i \partial D_i}{a_i \partial R_i} = \mu, \quad \forall i = \{1, \ldots, n\}. \tag{2.16}$$

The solution is found by an iterative algorithm. Let ε be a suitable tolerance, and j represent the number of attempts. It is sufficient to find the first value μ^j such that:

$$R_{\max} - \varepsilon \leq \sum_{i=1} a_i R_i(\mu^j) \leq R_{\max}. \tag{2.17}$$

2.3.5.2 Distortion Allocation Problem

The previous section introduced the minimization of the cost function with constraint imposed on rate. However, a different problem to be solved is the minimization of the cost function given a distortion constraint:

$$D(R) = \sum_{i=1}^{n} w_i D_i(R_i) \leq D_{\max} \tag{2.18}$$

The above constraint changes the Lagrangian cost function as following:

$$J(R) = \sum_{i=1}^{n} a_i R_i - \mu \left(\sum_{i=1}^{n} w_i D_i R_i - D_{\max} \right), \quad \text{with} \quad \mu \leq 0. \tag{2.19}$$

The zero-gradient condition is given by:

$$\frac{w_i \partial D_i}{a_i \partial R_i} = \frac{1}{\mu}, \quad \forall i = \{1, \ldots, n\}. \tag{2.20}$$

The algorithm proposed to find the best allocation vector is then quite similar to the previous one. Indeed, it is sufficient to change the termination condition as following:

$$D_{\max} - \varepsilon \leq \sum_{i=1}^{n} w_i D_i R_i(\mu^j) \leq D_{\max}. \tag{2.21}$$

2.4 Progress in Video Compression Standards

This section aims to be the overview of video compression standards. First of all, we define what is a video stream. Then, we discuss JPEG and JPEG2000 which are the most effective image compression standards. These two standards became the basic inspiration of developers who wanted to build video compression standards. Lastly, we introduce the progress of the compression techniques as they introduced in each video compression standard.

2.4.1 Video Stream

A natural visual scene $I(\boldsymbol{X}, t)$, where $\boldsymbol{X} \in \mathbb{R}^3$ and $t \in \mathbb{R}^+$, is a 3D stimulus which is spatially and temporally continuous. According to the law of optics, the 3D visual stimuli $I(\boldsymbol{X}, t)$ is projected onto the retina, which is the innest tissue of the eyes, via the lens (the optics of the eye is detailed in [Ögmen and Herzog, 2010]). Hence, the 3D luminance $I(\boldsymbol{X}, t)$ is simplified into a 2D luminance $\tilde{I}(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \mathbb{R}^2$ and $t \in \mathbb{R}^+$. However, even this 2D analog signal needs to be spatially and temporally sampled in a digital format. Digital videos are the representation of a sampled visual scene in digital form. A digital video stream $f(\boldsymbol{x}, t)$ which has been temporally sampled, is a group of pictures which change with respect to time as following:

$$f(\boldsymbol{x}, t) = \sum_{i=1}^{N} f_i(\boldsymbol{x}) \mathbf{1}_{[g_i, g_{i+1}]}(t), \tag{2.22}$$

where $\boldsymbol{x} \in \mathbb{R}^2$, $t \in \mathbb{R}$ is the observation time, $f_i(\boldsymbol{x})$ stands for the i^{th} picture of the video, N is the total number of pictures which form the video stream and $\mathbf{1}_{[g_i, g_{i+1}]}(t)$ is the indicator function which is equal to 1 if $g_i \leq t \leq g_{i+1}$, and 0 otherwise. Let's call frame period $T_i = g_{i+1} - g_i$ the duration for which a given picture $f_i(\boldsymbol{x})$ of the video stream exists. For simplicity it is assumed that $T_i = T$ because T_i is the same for every single picture of a video stream with a frame rate $1/T$.

Let $x_1, \ldots, x_n \in \mathbb{R}^2$ be some spatial samples of the i^{th} picture of the video stream and $f_i = (f_i(x_1), \ldots, f_i(x_n))$ the spatially sampled i^{th} picture. Each spatiotemporal sample of the video stream describes the brightness or the luminence and the color of the sample. The number n of spatial samples influences the quality of each picture (image or frame). The more the spatial samples, the higher the resolution of each picture. A monochrome picture requires only one number which represents the brightness. A color RGB picture requires three values per pixel for the Red, Green and Blue colors of light. The combination of red, green and blue in varying proportions generates any possible color:

$$Y_i(\boldsymbol{x}_k) = k_r R_i(\boldsymbol{x}_k) + k_g G_i(\boldsymbol{x}_k) + k_b B_i(\boldsymbol{x}_k), \tag{2.23}$$

where $Y_i(\boldsymbol{x}, t)$ is the grayscale intensity of the i^{th} spatiotemporal sample of the input visual stimulus, $R_i(\boldsymbol{x}_k)$ is the red color sample, $G_i(\boldsymbol{x}_k)$ the green color samples and $B_i(\boldsymbol{x}_k)$ the blue color sample, k_r,k_g and k_b are weighting factors which correspond to red, green and blue colors respectively. The $Y_i(\boldsymbol{x}_k) : C_i^b(\boldsymbol{x}_k) : C_i^r(\boldsymbol{x}_k)$ color space is a popular way of efficiently representing color images where $C_i^r(\boldsymbol{x}_k) = R_i(\boldsymbol{x}_k) - Y_i(\boldsymbol{x}_k)$, $C_i^g(\boldsymbol{x}_k) = G_i(\boldsymbol{x}_k) - Y_i(\boldsymbol{x}_k)$ and $C_i^b(\boldsymbol{x}_k) = B_i(\boldsymbol{x}_k) - Y_i(\boldsymbol{x}_k)$ are the *color differences* (chrominance or chroma). Only two of the three chrominance components need to be stored or transmitted since the third component can always be calculated from the other two.

The number of temporal samples influences the motion. A temporal sampling is called *progressive*, when the result is a series of complete frames. In case of incomplete frames the temporal sampling is called *interlaced* (the format and the use of an interlaced video is described in details in section 2.4.5.) The temporal samples correspond to the frame rate (the number of picture per second). The higher the frame rate, the smoothier the motion.

Format	Resolution	Nbr of Pixels
Standard Definition (SD)	720×576	141720
720p High Definition (HD)	1280×720	921600
1080p HD	1920×1080	2073600
UHD TV	3840×2160	8294400
2160p 4K UHD	4096×2160	8847360
8K UHD	7680×4320	33177600

Table 2.1: Video formats and their resolutions.

On the other hand, that requires more samples to be stored. In general a video stream of 10-20 pps (pictures per second) is considered to be of a low frame rate, 25-30 pps is the standard frame rate and 50-60 pps is a high frame rate video. Table 2.1 shows different video formats and their resolutions.

2.4.2 From JPEG to HEVC

Tracking the development of video compression algorithms, one will come to the following conclusion: all the video compression algorithms have the same origin which is the JPEG standard [ISO/IEC 10918-1:1994, 1994, Bhaskaran and Konstantinides, 1997]. The "JPEG" is an acronym of the Joint Photographic Expert Group which in 1986 established a standard for the sequential progressive encoding of continuous tone grayscale and color images. The "Joint" stands for International Organization for Standardization (ISO) and the International Telegraph and Telephone Consultative Committee (CCITT) which is a permanent organ of the International Telecommunication Union (ITU), the United Nations Specialized Agency in the field of telecommunications. JPEG2000 is a more recent standard released by the Joint Photographic Expert Group and it was intended as a successor of JPEG standard in many of its applications. However, even though JPEG2000 is more efficient with respect to compression ratio than JPEG, it was rarely preferred to be used due to its complexity. Back in 2000, when JPEG2000 was released, its format required a lot of memory to be process which was problematic when an average computer included around 64 MB of memory. In addition, JPEG2000 was an entirely different format based on new code, which means that the format was not backward compatible [Christopoulos et al., 2000, Santa-Cruz et al., 2002].

In the late 1980s, the Motion Picture Experts Group (MPEG) was formed with the purpose of deriving a standard for the coding of moving pictures and audio. It has since produced the standards for MPEG 1, MPEG-2, and MPEG-4. At the same time, the Video Coding Experts Group (VCEG) which is the sub group of ITU developed for example the H.261 and H.263 recommendations for video-conferencing over telephone lines. At the end of the 1990s, a new group was formed, the Joint Video Team (JVT), which consisted of both VCEG and MPEG. The purpose was to define a standard for the next generation video coding. The JVT released a series of standards like H.622/MPEG-2, H.264/MPEG-4/AVC which is termed Advanced Video Coding (AVC) and H.265 which is called High Efficiency Video Coding (HEVC). Figure 2.10 shows the progress in image and video compression standards over the last decades. The designers used JPEG standard as a basis to encode and decode key-pictures of a video stream introducing at the same time other methods to reduce temporal and spatial redundancy (i.e. inter-picture prediction, motion estimation, macroblocks, interlaced videos, deblocking, Discrete Cosine Transform (DCT) [Ahmed et al., 1974, Britanak, 2001], Discrete Wavelet Transform (DWT) [Mallat, 1999], quantization, entropy coding, etc). The second conclusion which is naturally raised is related to the complexity of these algorithms which increased during the years [Grois et al., 2013]. To achieve more efficient compression algorithms and improve the bitrate of the video codecs, the designers proposed more and more complex solutions.

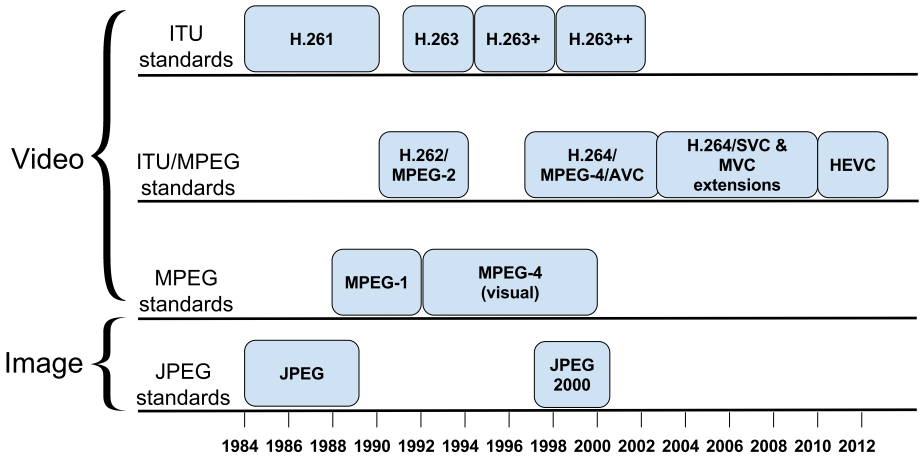


Figure 2.10: Standardization History.

2.4.3 Overview of JPEG and JPEG2000

Image compression algorithms have been used in numerous of applications like internet, digital photography, medical imaging, remote sensing, surveillance, facsimile, etc. The general structure of an image compression standard follows the coding principle which is illustrate in Figure 2.2. As it has been mentioned in section 2.2, there are three major steps which in case of JPEG are the DCT transform, the quantization and the Huffman entropy coding [Bhaskaran and Konstantinides, 1997]. The successor of JPEG, JPEG2000, follows the same principle but it uses DWT and Arithmetic entropy coding instead [Christopoulos et al., 2000, Santa-Cruz et al., 2002].

2.4.3.1 Discrete Cosine Transform (DCT)

The DCT is a basis in image and video compression standards. The basic computation is the transformation of an input block of a size $N \times N$, where $N = 64$, from the spatial to the DCT domain:

$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}, \tag{2.24}$$

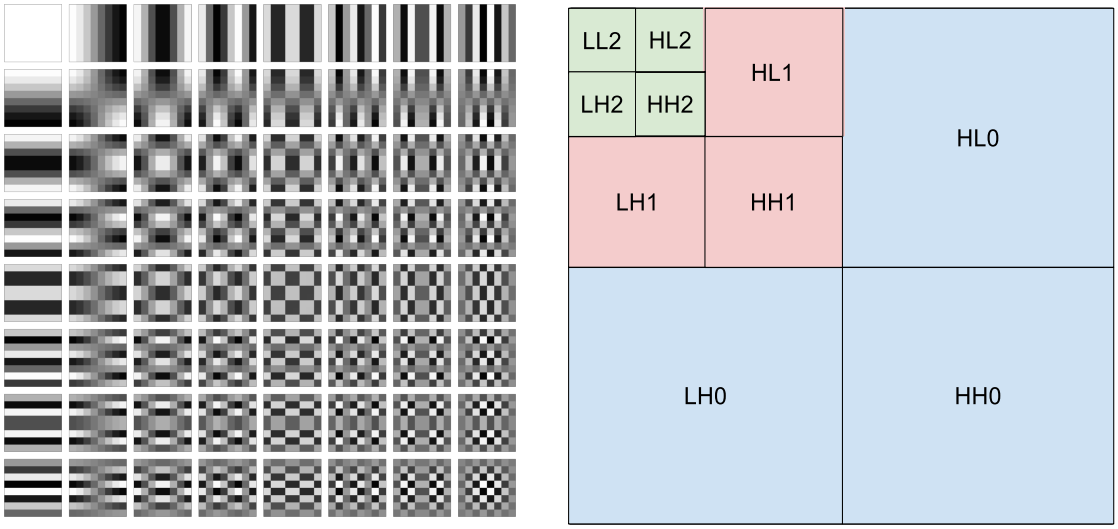
where $f(x,y)$ is the input image and

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if} \quad u = 0 \\ 1, & \text{otherwise} \end{cases} \quad \text{and} \quad C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if} \quad v = 0 \\ 1 & \text{otherwise.} \end{cases} \tag{2.25}$$

The blocksize had been initially chosen to be 8×8 for several reasons: first of all, from the computational point of view such a small size of block is not memory demanding. Secondly, if one increases the size of the block the efficiency of the algorithms is almost unchanged. Last but not least, the spatial correlation maybe eliminated in case of larger block. The block-based DCT decomposition is illustrated in Fig. 2.11 (a). The benefit of the DCT transform is its orthogonality, which means that it is invertible and leads to a perfect reconstruction of the input signal (see eq. 2.26). In addition, it has been proven that DCT decorrelates as well as Karhunen-Loève transform, sources with correlate coefficients

[Hamidi and Pearl, 1976].

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}. \tag{2.26}$$



(a) (b)

Figure 2.11: (a). Two dimensional DCT block-based decomposition frequencies. (b) Dyadic wavelet decomposition.

2.4.3.2 Discrete Wavelet Transform (DWT)

Among several wavelet transforms [Mallat, 1999, Fowler and Pesquet-Popescue, 2007, Pesquet-Popescu and Pesquet, 2011] in image processing have been used some of the most well adapted ones like Haar wavelets [Haar, 1910], 5/3 wavelets and 9/7 wavelets [Antonini et al., 1992]. A dyadic wavelet decomposition is illustrated in Fig. 2.11 (b). In the case of a spatial transform, long filters can be used in order to obtain a good decorrelation. The great support of 9/7 wavelets (9 samples for the analysis and 7 for the synthesis) and their bi-orthogonality, even nearly orthogonality, caused them a very efficient transform to be used in several image coding schemes, as JPEG2000.

In general wavelets decompose an image at different scales using a pyramidal algorithm architecture. The decomposition is along the vertical and horizontal directions and maintains constant the number of pixels required to describe the image. The wavelet function is generated by dilations and translations of a function ψ which is defined as following:

$$\psi_{a,b}(x) = |a|^{-1/2}\psi\left(\frac{x-b}{a}\right), \tag{2.27}$$

where $(a,b) \in \mathbb{R}$ and $a \neq 0$. High frequency wavelets correspond to $a < 1$ or narrow width, while low frequency wavelets have $a > 1$ or wider width. For a wavelet of orthogonal bases $\mathbb{L}^2$:

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n), \quad \text{where} \quad m,n \in \mathbb{Z}^2 \tag{2.28}$$

the wavelet coefficients are given by:

$$c_{m,n}(f) = \langle f, \psi_{m,n}\rangle = \sum f(x)\overline{\psi}_{m,n}(x). \tag{2.29}$$

One can represent any arbitrary function f as superposition of wavelets.

$$f(x) = c_{m,n}(f)\psi_{m,n}(x). \tag{2.30}$$

2.4.3.3 Quantization

Both JPEG and JPEG2000 use quantization in their lossy format. A typical quantization function is responsible to map several inputs to a single output. This process is irreversible and it causes a loss of information. The quantization has two basic formulations the uniform and the non-uniform. JPEG standard uses only the first one. There are two different uniform scalar quantizers, the midtread which has zero as one of its quantized values and the midrise which has no zeros. Let v the input of a uniform midrise scalar quantizer and $Q_q^*(v)$ its quantized value which is given as following:

$$Q_q^*(v) = \text{sgn}(v) q \Big\lfloor \frac{|v|}{q} + 1 \Big\rfloor, \tag{2.31}$$

where q the quantization step, $\lfloor . \rfloor$ is the floor operator and $\text{sgn}(v)$ stands for the sign of the input v. Concerning the JPEG encoding process, the input of the uniform scalar quantizer is the result of the DCT transform and its output is $F^Q(u,v) = Q_q^*(F(u,v))$. A modified

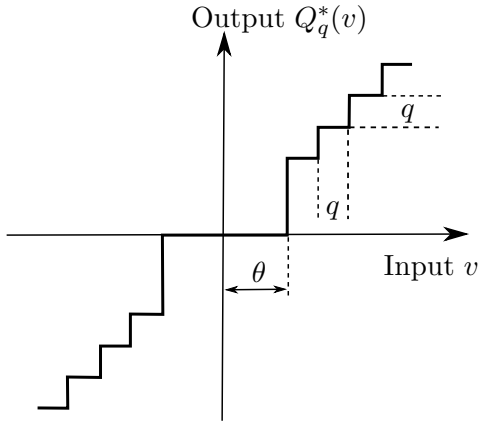


Figure 2.12: Dead-zone uniform quantizer.

version of the uniform scalar quantizer is the dead-zone quantizer. The dead-zone quantizer depends on a threshold θ which is responsible to discard all the inputs which are lower than θ while the rest of them are quantized with a uniform scalar quantizer:

$$Q_q^*(v) = \text{sgn}(v) q \max\Big(0, \Big\lfloor \frac{|v| - \theta}{q} + 1 \Big\rfloor \Big), \tag{2.32}$$

where 2θ is the size of the dead-zone, thus θ is half the dead-zone.

2.4.3.4 Entropy coding

The entropy coding has been explicitly described in section 2.3.2. Huffman and Arithmetic entropy coding which are used by JPEG and JPEG2000 have been explicitly described in section 2.3.2. JPEG2000 uses also the Embedded Block Coding with Optimal Truncation (EBCOT) coding [Taubman, 2000] which is a bit-plane coder before the Arithmetic coder.

2.4.3.5 Overview MJPEG and MJPEG2000

The performance of JPEG standard motivated people to apply JPEG not only to images but also to videos. A video stream is a group of pictures which change with respect to time (see eq. 2.22). When JPEG was applied to each picture of a video stream it succeeded in eliminating the spatial redundancy of the picture. In video compression terminology this is called *intraframe* coding. That was the first video compression format which was called Motion-JPEG (MJPEG). Motion-JPEG2000 (MJPEG2000) is an alternative to MJPEG

which uses JPEG2000 instead of JPEG. MJPEG2000 is based on the same architecture as MJPEG even though it was standardized many years later. However, both these formats are insufficient to reduce the temporal redundancy, which is called *interframe* coding.

In a video stream, usually, the only difference between sequential pictures is the camera moving or an object which is moving in the scene. As a result, it is not necessary to encode all these similar pictures but just the difference with respect to some reference picture. To enrich the performance of MJPEG in order to achieve better video compression, it is necessary to seek for solutions which enable to encode only what changes into a scene with respect to some reference picture. For that reason, there have been proposed several techniques which are enclosed in video compression standards. We are going to discuss the most important ones within a brief overview of the ITU/MPEG standards.

2.4.4 Overview of MPEG-1

MPEG-1 was released in 1990 introducing the most important techniques to reduce temporal redundancy. The goal of this standard was to achieve a bitrate of 1.5Mbit/s for a non-interlaced video of CIF picture format (352×288pixels) and 24-30pps (pictures per second). To achieve this goal the designers introduced the notion of *temporal prediction* which is also called interframe prediction or motion compensation. This prediction is used between samples of the current picture and a previously coded picture.

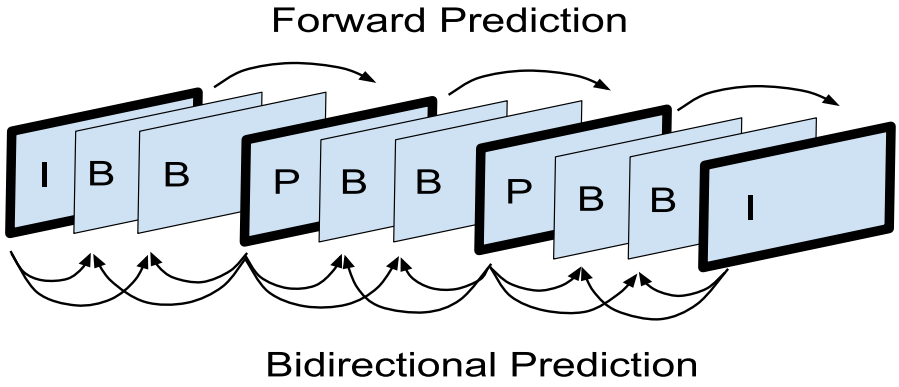


Figure 2.13: This Figure shows a GoP and the intra- and inter-picture prediction.

The pictures of a video stream $f(x,t)$ (see eq. 2.22) are separated into three different groups: the *Intra-pictures* (I-pictures), which are encoded following the block diagram of JPEG, the *Prediction-pictures* (P-pictures) and the *Bidirectional-pictures* (B-pictures) which are predicted using interframe method. The length between two I-pictures is called *Group of Pictures (GoP)*. MPEG-1 usually uses 15-18 pictures. However, this length may vary between 8 to 32 pictures for coding efficiency [Schwarz et al., 2007]. The P-pictures and B-pictures belong to the GoP (see Fig. 2.13). One would expect that predicting the value of each pixel in space or in time would be an efficient solution to reduce redundancy. However, this solution would be problematic in presence of noise. As a result, people introduced the *macroblocks* as a sample unit (see Fig. 2.14). Macroblocks and motion compensation are the basic properties of all the video compression standards.

2.4.4.1 Macroblocks

The macroblock, corresponding to a region of a picture, is the basic unit for motion compensated prediction which was first introduced in MPEG-1 and then was adopted by all the successor video compression standards (see Fig. 2.14). The dimension of each macroblock is 16×16 (or 8×8) and was chosen in order to provide efficient temporal redundancy under low computational requirement. As a result, each picture is partitioned in macroblocks which are used to estimate motion. The macroblocks of an I-picture are called I-macroblocks, the ones of P-picture are called P-macroblocks and lastly, B-macroblocks correspond to B-pictures.

2.4.4.2 Motion Compensation

Between two pictures, a reference frame and a current frame, one needs to find the prediction by subtracting the current picture by the reference picture. Figure 2.14 shows two pictures, which are subdivided into macroblocks. One is interested in predicting the macroblock of the current (blue) picture based on the macroblocks which belong to the searching area (green) of the reference picture. The simplest prediction is to subtract the current macroblock by the reference macroblock of the same position. However, this simple prediction results in a residual macroblock (yellow) of a lot of energy, which means that still there is a significant information which could be reduced. This is done by compensating motion.

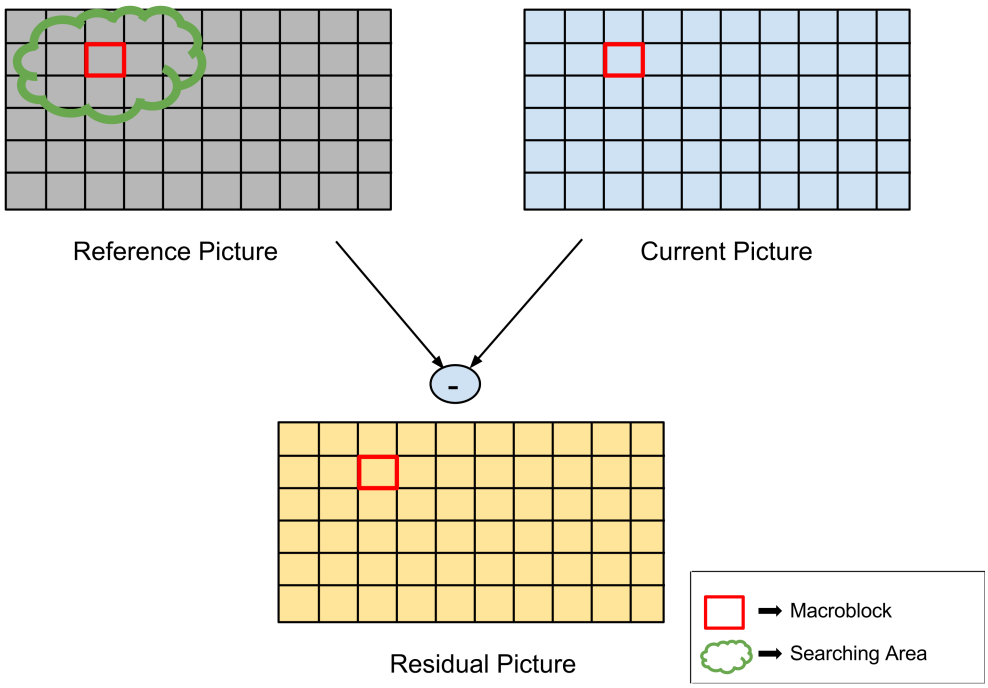


Figure 2.14: This figure shows a GoP and the intra- and inter-picture prediction.

Motion estimation determines the motion vectors which describe the transform between two pictures [Stiller and Konrad, 1999, Konrad, 2000, Pesquet-Popescu et al., 2014]. For each macroblock of a current picture we search in a searching area, around the same position of the reference frame (in Fig. 2.14 is illustrated by the green cloud), for the best matching. The best matching is related to the energy. The macroblock of the reference frame which minimizes the energy of the residual is chosen as the best one. The macroblock with the minimum MSE or minimum Sum of Absolute Difference (SAD) or the maximum correlation with respect to the reference picture becomes the predictor of the current macroblock and

it is subtracted from the current macroblock to form the residual macroblock. This process is called *motion compensation*. The residual of each macroblock is coded and transmitted to the decoder. In addition, the offset between the current macroblock and the position of the candidate region of the reference picture is also coded. This offset is called *motion vector*. A motion vector is used to recreate the predictor region in the previously coded reference picture. Then, the residual macroblock is decoded and added to the predictor for the reconstruction of the original macroblock.

When a picture is chosen to be the reference picture, it should be first encoded. The reference picture could be past or future picture with respect to the location of the current picture. If the motion estimation of the current picture does not match the reference picture, it means that there is a strong motion or a change of the scene and it is preferred that the current picture to be entirely encoded.

The I-pictures are encoded/decoded based on JPEG without any reference to other picture. As a result, for each 8×8 block which is contained into each macroblock of the I-pictures the encoder applies the 8×8 DCT transform, the scalar quantization and the entropy coding. Then, the output of the encoder is sent to the decoder. The decoder uses the entropy decoding, the de-quantization and the inverse DCT transform to reconstruct the I-picture.

To encode a P-picture, the previously I- or P-picture should be stored in both encoder and decoder. Motion estimation is performed on a macroblock basis between two current P-picture or the previous I- or P-picture which is described by motion vectors. One motion vector is calculated for each macroblock. The motion compensated prediction error is calculated by subtracting each pel in a macroblock of the current P-picture with its motion shifted counterpart in the previous I- or P-picture (pels are called the number of pixels on a screen). The predicted error is encoded and transmitted to the decoder. The encoding process is the same for B-pictures except they are able to do prediction not only for the following but also the previous pictures (see Fig. 2.13). The precision of motion vectors is 1/2 or 1/4 a pixel (*half pel* or *quarter pel*). The finer the precision of motion vectors is, the better the compression, but at the same time that would cause an increase of complexity.

The disadvantage of MPEG-1 is the generation of some block effects due to the division of the picture into macroblocks which are easily perceived by the visual system at low bitrates. In addition, this standard is unable to support high definition frame rate videos or interlaced videos.

2.4.5 Overview of MPEG-2

The successor of MPEG-1 is MPEG-2 which one the one hand, has similar format but it also has the capability to support interlaced video coding [ISO/IEC 10918-1:2000, 2000, Sikora, 1997, Bhaskaran and Konstantinides, 1997]. Interlaced video coding is a technique that doubles the frame rate perception of a video stream while it is displayed without consuming any extra bandwidth. A non-interlaced video is of a normal frame rate (25-30pps) . An interlaced video doubles the temporal resolution (50-60pps) in order to precisely estimate motion. However, every encoded picture is the result of a fusion between two consecutive pictures which are called fields. The first field is used to be displayed on the odd-lines of the picture while the second one on the even lines. Fusing the two fields gives the impression of motion. The advantage of interlaced videos is the double spatiotemporal resolution since it is optimized for SD and HD, achieving the same bitrate as MPEG-1. MPEG-2 is widely used for transmission of TV signal over satellite, cable, terrestrial emission and of standard of high definition videos onto DVDs.

The disadvantage of this method is the generation of artifact, known as *interlacing effects* or *combing*. These artifacts occur when an object moves very fast and its position is different when the two fields are captured. To overcome this problem, there are many simple methods like doubling the number of lines of one field and omitting the other or

anti-aliasing the signal by removing high frequency components or in case that the motion is along x-axis, one could shift one of the fields in order to match the new position. Last but not least, even though MPEG-2 has the same performance as MPEG-1, it is much more computationally demanding than MPEG-1 for compression.

2.4.6 Overview of H.264/MPEG-4/AVC

The increasing number of services and growing popularity of high definition TV were creating greater needs for higher coding efficiency than MPEG-2. In addition, other transmission media like cable modem, xDSL or UMTS require lower data rates than broadcasting channels, and enhance the gab of coding efficiency. The H.264/MPEG-4/AVC standard was released in 2003 and its target was to reduce half of the bitrate comparing to MPEG-2 [Pereira, 2000]. To reach this goal designers enforced the algorithm with many properties like variable block-size motion compensation, quarter-sample accurate motion compensation, multiple reference picture motion compensation, flexible interlaced scan video coding, directional spatial prediction for intraframe coding, deblocking filter, small block-size transform, CABAC entropy coding and Scalable Video Coding (SVC). This standard is currently used in variety of applications like Blu-ray discs, Streaming internet sources (YouTube, Vimeo, iTunes, etc), web softwares (Adobe Flash Player, Microsoft Silverlight), High Definition TeleVision (HDTV) broadcasting and Closed Circuit TV (CCTV) systems [T. Wiegand and Luthra, 2003]. We are going to briefly discuss the advantage of each one of the properties before we introduce the coding schema of H.264/MPEG-4/AVC standard [ISO/IEC 10918-1:2004, 2004] [Pereira and Ebrahimi, 2002, Ostermann et al., 2004].

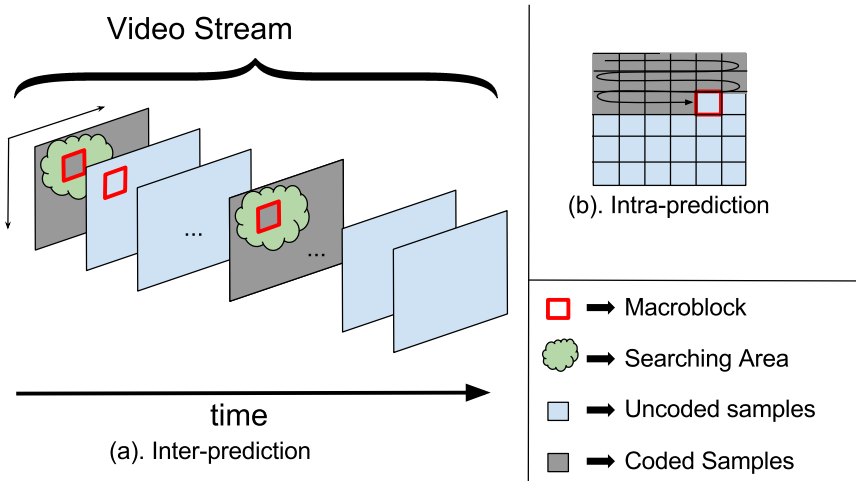


Figure 2.15: This Figure illustrates a macroblock (red block) which exist along the picture of a video stream whilst its position changes.

- **Variable block-size Motion Compensation:** This standard allows higher flexibility in the selection of motion compensation block-sizes which could vary between 16×16, 8×8 and 4×4. The smaller the size of the macroblock, the higher the precision of the motion vectors. Thus, the energy of the residual macroblock will be decreased more efficiently. However, reducing the size of the macroblock increases the complexity of the algorithms since more data need to be processed.

- **Quarter-sample Accurate Motion Compensation:** It is already mentioned that the precision of motion vectors was half pel in MPEG-1 and MPEG-2. In H.264/MPEG-4/AVC standards, the designers improved this results adding quarter pel for each motion vector sacrificing again the complexity of the algorithm which was increased with this improvement.

- **Multiple Reference Picture Motion Compensation:** Prediction coding of a current P-picture in MPEG-2 and its predecessors used only one previous I- or P-picture. The new design of H.264/MPEG-4/AVC, which was first introduced in H.263++, enables the encoder to select for motion compensation purposes among a larger number of pictures which have been decoded or stored in the decoder. Figure 2.15 (a) shows a video stream $f(\boldsymbol{x}, t)$ (see eq. (2.22)) which consists of a group of pictures. In this example is illustrated a GoP where I-picture are depicted in gray and P- and B-pictures in blue. Lets suppose that the current P-picture is $f_2(\boldsymbol{x})$ and one seeks for the best matching of the current P-macroblock. In previous standards, this P-picture would be possible to be predicted based on previously coded I- or P-pictures. So, in this example the reference frame would be only $f_1(\boldsymbol{x})$. However, H.264/MPEG-4/AVC allows $f_r(\boldsymbol{x})$ to be also a possible reference picture where one seeks for the best matching I- or P-macroblock. This extension had been applied also to B-pictures which are capable to predict values using previous or next I- or P- pictures. This property may allows a better matching of the macroblock but it also increases the complexity of the algorithm.

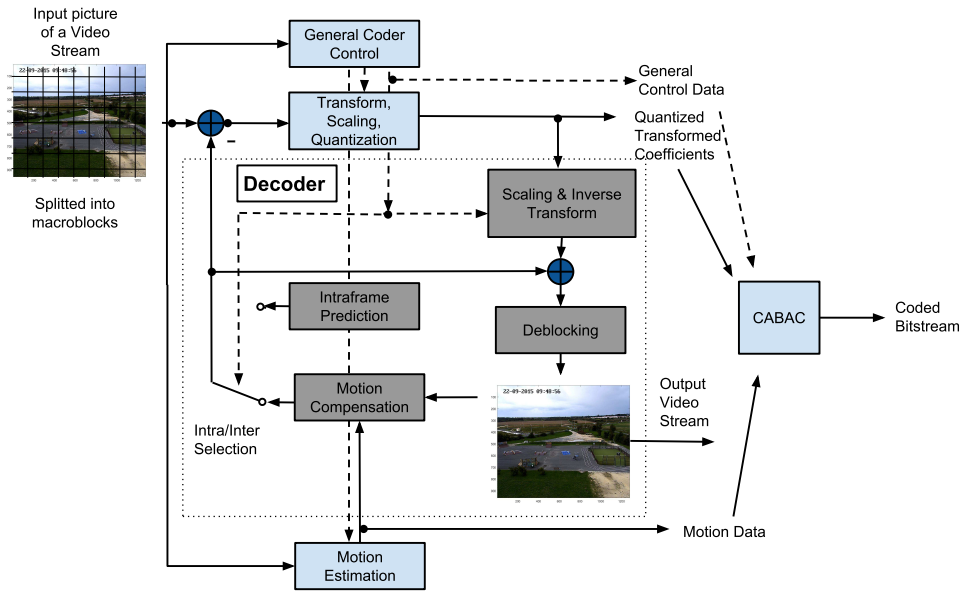
- **Directional Spatial Prediction for Intraframe Coding:** As it was discussed before, an intraframe coding was based on the coding schema of JPEG standard. However, H.264/MPEG-4/AVC improved this intraframe coding by introducing some intraframe predictions. This new technique is a spatial prediction which does not depend on other pictures but the current one. As usual, the picture is subdivided into macroblock. The first macroblock which is processed and entirely encoded following JPEG standard is the top/left macroblock. The rest macroblocks are predicted with respect to all the previously encoded macroblocks (positioned at left and top). Once the prediction has been generated, it is subtracted from the current block to form a residual in a similar way to inter prediction. The residual is transformed and encoded, together with an indication of how the prediction was generated. Figure 2.15 (b) shows an example of an intraframe prediction where the macroblocks are predicted in a raster-scan order.

- **Deblocking Filter:** Macroblocks are necessary in video coding. However, they cause some block artifacts which are originated from prediction and residual coding of the decoding process. A solution to this problem was given by an adaptive deblocking filter which improved video quality [Chebbo et al., 2009]. This filter was inserted into the motion compensation loop, so that this improvement of the quality can be used in interframe prediction and thus the algorithm will predict more efficiently.

- **Small block-size Transform:** As it is described above, all the video standards are based on DCT or DWT which is applied to 8×8 block. However, this block size causes some artifacts as "blocking" effects for DCT or "ringing" effects for DWT. H.264/MPEG-4/AVC corrects these artifacts by reducing the block size into 4×4. A small block size enables the encoder to be better locally adaptive to the signal, which causes its better representation.

- **Content-Adaptive Binary Arithmetic Coding (CABAC):** The CABAC entropy coding is based on Arithmetic Coding (see section 2.3.2.2) which was already introduced before. The combination CABAC and Context-Adaptive Variable-Length Coding (CAVLC) resulted in context-based adaptivity which further improved the coding efficiency.

- **Scalable Video Coding (SVC):** Scalable Video Coding (SVC) is a highly attractive solution to the problems posed by the characteristics of modern video transmission systems [Schwarz et al., 2007]. It is an extension of H.264/MPEG-4/AVC and its objective is to enable the encoding of a high-quality video bit stream that contains



!h

Figure 2.16: Basic H.264/MPEG-4/AVC encoding architecture.

one or more subset bit streams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the existing H.264/MPEG-4/AVC design with the same quantity of data as in the subset bit stream. Spatial and temporal scalability describe cases where subsets are selected from the initial video stream with reduced size (spatial) or frame rate (temporal). The quality scalability provides the spatiotemporal resolution of the video stream but the distortion which is measured is of a lower PSNR value. Quality scalability is also referred as fidelity of SNR scalability. Another option of scalability is the Regions Of Interest (ROI) of an pictures. For some applications, like CCTV systems it is necessary to provide higher resolution for some regions of the input scene while the rest could be displayed in lower quality.

Figure 2.16 is the encoding architecture of the H264/MPEG-4/AVC standard. The input picture which is subdivided into macroblocks is the input of the general coder control. In case of the top/left macroblock of an I-picture, it is transformed, scaled, quantized and encoded by the CABAC into a bitstream which is ready to be transmitted. For the rest of the I-macroblocks the general coder control is linked to the motion estimation. The intraframe prediction mode is selected which requires of previously encoded I-macroblocks to be decoded and used in order to predict the current I-macroblock. The best matching macroblock is subtracted by the current macroblock resulting in the residual macroblock which is transformed, scaled, quantized and sent to the CABAC. In case of a P- or B-macroblocks the general coder is linked to the motion estimation and motion compensation mode. Then, the past or future reference pictures/macroblocks are decoded to find the best matching. Once it is found within a searching area of the reference pictures it is subtracted by the current P- or B- picture which is encoded into a bitstream. As we have explained before, the motion vectors of each intra- or interframe prediction are also encoded by CABAC to inform the receiver about the exact position of the reference macroblock.

2.4.7 Overview of H.265

The H.265/HEVC is the latest standard in video compression which was released in order to provide almost 50% bitrate reduction comparing to H.264/MPEG-4/AVC and deal with HDTV and Ultra-HDTV signal [Sullivan et al., 2012]. The HEVC consists of a variety of

methods some of which are common with H.264/MPEG-4/AVC and prior standards, like block based coding tools of variable block size, block based motion compensation with a quarter-ample accuracy, spatial intraframe prediction, arithmetic coding and deblocking filter. Apparently, there are significant changes of some methods, like macroblocks, which on the one hand, improve the performance of the standard but on the other hand they increase its complexity. One of these changes was with respect to the macroblocks which were replaced by the Coding Tree Units (CTUs) known also as quadtree. Figure 2.17 shows the coding schema of HEVC which is similar to the one of H.264/MPEG-4/AVC except for the fact that instead of macroblocks the prediction and encoding process is applied to CTUs.

- **Coding Tree Units (CTUs) and Coding Tree Blocks (CTBs):** CTUs are larger block structures of up to 64×64 samples and can better subdivide the picture into variable sized structure. HEVC initially divides the picture into CTUs which can be later subdivided into Coding Tree Blocks (CTBs) of 64×64, 32×32, or 16×16 with a larger pixel block size usually increasing the coding efficiency.

- **Coding Units (CUs) and Coding Blocks (CBs):** The quadtree syntax of the CTU specifies the size and positions of its luma and chroma Coding Blocks (CBs). The root of the quadtree is associated with the CTU. Hence, the size of the luma CTB is the largest supported size for a luma CB. The splitting of a CTU into luma and chroma CBs is signaled jointly. One luma CB and ordinarily two chroma CBs, together with associated syntax, form a Coding Unit (CU). A CTB may contain only one CU or may be split to form multiple CUs, and each CU has an associated partitioning into prediction units (PUs) and a tree of transform units (TUs).

- **Prediction Units (PUs) and Prediction Blocks (PBs):** The decision whether to code a picture area using interpicture or intrapicture prediction is made at the CU level, this why the PU partitioning structure has its root at the CU level. HEVC supports variable PB sizes from 6464 down to 44 samples.

- **Transform Units (TUs)and Transform Blocks (TBs):** The prediction residual is coded using block transforms, as a result the TU tree structure has its root at the CU level. The luma CB residual may be identical to the luma transform block (TB) or may be further split into smaller luma TBs. The same applies to the chroma TBs. Integer basis functions similar to those of a discrete cosine transform (DCT) are defined for the square TB sizes 44, 88, 1616, and 3232. For the 44 transform of luma intrapicture prediction residuals, an integer transform derived from a form of discrete sine transform (DST) is alternatively specified.

2.5 Alternative Video Compression Algorithms

The indisputable progress of coding systems has been shown in section 2.4. The latest video compression standard HEVC has doubled the performance of its prior AVC. However, except for the performance, the computational cost has also been doubled, which is the main drawback of HEVC. It is easy to notice that scientists need to pay the price of complexity in order to achieve lower bitrates.

2.5.1 VP9

Some new attempts have been done in order to find the trade-off between the complexity and the coding efficiency which is higher or at least equal to H.264/MPEG-4/AVC or HEVC. The VP9 is an open source video coder proposed by Google. Its development started in 2011 and it was finally released in 2013. Its architecture is based on H.264/MPEG-4/AVC

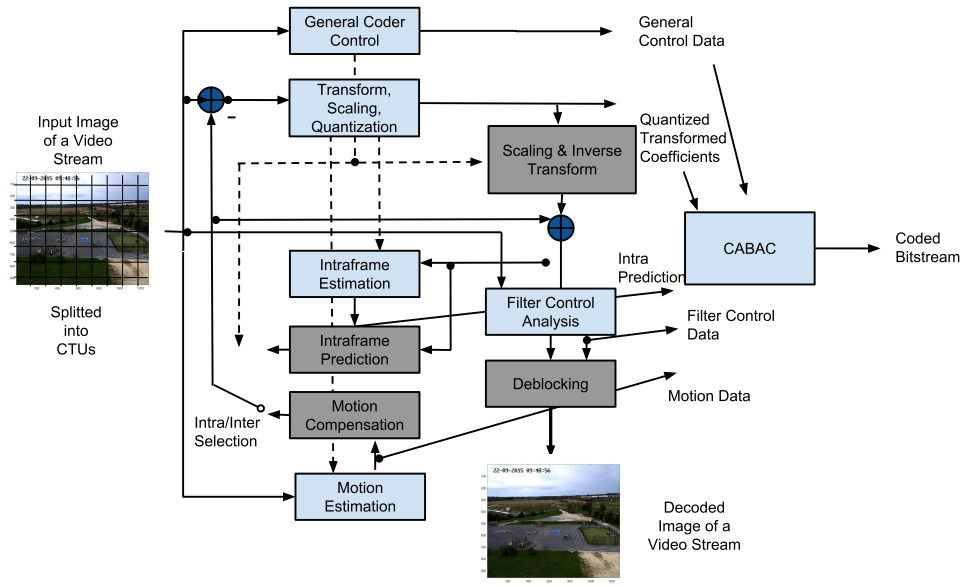


Figure 2.17: Basic HEVC encoding architecture.

standard as it uses block-based prediction, intraframe and interframe prediction, half pel accuracy, transformation, entropy coding and deblocking filter. The novelty of this standard is the superblock structure which replaces the macroblock. The idea was close to the CTU of HEVC while the designers also noticed that a HD video may show correlation over larger areas [Mukherjee et al., 2013]. An interesting comparison of H.264/MPEG-4/AVC, HEVC and VP9 is given in [Grois et al., 2013], where the authors concluded that the most efficient coding algorithm is HEVC providing 43.3% and 39/3% bitrate savings compared to VP9 and H.264/MPEG-4/AVC respectively. In addition, even if the performance of VP9 was superior compared to H.264/MPEG-4/AVC (bitrate gain 8.4%) the computational time of VP9 was 100times higher than H.264/MPEG-4/AVC.

2.5.2 Green Metadata Standard

Unfortunately, the above results show that it is very difficult to tackle the coding complexity and delay of video compression algorithms. In modern electronic devices, the complexity of the coding/decoding algorithm and the display units consume a lot of power. This power, especially in case like mobile phones or nomadic video surveillance systems, where the energy/battery is limited is very important. A new standard which is called "Green Metadata Standard" concerns about the energy-efficiency of video compression algorithms during the display, encoding and decoding process [Fernandes et al., 2015]. Green Metadata was standardized for energy-efficient video consumption without any loss in the Quality of Experience (QoE). Green Metadata reduces the energy consumption of H.264/MPEG-2/AVC even when QoE is maintained but this reduction gets higher when QoE varies.

Concerning the display processing, Green Metadata is kind of a successor of the Display Adaptation (DA) Model which is also called backlight dimming [Cheng and Pedram, 2004, Huang et al., 2013]. The role of DA is basically to take advantage of the negligible power consumption changes during the display, when the RGB values vary. Thus, DA was used to reduce the power consumption without sacrificing quality just by dimming the backlight of the Liquid Crystal Display (LCD) while at same time it adjusts the RGB values according to the dimming level. In [Fernandes et al., 2015], the authors provide some results in which backlight was reduced from 26-65 percent. Although, the DA technique is efficient in terms of saving energy, it produces a lot of artifacts. Consequently, the scaling up of the RGB values is not enough to sufficiently restore the quality. The Green Metadata standard

solves the quality problem using some contrast enhancement within the dynamic range that contains the majority of the RGB values. In fact, the new standards produces some metadata which consist of signaling RGB statistics, quality-level indicators and dynamic range bounds. The dimming of the backlight is set according to the RGB statistics. The dynamic-range bounds are used for contrast enhancement, improving the perceived quality. If the power reduction is small (high) then the backlight settings will be derived by RGB statistics associated with a high (low) quality level.

The power reduction can be also achieved in the encoder generating alternate low-quality and high-quality segments during the pre-processing of the encoding step (see Fig. 2.18). For low-quality segments, one is able to reduce the complexity because in practice it requires fewer encoding modes, fewer reference frames, smaller search range, etc. The Cross-Segment Decoding (XSD) is a technique which manages to enhance the decoding of the low-quality segments utilizing information by the high-quality segments. Finally, XSD enables higher quality of QoE reducing the average encoding complexity and therefore the power consumption.

For decoder power reduction, the major problem arises in term of high CPU frequency which is linked to the high power consumption. Low frequency values may cause some problems in decoding of complex pictures. Thus, metadata that indicate the picture-decoding complexity is embedded in the bit-stream which is transmitted to the receiver. This metadata is used by the receiver to set the GPU frequency at the lowest possible level which guarantees that the decoding completion will happen within the frame-rate deadlines.

Fig. 2.18 describes the functional architecture of a system which uses Green Metadata. The metadata which are produced during the pre-processing and/or the encoding process are forwarded to the Power Optimization Module of the receiver. The Power Optimization Module interprets them and then applies controls to reduce the power consumption during decoding and display of the video. This module is also responsible to send a Green feedback to the transmitter concerning the energy of the system (i.e. remaining battery of the phone). This feedback is used to adjust the encoding process of the transmitter. This method may increase the complexity of the first picture of the video stream but the rest of the pictures will be encoded, decoded and displayed in a very low complexity similar to the one of H.264/MPEGS-4/AVC.

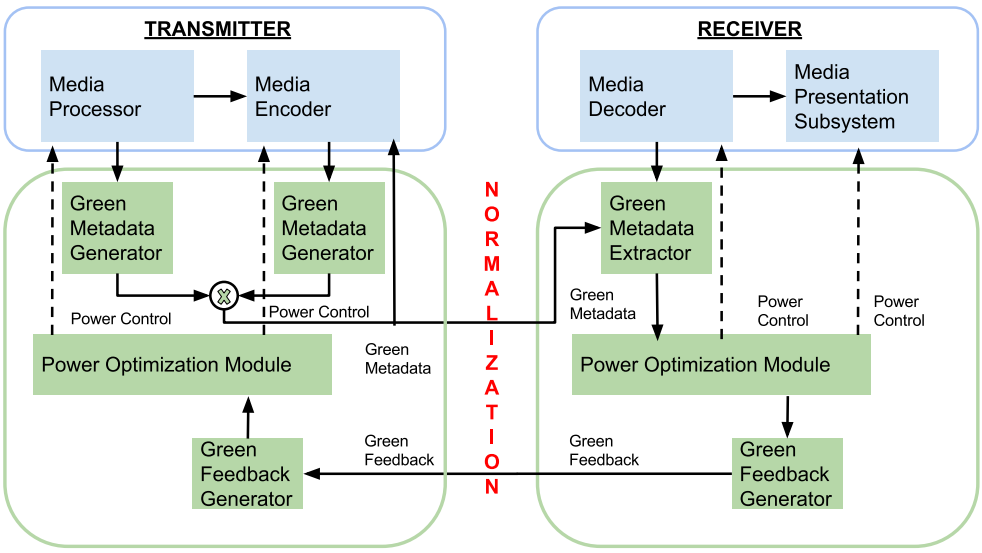


Figure 2.18: Functional architecture of a system that uses Green Metadata [Fernandes et al., 2015].

2.6 IEEE 1857 Standard

The beginning of this section explains that there are numerous of applications which are really demanding of low power consumption and complexity coding algorithms. One of these applications is the CCTV systems. Nowadays, CCTV cameras have been placed almost everywhere (i.e building, public places, streets, transport means, remote areas to protect civil areas from natural disasters, working areas, etc.). Their primary goal is to survey an area for 24h per day for security reasons. CCTV systems may survey areas where the infrastructure for data transmission over communication channels is poor and require low power consumption. In such a case, the increase of the computational cost and the power consumption of HEVC is problematic. One should also keep in mind that during the last few years almost every CCTV which is consisted of a single camera has been enriched by double or higher number of cameras. As a result, the amount of videos of CCTV is huge and it is impossible to be saved without efficient compression algorithms. However, according to Fig. 2.19, it seems that the progression rate of compression standards is too slow comparing to the explosive growth of the amount of data which need to be stored and transmitted.

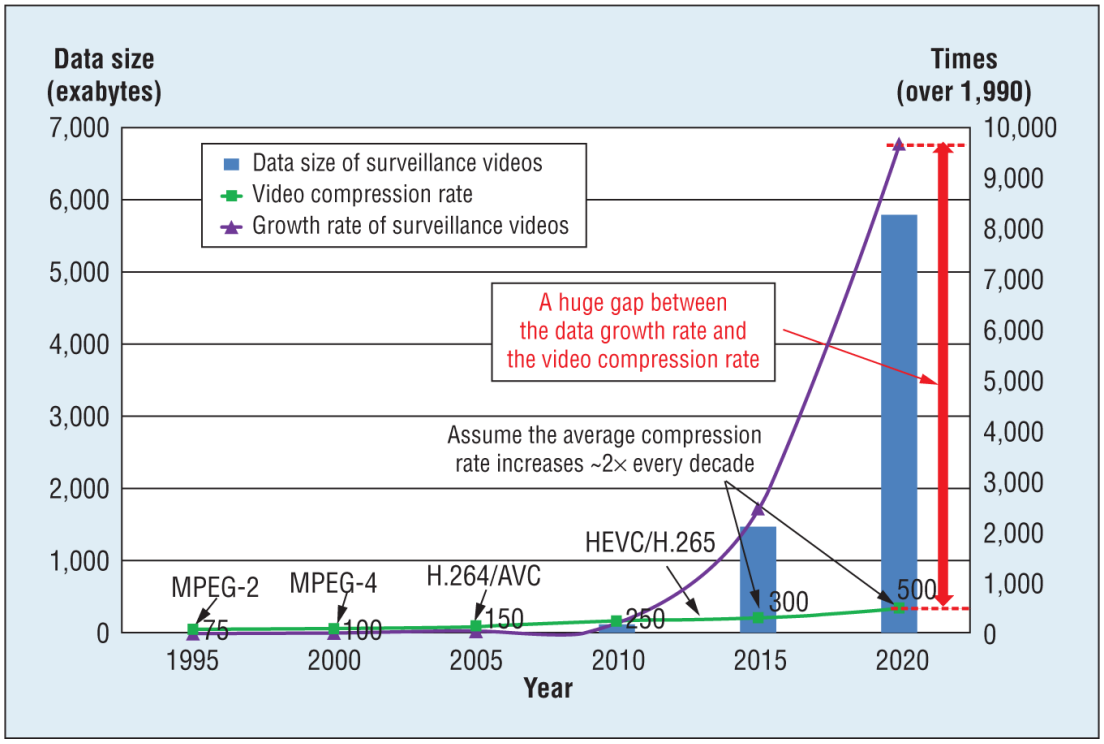


Figure 2.19: This graph illustrates the huge gad between the progress of compression algorithms and the increase of the amount of data captured for video surveillance reasons. It is expected according to the increase of rate that this gab is going to be bigger in the near future [Gao et al., 2013].

