

# Centralized and Distributed Semi-Parametric Compression of Piecewise Smooth Functions

Varit Chaisinthop

A Thesis submitted in fulfilment of requirements for the degree of  
Doctor of Philosophy of Imperial College London

Communications and Signal Processing Group  
Department of Electrical and Electronic Engineering  
Imperial College London

2010

I would like to dedicate this thesis to my loving parents ...

# Statement of Originality

I declare that the intellectual content of this thesis is the product of my own research work under the guidance of my thesis advisor, Dr. Pier Luigi Dragotti. Any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. The material of this thesis has not been submitted for any degree at any other academic or professional institution.

# Abstract

This thesis introduces novel wavelet-based semi-parametric centralized and distributed compression methods for a class of piecewise smooth functions. Our proposed compression schemes are based on a non-conventional transform coding structure with simple independent encoders and a complex joint decoder.

Current centralized state-of-the-art compression schemes are based on the conventional structure where an encoder is relatively complex and nonlinear. In addition, the setting usually allows the encoder to observe the entire source. Recently, there has been an increasing need for compression schemes where the encoder is lower in complexity and, instead, the decoder has to handle more computationally intensive tasks. Furthermore, the setup may involve multiple encoders, where each one can only partially observe the source. Such scenario is often referred to as distributed source coding.

In the first part, we focus on the dual situation of the centralized compression where the encoder is linear and the decoder is nonlinear. Our analysis is centered around a class of 1-D piecewise smooth functions. We show that, by incorporating parametric estimation into the decoding procedure, it is possible to achieve the same distortion-rate performance as that of a conventional wavelet-based compression scheme. We also present a new constructive approach to parametric estimation based on the sampling results of signals with finite rate of innovation.

The second part of the thesis focuses on the distributed compression scenario, where each independent encoder partially observes the 1-D piecewise smooth function. We propose a new wavelet-based distributed compression scheme that uses parametric esti-

---

mation to perform joint decoding. Our distortion-rate analysis shows that it is possible for the proposed scheme to achieve that same compression performance as that of a joint encoding scheme.

Lastly, we apply the proposed theoretical framework in the context of distributed image and video compression. We start by considering a simplified model of the video signal and show that we can achieve distortion-rate performance close to that of a joint encoding scheme. We then present practical compression schemes for real world signals. Our simulations confirm the improvement in performance over classical schemes, both in terms of the PSNR and the visual quality.

# Acknowledgements

First of all, I would like to express my deepest gratitude to my PhD advisor, Dr. Pier Luigi Dragotti, for his constant support, guidance and encouragement. He has always gone the extra mile in supervising and giving his students the opportunities that many could only dream of. I thank him for his enthusiasm to share his knowledge, ideas and wisdom, and for always greeting me with a smile.

I would then like to thank all my friends at Imperial. In particular, I am grateful to Kenko, Ake, Eagle, Nine, Kung and Kwang for the many joyful moments and laughs that we shared. I also wish to thank my lifelong friends at home, Tei, Jah, Phon and Nan, for always being there for me through my highs and lows.

Last but not least, I would never have finished this thesis if it was not for the support of my loving family who always believe in me. I am forever grateful to my Mum, my Dad and my sisters, Pim and Pat, for always being ever so kind, supportive and understanding.

# Contents

<b>Statement of Originality</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xvi</b>
<b>Abbreviations and Notations</b>	<b>xxii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem Statement and Motivations . . . . .	2
1.3 Previous Related Work . . . . .	5
1.4 Thesis Outline . . . . .	6
1.5 Original Contribution . . . . .	7
<b>Chapter 2. Theoretical Foundations of Centralized and Distributed Transform Coding</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Foundations of Centralized Transform Coding . . . . .	10
2.2.1 Transform Coding Overview . . . . .	10
2.2.2 Entropy code . . . . .	12
2.2.3 Scalar Quantization . . . . .	13
2.2.4 Distortion rate of scalar quantization . . . . .	15
2.2.5 Bit Allocation . . . . .	16
2.2.6 Optimal Transform for Gaussian Sources . . . . .	18
2.3 Distributed Transform Coding . . . . .	19
2.3.1 Foundations of Distributed Coding . . . . .	20
2.3.2 Distributed Karhunen-Loève Transform . . . . .	21
2.3.3 Practical Distributed Transform Coding with Side Information . . . . .	24
2.4 Summary . . . . .	27

<b>Chapter 3. State-of-the-Art Compression Methods with Wavelets</b>	<b>28</b>
3.1 Introduction . . . . .	28
3.2 The Wavelet Transform . . . . .	29
3.3 Lipschitz Regularity . . . . .	32
3.4 Linear and Nonlinear Approximation with Wavelets . . . . .	34
3.4.1 Linear approximation . . . . .	34
3.4.2 Nonlinear approximation . . . . .	36
3.5 Wavelet-Based Compression . . . . .	37
3.5.1 Distortion-rate results . . . . .	37
3.5.2 Embedded coding . . . . .	39
3.5.3 Wavelets in scalable video compression . . . . .	41
3.6 Summary . . . . .	42
<b>Chapter 4. Sampling of Signals with Finite Rate of Innovation</b>	<b>44</b>
4.1 Introduction . . . . .	44
4.2 Sampling Setup . . . . .	45
4.3 Signals with Finite Rate of Innovation . . . . .	46
4.4 Polynomial Reproducing Kernels . . . . .	48
4.4.1 Polynomial Reproduction Property . . . . .	48
4.4.2 Polynomial Reproduction Coefficients . . . . .	49
4.4.3 Moment-Samples Relationship . . . . .	51
4.5 Sampling Schemes for FRI Signals . . . . .	52
4.5.1 Streams of Diracs . . . . .	52
4.5.2 Streams of Differentiated Diracs . . . . .	54
4.5.3 Piecewise Polynomial Functions . . . . .	56
4.6 Sampling of 2-D FRI Signals . . . . .	57
4.6.1 2-D Sampling Setup . . . . .	57
4.6.2 2-D Signals with Finite Rate of Innovation . . . . .	58
4.6.3 2-D Polynomial Reproducing Kernels . . . . .	59
4.7 Summary . . . . .	62
<b>Chapter 5. Centralized Semi-Parametric Compression of Piecewise Smooth Functions</b>	<b>63</b>
5.1 Introduction . . . . .	63
5.2 Modeling of Piecewise Smooth Functions . . . . .	65
5.3 Semi-Parametric Compression Strategy . . . . .	66
5.3.1 Linear Approximation . . . . .	66
5.3.2 Semi-Parametric Compression Algorithm . . . . .	66
5.4 Cramér-Rao Bound of Parametric Estimation . . . . .	71
5.4.1 Derivation of Cramér-Rao Bound . . . . .	71
5.4.2 Evaluation of Cramér-Rao Bound . . . . .	73



5.5	Distortion-Rate Analysis . . . . .	75
5.6	Constructive Compression Algorithms . . . . .	81
5.6.1	FRI-Based Parametric Estimation Algorithm . . . . .	81
5.6.2	FRI-based Semi-Parametric Compression with Nonlinear Encoder . . . . .	83
5.7	Simulation Results . . . . .	84
5.7.1	Parametric Estimation Algorithm . . . . .	84
5.7.2	Semi-Parametric Compression Algorithm . . . . .	84
5.7.3	Semi-Parametric Compression Algorithm with Nonlinear Encoder . . . . .	88
5.8	Conclusion . . . . .	89
 <b>Chapter 6. Distributed Semi-Parametric Compression of Piecewise Smooth Functions</b>		<b>91</b>
6.1	Introduction . . . . .	91
6.2	Signal and Disparity Models . . . . .	92
6.3	Distributed Semi-Parametric Compression Strategies . . . . .	93
6.4	Distortion-Rate Analysis . . . . .	96
6.4.1	Disparity by Translation . . . . .	97
6.4.2	Disparity by Translation with Prediction Error . . . . .	100
6.4.3	Comparison with Independent and Joint Compression . . . . .	104
6.4.4	Extension to $N$ Signals . . . . .	106
6.5	Simulation results . . . . .	108
6.6	Conclusion . . . . .	109
 <b>Chapter 7. Distributed Semi-Parametric Compression of 2-D Signals</b>		<b>112</b>
7.1	Introduction . . . . .	112
7.2	Wavelet Approximation in 2-D . . . . .	113
7.3	Bi-level Polygon Case Study . . . . .	114
7.3.1	Signal Model . . . . .	114
7.3.2	Distributed Compression Strategy . . . . .	115
7.3.3	Quantization Error in Moment-Samples Relationship . . . . .	117
7.3.4	Distortion-Rate Analysis . . . . .	118
7.4	Real Objects with Affine Transform Disparity Model . . . . .	123
7.4.1	Signal Model . . . . .	123
7.4.2	Estimating Affine Parameters from Moments . . . . .	123
7.4.3	Distributed Compression Scheme . . . . .	127
7.4.4	Simulation Results . . . . .	128
7.5	Applications to Real Signals . . . . .	133
7.5.1	Array of Images . . . . .	133
7.5.2	Real Video Sequence with Fixed Background . . . . .	139
7.6	Conclusion . . . . .	144

<b>Chapter 8. Conclusions</b>	<b>145</b>
8.1 Thesis Summary . . . . .	145
8.2 Future Work . . . . .	147
<b>Appendix</b>	<b>150</b>
A.1 Proof of Theorem 10 . . . . .	150
A.2 Evaluation of Cramér-Rao Bound with B-Spline Scaling Functions . . .	152
<b>Bibliography</b>	<b>154</b>

# List of Figures

1.1	Illustration of the wavelet transform of an image. (a) original image; (b) the corresponding Haar wavelet transform with two decomposition levels.	3
2.1	Transform coding structure. A typical compression scheme consists of three elements: linear transform, quantization and entropy coding. . . .	11
2.2	Slepian-Wolf achievable rate region for distributed compression of two statistically dependent i.i.d. sources [52]. . . . .	21
2.3	Distributed KLT problem setups: (a) $L$ -terminals scenario where Encoder $l$ has access to the subvector $\mathbf{x}_l$ and the reconstruction of is performed jointly; (b) Two-terminals scenario where Encoder 2 applies a fixed known transform $T_2$ to the observed vector $\mathbf{x}_2$ and the decoder receives a noisy version $\mathbf{y}_2 = T_2\mathbf{x}_2 + \mathbf{z}_2$ . . . . .	22
2.4	Practical distributed transform coding scheme. A Wyner-Ziv encoder, which consists of a quantizer followed by a Slepian-Wolf (S-W) encoder, replaces the quantizer and the entropy encoder in the standard transform coding structure. . . . .	26
3.1	Wavelet decomposition: (a) a uniformly $\alpha$ -Lipschitz smooth; (b) a piecewise smooth function with $\alpha$ -Lipschitz pieces. The $N$ -term linear approximation error decays as $\varepsilon_l(N, f) \sim N^{-2\alpha}$ in (a) whereas $\varepsilon_l(N, f) \sim N^{-1}$ in (b). By using an adaptive grid of nonlinear approximation and, instead, keeping the $N$ largest coefficients, which include the wavelet coefficients in the cone of influence of singularities, we achieve a nonlinear approximation error with a decay of $\varepsilon_n(N, f) \sim N^{-2\alpha}$ . . . . .	36

3.2	Parent-offspring dependencies in the spatial-orientation tree of wavelet coefficients found in SPIHT algorithm [43]. . . . .	41
3.3	Examples of MCTF implementation with the LeGall 5/3 lifting scheme. Three decomposition levels are shown. The video frames are divided into even and odd frames. The even frames are used to form the prediction of the odd frames by means of motion compensation. Each motion compensated prediction is multiplied with the prediction coefficient and the high-pass subband is calculated by taking the difference. Similarly, the high-pass subbands are multiplied by the update coefficient to produce the low-pass subbands. Here, the prediction steps are denoted with $P$ and the update steps are denoted with $U$ . . . . .	43
4.1	Generic 1-D sampling setup: a continuous signal $f(t)$ , $t \in \mathbb{R}$ , is filtered by a sampling kernel $h(t) = \varphi(-t/T)$ and uniformly sampled by $\sum_n \delta(t-nT)$ to obtain a set of discrete samples $y_n = \langle f(t), \varphi(t/T - n) \rangle$ , where $n \in \mathbb{Z}$ . . . . .	46
4.2	Examples of 1-D signals with finite rate of innovation: (a) streams of Diracs; (b) piecewise polynomial signals. . . . .	47
4.3	Polynomial reproduction property of 1-D kernels as given by Equation (4.4): (a) Daubechies scaling function of order 3; (b) zeroth order polynomial ( $p = 0$ ); (c) first order polynomial ( $p = 1$ ); (d) second order polynomial ( $p = 2$ ); the blue lines in (b),(c) and (d) represent the reproduced polynomials and the red lines are the scaled and shifted versions of the db3 scaling function. . . . .	50
4.4	Generic 2-D sampling setup: a continuous signal $f(x, y)$ , $x, y \in \mathbb{R}$ , is filtered by a sampling kernel $h(x, y) = \varphi(-x/T_x, -y/T_y)$ and uniformly sampled by $\sum_m \sum_n \delta(x-mT_x, y-nT_y)$ to obtain a set of discrete samples $S_{m,n} = \langle f(x, y), \varphi(x/T_x - m, y/T_y - n) \rangle$ , where $m, n \in \mathbb{Z}$ and $T_x, T_y \in \mathbb{R}^+$ . . . . .	58
4.5	Examples of 2-D signals with finite rate of innovation: (a) a set of Diracs; (b) a bilevel polygon image. . . . .	59