An interesting solution to improve the intelligence of coding for CCTV systems was proposed in [Gao et al., 2013], where the authors introduced the IEEE 1857 Standard for Advanced Audio and Video Coding. The general framework of this algorithm is based on H.264/MPEG-4/AVC standard but it enables to double the surveillance video coding efficiency saving computational time. The IEEE 1857 takes advantage of the background and foreground data of the scene. As a result, if the input surveyed area is coded in the beginning, the number of I-pictures which are entirely coded/decoded could be significantly reduced since the GoP is much larger comparing to other standards. This is the basic improvement of IEEE 1857 standard which offers 45.89% bitrate gain and 45.86% time

gain comparing to H.264/MPEG-4/AVC.

2.7 Conclusion: What is the future of video compression?

A general true is that compression algorithms already stand close to a performance ceiling which does not respond to the requirements of the new technological devices. To our point of view, the basic drawback of the conventional architecture stems from the fact that videos are dynamic signals which are processed by methods proposed for static images (DCT, DWT, scalar quantization, etc.) whilst at the same time an increasing number of techniques is proposed in order to reduce temporal redundancy like motion estimation. Although people propose new techniques to further improve the trade-off between the bitrate and the reconstruction quality, finally the gain is very small. As a result, we believe that a video should be processed dynamically. Concerning this dynamicity, we propose that an interesting model to mimic is the retina.

The retina is part of the visual system which belongs to the central nervous systems. It could be considered as an efficient machine which is responsible to dynamically capture, transform and encode the visual stimulus. Finally, the input stimulus is transmitted to the brain in the form of spike trains. The retina is able to deal with very high resolution signals and it requires a highly spread sources of light. In [Salamo and Jakobs, 1996] the authors compared two different sources of light, each one of which propagated an intensity of 0.001 Watt to the retina. The first source of light was a laser pointer which is of a high spatial concentration, while the second one was a light bulb of 100 Watts which was spread in space. Despite the retina received the same amount of intensity of both the above sources, the laser pointer was able to cause permanent damage to the eye because it was focused on a small area of the retina.

In the literature, there have been already some attempts to build dynamic encoding systems based on neuroscience like the ones proposed by Masmoudi *et al* [Masmoudi et al., 2012, Masmoudi et al., 2013] and Lazar *et al* [Lazar and Pnevmatikakis, 2011]. The first model is a bio-inspired image codec which uses image processing tools to approximate neuroscientific models in order to encode images. The second one proposes a video encoding machine which is based on neuromathematical models related to the spike generation process.

In this thesis, we propose a novel retina-inspired video codec which is applied to each picture of a video stream like MJPEG and MJPEG2000. This codec follows the conventional coding principle (see Fig. 2.2) but each of the conventional processes has been replaced by dynamic models which are inspired by neuroscience. As a result, we proposed a dynamic retina-inspired transform [Doutsi et al., 2016] and a dynamic encoding process which are both involved to the generation of the code. This code is informative enough to reconstruct each picture of the video stream.

Part II

DYNAMIC FILTERING

Motivation

As explained in chapter 2, the evolution of video compression algorithms shows that the decrease of the bitrate is inversely proportional to the complexity of the algorithm. This is due to the fact the video compression standards are based on image compression standards using JPEG or JPEG2000 in order to encode key I-pictures within a GoP and estimation the motion of the rest P- and B-pictures of the GoP. As a result, videos are basically processed with static techniques proposed for still-images while at the same time multiple other processing tools are used to decrease the bitrate and the spatiotemporal redundancy of the video pictures (see section 2.4). Thanks to the increase of the complexity people are forced to seek for different solutions for compression. We propose that since videos are dynamic signals they should be dynamically processed. In this thesis we are interested in changing the conventional coding principle (see Fig. 2.20 (a)) being motivated by the visual system model.

The visual system is an efficient machine which dynamically processes the input visual stimulus. In literature, there are many neuroscientific models which try to fit neuroscientific measurements of different organisms i.e. salamander, cat, monkeys, etc. We believe that this dynamic models will upgrade and benefit the video compression algorithms in terms of complexity, computational cost and power consumption.

This part is dedicated to the study and analysis of the neuromathematical models which have been proposed to describe how the retina manages to dynamically filter the visual stimuli (see Fig. 2.20 (b)). There have been proposed several models to approximate this retinal filter starting from static spatial models [Kuffler, 1952, Marr and Hildreth, 1980, Marr, 1982] which turned through the years into dynamic models [Fleet et al., 1985, Wohrer and Kornprobst, 2009] which fit more precisely neuroscientific measurements. The evolution and the characteristics of each one model is described in chapter 3. However, the general formula of the static neuroscientific models is similar to the very well known and studied Gaussian and Laplacian pyramids which has been widely used in signal processing for image analysis and synthesis [Burt and Adelson, 1983, Adelson et al., 1984]. As a result, one could assume that the first static neuroscientic models have already motivated people in image and video processing comparing to the latest dynamic ones which have not. In this thesis we are motivated to utilize the dynamic retina filtering models in order to dynamically process videos (see Fig. 2.20 (c)). Our first contribution is introduced in chapter 4 which is included in this part. This chapter represents under which assumptions the dynamic OPL retina transform could be adopted into the conventional coding principle (see Fig. 2.2). We propose and we study a non-separable OPL retina-inspired filter which is briefly termed as retina-inspired filter. Last but not least, we dedicate a general formula for the retina-inspired filter which is a group of Weighted DoGs (WDoG) in order to enrich the image processing community with a novel dynamic decomposition method.

Being interested in compression, we need to ensure that the retina-inspired decomposition is invertible because we need to reconstruct the input signal (see Fig. 2.20 (c)). In chapter 5 we prove thanks to the frame theory that the retina-inspired decomposition is invertible. Thus, we provide results about a perfect reconstruction when the full retina-inspired frame is used. In addition, chapter 5 also studies what is the impact of noise in

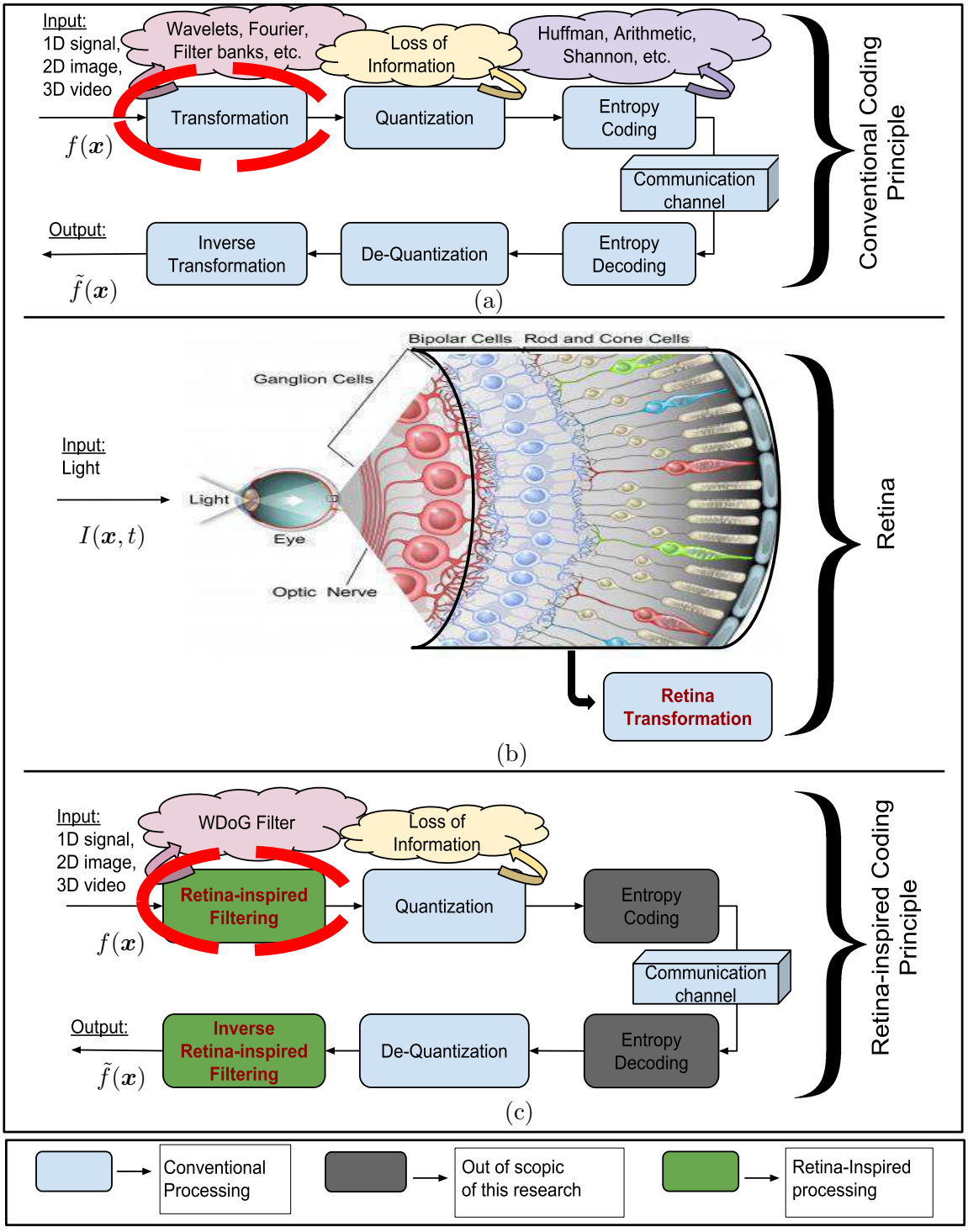


Figure 2.20: Motivation of the dynamic non-separable OPL retina-inspired filtering. (a) Conventional coding principle which consists of a static transform that we aim to replace with a dynamic one. (b) Retina cells which contribute to the dynamic retina transform. (c) Retina-inspired coding principle which consists of a retina-inspired filtering which has been proven to be invertible.

the reconstruction results.

Chapter 3

DoG Filters from Neuroscience to Image Processing

Contents

3.1 Introduction

This chapter is a brief introduction to the early visual system. We focus our attention on the retina and its Outer Plexiform Layer (OPL) which is responsible for capturing and transforming the input visual stimuli into electrical signal (current). We provide this assiduous study of the progress of OPL neuromathematical models because it is necessary for the signal processing community to uncover where the conventional models of image analysis/synthesis stand with respect to this progress. The OPL transform has been reported to be dynamic [Fleet et al., 1985, Wohrer and Kornprobst, 2009]. However, the first model was proposed by Kuffler [Kuffler, 1952] who approximated the OPL transform by a spatial Difference of Gaussian (DoG) filter. This model was improved by Marr [Marr and Hildreth, 1980, Marr, 1982] who introduced time resulting in a separable spatiotemporal DoG filter. Fleet in [Fleet et al., 1985] introduced a more accurate model

where space evolves with respect to time. This was the first dynamic (non-separable spatiotemporal) DoG filter which was improved by Wohrer in his work title as Virtual Retina [Wohrer and Kornprobst, 2009].

DoG filters are very well-known for image analysis/synthesis [Burt and Adelson, 1983]. However, they are not that efficient as the latest dynamic OPL filter. We are going to provide some retina-inspired encoding architectures which have adopted static DoG filters or DoG pyramidal filters, according to [Kuffler, 1952] and [Marr and Hildreth, 1980] respectively, like the Rank Order Coder (ROC) [Thorpe, 1990, Thorpe and Gautrais, 1998, Rullen and Thorpe, 2001, Thorpe et al., 2001] and its extensions [Masmoudi et al., 2012, Masmoudi et al., 2013]. However, these filter banks are very rough approximations comparing to the dynamic retina filtering. As a result, it would be interesting to study a dynamic retina filter for image synthesis/analysis and adopt it into the conventional coding principle, which is the first contribution of this thesis (see chapter 4).

3.2 Introduction to the Visual System

The visual system is part of the Central Nervous System (CNS). It consists of many different areas which participate to the coding of a visual stimulus. The most important and better studied areas are the retina, the optic nerve, the Lateral Geniculate Nucleus (LGN) and the visual cortex (Fig. 3.1) [Hubel, 1963]. The retina is a layer of tissue, lining the inner surface of the eye, which is responsible to capture, transform and encode the visual stimuli into a sequence of electrical impulses (spike trains). This code of spikes is transmitted through the optic nerve to the LGN cells which correlates not only spatially but also temporally the output signal of each of the two eyes in order to achieve a 3D "representation" of object space. The cells in visual cortex are more complex and sensitive to edge and orientation detection, motion estimation, discrimination of the shape or the color, etc.

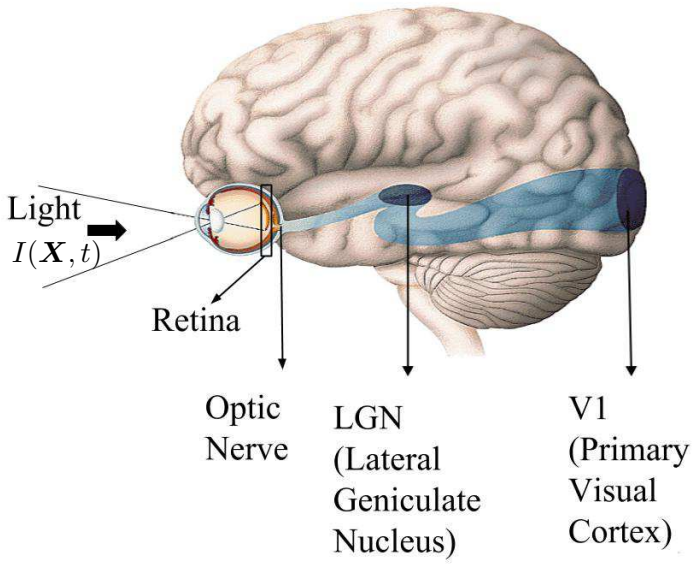


Figure 3.1: The visual system pathway.

3.2.1 Retina

The first and most important criterion for the luminance of light $I(\boldsymbol{X},t)$, where $\boldsymbol{X} \in R^3$ and $t \in \mathbb{R}^+$, which is the spatiotemporally varying visual stimulus, is to be transformed in order to fit the brain. This transformation takes place inside the *retina*

[Masland, 2001, Kolb, 2004, ter Haar Romeny, 2003, VanEssen et al., 2005]. The retina is a complex structure which is responsible for the light absorption and its transformation into electrical impulses. It consists of many different cells (Fig. 3.2) which differ with each other not only in shape but also in the way they act [Masland, 2011]. These cells are the photoreceptors, the horizontal cells, the bipolar cells, the amacrine cells and the ganglion cells.

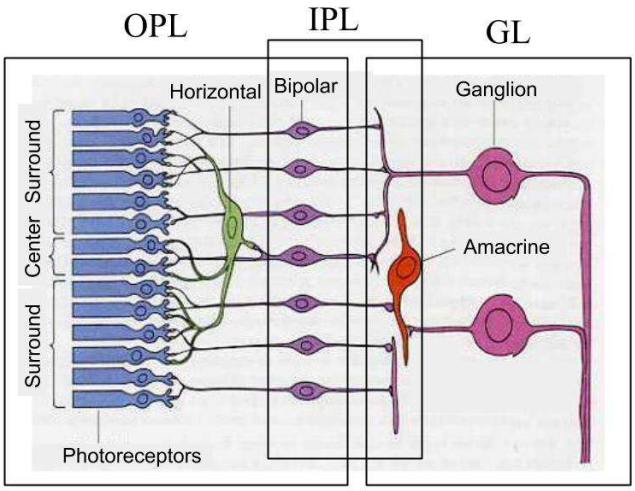


Figure 3.2: The retinal layers according to [Wohrer and Kornprobst, 2009]. This figure shows the connectivity and hierarchical structure of the retinal cells.

These cells form 3 layers; the Outer Plexiform Layer (OPL), the Inner Plexiform Layer (IPL) and the Ganglionic Layer (GL) each one linked to a different process necessary for the transmission of the visual information to the visual cortex [Wohrer and Kornprobst, 2009]. The OPL consists of photoreceptors, horizontal and bipolar cells, the IPL is structured by bipolar and amacrine cells and the GL by amacrine and ganglion cells. There is an overlapping between these cells which is due to the feedforward and feedback messages which are exchanged between the cells. The transformation of the visual stimuli into electrical signal happens by photoreceptors. Then, this signal has to be refined and controlled by the rest of the retinal cells in the OPL and IPL layers which are both filtering stages of the electrical signal. Finally, the transformed signal is sent to the GL where it is sampled in order to form a train of spikes (electrical impulses) [Wohrer and Kornprobst, 2009]. From all the above cells, only the ganglion cells are able to emit electrical impulses (spikes). As a result, the output of ganglion cells is a spiking train whereas the output of the rest ones is a change at their membrane potential.

In the literature, there have been many models which are based on neuroscientific measurements, trying to mathematically describe how each one of these layers works [Kuffler, 1952, Fleet et al., 1985, Marr and Hildreth, 1980, Marr, 1982, Wohrer and Kornprobst, 2009, Thorpe, 1990, Thorpe and Gautrais, 1998, Rullen and Thorpe, 2001]. In this section, we collect the most important models which refer to the OPL layer, we introduce the idea hidden behind each one of them and we interpret each model through the prism of information theory. However, we think, it would be wise to first continue by providing some abstract information about the role of each one of the retina OPL cells. The following subsections are only written for encyclopedic knowledge and they do not contribute to the rest of the chapter or the manuscript. Thus, a reader not interested in neuroscience could skip them.

3.2.1.1 Photoreceptors

The photoreceptors are of two different types: rods and cones. The rods are used to detect low levels of light whereas cones are used to detect color. In practice, there are three types of cones, blue, green and red cones which are completely related to color and they filter the visual stimuli under normal light conditions.

In most of the mammalian retinas, rods outnumber cones by approximately 20-folds. As a result, rods were considered to be the basic photoreceptor type. However, more recent studies have shown that cones are much more sensitive to light and they are excited much earlier than rods do. For example human rods have been computed to receive only 1 photon per 10 minutes [Masland, 2001]. In addition, even if the number of rods is higher, cones seem to have a much more complex network of postsynaptic cells. [Baylor et al., 1974, Baylor et al., 1979, Baylor et al., 1980]. Hence, most of the neuroscientific models refer to cones when they mathematically model the photoreceptors.

3.2.1.2 Horizontal Cells

The horizontal cells receive the electrical signal by photoreceptors through chemical synapses. One photoreceptor can propagate the signal to one or more horizontal cells. In addition, each horizontal cell is strongly connected to its neighbors. In the beginning, horizontal cells were considered to contribute to an edge enhancement by sharpening the visual stimuli at its edges. Another more interesting interpretation of their role was that they adjust the gain of the retina. While horizontal cells receive an excitatory signal from photo-receptors, the feedback which is sent from horizontal cells to rods and cones inhibits them. This is like they subtract a proportional value in order to locally adapt light [Masland, 2011]. The feedforward output of horizontal cells to bipolar cells is an average signal, which is interpreted in image processing as a strong blur.

3.2.1.3 Bipolar cells

A mammalian retina contains 9-11 different kinds of cone-driven bipolar cells. Each kind has its own number and distribution of synapses, as it is called the structure which permits a neuron to transmit a chemical or an electrical signal to another neuron. Individual cells have characteristic sets of neurotransmitter receptors. When the retina is stimulated by light there is a group of bipolar cells which are hyperpolarized (OFF-cells) and another group which is depolarized (ON-cell). The OFF and ON cells are of an equal number. Bipolar cells are further subdivided into two channels: the transient (high-pass) and the sustained (low-pass). Thus, there is the ON-transient, ON sustained, OFF-transient and OFF-sustained [Masland, 2001]. Bipolar cells receive a direct signal by one or more photoreceptors and a delayed signal from horizontal cells. In information theory, these two signals are interpreted as two versions of the visual stimuli which are both blurred in a different way.

3.3 OPL Approximation Models

According to the law of optics, the 3D visual stimuli $I(\boldsymbol{X}, t)$ is projected onto the retina via the lens (the optics of the eye is detailed in [Ögmen and Herzog, 2010]). Hence, the 3D luminance $I(\boldsymbol{X}, t)$ is simplified into a 2D luminance $f(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \mathbb{R}^2$ which is the input of the OPL layer. The OPL cells receive as an input the visual stimulus $f(\boldsymbol{x}, t)$ which is spatiotemporally transformed into an electrical signal. This transformation takes place inside the *Receptive Field (RF)* of each cell. Kuffler in [Kuffler, 1952] proposed to shape the RF by two concentric nested circles or ellipses, which are termed Center-Surround (CS). The smaller circle (or ellipse) corresponds to the center and the larger one is the surround. Let $\Omega_i \subseteq \mathbb{R}^2$ be the RF of a bipolar cell centered in $\boldsymbol{x}_i \in \Omega_i$. Let $A(\boldsymbol{x}_i, t)$ be the electrical

signal produced by the RF Ω_i when the input signal is $f(\boldsymbol{x}, t)$. Considering the bipolar cell is a linear time- and shift-invariant system, the following linear approximation of the OPL retinal-transform will be introduced in [Wohrer et al., 2009]:

$$A(\boldsymbol{x}_i, t) = \int_{t'=0}^{+\infty} \int_{\boldsymbol{x}' \in \Omega_i} K(\boldsymbol{x}_i - \boldsymbol{x}', t - t') f(\boldsymbol{x}', t') d\boldsymbol{x}' dt' \tag{3.1}$$

where $K(\boldsymbol{x}, t)$ is the spatiotemporal transform of a single bipolar cell, also known as the Point Spread Function (PSF) at time t. The above equation indicates that the electrical signal $A(\boldsymbol{x}_i, t)$ depends linearly on the spatial neighborhood and the past values of the input stimuli located in the RF Ω_i of the single bipolar cell centered in $\boldsymbol{x}_i$. Assuming that i) the number of cells is very large, ii) all the cells obey to the same spatiotemporal model (spatial invariance) and iii) the temporal point spread function $K(\boldsymbol{x}, t)$ is not restricted to the domain Ω_i, the spatiotemporal transform (3.1) is approximated by spatiotemporal convolution which has been already introduced in (3.15).

Many models of the temporal point spread function $K(\boldsymbol{x}, t)$ have been proposed. The most important ones, which are introduced below, are the spatial DoG [Kuffler, 1952], the separable spatiotemporal DoG [Marr and Hildreth, 1980] and the non-separable spatiotemporal DoG [Fleet et al., 1985].

3.3.1 Spatial DoG Filter

The first mathematical approximation of the OPL was proposed by Kuffler [Kuffler, 1952]. A bipolar cell receives its input signal directly from a group of photoreceptors and/or a group of horizontal cells (Fig. 3.3). On the one hand, the output of two or more photoreceptors is averaged and transmitted to the center of the RF of the bipolar cell in order to excite it. This is approximated by a Gaussian filter $G_{\sigma_C}(\boldsymbol{x})$ given by eq. (3.3). On the other hand, the same or higher number of photoreceptors is linked to horizontal cells. A horizontal cell is strongly connected to neighbor horizontal cells averaging twice the initial input stimulus. The output of one or more horizontal cells is then propagated to the surround of the RF of the bipolar cell in order to inhibit it [Hérault and Durette, 2007]. This signal is modeled by the Gaussian filter $G_{\sigma_S}(\boldsymbol{x})$ with $\sigma_c < \sigma_s$. As a result, the bipolar cell receives two signals of opposite signs. Finally, the CS activity $K(\boldsymbol{x}, t)$ of the RF of a bipolar cell is modeled as a DoG filter:

$$DoG(\boldsymbol{x}) = G_{\sigma_C}(\boldsymbol{x}) - G_{\sigma_S}(\boldsymbol{x}), \tag{3.2}$$

$$G_\sigma(\boldsymbol{x}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}\right) = G^*_\sigma(r), \tag{3.3}$$

where $\boldsymbol{x} \in \mathbb{R}^2$, $r = \|\boldsymbol{x}\|$ is the Euclidean norm of $\boldsymbol{x}$ and σ is the standard deviation which tunes the spread of the Gaussian filter. Kuffler assumed that all these processes happen instantaneously. Hence, the PSF $K(\boldsymbol{x}, t)$ is constant for all time t.

3.3.2 Separable Spatiotemporal DoG Filter

Another attempt to improve the static DoG filter was done by Marr [Marr, 1982] [Masmoudi et al., 2013]. Marr's theory contains an assumption of spatiotemporal separability adapting a temporal impulse response for each one of the decomposition layers which are formed by a static DoG:

$$K(\boldsymbol{x}, t) = H(t) DoG(\boldsymbol{x}), \tag{3.4}$$

where $H(t)$ is usually low-pass for "sustained" units, and band-pass for "transient" units. The goal of Marr's temporal function was to build a multiscale bank of DoG filters each one of which will have a resolution defined by $H(t)$.

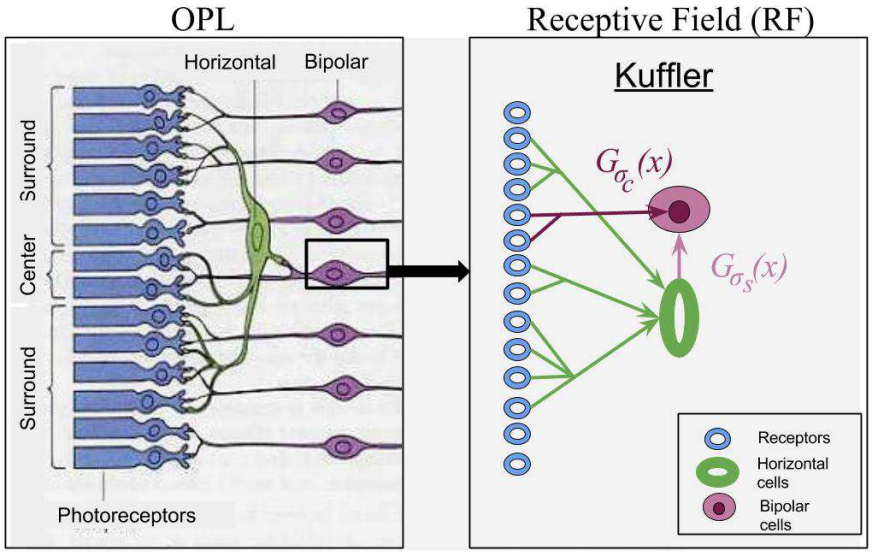


Figure 3.3: Propagation of the electrical signal to the receptive field of a bipolar cell according to [Kuffler, 1952]. The left figure is part of the retinal structure which corresponds to the OPL layer. The right figure is focused on a single bipolar cell and how the RF of this cell works.

3.3.3 Non-Separable Spatiotemporal Receptive Field

The models above were not accurate with time-varying stimuli $f(\boldsymbol{x}, t)$. Hence, Fleet [Fleet et al., 1985] proposed a non-separable spatiotemporal CS model as an extension of the DoG. Electrophysiological studies have shown that the center and the surround have different time courses of response. In addition, the temporal delay between the response of the center and surround areas of the cells receptive field should also be considered. This model was interpreted by Fleet as a precursor to the extraction of velocity specific information [Fleet et al., 1985].

The inseparability of space and time was highlighted and confirmed while studying the spectrum of the CS model. The dynamics of the model are due to the sensitivity changes of the response of the cells and the different phases of the CS areas [Fleet et al., 1985]. Similar models have been proposed in order to describe how does the receptive field of neurons work in areas which come after the retina like the Lateral Geniculate Nucleus (LGN) or cortical neurons [Wohrer et al., 2009, DeAngelis et al., 1993, D. Cai and Freeman, 1997]. The common point of all these models is the spatiotemporal inseparability, which confirms its importance.

A non-separable spatiotemporal retinal filtering as part of the visual system coding process is mathematically introduced as:

$$K(\boldsymbol{x}, t) = C(\boldsymbol{x}, t) - S(\boldsymbol{x}, t), \tag{3.5}$$

$$C(\boldsymbol{x}, t) = w_c G_{\sigma_C}(\boldsymbol{x}) H_c(t), \tag{3.6}$$

$$S(\boldsymbol{x}, t) = w_s G_{\sigma_S}(\boldsymbol{x}) H_s(t), \tag{3.7}$$

$$H_s(t) = \left(H_c \overset{t}{*} E_{\tau_S}\right)(t) \tag{3.8}$$

To explain the non-separable spatiotemporal retina-inspired filter, we need to focus again on bipolar cells. We have already mentioned how they receive two opposite signals in their RF. In this model the key is the hierarchy of the retinal cells and their connectivity. The temporal behavior of the retina cells is described in Fig. 3.4. The horizontal cells are

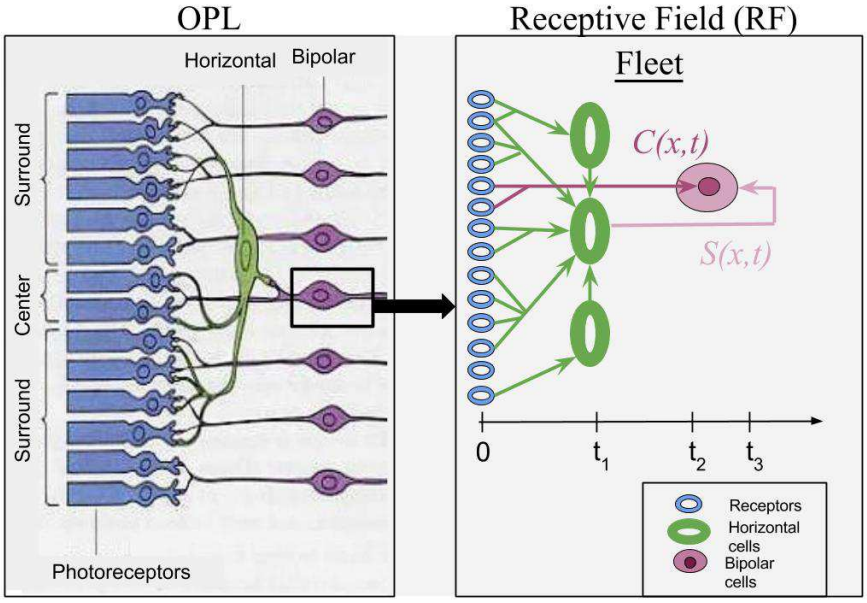


Figure 3.4: Propagation of the electrical signal to the receptive field of a bipolar cell according to Fleet [Fleet et al., 1985].

the first ones which receive an average signal by a group of photoreceptors at time t_1. A smaller group of photoreceptors propagates an excitatory signal to the center of bipolar cells $C(\boldsymbol{x}, t)$ at time t_2 while horizontal cells receive this signal and they communicate and exchange information with adjacent horizontal cells. This causes a small delay $E_{\tau_s}(t)$ until time t_3 in the propagation of the inhibitory signal $S(\boldsymbol{x}, t)$ coming from horizontal cells to bipolar cells.

3.3.4 Non-Separable Spatiotemporal Filter

The OPL layer describes the retina filtering as a non-separable spatiotemporal transform $K(\boldsymbol{x}, t)$:

$$K(\boldsymbol{x}, t) = C(\boldsymbol{x}, t) - S(\boldsymbol{x}, t), \tag{3.9}$$

$$C(\boldsymbol{x}, t) = w_c G_{\sigma_C}(\boldsymbol{x}) W(t), \tag{3.10}$$

$$S(\boldsymbol{x}, t) = w_s G_{\sigma_S}(\boldsymbol{x}) \left(W \overset{t}{*} E_{\tau_S} \right)(t), \tag{3.11}$$

where $C(\boldsymbol{x}, t)$ stands for the center and $S(\boldsymbol{x}, t)$ for the surround of the structure of the RF which is totally linked to the way the photoreceptors and the horizontal retinal cells are connected and propagate the stimuli, w_c and w_s are constant parameters, $G_{\sigma_C}(\boldsymbol{x})$ and $G_{\sigma_S}(\boldsymbol{x})$ are spatial Gaussian filters (see eq (3.3)) standing for the center (photoreceptors) and surround (horizontals) areas respectively, $W(t)$ is a low-pass filter and $E_{\tau_S}(t)$ is an exponential temporal filter.

The temporal filter $W(t)$ is given by (3.13) and describes the *Difference of Exponential (DoE)* which stands for the spatial variation with respect to time. It is modeled with temporal low-pass filters [Wohrer and Kornprobst, 2009]:

$$W(t) = \left(E_{\tau_{G,n}} \overset{t}{*} W_c \right)(t), \tag{3.12}$$

$$W_c(t) = \begin{cases} \delta_0 - w_c E_{\tau_C}(t), & \text{if } t \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.13}$$

where the gamma temporal filter $E_{\tau_G,n}(t)$ is defined by:

$$E_{\tau,n}(t) = \begin{cases} \dfrac{t^n \exp(-t/\tau)}{\tau^{n+1}} & \text{if } t \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{3.14}$$

with $n \in \mathbb{N}$, $\tau > 0$, $\delta_0(t)$ is the dirac function, $E_{\tau_C}(t)$ is an exponential temporal filter, and $\overset{t}{*}$ stands for the temporal convolution. The exponential temporal filter is given by (4.16) for $n = 0$.

The OPL transform $K(\boldsymbol{x}, t)$ is applied to the input signal $f(\boldsymbol{x}, t)$ resulting in the activation degree which is a continuous current $I_{OPL}(\boldsymbol{x}, t)$ which is the output of the OPL layer.

$$\begin{aligned} I_{\text{OPL}}(\boldsymbol{x}, t) &= \int_{t' \in \mathbb{R}} \int_{\boldsymbol{x}' \in \mathbb{R}^2} K(\boldsymbol{x} - \boldsymbol{x}', t - t') f(\boldsymbol{x}', t') d\boldsymbol{x}' dt' \\ &= (K \overset{x,t}{*} f)(\boldsymbol{x}, t), \end{aligned} \tag{3.15}$$

where $\overset{x,t}{*}$ is the spatiotemporal convolution between the input signal and the OPL filter.

3.4 DoG in Image Processing

The DoG filter which is used as a spatial transform in order to approximate the CS structure of the OPL cells is very well-known also in image processing. Burt and Adelson [Burt and Adelson, 1983] proposed the Gaussian pyramid for the analysis/synthesis of an image. The notion of pyramids is well known in image processing community as multiresolution techniques which produce several copies of the input signal and enable to decrease the sample density and resolution of the input image in regular steps [Adelson et al., 1984]. The Gaussian pyramid was the first one which was proposed to support an efficient scaled convolution and reduced the image representation. An input image $f(\boldsymbol{x})$ which is the 1^{st} layer K_1 of the Gaussian pyramid is filtered by a Gaussian Kernel $G(\boldsymbol{x})$. The filtered image $K_2(\boldsymbol{x}) = (G \overset{x}{*} f)(\boldsymbol{x})$ is the 2^{nd} layer of the Gaussian pyramid which is downsampled and filtered again with the same Gaussian kernel $G(\boldsymbol{x})$ to generate the 3^{rd} layer, etc (see Fig. 3.5). The Gaussian pyramid was extended into the Laplacian pyramid which is more efficient in terms of compression [Adelson et al., 1984]. The Laplacian pyramid generates its layers by subtracting every two layers of the Gaussian pyramid (see Fig. 3.5). The key point of the Laplacian pyramid is that it is an invertible transform which allows to perfectly reconstruct the input signal only by using its decomposition layers. The spatial DoG pyramid of Thorpe is similar to a Laplacian pyramid and it was used in his ROC encoder as a filter bank which approximates the OPL transform. However, seeking for retina-inspired transforms in order to be used in compression, Thorpe's filter bank has two important drawbacks: first of all, it is a very rough approximation of the dynamic retina transform. In addition, his DoG pyramid is not invertible which is necessary in compression.

3.4.1 Spatial DoG Pyramid

Thorpe proposed a pyramid of DoGs as an OPL approximation in order to roughly mimic the dynamic behavior of photoreceptors and horizontal cells before they reach bipolar cells [Thorpe, 1990]. This filter bank was proposed within the framework of the Rank Order Coder (ROC) which is going to be detailed in chapter 6, as a multiresolution spatial transform [Thorpe, 1990, Thorpe and Gautrais, 1998, Rullen and Thorpe, 2001, Thorpe et al., 2001], :

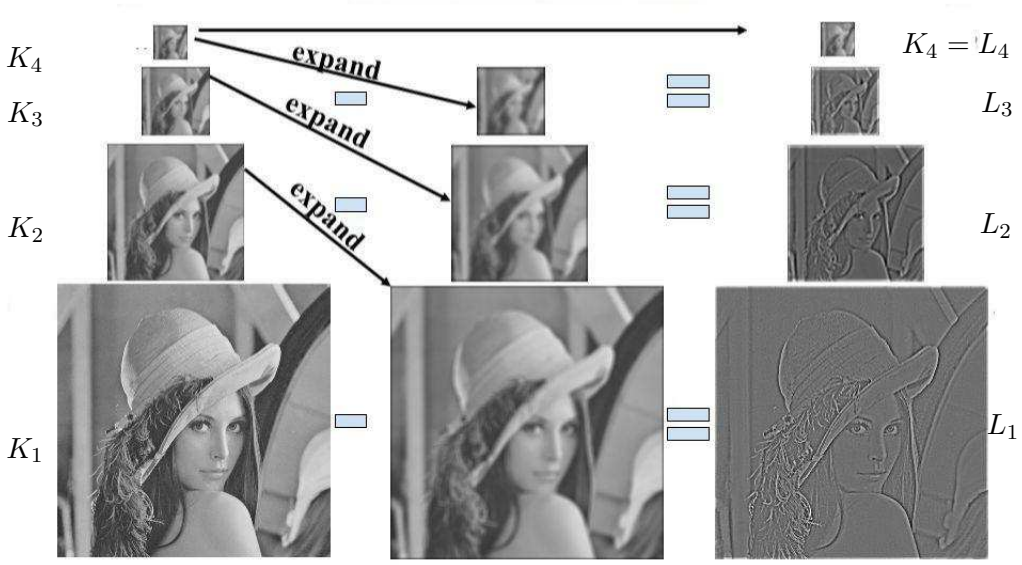


Figure 3.5: Gaussian Pyramid vs Laplacian Pyramid.

$$\mathrm{DoG}^k(\boldsymbol{x}) = G_{\sigma_c^k}(\boldsymbol{x}) - G_{\sigma_s^k}(\boldsymbol{x}), \tag{3.16}$$

where $G_{\sigma_c^k}(\boldsymbol{x})$, $G_{\sigma_s^k}(\boldsymbol{x})$ are two Gaussian filters with standard deviations σ_c^k and σ_s^k respectively such that $\sigma_s^1 = 3\sigma_c^1$, $\sigma_c^{k+1} = 0.5\,\sigma_c^k$, $\sigma_s^{k+1} = 0.5\,\sigma_s^k$ and $\sigma_c^K = 0.5$ pixels. Each filter has a size of $(2M_k + 1)^2$ with $M_k = 3\sigma_c^k$.

3.4.2 Invertible Spatial DoG Pyramid

Masmoudi was the first one who tackled Thorpe's problems in [Masmoudi et al., 2012]. He improved Thorpe's filter bank by proposing a rectification function to achieve a perfect reconstruction. This function was nothing more than a Gaussian filter considered as the zero layer of the transform pyramid, $DoG^0(x) = G_{\sigma_c^0}(x)$ [Masmoudi et al., 2012]. This rectification function was necessary to obtain a Laplacian-like multiscale and invertible filter bank which leads to a perfect reconstruction [Masmoudi et al., 2012, Kovacevic and Chebina, 2008]. Similar method was also used in [Perrinet et al., 2004] to achieve a reconstructable version of the Thorpe's filter. Masmoudi mathematically proved that his rectified DoG pyramid is a frame based on frame theory (see section 5.3.1). This property is necessary in compression algorithms.

3.4.3 Invertible Spatiotemporal DoG Pyramid

Masmoudi *et al* proposed another filter which was separable spatiotemporally combining Marr's and Thorpe's models. In more details, they represented the spatial CS structure using the DoG filter (3.2). In addition, they modeled the ability of retina to gradually compress all the details with respect to time using the DoG^k pyramid (see eq. 3.16) [Thorpe, 1990]. In this way, Masmoudi ensured the dynamic behavior of his filter introducing a time-delay function D_{t_k}. The value of t_k is an increasing function of k and it corresponds to the delay each subband DoG^k appears. In [Masmoudi et al., 2012] this function was proposed to be linearly increasing while in [Masmoudi et al., 2013] it was

proposed to be an exponential function. The activation of each subband was denoted as:

$$K(\boldsymbol{x}, t) = \mathrm{DoG}^k(\boldsymbol{x})\mathbf{1}_{[t \geq t_k]}(t), \tag{3.17}$$

where $\mathbb{1}_{(t \geq t_k)}$ is the indicator function. The above equation describes that each scale k exists only when $t \geq t_k$ ($\mathbb{1}_{(t \geq t_k)}(t) = 1$), otherwise there is no information transmitted ($\mathbb{1}_{(t \leq t_k)}(t) = 0$).

The pyramid of the difference of Gaussians, DoG^k, is applied to an input image $f(\boldsymbol{x})$. This spatial convolution is the activation degree which is similar to what the filter bank of Thorpe is his ROC model (see section 3.4.1).

$$A^k(\boldsymbol{x}) = DoG^k(\boldsymbol{x}) \overset{\boldsymbol{x}}{*} f(\boldsymbol{x}). \tag{3.18}$$

Masmoudi *et al* proposed that each scale k, is activated in a predefined time t_k, resulting that between two adjacent scales, k and $k+1$, there is a time delay $t_{delay} = t_{k+1} - t_k$. The time dependency is an increasing exponential function which is added to the coder/decoder by the following equation:

$$A^k(\boldsymbol{x}, t) = A^k(\boldsymbol{x})\mathbb{1}_{(t \geq t_k)}(t), \tag{3.19}$$

where $\mathbb{1}_{(t \geq t_k)}$ is the indicator function. The above equation describes that each scale k exists only when $t \geq t_k$ ($\mathbb{1}_{(t \geq t_k)}(t) = 1$), otherwise there is no information transmitted ($\mathbb{1}_{(t \leq t_k)}(t) = 0$).

The authors reconstructed the input stimulus $\tilde{f}(\boldsymbol{x})$ using the frame theory [Kovacevic and Chebina, 2008]:

$$\tilde{f}(\boldsymbol{x}) = \sum \tilde{A}^k(\boldsymbol{x}, t) \overset{x}{*} \widetilde{DoG}^k(\boldsymbol{x}). \tag{3.20}$$

where $\widetilde{DoG}^k$ is the dual of DoG^k. In more details, Masmoudi et al indicated that using **dual frames** it is able to progressively reconstruct an approximation of the input image [Masmoudi et al., 2012, Masmoudi et al., 2013]. As a result, dual frames make the model well conditioned and invertible.

3.5 Conclusion

This chapter introduced neuroscientific models which have been proposed to approximate the OPL retina transform. All these models are based on the DoG filter which is considered to precisely describe the CS structure of the neural RF. Initially, these models were static (spatial or separable spatiotemporal filters) until scientist realized that the CS structure dynamically transforms the visual stimuli (non-separable spatiotemporal filter). Thorpe was the first one who adopted some of the static neuroscientific models in his compression algorithm. However, in coding schemes (see Fig. 2.20 (a)), it is necessary to be proven that the transformation which is in use is invertible in order to ensure the reconstruction of the input signal. Thus, Thorpe needed to give some extra efforts not only to efficiently adapt the neuroscientific models to their systems but also to prove that they are invertible.

Thorpe's filter was mathematically proven to be invertible by Masmoudi but Masmoudi's filter was still not reliable enough. Although, Masmoudi tried to insert time into his filter proving that this time dependency does not influence the inversion of the filter, his filter bank still lacks of dynamicity especially comparing to models of the non-separable spatiotemporal RF and non-separable spatiotemporal filter. This dynamic transform of a bank of DoG filters which evolve in time is novel in signal processing community. Thus, in chapter 4 we are going to introduce a retina-inspired filter which is based on Wohrer's dynamic filter and in chapter 5 we will prove that this filter is invertible.

Chapter 4

Retina-inspired Filtering

Contents

4.1 Introduction

In chapter 3, we had a discussion about static and dynamic neuroscientific models which approximate the CS structure of the OPL retina layer. Being motivated by these models or the combination of these models people proposed bio-inspired filter which have been famous in image processing like the Gaussian pyramid, the Laplacian pyramid or more recently released filters like the separable spatiotemporal DoG pyramid proposed by Masmoudi in [Masmoudi et al., 2012]. We noticed that even though in [Masmoudi et al., 2012] the authors tried to mimic the dynamic OPL filter as proposed by Wohrer in [Wohrer and Kornprobst, 2009], their model was not dynamic and it could be further improved.

In this chapter, we aim to propose another more accurate simplification of OPL retina transform which allows us taking the advantage of its dynamicity. At the same time, our novel non-separable spatiotemporal OPL retina-inspired filter, which is also termed as retina-inspired filter, is easier to be analyzed and studied comparing to the original OPL transform and it also is easier to be proven as invertible.

In more details, we show that the retina-inspired filter is interpreted as a group of time varying Weighted Difference of Gaussians (WDoG) filters. That means that, at each time there is a new and different spatial WDoG filter which arises. Consequently, while time increases, the retina-inspired filter is able to extract different kinds of information from the input signal. We study the behavior of the WDoG bank in spatial and frequential domain and we prove that the bandwidth of this filter evolves in time which confirms the initial interpretation. We first represent some numerical results of this transform applied to 1D

signal which is straightforward and then we extend this to still-images and images retrieved from video streams, which are called in this document pictures.

Of course a reasonable question one could ask himself is what is the advantage of applying a dynamic filter to a still image. First of all, this is the first attempt to study and adopt a retina-inspired dynamic filter in signal processing. As a result, the simpler the input signal, the better the analysis of the impact of the filter. Secondly, the dynamic processing of real retinas exists even for still images. If one neglects the eyes' movements and just considers the human visual perception in the presence of a constant view, it is true that during the very first few milliseconds the scene seems blurry while more details enrich the clearanceness of the scene in time.

4.2 Definition

A still image does not vary in time when it is flashed. We assume that the visual stimulus is flashed for a given time $T > 0$ and it is constant during this time interval. This involves that the 2D visual stimulus is written as $f(\boldsymbol{x}, t) = f(\boldsymbol{x})\mathbb{1}_{[0,T]}(t)$ where $f(\boldsymbol{x}) \in L^1(\mathbb{R}^2)$ is a still image and $\mathbb{1}$ is the indicator function such that $\mathbb{1}_{[0,T]}(t) = 1$ if $0 \leq t \leq T$, otherwise 0. The space $L^1(\mathbb{R}^2)$ is the set of the Lebesgue integrable functions from $\mathbb{R}^2$ to $\mathbb{R}$. This time invariance enables us to simplify the spatiotemporal convolution (see eq. 3.15) of the OPL transform (introduced in section 3.3.4) as established in Proposition 1.

Proposition 1. *Assume $f(\boldsymbol{x}, t) = f(\boldsymbol{x})\mathbb{1}_{[0,T]}(t)$ for all $\boldsymbol{x} \in \mathbb{R}^2$ and all $t \in \mathbb{R}$. Then, the spatiotemporal convolution (3.15) turns into the spatial convolution:*

$$A(\boldsymbol{x}, t) = \phi(\boldsymbol{x}, t) \overset{x}{*} f(\boldsymbol{x}), \tag{4.1}$$

where $\phi(\boldsymbol{x}, t)$ is a spatiotemporal WDoG filter weighted by two temporal filters $R_c(t)$ and $R_s(t)$:

$$\phi(\boldsymbol{x}, t) = \begin{cases} w_c R_c(t) G_{\sigma_C}(\boldsymbol{x}) - w_s R_s(t) G_{\sigma_S}(\boldsymbol{x}) & \text{if } t \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{4.2}$$

$$R_c(t) = \mathbb{1}_{[0,+\infty)}(t) \int_{\max\{0,t-T\}}^{t} W(u)du, \tag{4.3}$$

$$R_s(t) = \mathbb{1}_{[0,+\infty)}(t) \int_{\max\{0,t-T\}}^{t} (W \overset{t}{*} E_{\tau_S})(u)du, \tag{4.4}$$

for all $\boldsymbol{x} \in \mathbb{R}^2$ and for all $t \in \mathbb{R}$.

Proof. The proof consists in calculating the convolution:

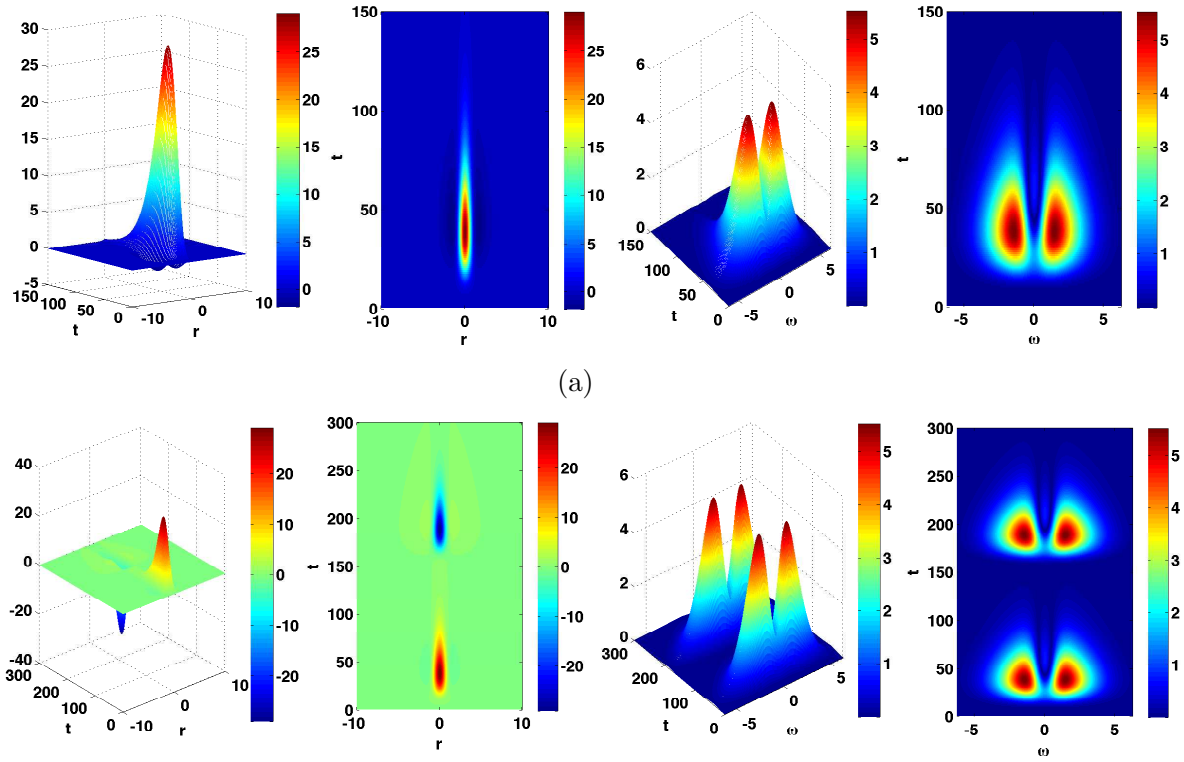
$$\begin{aligned} A(\boldsymbol{x}, t) &= K(\boldsymbol{x}, t) \overset{x,t}{*} f(\boldsymbol{x}, t) \\ &= (C(\boldsymbol{x}, t) - S(\boldsymbol{x}, t)) \overset{x,t}{*} f(\boldsymbol{x})\mathbb{1}_{[0,T]}(t) \\ &= w_c G_{\sigma_C}(\boldsymbol{x}) \overset{x}{*} f(\boldsymbol{x}) \left(W \overset{t}{*} \mathbb{1}_{[0,T]}(t)\right) \\ &\quad - w_s G_{\sigma_S}(\boldsymbol{x}) \overset{x}{*} f(\boldsymbol{x}) \left(\left(W \overset{t}{*} E_{\tau_S}\right) \overset{t}{*} \mathbb{1}_{[0,T]}(t)\right). \end{aligned}$$

For an integrable function $U(t)$, a short calculation shows that

$$U \overset{t}{*} \mathbb{1}_{[0,T]}(t) = \begin{cases} 0, & \text{if } t < 0, \\ \displaystyle\int_{\max\{0,t-T\}}^{t} U(u)du, & \text{otherwise.} \end{cases} \tag{4.5}$$

Then, it is straighforward to derive (4.1) with $\phi(\boldsymbol{x}, t)$ defined in (4.2). □

This proposition is fundamental because it turns the spatio-temporal filtering of the still image $f(\boldsymbol{x})$ with the retinal filter $K(\boldsymbol{x}, t)$ into a simpler spatial convolution with the retina-inspired filter $\phi(\boldsymbol{x}, t)$ which is a group of time-varying WDoGs. The temporal functions $R_c(t)$ and $R_s(t)$ act like weights that modify the WDoG spatial spectrum with respect to time (see Section 4.4). Finally, the retina-inspired filter is built by the difference of two separable filters which evolve in time mimicking the non-separable behavior of the OPL layer. A DoG filter is an isotropic function due to the property of the circular symmetry. As a results, for a fixed time $t \in \mathbb{R}$, the retina-inspired filter $\phi(\boldsymbol{x}, t)$ is spatially isotropic. Thus, it can be simplified by $\phi(r, t)$ where the radius r is the norm of $\boldsymbol{x}$. Fig. 4.1 plots the retinal filter $\phi(r, t)$ as a function of $r = \|\boldsymbol{x}\|_2$ for two different cases related to the temporal filters $R_c(t)$ and $R_s(t)$ which are studied in the following subsection. For simplicity, it is assumed that $r \in \mathbb{R}$ and $\phi(r, t)$ is symmetric around $r = 0$ for all t. The parameters have been tuned according to neuroscientific results [Masmoudi et al., 2012] which approximate the retinal spectrum and the speed of the retinal processing. The spectrum of a DoG filter is also a DoG. As a result, the spectrum of the retina-inspired filter is also a spatially isotropic function. This spectrum which is denoted $\hat{\phi}(\omega, t)$, where ω is the spatial angular frequency related to r, is shown in Fig. 4.1.