4.6	Polynomial reproduction property of 2-D kernels as given by Equation (4.24): (a) 2-D Daubechies scaling function of order 4; (b) zeroth order polynomial with $p = 0, q = 0$ ; (c) first order polynomial along $x$ -direction with $p = 1, q = 0$ ; (d) first order polynomial along $y$ -direction with $p = 0, q = 1$ . Note that the grid scale used in (a) is different from (b),(c) and (d) . . . . .	61
5.1	Illustration of a piecewise smooth function. (a) A piecewise smooth function $f(t) = f_p(t) + f_\alpha(t)$ ; (b) coefficients of $f(t)$ , the high-pass coefficients in the boxes are in the cone of influence of discontinuities represented by the index set $\mathcal{J}_p$ and the coefficients outside the boxes are in the set $\mathcal{J}_\alpha$ ; (c) a piecewise polynomial function $f_p(t)$ ; (d) coefficients of $f_p(t)$ , the high-pass coefficients in the boxes correspond to the index set $\mathcal{J}_p$ , which can be predicted using parametric estimation; (e) a smooth $\alpha$ -Lipschitz function $f_\alpha(t)$ ; (f) coefficients of $f_\alpha(t)$ . . . . .	68
5.2	Evaluation of $J_{t_0,A}$ in (5.18) using B-spline scaling functions of order $P$ , where $0 < P \leq 10$ . (a) Plots of $(J_{t_0,A})_{11}$ against $P$ . (b) Plots of $(J_{t_0,A})_{22}$ against $P$ . . . . .	75
5.3	Error in the estimation of $t_0$ . (a) MSE in the estimation of $t_0$ using Algorithm 2 in comparison with the corresponding CRB. (b) retrieved locations $\bar{t}_0$ using Algorithm 2 and the corresponding CRB. . . . .	85
5.4	Error in the estimation of $A$ . (a) MSE in the estimation of $A$ using Algorithm 2 in comparison with the corresponding CRB. (b) retrieved amplitudes $\bar{A}$ using Algorithm 2 and the corresponding CRB. . . . .	86
5.5	Distortion-rate plots (log scale) for the compression of piecewise smooth function with one discontinuity and $\alpha = 1$ . The proposed semi-parametric compression scheme achieves the decay rate of $R^{-2\alpha}$ whereas the distortion of a linear approximation based scheme decayed as $R^{-1}$ . . . . .	86
5.6	Plots of the original signal, the reconstructed signals with a linear approximation based scheme and a semi-parametric scheme. The original function is piecewise smooth with one discontinuity and $\alpha = 1$ . . . . .	87

5.7	Distortion-rate plots (log scale) for the compression of piecewise smooth function with one discontinuity and $\alpha = 0.95$ . The variance $\sigma_\alpha^2$ of $c_{J,n} = \langle f_\alpha(t), \varphi_{J,n}(t) \rangle$ causes the behavior of the $D(R)$ curve to change from $D(R) \sim R^{-2\alpha}$ to $D(R) \sim C\sigma_\alpha^2$ after a certain rate point. . . . .	87
5.8	Distortion-rate plots (log scale) for the compression of piecewise smooth function with two discontinuities and $\alpha = 2.5$ . The proposed semi-parametric compression scheme achieves the decay rate of $R^{-2\alpha}$ . . . . .	88
5.9	Plots of the original signal, the reconstructed signals with a linear approximation based scheme and a semi-parametric scheme. The original function is piecewise smooth with two discontinuities and $\alpha = 2.5$ . . . . .	88
5.10	PSNR plots of semi-parametric compression algorithm with nonlinear encoder; at high rates, the combined semi-parametric compression with 1-D SPIHT achieved a gain of approximately 1-3 dB. . . . .	89
6.1	Distributed compression scenario. A set of $N$ functions $\{f_i(t)\}_{1 \leq i \leq N}$ are individually encoded but are jointly decoded. . . . .	93
6.2	Examples of the scan lines taken from stereo images shown in (a) and (b). (c) A scan line of the first image $f_1(x, y')$ ; (d) a scan line from the second image $f_2(x, y')$ ; (e) the prediction error given by $f_2(x, y') - f_1(x - \tau_x, y')$ , where $\tau_x$ denotes the shift parameter. . . . .	94
6.3	Distributed compression problem setup using the disparity-by-translation model with prediction error. Each function is piecewise smooth and $f_i(t) = f_1(t - \tau_i) + \epsilon_{i\alpha}(t)$ , $i = 2, \dots, N$ . . . . .	95
6.4	Plots of the function in (6.24). The gain in compression performance over an independent scheme increases with the degree of smoothness of the function. From the log-log plot in (b), we can see that the performance gain exhibits a super-exponential decay with increasing $\beta^*$ . . . . .	105
6.5	The plot of $\frac{D(R)}{D_{joint}(R)}$ against $\frac{\beta^*}{\beta}$ . The redundancy increases linearly with the ratio $\frac{\beta^*}{\beta}$ . . . . .	106
6.6	Plots of the two original piecewise smooth functions $f_1(t)$ and $f_2(t)$ used in the first simulation, where $f_2(t) = f_1(t - \tau)$ . . . . .	109

6.7	Distortion-rate plots of compression schemes based on disparity-by-translation model (log-log scale in (a) and decibel scale in (b)). The proposed distributed semi-parametric compression scheme in Algorithm 4 outperforms the independent compression scheme by a factor of $\frac{D_{ind}(R)}{D(R)} \approx 28$ ( $\approx 15$ dB), where the predicted factor is $2^{2\alpha} = 36.76$ (15.65 dB). The joint encoding scheme outperforms the proposed scheme by approximately 2 dB	110
6.8	Plots of the two original piecewise smooth functions $f_1(t)$ and $f_2(t)$ used in the second simulation, where $f_2(t) = f_1(t - \tau) + \epsilon_\alpha(t)$ .	110
6.9	Distortion-rate plots of compression schemes based on disparity-by-translation model with prediction error (log-log scale in (a) and decibel scale in (b)). The proposed distributed semi-parametric compression scheme in Algorithm 4 outperforms the independent compression scheme by a factor of $\frac{D_{ind}(R)}{D(R)} \approx 2.33$ , which is close to the predicted value of 2.26 (from (6.24)). The achieved $D(R)$ is comparable to that of the joint encoder scheme.	111
7.1	Examples of equilateral bilevel polygon signals: (a) a polygon with four corner points; (b) illustration of the distortion due to the error in the retrieved translation vector.	114
7.2	Plots of $D(R)$ functions (in log scale) of the proposed distributed semi-parametric, independent and joint compression schemes for four sequences of translating bi-level polygons with different number of corner points $K$ : (a) $K < L_m L_n / 2$ ; (b) $K = L_m L_n / 2$ ; (c) $K > L_m L_n / 2$ and (d) $K \gg (N - 1) L_m L_n / 2$ . As the complexity of the polygon increases, i.e. as $K$ increases, the bound $D_{distr}(R)$ approaches $D_{joint}(R)$ .	122
7.3	The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 7. The input sequence shown is shown in Figure 7.5 (a).	129
7.4	The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 7. The input sequence shown is shown in Figure 7.6 (a).	130

7.5	Illustration of the compression of the first sequence with a real object whose disparity is described by the affine transform: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 7 at 0.04 bpp with the PSNR of 333.96 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.04 bpp with the PSNR of 33.22 dB. . . . .	131
7.6	Illustration of the compression of the second sequence with a real object whose disparity is described by the affine transform: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 7 at 0.043 bpp with the PSNR of 30.1 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.045 bpp with the PSNR of 29.3 dB. . . . .	132
7.7	The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 8. The input sequence shown is shown in Figure 7.9 (a). . . . .	135
7.8	The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 8. The input sequence shown is shown in Figure 7.10 (a). . . . .	136
7.9	The first illustration of the compression of an array of images whose disparity is described by a translation vector. (a) the original images; (b) the reconstructed images using the proposed distributed semi-parametric compression scheme at 0.07 bpp with the PSNR of 31.9 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.066 bpp with the PSNR of 28.35 dB. . . . .	137
7.10	The second illustration of the compression of an array of images whose disparity is described by a translation vector. (a) the original images; (b) the reconstructed images using the proposed distributed semi-parametric compression scheme at 0.34 bpp with the PSNR of 28.1 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.31 bpp with the PSNR of 26.5 dB. . . . .	138



7.11	The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 9. The input sequence is shown in Figure 7.13 (a). . . . .	141
7.12	Illustration of the segmentation results of the key frame and the corresponding low-pass coefficients: (a) the extracted background; (b) the segmentation of the key frame; (c) the segmentation of the low-pass coefficients in the second frame. . . . .	142
7.13	Illustration of the compression of the highway sequence: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 9 at 0.2 bpp with the PSNR of 32.3 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.18 bpp with the PSNR of 29.9 dB. . . . .	143

# Abbreviations and Notations

## Abbreviations

bpp	Bits per pixel
CRB	Cramér-Rao Bound
DCT	Discrete Cosine Transform
DSC	Distributed Source Coding
DWT	Discrete Wavelet Transform
FRI	Finite Rate of Innovation
GOP	Group Of Pictures
KLT	Karhunen-Loève Transform
LSB	Least Significant Bit
MSB	Most Significant Bit
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
SNR	Signal-to-Noise Ratio
SPIHT	Set Partitioning In Hierarchical Trees

## Notations

$*$	Convolution operator
$\otimes$	Tensor product operator
$E[\cdot]$	Expectation operator
$\alpha$	Degree of smoothness of a function with Lipschitz regularity
$\mathbf{L}_2(\mathbb{R})$	Space of all square-integrable functions
$f(t)$	Continuous 1-D function
$f_i(t)$	Continuous 1-D function observed by Encoder $i$
$\hat{f}_i(t)$	Reconstructed version of $f_i(t)$
$\tilde{f}_i(t)$	Predicted version of $f_i(t)$ given by $\hat{f}_1(t - \hat{\tau})$
$f^{(R)}(t)$	$R$ -th derivative of $f(t)$
$f_p(t)$	Piecewise polynomial function
$\hat{f}_p(t)$	Reconstructed version of $f_p(t)$ obtained from parametric estimation
$f_\alpha(t)$	Uniformly Lipschitz $\alpha$ smooth function
$f_N(t)$	$N$ -term linear approximation of $f(t)$
$f_{\mathcal{N}}(t)$	$N$ -term nonlinear approximation of $f(t)$
$f(x, y)$	Continuous 2-D function
$f_i(x, y)$	Continuous 2-D function observed by Encoder $i$
$s(t)$	Continuous step function
$\psi$	Wavelet function
$\psi_{j,n}(t)$	Shifted and dilated version of $\psi(t)$ at resolution $2^j$
$\psi_{m,n}^j(x, y)$	Shifted and dilated version of $\psi(x, y)$ at resolution $2^j$