(a)

(b)

Figure 4.1: (Left to right) The retina-inspired filter $\phi(r, t)$ as a function of $r \in \mathbb{R}$ and $t \in \mathbb{R}$. The top view of $\phi(r, t)$. The retina-inspired filter spectrum $\hat{\phi}(\omega, t)$. The top-view of $|\hat{\phi}(\omega, t)|$. (a) The top line corresponds to the case $T = +\infty$. (b) The bottom line stands for the case $T < +\infty$ (Parameters for (a) and (b): $T = 150$ ms, $\tau_C = 20$ ms, $\tau_S = 4$ ms, $\tau_G = 5$ ms, $n = 5$, $w_S = 1$, $w_C = 1$, $\sigma_c = 0.5$ and $\sigma_s = 3\sigma_c$).

4.3 Spatiotemporal Behavior and Convergence

We are interested in studying the two temporal filters $R_c(t)$ and $R_s(t)$ which are responsible for the spatiotemporal behavior of the retina-inspired filter. First of all, we calculate their

closed-form model in Proposition 2. This proposition is based on the following lemma.

Lemma 1. Assume $t \geq 0$, then

$$\begin{aligned} J_c(t) &= \int_0^t W(u)du \\ &= P_n(t)\exp\left(\frac{-t}{\tau_G}\right) + \alpha_c \exp\left(\frac{-t}{\tau_C}\right) + \gamma_c, \end{aligned} \tag{4.6}$$

where $P_n(t)$ is a polynomial function in t of order n and α_c, γ_c are two reals, and

$$\begin{aligned} J_s(t) &= \int_0^t (W \overset{t}{*} E_{\tau_S})(u)du \\ &= Q_n(t)\exp\left(\frac{-t}{\tau_G}\right) + \alpha_s \exp\left(\frac{-t}{\tau_S}\right) + \beta_s \exp\left(\frac{-t}{\tau_C}\right) + \gamma_s, \end{aligned} \tag{4.7}$$

where $Q_n(t)$ is a polynomial function in t of order n and α_s, β_s and γ_s are some reals.

Proof. See the Appendix A. □

Proposition 2. *The temporal weights $R_c(t)$ in (4.3) and $R_s(t)$ in (4.4) satisfy:*

$$R_c(t) = \begin{cases} J_c(t) & \text{if } 0 \leq t \leq T, \\ J_c(t) - J_c(t-T) & \text{if } T < t, \end{cases} \tag{4.8}$$

$$R_s(t) = \begin{cases} J_s(t) & \text{if } 0 \leq t \leq T, \\ J_s(t) - J_s(t-T) & \text{if } T < t, \end{cases} \tag{4.9}$$

where $J_c(t)$ is given in (4.6) and $J_s(t)$ is given in (4.7).

Proof. The proof is based on the fact that $R_c(t) = J_c(t)$ for $0 \leq t \leq T$ and

$$R_c(t) = \int_{t-T}^{t} W(u)du = \int_0^t W(u)du - \int_0^{t-T} W(u)du$$

for $t > T$. Lemma 1 is used to deduce (4.8). The same equalities hold for $R_s(t)$. □

The temporal weights $R_c(t)$ and $R_s(t)$ are illustrated in Fig. 4.2. The parameters in (a) and (b) have been tuned according to the parameters of Fig. 4.1 for $T = +\infty$ and $T < +\infty$ respectively. We should note that their shapes are very similar except that the surround temporal filter $R_s(t)$ appears with a small delay $E_{\tau_S}(t)$ with respect to the center one. There is a high impact of the above characteristic on the spatiotemporal evolution of the filter. The delay $E_{\tau_S}(t)$ is crucially important because for the very first few milliseconds, while $R_s(t)$ does not yet exist, the second term of the WDoG is zero. As a result, at the very beginning, the retina-inspired filter is a pure Gaussian with a very low amplitude since it is weighted by $R_c(t)$. Finally, we can note that $R_c(t)$ and $R_s(t)$ converge to a constant value. Hence, $\phi(\boldsymbol{x}, t)$ also converges as $t \to +\infty$. This convergence is established in the following proposition.

Proposition 3. *The filter $\phi(\boldsymbol{x}, t)$ is a continuous and infinitely differential function over $\mathbb{R}^2 \times \mathbb{R}$ such that $\phi(\boldsymbol{x}, 0) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^2$. If $T = +\infty$, then $\phi(\boldsymbol{x}, t)$ converges uniformly toward $\phi(\boldsymbol{x})$ where $\phi(\boldsymbol{x})$ is the WDoG filter:*

$$\phi(\boldsymbol{x}) = w_c\gamma_c G_{\sigma_C}(\boldsymbol{x}) - w_s\gamma_s G_{\sigma_S}(\boldsymbol{x}), \tag{4.10}$$

with γ_c and γ_s defined in (4.6)-(4.7). If $T < +\infty$, the filter vanishes uniformly as $t \to +\infty$:

$$\lim_{t \to +\infty} \sup_{\boldsymbol{x} \in \mathbb{R}^2} \mid \phi(\boldsymbol{x}, t) \mid = 0. \tag{4.11}$$

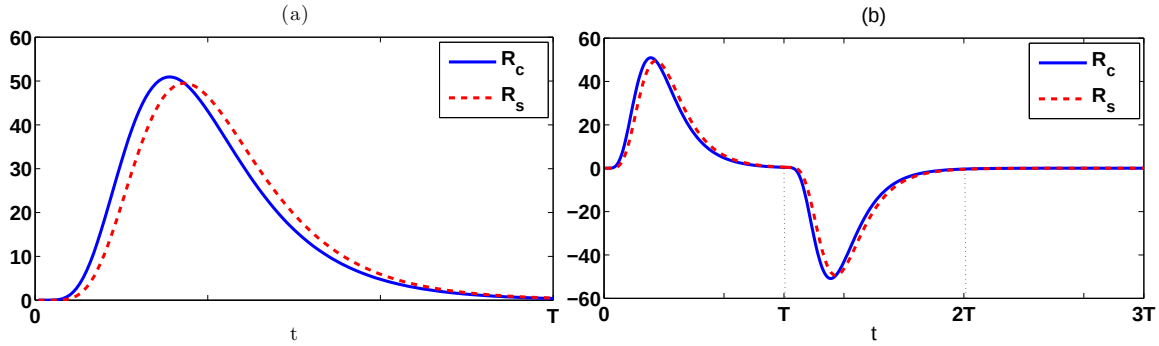


Figure 4.2: Temporal filters $R_c(t)$ and $R_s(t)$. The convergence of the filters depends on the value of time T. Subplot (a) corresponds to the case $T = +\infty$, while subplot (b) corresponds to the case $T < +\infty$ (Parameters: $T = 150$ ms, $\tau_C = 20$ ms, $\tau_S = 4$ ms, $\tau_G = 5$ ms, $n = 5$).

Proof. The uniform convergence (4.11) results from the definition of $\phi(\boldsymbol{x}, t)$ in (4.2) since $G_{\sigma_C}(\boldsymbol{x})$ and $G_{\sigma_S}(\boldsymbol{x})$ are bounded and $R_c(t)$ and $R_s(t)$ converge to 0 according to Proposition 2 and Lemma 1. When $T = +\infty$ and $t \to +\infty$, $R_c(t)$ and $R_s(t)$ converge to γ_c and γ_s respectively. Hence, $\phi(\boldsymbol{x}, t)$ converges uniformly toward $\phi(\boldsymbol{x})$. □

Figure 4.1 depicts the two different cases of Proposition 3. A brief discussion about this proposition is that, while time T increases to infinity, the retina-inspired filter turns into a static WDoG. This is totally concurrent with the neuroscientific assumptions about the time limits of the visual system. Neuroscientists have proposed that the objects categorization of a single image which is propagated from the retina to the brain needs approximately 100 ms [Liu et al., 2009] before the next image is processed. Recent studies have proven that a simple comprehension lasts approximately 13 ms [Potter et al., 2014]. In any case, there exists a time t_c when, even if the photoreceptors will continue capturing the same signal, all the necessary information which needs to be processed has already been transmitted to the brain. From a theoretical point of view, the existence of t_c is deduced from the uniform convergence established in Proposition 3. A sequence of functions $g_n, n = 1, 2, 3, \ldots$ is said to be uniformly convergent to g for a set E of values of x if, for each $\varepsilon > 0$, an integer N can be found such that $|g_n(\boldsymbol{x}) - g(\boldsymbol{x})| < \varepsilon$ for $n \geq N$ and all $\boldsymbol{x} \in R$. In fact, given $\varepsilon > 0$, the time $t_c = t_c(\varepsilon)$ can be defined when the uniform convergence is achieved up to ε. As a result, the retina-inspired filter $\phi(\boldsymbol{x}, t)$ converges to $\phi(\boldsymbol{x})$ if $|\phi(\boldsymbol{x}, t_\varepsilon) - \phi(\boldsymbol{x})| < \varepsilon$.

4.4 Weighted DoG Analysis

This section aims to study the WDoG filter, i.e., to approximate its spatial and frequential response. Without any loss of generality, a WDoG is defined by:

$$\varphi(\boldsymbol{x}) = aG_{\sigma_a}(\boldsymbol{x}) - bG_{\sigma_b}(\boldsymbol{x}) \tag{4.12}$$

where $a, b > 0$ and $\sigma_b^2 > \sigma_a^2$. The retina-inspired filter $\phi(\boldsymbol{x}, t)$ consists of a group of WDoG with the coefficients

$$a = w_c R_c(t) = a(t),\ b = w_s R_s(t) = b(t), \tag{4.13}$$

that are time dependent, $\sigma_a = \sigma_c$ and $\sigma_b = \sigma_s$. Since the WDoG is symmetric, we define the WDoG according to the radial coordinate $r = \|\boldsymbol{x}\|_2$:

$$\varphi(r) = aG_{\sigma_a}(r) - bG_{\sigma_b}(r). \tag{4.14}$$

Since $\varphi(r)$ is symmetric around 0, we assume that $r \in \mathbb{R}$ (and not only to $\mathbb{R}^+$) to ensure a better legibility of the results and to study more easily the spectrum of the WDoG.

According to the couple (a, b), the WDoG has eigth shapes which are depicted in Fig. 4.3.

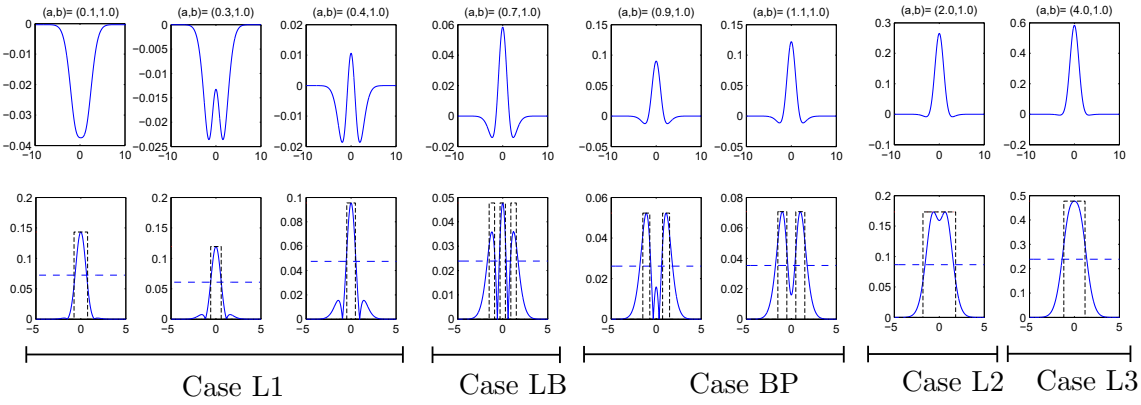


Figure 4.3: Eight typical shapes of the WDoG $\varphi(r)$ (first row) and the WDoG spectrum modulus $|\hat{\varphi}(\omega)|$ (second row) w.r.t. some values of the couple (a, b). The dotted line represents the half of the maximum value of the spectrum modulus. The rectangles in dashed line represent the approximated bandwidth of the spectrum. There are five specific cases: 1) lowpass L1, 2) lowpass/bandpass LB, 3) bandpass BP, 4) lowpass L2 and 5) lowpass L3.

The parameter $b = 1$ is fixed. According to the value of $a \in \{0.1, 0.3, 0.4, 0.7, 0.9, 1.1, 2, 4\}$, the WDoG is either a lowpass filter (cases L1, L2 or L3) or a bandpass filter (case BP) or a mixed lowpass/bandpass filter (case LB). The conventional DoG filter ($a = b = 1$) is a special case of BP. When a is changing, the bandwidth is also changing but the WDoG is always constrained to one of these eight shapes, corresponding to five behaviors. This section studies these behaviors in the space domain and in the frequency domain.

4.4.1 WDoG in Space Domain

Let us study the variations of $\varphi(r)$. The first derivative of the WDoG, which is differentiable for all $r \in \mathbb{R}$, is given by

$$\varphi'(r) = r\left(-\frac{a}{\sigma_a^2}G_{\sigma_a}(r) + \frac{b}{\sigma_b^2}G_{\sigma_b}(r)\right). \tag{4.15}$$

Let

$$\gamma = \gamma(a, b) = \frac{b\sigma_a^4}{a\sigma_b^4}. \tag{4.16}$$

A short analysis of the roots of $\varphi'(r)$ shows that two cases occur: i) if $\gamma \geq 1$, there is only one root $r = 0$, ii) if $\gamma < 1$, there are three roots $r = 0$, $r_1 > 0$ and $-r_1$ with

$$r_1 = \sigma_a\sigma_b\sqrt{\frac{2\ln(\gamma)}{\sigma_b^2 - \sigma_a^2}}. \tag{4.17}$$

As an illustration, case i) corresponds to the first left curve in Fig. 4.3 and case ii) corresponds to all the other curves. It is then easy to determine the positive intervals and negative intervals of $\varphi'(r)$ and, hence, to determine when $\varphi(r)$ is increasing or decreasing.

Proposition 4. *If $\gamma \geq 1$, $\varphi(r)$ is negative for all $r \in \mathbb{R}$ and its minimum is*

$$\varphi_0 = \min_{r\in\mathbb{R}} \varphi(r) = \varphi(0) = \frac{a\sigma_b^2 - b\sigma_a^2}{2\pi\sigma_c^2\sigma_s^2} < 0. \tag{4.18}$$

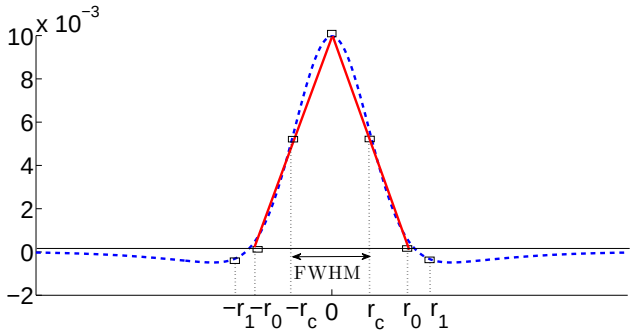


Figure 4.4: Approximate FWHM interval of a WDoG based on a triangle approximation.

Otherwise, if $\gamma < 1$*, then* $\varphi(r)$ *is negative and decreasing over* $[-\infty, -r_1]$*, increasing over* $[-r_1, 0]$*, decreasing over* $[0, r_1]$ *and finally increasing and negative over* $[r_1, +\infty]$*. The global minima are* $-r_1$ *and* r_1*.*

Proof. The proof is straightforward by studying the sign of $\varphi'(r)$ over the intervals given in the proposition. □

The Full Width Half Maximum (FWHM) response of the symmetric WDoG is given by the interval $[-r_C, r_C]$ where r_C satisfies:

$$\varphi(r_C) = \frac{\max_{r\in\mathbb{R}} \varphi(r)}{2}. \tag{4.19}$$

The FWHM is not relevant for case L1 (see Fig. 4.3) because the shape of the WDoG is significantly different from a peak. The FWHM is relevant for the other cases when the peak at $r = 0$ is sufficiently large, i.e., when $\varphi(0) > -2\varphi(r_1)$. The exact calculation of FWHM is difficult. Hence, we approximate r_C in (4.19) using the following method which is proposed in [Birch et al., 2010]. We first calculate the maximum response. Then, we compute a straight line from each maximum response to the first intercept with the r-axis (see Fig. 4.4 red solid line). We compute half of this line and we approximate its correspondent value. The resulting FWHM is illustrated in Fig. 4.4. A straightforward calculation shows that the first intercept of $\varphi(r)$ with the r-axis is

$$r_0 = \sigma_a\sigma_b\sqrt{\frac{2\ln\left(\frac{a\sigma_b^2}{b\sigma_a^2}\right)}{\sigma_b^2 - \sigma_a^2}}. \tag{4.20}$$

where $b\sigma_a^2 > a\sigma_b^2$ since $\varphi(0) > 0$. It follows that $r_C \approx \frac{r_0}{2}$ by using the triangle approximation.

4.4.2 WDoG in Frequency Domain

The Fourier transform of the WDoG in (4.14) is

$$\hat{\varphi}(\omega) = a\hat{G}_{\sigma_a}(\omega) - b\hat{G}_{\sigma_b}(\omega) \tag{4.21}$$

where

$$\hat{G}_\sigma(\omega) = \frac{1}{2\pi}\exp\left(-\frac{\omega^2\sigma^2}{2}\right), \tag{4.22}$$

and $\omega \in \mathbb{R}$ denotes the spatial angular frequency associated to r. The WDoG bandwidth $B = B(a, b)$ refers to the frequency range in which the spectrum $\hat{\varphi}(\omega)$ is above a threshold

value. The threshold value is defined relative to the maximum value and the points where the spectrum is half its maximum value, i.e., we need to find all the solutions $\bar{\omega}$ of

$$\hat{\varphi}(\bar{\omega}) = \frac{\max_{\omega\in\mathbb{R}} \hat{\varphi}(\omega)}{2}. \tag{4.23}$$

For each case identified in Fig. 4.3, the derivation of a closed form expression of the bandwidth is very tricky. Hence, we propose some simple approximations. For cases L1, LB, BP, each part of the bandwidth is approximated by a triangle. A triangle is formed by taking the zero position, the maximum position and double the turning point as already illustrated in Fig. 4.4. For cases L2 and L3, a better approximation is obtained by considering that the WDoG is almost equivalent to a Gaussian function.

Let us determine the zero position and the maximum position of the WDoG. The solutions of $\hat{\varphi}(\omega) = 0$ are the two opposite roots ω_0 and $-\omega_0$ with

$$\omega_0 = \sqrt{\frac{2}{\sigma_b^2 - \sigma_a^2} \ln\left(\frac{b}{a}\right)} \tag{4.24}$$

when $b > a$. Otherwise, when $b \leq a$, the WDoG is always positive with the single root $\omega = 0$ in the special case $a = b$. Since the WDoG is differentiable, the extrema are the solutions of $\hat{\varphi}'(\omega) = 0$ where $\hat{\varphi}'(\omega)$ is the first derivative of $\hat{\varphi}(\omega)$. A short calculation shows that

$$\hat{\varphi}'(\omega) = \frac{\omega}{2\pi}\left(-a\sigma_a^2 \exp\left(-\frac{\omega^2\sigma_a^2}{2}\right) + b\sigma_b^2 \exp\left(-\frac{\omega^2\sigma_b^2}{2}\right)\right).$$

A first extrema is $\omega = 0$. The other extrema are the solutions of

$$\exp\left(\frac{\omega^2(\sigma_b^2 - \sigma_a^2)}{2}\right) = \frac{b\sigma_b^2}{a\sigma_a^2} = \varrho(a, b). \tag{4.25}$$

If $\varrho(a, b) \leq 1$, there is no other extrema. Otherwise, there are two opposite extrema $\omega_1 > 0$ and $-\omega_1$ where

$$\omega_1 = \sqrt{\frac{2}{\sigma_b^2 - \sigma_a^2} \ln(\varrho(a, b))}. \tag{4.26}$$

From (4.25), it is clear that $\hat{\varphi}'(\omega) > 0$ for $0 < \omega < \omega_1$ and $\hat{\varphi}'(\omega) > 0$ for $\omega_1 < \omega$. Since $\lim_{\omega\to\pm\infty} \hat{\varphi}(\omega) = 0$, the frequency ω_1, if it exists, is a global maximum. We can also show that $-\omega_1$ is a global maximum. We obtain the following proposition.

Proposition 5. *If $\varrho(a, b) > 1$, then the maximum is given by*

$$\begin{aligned} \max_{\omega\in\mathbb{R}} \hat{\varphi}(\omega) &= \hat{\varphi}_1 = \hat{\varphi}(\omega_1) = \hat{\varphi}(-\omega_1) \\ &= \frac{a(\sigma_b^2 - \sigma_a^2)}{2\pi}\left(\frac{1}{\varrho(a, b)}\right)^{\frac{\sigma_a^2}{\sigma_b^2 - \sigma_a^2}} \end{aligned} \tag{4.27}$$

with ω_1 given in (4.26). Otherwise, if $\varrho(a, b) \leq 1$, then

$$\max_{\omega\in\mathbb{R}} \hat{\varphi}(\omega) = \hat{\varphi}_0 = \hat{\varphi}(0) = \frac{a - b}{2\pi} > 0.$$

Proof. If $\varrho(a, b) \leq 1$, there is only one extrema $\omega = 0$. Since $\hat{\varphi}'(\omega)$ has the sign of $-\omega$, $\hat{\varphi}$ is increasing when $\omega < 0$ and decreasing when $\omega > 0$. Hence, 0 is a positive global maximum. If $\varrho(a, b) > 1$, it has been already shown that the maximum are located at ω_1 and $-\omega_1$.

The maximum value in (4.27) is obtained by inserting ω_1, given in (4.26), in (4.21). □

According to the couple (a, b), the WDoG has three possible behaviors: lowpass, bandpass or lowpass/bandpass. Hence, the total bandwidth $B = B(a, b)$, including negative and positive frequencies, could be given by one of the three following forms:

- Lowpass: $B = [-\omega_H, \omega_H]$ with $\omega_H > 0$,
- Bandpass: $B = [-\omega_H, -\omega_L] \cup [\omega_L, \omega_H]$ with $0 < \omega_L < \omega_H$,
- Lowpass/bandpass: $B = [-\omega_C, \omega_C] \cup [-\omega_H, -\omega_L] \cup [\omega_L, \omega_H]$ with $0 < \omega_C < \omega_L < \omega_H$.

The following proposition gives the bandwidth $B(a, b)$ with respect to the couple (a, b).

Proposition 6. *According to the value of $\varrho = \varrho(a, b)$, the WDoG $\hat{\varphi}(\omega)$ satisfies one of the following cases:*

1. *If $\varrho > 1$ and $|\hat{\varphi}_0| \geq \hat{\varphi}_1$*
   (a) ***Case L1:** if $|\hat{\varphi}_0| \geq 2\hat{\varphi}_1$, then $\hat{\varphi}(\omega)$ is lowpass with*
   $$\omega_H \simeq \frac{\omega_0}{2},$$
   (b) ***Case LB:** if $|\hat{\varphi}_0| < 2\hat{\varphi}_1$, then $\hat{\varphi}(\omega)$ is lowpass/bandpass with*
   $$\omega_C \simeq \frac{\omega_0}{2},\ \omega_L \simeq \frac{\omega_0 + \omega_1}{2},\ \omega_H \simeq \frac{\omega_0 + 3\omega_1}{2},$$
2. *If $\varrho > 1$ and $|\hat{\varphi}_0| < \hat{\varphi}_1$*
   (a) ***Case BP:** if $|\hat{\varphi}_0| \leq \frac{\hat{\varphi}_1}{2}$, then $\hat{\varphi}(\omega)$ is bandpass with*
   $$\omega_L \simeq \frac{\omega_0 + \omega_1}{2},\ \omega_H \simeq \frac{\omega_0 + 3\omega_1}{2},$$
   (b) ***Case L2:** if $|\hat{\varphi}_0| > \frac{\hat{\varphi}_1}{2}$, then $\hat{\varphi}(\omega)$ is lowpass with*
   $$\omega_H \simeq \omega_1 + \frac{\sqrt{2\ln(2)}}{\sigma_a},$$
3. *If $\varrho \leq 1$, which corresponds to **Case L3**, then $\hat{\varphi}(\omega)$ is lowpass with*
   $$\omega_H \simeq \frac{\sqrt{2\ln(2)}}{\sigma_a}.$$

Proof. The behavior of the filter depends on the maximum values $\hat{\varphi}_0$ or $\hat{\varphi}_1$, the maximum position and the zero position. In case of L1, LB and BP, each part of the bandwidth is approximated by a triangle. A short calculation gives the bounds of the triangle. In case of L2, we use two Gaussians to approximate the central part of the spectrum. The bandwidth of a Gaussian function with zero mean and standard deviation σ is given by

$$B_\sigma = \left[-\sigma\sqrt{2\ln(2)}, \sigma\sqrt{2\ln(2)}\right].$$

In case of L3, we use a single Gaussian approximation. □

The retina-inspired filter is filtering the input image $f(\boldsymbol{x})$ by using a WDoG varying in time. By controling the trajectory $(a(t), b(t))$, this filter is able to explore the frequency spectrum of the input image.

In Fig. 4.5 (a) we illustrate the bandwidth $B = B(a, b)$ of the filter as a function of time according to Proposition 6 using bio-plausible parameters given in [Wohrer and Kornprobst, 2009] and described in Fig. 4.1. The approximation consists of 3 different cases which appear progressively in time: for $1 \leq t \leq 5$ ms the bandwidth is approximated according to L3, for $5 < t \leq 23$ ms it follows L2 and finally, for $t > 23$ ms it

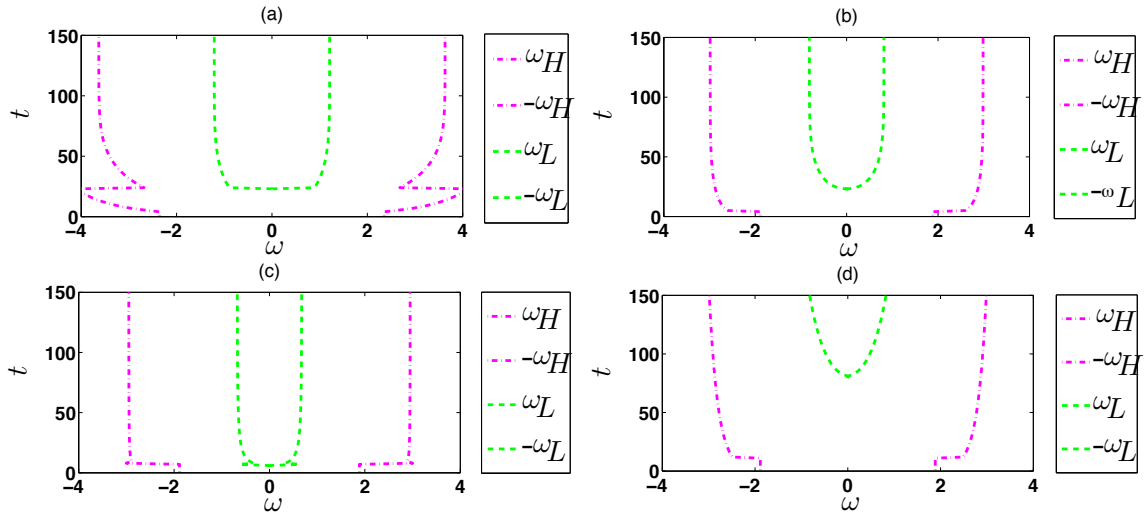


Figure 4.5: For bio-plausible parameters described in Fig. 4.1 we compute in (a) the approximation of the evolution of the bandwidth with respect to time which is computed according to Proposition 6 and in (b) the exact evolution of the bandwidth with respect to time which is numerically computed. (c). The exact solution of the bandwidth which behaves as bandpass ($\tau_C = 40$ ms, $\tau_S = 0.5$ ms, $\tau_G = 9$ ms). (d). The exact solution of the bandwidth which behaves as lowpass for longer time ($\tau_C = 30$ ms, $\tau_S = 20$ ms, $\tau_G = 15$ ms, $n = 5$).

belongs to BP. The exact bandwidth is computed by solving the equation (4.23) using the Matlab^{TM} *fsolve* function and it is depicted in Fig. 4.5 (b). Concerning the exact solution one should notice that the bandwidth evolves in time in a very similar way depicted that the filter behaves as a lowpass for $1 \leq t \leq 23$ ms while it becomes bandpass for $t > 23$ ms. Comparing these two plots, we confirm the accuracy of Proposition 6 is not perfect because of the triangle approximation. This approximation of the DoG filter has already been used in previous works getting similar accuracy results [Birch et al., 2010]. It would be also important to be mentioned that the approximation is not continuous but it has been proposed for each case separately. This is the reason why some discontinuities appear. The second row of Fig. 4.5 reminds the reader that if one chooses different set of parameters, not only the behavior but also the evolution of the bandwidth of the filter will change with respect to time .

4.5 Numerical Results

This section aims to represent the retina-inspired filtering results. Concerning the two different cases when $T = +\infty$ and $T < +\infty$, one could notice according to Fig. 4.2 that the second one is separated into three regions for $t \in [0, T]$, $t \in (T, 2T]$ and $t > 2T$. Concerning the first region, the retina-inspired filter would evolve exactly as in case $T = +\infty$. The second region is the inverse of the first group and the third region corresponds to the zero value because of the zero convergence. According to Fig. 4.1 it is obvious that at least for bio-plausible parameters, the two first groups have exactly the same spectrum. As a result, we concentrate the analysis on the first group which is identical to $T = +\infty$.

4.5.1 1D Input Signal

Due to the limit of space it is impossible to show the complete evolution of space in time when the filter is applied to an image. For this reason we first illustrate the impact of the retina-inspired filter to an 1D signal. This signal is a piecewise constant signal which

remains constant with respect to time. This constant effect is completely equivalent when we flash a still-image for a given time T. As explained above we are going to concentrate the analysis when $T = +\infty$ which is the time the 1D signal exists. All the mathematical notations which are given above are consistent of this example if one assumes just that $\boldsymbol{x} \in \mathbb{R}$ instead of $\mathbb{R}^2$.

Figure 4.6 (a) shows the piecewise constant signal $f(\boldsymbol{x}, t)$ which exists for time $T = +\infty$. This signal is filtered by the retina-inspired filter $\phi(\boldsymbol{x}, t)$ (see Fig. 4.6 (b)) resulting in the activation degree $A(\boldsymbol{x}, t)$ which is illustrated in Fig. 4.6 (c). Obviously, the retina-inspired filter has a great impact on the input signal not only in space but also in time. The filtered 1D signal appears to be smoothed with respect to space but the way it is smoothed changes along time due to the dynamic behavior of the retina-inspired filter. In addition, the DoG bases of each subband of the retina-inspired filter introduces some high contrast effects every time the intensity of the input signal changes.

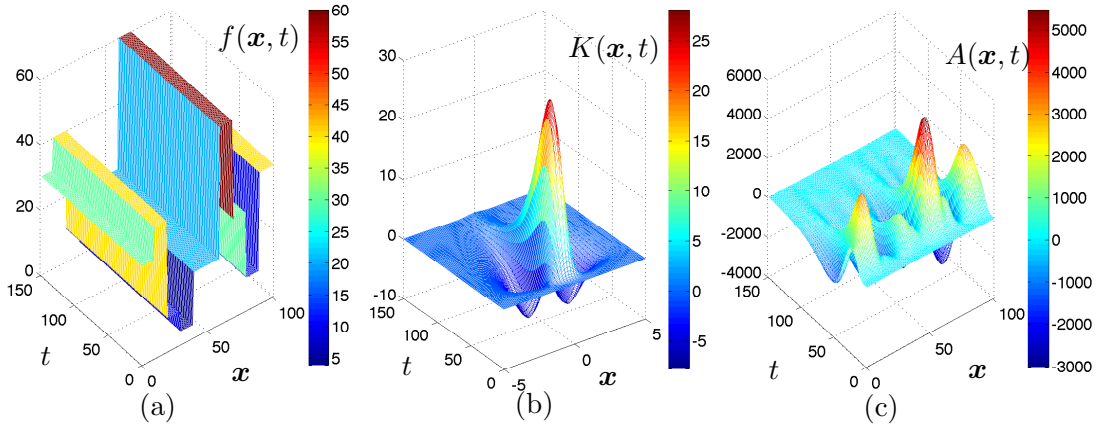


Figure 4.6: This figure shows the complete impact of the retina-inspired filter when it is applied to an 1D signal $f(\boldsymbol{x}, t)$, $\boldsymbol{x}, t \in \mathbb{R}$, which exists for a given time $T = 150$ ms. (a) Piecewise 1D signal (b) 1D retina-inspired filter (c) Retina-inspired decomposition of the 1D signal.

4.5.2 Image Input Signal

In Fig. 4.7, we represent the filtering results while the retina-inspired filter is applied to an image for bio-plausible parameters $w_c = w_s = 1$. The images of left and the middle columns belong to the database USC-SIPI [Weber, 1977]. The image of the right column is a picture retrieved from a video stream captured by our partner, 4G TECHNOLOGY, for video surveillance reasons. The size of the images is $n = 512 \times 512$ pixels. We have decided to illustrate only 5 out of 150 decomposition layers for each experiment. The evolution of the retina-inspired filter is according to the bandwidth of Fig. 4.5 (b). In Fig. 4.8 we tested different parameters for the constant weights w_c and w_s. The first column of Fig. 4.8 corresponds to the bio-plausible parameters where $w_c = w_s = 1$. The middle and the left columns correspond to non bio-plausible parameters but very close to the ones of the left column. It is interesting that while w_c decreases there is a stronger categorization of the L2, L3, LB and BP cases.

In all the above cases, it is easy to observe that while time increases the filtering results change. Thus, in the beginning, the retina-inspired filter smooths the image emphasizing to its low frequency components. However, while time increases, low and high frequency components appear in the filtering results emphasizing the contours of the image. Another important remark is related to the scale of the top pictures which correspond to the first decomposition layer. Apparently, this layer seems to be identical to the original. How-

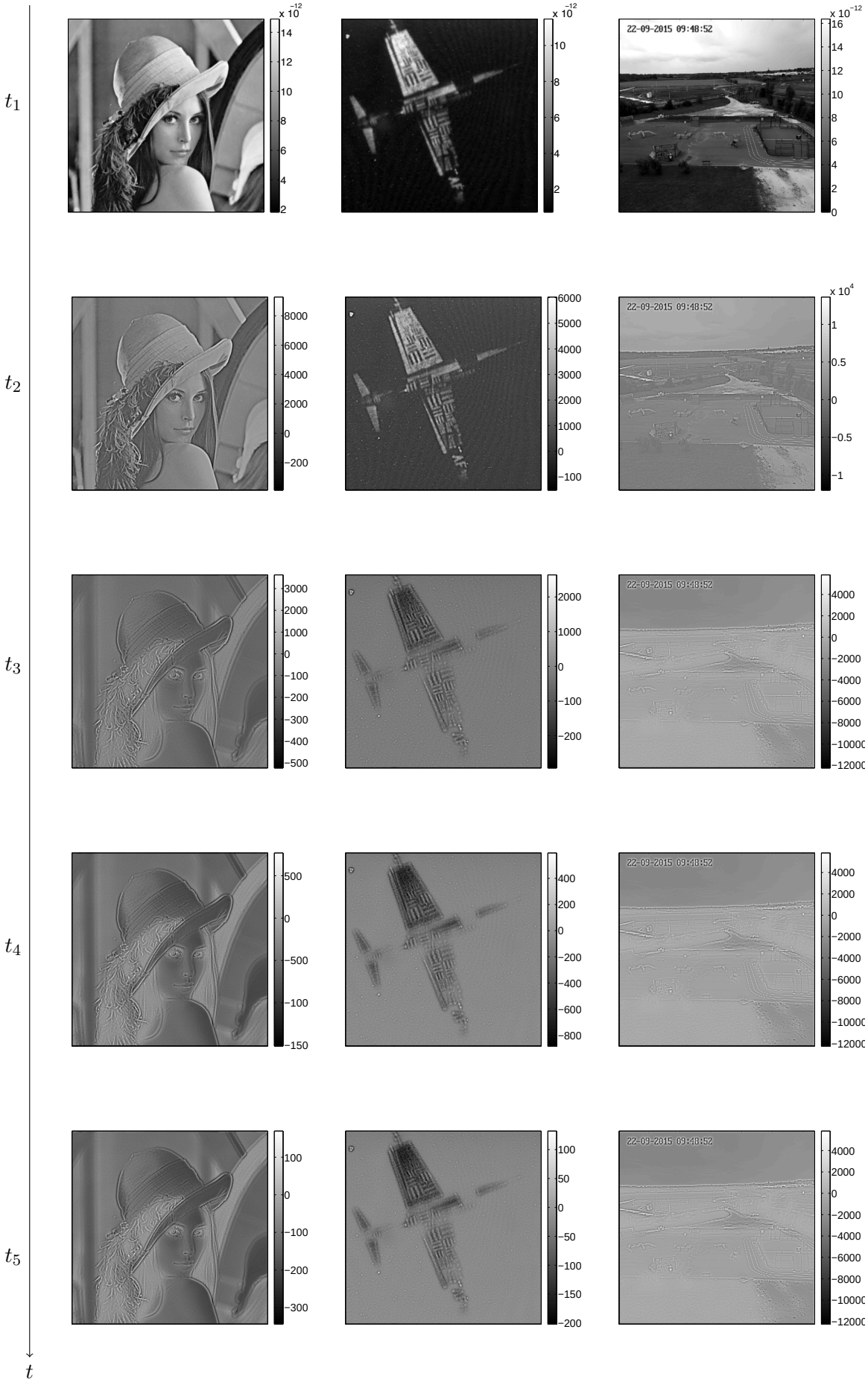


Figure 4.7: Decomposition of different kind of images using the retina-inspired non-separable spatiotemporal filter for a bio-plausible set of parameters $w_c = w_s = 1$. From the top to the bottom: $t_1 = 1$ ms, $t_2 = 30$ ms, $t_3 = 60$ ms, $t_4 = 90$ ms and $t_5 = 120$ ms.

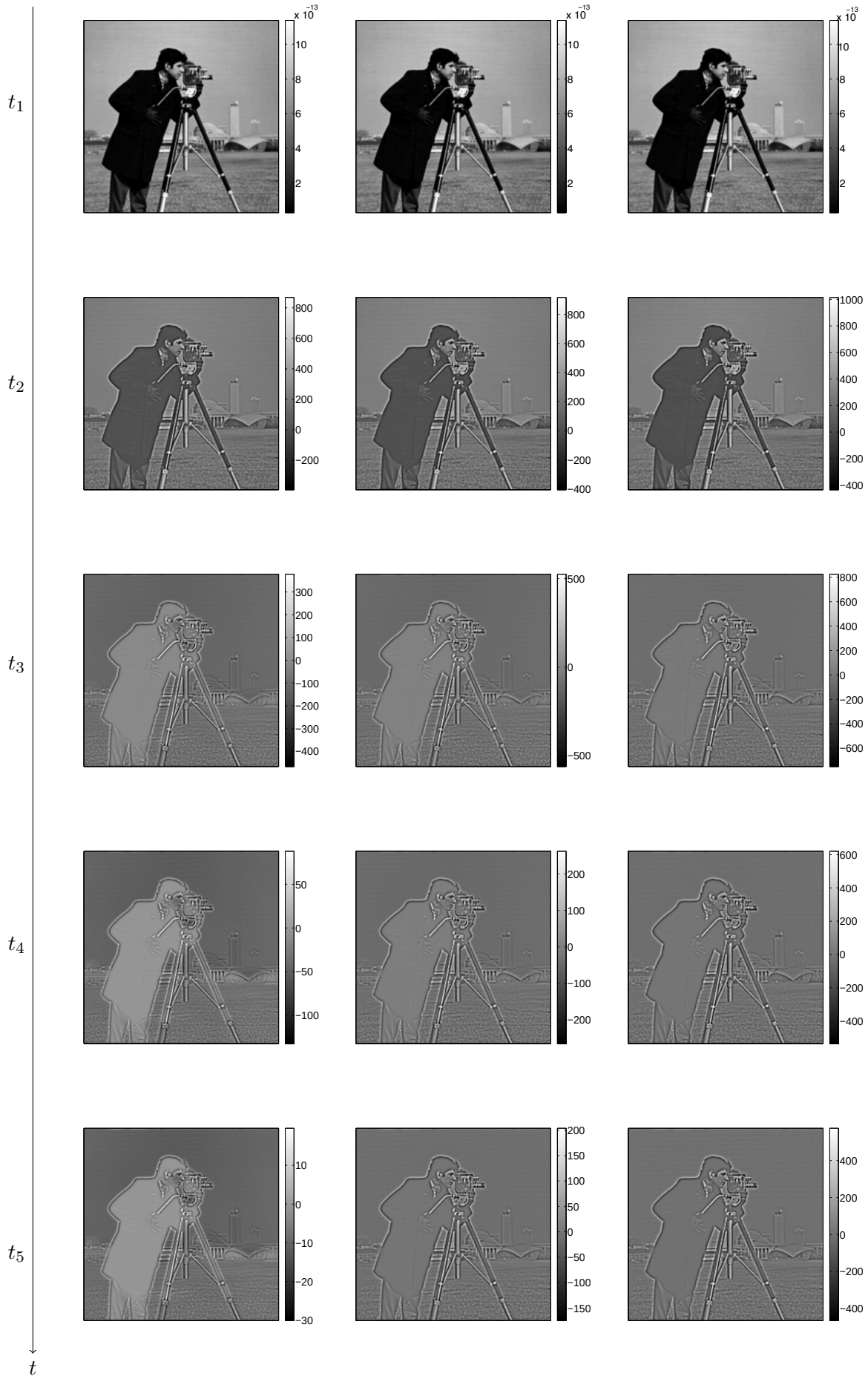


Figure 4.8: Decomposition of an image using the retina-inspired filter. Left: $w_c = w_s = 1$. Middle: $w_c = 0.9$ and $w_s = 1$. Right: $w_c = 0.7$ and $w_s = 1$. From the top to the bottom: $t_1 = 1$ ms, $t_2 = 30$ ms, $t_3 = 60$ ms, $t_4 = 90$ ms and $t_5 = 120$ ms.

ever, its scale is very low (i.e. 10^{-12}) which means that in the presence of noise or after quantization all this information will be completely lost.

4.6 Conclusion

This chapter has introduced a non-separable spatiotemporal retina-inspired filter based on a realistic model of the OPL. This filter is a groundbreaking analysis of neuroscience for image processing. The retina-inspired filter keeps the dynamic behavior of the non-separable spatiotemporal OPL neuroscientific model which was introduced by Fleet and it was adapted in Virtual Retina. However, under the assumption that the input signal is an image which is flashed for a given time, the retina-inspired filter turns into a group of time-varying WDoGs. This chapter proposes a spatial and frequential analysis of a general WDoG function. This study established that there are 5 different approximation cases of a WDoG filter depending on the values of the weights of the two Gaussian filters: 1) lowpass L1, 2) lowpass/bandpass LB, 3) bandpass BP, 4) lowpass L2 and 5) lowpass L3. The retina-inspired filter consists of L3, L2 and BP which appear in this order progressively in time. Hence, it enables the extraction of different kinds of data while time increases. Initially, the filter is a lowpass filter generating low frequency copies of the input signal and it turns into a bandpass filter enabling to extract high frequencies. As underlined in the introduction, the retina-inspired filter is a great improvement of other bio-inspired filters which are simpler and not as much accurate as the neuroscientific approximations of the retina. Hence, this is certainly of a great interest to image processing field.

We aim to adapt the retina-inspired filter in a bio-inspired coding principle. As a result, the following chapter proves that this transform is invertible and it allows a perfect reconstruction of the input signal.

Chapter 5

Inverse Retina-Inspired Filtering

Contents

5.1 Introduction

According to the retina-inspired coding principle (see Fig. 2.20) which was introduced in chapter II we aim to encode an input image $f(\boldsymbol{x}, t)$ which is flashed for a given time T mimicking the way the retina encodes the visual stimulus. The output of the encoding process is a code of spikes based on the lossy compression format. These spike trains are used to reconstruct a version of the input signal $\tilde{f}(\boldsymbol{x}, t)$ which is visually and numerically close to the original signal $f(\boldsymbol{x}, t)$. The retina tissue and the neuromathematical models including Virtual Retina, which approximate its encoding processing are interesting methods to be adapted in the encoding path. However, the visual cortex does not guarantee that the code of spikes which includes all the information about the visual signal, enables a high reconstruction quality. The reason is simple, the visual cortex uses the code of spikes to analyze it and to communicate with the rest of the brain cortices instead of reconstructing the input signal. However, in this thesis the decoding process is necessary for the reconstruction and we need to mathematically prove that it exists.

The retina-inspired filter $\phi(\boldsymbol{x}, t)$ which was introduced in chapter 4 is the first step of the retina-inspired encoding chain (see Fig. 2.20). In this chapter, we prove that the retina-inspired transform is invertible such that we are able to perfectly reconstruct the input signal i.e. $\tilde{f}(\boldsymbol{x}, t) = f(\boldsymbol{x}, t)$ based on the retina-inspired decomposition layers. This proof was based on frame theory which is detailed in the section 5.3. Then, in section 5.4 the pseudo-inverse frame is introduced in a matrix form. Due to the high computational cost of the pseudo-inversion, we utilize in section 5.5 the conjugate gradient descend as a simpler, faster and more efficient numerical reconstruction method. In section 5.6 we illustrate the

reconstruction numerical results when all the coefficients of the retina-inspired frame are used. Last but not least, we also show that the reconstruction based on the retina-inspired decomposition performs well even in the presence of random noise. Some interesting results concerning the quality of the reconstruction with noise are that although the distortion rate increases when the range of noise increases, the PSNR values remain above 30 dB. This is a typical measurement for good quality reconstruction images in compression. In addition, the SSIM ≈ 0.9, very close to 1, which corresponds to the perfect reconstruction (see section 2.3.1.2).

5.2 Discrete Retina-inspired Filter

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^n$, where $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ and $t_1, \ldots, t_m \in \mathbb{R}^+$, where $t_1 < t_2 < \ldots < t_m$ be some sets of spatial and temporal sampling points. As a consequence, the continuous spatial convolution (4.1) is approximated by the discrete convolution:

$$\begin{aligned} A(\boldsymbol{x}_k, t_j) &= \phi(\boldsymbol{x}_k, t_j) \circledast f(\boldsymbol{x}_k) \\ &= \sum_{i=1}^{n} \phi(\boldsymbol{x}_k - \boldsymbol{x}_i, t_j) f(\boldsymbol{x}_i), \forall k, j. \end{aligned} \tag{5.1}$$

Let $f = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)) = (f_1, \ldots, f_n)$ be the discretized image and $\|f\|$ its Euclidean norm and let also $\varphi_{k,j}$ be the row vector of $\mathbb{R}^n$ defined by

$$\varphi_{k,j} = [\phi(\boldsymbol{x}_k - \boldsymbol{x}_1, t_j), \ldots, \phi(\boldsymbol{x}_k - \boldsymbol{x}_n, t_j)]\,. \tag{5.2}$$

Let us denote $\hat{\phi}_{t_j}(\xi)$ the discrete Fourier transform of the vector $(\phi(\boldsymbol{x}_1, t_j), \ldots, \phi(\boldsymbol{x}_n, t_j))$. The matrix form the discrete convolution is given by:

$$\underbrace{\begin{bmatrix} A(\boldsymbol{x}_1, t_1) \\ A(\boldsymbol{x}_2, t_1) \\ \vdots \\ A(\boldsymbol{x}_n, t_1) \\ A(\boldsymbol{x}_1, t_2) \\ A(\boldsymbol{x}_2, t_2) \\ \vdots \\ A(\boldsymbol{x}_n, t_2) \\ \vdots \\ A(\boldsymbol{x}_1, t_m) \\ A(\boldsymbol{x}_2, t_m) \\ \vdots \\ A(\boldsymbol{x}_n, t_m) \end{bmatrix}}_{nm \times 1} = \underbrace{\begin{bmatrix} \phi(\boldsymbol{x}_1 - \boldsymbol{x}_1, t_1) & \phi(\boldsymbol{x}_1 - \boldsymbol{x}_2, t_1) & \ldots & \phi(\boldsymbol{x}_1 - \boldsymbol{x}_n, t_1) \\ \phi(\boldsymbol{x}_2 - \boldsymbol{x}_1, t_1) & \phi(\boldsymbol{x}_2 - \boldsymbol{x}_2, t_1) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_1) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_1) & \phi(\boldsymbol{x}_n - \boldsymbol{x}_2, t_1) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_1) \\ \phi(\boldsymbol{x}_1 - \boldsymbol{x}_1, t_2) & \phi(\boldsymbol{x}_1 - \boldsymbol{x}_2, t_2) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_2) \\ \phi(\boldsymbol{x}_2 - \boldsymbol{x}_2, t_2) & \phi(\boldsymbol{x}_2 - \boldsymbol{x}_2, t_2) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_2) & \phi(\boldsymbol{x}_n - \boldsymbol{x}_2, t_2) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\boldsymbol{x}_1 - \boldsymbol{x}_1, t_m) & \phi(\boldsymbol{x}_1 - \boldsymbol{x}_2, t_m) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_m) \\ \phi(\boldsymbol{x}_2 - \boldsymbol{x}_2, t_m) & \phi(\boldsymbol{x}_2 - \boldsymbol{x}_2, t_m) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_m) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_m) & \phi(\boldsymbol{x}_n - \boldsymbol{x}_2, t_m) & \ldots & \phi(\boldsymbol{x}_n - \boldsymbol{x}_n, t_m) \end{bmatrix}}_{nm \times n} \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_n) \end{bmatrix} \Leftrightarrow$$

$$\underbrace{\begin{bmatrix} A(\boldsymbol{x}_1,t_1)\\ A(\boldsymbol{x}_2,t_1)\\ \vdots\\ A(\boldsymbol{x}_n,t_1)\\ A(\boldsymbol{x}_1,t_2)\\ A(\boldsymbol{x}_2,t_2)\\ \vdots\\ A(\boldsymbol{x}_n,t_2)\\ \vdots\\ A(\boldsymbol{x}_1,t_m)\\ A(\boldsymbol{x}_2,t_m)\\ \vdots\\ A(\boldsymbol{x}_n,t_m) \end{bmatrix}}_{nm\times 1} = \underbrace{\begin{bmatrix} \varphi_{1,1}\\ \varphi_{2,1}\\ \vdots\\ \varphi_{n,1}\\ \varphi_{1,2}\\ \varphi_{2,2}\\ \vdots\\ \varphi_{n,2}\\ \vdots\\ \varphi_{1,m}\\ \varphi_{2,m}\\ \vdots\\ \varphi_{n,m} \end{bmatrix}}_{nm\times n} \underbrace{\begin{bmatrix} f(\boldsymbol{x}_1)\\ f(\boldsymbol{x}_2)\\ \vdots\\ f(\boldsymbol{x}_n) \end{bmatrix}}_{n\times 1} \Leftrightarrow$$

$$\underbrace{\begin{bmatrix} A_{t_1}\\ \vdots\\ A_{t_m} \end{bmatrix}}_{nm\times 1} = \underbrace{\begin{bmatrix} \phi_1\\ \vdots\\ \phi_m \end{bmatrix}}_{nm\times n} \underbrace{\begin{bmatrix} f_1\\ \vdots\\ f_n \end{bmatrix}}_{n\times 1} \Leftrightarrow$$

$$A_{nm\times 1} = \Phi_{nm\times n} f_{n\times 1},$$

where

$$A_{t_j} = \begin{bmatrix} A(x_1,t_j)\\ \vdots\\ A(x_n,t_j) \end{bmatrix} \quad \text{and} \quad \phi_j = \begin{bmatrix} \varphi_{1,j}\\ \vdots\\ \varphi_{n,j} \end{bmatrix} = \begin{bmatrix} [\phi(\boldsymbol{x}_1-\boldsymbol{x}_1,j),\ldots,\phi(\boldsymbol{x}_1-\boldsymbol{x}_n,j)]\\ \vdots\\ [\phi(\boldsymbol{x}_n-\boldsymbol{x}_1,j),\ldots,\phi(\boldsymbol{x}_n-\boldsymbol{x}_n,j)] \end{bmatrix}. \tag{5.3}$$

5.3 Retina-inspired Frame

In this section, we are going to prove that the retina-inspired filter is invertible using the frame theory. This proof is the second contribution of this thesis since it allows us to use the retina-inspired decomposition in terms of compression.

5.3.1 Frame Theory

The frame theory was originated by Duffin and Schaeffer [Duffin and Schaeffer, 1952]. Their motivation was to establish general conditions under which one can recover a vector f in Hilbert space $\mathbf{H}$ from its inner product with a family of vectors $\{g_n\}_{n\in\Gamma}$. The index of Γ maybe finite or infinite. The following frame definition gives an energy equivalent to invert the operator U defined by:

$$\forall n \in \Gamma, \quad Uf[n] = \langle f, g_n\rangle.$$

The definition of a frame according to the frame theory is given by:

Definition 5.3.1. A sequence $\{g_n\}_{n\in\Gamma}$ is a frame of $\mathbf{H}$ if there exist two constants $\alpha > 0$ and $\beta > 0$ such that for any $f \in \mathbf{H}$

$$\alpha\|f\|^2 \leq \sum_{n\in\Gamma} |\langle f, g_n\rangle|^2 \leq \beta\|f\|^2. \tag{5.4}$$

When $\alpha = \beta$ the frame is said to be tight.