$\tilde{\psi}$	Dual function of $\psi$
$\varphi$	Scaling function or sampling kernel
$\Phi$	Fourier transform of $\varphi$
$\varphi_{J,n}(t)$	Shifted and dilated version of $\varphi(t)$ at resolution $2^J$
$\varphi_{m,n}^J(x, y)$	Shifted and dilated version of $\varphi(x, y)$ at resolution $2^J$
$\tilde{\varphi}$	Dual function of $\varphi$
$\beta_P$	B-spline function of order $P$
$B_P$	Fourier transform of $\beta_P$
$d_{j,n}$	1-D wavelet (or high-pass) coefficients at resolution $2^j$
$\bar{d}_{j,n}$	Quantized version of $d_{j,n}$
$\hat{d}_{j,n}$	Predicted version of $d_{j,n}$ where $\hat{d}_{j,n} = \langle \hat{f}_p(t), \psi_{j,n}(t) \rangle$
$\tilde{d}_{j,n}$	Decoded version of $d_{j,n}$ that is obtained using error correction code
$d_{i,j,n}$	1-D wavelet (or high-pass) coefficients of $f_i(t)$ at resolution $2^j$
$\tilde{d}_{i,j,n}$	Predicted version of $d_{i,j,n}$ given by $\langle \tilde{f}_i(t), \psi_{j,n}(t) \rangle$
$\hat{d}_{i,j,n}$	Decoded version of $d_{i,j,n}$ that is obtained using error correction code
$d_{m,n}^j$	2-D wavelet (or high-pass) coefficients at resolution $2^j$
$c_{J,n}$	1-D scaling (or low-pass) coefficients at resolution $2^J$
$\bar{c}_{J,n}$	Quantized version of $c_{J,n}$
$c_{i,J,n}$	1-D scaling (or low-pass) coefficients of $f_i(t)$ at resolution $2^J$
$\bar{c}_{i,J,n}$	Quantized version of $c_{i,J,n}$
$c_{m,n}^J$	2-D scaling (or low-pass) coefficients at resolution $2^J$
$\bar{c}_{m,n}^J$	Quantized version of $c_{m,n}^J$

$\epsilon_{m,n}^q$	Quantization noise of $\bar{c}_{m,n}^J$
$\bar{y}_n$	Quantized version of $y_n$
$y_n$	Samples of $f(t)$
$z_n^{(R)}$	$R$ -th order finite difference of $y_n$
$\bar{z}_n^{(R)}$	$R$ -th order finite difference of $\bar{y}_n$
$\epsilon_n$	i.i.d. additive Gaussian noise
$\epsilon_n^q$	Quantization noise of $\bar{y}_n$
$\sigma_q^2$	Variance of $\epsilon_n^q$
$y_n^\alpha$	Low-pass coefficients (samples) of $f_\alpha(t)$
$\sigma_\alpha^2$	Variance of $y_n^\alpha$
$y_n^s$	Low-pass coefficients (samples) of $s(t)$
$\epsilon_n^s$	Noise term given by $y_n^\alpha + \epsilon_n^q$
$\sigma_\epsilon^2$	Variance of $\epsilon_n^s$ given by $\sigma_q^2 + \sigma_\alpha^2$
$\epsilon_t$	Error in estimation of location of step in step function $s(t)$
$\epsilon_A$	Error in estimation of amplitude of step in step function $s(t)$
$c_n^{(p)}$	Coefficients for 1-D $t^p$ polynomial reproduction
$S_{m,n}$	Samples of $f(x, y)$
$\mathcal{J}_p$	Set of indices of coefficients in cone of influence of discontinuities of $f_p(t)$
$c_{m,n}^{(p,q)}$	Coefficients for 2-D $x^p y^q$ polynomial reproduction
$M_p$	$p$ -th order continuous moment of $f(t)$
$M_p^{(R)}$	$p$ -th order continuous moment of $f^{(R)}(t)$
$M_p^{(R)}$	Estimation of $M_p^{(R)}$

$\widehat{M}_{p,q}^i$	Estimate of $M_{p,q}^i$ obtained given by $\sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^{(p,q)} \bar{c}_{m,n}^J$
$M_{p,q}^i$	$(p+q)$ -th order continuous moment of $f_i(x, y)$
$M_{p,q}$	$(p+q)$ -th order continuous moment of $f(x, y)$
$w_{p,q}^i$	Error in estimate of $M_{p,q}^i$
$\mu_{p,q}$	$(p+q)$ -th order central moment of $f(x, y)$
$\kappa_{p,q}$	$(p+q)$ -th order complex moment of $f(x, y)$
$\delta_{m,n}$	Kronecker's delta function
$\Sigma$	Covariance matrix
$T$	Sampling period
$\Delta$	Quantizer step size
$\tau$	Shift parameter
$\hat{\tau}$	Estimation of $\tau$
$\ell$	Length of each side of bi-level polygon
$\epsilon_{i_\alpha}$	$\alpha$ -Lipschitz smooth prediction error given by $f_i(t) - f_1(t - \tau_i)$
$\hat{\epsilon}_{i_\alpha}$	Reconstructed version of $\epsilon_{i_\alpha}$
$\beta$	Parameter that relates power of $\epsilon_{i_\alpha}$ to $f_1(t)$
$\beta^*$	Estimate of $\beta$
$\boldsymbol{\tau}$	Translation vector
$\hat{\boldsymbol{\tau}}$	Estimate of $\boldsymbol{\tau}$
$\epsilon_{\tau_x}$	Error in estimation of $x$ -component of $\boldsymbol{\tau}$
$\epsilon_{\tau_y}$	Error in estimation of $y$ -component of $\boldsymbol{\tau}$
$D$	Distortion

$R$	Bit rate
$D(R)$	Distortion-rate function

# CHAPTER 1

## Introduction

### 1.1 Background

Compression technologies have become a vital part in our modern-day multimedia and visual communication systems. Over the last two decades, a considerable amount of research has been carried out in an attempt to determine an efficient digital representation of images and videos [26, 51, 58, 24]. The vast number of applications includes digital broadcast, internet video streaming, media storage and many more. A number of international standards for image and video coding have been established since the 1990s; these include, for example, the JPEG [69] and JPEG-2000 [61] (by the Joint Photographic Experts Group), the MPEG-1/2/4 [50, 58] (by the Moving Picture Experts Group) and the H.261/3/4 [36] (by the International Telecommunication Union).