If the frame condition is satisfied then U is called frame operator. Condition (5.4) is necessary and sufficient to guarantee that U is invertible, with a bounded inverse. Thus, a frame defines a complete and stable representation of the signal which may also be redundant. When the frame is normalized $\|g_n\| = 1$, this redundancy is measured by the frame bounds α and β. If $\{g_n\}_{n\in\Gamma}$ are linearly independent then it is proven in [Mallat, 1999] that $\alpha \leq 1 \leq \beta$. The frame is an orthonormal bases if and only if $\alpha = \beta = 1$. This is verified if in (5.4) we set $f = g_n$. If $\alpha > 1$, then the frame is redundant and α can be interpreted as a minimum redundancy factor [Mallat, 1999].

5.3.2 Frame Proof

Interestingly, one can prove that the retina-inspired family of vectors $\Phi = \left\{\varphi_{k,j}\right\}_{1\leq k\leq n, 1\leq j\leq m}$, is a frame, where $\varphi_{k,j}$ is given by eq. (5.2) based on frame theory (see section 5.3.1). Thanks to this, we are able to reconstruct the input image by inverting the retina-inspired filter.

Proposition 7. *The family of vectors Φ is a frame, because there exist two scalars $0 < \alpha \leq \beta < \infty$ such that:*

$$\alpha\|f\|^2 \leq \sum_{j=1}^{m}\sum_{k=1}^{n}|A(\boldsymbol{x}_k, t_j)|^2 \leq \beta\|f\|^2, \tag{5.5}$$

where

$$\alpha = \min_{\xi_k}\left\{\frac{1}{n}\sum_{j=1}^{m}\left|\hat{\phi}(\xi_k, t_j)\right|^2\right\} > 0, \tag{5.6}$$

$$\beta = \sum_{j=1}^{m}\sum_{k=1}^{n}\sum_{i=1}^{n}\phi^2(\boldsymbol{x}_k - \boldsymbol{x}_i, t_j). \tag{5.7}$$

Proof. This proof establishes that the non-separable spatiotemporal filter is a frame according to [Kovacevic and Chebina, 2008]. First, we study the lower bound and next the upper bound. We are going to show that the existence of the lower bound α strongly depends on the temporal sampling rule.

- **Lower Bound**

  First of all, we use the Parserval Theorem to transform the coefficients after the retina-inspired filter in Fourier domain:

$$\begin{aligned}\sum_{j=1}^{m}\sum_{k=1}^{n}\left|A(\boldsymbol{x}_k, t_j)\right|^2 &= \sum_{j=1}^{m}\sum_{k=1}^{n}\frac{1}{n}\left|\hat{A}(\xi_k, t_j)\right|^2 \\ &= \frac{1}{n}\sum_{j=1}^{m}\sum_{k=1}^{n}\left|\hat{\phi}(\xi_k, t_j)\hat{f}(\xi_k)\right|^2 \\ &= \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{m}\left|\hat{\phi}(\xi_k, t_j)\right|^2\left|\hat{f}(\xi_k)\right|^2,\end{aligned}$$

  where $\xi_k, \forall k$ is the ordinary frequency which is by definition $\xi_k = \omega_k/2\pi$, where ω_k is the angular frequency. To succeed in the definition of the lower bound α we need to prove that

$$\sum_{j=1}^{m}\left|\hat{\phi}(\xi_k, t_j)\right|^2 \neq 0, \forall\xi_k, \tag{5.8}$$

which guarantees that the spectrum of the input signal will remain exactly the same even after the retina-inspired transform.

Let us suppose by contradiction that there exists ξ_k such that

$$\begin{aligned} \sum_{j=1}^{m} \left|\hat{\phi}(\xi_k, t_j)\right|^2 &= \sum_{j=1}^{m} \left|\hat{\varphi}_{t_j}(\xi_k)\right|^2 = 0 \quad \Leftrightarrow \\ \sum_{j=1}^{m} \left|w_c R_c(t_j)\hat{G}_c(\xi_k) - w_s R_s(t_j)\hat{G}_s(\xi_k)\right|^2 &= 0, \end{aligned} \tag{5.9}$$

which means that

$$w_c R_c(t_j)\hat{G}_c(\xi_k) = w_s R_s(t_j)\hat{G}_s(\xi_k). \tag{5.10}$$

Based on equation (4.8) one can show that $\forall t_j \ R_c(t_j) \neq 0$. In addition, the Fourier transform of a Gaussian is again a Gaussian which means that $\forall \xi_k, \ \hat{G}_c(\xi_k) \neq 0$ and $\hat{G}_s(\xi_k) \neq 0$. As a result, since $w_c, w_s \neq 0$ too, we can rewrite (5.10) as following:

$$\frac{\hat{G}_c(\xi_k)}{\hat{G}_s(\xi_k)} = \frac{w_s R_s(t_j)}{w_c R_c(t_j)}, \ \forall t_j. \tag{5.11}$$

The left side ratio of eq. (5.11) is constant $\forall j$ because $\hat{G}_s(\xi_k)$ and $\hat{G}_c(\xi_k)$ do not depend on time. However, the right side of eq. (5.11) is a function of time. Thus, the rule of the temporal sampling has an impact on the right side ratio. Depending on the temporal sampling rule, two different cases may occur:

Case 1: Let us suppose $\exists(t_i, t_j)$ where $t_i \neq t_j$, such that:

$$\frac{w_s R_s(t_i)}{w_c R_c(t_i)} \neq \frac{w_s R_s(t_j)}{w_c R_c(t_j)}. \tag{5.12}$$

In this case, the right side ratio of eq. (5.11) is not constant $\forall j$ thus, eq. (5.8) is true and we conclude that there exists the lower bound α.

Case 2: Another possible sampling scenario results in eq. (5.13) which means that $\forall j$ the temporal samples have been selected such that the ratio is a constant c:

$$\frac{w_s R_s(t_j)}{w_c R_c(t_j)} = c, \ \forall j. \tag{5.13}$$

This is the case when $\hat{\varphi}_{t_j}(\xi_k) = 0, \forall k$. We have proven in chapter 4 (see eq. (5.14)) that $\forall j$ when $\hat{\varphi}_{t_j}(\omega) = 0$, there exist two roots $\omega_{0,j}$ and $-\omega_{0,j}$ given by:

$$\begin{aligned} \omega_{0,j} &= \sqrt{\frac{2}{\sigma_s^2 - \sigma_c^2} \ln\left(\frac{R_s(t_j)}{R_c(t_j)}\right)} \quad \Leftrightarrow \\ \omega_0 &= \sqrt{\frac{2}{\sigma_s^2 - \sigma_c^2} \ln(c)}. \end{aligned} \tag{5.14}$$

Case 2.1 The lower bound α exists,

- If ω_0 exists and $\omega_k \neq \omega_0$. That means that even if there exists a root the sampling rule has discarded this root from the set of the angular frequencies ω_k. Consequently, the spectrum of the signal will remain the same.
- If ω_0 does not exist, which also means that the spectrum keeps all its frequencies.

Case 2.2 The lower bound α does not exist if ω_0 exists and $\omega_k = \omega_0$. In this case the spectrum changes. Thus, Φ is impossible to be a frame.

For the cases 1 and 2.1 we have proven that eq. (5.8) is true which means that the lower bound is given as:

$$\alpha = \min_{\xi_k} \Big\{ \frac{1}{n} \sum_{j=1}^{m} \Big| \hat{\phi}(\xi_k, t_j) \Big|^2 \Big\} > 0.$$

- **Upper Bound**

We are using the Cauchy-Schwarz inequality to calculate the upper bound of the non-separable spatiotemporal frame:

$$\begin{aligned}
\sum_{j=1}^{m}\sum_{k=1}^{n} \Big| A(\boldsymbol{x}_k, t_j) \Big|^2 &= \sum_{j=1}^{m}\sum_{k=1}^{n} \Big| \phi(\boldsymbol{x}_k, t_j) \circledast f(\boldsymbol{x}_k) \Big|^2 \\
&= \sum_{j=1}^{m}\sum_{k=1}^{n} \Big| \sum_{i=1}^{n} \phi(\boldsymbol{x}_k - \boldsymbol{x}_i, t_j) f(\boldsymbol{x}_i) \Big|^2 \\
&= \sum_{j=1}^{m}\sum_{k=1}^{n} \Big| \sum_{i=1}^{n} \varphi_{k,j}(\boldsymbol{x}_i) f(\boldsymbol{x}_i) \Big|^2 \\
&\leq \sum_{j=1}^{m}\sum_{k=1}^{n} \Big(\Big| \sum_{i=1}^{n} \varphi_{k,j}^2(\boldsymbol{x}_i) \Big| \Big| \sum_{i=1}^{n} f^2(\boldsymbol{x}_i) \Big| \Big) \\
&= \Big(\sum_{j=1}^{m}\sum_{k=1}^{n} \Big| \sum_{i=1}^{n} \varphi_{k,j}^2(\boldsymbol{x}_i) \Big| \Big) \Big| \sum_{i=1}^{n} f^2(\boldsymbol{x}_i) \Big| \\
&= \Big(\Big| \sum_{j=1}^{m}\sum_{k=1}^{n}\sum_{i=1}^{n} \phi^2(\boldsymbol{x}_k - \boldsymbol{x}_i, t_j) \Big| \Big) \Big| \sum_{i=1}^{n} f^2(\boldsymbol{x}_i) \Big| \\
&= \beta \Big\| f \Big\|^2.
\end{aligned}$$

□

5.4 Pseudo-inverse Frame

In Proposition 7 it is proven that the non-separable spatiotemporal filter is a frame hence, the filter is invertible meaning that it is possible to reconstruct the input image. The optimal reconstruction results are given when all the coefficients $A(\boldsymbol{x}_k, t_j)$ are available at the final discrete time t_m.

In practice, we need to solve the linear system $A = \Phi f$ and reconstruct $\tilde{f}$ which according to the Proposition 7 should be $\tilde{f} = f$. At time t_m, the exact estimation of f is given by $\hat{f}_{t_m}$ according to:

$$\tilde{f}_{t_m} = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top A, \tag{5.15}$$

where Φ^{-1} denotes the inverse of a matrix Φ and $\Phi^\top$ denotes its transpose. In the previous section, we demonstrated that the retina-inspired filter Φ is a frame. As a result, we can define as $\Phi^T\Phi$ its frame operator. According to [Conway et al., 1996], the frame operator is bounded, invertible, self-adjoined and positive. Since $\Phi^T\Phi$ is invertible, it exists $(\Phi^T\Phi)^{-1}$.

This is the last step to build the canonical dual frame of a frame Φ [Masmoudi et al., 2012, Kovačević and Vetterli, 1992, Kovacevic and Chebina, 2008], which is necessary to have a perfect decoding at time t_m, and it is defined as $(\Phi^\top\Phi)^{-1}\Phi^\top$.

However, in practice it is impossible to compute the dual frame since the size of matrix Φ is too big and this would be time consuming and resource demanding. One way to solve this problem is to use the conjugate gradient descent which is one of the most efficient iterative methods.

5.5 Conjugate Gradient

The conjugate gradient method solves the same linear system $\Phi f = A$ by minimizing the quadratic function [Shewchuk, 1994]:

$$y(f) = \frac{1}{2} f^T \Phi f - A^T f + c. \tag{5.16}$$

The gradient of a quadratic form is defined as:

$$\bigtriangledown y(f) = \begin{bmatrix} \frac{\partial}{\partial f_1} y(f) \\ \frac{\partial}{\partial f_2} y(f) \\ \vdots \\ \frac{\partial}{\partial f_n} y(f) \end{bmatrix}. \tag{5.17}$$

There are two constraints which need to be verified to use the conjugate gradient to solve the linear system $\Phi f = A$: the matrix Φ should be symmetric and positive-definite. If Φ is symmetric then equation 5.16 reduced to:

$$\bigtriangledown y(f) = \Phi f - A. \tag{5.18}$$

In our case, the retina-inspired filter Φ is symmetric but not positive-definite. In order to overcome this difficulty we multiply with its transpose Φ^T. Now, the matrix $\overline{\Phi} = \Phi^T\Phi$ is positive-definite so we alternate equation (5.16) into the following:

$$y = \frac{1}{2} f^T \overline{\Phi} f - \overline{A}^T f + c. \tag{5.19}$$

where $\overline{A} = \Phi^T A$. Finally, what we need to solve is $\bigtriangledown y(f) = 0$ which corresponds to $\overline{\Phi} f = \overline{A}$. For each iteration we need to estimate the residual $r = \overline{A} - \overline{\Phi} f$ and the direction of the gradient p which is perpendicular to the residual. We first initialize the solution we want to estimate $\tilde{f}_0 = 0$, the residual of each iteration $r_0 = \overline{A}$ and the direction of the gradient $p = r_0$.

The maximum number of iterations is related to the size of the input signal (in our case is n) and the quality of the reconstruction, so the convergence of the algorithm to the optimal solution is related to the condition number of $\overline{\Phi}$ (see Algorithm 1). An important remark is that the Euclidean error ϵ_i which is computed for each iteration of the conjugate gradient descent is a strictly decreasing function. As a result, it holds that $\epsilon_i > \epsilon_{i+1}$ [Gilbert and Nocedal, 1992].

Algorithm 1 Conjugate Gradient Descent

Initialize:
$\tilde{f}_0 = 0$
$r_0 = \overline{A}$
$p_0 = r_0$
for `i = 1:Max_Nbr_Iterations` **do**
  $a_i \leftarrow \frac{r_{i-1}^T r_{i-1}}{p_{i-1}^T \overline{\Phi} p_{i-1}}$ ▷ Compute the step length
  $\tilde{f}_i \leftarrow \tilde{f}_{i-1} + a_i p_{i-1}$ ▷ Update the solution
  $b_i \leftarrow \frac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$ ▷ Compute the gradient correction
  $p_i \leftarrow r_i + b_i p_{i-1}$ ▷ Update the direction
  if $\epsilon_i = ||\overline{A} - \overline{\Phi} \circledast \tilde{f}_i|| \leq \varepsilon$ **then** ▷ 1^{st} Criterion
    `break;`
  end if
end for
return $\tilde{f}_i$ ▷ The reconstructed image is f_i

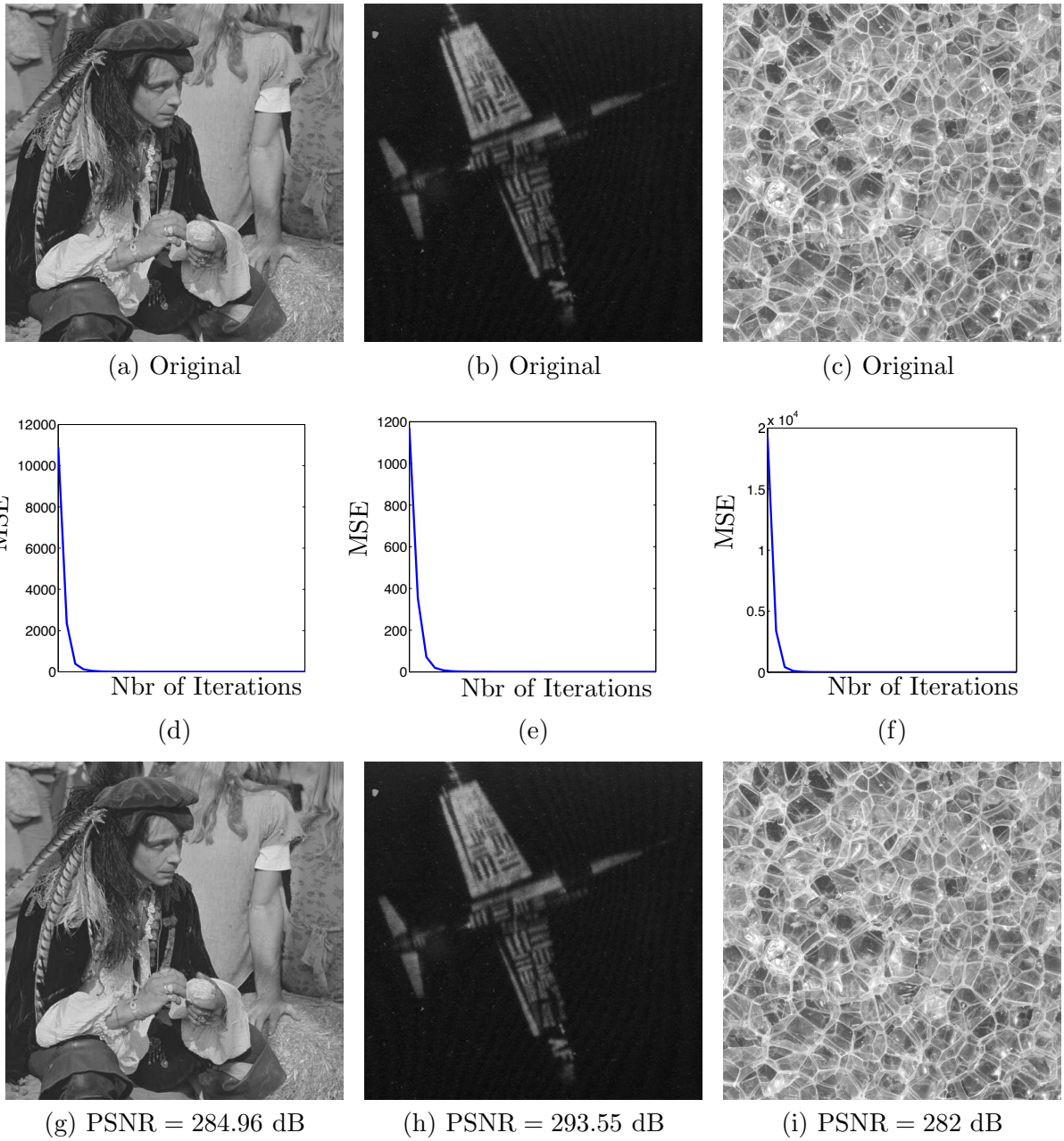
5.6 Numerical Results

This section illustrates some numerical reconstruction results. We tested 100 grayscale images of size $n = 512 \times 512$ taken from the database USC-SIPI [Weber, 1977]. This database contains different kinds of images including the standard in image processing, satellite images, portraits, textures, natural scenes, etc. The results we are going to illustrate later on cover a big variety of input scenes. The reconstruction was achieved using the conjugate gradient descent of maximum number of iterations Max_Nbr_Iterations=100000. This number was sufficiently large for all the experiments we run and by far smaller than the maximum number of iterations the conjugate gradient method requires to provide the optimal solution, which equals the size of the matrix (in our case n). In addition, the following two stopping criteria are tested for each iteration:

$$\epsilon_i = ||\overline{A} - \overline{\Phi} \circledast \tilde{f}_i|| \leq \varepsilon, \tag{5.20}$$

where $\varepsilon = 10^{-20}$. If the above criterion comes true, the conjugate gradient descend algorithm stops generating the reconstructed signal. The criterion ϵ measures the error between the original retina-inspired decomposition and the one which occurs when the reconstructed image $\tilde{f}$ is used. This is a necessary criterion which is used in case there is no a priori information about the original signal.

Fig. 5.1 shows three different kinds of images (portrait, object, texture) which were retina-inspired filtered and they were reconstructed using all the frame coefficients. It also illustrated the conjugate function for each iteration as a strictly decreasing function. The first column of Table 5.1 shows the perfect reconstruction results for all the 100 images we used from USC-SIPI. We compute the mean and the variance values of ϵ, the mean value of the number of iterations which were necessary for the reconstruction and the mean PSNR and SSIM values (see sections 2.3.1.2 and 2.3.1.3).



(a) Original (b) Original (c) Original

(d) (e) (f)

(g) PSNR = 284.96 dB (h) PSNR = 293.55 dB (i) PSNR = 282 dB

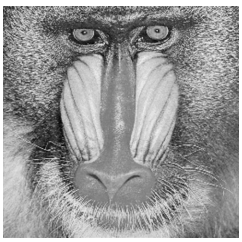
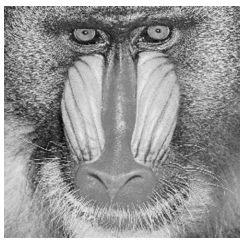
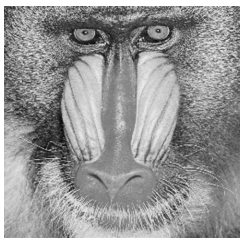
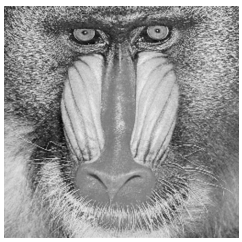
Figure 5.1: Reconstruction of different kinds of images using the complete retina-inspired frame, which consists of 150 layers. The top line depicts the original images, the middle line illustrates the MSE, which is measured for each iteration until one of the three stopping criterion (eq. (5.20)) will come true. The bottom line shows the reconstructed images.

5.6.1 Progressive Reconstruction

The retina-inspired filtering is a dynamic and invertible transform which performs according to the OPL retinal layer. We have proven that using the complete retina-inspired frame the reconstruction is perfect. However, the dynamic nature of the retina-inspired transform raises some questions about the qualitative progress of the reconstruction results with respect to time. Figure 5.2 shows the progressive reconstruction of a still-image which has been filtered by the retina-inspired transform, when only few of the decomposition layers participate to its synthesis.

The results show that even a small number of decomposition layers also leads to a perfect reconstruction. This is obvious due to the nature of the retina-inspired filter which

consists of some decomposition layers which are lowpass filters (see section 4.3). However, this scenario will never be used in practice not only due to the noisy inputs but also because of the quantization which is necessary in compression.

Original Baboon Picture
512 × 512 pixels

t = 10ms
PSNR = 277.18 dB
SSIM = 1

t = 100ms
PSNR = 282.49 dB
SSIM= 1

t = 150ms
PSNR = 28.19dB
SSIM = 1




52^{th} Foreman Picture
512 × 512 pixels

t = 10ms
PSNR = 272.29dB
SSIM = 1

t = 100ms
PSNR = 283.64dB
SSIM= 1

t = 150ms
PSNR = 282.5523dB
SSIM = 1

Figure 5.2: (a) Progressive Reconstruction of the baboon image of the size 512×512 pixels for different time interval. (b) Progressive Reconstruction of the 52^{nd} picture of the size 512×512 pixels of the very well-known foreman video stream for different time intervals.

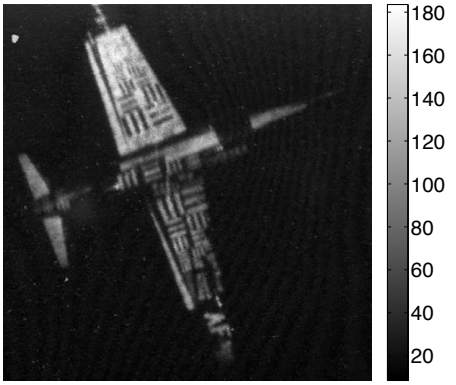
5.6.2 Additive White Gaussian Noise

We have represented some numerical results about retina-inspired decomposition and reconstruction in absence of noise. However, this scenario is not at all realistic since the visual scene is always noisy due to the eye movements. As a result, we would like to study the impact of noise both in decomposition and reconstruction results. As we have already noticed in chapter 4 even though the first decomposition layers include almost all the information about the input signal, their scale is too low (10^{-12}). As a result, it is expected that the presence of noise will influence them.

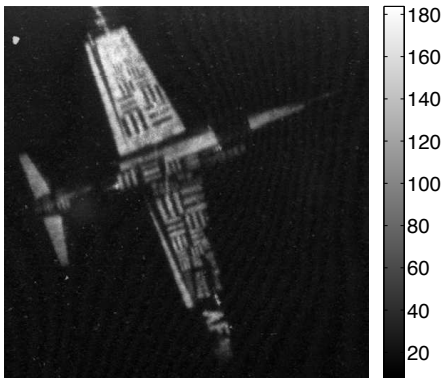
We tested the Additive White Gaussian Noise (AWGN) $\eta(\boldsymbol{x}_k), \forall k$, which is a kind of random additive white noise. The white noise has a constant Power Spectral Density (PSD) which describes how the power P of the signal x is distributed over frequencies and it is defined as the square of a signal:

$$P = \lim_{T\to\infty} \int_{-T}^{T} |x(t)|^2 dt \tag{5.21}$$

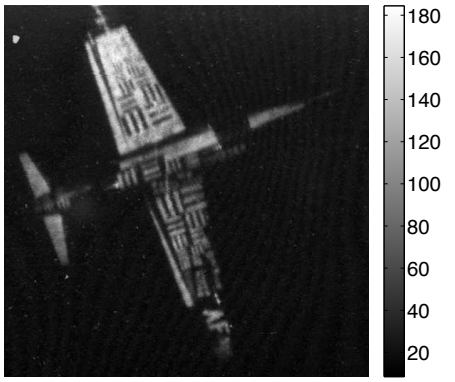
The white noise is a stationary signal which is determined by the unchanged Joint Probability Distribution (JPD) when it is shifted in time. Consequently, the parameters like mean μ and variance σ_η^2 do not change in time. The AWGN is a special case of white noise when there is a normal distribution of a zero mean. Thus, in discrete domain, the



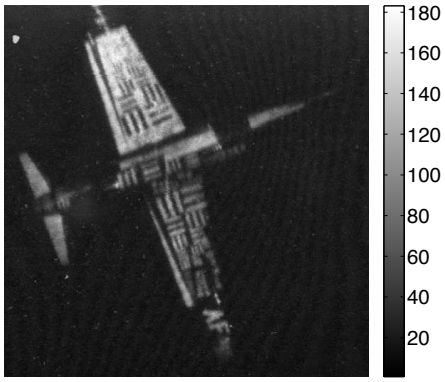
PSNR = 293.556 dB
SSIM = 1
No noise



PSNR = 55.03 dB
SSIM = 0.9987
$\sigma^2 = 25$



PSNR = 51.21 dB
SSIM = 0.9960
$\sigma^2 = 100$



PSNR = 37.58 dB
SSIM = 0.9251
$\sigma^2 = 10^4$

Figure 5.3: Reconstruction of noisy retina-inspired decomposition. The quality of each reconstruction is measured using the PSNR metric which decreases while the range of the noise increase.

AWGN is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean ($\mu = 0$) and finite variance σ_η^2.

The AWGN is applied to the retina-inspired transformed signal $A(\boldsymbol{x}_k, t_j)$ as following:

$$\begin{aligned} A_\eta(\boldsymbol{x}_k, t_j) &= A(\boldsymbol{x}_k, t_j) + \eta(\boldsymbol{x}_k) \\ &= \phi(\boldsymbol{x}_k, t_j) \circledast f(\boldsymbol{x}_k) + \eta(\boldsymbol{x}_k) \end{aligned} \tag{5.22}$$

The spread of a Gaussian distribution changes with respect to its variance. In case of the AWGN, the higher the variance, the stronger the impact of the noise to the signal. Thus, if σ_η^2 is too small, the impact of the noise will be imperceptible. However, in case of a large σ_η^2 the noise will influence almost all the spectrum of the input signal. Figures 5.5 and 5.6 illustrate 5 retina-inspired decomposition (for $t_1 = 1$ ms, $t_2 = 30$ ms, $t_3 = 60$ ms, $t_4 = 90$ ms and $t_5 = 120$ ms) of plane and lena images for 4 different scenarios: 1) no noise, when variance of the AWGN equals 2) $\sigma_\eta^2 = 25$, 3) $\sigma_\eta^2 = 100$ and 4)$\sigma_\eta^2 = 10^4$. The numerical results show that the first decomposition layers are completely blurred due to the

PSNR = 286.051 dB
SSIM = 1
No noise

PSNR = 53.83 dB
SSIM = 0.9994
$\sigma^2 = 25$

PSNR = 49.60 dB
SSIM = 0.9977
$\sigma^2 = 100$

PSNR = 36.51 dB
SSIM = 0.9578
$\sigma^2 = 10^4$

Figure 5.4: Reconstruction of noisy retina-inspired decomposition. The quality of each reconstruction is measured using the PSNR metric which decreases while the range of the noise increase.

noise for all the three different cases. This happens due to the very small range of the first decomposition layers ($\approx 10^{-12}$). Concerning the rest of the layers the impact of the noise is related to the range of each layer. When the range of a layer is small, a strong AWGN enables to blur the signal (see the last row of Figures 5.5 and 5.6). However, if the range of the layer is high (see the second row of Figures 5.5 and 5.6) even a strong noise AWGN (i.e. right column where $\sigma_\eta^2 = 10^4$) would be unable to dramatically change the amplitude of the signal.

Figure 5.3 and 5.4 show the reconstruction results based on the noisy decomposition layers. We measured the quality of the reconstruction using the PSNR and the SSIM metrics (see section 2.3.1.2 and 2.3.1.3). According to the values of the two metrics, while noise increases the quality of the reconstruction decreases. However, not only the PNSR results but also the visual quality of the reconstructed images confess that the the redundancy of the retina-inspired frame is efficient enough to allow low distortion. Figure 5.7 illustrates the mean PSNR and SSIM rates versus the different noise scenarios. Table 5.1 shows the reconstruction results of the 3 different noisy scenarios applied to the set of 100 images

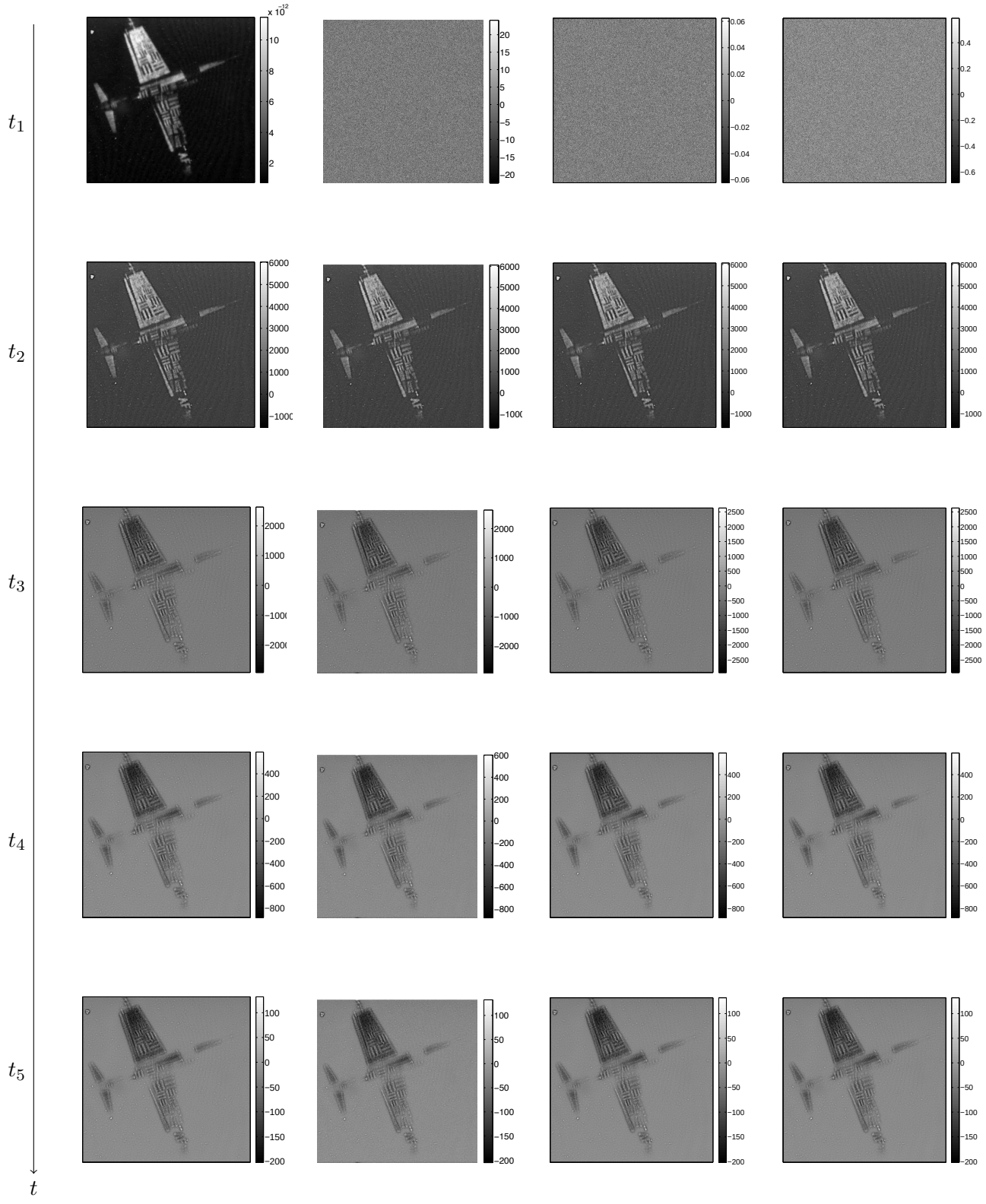


Figure 5.5: Decomposition of the plane image using the retina-inspired non-separable spatiotemporal filter for a bio-plausible set of parameters $w_c = w_s = 1$. From the left to the right: The first column corresponds to the decomposition of no noise ($\sigma_\eta^2 = 0$), the next ones include noise of the following variance $\sigma_\eta^2 = 25$, $\sigma_\eta^2 = 100$ and $\sigma_\eta^2 = 10^4$. From the top to the bottom: $t_1 = 1$ ms, $t_2 = 30$ ms, $t_3 = 60$ ms, $t_4 = 90$ ms and $t_5 = 120$ ms.

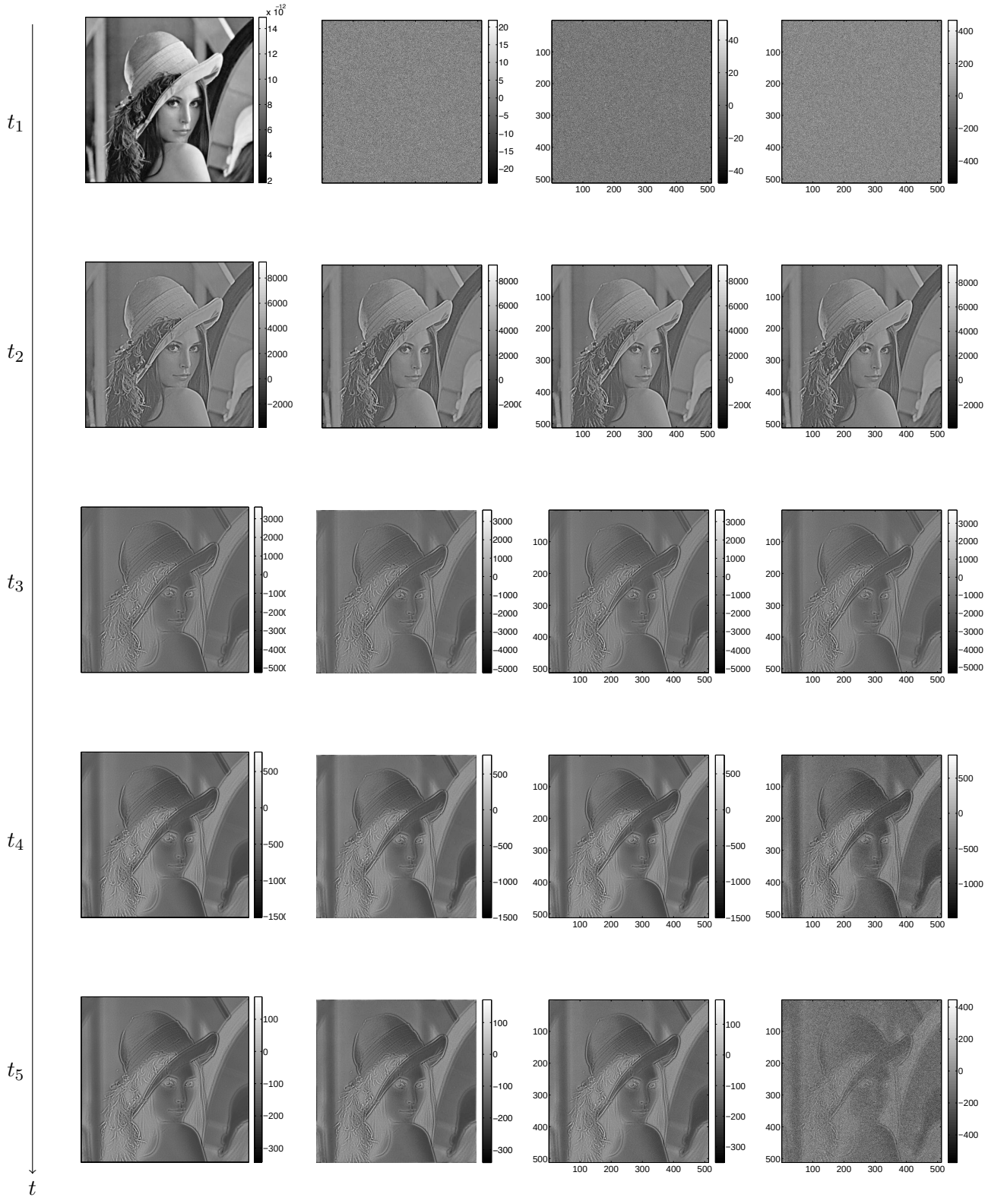
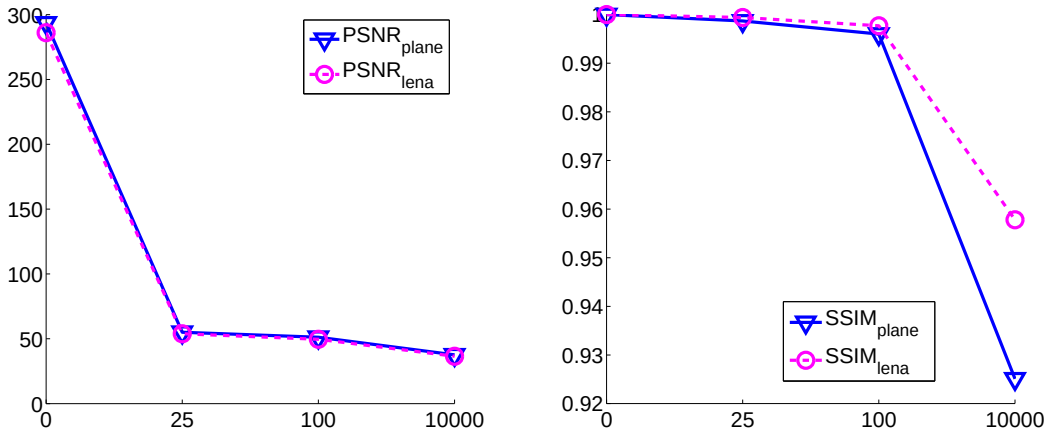


Figure 5.6: Decomposition of lena images using the retina-inspired non-separable spatiotemporal filter for a bio-plausible set of parameters $w_c = w_s = 1$. From the left to the right: The first column corresponds to the decomposition of no noise ($\sigma_\eta^2 = 0$), the next ones include noise of the following variance $\sigma_\eta^2 = 25$, $\sigma_\eta^2 = 100$ and $\sigma_\eta^2 = 10^4$. From the top to the bottom: $t_1 = 1$ ms, $t_2 = 30$ ms, $t_3 = 60$ ms, $t_4 = 90$ ms and $t_5 = 120$ ms.

taken from USC-SIPI database. As expected, while the variance of noise increases the mean PSNR and the SSIM values decrease. On the other hand, even for a strong AWGN noise ($\sigma_\eta^2 = 10^4$) these values still remain sufficient high which guarantees high reconstruction quality.

Range of Noise :	No Noise	$\sigma_\eta^2 = 25$	$\sigma_\eta^2 = 100$	$\sigma_\eta^2 = 10^4$
Mean ϵ	1.7368×10^{-16}	1.3513×10^3	7.1874×10^3	4.1677×10^6
Variance ϵ	3.8134×10^{-33}	1.2810×10^7	3.1500×10^8	6.4219×10^{12}
Mean Iterations	748	145	102	25
Mean PSNR	281.42	63.85	58.47	43.08
Mean SSIM	1	0.9067	0.9011	0.8513

Table 5.1: Reconstruction results using the inverse retina-inspired frame. The results represent 4 different scenarios of noise: 1) no noise, 2) $\sigma_\eta^2 = 15$, 3) $\sigma_\eta^2 = 100$ and 4) $\sigma_\eta^2 = 10^4$.



(a) (b)

Figure 5.7: (a). PSNR rate vs Noise. The x-axis corresponds to the range of noise and the y-axis is the PSNR value. (a). SSIM rate vs Noise. The x-axis corresponds to the range of noise and the y-axis to the SSIM values. (For this experiments we test the 7_2_01.tiff and the lena.tiff images taken from the USC-SIPI database, both of the size 512×512 pixels.)

5.7 Conclusion

In this chapter, we have mathematically proven that the retina-inspired filter is a frame according to the frame theory. We show the existence of the lower and the upper bounds of the frame and we provide their exact formulas. According to the frame theory, it means that the retina-inspired transform is invertible. Thus, we proposed a pseudo-inverse reconstruction model. However, due to the high computational cost of this method, in practice we used an alternative and more efficient computationally algorithm, the conjugate gradient for reconstruction.

We also illustrate some numerical results which guarantee that the reconstruction using all the retina-inspired frame coefficients is perfect. Last but not least, we introduced some AWGN to the retina-inspired decomposition in order to be more realistic and to show the impact of noise on the reconstruction quality. Concerning the decomposition, we observed that the very first decomposition layers which are lowpass filtered are completely blurred due to their low intensity range. However, the reconstruction quality even in presence

of AWGN is not dramatically decreased. This is due to the redundancy of the retina-inspired filter. In fact, we show that the distortion rate is influenced due to the presence of noise however, the quality metric PSNR gives promising results comparing the usual measurements in compression ($\approx$ 30dB).

Part III

DYNAMIC QUANTIZATION

Motivation

In this part we aim to reduce the spatiotemporal redundancy of the retina-inspired frame in order to efficiently compress the input signal into a binary code. This is necessary in lossy compression as we have already introduced in chapter 2 section 2.1.2. A very well-known model to eliminate some meaningless coefficients is the static dead-zone scalar quantizer which was introduced in 2.4.3.3 and it is used in conventional coding principle (see Fig. 5.8 (a)). This model performs well for static transforms. However, as we have explicitly introduced in chapter 4 the retina-inspired transform is a dynamic filter which should be dynamically encoded. Thus, in this chapter, we are motivated to improve the performance of a static dead-zone scalar quantizer being inspired by the dynamic encoding which happens inside the Ganglionic Layer (GL) of the retina by the ganglion cells (see Fig. 5.8 (b)).

The ganglion cells progressively receive the electrical current which has been generated into the previous retina layers (OPL and IPL). These cells are responsible to dynamically build a code of sequence of spikes (spike trains). This code is propagated to the next layers of the visual pathway. There have been proposed several models which approximate the generation of this code. In chapter 6, we represent some of these models focusing on the two most important ones, the Rank Order Coder (ROC) and the Leaky Integrate and Fire (LIF). These two models are very different each other although, both of them share some common assumptions for instance, about the importance of the information which is hidden at the time the first spike of each neuron is emitted. First of all, the ROC model is a static model, while the LIF is a dynamic one. In addition, the LIF model is mathematically better defined comparing to ROC especially concerning the reconstruction process. As a result, the LIF model enables a perfect reconstruction of the input signal based on its output, which is the code of spikes, if the observation window is very large. Hence, in terms of compression, it seems that the LIF model would be more accurate to be used.

Chapter 6 is an introduction to the most famous and widely used neuroscientific models which describe the spiking generation mechanisms of neurons. We also discuss under which assumptions these models were built. In addition, once the neural code is built, we present how this code is interpreted and linked to the input signal. Comparing different interpretation ways we conclude that the model which enables the best matching of the firing rate to the input stimulus is the LIF. In chapter 7, we release of novel dead-zone quantizer which is based on the LIF model. Under some assumptions, we propose how to approximate the LIF by a dead-zone scalar quantizer which enables the progressive reconstruction of the input signal under some time limitations. The novel quantizer is called retina-inspired quantizer or LIF dead-zone quantizer or LIF Quantizer (LIFQ). There are four different models of the retina-inspired quantizer, the perfect-LIFQ, the uniform-LIFQ, the adaptive LIFQ and the optimized-LIFQ depending on the step which is chosen for the quantization of each decomposition layer. We provide results concerning the above models of LIFQ and we prove that the optimized-LIFQ outperforms not only the other LIFQs but also the standards JPEG and JPEG2000 for bitrates higher than 1 bpp.

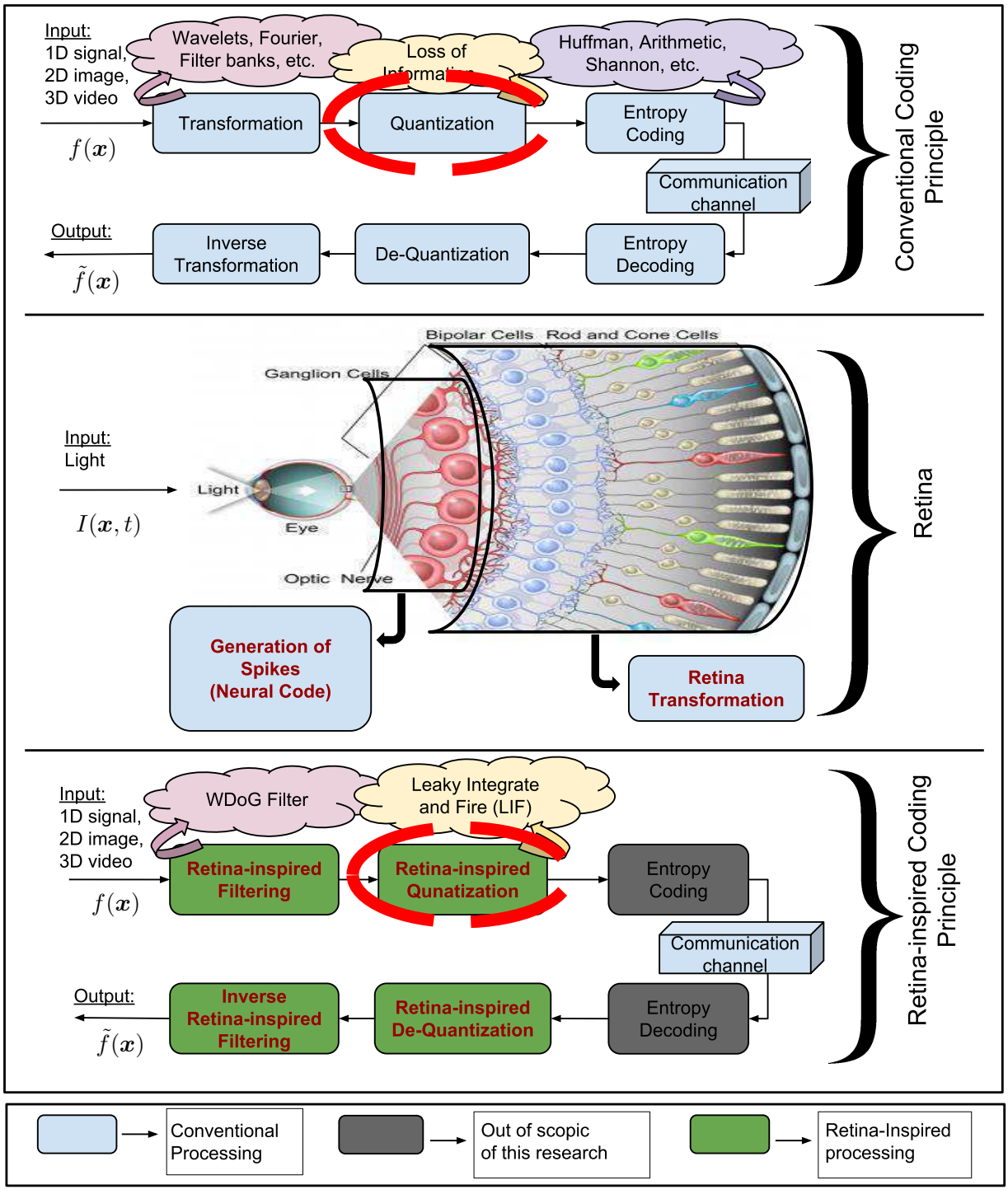


Figure 5.8: Motivation of the dynamic encoding. (a) Conventional coding principle which consists of a static quantization. (b) Generation of spike trains according to the dynamic generation of spikes based on the LIF model. (c) Retina-inspired coding principle based on the LIF model.

Chapter 6

Generation of Spikes

Contents

6.1 Introduction

This chapter is an introduction to the physical role of the ganglion cells which belong to the GL retina layer. These cells are the only retinal neurons which are able to produce a neural code of electrical impulses (spikes). This code is sent to the visual cortex of the brain to be further analyzed. The aim of the retina coding is to transmit enough information about the input stimulus on the retina to allow the identification of objects and events. This information can be locally conveyed by analog electrical mechanisms, but over long distances it should be encoded into spatiotemporal spike trains which are generated by a population of neurons. This structure is more efficient not only to prevent any lose of information but also to accelerate the speed of the transmission.

We are interested in exploiting the generation of spikes and consequently the code of spikes. Generally, the firing rate is considered to be a stochastic process. There have been proposed many models which approximate the statistical link between the input signal and

the code of spikes. Some of these models consider the ganglion cells as an interconnected network. This network produces some strong non-linearities in the way the spikes are generated for a given input signal. These non-linearities are very important to be included in a mathematical model which approximates the neural behavior even if their complexity is high. However, there are also other models which assume that neurons spike individually. On the one hand, these models are not very reliable but on the other hand, their complexity is lower and the way they link the stimulus and the sequence of spikes is much easier to be interpreted.

In this thesis, we are interested in adopting a model which describes how the neurons spike in the conventional coding principle (see Fig. 5.8). This model will reduce the spatiotemporal redundancy of the retina-inspired transformed input signal by generating the code of spikes. This code will be used to reconstruct the input signal with the minimum possible distortion. Thus, we need a system which allows to interpret the code of spikes by providing a link between the input signal and the firing rate. As a result, it seems that the second group of models where the neurons are considered to be independent (*independent spike hypothesis*) is easier to be adapted in a such a codec. This is the first attempt of image coding with neurons.

Through the prism of the independent spike hypothesis, there are several ways in which neuroscientists modeled the generation of spikes. The firing rate can be computed based on the Michaelis-Menten function (see section 6.3.1.1). Generally, the input signal is supposed to be described by the "mean firing rate" of the ganglion cells. A quick glance at the experimental literature shows that once the firing rate is generated there are different counting methods to compute the mean firing rate. Averaging the rate over time or over different repetitions or over a population of neurons are some possible ways to compute a rate code. It is clear, however, that these approaches neglect almost all the information which is possibly contained in the exact timing of the spikes. A strong argument to avoid this kind of codes is that the brain activity is very fast. Thus, it is impossible to produce a high number of spikes which allows the temporal averaging. In addition, the brain is unable to encode multiple times the same event under the same circumstances in order to average the firing rate. However, the rate codes have been widely used in literature and they deserve a short discussion in this thesis (see sections 6.3.1).

Another coding schema is based on the exact time each neuron emits its first spike. This time carries all the necessary information about the input such that even if the neuron continues to spike the rest of the spikes can be neglected. This time code is more efficient comparing to the rate codes especially if one considers the time constraint which is imposed for the propagation of the signal from the retina to the brain. To build such a time code we should imagine that a neuron is inhibited just after the release of its first spike. Since a neuron spikes only once then, it is clear that all the necessary information is conveyed by the time instead of the number of neurons (see section 6.3.2). The same assumptions with the time code shares also the rank code while instead of the delay of the the arrival of the first spike, the rank code computes the order the neurons spike (see section 6.3.3). In section 6.4 we compare the performances of the rate, time and rank codes under the time constraint in order to conclude which one interprets the spikes in the best way allowing the most faithful reconstruction of the input stimulus. Last but not least, this chapter finishes with section 6.5 which is dedicated to some related works in compression where spikes were also used.

6.2 Ganglion Cells

The morphology and the functional type of ganglion cells have been established long ago. Ganglion cells belong to the GL retina layer. They receive a signal from bipolar and amacrine cells which are parts of the OPL and IPL retina layers respectively. The ganglion

cells are the most distinctive retina neurons in terms of biochemical markers and their dendrites architectures are dramatically variable from species to species - while this variation is not seen for the cells which belong to the OPL. As a result, the taxonomy of the cells has been very challenging [Masland, 2011].

6.2.1 Morphology

The ganglion cells have a receptive field which is organized as two concentric circles. There are ON-center and OFF-center ganglion cells. ON-center ganglion cells are activated when a spot of light falls into the center of their receptive fields, whereas OFF-center ganglion cells fire in response to light falling on their fields' periphery leaving their center dark (in terms of simplicity, for the rest of this thesis we will only refer to the ON-center cells). Ganglion cells have also a receptive field with a Mexican-hat shape (modeled with a DoG), reflecting their integration of opposing information about centers and surrounds. Electrical recordings show that several types of ganglion cells do not have concentric organization, especially in animals whose eyes lack of fovea. This includes most non-mammalian species and mammalian species that have retinas with visual streaks.

In mammalian species there exist very small ganglion cells which are called *midget* ganglion cells, because of their tiny dendrite trees exist inside the mammalian fovea. *Fovea* is a tiny dimple in the retina which consists of smaller in size but very compact number of cones. When the light falls onto one of these cones which is connected in a one-to-one ratio with a single midget bipolar and the last one with midget ganglion cell, it relays point-to-point image- very sharp and brilliant copy of the input visual signal - to the brain [Kolb, 2004].

6.2.2 Functionality

In [Adrian, 1926, Adrian, 1928], the author was the first one who saw that an individual neuron is able to emit a spike. In addition, he specified that there is no other information which is propagated to the brain except for the time of spikes. The input signal which reaches a single neuron is able either to generate an action potential driving the neuron to spike, or to keep the neuron in silence. Secondly, another important remark of Adrian was that the intensity of the stimulus is indicated by the rate (or otherwise the frequency) of spikes. Thus, the higher the intensity of the stimulus, the higher the firing rate. This is the idea of rate coding (measuring the number of spikes within a fixed time window). Last but not least, the third discovery of Adrian was that while the stimulus is constant, the spike rate begins to decline. This is called *adaptation* and is an approach which is used in order to describe the phenomenon of becoming unaware of a constant stimulus basing on the history of the stimulation. Adrian's experiments on neural coding consist a large fraction about what we know about spikes which is the language of the brain [Rieke et al., 1999].