The basic problem of compression or source coding is to convey the source data with the highest possible fidelity within an available bit budget where the fundamental trade-off is made between the bit rate and the fidelity. The *efficiency* or the *distortion-rate performance* of a coding system is the measure of how well this trade-off can be made [13]. In many practical applications, however, the issues of complexity must also be taken into account. One can, therefore, state that the design objective of a compression system is to achieve an optimal distortion-rate performance given the maximum allowed complexity. Therefore, in the context of compression, optimality



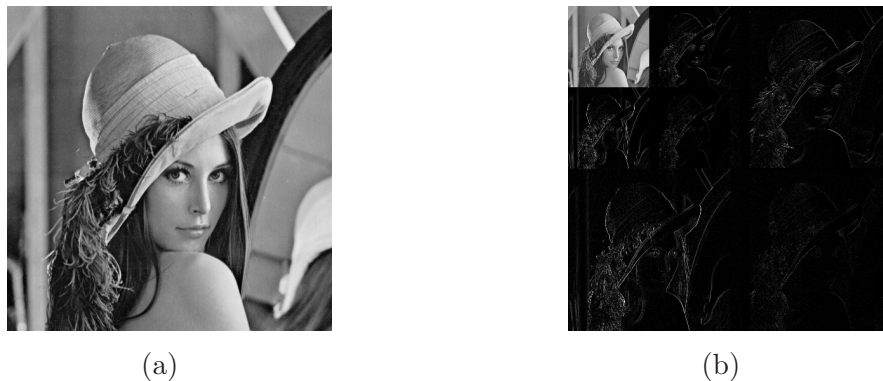
means that the achieved distortion, which is usually measured as the mean-squared error, is minimum for a given rate.

Over the last two decades, transform coding has emerged as the dominating strategy for compression due to its efficiency and simplicity [26]. In order to obtain a good compression result, one needs to reduce the spatial and temporal redundancies contained in the observed signal. The use of the transform allows the encoder to achieve this task efficiently. Even though the transform is linear, the process of selecting which transform coefficients to encode is usually highly adaptive and nonlinear in nature. In addition, for video compression, the increase in computational capability of computers has made possible the inclusion of sophisticated joint encoding techniques such as motion compensated prediction (MCP), which further reduces the temporal redundancy of the signal. Thus, the encoding process of today's conventional compression schemes is highly complex while the decoding process is much less computationally intensive.

In recent years, the growth in the area of sensor network and uplink-rich media applications based on mobile devices has given rise to a new paradigm in compression known as distributed source coding [41, 24]. An example of such applications is the capturing of a scene with an array of cameras. In contrast to the centralized setting, the source is partially observed by a number of independent encoders, which are not able to communicate to one another. The observations, however, can be jointly decoded. Moreover, in some cases, there is a limit on the computational power of the acquisition device. Therefore, such scenarios impose a new set of requirements for compression, which are low-complexity encoder, robustness and high compression efficiency. While important results and optimality conditions have been provided for the classical centralized case, this new paradigm present new challenges in compression technology and many questions remain largely open. Here, optimality refers to the minimization of the total distortion for a given total rate used by every independent encoder.

## 1.2 Problem Statement and Motivations

The core structure of transform coding consists of three elements, namely, the linear transform, quantization and lossless entropy code. Given an observed source vector  $f \in$



**Figure 1.1:** Illustration of the wavelet transform of an image. (a) original image; (b) the corresponding Haar wavelet transform with two decomposition levels.

$\mathbf{L}_2(\mathbb{R})$ , the linear transform exploits the redundancy contained within  $f$  by decomposing it over the basis  $\mathcal{B} = \{g_m\}$  of  $\mathbf{L}_2(\mathbb{R})$ . A quantizer then maps the value of the transform coefficients to some discrete set  $\mathcal{J}$  and the entropy code performs a reversible mapping of  $\mathcal{J}$  to a bit stream. This process is then simply reversed at the decoder. One can, therefore, state that the design objective of any transform code is to optimize each stage of the encoding and decoding processes, given the knowledge of the source and the complexity constraint, such that the distortion of the reconstructed signal is minimized for a given rate.

One of the most important transforms in image compression is the wavelet transform, which is used in state-of-the-art image compression standards such as the JPEG-2000 [61]. Figure 1.1 shows an example of the wavelet transform of an image. The theoretical study of the performance of wavelets in compression is usually based on a class of piecewise smooth functions, which is used as a simplified model of a row (or column) of an image [11]. For a conventional centralized transform coding setting, it has been shown that a wavelet-based compression strategy that employs a nonlinear encoder and a linear decoder produces the best distortion-rate performance. The problem of finding the best strategy for a wavelet-based distributed source coding, however, remains largely open.

For the purpose of investigating the impact of the structural change in complexity, it is natural to first ask the following question: given the same observation of the source, can we still achieve the same distortion-rate performance by using an encoder

that is linear and, instead, a decoder that is nonlinear? This scenario represents the dual situation of the traditional one. In order to obtain a distortion-rate analysis that is comparable to the centralized case (see [11]), we consider the same piecewise smooth model of the signal. One possibility is to include a form of nonlinear parametric estimation technique in the decoding process to allow the decoder to partially predict the structure of the signal. This study gives us new insights into the problem and represents a new approach to wavelet-based compression, which can also be applied to the distributed coding scenario.

The second problem that follows is how to apply the concept of nonlinear decoding in the distributed source coding setting. In order to solve the problem, one needs to first find an appropriate model that describes the difference between each observed signal. For images and videos, one of the possible candidates is the geometric transformation. This is analogous to the use of MCP algorithm in video compression, where blocks of previously decoded frames are shifted to form the prediction of the current frame. More precisely, given an observed 2-D signal  $f_i$  at the  $i$ -th encoder, the function  $f_i$  can be predicted from the reference observation, say  $f_1$ , as

$$f_i(x, y) = f_1(T_i(x, y)) + \epsilon_i(x, y), \quad i \neq 1, \quad x, y \in \mathbb{R},$$

where  $\epsilon_i(x, y)$  is the prediction error. The transformation matrix  $T_i$  can represent, for example, a simple translation, shearing, rotation and re-scaling. This model also fits in well with the concept of joint decoding with parametric estimation as  $T_i$  can be estimated at the decoder. Since each encoder does not have access to the reference observation  $f_1$ , the next challenge is to find an appropriate quantization strategy to encode the prediction error.

In order to gain a deeper understanding of the wavelet-based distributed compression problem, it is logical to start the analysis from the simplified signal model, which is the 1-D piecewise smooth function. Lastly, we apply the new theoretical framework in the context of practical distributed images and video compression. This also includes the problem of finding an appropriate constructive parametric estimation algorithm that

can be employed by the decoder.

### 1.3 Previous Related Work

The problem of the classical centralized transform coding has been well studied in the literature over the last two decades [26]. The results obtained can be divided into two categories based on stationarity of the source. For stationary sources, some of the most well-known results include the optimality of the Karhunen-Loève transform (KLT) for Gaussian sources [30].