In the visual system, the rate coding is at first produced by ganglion cells in GL and it indicates the strength of the stimuli. Ganglion cells are the only retina cells which are able to emit spikes. However, it has been shown in [Thorpe and Imbert, 1989] that the neurons in primate brain are able to respond selectively 100-150ms after the stimulus is offset. In addition, 150 ms is enough to categorize a complex visual scene which was never seen before. Given that, there are approximately 10 different processing layers in the visual pathway between the photoreceptors and the visual cortex with an average processing time for each layer about 10ms [Perrett et al., 1982] (see Fig. 6.1). Given that, the cortical neurons rarely fire with a rate above 100Hz, meaning that each individual neuron fires either none or one spike. The processing speed is a strong constraint concerning the firing rate of neurons. Under this constraint some famous spike generation models, like the rate codes, become probably too inefficient to account the rapid information transmission. However, we are going to explain in the following sections that, without this constraint, these models

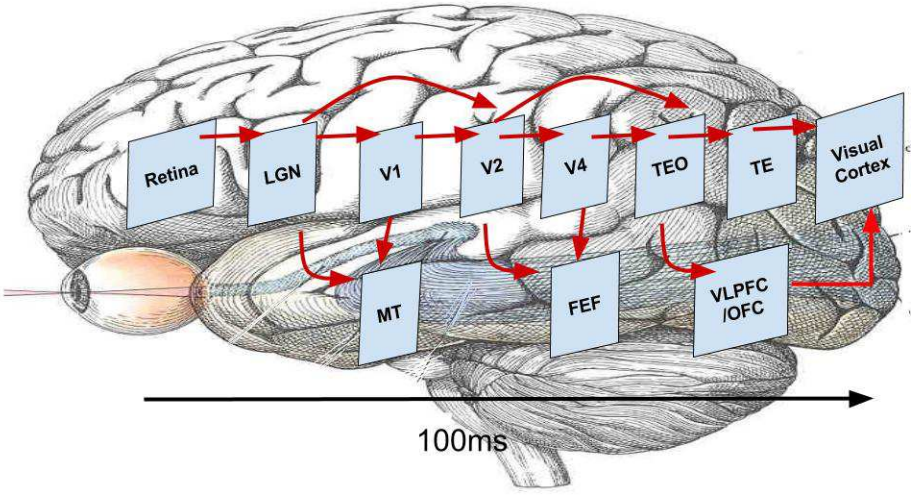


Figure 6.1: Visual pathway chain. This representation aims to represent the different processing layers into the visual pathway and it does not correspond to the real structure of these processing steps. In addition, studying the visual pathway is out of the scope of this thesis.

manage to eliminate noise over a large number of repetitions and produce a reliable rate code [Heeger, 2000]. Nevertheless, as we will explain in section 6.4, since the time constraint eliminates the reliability of the rate codes, people seek for other solutions like the time or the rank codes which seem to encode the input stimulus more reliably.

6.3 Spike Generation Models

The retina is a multilayer structure which takes as an input the visual stimuli $I(\boldsymbol{x}, t)$ where $\boldsymbol{x} \in \mathbb{R}^2, t \in \mathbb{R}$ and transforms it into a group of spikes. This emission of spikes happens inside the GL layer of the retina which consists of the ganglion cells. The emission of spikes in ganglions and their individual physical role has been overlooked during the last decades. In the early experiments of Adrian and Hartline, the response of a neuron was measured by counting the number of the emitted spikes in a fixed time observation interval following the onset of the input stimulus. In modern experiments one repeats the same stimulus many times in order to average the spike trains. However, it has been observed that the spike trains are not identical, which means that there is a degree of randomness in the neural response.

Due to the high number of possible spike sequences, we should rely on some statistical models that allow us to estimate the probability of an arbitrary spike sequence occurring, given our knowledge of the responses actually recorded. A firing rate $r(t)$ determines the probability of firing a single spike in a small interval around the time t. This rate, $r(t)$ is not in general a sufficient information to predict the probabilities of spike sequences. In reality, there is a statistical dependency between the spikes, since the presence of a single spike could effect the generation of another one. This dependency is due to the connection of the neurons which form a network that allows them to exchange information with each other. This is why a precise neural model should not consider neurons as individual cells. In that sense, a spike was seen as the mean to transfer faster over long distances a continuous

graded signal coming from the prior to ganglion cells. However, in such a case the theoretical framework to describe the generation of spikes was assigned either to complex non-linear dynamic models (i.e the Hodgkin-Huxley, FitzHugh-Nagumo, Mainen-Sejnowski, etc) which modeled the spikes as fast oscillations or to non-linear systems (i.e Gerstner, Izhikevich, etc) which approximate the spikes as explosions in finite time.

6.3.1 Rate Codes

To overcome the complexity of these non-linear systems and for interpretative reasons of the spikes, people assumed that each neuron is independent. The output of a spiking neuron is its firing rate. However, the time each neuron produces its spike train for a given stimulus is highly irregular. This irregularity might arise from some stochastic forces [Heeger, 2000]. In this case, the irregular interspike intervals reflect a random process and imply that an instantaneous spike rate (mean firing rate) can be obtained either by averaging the spikes of an individual neuron (spike count), or by averaging the firing rate over multiple repetitions of the same experiment (spike density) or by averaging the pooled responses of many individual neurons (population activity) [Gerstner and Kistler, 2002] (see Fig. 6.3). One way to compute the firing rate for a given input stimulus is to use the Michaelis-Menten function.

6.3.1.1 Michaelis-Menten Function

The Michaelis Menten function (see Fig. 6.2 A) is described as following:

$$r(I) = \frac{aI}{(b+I)}, \tag{6.1}$$

where r is the firing rate, I is the input contrast intensity, a is the maximum firing rate and b the intensity for which the firing rate is $a/2$. Based on the above model, the mean interspike interval or the delay, $d(I)$, between two spikes could be computed as:

$$\begin{aligned} d(I) &= \frac{1}{r(I)} \\ &= d_{ref} + \frac{b}{aI}, \end{aligned} \tag{6.2}$$

where $d_{ref} = 1/a$ is the refractory period when each neuron remains silent after the emission of the spike i and before the emission of the spike $i+1$. The best coding scheme to transmit the mean firing rate would be by a regular spike train with intervals $d(I)$. However, in practice this is impossible due to the interspike irregularities. The irregularities are considered as noise which is modeled using the Poisson process. Applying the Poisson process on the theoretical firing rate $r(I)$, one is able to obtain the spike train and build a rate code based on the average number of spikes. The smaller the number of the d_{ref}, the higher the benefit any of the three different rate codes because the neuron will spike its maximum rate a. Thus, the Poisson process estimates an accurate firing rate when the number of spikes is high (see Fig. 6.2 B). If the firing rate is independent of time then, the homogeneous Poisson process is used whilst, in different case, we use the in-homogeneous Poisson process.

6.3.1.2 Rate as a Spike Count

The definition of the rate which computes the mean firing rate r_m of a single neuron (see Fig. 6.3 (a)), counting the number of spikes n within an observation window T is given as following:

$$r_m = \frac{n}{T}. \tag{6.3}$$

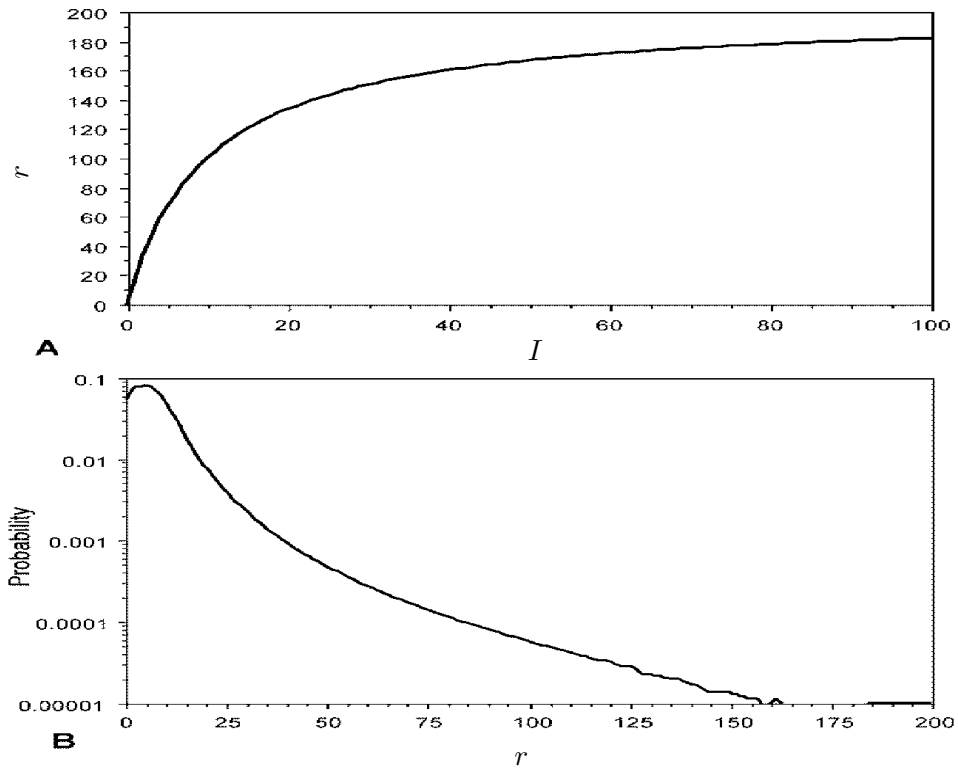


Figure 6.2: (A) Theoretical firing rate as a function of the input contrast. Michaelis-Menten function, parameters a/b=2000, $d_{ref} = 0.005$. (B) Distribution of firing rates as derived from the distribution of contrast levels, calculated over more than 3000 natural images (extracted from [Rullen and Thorpe, 2001]).

6.3.1.3 Rate as a Spike Density

The second case of rate code is illustrated in Fig. 6.3 (b), where given an input signal the mean firing rate is given by:

$$r_m(t) = \frac{1}{\Delta t} \frac{n_K(t; t + \Delta t)}{K}, \tag{6.4}$$

where K is the number of repetitions (in our example K=3), $n_K(t; t + \Delta t)$ is the number of spikes measured between time t and $t + \Delta t$. The number of spikes which are counted in $[t, t + \Delta t]$ is called spike density.

6.3.1.4 Rate as a Population Activity

The last rate code we are going to introduce is based on a population of neurons. The number of neurons within the brain is huge. Ideally, the neurons with the same properties should belong to the same population. Thus, the spikes of the population m should be sent to the population n. The mean population activity if depicted in Fig. 6.3 (c) and it is given by:

$$r_m(t) = \frac{1}{\Delta t} \frac{n_{act}(t; t + \Delta t)}{N}, \tag{6.5}$$

where N is the number of neuron in the population and $n_{act}(t; t + \Delta t)$ is the number of the active neurons within the interval t and $t + \Delta t$. In all the above scenarios, the precise time of an individual spike contains little information.

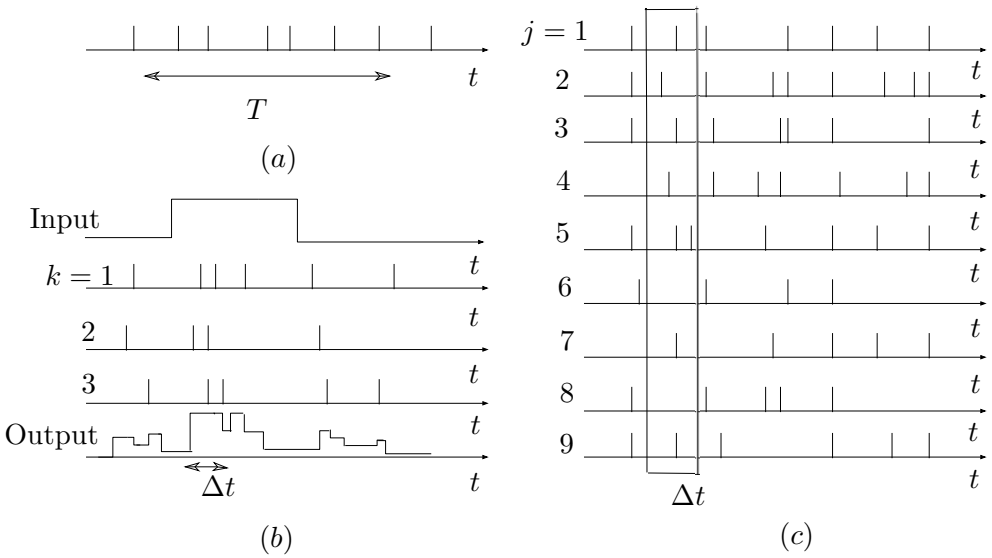


Figure 6.3: Rate Codes. (a) It concerns the mean firing rate of a single neuron. (b) It depicts the counting process of the mean firing rate over $K = 3$ repetitions of the same experiment. The higher the number of repetitions, the better the approximation of the input signal. Obviously, in this examples the estimation is poor because of the small number of repetitions. (c) This is an example of averaging the number of spikes over a population of $N = 9$ neurons for a given subinterval Δt.

6.3.2 Time Code: Leaky Integrate and Fire (LIF)

This section describes a time coder which stands under the assumption that the information which is hidden in each individual spike (the arrival time and/or the interspike interval) is sufficient to describe the input stimulus. Thus, the fastest a spike arrives, the strongest the visual stimuli. Of course, this rule has been proposed under a strong assumption that the neurons have a local sensitivity. This is true also in auditory systems for which the arrival time of a spike is used to estimate the distance of the auditory stimuli [Poggio and Koch, 1986]. In addition, according to [Meister and Berry, 1999], the retina is designed to eliminate redundancy. Thus, even if there is an overlap between ganglion cells it is considered to be relatively small. In fact, the overlapping happens but it takes place between different types of ganglion cells which means that each cell has a different functionality.

The LIF model [Gerstner and Kistler, 2002] is a very well know model which has been widely used in literature [Rieke et al., 1999, Wohrer et al., 2009, Masmoudi et al., 2013, Lazar and Pnevmatikakis, 2011, Jolivet et al., 2004, Cardarilli et al., 2013]. As it is already mentioned, the LIF model approximates the neural spiking mechanism. The basic circuit LIF model is given by:

$$I(t) = \frac{V(t)}{R} + C\frac{dV}{dt}, \tag{6.6}$$

where $I(t)$ is the input current, C the membrane capacitor of a neuron which is in parallel with the resistor R and $V(t)$ is the voltage across the resistor. If we multiply eq. (6.6) by R, we introduce the time constant $\tau_m = RC$ of the leaky integrator. This yields the standard form:

$$\tau_m \frac{dV}{dt} = V(t) + RI(t). \tag{6.7}$$

Whenever, the membrane potential of a neuron crosses a threshold θ, where $\theta > 0$, the neuron spikes. The moment the neuron spikes is called firing time. For a given threshold θ

a neuron spikes according to the following law:

$$\text{if} \quad V(t) \geq \theta \quad \text{and} \quad V(t) > 0 \Rightarrow t^f = t, \tag{6.8}$$

where t^f is the firing time of a neuron. Immediately after the emission of a spike the potential is reset to a value $V_r < \theta$. Since spikes are stereotyped events, i.e. with nearly identical shapes, they are fully characterized by their firing time.

Lets assume that LIF is applied to the input of a ganglion cell, which is constant for a given time T:

$$I(t) = I_0 \mathbf{1}_{[0 \leq t \leq T]}(t), \tag{6.9}$$

where $\mathbf{1}$ is the indicator function which is equal to 1 if $0 \leq t \leq T$, and 0 otherwise. The LIF has a double role: first of all, it sets a threshold θ which is the first criterion to decide if the coefficient will spike or not. Hence, if the value exits the threshold, the neuron will spike otherwise, it will remain silent. Secondly, the LIF generates the spike train which encodes the input signal I_0. This spike train requires the estimation of the delay $d(I_0)$ between each two sequential firing times t^i, t^{i+1}. This delay is the same for each spike of the spike train including the first one. This is obvious since after each spike a neuron is set to V_r and the firing process is repeated. In addition, the delay strongly depends on the intensity I_0. Thus, the stronger the input signal, the faster the emission of spike of the first spike which results in a small delay. On the contrary, a weak input signal requires large trigger time in order to spike. Assuming that the first spike arrives at time t^1, the trajectory of the membrane potential can be found by integrating eq. (6.7) with the initial condition $V(t^1) = 0$ (see eq. (6.10)):

$$V(t) \quad = \quad RI_0 \left[1 - \exp(-\frac{t - t^1}{\tau_m}) \right]. \tag{6.10}$$

The asymptotic value RI_0 in eq. (6.10) determines the generation of the spikes: if $RI_0 \leq \theta$ there is no spike, otherwise a spike arrives. The membrane potential reaches again the threshold θ at time t_2, which can be found from the threshold condition $V(t^2) = \theta$ (see eq. (6.11)).

$$\theta \quad = \quad RI_0 \left[1 - \exp(-\frac{t^2 - t^1}{\tau_m}) \right]. \tag{6.11}$$

Solving eq. (6.11) with respect to the delay $t^2 - t^1$ of the second spike, we derive a general formula for the delay given a constant input signal:

$$d(I_0) \quad = \quad -\tau_m \ln \left[\frac{RI_0 - \theta}{RI_0} \right]. \tag{6.12}$$

The LIF is an efficient model which also enables to take into account the refractory period of a neuron. Let suppose that a neuron spikes with a delay $d(I_0)$ for a given constant input stimulus I_0. Without any refractory period the firing rate of the neuron is $r = 1/d(I_0)$. However, after the emission of each spike there is an interval d_{ref} during which the neuron is unable to emit any spike. The definition of the firing rate r including d_{ref} is given by:

$$r \quad = \quad \frac{1}{d_{ref} + d(I_0)}. \tag{6.13}$$

The refractory period is out of the spectrum of this research. The only reason it is mentioned is to justify the reliability of the LIF model comparing to Poisson process which violates this neural feature. After neglecting the refractory period and simplifying the circuit notation, given a constant input value v for a given threshold θ, the delay $d(v)$ is given by:

$$d(v) = \begin{cases} +\infty & \text{if} \quad v < \theta, \\ h(v;\theta) = -\tau_m \ln \left[1 - \dfrac{\theta}{v} \right] & \text{if} \quad v \geq \theta. \end{cases} \tag{6.14}$$

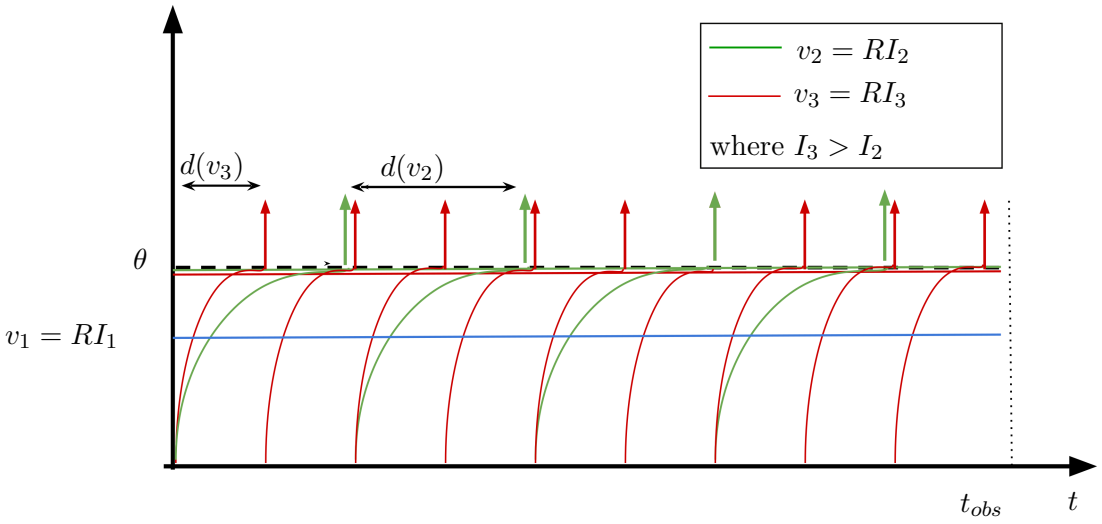


Figure 6.4: This figure illustrates the LIF model without any refractory period d_{ref}. For a given observation window t_{obs} and threshold θ, if the intensity $I > \theta$, the neuron spikes (i.e. I_2, I_3) otherwise it remains silent (i.e. I_1). The higher the intensity of the input v is ($I_3 > I_2$ which also corresponds to $v_3 > v_2$), the faster the first spike will be emitted ($d(v_3) < d(v_2)$) and the most compact the spike train will be. For this experiment the potential just after the emission of a spike is $V_r = 0$.

Figure 6.4 illustrates an examples of the LIF model for three different inputs $v_1(t)$, $v_2(t)$ and $v_3(t)$ where $|v_1(t)| > |v_2(t)| > |v_3(t)|$. For a given threshold θ, the intensities $v_1(t) > \theta$ and $v_2(t) > \theta$ are able to spike, so we are able to compute the delays $d(v_1)$ and $d(v_1)$ which correspond to the time the first spike appears. In addition, since the amplitude of $v_1(t)$ is higher than $v_2(t)$, the delay $d(v_1) < d(v_2)$. Consequently, the firing rate of $v_1(t)$ will be more compact than the one of $v_2(t)$. In contrast to v_1 and v_2, the third intensity $v_3(t) < \theta$ remains silent and its spiking delay $d(v_3) \approx \infty$.

6.3.3 Rank Code

Another coder which shares the same assumption with the LIF is the rank coder. The importance of the first spike plays also a key role to build this code. Once again, each contrast intensity is associated to a specific spike train. The arrival time of the first spike within a spike train depends on the intensity of the input stimulus I. Then, the contrast intensity is linked to the arrival rank of the first spike: a stronger stimulus corresponds to a fast arrival of a spike (low rank) while a weak stimulus results in a late or no response (high rank). Let I_1 and I_2, where $I_1 < I_2$ two different input intensities which are the inputs of two neurons i and j, where $i \neq j$ respectively. Then, the rank R of neuron j will be lower than the rank of neuron i ($R_i > R_j$). Each rank R is linked to a weight $w(R)$. The stronger the stimulus, the higher the weight ($w(R_i) < w(R_j)$). These weights were adjusted with a Look-Up-Table (LUT), which allows to look-up for the most likely intensity value with a given rank. This Look-Up-Table was experimentally defined after testing several grayscale images [Thorpe and Gautrais, 1998, Rullen and Thorpe, 2001, Perrinet et al., 2004].

6.4 How to interpret the spikes?

In this thesis, we are interested in building a codec which encodes and decodes an input stimulus using (and/or interpreting) spike trains. Thus, it is important not only to be able to produce a code of spikes but also it is necessary to use this code in order to reconstruct the input stimulus with the minimum distortion (see chapter 2, section 2.3.1). As a result, we need to interpret the spike trains. We have represented different coding schemes each one of which is efficient under different circumstances and constraints. In fact, the time constraint for a video codec is imposed due to the time T that a picture $f(\boldsymbol{x})$ of a video stream is flashed (the definition of a video stream $f(\boldsymbol{x}, t)$ as a sequence of N pictures $f(\boldsymbol{x})$ is given by eq. (2.22) in chapter 2). Hence, we need to use a coding scheme which performs well to this constraint.

According to [Thorpe et al., 2001], the rate codes are efficient when the observation window is sufficiently large resulting in a high number of spikes. However, under some processing speed constraints the interpretation of the spike train could be very poor. The reason is that for a very small observation window each neuron will be able to emit only one or none spike. Thus, counting only one or none spike is impossible to estimate a good firing rate. Fortunately, the rate coding is not the only coding schema. There exist the time and the rank codes which are more adapted to the time constraints. In the next section, we compare the rate, rank and time codes in terms of how faithfully they are able to represent the input stimulus.

Table 6.1 shows two examples of contrast intensities which are encoded by each coder under the time constraint. In the first example, the activation membrane threshold is $\theta_1 = 100$ whilst in the second one it is $\theta_2 = 9$. In other words, if the intensity is higher than the threshold the neuron will fire, otherwise it will remain silent. The reconstruction quality using the count rate code is impossible to be high, because the range of the input signal is roughly assigned to only two possible values. Even if one converts the counting process into a binary code where the emission of a spike is encoded using "1" and the silence using "0" the representation of the input stimulus would be also weak. The reason is simple, every intensity of the input stimulus will be labeled as spiking or silent using the counting code and "1" or "0" using the binary code (see Table 6.1). As a result, using these coders we will not be able to distinguish and recover the different input intensities.

The code of spikes could be more informative than the ones given by count and binary coders if one uses the rank encoder. If we focus only on the first example of Table 6.1, the rank coder computes a unique rank for each different intensity. As explained above, the higher the intensity the lower the rank. Consequently, one would expect that using the LUT each rank should be mapped to the correct intensity. However, if we now compare the rank code between the two examples it would be easy to conclude that the rank is exactly the same for two different intensity values. Thus, the reconstruction based on a unique LUT will be poor at least for one of the two examples. This LUT is the basic drawback of the rank encoder. The LUT has been built using a group of images with similar statistical properties. Hence, if these properties are different, the LUT will fail to match each rank to the correct reconstruction intensity.

Last but not least, it seems that the most accurate model is the time coder which encodes the delay of spike arrival. This delay is related to the intensity of the input signal. As a result, a high intensity would lead to a short delay and vise versa. An important remark is that the time coder requires an a priori knowledge concerning the time origin in order to compute the delay. Comparing the two intensity examples of Table 6.1 the time coder calculates a distinct delay value for each different intensity. Therefore, each delay will be able to correctly reconstruct the input intensity. The above comparison concludes that the LIF model is the most accurate encoder which allows to interpret the code of spikes and find out the best link between the interspike arrival and the input intensity.

Example 1

Intensity	Count	Binary	Rank	Time
155	spike	1	3	18
202	spike	1	2	13
220	spike	1	1	11
112	spike	1	4	22
99	-	0	0	0

Example 2

Intensity	Count	Binary	Rank	Time
10	spike	1	3	122
19	spike	1	2	111
20	spike	1	1	110
9	spike	1	4	129
8	-	0	0	0

Table 6.1: In this Table we present two sets of contrast intensities which are encoded by 4 different coders. The threshold for the left sequence is $\theta_1 = 100$ whilst for the right is $\theta_2 = 9$. The count coder reports if the input intensity is equal/higher the threshold resulting in the emission of spike or not. Similar code is generated by the binary coder with the difference that if a neuron fires, this is coded by "1" otherwise by "0". The rank coder encodes in which order the intensities trigger the neurons to spike. Last but not least, the time coder encodes the delay each intensity requires to activate a neurons. Comparing the two examples only the time coder enables to generate a unique code for each intensity.

6.5 Spikes in coding systems

During the last years, the usage of spikes as a mean to encode an input stimulus has gained a lot of interest. Spikes have been used in many different scientific fields like vision sensors, brain implant, scene recognition, compression, etc. The event-based Dynamic Vision Sensor (DVS) silicon retinas are some postprocessing devices which use asynchronous spikes to eliminate the redundancy between the pictures of a video stream and track motion [Lee et al., 2014]. Some recently released spine and brain implants are able to stimulate lumbar spines and recover paralyzed monkeys enabling them walking again. In fact, these implants record electrical signals (spikes) from the motor cortex. These signals are decoded and translated into commands in order to be sent to other electrodes implanted in the monkeys' lumbar spines and stimulate again the injured spinal cord allowing the monkeys natural movement commands to use their legs again.

In this thesis, we are interested in compression algorithms thus, we are going to provide some more details about related coding systems which are based on spikes. Thorpe was the first one who tried to build a codec, named as Rank Order Coder (ROC), based on spike trains in order to prove that a very small number of spikes is enough to identify objects and events inside a scene. Masmoudi *et al* noticed that Thorpe's model was very close to the coding principle which is used in image processing. Thus, they tried to improve any of its limitations and enhance the mathematical background by proving that not only the encoding but also the decoding pathway exists (see section 6.5.2). Time Encoding Machine (TEM) is another interesting codec which uses spikes based on the LIF model in order to faithfully reconstruct video streams (see section 6.5.3) [Lazar and Pnevmatikakis, 2011].

6.5.1 Rank Order Coder (ROC)

The ROC model was proposed by Thorpe [Thorpe, 1990] as a complete architecture of coding and decoding natural images. The ROC model [Thorpe, 1990, Thorpe and Gautrais, 1998, Rullen and Thorpe, 2001] is a bio-plausible generator and decoder of spikes (Fig. 6.5). The first step of the ROC model is the convolution of a still image, $f(\boldsymbol{x})$, with a spatial DoG pyramid given by:

$$A^k(\boldsymbol{x}) = DoG^k(\boldsymbol{x}) \overset{\boldsymbol{x}}{*} f(\boldsymbol{x}), \tag{6.15}$$

where $DoG^k(\boldsymbol{x})$ is given by eq. (3.16), $\boldsymbol{x} \in \mathbb{R}^2$, $\overset{\boldsymbol{x}}{*}$ denotes the spatial convolution and k is the layer index. Each layer k of this pyramid approxi-

mates a scale of the Center-Surround (CS) structure of the OPL cells (photoreceptor, horizontal and bipolar cells) [Rullen and Thorpe, 2001, Thorpe and Gautrais, 1998, Wohrer and Kornprobst, 2009]. Thorpe assumed that all the layers are fed simultaneously to the neurons in order to spike.

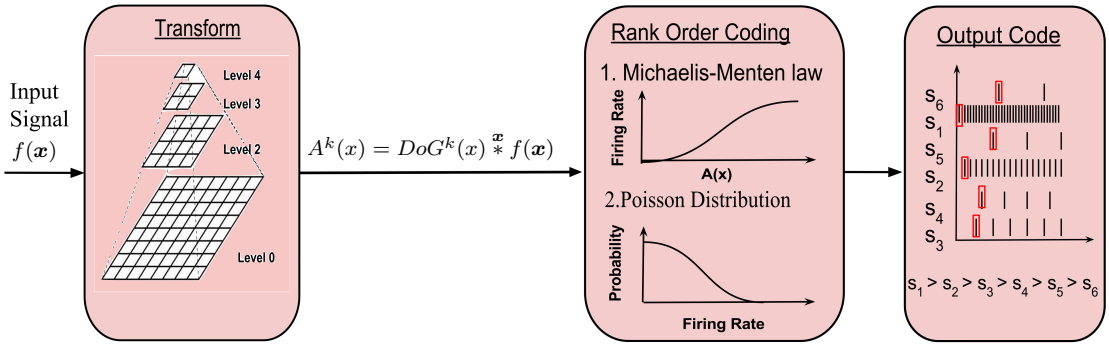


Figure 6.5: ROC encoding architecture: the input signal is transformed, then the firing rate deduced from the transformed signal is used to produce random spike trains. Each spike train is a Poisson process.

Let $A(\boldsymbol{x}) = (A^1(\boldsymbol{x}), \ldots, A^L(\boldsymbol{x}))$ be the input of the ROC model. Each contrast intensity $A^k(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i$ with $i = 1, \ldots, n$ is a given location, is converted into a spike train by a specific spiking neuron. For this purpose, the firing rate $r(A^k(\boldsymbol{x}_i))$ of the spiking neuron is given by the Michaelis-Nenten formula (see eq. (6.1)). Thorpe used a Poisson process to produce the spike train which encodes $A^k(\boldsymbol{x}_i)$. The Poisson process is a probabilistic mechanism known to induce noise on the precise timing of the firing events. Each contrast intensity is associated to a specific spike train. The arrival time of the first spike within a spike train depends on the intensity of the coefficient $A^k(\boldsymbol{x}_r)$ (see section 6.3.3). Thorpe argued that only 1% of the total number of spikes is enough to identify the input scene. Of course, the goal of the ROC was far from compression algorithms meaning that Thorpe did not aim to use spikes in order to decode the input signal with the minimum distortion.

6.5.2 Extension of ROC

The architecture of ROC model motivated Masmoudi *et al* to build the first bio-inspired coding system [Masmoudi et al., 2010]. However, they had to face ROC's limitations and detect possible solutions in order to first of all, improve its performance and secondly, prove that the decoding pathway is mathematically stable. First of all, the authors in [Masmoudi et al., 2012] proved that the spatial DoG pyramid is invertible introducing a rectification function (see details in 3.4.2). In addition, they improved the performance of the rank encoder proposing a different way to generate the LUT. As explained in section 6.4, the LUT of ROC was experimentally defined for a set of images of a given range. As a results, this LUT was impossible to be used for images of different range. Masmoudi *et al* proposed to create a LUT which is directly linked to the range of the input stimulus. This LUT ensures that the reconstruction values which are assigned at each rank will be close enough to the original input values.

6.5.3 Time Encoding Machine (TEM)

An interesting codec which is based on Integrate and Fire (IF) model was proposed by Lazar *et al* in order to faithfully reconstruct an input stimuli [Lazar and Pnevmatikakis, 2011]. The IF model is similar to the LIF without the leakage term. The new codec is called Time Encoding Machine (TEM) and it is illustrated in Figure 6.6. The TEM is applied to video streams $f(\boldsymbol{x}, t)$ (see eq. (2.22) in chapter 2) which belong to Ξ; a set of band-limited

functions. The video stream is first filtered by a kernel $\boldsymbol{h}^m : \mathbb{R} \mapsto \mathbb{R}^N$ for all neurons $m, m = 1, 2, \ldots, M$ and then it is biased by a constant amount $+(-)b$, which guarantees that the signal will be a positive (or negative) increasing (decreasing) function of time:

$$\begin{aligned} A(\boldsymbol{x}, t) &= [A_1(t), A_2(t), \ldots, A_M(t)] \\ &= \left[(h^1 * v)(\boldsymbol{x}, t) + b^1, (h^2 * v)(\boldsymbol{x}, t) + b^2, \ldots, (h^M * v)(\boldsymbol{x}, t) + b^M\right]^T \\ &= (\boldsymbol{h} * v)(\boldsymbol{x}, t) + \boldsymbol{b}, \end{aligned} \tag{6.16}$$

where T denotes the transpose. Each term of the transformed results is the input of the IF model $A_m(t) = I^m_{\text{Gang}}(t)$, which is also called by the authors "t-transform" and it is responsible to estimate the relation between the input and the output t^m_k of the TEM which denotes the spike train. This transform depends on a threshold $\theta^m = \kappa^m \delta^m$, where κ^m is an integration constant and $+(-)\delta^m$ an excitation (inhibition) threshold:

$$\int_{t^m_k}^{t^m_{k+1}} I^m_{\text{Gang}}(u)du = q^m_k = \theta^m - b^m(t^m_{k+1} - t^m_k), \tag{6.17}$$

for all $k \in \mathbb{Z}$ and all $m, m = 1, \ldots, M$. The authors showed that the "t-transform" can be

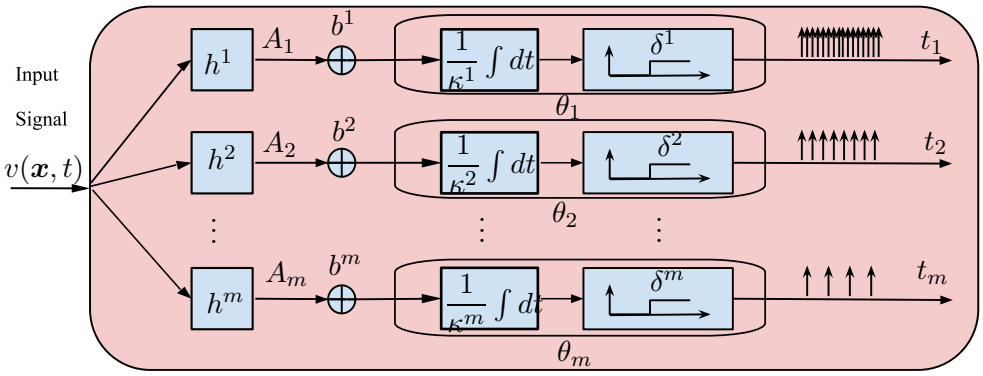


Figure 6.6: Time Encoding Machine [Lazar and Pnevmatikakis, 2011].

written as an inner product:

$$\langle v, \phi^m_k \rangle = q^m_k, \tag{6.18}$$

where $\phi^m_k = \tilde{\boldsymbol{h}}^m * g * \mathbf{1}_{[t^m_k, t^m_{k+1}]}$, where $\tilde{\boldsymbol{h}}^m$ is the involution of $\boldsymbol{h}^m$ and $g(t) = \sin(\Omega t)/\pi t$ is the impulse response of a lowpass filter with a cut-off frequency Ω. They also proved that the new function ϕ^m_k is invertible based on frame theory [Kovačević and Vetterli, 1992] which means that a faithful reconstruction can be achieved.

6.5.4 A/D Bio-inspired Converter

The authors in [Masmoudi et al., 2013] proposed a novel Analog to Digital (A/D) converter which consists of more reliable models concerning the neuroscientific background (i.e. the A/D converter used LIF instead of IF or ROC) and produces synchronous spikes instead of asynchronous which were used in TEM. The primary goal of the A/D converter is to reduce the bitrate encoding a scalar value which corresponds to the interval each spike arises. This is more efficient than encoding the exact spike arrival time which is a float number and requires higher number of bits to be encoded.

Generally the A/D converter uses the LIF model to generate the spike trains and the rate coding to encode the spike trains. For a given observation window $[0, t_{obs}]$, where $0 \leq t_{obs} \leq T$, the authors generated the spike train of each neuron and then, they counted

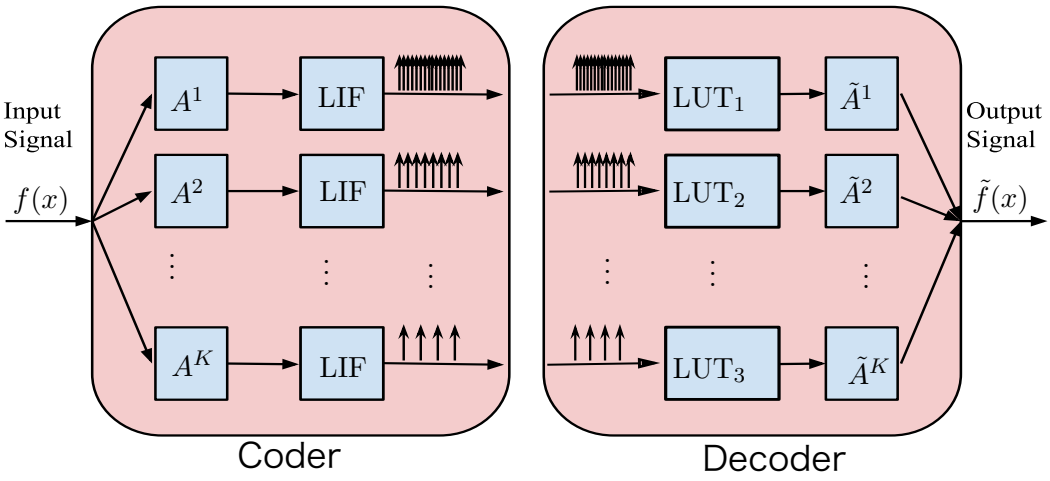


Figure 6.7: A/D converter using spikes.

the number of spikes N_s. This number increased while the observation window t_{obs} increases. The intensity of the input was strictly related to the number of spikes due to the LIF model. Thus, the higher the intensity, the higher the density of the firing rate. In this case, even for a small observation window t_{obs} the number of spikes will be higher for the more informative inputs. The A/D converter was the first complete encoding/decoding bio-inspired system.

Figure 6.7 shows how the A/D converter is applied to the input signal. The input image $f(x)$ is decomposed into several subbands A^k with $1 \leq k \leq K$ due to the Invertible Spatiotemporal DoG Pyramid which was introduced in section 3.4.3 (see eq. (3.19)). The authors applied the LIF model to each subband for a given observation window t_{obs} and they counted the number of spikes N_s each one of the intensities produced. Since, several different intensities may emit the same number of spikes, they assigned this number of spikes to the full range of these intensities. Moreover, each number of spikes is linked to a unique reconstruction value which was used in order to reconstruct all the input intensities which emitted this number of spikes. The number of spikes and its reconstruction value are saved to a LUT which is used in the decoding pathway.

6.6 Conclusion

This chapter was an introduction to the Ganglionic Layer (GL) of the retina tissue where the ganglion cells generate the retinal code. We presented different coding schemes to interpret the code of spikes like the rate, rank and time coders. Under the limitations of time the comparison of all these models showed that the time coder performs better in terms of how faithful the representation of the input signal is. A very well known neuromathematical model for time coding is the LIF which assumes that the time of the first spike arrival is the necessary information which enables to interpret the code of spikes and reconstruct a faithful representation of the input signal. According to the last section of this chapter, the LIF model is a good candidate which has been already used in compression schemes in order to encode in a bio-plausible way the input stimulus.

The above study was necessary for the next chapter of this thesis which is related to the retina-inspired quantization model. We aim to use the LIF model in order to build a dynamic quantizer inspired by the spiking mechanism of neurons. Generally, as we have already discussed the goal of this thesis is to build a retina-inspired video codec which adopts neuromathematical models proposed in order to explain how the retina transforms and encodes the visual stimulus. In chapter 4, we introduced the retina-inspired filtering

which approximates the OPL retina transformation. We have also proven in chapter 5 that this filter is invertible thus, it is suitable to be used in the coding principle (see Fig. 5.8). To integrate the retina-inspired codec we seek for neuromathematical encoding models which allow to reconstruct an accurate copy of the input signal interpreting spikes. To our point of view, the LIF model seems to be an efficient candidate to be adopted in the retina-inspired codec.

Chapter 7

LIF Quantizer

Contents

7.1 Introduction

According to the conventional coding principle (see section 2.2, Fig. 2.2), the necessary steps in compression are the transformation, quantization and entropy coding. In this thesis, we aim to build a retina-inspired coding principle (see Fig. 2.20). The retina-inspired transformation has been already presented in the chapter 4. It has been also proven in chapter 5 that this transform is a frame according to the frame theory hence, it enables a perfect reconstruction of the input signal. However, the retina-inspired frame is very redundant both in space and time. In conventional coding principle the redundancy is reduced using quantization (see section 2.4.3.3). The already existing quantizers are static, meaning that they should be applied to the full retina-inspired frame. In that sense, the dynamic properties of the retina-inspired filtering are eliminated.

In this chapter, we propose a retina-inspired quantizer being motivated by models which approximate the generation of the neural code, which is a code of spikes. These models are supposed to dynamically encode their input signal. In chapter 6, we have described several models which approximate this neural coding, the most efficient and accurate of which is the LIF (see section 6.3.2). Here, we aim to link the neuroscientific LIF model with the conventional quantization. This connection results in the construction of a retina-inspired quantizer, which is termed as *LIF dead-zone quantizer* or *LIF-quantizer* or *LIFQ*. We also explain how the LIFQ is applied to the retina-inspired frame in order to reduce its redundancy.

Depending on the value of the quantization step, we propose three different kinds of LIFQ: the first one is called *perfect-LIF dead-zone quantizer* or *perfect-LIFQ* which is

similar to the Integrate and Fire (IF) or Threshold And Fire (TAF) model. It is called perfect because a threshold θ is the only criterion to discard some intensities; coefficients above the threshold remain the same. Another more advanced model is the *uniform-LIF dead-zone quantizer* or *uniform-LIFQ*. In this model, the intensities which exit the threshold θ are quantized using a given quantization step q which is unique for all the decomposition layers. In addition, we present the *adaptive-LIF dead-zone quantizer* or *adaptive-LIFQ*, which adapts the value of the quantization step with respect to the energy of each decomposition layer. We first propose some experimental evolution of the value of the quantization step for each subbands, before we introduce the optimization of the bit-allocation method which tunes the quantization step according to the energy of each subband (see details in section 2.3). This is called *optimized-LIF dead-zone quantizer* or *optimized-LIFQ*. We present some numerical results to defend the efficiency of all the LIFQ models and we compare the performance of the retina-inspired codec, when it is applied to a still-image, to JPEG and JPEG2000 standards. Last but not least, we introduce some progressive reconstruction results where there is a progressive increase of the number of the retina-inspired decomposition layers which appear and are involved in the reconstruction. These results are important to show the improvement of the reconstruction quality with respect to time.

7.2 LIF Quantizer

In this section, we approximate the LIF using the dead-zone scalar quantizer which is a very well-known model in compression domain (see section 2.4.3.3) [Bhaskaran and Konstantinides, 1997, Taubman and Marcellin, 2002, Richardson, 2011]. Our motivation is to find out a way to assign spikes into intensity. We recall the definition of the LIF (see eq. (6.14)), without any refractory period d_{ref}, which has been introduced in details in section 6.3.2:

$$d(v) = \begin{cases} +\infty & \text{if} \quad v < \theta, \\ h(v;\theta) = -\tau_m \ln\left[1 - \dfrac{\theta}{v}\right] & \text{if} \quad v \geq \theta. \end{cases} \tag{7.1}$$

In the LIF model, each spike is described by its arrival delay $d(v)$. In addition, this delay is strongly related to the intensity v of the input. The arrival of the first spike for each input intensity corresponds to its magnitude. For instance, a high intensity will spike sooner than a lower intensity. As explain in chapter 6, the LIF is a temporal coder thus, it is based on the assumption of the importance of the first spike. However, let us suppose that the decoder "receives" the output of the LIF after an infinite observation window t_{obs}. As a result, each intensity will produce a spike train of a high density. This density is related to the delay of the first spike (see Fig. 6.4 in section 6). The density is described by the number of spikes N_s which is given by:

$$N_s = \left\lfloor \frac{t_{obs}}{d(v)} \right\rfloor, \tag{7.2}$$

The number of spikes depends on three parameters: the intensity of the input signal v, the value of threshold θ and the observation time t_{obs}. For given θ and t_{obs} the higher the intensity v, the smaller the delay $d(v)$. Moreover, the higher the value of θ, the less the number of the ganglion cells which are going to spike. Last but not least, the longer the observation time t_{obs} is, the more the spikes each ganglion cell is going to emit. In this thesis, we are interested in coding and decoding a video stream. As a result, the code of spikes should enable the reconstruction of the input intensity.

7.2.1 Decoding spikes

For the ideal scenario that the observation window is very large, the number of spikes N_s which is emitted for each different input intensity v will be efficient to precisely approximate the delay $\tilde{d}(v)$ according to the following formula:

$$\tilde{d}(v) = \frac{t_{obs}}{N_s}. \tag{7.3}$$

Based on the general formula of the LIF model which is recalled in the beginning of this section, if one knows the delay $d(v)$ is able to recover the input intensity v. Consequently, if we know the approximation of the delay $\tilde{d}(v)$, we will manage to reconstruct an approximation of the input intensity using the function $h^{-1}(\cdot;\theta)$ which is the inverse function of $h(v;\theta)$:

$$\tilde{v} = h^{-1}(\tilde{d}(v), \theta). \tag{7.4}$$

Figure 7.1 shows the reconstruction results based on the number of spikes which have been produced according to the LIF model for different values of threshold θ. In our experiments we have tuned the parameters such that the observation window that each subband is encoded will be $t_{obs} \in \{0.3, 1, 10\}$ ms. We observe that for very high values of θ only few of the ganglion cells are supposed to emit a spike, while at the same time the number of spikes of the higher intensity inputs is low. In such a case, the reconstruction encloses the contours of the scene which is enriched with some texture when the value of θ decreases. In fact, the value of θ is inversely related to the percentage of the total number of coefficients p which are activated. The lower the threshold, the higher the number (percentage %) of neurons which spike.

As explained in chapter 6, there have been already some attempts to use spikes in image and video compression algorithms. The LIF model was used as a spike generation mechanism which converts the input signal into spikes. Thus, each input intensity corresponds to a spike train. Counting the number of spikes, the authors in [Masmoudi et al., 2013] managed to reconstruct the input intensity. In fact, the authors managed to assign the number of spikes into intensity values using LUT. The accuracy of this reconstruction was related to the observation window t_{obs}. If we assume that an input intensity should be encoded and decoded within t_{obs}, the larger the observation window, the higher the number of spikes which allows a better estimation of the input intensity value. Masmoudi *et al* in [Masmoudi et al., 2013] proposed that the LIF model and the counting process could be approximated by the A/D converter (see section 6.5.4). This converter is a quantizer which evolves in time from a uniform to a non-uniform mode. In other words, a given input intensity enables the dynamic generation of spikes during an observation window t_{obs}. High intensity values correspond to high density spike trains which means high number of spikes. On the other hand, low values correspond to sparse codes of small number of spikes. The higher the number of spikes is, the better the approximation of the input signal.

According to the above analysis, it is easier to understand why the authors in [Masmoudi et al., 2013] approximated the LIF model by an A/D converter. This encoding process of counting the number of spikes is equivalent to a quantizer with a non-uniform quantization step (see Fig. 7.2). For a given observation window t_{obs}, the different input intensities are going to be quantized by the number of spikes. The higher the intensity, the more accurate the quantization.

We are also interested in approximating the LIF model by a dead-zone scalar quantizer Q_q^* in order to be well adapted to the compression systems. We recall the formula of the uniform dead-zone quantizer which was introduced in section 2.4.3.3 (see eq. (2.32)):

$$Q_q^*(v) = \mathrm{sgn}(v) q \max\left(0, \left\lfloor \frac{|v| - \theta}{q} + 1 \right\rfloor\right),$$

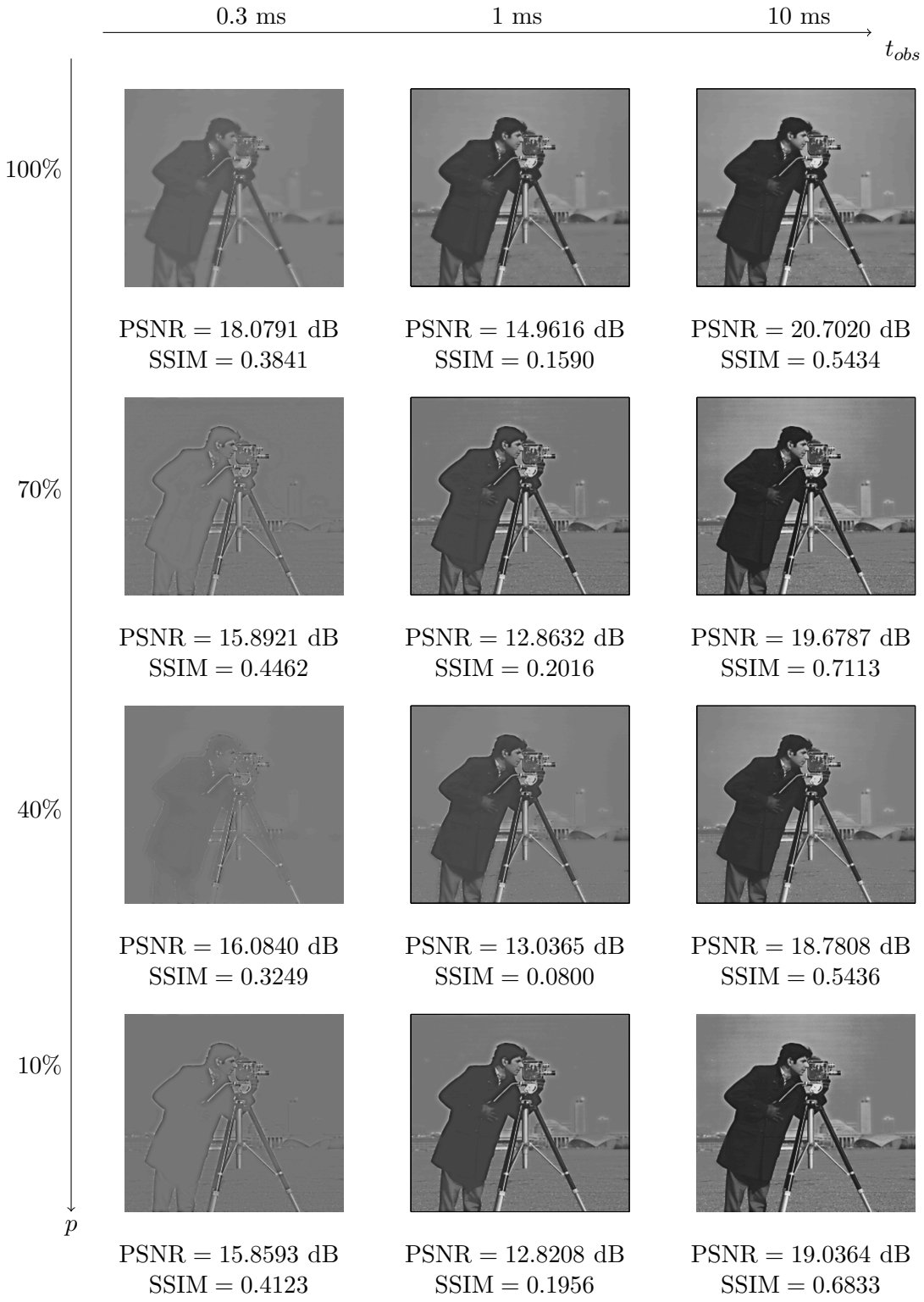


Figure 7.1: Decoding spikes based on equations (7.3) and (7.4). The figure illustrates reconstruction results for different values of p and t_{obs}.

where v is the input signal, $\text{sgn}(v)$ is the sign of the input value, θ is half the dead-zone and q the quantization step. The following section introduces an important contribution with respect to the dead-zone of the quantizer. We show that the dead-zone is related to the time constraint which is imposed by the input signal.