For non-stationary piecewise smooth signals, recent research has demonstrated that the wavelet transform is the best transform and that the best compression strategy is based around nonlinear approximation strategies [66, 11, 34]. A number of important wavelet-based image compression algorithms have been proposed since the 1990s [48, 43, 60, 61]. Their new insights have established the potential of wavelets in compression. Many articles and books are available for an in-depth review of the wavelet theory [67, 56, 66, 65, 34].

Following the developments of wavelet theory, which focuses on the projection of a signal onto an approximation subspace under the multiresolution framework, a new sampling theory has emerged [68, 14, 49]. The new sampling theory allows a perfect reconstruction of a class of signals called signals with Finite Rate of Innovation (FRI), which also includes some non-bandlimited signals. One of the main characteristics of FRI signals is that they can be completely described by a parametric representation, which has a finite degree of freedom. This development has given us new insights into the connection between the scaling coefficients of the wavelet transform and the parametric representation of the signal. This sampling theory has also been applied in the field of image registration, which is also based around parametric estimation [7].

Recently, the study of distributed KLT of stationary Gaussian sources has been proposed in [20]. The authors demonstrated that the KLT is still the optimal transform, however, the structure of the transform needs to be modified in order to achieve optimality. With the inspiration from the theoretical results obtained by Slepian and Wolf [52] and by Wyner and Ziv [71], many practical distributed compression schemes for non-

stationary sources such as images and videos have also been proposed [41, 2, 24, 28, 15]. These practical designs modify the way transform coefficients are quantized by applying the concept of advanced channel codes [1, 44, 40, 19, 53]. Practical schemes that use the wavelet transform together with channel coding techniques are presented in [9, 18].

## 1.4 Thesis Outline

The overall research described in this thesis starts with a set of theoretical results of the wavelet-based compression of 1-D piecewise smooth functions under a new set of requirement imposed by the distributed source coding problem. We then investigate the potential applications of our findings in the context of distributed image and video compression.

Chapter 2 reviews the theoretical foundations of centralized and distributed transform coding. Firstly, the well-known results and optimality conditions of each element in the centralized transform coding structure are presented. This is then followed by an overview of the theoretical results from Slepian and Wolf for lossless distributed coding, and Wyner and Ziv for lossy distributed coding. We then review the recent developments in practical distributed transform coding schemes, which includes the distributed KLT for Gaussian sources as well as other practical distributed compression schemes for images and videos.

The review of the wavelet theory is then given in Chapter 3. In particular, we focus on the linear and nonlinear approximation results of piecewise smooth functions and their relation to the compression performance. We also give an overview of the practical wavelet-based image and video compression algorithms.

Chapter 4 presents a review of the sampling theory of FRI signals. In particular, we focus on the polynomial reproduction properties of the sampling kernels. Three sampling schemes that allow perfect reconstruction of non-bandlimited signals by means of parametric estimation are then presented. The extension of the theory to 2-D cases is also included.

In Chapter 5, we present a novel centralized wavelet-based compression scheme that employs a linear encoder and a nonlinear decoder. We start by presenting our model

of the piecewise smooth function. The new semi-parametric method for compression is then proposed. We show, using the distortion-rate analysis, that by including parametric estimation in the decoding process, the distortion-rate performance of the proposed scheme can be comparable to that of a conventional scheme with a nonlinear encoder. This finding is confirmed with simulation results. In this chapter, we also present a practical parametric estimation based on the results from sampling theory of FRI signals.

Chapter 6 extends the results of the centralized scenario to the case where a piecewise smooth function is observed by a number of multiple independent encoders. We start by presenting the model that describes the disparity between each observed signal. A distributed semi-parametric compression scheme is then proposed, where the decoder employs a parametric estimation algorithm in order to perform joint decoding. The distortion-rate analysis of the proposed scheme is then given. In particular, we show that the distortion-rate function of the proposed scheme can be comparable to that of a joint encoding scheme.

The work in Chapter 7 focuses on the application of the proposed theoretical framework in the context of distributed image and video compression. In order to gain a deeper understanding, we first look at a case study where we develop a distributed compression algorithm for a simple synthetic video sequence. We then extend the model to include a real object whose motion can be described by an affine transform. Here, we present a compression scheme whose decoder can estimate the affine transform parameter by using the results of the sampling theory of FRI signals. Finally, we propose two practical distributed compression schemes for a set of images obtained from an array of cameras and a real video sequence with a fixed background.

Finally, the thesis is concluded in Chapter 8, where we also presents some ideas and remarks for future work.

## 1.5 Original Contribution

The main contribution of this thesis is the development of the new semi-parametric approach to the wavelet-based centralized and distributed compression. The original

research work presented in Chapter 5, 6 and 7 of this thesis has led to the following publications:

- V. Chaisinthop and P.L. Dragotti, “Distributed video coding based on sampling of signals with finite rate of innovation,” *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing, Wavelets XII*, vol. 6701, San Diego, CA, USA, August 2007.
- V. Chaisinthop and P.L. Dragotti. “A new approach to distributed video coding using sampling of signals with finite rate of innovation,” *Proceedings of Picture Coding Symposium (PCS)*, Lisbon, Portugal, November 2007.
- V. Chaisinthop and P.L. Dragotti. “Distributed transform coding”, in *Distributed Source Coding: Theory, Algorithms, and Applications*, Academic Press, January 2009.
- V. Chaisinthop and P.L. Dragotti. “Semi-parametric compression of piecewise smooth functions,” *Proceedings of European Conference on Signal Processing (EU-SIPCO)*, Glasgow, UK, August 2009.
- V. Chaisinthop and P.L. Dragotti. “Centralized and distributed semi-parametric compression of piecewise smooth functions,” *IEEE Transactions on Signal Processing*, to be submitted.

# CHAPTER 2

## Theoretical Foundations of Centralized and Distributed Transform Coding

### 2.1 Introduction

Compression or source coding is undeniably one of the most important problems in modern signal processing and communications. Due to its high efficiency and low complexity, transform coding has emerged as the dominating compression strategy. Thus, it is not surprising that transform coders are present in most of today's compression standards. Over the last three decades, many important results and optimality conditions have been derived for a classical centralized scenario where the source can be observed by a single encoder (see [26] for an overview of this topic). For example, it is now well known that when the source is Gaussian, the Karhunen-Loève Transform (KLT) is the optimal transform [30, 26].