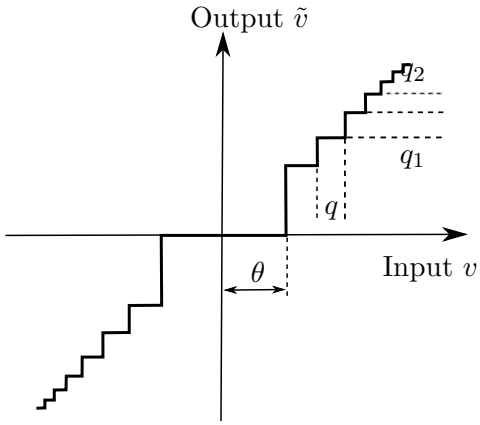


Figure 7.2: A/D converter as a non-uniform quantizer. For a given observation window t_{obs}, high intensities will be encoded better than the lower intensities using the LIF model.

7.2.2 Dead-zone

As explained before, for every coefficient $v < \theta$ the emission of spike is forbidden. Thus, this group of coefficients remains silent. As a result, a dead-zone is necessary to set to zero all the coefficients which are kept in silence due to the threshold θ. In addition, the dead-zone should also takes under consideration that the image is flashed for a given time T (see Proposition 1 in chapter 4). Thus, we assume that there is a maximum delay $d_{\max}$ during which the image should be reconstructed and we have chosen this delay to be related to the time the image is flashed, i.e. $d_{\max} = T$. In addition, since each image is decomposed into n subbands, there is a delay d_{obs} during which each subband should be encoded. This delay equals the observation window $d_{obs} = t_{obs}$ of each layer and it could be also given by:

$$d_{obs} = \frac{d_{\max}}{n} \tag{7.5}$$

For a given intensity v of a subband, according to the properties of $h(v;\theta)$, satisfying the observation delay d_{obs} is equivalent to encode only the values v whose intensity is larger than $\lambda = h^{-1}(d_{obs};\theta)$ where $h^{-1}(\cdot;\theta)$ is the inverse function of $h(v;\theta)$. A short calculation shows that:

$$\lambda = \lambda(d_{obs}) = \frac{\theta}{1 - \exp\left(-\dfrac{d_{obs}}{\tau_m}\right)}, \tag{7.6}$$

which involves that $\lambda > \theta$. For a given d_{obs} and since τ_m is a constant, λ depends on θ (see Fig. 7.3). As a result, the LIFQ is now given by:

$$Q_q(v) = \mathrm{sgn}(v) q \max\left(0, \left\lfloor \frac{|v| - \lambda}{q} + 1 \right\rfloor\right), \tag{7.7}$$

where λ is half the dead-zone.

Figure 7.4 represents how the LIF dead-zone quantizer (LIFQ) is applied to the retina-inspired frame. It describes that each decomposition layer A_{t_j} is the input of the LIFQ, Q_{q_j}, with quantization step q_j. For each input retina-inspired frame coefficient $A(\boldsymbol{x}_k, t_j)$, the output is the quantized value $A^q(\boldsymbol{x}_k, t_j) = Q_q(A(\boldsymbol{x}_k, t_j))$. There are several possible architectures to be studied but we have decided to apply the Q_q to each retina-inspired decomposition layer.

7.2.3 Quantization

In this section, we explain why the quantization is necessary to approximate the LIF model. The LIF model assigns an input intensity v to a unique time delay $d(v)$ and/or number of

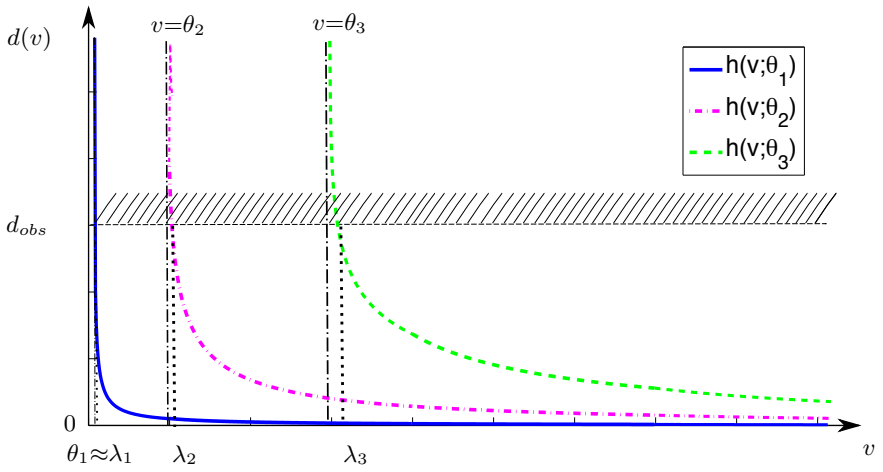


Figure 7.3: Delay $d(v)$ as a function of v: the quantization dead-zone $[0, 2\lambda]$ is imposed by d_{obs} for different $\lambda \in \{\lambda_1, \lambda_2, \lambda_3\}$ and $\theta_1 < \theta_2 < \theta_3$. Thus, for a given intensity v none of the neurons will be able to spike if $d(v) \geq d_{obs}$. For a given value of θ, while d_{obs} increases, λ turns to θ.

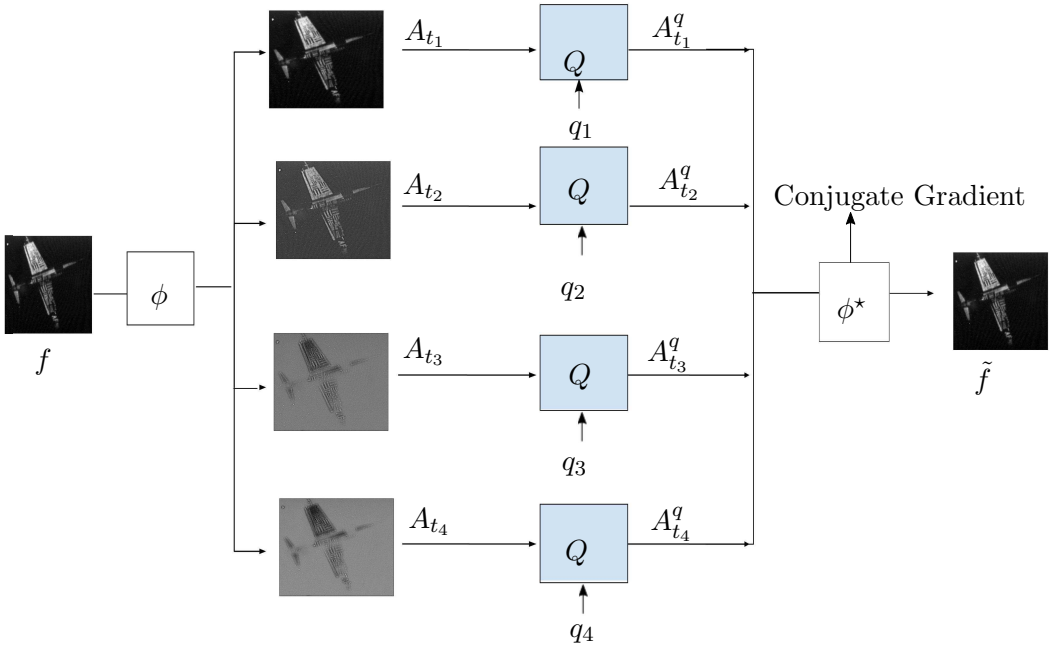


Figure 7.4: Retina-inspired quantizer LIFQ applied to the retina-inspired frame.

spikes N_s (see eq. (7.1) and (7.2) respectively). If one knows the delay $d(v)$ of the first spike and/or the number of spikes N_s, using the $h^{-1}(d;\theta)$ function, he can approximate the input intensity $\tilde{v}$. Lets now have a closer look at the estimation of the delay $\tilde{d}(v)$. For a fixed observation window t_{obs} the delay is given by:

$$\tilde{d}(v) = \begin{cases} \infty, & \text{if} \quad N_s = 0 \quad \text{and} \quad v < \theta \\ \dfrac{t_{obs}}{N_s} & \text{if} \quad N_s > 0 \quad \text{and} \quad v \geq \theta. \end{cases} \tag{7.8}$$

As a result $\tilde{d}(v) \in \left\{\infty, \frac{t_{obs}}{1}, \frac{t_{obs}}{2}, \ldots, \frac{t_{obs}}{k} \ldots\right\}$. Consequently, due to eq. (7.4) the reconstructed value $\tilde{v}$ is going to be given by:

$$\tilde{v} = \begin{cases} 0, & \text{if} \quad \tilde{d}(v) = \infty \\ h^{-1}(\tilde{d}(v), \theta) & \text{if} \quad \tilde{d}(v) < \infty. \end{cases} \tag{7.9}$$

Thus, $\tilde{v} \in \left\{0, h^{-1}\left(\frac{t_{obs}}{1}, \theta\right), h^{-1}\left(\frac{t_{obs}}{2}, \theta\right), \ldots, h^{-1}\left(\frac{t_{obs}}{k}, \theta\right), \ldots\right\}$. Figure 7.5 explains how the LIF model can be approximated by a quantizer. For a given input value which is below the threshold $v \leq \theta$, there will be no spike emitted $N_s = 0$, which means that the delay $\tilde{d}(v) \to \infty$. In such a case, all the values which belong to $c_0 = \{v \mid t_{obs} < d(v)\}$ will be encoded by the value $\tilde{v}_0 = 0$. Let us now suppose that only one spike arrives for the input signal, $N_s = 1$. Based on eq. (7.3) it means that $\tilde{d}(v) = t_{obs}$. All the input values which belong to the interval between t_{obs} and $\frac{t_{obs}}{2}$ will be decoded by the the value $\tilde{v}_1$. All the input values which belong to the interval between $\frac{t_{obs}}{3}$ and $\frac{t_{obs}}{2}$ will be decoded by the the value $\tilde{v}_2$, etc. Finally, one could define the cluster or in other words the quantization intervals as following:

$$\begin{aligned} c_0 &= \{v \mid d(v) > t_{obs}\} \\ c_k &= \left\{v \mid \frac{t_{obs}}{k+1} < d(v) \leq \frac{t_{obs}}{k}\right\}, \quad \forall k \in \mathbb{N}^+. \end{aligned} \tag{7.10}$$

An interesting remark related to the observation window t_{obs} is that the higher the value of t_{obs}, the smaller the size of the clusters, which leads to a higher precision in terms of reconstruction. Equivalently, for a given t_{obs} the higher v, the higher the precision of the reconstruction.

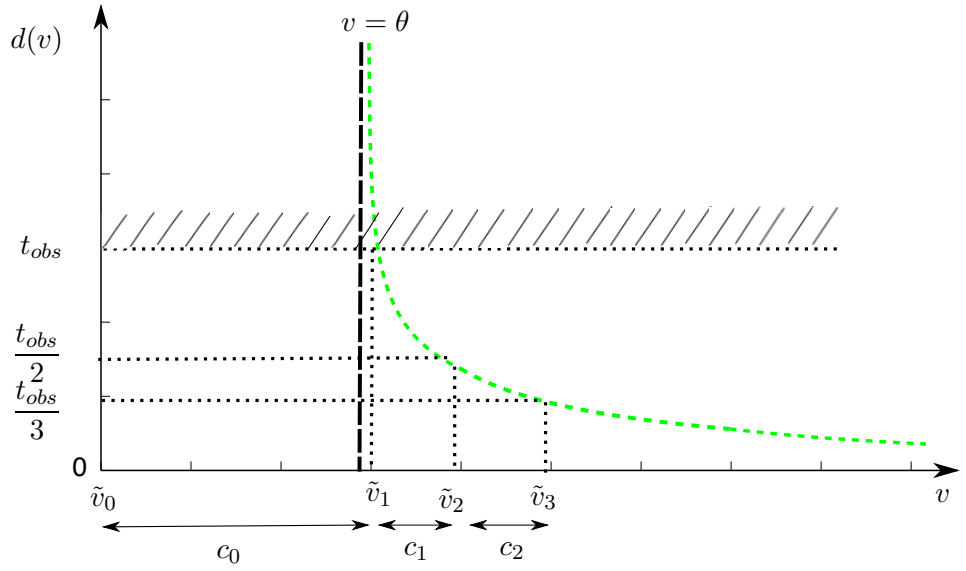


Figure 7.5: This figure introduces the notion of quantization in the LIF mode. Concerning the number of spikes N_s, the model provides different clusters/quantization interval which encode a group of input coefficients.

For a very large observation window t_{obs}, the number of spikes N_s is large enough such that it allows to perfectly reconstruct the delay $\tilde{d}(v) \approx d(v)$ and the input intensity $\tilde{v} \approx v$ according to Fig. 7.6 (a). This figure shows that half the dead-zone λ (see eq. (7.6))

is the only criterion to discard some intensities. Intensities which are higher than λ will remain the same. Concerning the LIFQ, the same behavior occurs when $q \approx \varepsilon$, with ε very small. Such a quantization step is interpreted by the number of spikes N_{s_1} which allows to reconstruct $\tilde{v}_1$ close to the asymptote $v = \theta$ in Fig. 7.5. This quantizer is called perfect-LIFQ and it eliminates some coefficients only with respect to λ. Now, if we reduce the observation window, we introduce some quantization to the LIF model which is depicted in Fig. 7.6 (b). Comparing this quantizer with the one of Masmoudi (see Fig. 7.2) the only difference is with respect to the bounded dead-zone 2λ. Let us suppose that for a given observation window t_{obs}, one should encode the input intensities. High intensities should be encoded better than the lower ones because ideally they would emit more spikes within the observation window t_{obs}. As a result, the quantization step should be adapted to the intensity value. In literature, there is the Lloyd quantizer where its quantization levels are at the center of mass of inputs probability density function between the corresponding decision levels, while decision levels are averages of neighboring quantization levels.[Lloyd, 1982].

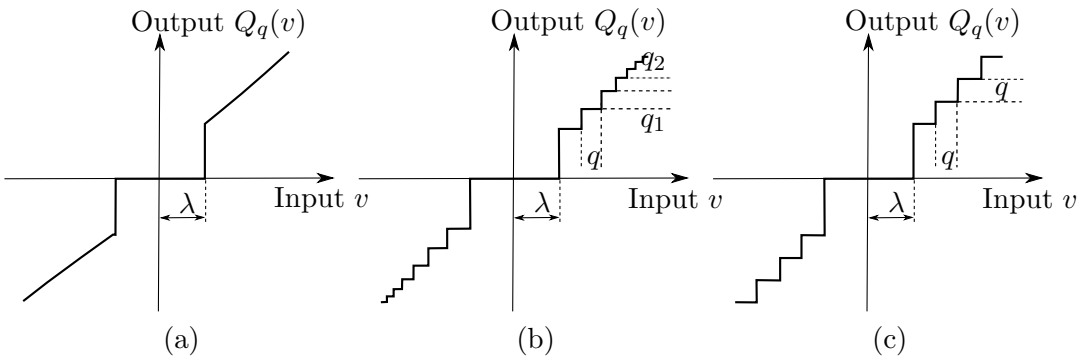


Figure 7.6: LIF dead-zone quantizer. (a) Perfect-LIF dead-zone quantizer, which discards values below a threshold λ while the rest are perfectly encoded. (b) Ideal LIF dead-zone quantizer where the quantization step varies according to the intensity of the input. (c) Uniform-LIF dead-zone quantizer, where the quantization step is unique for a retina-inspired decomposition layer.

In this thesis, in terms of simplicity, we first assume that all the coefficients of the retina-inspired frame are quantized in the same way. Hence, we define a uniform quantization step q (see Fig. 7.6 (c)). This model is called uniform-LIFQ because it uses a global quantization step q for each layer (i.e. in Fig. 7.4 $q_1 = q_2 = q_3 = q_4 = q_5$). Then, we extend the uniform-LIFQ into the adaptive-LIFQ. The adaptive-LIFQ uses different quantization steps (i.e. in Fig 7.4 $q_1 \neq q_2 \neq q_3 \neq q_4 \neq q_5$) for each retina-inspired layer. This quantization step is defined with respect to the evolution of energy and the bandwidth of each subband (see section 4.4). We implement and compare the perfect-LIFQ, the uniform-LIFQ and the adaptive-LIFQ to the LIF model. In addition, we propose the optimized-LIFQ where the quantization step has been optimized with respect to the rate-distortion theory (see section 2.3).

7.2.4 Perfect-LIF dead-zone Quantizer

This section studies the performance of the LIF dead-zone quantizer when the quantization step $q \to 0$. The goal is to discover the impact of the dead-zone 2λ on the quality of the reconstruction. We call this model perfect-LIF dead-zone quantizer because there is no quantization (see Fig. 7.6 (a)). The perfect-LIFQ is similar to the well-known model Threshold And Integrate (TAF). The TAF model takes as an input a sequence of coefficients v and using a threshold θ if $v \geq \theta$ it emits a spike otherwise ($v < \theta$) it remains silent. The intensities which are above the threshold are perfectly encoded. If we assume that the dead-zone $2\lambda = 0$ then, all the retina-inspired intensities will be able to spike. Moreover,

under no quantization and having proven that the retina-inspired filter is a frame, such a scenario would lead to the perfect reconstruction $f(\boldsymbol{x}) = \tilde{f}(\boldsymbol{x})$ which has been introduced in Chapter 5. However, while the dead-zone 2λ increases, there will be less and less activate coefficients thus, the reconstruction quality will be reduced.

The perfect-LIF quantizer model is a rough way to reduce the spatiotemporal redundancy of the retina-inspired frame. Similar approaches were introduced in [Thorpe et al., 2001, Masmoudi et al., 2012]. The authors proposed to eliminate some coefficients of their frames, keeping only the most informative ones, in order to study how the quality of the reconstruction is influenced and what is the amount of the active coefficients one needs to identify the object in the input scene. The method proposed in [Thorpe et al., 2001, Masmoudi et al., 2012] is completely equivalent to the Perfect-LIFQ if one considers that the value of λ is also inversely related to the percentage of coefficients p. Figure 7.7 shows some reconstruction results of perfect-LIFQ for different values of the threshold λ. The smaller the λ, the higher the percentage of the active coefficients. From the top to the bottom, Figure 7.7 illustrates the reconstruction results of cameraman image when p corresponds to 0.5%, 1%, 5%, 10% and 100% of excitatory coefficients. We compare the performance of the perfect-LIFQ when it is applied to the retina-inspired frame and other decomposition schema. The left column of Fig. 7.7 shows results of the perfect-LIFQ when it is applied to Thorpe's spatial DoG Pyramid (see section 3.4.1), the middle column corresponds to Masmoudi's spatiotemporal DoG pyramid (see section 3.4.2) and the right column illustrated the reconstruction results when the perfect-LIF quantizer is applied to the retina-inspired frame.

To evaluate the quality of the reconstruction we measure the PSNR value in dB (see section 2.3.1.2). Although, the PSNR results are lower in our case compared to the values of Thorpe's and Masmoudi's, one is easy to observe that the visual quality of the reconstruction in our case is higher. Starting from low to high p values (very small to large half the dead-zone λ) the first two approaches result in very blurred versions of the original signal which are enriched in details. However, the retina-inspired way performs differently: the basic contours of the objects of the scene are detailed even for small p values. In addition, when p increases there is a gain in the texture of the scene until we reach the perfect result. These results where obvious, since the higher intensities belong to the retina-inspired decomposition layers which are more assigned to the higher frequencies instead of lower frequencies. However, the low frequency copies of the signal belong to the very first retina-inspired decomposition layers which are of small intensities.

We have tested the perfect quantizer to 100 grayscale images of a size $n = 512 \times 512$ pixels, taken from the USC-SIPI database [Weber, 1977]. For a single image, the amount of time the algorithm needs in order to filter, quantize and reconstruct the input image is $\approx$ 20 sec using MATLAB running on a laptop with a 2.6 GHz Intel Core i7 processor, 8 GB 1600 MHz DDR3 memory and NVIDIA GeForce GT 650M 1024 MB graphics card. Table 7.1 shows some interesting results about the MSE which decreases while the amount p of the coefficients which are used for the reconstruction increases. This behavior leads the PSNR value to an exponential increasing rate (see Fig. 7.8 (a)). Except for the PSNR we confirm the efficiency of the combination of the retina-inspired filter and the perfect-LIF quantizer computing the SSIM which also increases while p decreases (see Fig. 7.8 (b)).

7.2.5 Uniform-LIF dead-zone Quantizer

In this section, we study the uniform-LIF dead-zone quantizer which performs according to Figure 7.6 (c) based on a uniform quantization step q. The smaller the quantization step is, the better the approximation of the intensity of the signal. One way to interpret the strong assumption of the uniform-LIFQ is to consider that the length of the observation window t_{obs} is inversely related to the intensity. The higher the intensity, the smaller the

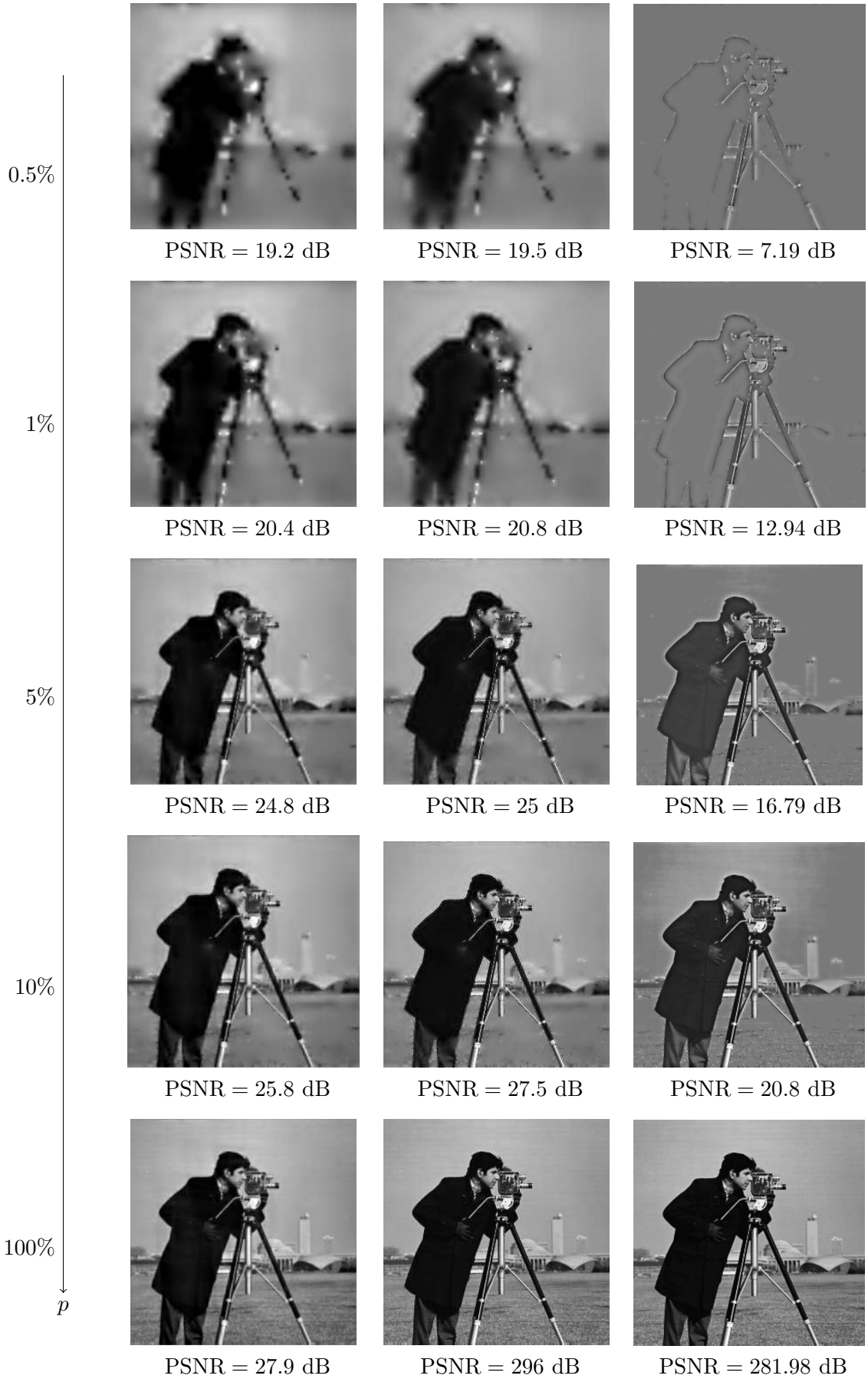
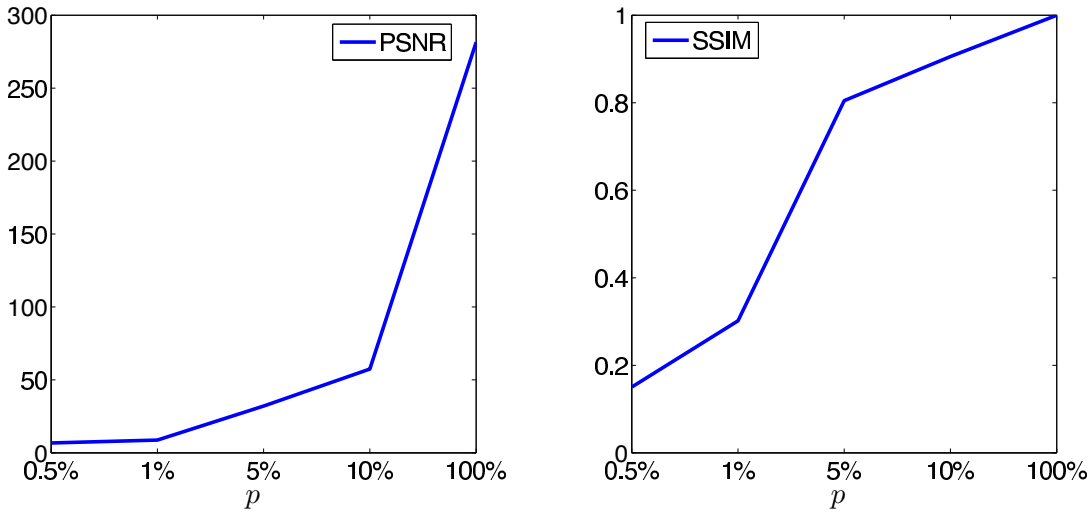


Figure 7.7: Reconstruction of an image using the perfect-LIF quantizer applied to Thorpe's DoG pyramid filter bank (left column), Masmoudi's spatiotemporal DoG pyramid (middle column) and the retina-inspired frame (right column). From the top to the bottom: The value of p increases corresponding to 0.5%, 1%, 5%, 10% and 100% of the total number of coefficients.

Percentage p	**0.5%**	**1%**	**5%**	**10%**	**100%**
Mean of ϵ	8.50×10^{9}	1.17×10^{10}	1.26×10^{10}	5.15×10^{9}	1.73×10^{-16}
Variance of ϵ	6.02×10^{20}	1.52×10^{21}	2.09×10^{21}	3.69×10^{20}	3.81×10^{-33}
Average Nbr of Iterations	7	8	10	12	748
Average PSNR	17.66	18.2	21.65	25.06	281.42
Average SSIM	0.10	0.16	0.43	0.64	1

Table 7.1: This Table shows the relation between the percentage of coefficients which participate to the reconstruction and the mean PSNR and SSIM metrics when the perfect-LIF quantizer is used. For these experiments we used 100 grayscale images of a size 512×512 pixels taken from the USC-SIPI database [Weber, 1977]. The value variable ϵ represents the Euclidean error which is measured during the conjugate gradient (see Algorithm 1) in chapter 5.



(a) (b)

Figure 7.8: This figure depicts the (a) PSNR and (b) SSIM metrics which evaluate the reconstruction results using the perfect-LIF quantizer. These are the average PSNR and SSIM values of 100 grayscale images of 8 bpp and size 512×512 pixels taken from the USC-SIPI database. For these experiments, we first applied to each image the retina-inspired filtering, the perfect-LIF quantizer and the retina-inspired reconstruction.

observation window. As a result, if the quantization step is small, every intensity will be encoded precisely. Otherwise, the estimations will be rough. We should remark that to be able to reduce the redundancy of the retina-inspired frame, the quantization step should be correspondent to most of the decomposition layers. Consequently, the uniform-LIFQ will completely discard the very first retina-inspired decomposition layers which are of a very low energy and process the rest of the layers according to their energy. This is reliable because the first layers are not able to activate the neurons due to their low energy.

To prove the efficiency of this model we show that using the PSNR metric, the number of bits required to reconstruct an image of quality > 30 dB is lower than 1 bpp. Consequently,

Figure 7.9: Reconstruction results using the uniform-LIF quantizer for different quantization steps q and widths of the dead-zone 2λ. From the top to the bottom, $q = \{1400, 1000, 600, 200, 1\}$ and from the left to the right, $p = \{100\%, 70\%, 40\%, 10\%\}$.

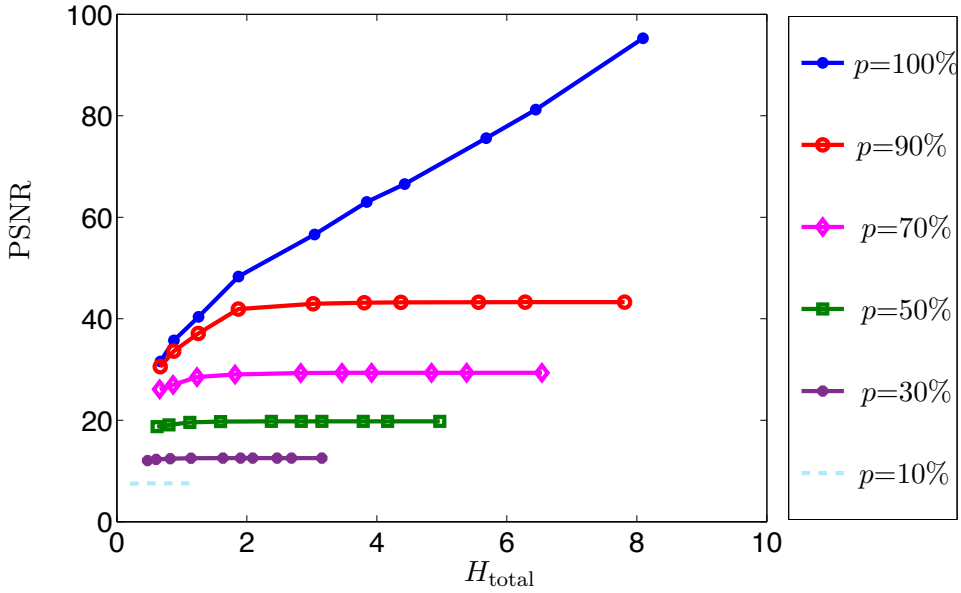


Figure 7.10: Uniform-LIFQ tested on 100 images taken form USC-SIPI database [Weber, 1977]. We compute the mean PSNR for different values of quantization step q and half the dead-zone λ ($q \in \{1400, 1200, 1000, 800, 600, 400, 200, 100, 50, 40, 30, 20, 10, 5, 4, 3, 2, 1\}$ and λ is controlled by the % of neurons p).

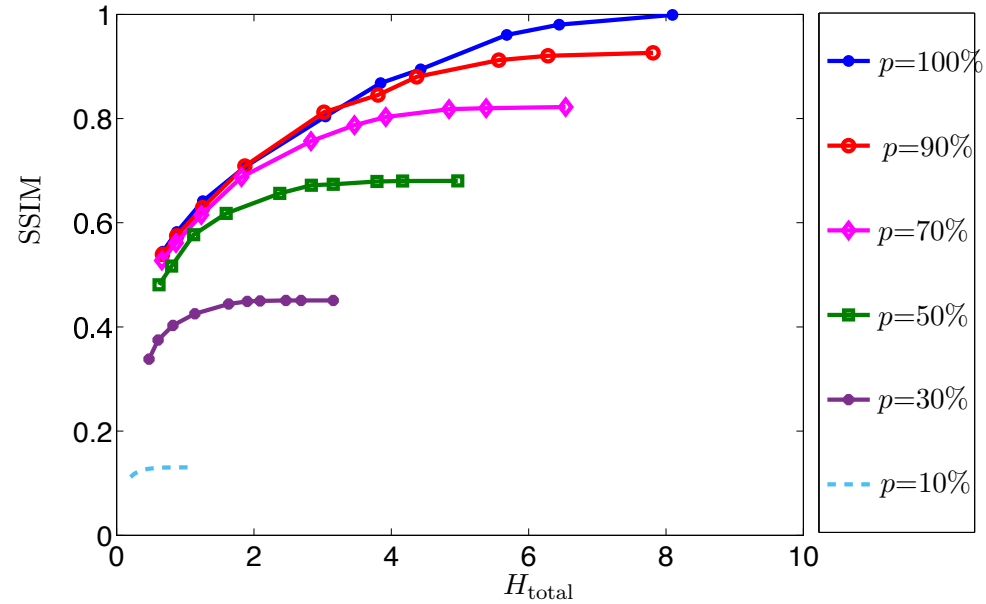


Figure 7.11: Uniform-LIFQ tested on 100 images taken form USC-SIPI database [Weber, 1977]. We compute the mean SSIM for different values of quantization step q and half the dead-zone λ ($q \in \{1400, 1200, 1000, 800, 600, 400, 200, 100, 50, 40, 30, 20, 10, 5, 4, 3, 2, 1\}$ and λ is controlled by the % of neurons p).

the retina-inspired quantization performs well in terms of compression. We first compute the number of bits using the Shannon Entropy for each retina-inspired decomposition layer which is introduced in Chapter 2 (see section 2.3.2). Then, we calculate the total entropy H_{total} which corresponds to the number of bits for the full retina-inspired frame according

to:

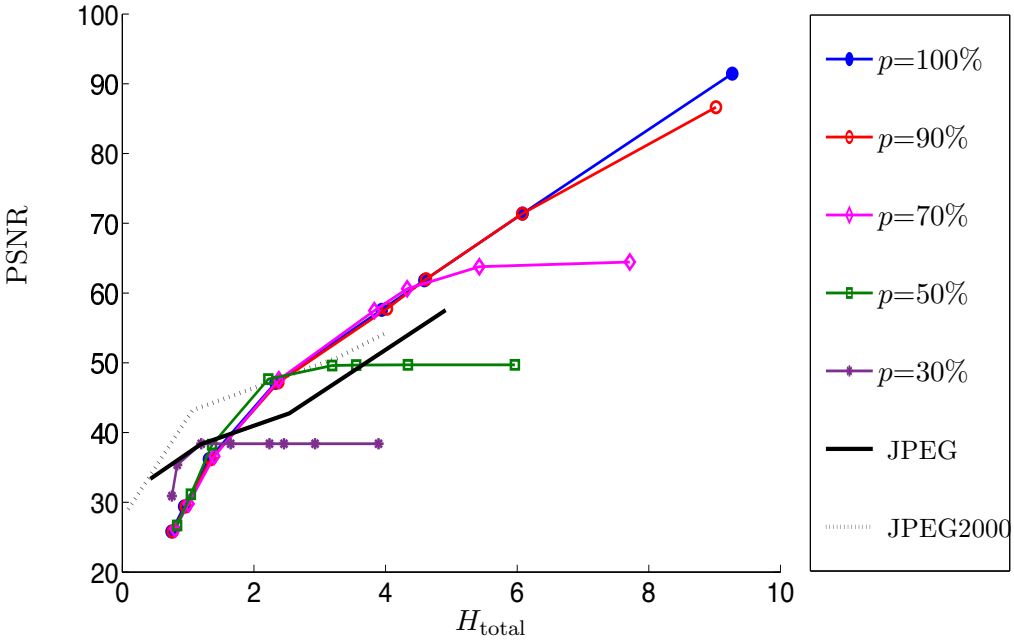


Figure 7.12: Uniform-LIFQ tested on lena image of a size 512× 512 pixels. PSNR for different values of quantization step q and half the dead-zone λ. ($q \in \{1400, 1000, 600, 200, 100, 50, 10, 5, 1\}$. The value of λ is controlled by the % of neurons p.)

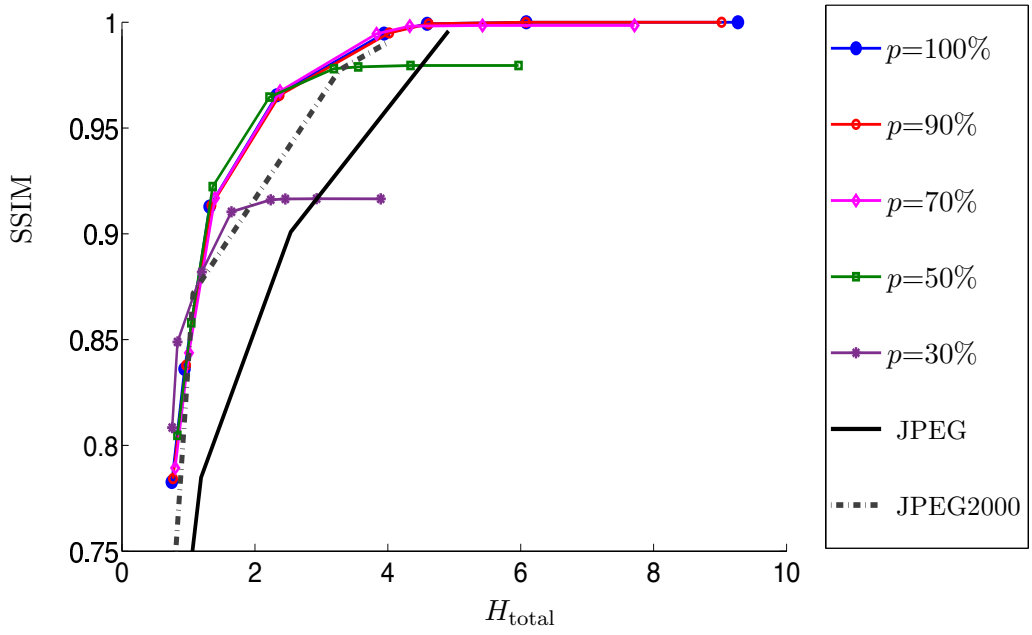


Figure 7.13: Uniform-LIFQ tested on lena image of a size 512× 512 pixels. SSIM for different values of quantization step q and dead-zone λ. ($q \in \{1400, 1000, 600, 300, 100, 50, 30, 10, 5, 1\}$. The value of λ is controlled by the % of neurons p.)

$$H_{\text{total}} = \frac{1}{m}\sum_{j=1}^{m} H_j, \tag{7.11}$$

where H_j is the Shannon entropy given by eq. (2.10), j stands for each decomposition layer and m is the total number of the layers. Figure 7.9 shows numerical results of the uniform-LIF dead-zone quantizer for different values of p and q. It is easy to observe that while the number of p increases the distortion between the original and the reconstructed image decreases. The Shannon entropy is a nice approximation method of the bitrate but we would like to provide some more accurate results in order to be comparable to standards.

Figures 7.10 and 7.11 show the performance of the uniform-LIFQ using the mean PSNR and the mean SSIM values respectively, for 100 images taken from the USC-SIPI database [Weber, 1977]. For a given dead-zone 2λ, while the quantization step decreases the mean value of quality of the reconstruction results increases. Moreover, while the size of the dead-zone decreases which means that the percentage p should increase, the quality also increases.

Figure 7.12 and 7.13 illustrate the evolution of PSNR and SSIM respectively versus H_{total} in function of different quantization steps q and widths of the dead-zone 2λ. While q decreases the quantity of the reconstruction increases, but what also gets higher is the number of bits which are required in order to store the signal. However, interestingly, we observe that the PSNR values are $> 30dB$ for very low total entropy values ($H_{\text{total}} \leq$1bits), which means that an acceptable reconstruction quality required less than 1bit to be stored while the original image would require 8bits to be stored. We also compare the performance of the uniform-LIFQ with JPEG and JPEG2000 standards. To provide fair results, we encoded raw grayscale images from USC-SIPI database using JPEG and JPEG2000. Then, we calculated the bitrate H_{total} just by dividing the total number of bits that each image requires on the memory disk by the size of the image ($n = 512 \times 512$ pixels).

Figures 7.14 and 7.15 compare the visual quality between the retina-inspired codec with the uniform-LIFQ and the JPEG and JPEG2000 standards for the "low" and the "medium" qualities. We do not provide any results concerning the "good" and "high" qualities because our codec outperforms the standards. According to Fig. 7.14 (a) and (b), our codec performs similarly to JPEG for very low bitrates while for higher bitrates the quality of the reconstructed image measured by PSNR is higher with the retina-inspired codec than JPEG (see Fig. 7.14(d) and (c)). The most interesting comparison is illustrated in Fig. 7.15 where especially for low bitrates, one is able to extract more details concerning the objects inside the living-room (i.e. the window and the flowers in the background, the armchair and the furniture where the phone is placed, the contours of the room, etc.). Thus, although our method provides lower PSNR values, the visual quality of the reconstructed image allows to detect much more details than JPEG2000. Of course, while the bitrate increases (see Fig. 7.15 (c) and (d)) the details also increase.

7.2.6 Adaptive-LIF dead-zone Quantizer

The strong assumption concerning the quantization step of the uniform-LIFQ drove us to build the adaptive-LIFQ dead-zone quantizer. We propose that the quantization step q should vary according to the energy of each retina-inspired decomposition layer. One solution would be to encode more precisely the layers which enclose more data (i.e. the layers which have been lowpass filtered, see chapter 4) than the layers which have been bandpass filtered. Another solution would be to discard the lowpass decomposition layers, because their energy is very small, and keep only the high energy layers. To develop this method each decomposition layer j is linked to a weight w_j which tunes the value of the quantization step. We call this quantizer adaptive-LIFQ.

(a)
JPEG
PSNR = 31.0993 dB
$H_{\text{total}} = 0.5798$ bpp
"low" quality

(b)
Retina-inspired Codec
PSNR = 31.0257 dB
$H_{\text{total}} = 0.5821$ bpp
$p = 20\%$ and $q = 1200$

(c)
JPEG
PSNR = 31.9132 dB
$H_{\text{total}} = 0.6713$ bpp
"medium" quality

(d)
Retina-inspired Codec
PSNR = 36.0666 dB
$H_{\text{total}} = 0.6795$ bpp
$p = 20\%$ and $q = 800$

Figure 7.14: Visual comparison between the retina-inspired codec with the uniform-LIF quantizer and JPEG standard for "low" and "medium" qualities.

(a)
JPEG2000
PSNR = 25.7915 dB
$H_{\text{total}} = 0.121$ bpp
"low" quality

(b)
Retina-inspired Codec
PSNR = 18.8031 dB
$H_{\text{total}} = 0.192$ bpp
$p = 5\%$ and $q = 5000$

(c)
JPEG2000
PSNR = 31.60 dB
$H_{\text{total}} = 0.3967$ bpp
"medium" quality

(d)
Retina-inspired Codec
PSNR = 22.3693 dB
$H_{\text{total}} = 0.4462$ bpp
$P = 20\%$ and $q = 2500$

Figure 7.15: Visual comparison between the retina-inspired codec with the uniform-LIF quantizer and JPEG2000 standard for "low" and "medium" qualities.

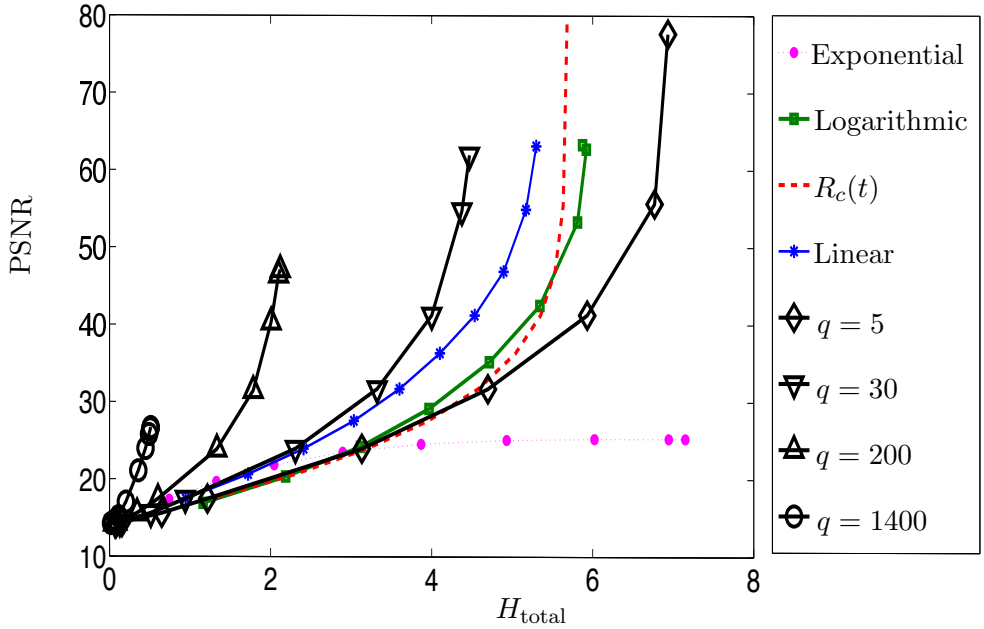


Figure 7.16: The figure shows the adaptive-LIFQ tested on lena image of a size 512× 512 pixels for the linear, exponential, logarithmic and $R_c(t)$ weight cases. We compare these results to four different cases of the uniform-LIF with $q \in \{5, 30, 200, 1400\}$ and $p \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$. In this experiment, each curve which corresponds to the uniform-LIF quantizer corresponds to a given q for different p values.

Some first results concerning the bit allocation between the retina-inspired decomposition layers are shown in Figure 7.16. For these experiments we assume that for a given decomposition layer j, the quantization step q_j depends on the energy of each decomposition layer with 4 different possible ways:

1. Linear: $q(j) = j$
2. Exponential: $q(j) = \exp(j)$
3. Logarithmic: $q(j) = \log(j)$
4. According to the temporal function $R_c(t)$ (see eq. (4.3)): $q(j) = \dfrac{1}{R_c(j)}$

Figure 7.16 shows results of PSNR vs the total Entropy, H_{total}, using the adaptive-LIFQ when q changes for each layer A_{t_j}. Concerning the width of the dead-zone, it is tuned such that the percentage of coefficients which are active will be $p \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$. According to the figure, the optimal trade-off between the bitrate and the PSNR is given by the linear case and the worst case is the exponential. We compare the adaptive-LIFQ with the uniform-LIFQ of 4 different quantization steps $q \in \{5, 30, 200, 1400\}$ and $p \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$. The uniform-LIFQ performs better than the adaptive-LIFQ if one compares the PSNR values for given bitrates. This is normal since the functions which are used in the adaptive-LIFQ are just an approximation of the optimal solution of q_j. To optimize this solution people try to find the trade-off between the distortion and the bitrate (see chapter 2).

7.2.7 Optimized-LIF dead-zone Quantizer

Let $D_j = D_j(R_j)$ be the Rate-Distortion (RD) curve for a given retina-inspired decomposition layer j (see details in section 2.3). We assume the MSE (see (eq. 2.1)) as the metric of the distortion D_j between the original and the decoded subband j and the Shannon entropy H (see eq. (2.10)) to estimate the rate R_j. We also assume that the RD curve is convex which is a realistic assumption since when the rate R_j increases, the distortion D_j decreases (see Fig. 2.9). In the bit allocation problem, the constraint is related to the total bitrate, R_{total} (see eq. 7.11) which should be bounded by a maximum rate, $R_{\max}$, such that:

$$R_{\text{total}} \leq R_{\max}. \tag{7.12}$$

In our problem, both the distortion D_j and the bitrate R_j are functions of (λ, q). As a result, the analytic expression of the rate optimization problem could be defined as following:

$$\begin{aligned} J(\lambda,q) &= D(\lambda,q) + \mu\ R(\lambda,q) \\ &= \frac{1}{m}\sum_{j=1}^{m} w_j D_j(\lambda_j,q_j) + \mu\ \frac{1}{m}\sum_{j=1}^{m} a_i R_j(\lambda_j,q_j) \\ &= \frac{1}{m}\sum_{j=1}^{m} w_j \text{MSE}_j(\lambda_j,q_j) + \mu\ \frac{1}{m}\sum_{j=1}^{m} a_i H_j(\lambda_j,q_j), \quad \mu \geq 0. \end{aligned} \tag{7.13}$$

The goal is to find for each subband j the values of q_j and λ_j such that the function $J(\lambda, q)$ will be minimized. The above optimization problem has a solution when eq. (7.14) comes true. Then, by varying the Lagrange operator μ, one can span the graph of $D_j(R_j)$ and find the optimal alignment of the RD curve for different complexities.

$$\begin{aligned} \frac{\partial J(\lambda,q)}{\partial q} &= 0 \Rightarrow \\ \frac{w_j \partial D_j(\lambda_j,q_j)}{a_j \partial R_j(\lambda_j,q_j)} &= -\mu, \Rightarrow \\ \frac{\partial D_j(\lambda_j,q_j)}{\partial R_j(\lambda_j,q_j)} &= -\chi_j \mu, \quad \forall j = \{1,\ldots,m\} \quad \text{and} \quad \mu \geq 0. \end{aligned} \tag{7.14}$$

where $\chi_j = \dfrac{a_j}{w_j}$ with $w_j \neq 0, \forall j$. We are aware of works which focus on the analysis of the statistical distribution of the transformed signal which is highly important for optimization [Perrinet, 2010, Perrinet, 2015]. However, this study is out of the scope of this manuscript. In this section, we would like to confirm that the performance of the retina-inspired codec is promising and it could be improved even if one uses a very naive optimization method. In that sense, instead of proving what is the statistical distribution of the signal, we aim to approximate it using already known models. Let us assume for seek of simplicity that there is a relation between the values of the dead-zone and the quantization step. In [Parisot, 2003] the authors prove that if the distribution of the signal which is quantized is a Laplacian, Gaussian or Generalized Gaussian then, there is a relation between the dead-zone 2λ and the quantization step q. Here, we assume that the distribution of the decomposition layers can be approximated by a Laplacian or a Gaussian function such that the relation of the dead-zone and the quantization step will be given by $2\lambda = 2q$ or $2\lambda = q$ respectively for high bitrates. Figure 7.17 shows the distribution of some decomposition

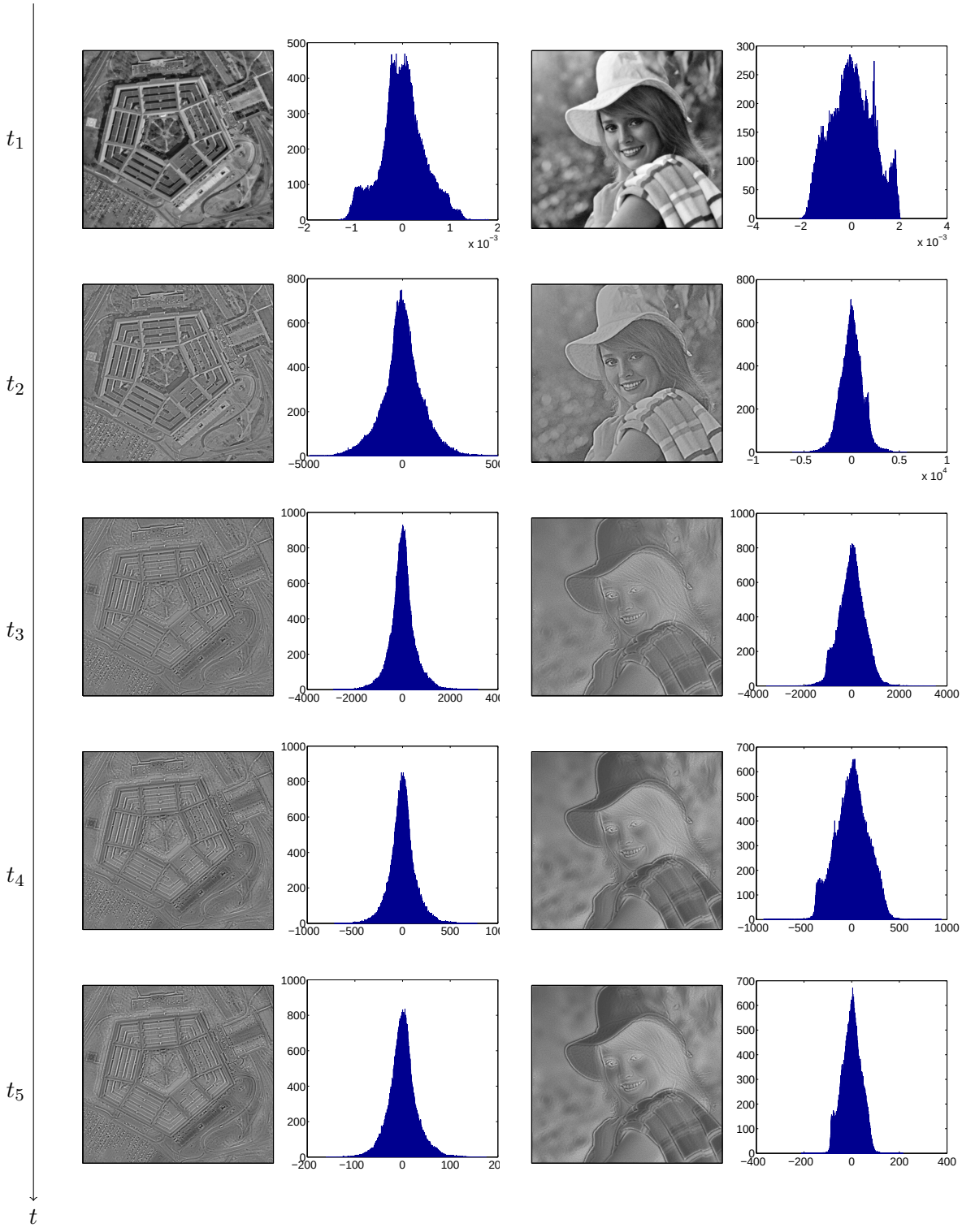


Figure 7.17: The retina-inspired decomposition layers which is biased to be centered at zero. We assume that the above distributions can be roughly approximated by the Laplacian or the Gaussian function. If one would like to find out the best model which approximates the distribution of the decomposition layers, he should study the distribution using the Kolmogorov-Smirnov test.

layers which could be roughly approximated by both Laplacian or Gaussian. Under this assumption, optimizing the value of q leads also to the optimization of the value of λ.

In practice, in order to compute the optimal RD curve, we need first to build the point-cloud of different pairs of D_j and R_j for each subband (see Fig. 2.9). Then, since the

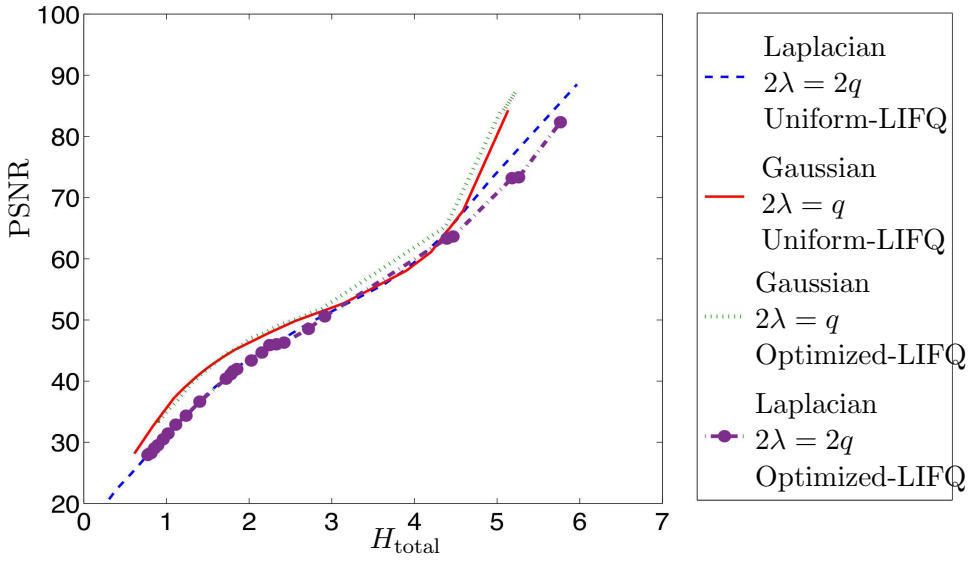


Figure 7.18: This figure compares the performance of the optimized-LIFQ and the uniform-LIFQ under the same condition with respect to the relation between the dead-zone 2λ and the quantization step q. We tested two different cases based on [Parisot, 2003]: (a) when the distribution of the subband is a Laplacian function leading to $2\lambda = 2q$ and (b) a Gaussian function leading to $2\lambda = q$. The performance of the two LIFQs is almost the same. For this experiment we used lena image of size 512×512 pixels at 8 bpp.

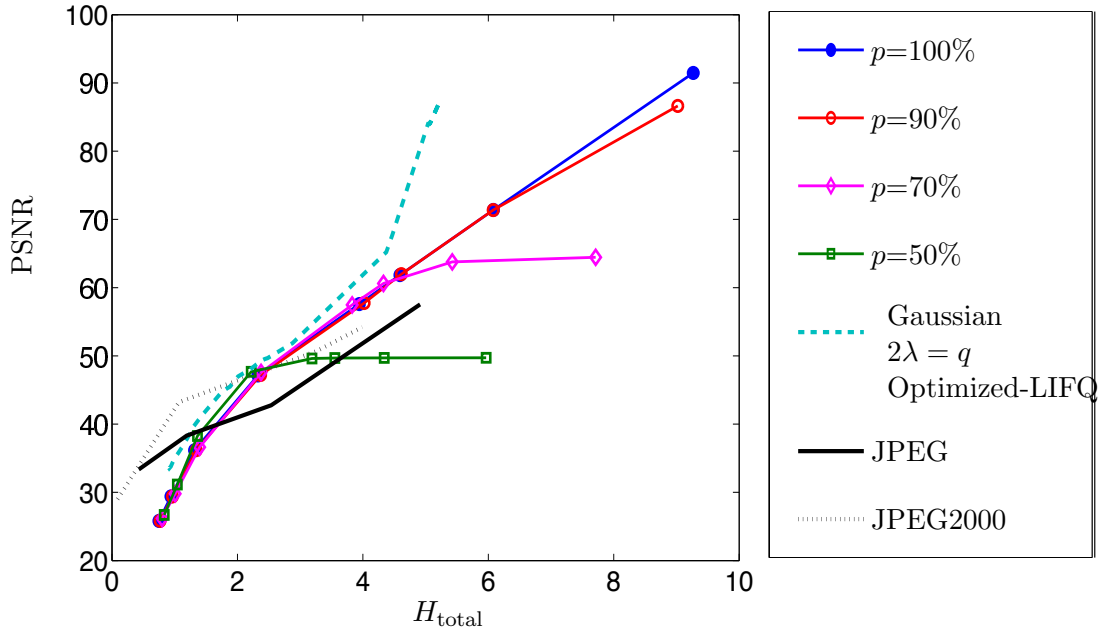


Figure 7.19: This figure compares the performance of the optimized-LIFQ for $2\lambda = q$ and the uniform-LIFQ when λ is tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and $q \in \{1400, 1000, 600, 200, 100, 50, 10, 1\}$. The performance of the the optimized-LIFQ is better than the uniform-LIFQ. In addition we compare the optimized-LIFQ to JPEG and JPEG2000. Our codec outperforms both the standards for bitrates ≥ 2 bpp. In addition, the retina-inspired codec results in higher PSNR values comparing to JPEG for bitrate ≥ 1 bpp. For this experiment we used lena image of the size 512×512 pixels at 8 bpp.

Algorithm 2 Find the μ of each subband

1: **for** `j = 1:m` **do** ▷ m is the total number of decomposition layers
2: **for** `q = 1:2000` **do**
3: $q_1 = q + \varepsilon$ ▷ Quantization Step
4: $q_2 = q - \varepsilon$ ▷ Quantization Step
5: $\lambda_1 = q_1$ ▷ Half deadzone for Laplacian distribution
6: $\lambda_2 = q_2$ ▷ Half deadzone for Laplacian distribution
7: $Q_{q_1}^{\lambda_1}(v) = \mathtt{sgn}(v)\max\left(0, \left\lfloor \frac{|v| - \lambda_1}{q_1} + 1 \right\rfloor\right)$ ▷ LIFQ
8: $Q_{q_2}^{\lambda_2}(v) = \mathtt{sgn}(v)\max\left(0, \left\lfloor \frac{|v| - \lambda_2}{q_2} + 1 \right\rfloor\right)$ ▷ LIFQ
9: $D_{q_1}^{\lambda_1} = \mathtt{MSE}(Q_{q_1}^{\lambda_1}(v), v)$ ▷ Compute the the Distortion for q_1
10: $D_{q_2}^{\lambda_2} = \mathtt{MSE}(Q_{q_2}^{\lambda_2}(v), v)$ ▷ Compute the the Distortion for q_2
11: $R_{q_2}^{\lambda_2} = H_{q_1}$ ▷ Compute the Shannon Entropy for q_1
12: $R_{q_1}^{\lambda_1} = H_{q_2}$ ▷ Compute the Shannon Entropy for q_2
13: $\mu_j(q_j, \lambda_j) = \frac{D_{q_1}^{\lambda_1} - D_{q_2}^{\lambda_2}}{R_{q_1}^{\lambda_1} - R_{q_2}^{\lambda_2}}$ ▷ Compute the Lagrange operators μ
14: **end for**
15: **end for**

optimal Lagrange operator μ is unknown, we need to compute several values of μ. Each μ value is assigned to a given pair of (q_j, λ_j) for each subband j which corresponds to a point of the RD curve. We provide a pseudo-algorithm (see Algorithm 2) which describes how we have computed the different values of $\mu^j(q_j, \lambda_j)$ for each retina-inspired decomposition layer j. Finally, if we select a value of μ we are able to find the correspondent q_j and λ_j for each decomposition layer and then compute the Lagrange optimization criterion (see eq. (7.13)).

Figure 7.18 illustrates the performance of the optimized-LIFQ compared to the uniform-LIFQ for two different cases: when the deadzone $2\lambda = 2q$ which corresponds to the Laplacian distribution and $2\lambda = q$ which corresponds to the Gaussian distribution [Parisot, 2003]. For this experiment we have used exactly the same conditions with respect to the relation between the dead-zone and the quantization step for the uniform-LIFQ. According to the results, we come to the following two conclusions: first of all, the Gaussian distribution ($2\lambda = q$) approximates better the distribution of the retina-inspired decomposition layers. In addition, the optimized-LIFQ performs almost the same with the uniform-LIFQ under the same conditions. One would expect that for a given bitrate the optimized-LIFQ would result in higher PSNR values. However, we interpret our results according to the strong assumption we made concerning the weighted factor χ_j (see eq. 7.14). In our experiments we used $a_j = 1, \forall j$ due to the fact that all the retina-inspired subbands have the same size and $\chi_j = 1$ which is true only for orthogonal filters. However, the retina-inspired filter is not orthogonal as a result $\chi_j = \frac{1}{w_j}$. In [Usevitch, 1996], the author provides some solutions concerning the estimation of χ_j for biorthogonal filters which could be probably a possible solution to lead the results into higher gain in terms of bitrate. Figure 7.19 shows the comparison between the optimized-LIFQ and the uniform-LIFQ when the second one is tuned without any relation between the dead-zone and the quantization step, according to section 7.2.5. As it is expected, the optimized-LIFQ outperforms the uniform-LIFQ. In

the same plot we also compare the optimized-LIFQ to JPEG and JPEG2000. Our codec is more efficient than JPEG for bitrates ≥ 1 bpp and JPEG2000 for bitrates ≥ 2 bpp.

7.3 Progressive Reconstruction

As explained in chapter 4 the retina-inspired filter is a spatiotemporal transform which results in a multilayer decomposition. Each one of these layers is a temporal sample which arises due to the spatial transform of an input still-image (picture of a video stream) with the corresponding retina-inspired DoG filter. We have proven in chapter 5 that the full retina-inspired decomposition is a frame. Thus, using all the retina-inspired subbands, one is able to perfectly reconstruct the input signal. In chapter 5 we have also introduced the notion of progressive reconstruction when, in the absence of noise, we are able to perfectly reconstruct the input signal keeping only few of the first decomposition layers (see Fig. 7.20).

$t = 20$ ms	$t = 30$ ms	$t = 40$ ms	$t = 50$ ms	$t = 100$ ms
PSNR = 16.02 dB	PSNR = 18.34 dB	PSNR = 21.20 dB	PSNR = 26.30 dB	PSNR = 296 dB
SSIM = 0.48	SSIM = 0.55	SSIM = 0.65	SSIM = 0.84	SSIM = 1
$H = 0.006$ bpp	$H = 0.077$ bpp	$H = 0.23$ bpp	$H = 1.39$ bpp	$H \approx 8$ bpp

$t = 20$ ms	$t = 30$ ms	$t = 40$ ms	$t = 50$ ms	$t = 100$ ms
PSNR = 12.85 dB	PSNR = 13.42 dB	PSNR = 15.80 dB	PSNR = 33.12 dB	PSNR = 283 dB
SSIM = 0.0519	SSIM = 0.0991	SSIM = 0.1282	SSIM = 0.7688	SSIM = 1
$H = 0.006$ bpp	$H = 0.077$ bpp	$H = 0.23$ bpp	$H = 1.39$ bpp	$H \approx 8$ bpp

Figure 7.20: Progressive Reconstruction. The top line shows results of the progressive reconstruction using Masmoudi's invertible spatiotemporal DoG pyramid (see section 3.4.3) and his A/D converter. The bottom line shows the retina-inspired filtering and the uniform-LIF dead-zone quantizer. ($p = 100\%$).

The progressive reconstruction has an important meaning in terms of compression especially when one is interested in dynamic encoding/decoding. In that sense, this section is oriented to show that the uniform-LIFQ performs better comparing to other bio-inspired models like the A/D converter proposed in [Masmoudi et al., 2013]. The A/D converter was applied to still-images which were flashed for a given time. Figure 7.20 compares some progressive reconstruction results between the A/D model and the uniform-LIFQ. We managed to achieve the same bitrates while the number of the decomposition layers was increasing and we were interested in the reconstruction quality. Although the PSNR and the SSIM values are higher in case of the A/D converter, the visual results show that our system performs better in terms of contrast. Obviously, without any a priory knowledge

of the input signal, the uniform-LIFQ tested for $t = 20$ ms is able to depict the foreground of the scene.

7.4 Conclusion

In this chapter, we have introduced the LIF dead-zone quantizer (LIFQ) which is an approximation of the LIF neural spiking mechanism. Depending on a threshold, the LIF model computes what is the delay each input intensity needs to emit the first spike of the spike train. If we know the delay, we are able to reconstruct the input intensity. The LIFQ performs in the same way using the dead-zone to decide which coefficients of the input signal are active. The quantization however is also necessary in order to reduce the number of bits which is required to store the real values of the time delay.

We studied different cases concerning the quantization step q. The simplest model was the perfect-LIFQ, where the reduction of the spatiotemporal redundancy was achieved only by the dead-zone. The uniform-LIFQ introduces a uniform quantization step for all the retina-inspired decomposition layers and the adaptive-LIFQ in which the quantization step changes with respect to the range of the decomposition layers. Last but not least, we describe what is the optimization RD method one should apply to the retina-inspired frame in order to optimize the reconstruction results.

The retina-inspired filter and the LIFQ build the retina-inspired codec. This codec has been applied to still-images which are flashed for a given time. Comparing the reconstruction results between the retina-inspired codec and JPEG and JPEG2000 we conclude that our coding system performs beyond the standards for bitrates higher than 1 bpp for JPEG and 2 bpp for JPEG2000 for lena image while the visual quality of the reconstruction is better for lower bitrates. We also compare the retina-inspired codec to other bio-inspired coding systems concerning their dynamic behavior. The progressive reconstruction using the retina-inspired codec enables better description of the scene even for very low bitrates.

Part IV

APPLICATIONS

Chapter 8

Application on Video Surveillance Data

Contents

In this chapter, we introduce video surveillance systems as an application of the retina-inspired codec. We have chosen these systems due to the 4G-TECHNOLOGY, which is the industrial partner of this thesis. The 4G-TECHNOLOGY is a group of experts who work on video surveillance systems. They have released the EViBOX which is an efficient machine designed to provide a 24h survey of public and remote areas. To be easier to understand the architecture of EViBOX, we need first to provide a general state-of-the-art in video surveillance over Wireless Sensor Networks (WSN). We represent the most common network infrastructure for video transmission over wireless channels and technologies for video capture and compression of the ultimate goal to maximize the received video quality under the resource limitations. These technologies provide power-efficiency solutions which is the major concern of nomadic video surveillance systems. Moreover, this chapter represents the architecture of EViBOX and its advantage with respect to other systems. Last but not least, we discuss why the retina-inspired codec is beneficial with respect to the current format of EViBOX, how it is applied to the system and which are the benefits of adopting this novel codec in video surveillance systems.

8.1 Video Surveillance over WSN

Video Surveillance over WSN are used in various cyber-physical systems including traffic analysis, healthcare, public safety, wildlife tracking and environment/weather monitoring. In current systems, each source node is usually equipped with one or more cameras, a microprocessor, the storage unit, a transceiver, and a power supply. The basic functions of each node consist of the capture of the video, the compression and the transmission. However, for video surveillance systems with real-time demands the processing and the transmission over wireless channels of large amount of data is really challenging.

In literature, there are famous video surveillance systems all over the world monitoring different scenarios like the traffic system of Irving in Texas which is implemented by seventy pan-tilt-zoom (PTZ) CCTV (closed-circuit television) cameras [Leader, 2004], the traffic monitoring system at the University of Minnesota (UMN, 2005) [Hourdakis et al., 2005], or the system of University of North Texas which was also used for traffic surveillance [Luo, 2011]. Other systems have been used for weather monitoring like FireWxNet [Hartung et al., 2006], the Smart Camera Network System (SCNS) which was used for security monitoring in a railway station [Kawamura et al., 2011], for indoor surveillance system in a multi-floor department building at the University of Massachusetts-Lowell [N. Li and Wang, 2010] or for surveillance in a wide social area like metropolis like PRISMATICA [Lo et al., 2003]. The common problem of all the above systems is the sensor deployment and the system configuration for video communication. However, this is completely out of the scope of this research.

8.1.1 Transmission Constraints

The unwired node connection facility in WSNs comes with some typical problems for data transmission. Among them are line-of-sight obstruction, signal attenuation and interference, data security, and channel bandwidth or power constraint. In terms of efficient coding and transmission of the data one should find the trade-off between the bandwidth, the power consumption and the computational cost of the system. The large number of camera nodes and the big amount of data are always drawbacks for video surveillance systems. Thus, optimizing the configuration of the system could be itself a first solution to the power-efficiency problem. For instance, if the area which is surveyed is a small-scale environment then, a point-to-point communication of cameras is efficient for real-time observation. However, if it is a large public area, it will be necessary to adopt a communication system between the camera nodes.

8.1.2 Network Topologies

There are different topologies between the nodes (see Fig. 8.1). The star system proposed in PRISMATICA [Lo et al., 2003] is a system where each device deals with a relatively small area without necessarily being able to capture the global area of interest. If within the small local area there is an abnormal detection, the periphery node signals the central supervisor node. Another architecture which is more efficient with respect to the energy consumption is the SensEye [Kulkarni et al., 2005]. This system has a tree structure where the route node is the high resolution and computationally more expensive node which receives a signal by the leaf camera nodes when it is necessary. The leaf nodes are small low cost cameras which work for longer time comparing to the route. If the functionality and computational capability are equally distributed among the sensor nodes, a mesh network is more appropriate where the position of the target object is detected and the nearest camera is selected in order to track it [Kawamura et al., 2011]. The Hybrid-resolution smart cameras introduced in [Hengstler et al., 2006] provide some low energy cost. These kind of hybrid-resolution systems uses two cameras of different resolutions. A low resolution

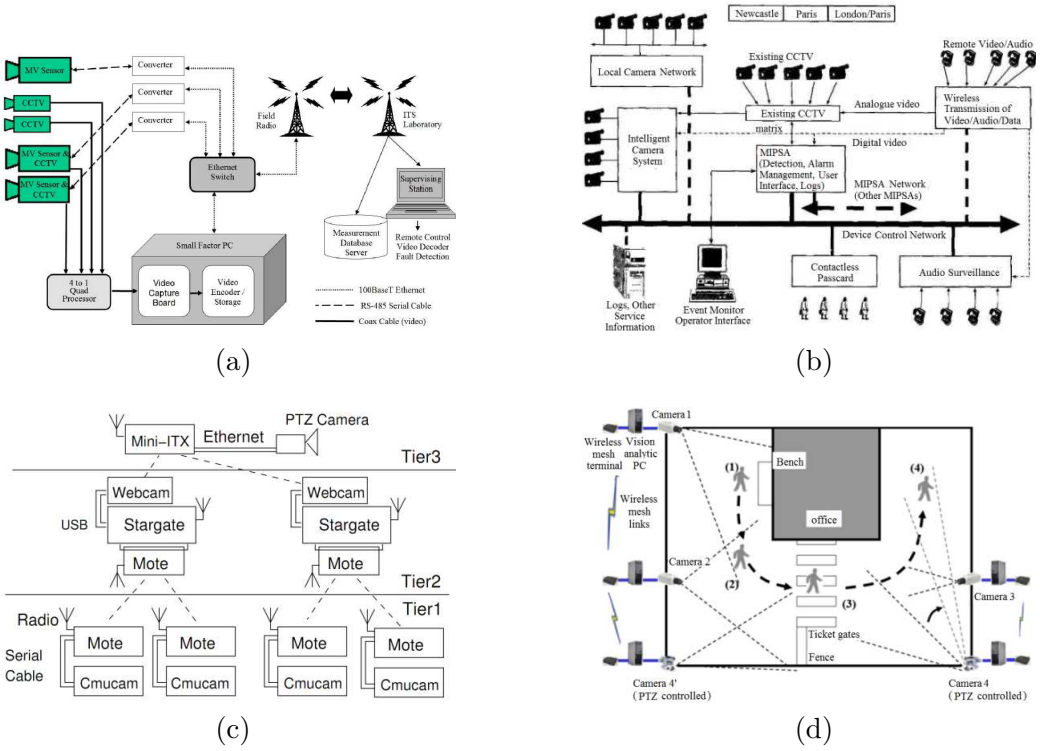


Figure 8.1: Network topology. (a) Point-to-point [Hourdakis et al., 2005]. (b) Star (PRISMATICA) [Lo et al., 2003]. (c) Tree (SensEye) [Kulkarni et al., 2005]. (d) Mesh [Kawamura et al., 2011]. (This Figure was published in [Ye et al., 2013]).

camera estimates the target object from the image data and the high resolution camera marks the position of the object and transmits only the video data inside the target region. Similar multiresolution strategy was also used in [Wang et al., 2009].

8.1.3 Pre-processing solutions

Although, the topology of WSN seems to be useful to reduce the size of surveillance videos which need to be stored and/or transmitted, there are several other much more effectual solutions. These solutions are related to the pre-processing of surveillance videos. First of all, we need to mention that the output of each camera node is encoded using video coding standards including JPEG, JPEG2000, Motion JPEG, MPEG, H.26x. The compression ratio of these standards is high but these algorithms have been built to compress videos in general and not especially surveillance videos. Thus, once the data are captured and before they are transmitted, one should take advantage of the special attribute of surveillance videos. These videos are often captured by stationary cameras that always stand towards the same scene for a long time. As a result, there is always a similar (or stationary) background information of the scene while the foreground changes. However, people are always interested in the foreground and moving target objects.

There is a great number of video coding and transmission techniques dedicated to the differentiation of the foreground and background, such as the Unequal Error Protection (UEP). The idea of UEP is to allocate more resources to the parts of the video which have a great impact on video quality, like the Regions of Interest (ROI) instead of the rest of the scene. Target or moving objects are able to be encoded more precisely than other less significant parts [Wang et al., 2005]. There have been proposed several background subtraction techniques reviewed in [Piccardi, 2004, McHugh et al., 2009] like Running Gaussian Average, Temporal Median Filter, Mixture of Gaussian, Kernel Density Estimation (KDE),

Sequential KD Approximation, Cooccurence of image variations and Eigen Background Technique. Some other energy saving power strategies include data filtering, buffering and adaptive message discarding [Feng et al., 2003]. Using the above techniques, it would be probably easier to build more autonomous systems, which are able to understand the events occurring in a scene because this is one of the biggest open issues in video surveillance system [Regazzoni et al., 2010].

8.2 4G-TECHNOLOGY

The 4G-TECHNOLOGY was founded in 2008 in order to provide hightech solutions related to interception, communication, analysis and processing, geolocalization and video surveillance problems. The company is active basically in France and it is manned by engineers, specialized technicians and dealers. The strongest activity of the company is related to surveillance. They have built two systems: the BSVi and the EViBOX/BVi which are widely provided in the department of Alpes-Maritimes.

8.2.1 BSVi Architecture

BSVi is a standalone video encoder which produces ciphered local recordings of special authorization saved on extractable disks. This special authorization offers high security to the recordings in case of theft. Summarizing the BSVi architecture, a HD camera records a digital video stream of 25 fps, which is encoded by H.264 standard format. The system is self-efficient with respect to the energy since it is compatible with different power technologies (solar power, batteries, fuel cells, etc.).

8.2.2 EViBOX/BVi Architecture

EViBOX is an Audio/Video recorder and encoder which adapts its performance according to the current or the future needs of a client. The goal of this machine is to use the minimum possible bandwidth for real-time broadcast over the available networks (4G, 3G, satellite, WiFi, ADSL, etc.). Figure 8.2 shows the complete architecture between the transmitter camera node and the receiver/client. A controlling PTZ camera, which is a camera that is capable of remote directional and zoom control, records the desired area producing data of H.264 standard format. These data are transmitted through an Ethernet channel to the EViBOX which is responsible on the one hand, to reconstruct the data using an H.264 decoder and then transcode them. The transcoding process inside the EViBOX uses a 4G-encoder which is similar to H.264 format but it is more efficiently tuned in order to fit the network bandwidth. Once the surveillance video has been transcoded it is sent through the VPN to the receiver. The receiver could reconstruct the encoded signal using the EViPack software which is provided by the 4G-TECHNOLOGY and then display the data on a machine (PC, laptop, tablet, smartphone, etc.). Another possible scenario is to provide the encoded data to a Video Management Systems (VMS) in order to be displayed. Each PTZ camera is connected to an EViPROXY device which stands as a virtual camera to the VMS. In case of large camera networks, the VMS reduces the workload using the EViPROXYs and retrieves only the data which are captured by the camera placed closer to the area of interest. The EViPROXY is a 4G-decoder which reconstructs the video stream before it is displayed.

8.2.3 Surveillance Scenarios

The benefits of the EViBOX architecture are related to the network bandwidth constraints. The bandwidth varies depending on the location and the characteristics of the area which is surveyed and it determines the trade-off between the allowed bitrate and the distortion.

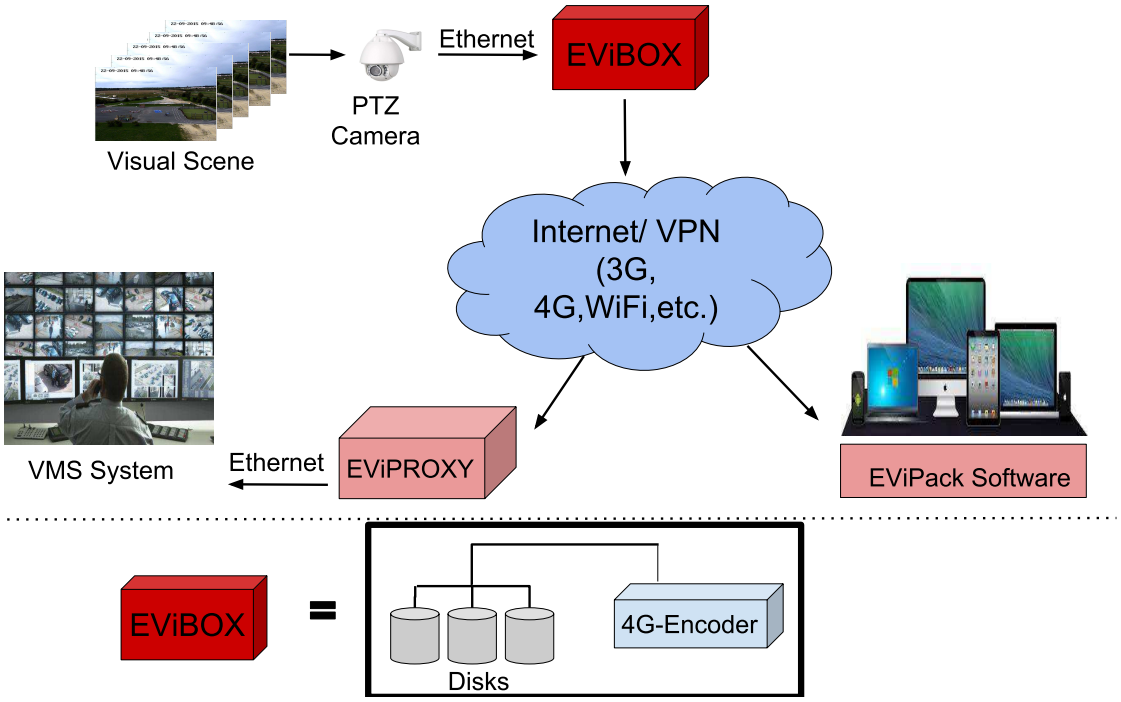


Figure 8.2: The architecture of the nomadic video surveillance system which uses the EViBOX/BVi machine/software.

There are many different scenarios which could be categorized in 3 general cases each one of which seems to demand different trade-off. We call these cases: working area, traffic and public area.

8.2.3.1 Working Area

A video stream which belongs to this case consists of a static background which is known to the receiver. The goal is to detect the motion of an external object/person in the foreground. Hence, the foreground "activates" the system when it changes. When there is no change into the scene the system is considered to be "disactivated". We have to mention here that there is a possibility for the background to be almost static. For instance, we aim to record an area inside the forest. This scene would never be absolutely static since there is always a motion of the leafs due to the wind or some weather phenomenon (rain, snow, etc.) which activates the system. As a result, there maybe some false alarms. To avoid these false alarms we may first of all define very well the background frame and then, we can also analyze the kind of motion. If the motion suddenly starts to happen in the whole frame this signal should not be recorded (i.e weather phenomena). In addition, if the motion is around very small radius comparing to the input one then it should also be neglected.

Given a static background we try to detect a motion of one of the included objects within a predefined area. In this case, we should also be aware of the meaningless motion which maybe exist according to the description of the previous case.

8.2.3.2 Traffic

Given a scene which is of a high motion (i.e. video surveillance in a highway) we aim to transmit a very low quality of the input signal while there are no abnormalities (i.e all the cars are moving to the same direction with a regular speed average constant). However, in case there is an "event" we need to increase the quality adding more high frequencies and precise in a better way the input signal. Using the term "event" we refer to:

1. the vigorous stop of the motion (i.e car accident)

2. motion of a different direction of the predefined one (i.e a person who tries to cross the road or a car which is moving in a opposite/not allowed direction in a high way.)

3. recognition of a prohibited size/shape of an object (i.e tracks moving in a part of the highway where it is forbidden).

8.2.3.3 Public Area

This is the most general case where the capture signal consists of a high and random motion. As a result, almost every frame should be encoded and transmitted to the receiver. In such a case, it is very demanding to define an "event" because the scene could be a walking area, the city center, etc. Some times, in these case, we combine the video with an audio signal to be able to precise the spatial origin of the sound. Hence, a high resolution signal is assigned to this spatial area while the resolution remains lower for the rest of the captured scene.

8.3 Our Contributions

In section 8.2, we introduced the EViBOX and the current system which is promoted by 4G-TECHNOLOGY. This system could be sufficiently improved using the retina-inspired codec and this section is dedicated to discuss the benefits of using our codec in a system like EViBOX. Before we propose some important scenarios which will benefit the performance of the EViBOX due to the retina-inspired codec, we need first to explain how this codec is applied to a video stream.

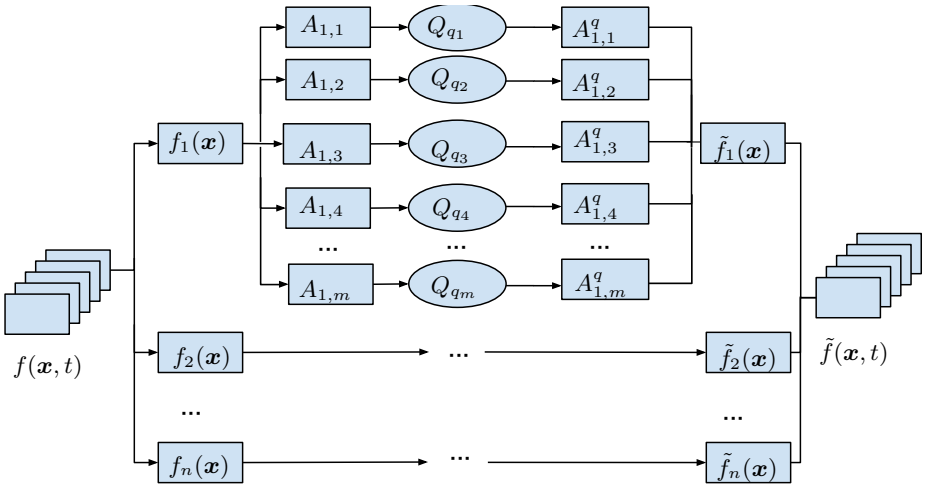


Figure 8.3: Video Codec Schema.

The retina-inspired codec is promoted as a video codec which is applied separately to each picture f_i of a video stream $f(\boldsymbol{x}, t)$ (Fig. 8.3). It is necessary to remind the reader that a video $f(\boldsymbol{x}, t)$ has been described as a group of N sequential images, each one of which appears for a given time T (see chapter 2). According to the vision of the first video codec designers, an algorithm which efficiently encodes a still-image could be used to encode the pictures of a video stream like in MJPEG and MJPEG2000 (see section 2.4.2). This compression will be very rough without taking under consideration the strong similarities between sequential pictures. Motion estimation algorithms could improve the total bitrate but this is out of the aspects of this work. Thus, without any motion estimation, our novel codec can not be compared to the current standards like the H.264/MPEG-4/AVC

(a) (b)

Figure 8.4: This figure shows some Regions of Interets (ROIs) in a picture of video streams captured from 4G-TECHNOLOGY. In picture (a) the receiver is interested in the car number plates while in (b) to the material/equipment in a working area. The retina-inspired codec is able to provide to the receiver some copies of these ROIs with a lower distortion than the parts of the scene which are out of the ROIs.

or the H.265/HEVC. A fair comparison of retina-inspired codec would be with MJPEG, MJPEG2000 and maybe also MPEG-2, since the motion estimation of this standard is very naive.

Concerning the EViBOX, the retina-inspired codec will not replace the encoding system of camera but the transcoding system of EViBOX (see Fig. 8.2). We represent how the EViBOX is going to adopt our codec and how its special features and its strong dependence on time will benefit the system. The first possible scenario is to take the advantage of the dynamic behavior of the retina-inspired codec in a surveillance system by tuning the visual quality of the reconstructed signal. According to the progressive reconstruction which has been introduced in chapter 7, the quality of the reconstruction can be dramatically improved while the number of the retina-inspired decomposition layer increases with respect to time. Since most of the surveillance systems are linked to some detection algorithms, one could take the advantage of these algorithms to tune the quality of the reconstructed video stream. Consequently, if nothing interesting is detected into the scene, it is unnecessary for the transmitter to propagate very high quality of the input signal and "wasting" energy and bandwidth. In such a case, transmitting a low or medium reconstruction quality of the captured signal using a few first layers of the retina-inspired frame will be enough for the receiver. However, if an event which is detected needs to be further analyzed, the transmitter could increase the quality of the reconstruction providing the full set of the retina-inspired decomposition layers which are finer quantized.

Another case which is more challenging is to separate the visual scene into Regions Of Interests (ROIs). The ROIs are a reference to some areas of the visual scene for which the surveillance system has been placed. For instance, some interesting examples are given in Fig. 8.4. Fig. 8.4 (b) shows a working area where the employer needs to secure the equipment 24h/day (tracks, machines, construction materials, etc.). As a result, it would be wise to provide higher quality signal to the areas where the ROIs are and lower quality to the rest of the scene. In Fig. 8.4 (a) we represent another example which is related to the road survey. In this case, a region of interest could be selected part of the input scene which allows for example to detect the car number plates. As a result, this region is transmitted in the sharpest possible way comparing to the rest of the visual scene. In such a case, everything which is included into the ROIs are going to be transmitted in a sharp

and high quality way whilst the rest of the visual scene will be propagated with a lower quality.

The next section represents some numerical results of the retina-inspired codec applied to some well-known video streams (i.e. foreman and bus). We aim to illustrate the visual quality for different bitrates and distortions and compare the retina-inspired codec with the state-of-the-art which in our case is MJPEG, MPEG-2 and MJPEG2000.

8.4 Video Numerical Results

This section is dedicated to illustrate some numerical results concerning the retina-inspired codec when it is applied to video streams. As explained before, we have applied the coder to each picture of the video stream separately. Our goal is to show that the retina-inspired codec is more efficient comparing to MJPEG which is the standard that is also applied to video streams in the same way without motion estimation. The results we obtained were encouraging enough to continue the comparison using MPEG-2 which is the first standard that uses a very naive motion estimation method. What we have noticed is that although the PSNR value of MPEG-2 was higher than the retina-inspired codec for very lower bitrates, the visual results are much better using our coding system. We are going to provide results only related to the uniform-LIFQ since the impact of the adaptive-LIFQ and the optimized-LIFQ has been extensively detailed in chapter 7. We have also tested other video streams such as the bowing and the bus videos obtaining similar results. At this point we need to highlight that the video streams $f(\boldsymbol{x}, t)$ which are used in our experimental results have been pre-processed using the free software "Total Video Audio Converter 4". The software receives as an input a video streams of CIF format (resolution of each picture 352×288 pixels) of YCbCr (see section 2.4.1). The output video converts each color picture $f_i(\boldsymbol{x})$ into a grayscale picture of the size 256×256 pixels. At the same time, we set the total number of the pictures $N = 100$, keeping only the first 100 pictures of the original video stream.

Figures 8.5 and 8.6 show the performance of the retina-inspired codec with a uniform-LIFQ, when it is applied to the very well-known foreman video, using PSNR and SSIM quality metrics respectively. Figure 8.5 shows the PSNR_v value (see eq. (8.1)) vs the entropy value H_v of the whole video stream (see eq. (8.3)) measured for the retina-inspired codec and the MJPEG and MPEG-2 standards. Concerning the standards, we obtained four different qualities "low", "medium", "good" and "high" each one of which corresponds to a certain H_v bitrate. The retina-inspired codec performs better comparing to MJPEG concerning the medium, good and high qualities. In addition, it is comparable to MPEG-2 concerning the high quality. Figure 8.7 compares the visual quality of the 60^{th} picture of the foreman video stream encoded by MJPEG and the retina-inspired codec.

Figures 8.8 and 8.9 show that the retina-inspired codec performs in the same way as MJPEG for "medium", "good" and "high" qualities. Figures 8.10 and 8.11 illustrate that our codec outperforms MJPEG for "medium", "good" and "high" qualities and it is also better than MPEG-2 for "good" and "high" qualities. Figures 8.12, 8.13 and 8.14 approve the above results depicting the visual quality of some extracted pictures of the above video streams which are encoded by the retina-inspired codec and the two standards MJPEG and MPEG-2.

$$\text{PSNR}_v = 10 \log_{10} \frac{(2^n - 1)^2}{\text{MSE}_{\text{total}}}, \tag{8.1}$$

where

$$\text{MSE}_{\text{total}} = \frac{1}{N} \sum_{i=1}^{N} \text{MSE}^i, \tag{8.2}$$

where MSE^i is given by eq. (2.1).

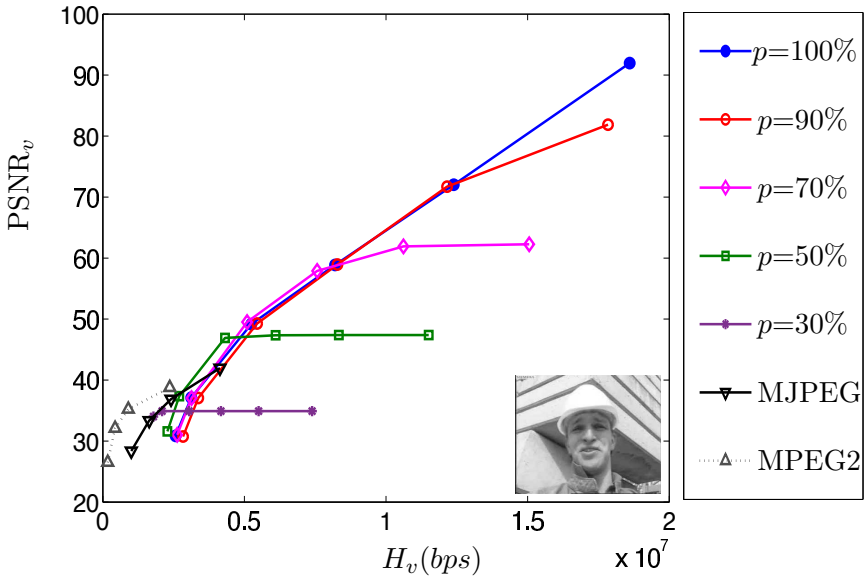


Figure 8.5: This figure compares using PSNR metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for foreman video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec outperforms MJPEG for the "medium", "good" and "high" qualities while it is also comparable to MPEG-2 for its "high" quality.

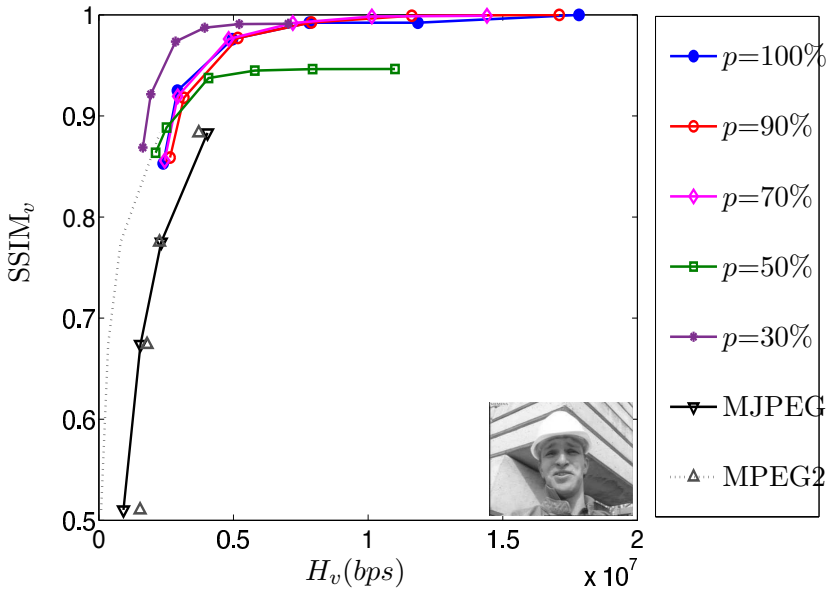


Figure 8.6: This figure compares using SSIM metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for foreman video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec outperforms MJPEG for the "medium", "good" and "high" qualities and MPEG-2 for its "high" quality.

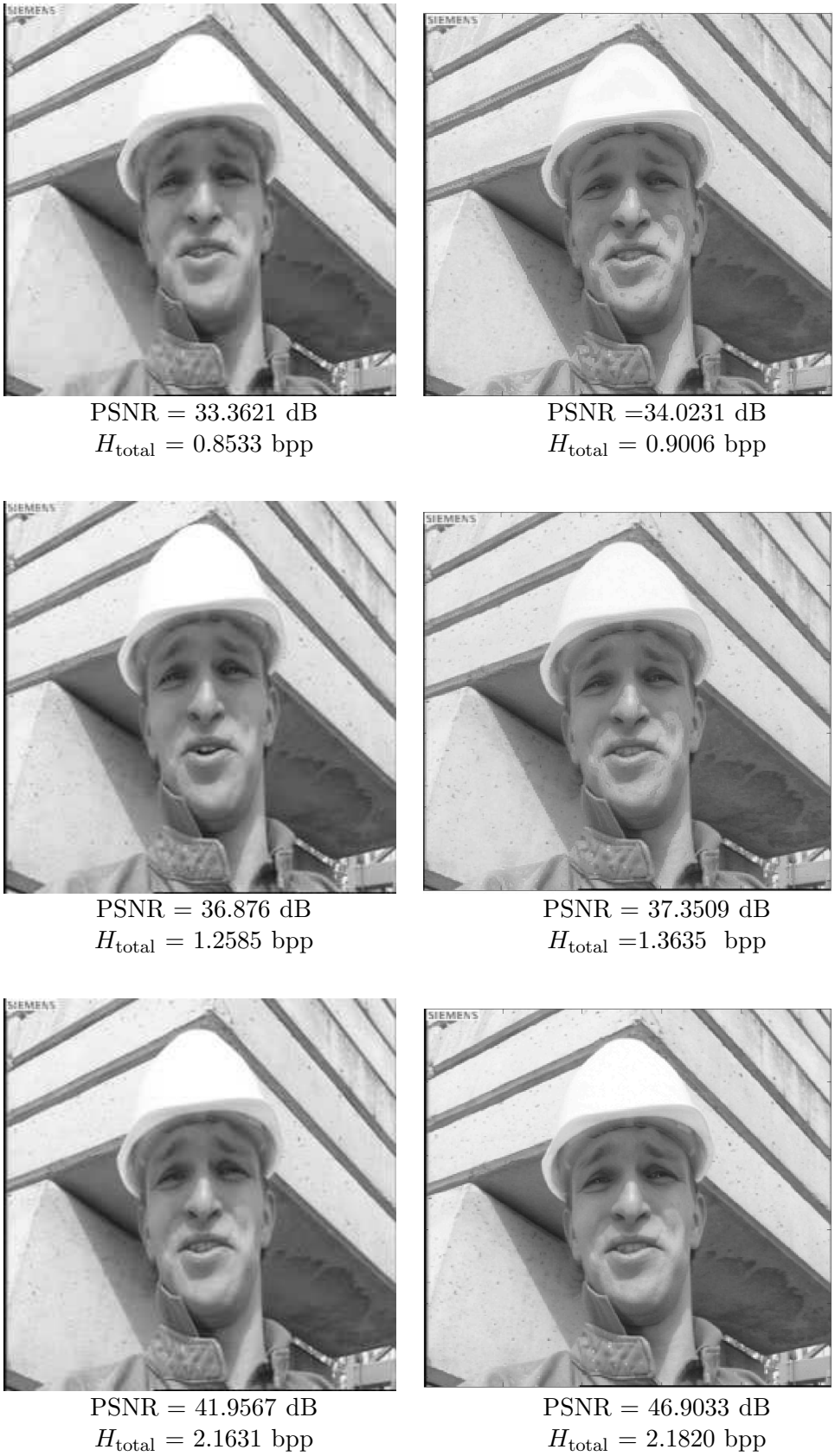


Figure 8.7: This figure compares the visual quality of the 60^{th} picture of the foreman video stream encoded by MJPEG (left column) and the retina-inspired codec (right column).

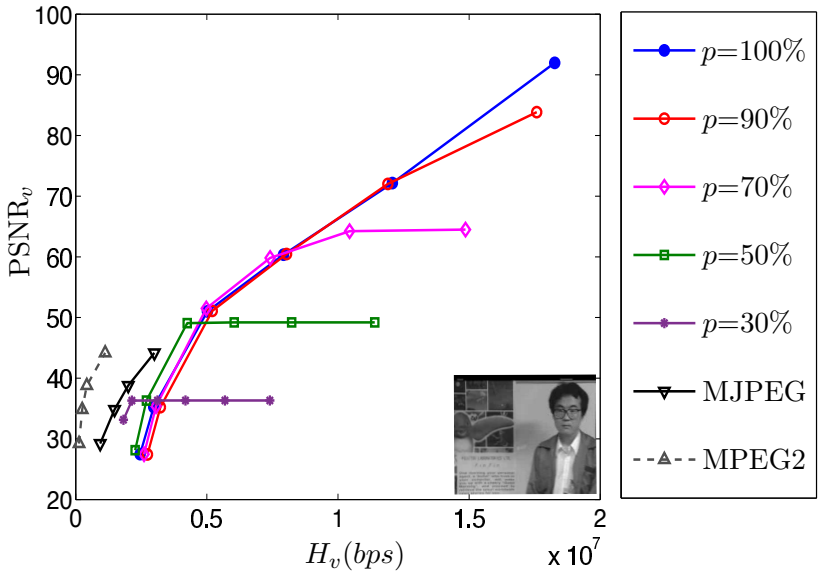


Figure 8.8: This figure compares using PSNR metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for bowing video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec is very close to MJPEG for the "medium", "good" and "high" qualities.

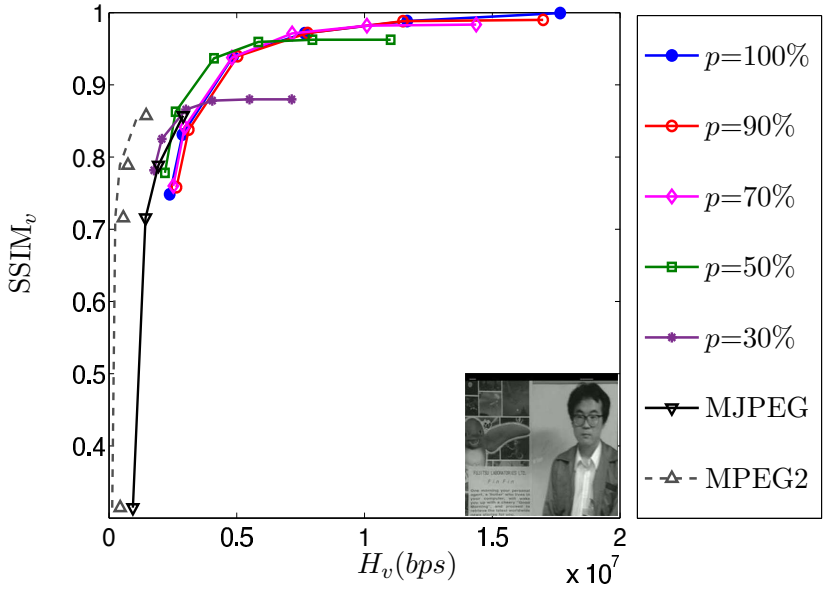


Figure 8.9: This figure compares using SSIM metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for foreman video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec is very close to MJPEG for the "medium", "good" and "high" qualities.

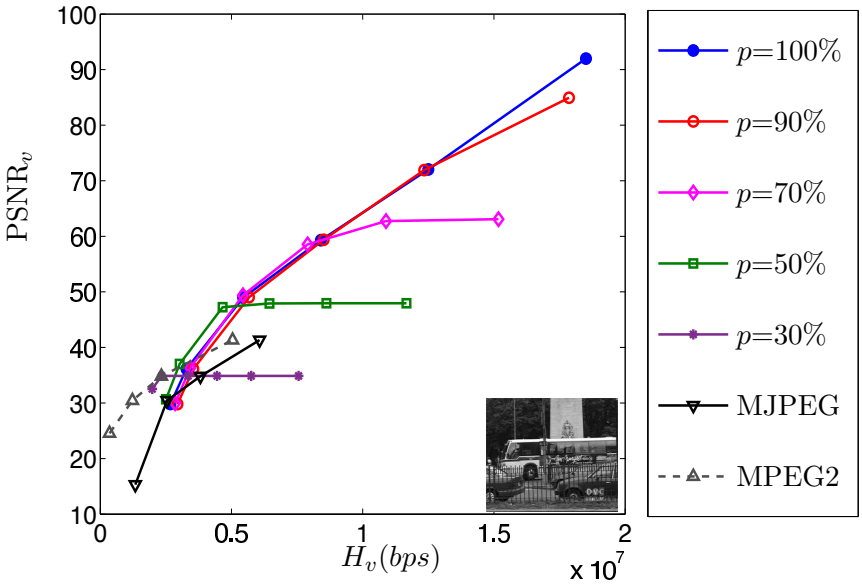


Figure 8.10: This figure compares using PSNR metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for bus video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec outperforms MJPEG for the "medium", "good" and "high" qualities while it is also better than MPEG-2 for its "good" and "high" qualities.

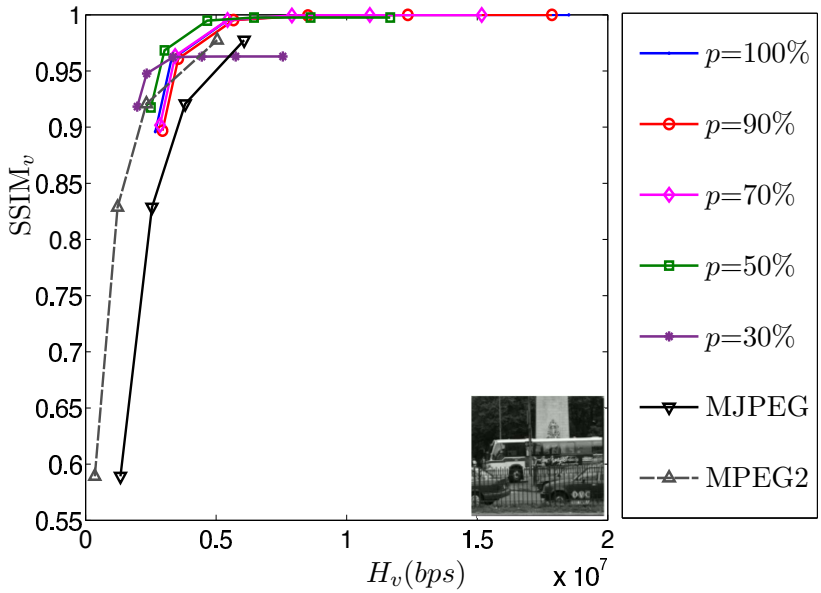
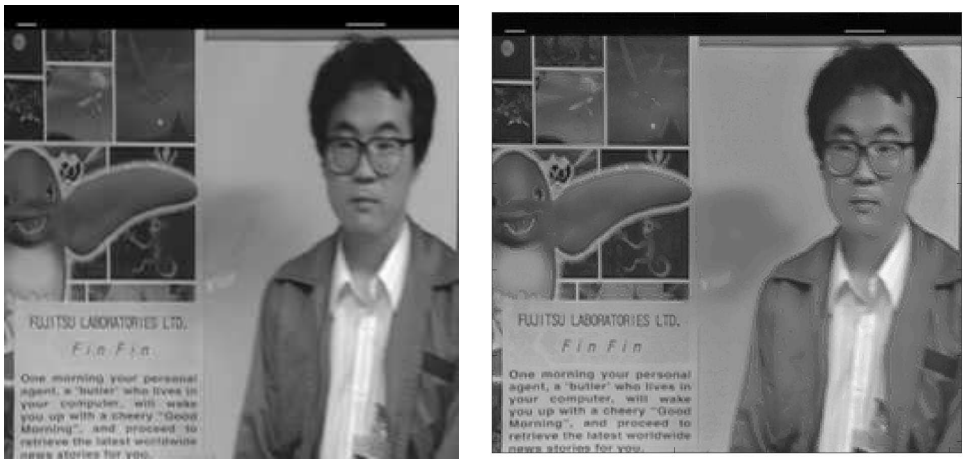


Figure 8.11: This figure compares using SSIM metric, the performance of MJPEG and MPEG-2 standards to the retina-inspired codec with uniform-LIFQ for foreman video. We tested the uniform-LIFQ for different λ values tuned according to $p \in \{100\%, 90\%, 70\%, 50\%\}$ and quantization steps $q \in \{1400, 800, 200, 50, 10, 1\}$. The two standards have been tested for 4 different qualities: "low", "medium", "good" and "high". The retina-inspired codec outperforms MJPEG for the "medium", "good" and "high" qualities while it is also better than MPEG-2 for its "good" and "high" qualities.

PSNR = 34.8238 dB
H_{total} = 0.7520 bpp

PSNR = 33.14 dB
H_{total} = 0.9258 bpp

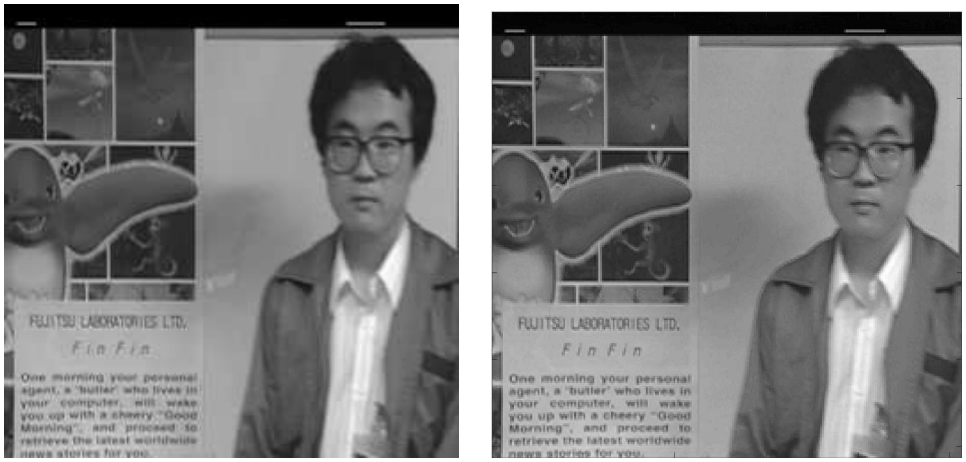
medium



PSNR = 38.7909 dB
H_{total} = 1.0181 bpp

PSNR = 36.31 dB
H_{total} = 1.0893 bpp

good



PSNR = 44.1689 dB
H_{total} = 1.5259 bpp

PSNR = 45.07 dB
H_{total} = 1.42 bpp

high

Quality
MJPEG

Figure 8.12: This figure compares the visual quality of the 60^{th} picture of the bowing video stream encoded by MJPEG (left column)and the retina-inspired codec (right column).

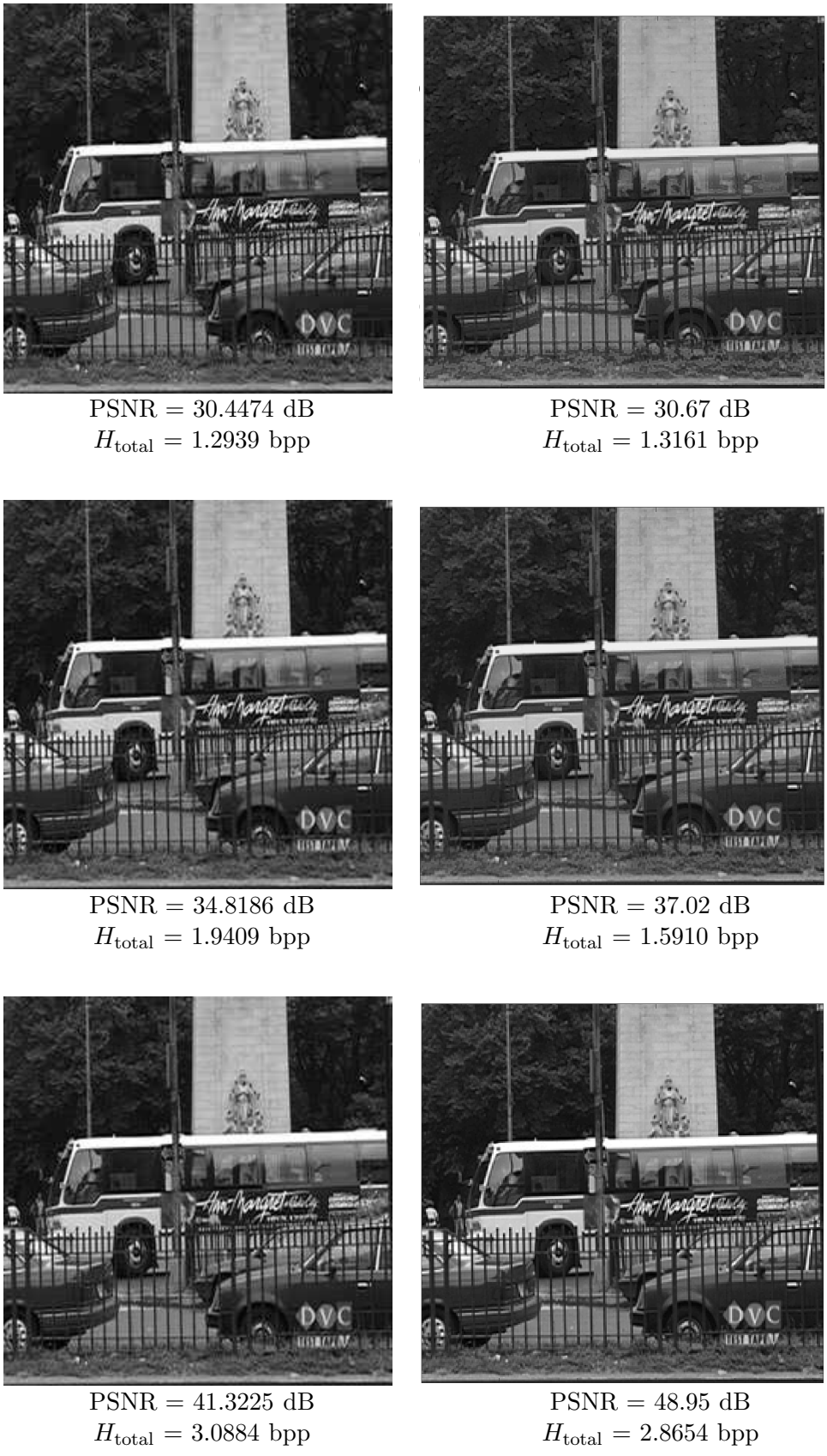


Figure 8.13: This figure compares the visual quality of the 60^{th} picture of the bus video stream encoded by MJPEG (left column)and the retina-inspired codec (right column).

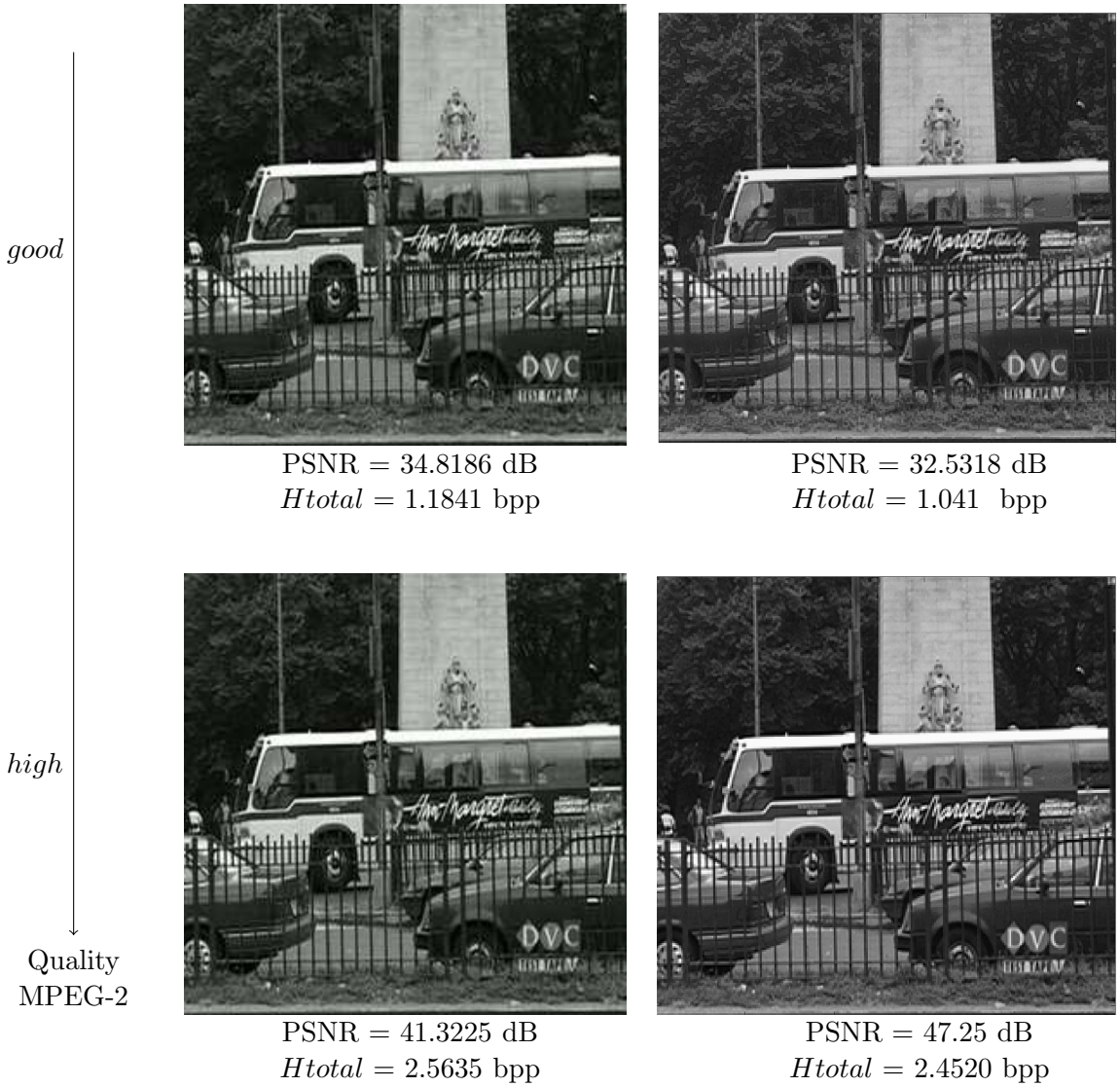


Figure 8.14: This figure compares the visual quality of the 60^{th} picture of the bus video stream encoded by MPEG-2 (left column) and the retina-inspired codec (right column).

$$H_v = \frac{1}{N}\sum_{i=1}^{N} H_{total}^i, \tag{8.3}$$

where H_{total}^i is given by eq. (7.11).

$$\mathrm{SSIM}_v = \frac{1}{N}\sum_{i=1}^{N} \mathrm{SSIM}^i, \tag{8.4}$$

where SSIM^i is given by eq. (2.3).

8.5 Conclusion

This chapter was a brief introduction to video surveillance systems. We presented several different models concerning the architecture of these systems, their constraints and needs. We also introduced our industrial partner, the 4G-TECHNOLOGY, which is a company

that provides systems and services for surveillance. We discuss that EViBOX is the main product of 4G-TECHNOLOGY which could be benefited by our retina-inspired video codec. Last but not least, we provide numerical results of the retina-inspired codec applied to well-known video streams and a video stream captured by our partner. We compare our results to MJPEG and MPEG-2 illustrating some graphs but also some pictures in order to be easier to compare the quality of the reconstruction. Our method outperforms MJPEG in PSNR and SSIM for given bitrates and it seems to be promising comparing to MPEG-2 although our method does not consider yet any motion estimation.

Chapter 9

General Conclusion

This thesis has dealt with a novel video coding architecture which is inspired by the retina. The basic characteristic of the retina is its capacity to dynamically process and encode the visual stimulus. We prove in this thesis that by deriving models based on equations which approximate the retina, we are able to build a novel and very efficient coding system. Here, we proposed a coding algorithm which is called retina-inspired video codec. This codec consists of two basic processing steps which are also the main contributions of this thesis: the retina-inspired filtering and the LIF quantizer which is a dynamic quantizer.

9.1 Contributions

9.1.1 Retina-inspired Filtering

The retina-inspired filter is a groundbreaking analysis of neuroscience for image processing. We propose a detailed background of neuromathematical models which are all based on DoG functions. The properties of these functions are very well known in image processing since they have been widely used. Under the strong assumption that the input signal is constant in time, we propose the retina-inspired filter. This filter is a group of time-varying WDoGs. In literature, there is no any other similar work which studies WDoG filters, so this is an important contribution especially due to the efficiency of these filters. We propose a spatial and frequential analysis of a general WDoG function. The retina-inspired filter enables the extraction of different kinds of data while time increases. We have also proven that the filter is a lowpass filter generating low frequency copies of the input signal and it turns with time into a bandpass filter enabling to extract high frequencies. The retina-inspired filter is a great improvement of other bio-inspired filters which are simpler and not as accurate as neuroscientific approximations of the retina. Hence, this is certainly of a great interest for the image processing field. We have also mathematically proven that the retina-inspired filter is a frame according to the frame theory. That means that the retina-inspired transform is invertible. We also illustrate some numerical results which guarantee that the reconstruction using all the retina-inspired frame coefficients is perfect in absence of any perturbations. Last but not least, we introduced some AWGN to the retina-inspired decomposition in order to be more realistic and to show the impact of noise on the reconstruction quality. One observes that the redundancy of the retina-inspired decomposition is high enough to reduce the impact of the noise and provide high reconstruction quality results. In fact, we show that although the presence of noise influences the quality of the decomposition layers, the quality of the reconstructed signal, which is measured by PSNR or SSIM, remains high.

9.1.2 LIF Quantizer

The retina-inspired quantizer which is also termed LIF Quantizer (LIFQ) is a model motivated to perform according to LIF spike generator. First of all, we provide a comparison of neuromathematical models which approximate the generation of the neural code. Based on this background we explain that the LIF model seems to be very efficient to describe the emission of spikes. This decision was further enhanced under the initial assumption that the input signal is constant in time and under some time constraints which accompany the real-time encoding and decoding of a video stream. The LIF model assumes that the time of the first spike arrival is the only necessary information which needs to be propagated to the brain. We figured out important similarities between the behavior of the LIF and a uniform dead-zone scalar quantizer which is a very well known and widely used model in compression algorithms. Thus, we proposed the novel LIFQ which is close to the conventional quantization but it enables to take under consideration the limited time we are allowed to encode and decode the input signal according to the performance of LIF. The LIFQ was a necessary processing step in order to reduce the number of bits which is required to store the real values of the time delay. We studied different cases concerning the quantization step. The simplest model was the perfect-LIFQ where the reduction of the spatiotemporal redundancy was achieved only by the dead-zone; the values out of the deadzone are not quantized. The uniform-LIFQ introduces a uniform quantization step for all the retina-inspired decomposition layers. The adaptive-LIFQ in which the quantization step changes with respect to the range of the decomposition layers and the method of the optimized-LIFQ where the quantization step is computed according to the optimization Rate-Distortion (RD) theory. Last but not least, we show that the optimized-LIFQ outperforms the rest LIFQ models.

9.1.3 Retina-inspired image and video codec

The retina-inspired filter and the LIFQ build the retina-inspired codec. This codec has been applied to still-images which are flashed for a given time. Comparing the reconstruction results of the retina-inspired codec to JPEG and JPEG2000 we conclude that our coding system performs beyond the standard in terms of compression for bitrates higher than 1 bpp while the visual quality of the reconstruction is better for lower bitrates. We also compare the retina-inspired codec to other bio-inspired coding systems concerning their dynamic behavior. The progressive reconstruction using the retina-inspired codec enables better description of the scene even for very low bitrates. Last but not least, we applied the retina-inspired codec to each picture of a video stream. In such a way we were also allowed to compare our results to MJPEG. Our coding system outperforms MJPEG since we provide higher reconstruction quality evaluated by PSNR and SSIM metrics. These results were encouraging enough to compare our model with MPEG-2 standard which uses motion estimation and it is expected to perform much better than the retina-inspired codec. However, our codec is comparable to MPEG-2 for high quality video streams while for lower qualities it seems that the retina-inspired codec outlines better the content of the input scene.

9.2 Perspectives

The retina-inspired codec is a novel and very promising coding algorithm. Of course such a codec opens a lot of perspectives which are interested to be studied. In this section, we introduce some extensions of the proposed models related to neuroscientific inspirations, information technology and video surveillance applications

- Neuroscience:

- In this thesis, we have released a novel WDoG filter which is inspired by the behavior of the OPL retina cells. This filter was derived under the assumption that the input signal is constant in time. However, this assumption limits the dynamic behavior of the model. It would be very interesting to study how the retina-inspired filter could be applied to an input signal which evolves along time. The initial assumption we made in this thesis that the input signal is constant with respect to time serves mainly educational reasons since it allows us to study the behavior of the retina-inspired filter. However, the OPL retina layer processes the visual stimulus on the fly and this is why we need to overcome the initial assumption. Of course, we should always keep in mind that this retina-inspired transform aims to be used for compression and we should always guarantee that it is invertible.
- Another important issue concerning the progress of the current retina-inspired codec is to improve the quantization process. First of all, in this thesis we approximated the LIF model with a uniform dead-zone scalar quantizer. However, it would be also interesting to implement and study the LIF model in terms of coding. We should also generate the code of spikes and use this code which is more realistic in order to reconstruct. Another aspect would related to the dynamic behavior of the LIF model. Not only the OPL transform but also the spike generation is a dynamic process. Thus, we should also extend the LIFQ and apply it to an input signal which evolves in time.
- The LIF model was proposed under some strong assumptions which are linked to the connectivity between the neurons. Another interesting perspective is to use other kind of models which concern the interconnection and the feedback which is sent between the neurons.

- Information Theory:
  - The WDoG is a novel decomposition which has been used only in our retina-inspired coding system. It would be interesting if this filter could be adopted by the already existing coding systems in order to be compared to the conventional multilayer transforms like DCT or DWT.
  - The WDoG filter could be also tested in different approaches like the image analysis including edge detection, object tracking, medical image analysis or High-Dynamic-Range Imaging (HDRI). It is expected that the dynamic behavior of the filter and the origins of the model will perform good in terms of perception.
  - In this thesis, we have proposed a LIFQ model which approximates the spike generation neural process. It would be challenging to use this novel quantizer into standards and replace the current and conventional quantization methods. Then, it would be possible to conclude whether the LIFQ improves or not the compression performance.
  - The proposed codec is the first approximation of a compression system which is based on neuromathematical models. Thus, there are several limitations which could be improved. First of all, the way we apply the codec to a video stream is very rough like MJPEG and MJPEG2000. According to this way, we are not able to compute any motion estimation. It would be interesting to adopt some motion compensation models in order to be comparable to the latest standards.

- Video Surveillance:
  - The proposed codec was implemented in MATLAB for experimental simplicity. It would be necessary to use a low-level programming language to execute this code in a more efficient way.

- The retina-inspired codec was proposed through the prism of video surveillance systems. This codec could be adopted by a surveillance system in a pre- or post-processing tool which allows adaptive visual quality with respect to some Regions Of Interests (ROIs).
- Taking the advantage of the dynamic behavior of the retina-inspired codec in a surveillance system, one could tune the visual quality of the reconstructed signal. Since most of the surveillance systems are linked to some detection algorithms, one could take advantage of these algorithms and tune the quality of the reconstructed video stream.

Part V

APPENDIX

Appendix A

Proof of Lemma 1

This appendix proves Lemma 1. It is shown that both $J_c(t)$ and $J_s(t)$ can be defined in a closed form as polynomial functions which are attenuated by exponential ones. These functions will be essential to calculate $R_c(t)$ and $R_s(t)$. The calculation of $J_c(t)$ and $J_s(t)$ are based on the following lemma whose proof is straightforward.

Lemma 2. Let ω a real value, $t \geq 0$ and n a positive integer. Using an integration by parts, we obtain the following equality:

$$\int_0^t u^n \exp(-\omega u)\, du \quad = \quad P_n(t) \exp(-\omega t) + c,$$

where

$$P_n(t) = \sum_{k=0}^{n} -\frac{n!}{(n-k)!\,\omega^{k+1}} t^{n-k}$$

is a polynomial function in t of order n whose coefficients depend on ω and c is a constant value.

A.1 Closed-form of $J_c(t)$

Assume that $0 \leq t \leq T$. It follows that:

$$J_c(t) \quad = \quad \int_{u=0}^{t} W(u)du,$$

which yields

$$J_c(t) \quad = \quad \int_{u=0}^{t} E_{\tau_G,n} \overset{t}{*} (\delta_0 - w_C E_{\tau_C})(u)du$$

$$= \quad \int_{u=0}^{t} E_{\tau_G,n}(u)du - w_C \int_{u=0}^{t} E_{\tau_G,n} \overset{t}{*} E_{\tau_C}(u)du.$$

The definition of the gamma and exponential filters yields:

$$J_c(t) \quad = \quad \frac{1}{\tau_G^{n+1}} \int_{u=0}^{t} u^n \exp(-au)\, du$$

$$- \quad \frac{w_C}{\tau_G^{n+1}\tau_C} \left(\int_{u=0}^{t} \exp\left(\frac{-u}{\tau_C}\right) \int_{v=0}^{u} v^n \exp(-bv)\, dvdu \right).$$

where $a = \frac{1}{\tau_G}$, $b = \frac{\tau_C - \tau_G}{\tau_G \tau_C}$. Using Lemma 2, we get

$$\begin{aligned} J_c(t) &= \sum_{k=0}^{n} -\frac{(n!)}{(n-k)!a^{k+1}\tau_G^{n+1}} t^{n-k} \exp(-at) \\ &- \sum_{k=0}^{n}\sum_{l=0}^{m} \frac{n! w_c}{(m-l)! b^{k+1} a^{l+1} \tau_G^{n+1} \tau_C} t^{m-l} \exp(-at) \\ &+ \frac{n! w_c}{b^{n+1}\tau_G^{n+1}} \exp\left(\frac{-t}{\tau_C}\right) \\ &+ \sum_{k=0}^{n} \frac{n! w_c}{b^{k+1} a^{m+1} \tau_G^{n+1} \tau_C} - \frac{n! w_c}{b^{n+1}\tau_G^{n+1}} + n! \end{aligned}$$

where $m = n - k$. Finally,

$$J_c(t) = P_n(t) \exp\left(\frac{-t}{\tau_G}\right) + \alpha_c \exp\left(\frac{-t}{\tau_C}\right) + \gamma_c$$

where $P_n(t)$ is a polynomial function in t of order n and α_c and γ_c are two reals.

A.2 Closed-form of $J_s(t)$

The method used to calculate $J_c(t)$ can be applied to $J_s(t)$. This leads to:

$$\begin{aligned} J_s(t) &= \int_{u=0}^{t} (W \overset{t}{*} E_{\tau_S})(u) du \\ &= \int_{u=0}^{t} E_{\tau_G,n} * (\delta_0 - w_C E_{\tau_C}) * E_{\tau_S}(u) du \\ &= \int_{u=0}^{t} E_{\tau_G,n} * E_{\tau_S}(u) du \\ &\quad - w_c \int_{u=0}^{t} E_{\tau_G,n} * E_{\tau_C} * E_{\tau_S}(u) du. \end{aligned}$$

Using Lemma 2, we get

$$\begin{aligned} J_s(t) &= \frac{1}{\tau_G^{n+1}\tau_S} \left(\sum_{k=0}^{n}\sum_{l=0}^{m} \frac{n! t^{m-l} \exp(-at)}{g^{k+1} a^{l+1} (m-l)!} \right. \\ &- \left. \sum_{k=0}^{n} \frac{n!}{g^{k+1}a^{m+1}} + \frac{n!\tau_S}{g^{n+1}} \left(1 - \exp\left(\frac{-t}{\tau_S}\right)\right) \right) \\ &- \frac{w_c}{\tau_G^{n+1}\tau_C\tau_S} \left(\sum_{k=0}^{n}\sum_{l=0}^{m}\sum_{r=0}^{p} -\frac{n! t^{p-r} \exp(-at)}{(p-r)! b^{k+1} g^{l+1} a^{r+1}} \right. \\ &+ \left. \sum_{k=0}^{n}\sum_{l=0}^{m} \frac{n!}{b^{k+1} g^{l+1} a^{p+1}} \right. \\ &- \left. \sum_{k=0}^{n} \frac{n!\tau_S}{b^{k+1} g^{m+1}} \left(1 - \exp\left(\frac{-t}{\tau_S}\right)\right) \right) \end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{n!\tau_S}{b^{n+1}\phi}\left(1-\exp\left(\frac{-t}{\tau_S}\right)\right) \\
&- \quad \frac{n!\tau_C}{b^{n+1}\phi}\left(1-\exp\left(\frac{-t}{\tau_C}\right)\right)\Bigg),
\end{aligned}
$$

with the variables

$$g=\frac{\tau_S-\tau_G}{\tau_G\tau_S},\ \phi=\frac{\tau_S-\tau_C}{\tau_C\tau_S},$$

$p=m-l$ and $m=n-k$. It follows that:

$$
\begin{aligned}
J_s(t) \quad = \quad & Q_n(t)\exp\left(\frac{-t}{\tau_G}\right)+\alpha_s\exp\left(\frac{-t}{\tau_S}\right) \\
& +\beta_s\exp\left(\frac{-t}{\tau_C}\right)+\gamma_s
\end{aligned}
$$

where $Q_n(t)$ is a polynomial function in t of order n and α_s, β_s and γ_s are some reals. This ends the proof.

Appendix B

List of Symbols

$f(\boldsymbol{x})$	Input image
$\tilde{f}(\boldsymbol{x})$	Reconstructed image
$V(\boldsymbol{x}, t)$	Video stream
N	Number of frames
T	Time an image/picture if flashed
p_i	Probability of a symbol i
H_j	Shannon Entropy of the j subband
H_{total}	Total Shannon Entropy
J	Rate-Distortion cost function
D	Total Distortion
R	Total Rate
$R_{\max}$	Maximum Rate
μ	Lagrange multiplier
v	The input of a quantizer
q	Quantization step
$\text{sgn}(v)$	Sign of an input v
$Q_q(v)$	Quantizer
λ	Dead-zone threshold
$Q_q^\lambda(v)$	Dead-zone quantizer
$K(\boldsymbol{x}, t)$	Spatiotemporal kernel
$A(\boldsymbol{x}, t)$	Activation degree
G_{σ_c}	Center Gaussian in space
$\hat{G}_{\sigma_c}$	Center Gaussian in frequency
G_{σ_s}	Surround Gaussian in space
$\hat{G}_{\sigma_s}$	Surround Gaussian in frequency
σ_c	Center standard deviation
σ_c	Surround standard deviation
$DoG(\boldsymbol{x})$	Spatial DoG
$DoG^k(\boldsymbol{x})$	Spatial DoG pyramid
$DoG(\boldsymbol{x}, t)$	Spatiotemporal Difference of Gaussian
$W(t)$	Difference of Exponentials
$E_{\tau,n}$	Gamma temporal filter
$\phi(\boldsymbol{x}, t)$	Retina-inspired filter in space
$\hat{\phi}(\boldsymbol{x}, t)$	Retina-inspired filter in frequency
$R_c(t)$	Center temporal filter
$R_s(t)$	Surround temporal filter
$P_n(t)$	Polynomial function in t of order n
r	Radial coordinate
ω	Angular frequency
ξ	Ordinary frequency

α	Lower bound of the frame
β	Upper bound of the frame
$C(\boldsymbol{x}, t)$	Center spatiotemporal filter
$S(\boldsymbol{x}, t)$	Surround spatiotemporal filter
Φ^{-1}	Inverse of the matrix Φ
Φ^{T}	Transpose of the matrix Φ
$\eta(\boldsymbol{x})$	Additive White Gaussian Noise
$r(I)$	Firing rate
r_m	Mean firing rate
d_{ref}	Refractory period
C	Capacitor
$V(t)$	Voltage of the resistor
V_r	Reset voltage
R	Resistor
$I(t)$	Input current
θ	Threshold
t^f	Firing time
$d(v)$,	Delay of spike arrival for an intensity v
$d_{\max}$	Maximum reconstruction delay
t_{obs}	Observation window
N_s	Number of spikes

Appendix C

List of Abbreviations

A/D	Analog to Digital
ADSL	Asymmetric Digital Subscriber Lines
AVC	Advanced Video Coding
AWGN	Additive White Gaussian Noise
BP	Bandpass
bpp	Bits per pixel
CABAC	Content-Adaptive Binary Arithmetic Coding
CB	Coding Block
CCITT	International Telegraph and Telephone Consultative Committee
CCTV	Closed-Circuit TeleVision
CIF	Common Intermediate Format
CNS	Central Nervous System
CPU	Central Processing Unit
CTU	Coding Tree Unit
CTB	Coding Tree Block
CU	Coding Unit
DA	Display Adaptation
DCT	Discrete Cosine Transform
DST	Discrete Sine Transform
DVD	Digital Video/Versatile Discs
DWT	Discrete Wavelet Transform
DoE	Difference of Exponential
DoG	Difference of Gaussian
EBCOT	Embedded Block Coding with Optimal Truncation
EOB	End Of suBbands
EOI	End Of Image
FWHM	Full Width Half Maximum
GL	Ganglionic Layer
GoP	Group of Pictures
HD	High Definition
HDTV	High Definition TeleVision
HEVC	High Efficiency Video Coding
HFC	Hubric Fiber Cable Network
HVS	Human Visual System
IF	Integrate and Fire
IPL	Inner Plexiform Layer
ISO	International Organization for Standardization
ITU	International Telecommunication Union
JPD	Joint Probability Distribution
JPEG	Joint Photographic Expert Group
JVT	Joint Video Team
KDE	Kernel Density Estimation

L1, L2	Lowpass
LB	Lowpass/Bandpass
LCD	Liquid Crystal Display
LIF	Leaky Integrate and Fire
LIFQ	Leaky Integrate and Fire Quantizer
LGN	Lateral Geniculate Nucleus
LSB	Least Significant Bit
LUT	Look-Up-Table
MJPEG	Motion Joint Photographic Expert Group
MPEG	Motion Picture Experts Group
MSB	Most Significant Bit
MSE	Mean Square Error
OPL	Outer Plexiform Layer
PB	Prediction Block
pps	pixels per second
PSF	Point Spread Function
PSNR	Peak Signal to Noise Ration
PTZ	Pan-Tilt Zoom
PU	Prediction Unit
QoE	Quality of Experience
RD	Rate-Distortion
RGB	Red Green Blue
RF	Receptive Field
ROC	Rank Order Coder
ROI	Region Of Interest
SCNS	Smart Camera Network System
SD	Standard Definition
SSIM	Structure SIMilarities
SVC	Scalable Video Coding
TAF	Threshold And Fire
TB	Transform Block
TEM	Time Encoding Machine
TU	Transform Unit
TV	TeleVision
PSD	Power Spectral Density
UEP	Unequal Error Protection
USC-SIPI	University of South California Signal and Image Processing Institution
VMS	Video Management Systems
WDoG	Weighted Difference of Gaussian
WSN	Wireless Sensor Networks
XSD	Cross-Segment Decoding

Bibliography

[Abdelgattah and Mohiuddin, 2010] Abdelgattah, E. and Mohiuddin, A. (2010). Performance analysis of multimedia compression algorithms. *International Journal of Computer Science and Information Technology (IJCSIT)*, 2(5).

[Adelson et al., 1984] Adelson, E., Andelson, C., Bergen, J., Burt, P., and Ogden, J. (1984). Pyramid methods in image processing. *RCA engineer*, 29(6):33–41.

[Adrian, 1926] Adrian, E. D. (1926). The impulses produces by sensory nerve endings. 61(1):47–72.

[Adrian, 1928] Adrian, E. D. (1928). *The Basis of Sensation*. The Action of the Sense Organs W. W Norton, New York.

[Ahmed et al., 1974] Ahmed, N., Natarajan, T., and Rao., K. R. (1974). Discrete cosine transform. *IEEE Transactions on Computers*, 100(1):90–93.

[Antonini et al., 1992] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies., I. (1992). Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220.

[Baylor et al., 1974] Baylor, D. A., Hodgkin, A. L., and Lamb, T. (1974). The electrical response of turtle cones to flashes and steps of light. *J Physiology*, (242):685–727.

[Baylor et al., 1979] Baylor, D. A., Lamb, T., and Yau, K. W. (1979). Responses of retinal rods to single photons. *J Physiology*, 288(613-634).

[Baylor et al., 1980] Baylor, D. A., Matthews, G., and Yau, K. W. (1980). Two components of electrical dark noise in toad retinal rod outer segments. *J Physiology*, (309):591–621.

[Bhaskaran and Konstantinides, 1997] Bhaskaran, V. and Konstantinides, K. (1997). *Image and Video Compression Standards Algorithms and Architectures*. Springer Science and Business Media LLC.

[Birch et al., 2010] Birch, P., Mitra, B., Bangalore, N. M., Rehman, S., Young, R., and Chatwin, C. (2010). Approximate bandpass and frequency response models of the difference of gaussian filter. *Optics Communications*, 283:4942–4948.

[Britanak, 2001] Britanak, V. (2001). *The Transform and Data Compression Handbook -Discrete Cosine and Sine Transforms*. CRC Press LLC.

[Burt and Adelson, 1983] Burt, P. and Adelson, E. (1983). The Laplacian pyramid as a compact image code. *IEEE Transactions on Communication*, 31(4):532–540.

[Cardarilli et al., 2013] Cardarilli, G. C., Cristini, A., Nunzio, L. D., Re, M., Salerno, M., and Susi, G. (2013). Spiking neural networks based on LIF with latency: Simulation and synchronization effects. *Asilomar Conference on Signals, Systems and Computers*.

[Chebbo et al., 2009] Chebbo, S., Durieux, P., and Pesquet-Popescu, B. (2009). Adaptive Deblocking filter for DCT coded video. *Proc. Fourth Int. Workshop on Video Processing and Quality Metrics for Consumer Electr. (VPQM'09),*.

[Cheng and Pedram, 2004] Cheng, W. C. and Pedram, M. (2004). Power Minimization in a Backlit TFTLCD Display by Concurrent Brightness and Contrast Scaling. *IEEE Transaction on Consumer Electronics*, 50(1):25–32.

[Christopoulos et al., 2000] Christopoulos, C., Skodras, A., and Ebrahimi, T. (2000). The JPEG2000 still image coding system: An overview. *Transactions on Consumer Electronics*, 46(4):1103–1127.

[Conway et al., 1996] Conway, J., Hardin, R. H., and Sloane, N. J. (1996). Packing line, planes, etc: Packings in Grassmanian spaces. *Experimental Mathematics*, 5(2):139–159.

[D. Cai and Freeman, 1997] D. Cai, G. D. and Freeman, R. (1997). Spatiotemporal receptive field organization in the Lateral Geniculate Nucleus of cats and kittens. *The American Physiological Society*, 22(3077):1045–1061.

[DeAngelis et al., 1993] DeAngelis, G., Ohzawa, I., and Freeman, R. (1993). Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex. i. general characteristics and postnatal development. *Journal of Neurophysiology*, 69(4):1091–1117.

[der Weken et al., 2002] der Weken, D. V., Nachtegael, M., and Kerre, E. E. (2002). Image quality evaluation. *Proceedings of the* 6^{th} *International Conference on Signal Processing*, 1(711-714).

[Doutsi et al., 2016] Doutsi, E., Fillatre, L., Antonini, M., and Gaulmin, J. (2016). Retina-inspired filtering. *IEEE Transactions on Image Processing*.

[Duffin and Schaeffer, 1952] Duffin, R. J. and Schaeffer, A. C. (1952). A class of nonharmonic Fourier series. *Transactions on American Mathematical Society*, 72:341–366.

[Eskicioglu and Fisher, 1995] Eskicioglu, A. M. and Fisher, P. S. (1995). Image quality measures and their performance. *IEEE Transactions on Communication*, 43(12):2959–2965.

[Everett, 1963] Everett, H. (1963). Generalized Lagrange miltiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11:399–417.

[Feng et al., 2003] Feng, W., Code, B., Shea, M., and Feng, W. (2003). Panoptes: A scalable architecture for video sensor networking application. *Proceedings of ACM Multimedia*, (151-167).

[Fernandes et al., 2015] Fernandes, F. C., Ducloux, X., Ma, Z., Faramarzi, E., Gendron, P., and Wen, J. (2015). The green metadata standard for energy-efficient video consumption. *IEEE Computer Society*.

[Fleet et al., 1985] Fleet, D., Hallett, P., and Jepson, A. (1985). Spatiotemporal inseparability in early visual processing. *Biological Cybernetics*, 52:153–164.

[Fowler and Pesquet-Popescue, 2007] Fowler, J. and Pesquet-Popescue, B. (2007). An overview on wavelets in source coding, communications, and networks. *EURASIP Journal on Image and Video Processing*, page 27.

[Gao et al., 2013] Gao, W., Tian, Y., T.huang, Ma, S., and X.Zhang (2013). IEEE 1857 standard empowering smart video surveillance systems. *IEEE Intelligent Systems*.

[Gersho and Gray, 1992] Gersho, A. and Gray, R. M. (1992). *Vector Quantization and Signal Compression*. Kluwer Academic.

[Gerstner and Kistler, 2002] Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models: An Introduction*. Cambridge University Press, New York, NY, USA.

[Gilbert and Nocedal, 1992] Gilbert, J. C. and Nocedal, J. (1992). Global Convergence Properties of Conjugate Gradient Methods for Optimization. *SIAM Journal on Optimization*, (1):21–42.

[Grois et al., 2013] Grois, D., Marpe, D., A.Mulayoff, and O.Hadar (2013). Performance comparison of h.265/mpeg-hevc, vp9, and h.264/mpeg-avc encoders. 30^{th} *Picture Coding Symposium 2013 (PCS 2013)*.

[Haar, 1910] Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, pages 331–371.

[Hamidi and Pearl, 1976] Hamidi, M. and Pearl, J. (1976). Comparison of the cosine and Fourier transforms of Markov-I signals. *IEEE Transactions on Accustic Speech and Signal Processing ASSP*, 24:428–429.

[Hartung et al., 2006] Hartung, C., Han, R., Seielstad, C., and Holbrook, S. (2006). FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments,. *Proceedings of 4th International Conference on Mobile Systems, Applications and Services*, pages 28–41.

[Heeger, 2000] Heeger, D. (2000). Poisson model of spike generation. Technical report.

[Hengstler et al., 2006] Hengstler, S., Prashanth, D., Fong, S., and Ahgajan, H. (2006). MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. *Proceedings of International Symposium of Information Processing Sensor and Networks*, 360-369.

[Hérault and Durette, 2007] Hérault, J. and Durette, B. (2007). *Modeling Visual Perception for Image Processing*, pages 662–675. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Horé and Ziou, 2010] Horé, A. and Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. 20^{th} *International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey,*, pages 2366–2369.

[Hourdakis et al., 2005] Hourdakis, J., Morris, T., Michalopoulos, P., and Wood, K. (2005). Advanced portable wireless measurement and observation station,. Technical Report CTS 05-07, Center for Transportation Studies in Univ. Minnesota, Minneapolis, MN, USA.

[Howard and Vitter, 1992] Howard, P. and Vitter, J. S. (1992). Practical implementation of arithmetic coding. Technical Report 92-12, Brown University Department of Science.

[Huang et al., 2013] Huang, T.-H., Shih, K.-T., Yeh, S.-L., and Chen, H. H. (2013). Enhancement of backlight-scaled images. *IEEE Transactions on Image Processing*, 22(12):4587–4597.

[Hubel, 1963] Hubel, D. (1963). The visual cortex of the brain. *American Scientist the magazine of Sigma Xi, The Scientific Research Society*, 209(5):54–63.

[Huffman, 1952] Huffman, D. (1952). A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the I.R.E*, pages 1098–1101.

[ISO/IEC 10918-1:1994, 1994] ISO/IEC 10918-1:1994 (1994). Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines. *JPEG*.

[ISO/IEC 10918-1:2000, 2000] ISO/IEC 10918-1:2000 (2000). Information technology - ceneric coding of movind pictures and associated audio information: Video. *MPEG-2 Video*.

[ISO/IEC 10918-1:2004, 2004] ISO/IEC 10918-1:2004 (2004). Information technology - coding of audio-visual objects - part 2:Visual. *MPEG-4 Video*.

[Jolivet et al., 2004] Jolivet, R., Lewis, T. J., and Gerstner, W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. *Journal of Neurophysiology*, 92:959–976.

[Kawamura et al., 2011] Kawamura, A., Yoshimitsu, Y., Kajitani, K., Naito, T., Fujimura, K., and Kamijo, S. (2011). Smart camera network system for use in railway stations. *Proceedings of International Conference on Systems, Man and Cybernetics*.

[Kolb, 2004] Kolb, H. (2004). How the retina works. *American Scientist the magazine of Sigma Xi, The Scientific Research Society*, 91:28–35.

[Konrad, 2000] Konrad, J. (2000). Motion detection and estimation. *Handbook of Image and Video Processing*, 207(225).

[Kovacevic and Chebina, 2008] Kovacevic, J. and Chebina, A. (2008). An introduction to frames. *Signal Processing*, 2(1):1–94.

[Kovačević and Vetterli, 1992] Kovačević, J. and Vetterli, M. (1992). Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for r^n. *IEEE Transactions on Information Theory*, 38(2):533–555.

[Kuffler, 1952] Kuffler, S. (1952). Neurons in the retina: Organization, inhibition and excitation problems. *Cold Spring Harbor Symposia on Quantitative Biology*, 17:281–292.

[Kulkarni et al., 2005] Kulkarni, P., Ganesan, D., Shenoy, P., and Lu, Q. (2005). SensEye: A mulit-tier camera sensor network. *Proceedings of ACM Multimedia*, pages 229–238.

[Lazar and Pnevmatikakis, 2011] Lazar, A. A. and Pnevmatikakis, A. (2011). Video time encoding machines. *IEEE Transaction on Neural Networks*, 22(3):461–473.

[Leader, 2004] Leader, S. (2004). Telecommunications handbook for transportation professionals—the basics of telecommunications,. Technical Report FHWA-HOP-04-034, Federal Highway Administration, Washington, DC, USA,.

[Lee et al., 2014] Lee, J. H., Delbruck, T., Pfeiffer, M., Park, P. K. J., Shin, C.-W., Ryu, H. E., and Kang, B. C. (2014). Real-time gesture interface based on event-driven processing from stereo silicon retinas. *IEEE Transaction on Neural Networks and Learning Systems*.

[Liu et al., 2009] Liu, H., Agam, Y., Madsen, J. R., and Kreiman, G. (2009). Timing, timing, timing: Fast decoding of object information intracranial field ptentials in human visual cortex. *Neuron*, 62:281–290.

[Lloyd, 1982] Lloyd, S. (1982). Least Square Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

[Lo et al., 2003] Lo, B. P. L., Sun, J., and Velastin, S. A. (2003). Fusing visual and audio information in a distributed intelligent surveillance system for public transport systems. *Acta Automatica Sinica*, 29(3):393–207.

[Luo, 2011] Luo, N. (2011). A wireless traffic surveillance system using video analytics. Master's thesis, Department Computer Science Engineering, University of North Texas.

[Mallat, 1999] Mallat, S. (1999). *A Wavelet Tour of Signal Processing*. Academic Press, 2^{nd} edition.

[Marr, 1982] Marr, D. (1982). *Vision*. The MIT Press.

[Marr and Hildreth, 1980] Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217.

[Masland, 2001] Masland, R. (2001). The fundamental plan of the retina. *Natural Neuroscience*, 4(9):877–886.

[Masland, 2011] Masland, R. (2011). Cell populations of the retina: The proctor lecture. *Investigatve Ophthalmology and visual Science*, 52(7):4581–4591.

[Masmoudi et al., 2012] Masmoudi, K., Antonini, M., and Kornprobst, P. (2012). Frames for exact inversion of the rank order coder. *IEEE Transaction on Neural Networks*, 23(2):353–359.

[Masmoudi et al., 2013] Masmoudi, K., Antonini, M., and Kornprobst, P. (2013). Streaming an image through the eye: The retina seen as a dithered scalable image coder. *Signal processing Image Communication*, 28(8):856–869.

[Masmoudi et al., 2010] Masmoudi, K., Antonini, M., Kornprobst, P., and Perrinet, L. (2010). A novel bio-inspired static image compression scheme for noisy data transmission over low-bandwidth channels. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3506–3509.

[McHugh et al., 2009] McHugh, J., Konrad, J., Sligrama, V., and Jodoin, P. (2009). Foreground-adaptive backround subtraction. *IEEE Signal Processing Magazine*, 16(5):390–393.

[Meister and Berry, 1999] Meister, M. and Berry, M. J. (1999). The neural code of the retina. *Neuron*, 22(435-450).

[Mishra and Singh, 2015] Mishra, E. and Singh, K. K. (2015). Comparision of various lossless image compression techniques. *International Journal of Engineering Research and Applications ISSN*, 5(6):36–39.

[Mukherjee et al., 2013] Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., Xu, Y., and Blakeltje, R. (2013). The latest open-source video codec vp9 - an overview and preliminary results. *Picture Coding Symposium (PCS)*, pages 390 – 393.

[N. Li and Wang, 2010] N. Li, B. Yan, G. C. P. G. and Wang, J. (2010). Design and implementation of a sensor-based wireless camera system for continuous monitoring in assistive environments. *Journal Personal and Ubiquitous Computing*.

[Ögmen and Herzog, 2010] Ögmen, H. and Herzog, M. H. (2010). The geometry of visual perception: Retinotopic and nonretinotopic representations in the human visual system. *Proceedings of the IEEE*, 98(3):479–492.

[Ortega and Ramchandran, 1998] Ortega, A. and Ramchandran, K. (1998). Rate-distorsion methods for image and video compression. *IEEE Signal Processing Magazine*.

[Ostermann et al., 2004] Ostermann, J., Bormans, J., List, P., Marpe, D., Narroschke, N., Pereira, F., Stockhammer, T., and Wedi, T. (2004). Video coding with H.264/AVC: tools, performance and complexity. *IEEE Circuit and System magazine*, 4(1):7–28.

[Parisot, 2003] Parisot, C. (2003). *Allocations basées modèles et transformée en ondelettes au fil de l'eau pour le codage des images et des vidéos*. PhD thesis, University of Nice, Sophia-Antipolis.

[Pereira, 2000] Pereira, F. (2000). MPEG-2: Why, what, how and when? *Signal Processing: Image Communication*, 15(4-5):271–279.

[Pereira and Ebrahimi, 2002] Pereira, F. and Ebrahimi, T. (2002). The MPEG-4 book. *Prentice Hall Professional*.

[Perrett et al., 1982] Perrett, D. I., Rolls, E. T., and Caan, W. C. (1982). Visual neurons responsive to faces in the monkey temporal cortec. *Experimental Brain Research*, 47:329–342.

[Perrinet et al., 2004] Perrinet, L., Samuelides, M., and Thorpe, S. (2004). Coding static natural images using spiking event times: so neurons cooperate? *IEEE Transaction on Neural Networks*, 15(5):1164–1175.

[Perrinet, 2010] Perrinet, L. U. (2010). Role of homeostasis in learning sparse representations. *Neural Computation, Massachusetts Institute of Technology Press (MIT Press),*, 22(7):1812–36.

[Perrinet, 2015] Perrinet, L. U. (2015). *Biologically inspired computer vision*, chapter Sparse models for Computer Vision. Number 14. Wiley-VCH Verlag GmbH & Co. KGaA.

[Pesquet-Popescu et al., 2014] Pesquet-Popescu, B., Cagnazzo, M., and Dufaux, F. (2014). Motion-estimation - A Video Coding Viewpoint. *Academic Press Library in Signal Processing: Image and Video Compression and Multimedia*, 5:27.

[Pesquet-Popescu and Pesquet, 2011] Pesquet-Popescu, B. and Pesquet, J. (2011). Wavelets and image processing. *Image Processing*, pages 181–204.

[Piccardi, 2004] Piccardi, M. (2004). Background subtraction techniques: a review. *IEEE International Conference on Systems, man and cybernets*.

[Poggio and Koch, 1986] Poggio, T. and Koch, C. (1986). Synpases that commpute motion. *Scientific American*, 256:46–71.

[Potter et al., 2014] Potter, M. C., Wyble, B., Hagmann, C. E., and McCourt, E. S. (2014). Detecting meaning in RSVP at 13ms per picture. *Attention, Perception & Phychophysics*, 76(2):270–279.

[Regazzoni et al., 2010] Regazzoni, C., Cavallaro, A., Wu, Y., Konrad, J., and Hampapur, A. (2010). Video analytics for Surveillance: Theory and Practice. *IEEE Signal Processing Magazine*, 22(9):3285–3496.

[Richardson, 2011] Richardson, I. E. (2011). *The H.264 Advanced Video Compression Standard*.

[Rieke et al., 1999] Rieke, F., Warland, D., Steveninck, R. R., and Bialek, W. (1999). *Spikes: Exploring the neural code*. Computational Neuroscience. MIT Press.

[Rissanen and Langdon, 1979] Rissanen, J. and Langdon, G. G. (1979). Arithmetic coding. *IBM Journal of Research and Development*, 23(2):149–162.

[Rullen and Thorpe, 2001] Rullen, R. V. and Thorpe, S. J. (2001). Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. *Natural Neuroscience*, 13:1255–1283.

[Salamo and Jakobs, 1996] Salamo, G. and Jakobs, T. (1996). Laser pointers: are they safe for use by children? *Augmentative and Alternative Communication AAC*, 12:47–51.

[Santa-Cruz et al., 2002] Santa-Cruz, D., Grosbois, R., and Ebrahimi, T. (2002). Jpeg 2000 performance evaluation and assessment. *Signal Processing: Image Communication*, 17(1):113–130.

[Schwarz et al., 2007] Schwarz, H., Marpe, D., and Wiegand, T. (2007). Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transaction on Circuits and Systems for Video Technology*, 17(9):1103–1120.

[Shapiro, 1993] Shapiro, J. M. (1993). Embedded Image Coding Using Zerotrees of Wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462.

[Shewchuk, 1994] Shewchuk, J. R. (1994). An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.

[Sikora, 1997] Sikora, T. (1997). Mpeg digital video coding standards. *IEEE Signal Processing Magazine*, 14(5):82–100.

[Stiller and Konrad, 1999] Stiller, C. and Konrad, J. (1999). Estimation motion in image sequences. *IEEE Signal Processing Magazine*, 16(4):70–91.

[Sullivan et al., 2012] Sullivan, G., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transaction on Circuits and Systems for Video Technology*, 22(12):1649–1669.

[T. Wiegand and Luthra, 2003] T. Wiegand, G. J. Sullivan, G. B. and Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576.

[Taubman, 2000] Taubman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170.

[Taubman and Marcellin, 2002] Taubman, D. and Marcellin, M. (2002). *JPEG2000 Image Compression Fundamentals Standards and Practice*. Kluwer Academic Publishers Dordrecht.

[Teo and Heeger, 1994] Teo, P. C. and Heeger, D. (1994). Perceptual image distortion. *Proceedings of the* 1^{st} *IEEE International Conference on Image Processing*, (982-986).

[ter Haar Romeny, 2003] ter Haar Romeny, B. M. (2003). *Front-End Vision and Multi-Scale Image Analysis*, computational imaging and vision The front-end visual system: the retina, pages 153–165. Springer Netherlands.

[Thorpe et al., 2001] Thorpe, S., Delorme, A., and Rullen, R. V. (2001). Spike-based strategies for rapid processing. *Neural Networks*, 14:715–725.

[Thorpe, 1990] Thorpe, S. J. (1990). Spike arrivel times: A highly efficient coding scheme for neural network. *Parallel Processing in neural Systems and Computers*, pages 91–94.

[Thorpe and Gautrais, 1998] Thorpe, S. J. and Gautrais, J. (1998). Rank Order Coding: A new coding scheme for rapid processing in neural network. *Computational Neuroscience: Trends in Research*, pages 113–118.

[Thorpe and Imbert, 1989] Thorpe, S. J. and Imbert, M. (1989). Biological constraints on connectionist models. *Connectionism in perspective*, (63-92).

[Tomar and Jain, 2016] Tomar, R. R. S. and Jain, K. (2016). Lossless image compression using differential pulse code modulation and its application. *Internationl Journal of Signal Processing, Image Processing and Pattern Recognition*, 9(1):197–202.

[Tsai, 1998] Tsai, M. J. (1998). Stack-Run-End compression for low bit rate color image communication. *IEEE International Conference on Acoustics, Speech and Signal Processing*.

[Tsai et al., 1996] Tsai, M. J., Villasenor, J. D., and Chen, F. (1996). Stac-Run Coding for Low Bit Rate Image Communication. *IEEE International Conference in Image Processing (ICIP)*.

[Tseng et al., 2006] Tseng, C., Wang, H., and Yang, J. (2006). Enhanced intra4x4 mode decision for H.264/AVC codes. *IEEE Transaction on Circuits and Systems for Video Technology*, 15(3):378–401.

[Usevitch, 1996] Usevitch, B. E. (1996). Optimal bit allocation for biorthogonal wavelet coding. *In Proceedings of Data Compression Conference*, pages 387–395.

[VanEssen et al., 2005] VanEssen, D., Adelson, C., and Felleman, D. (2005). Information processing in the primate visual system: An integrated systems perspective. *Science*, 255(5043):419–423.

[Vitter, 1987] Vitter, J. S. (1987). Design and analysis of dynamic huffman codes. *Association for Computing Machinery*, 34(4):825–845.

[Wang et al., 2005] Wang, H., Zhai, F., Eisenberg, Y., and Katsaggelos, A. K. (2005). Cost-distrortion optimized unequal error protection for object-based video communications. *IEEE Transaction on Circuits and Systems for Video Technology*, 15(12):1505–1516.

[Wang et al., 2009] Wang, X., Wang, S., and Bi, D. (2009). Distributres viseal-target-surveillance system in wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 39(5):1134–1146.

[Wang et al., 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

[Weber, 1977] Weber, A. (1977). The USC-SIPI Image Database.

[Witten et al., 1987] Witten, I. H., Neal, R. M., and Cleary, J. G. (1987). Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540.

[Wohrer and Kornprobst, 2009] Wohrer, A. and Kornprobst, P. (2009). Virtual retina: A biological retina model and simulator, with constrast gain control. *Journal of Computational Neuroscience*, 26(2):219–249.

[Wohrer et al., 2009] Wohrer, A., Kornprobst, P., and Antonini, M. (2009). Retinal filtering and image reconstruction. Technical report, Inria Research ReportRR- 6960.

[Ye et al., 2013] Ye, Y., Ci, S., Katsaggelos, A., Liu, Y., and Qian, A. Y. (2013). Wireless video surveillance: A survey. *IEEE Access*, 1:646–660.