Recently, a new paradigm in compression called Distributed Source Coding (DSC) has emerged as a result of a fast growing number of sensor networks seen in today's applications. In contrast to the centralized scenario, the source is partially observed by independent encoders, which are required to perform compression locally. It is then natural to wonder how the classical centralized transform coding strategy is going to change under this new scenario. In order to provide a precise set of answers, one needs to first reconsider each module in the transform coding architecture; this includes the



transform itself, quantization and bit allocation strategies as well as the entropy code.

Recent research has provided us with some precise answers. For example, when the source is Gaussian, it has been shown that the KLT is still the best transform and it is optimal in some cases [20]. Other optimality conditions for transforms in high bit-rate regimes have also been proved [18, 42]. If, however, the Gaussian and high bit rate assumptions are relaxed then the problem of distributed transform coding remains largely open. Current designs of distributed transform coders for non-Gaussian sources are usually based on heuristics. Moreover, most of the modification occurs in the quantization and entropy coding stages but not in the local transform. For example, in distributed video coding, the structure of the discrete cosine transform (DCT) [4] remains unchanged but the transform coefficients are quantized differently in order to exploit the redundancies in the correlated information available at the decoder. While the strategy is effective and can be generalized, it is not necessarily optimal. For a more in-depth coverage of the recent development in DSC, we refer to [15].

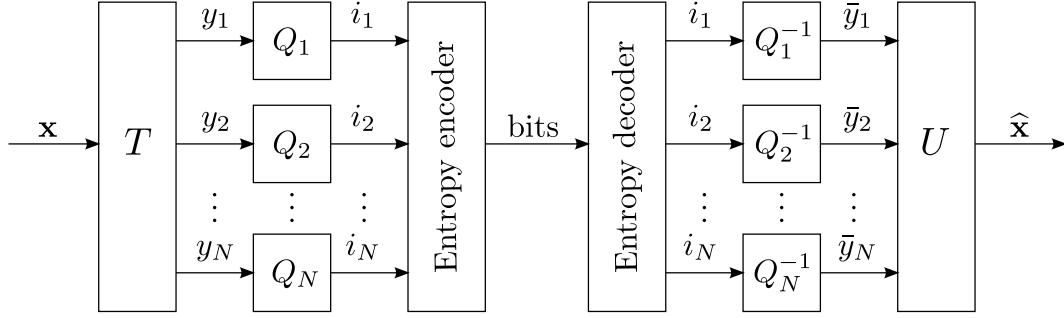
In this chapter, we review the theoretical foundations of centralized and distributed compression methods. We primarily focus on sources which are statistically stationary. The review is divided into two parts: in Section 2.2, we discuss the foundations of centralized transform coding, then, in Section 2.3, a brief overview of distributed transform coding is presented. A summary is then given in Section 2.4.

## 2.2 Foundations of Centralized Transform Coding

For the first part of our review, we look into each key element in a transform coding architecture. We refer to [26, 34] for an excellent overview of this topic.

### 2.2.1 Transform Coding Overview

Figure 2.1 shows a typical compression scheme based on transform coding structure. The goal of the encoder is to map the input  $x$  to a bitstream of finite length. First, the transform decomposes a signal in a basis and quantizes the transform coefficients. This is followed by a lossless entropy coder that maps the quantized coefficients to a bit stream. Such modularization of the encoding process allows ‘simple coding’, which



**Figure 2.1:** Transform coding structure. A typical compression scheme consists of three elements: linear transform, quantization and entropy coding.

means the use of a scalar quantizer and a scalar entropy encoder, to be used with high efficiency. Given a source vector  $\mathbf{x} \in \mathbb{R}^N$ , the simplicity of a transform coder enables  $x$  with a large value of  $N$  to be encoded. This is one of the main reasons that make transform codes the most widely used source codes.

The decomposition of  $\mathbf{x}$  over basis  $\mathcal{B} = \{\mathbf{g}_m\}_{0 \leq m < N}$  of  $\mathbb{R}^N$  can be written as follows:

$$\mathbf{x} = \sum_{m=0}^{N-1} y_m \mathbf{g}_m, \quad (2.1)$$

where the transform coefficients  $y_m$  are given by

$$y_m = \langle \mathbf{x}, \tilde{\mathbf{g}}_m \rangle = \sum_{n=0}^{N-1} x_n \tilde{g}_{m,n}.$$

Here,  $\langle \cdot, \cdot \rangle$  is the inner product operator and  $\tilde{\mathbf{g}}_m$  is the dual of  $\mathbf{g}_m$ . More precisely,  $\tilde{\mathbf{g}}_m$  is such that  $\langle \tilde{\mathbf{g}}_m, \mathbf{g}_n \rangle = \delta_{m,n}$  with  $\delta_{m,n}$  denoting the Kronecker's delta function. When the basis  $\mathcal{B}$  is orthogonal,  $\tilde{\mathbf{g}}_m = \mathbf{g}_m$ . The decomposition of  $\mathbf{x}$  in its transform coefficients  $\mathbf{y}$  can be written in matrix form as

$$\mathbf{y} = T\mathbf{x},$$

where the rows of the  $N \times N$  matrix  $T$  correspond to the dual-basis vectors  $\{\tilde{\mathbf{g}}_m\}_{1 \leq m \leq N}$ . The motivation behind the transform is to exploit the redundancy within  $\mathbf{x}$ . A quantizer then maps  $\mathbf{y} \in \mathbb{R}^N$  to some discrete set  $\mathcal{J}$  and, finally, the lossless entropy encoder performs a reversible mapping from  $\mathcal{J}$  to a bit stream. The decoder essentially reverses

the encoding process to obtain the approximation of the source given by

$$\hat{\mathbf{x}} = \sum_{m=0}^{N-1} \bar{y}_m \mathbf{g}_m,$$

where  $\bar{y}_m$  denotes the quantized coefficients. In matrix form, this can be written as

$$\hat{\mathbf{x}} = U\bar{\mathbf{y}},$$

where the columns of  $U$  correspond to the basis vectors  $\{\mathbf{g}_m\}_{1 \leq m \leq N}$ . In the orthogonal case, we have that  $U = T^T$ .

Quantization is a lossy process, which introduces error into the compression scheme. The quality of a lossy encoder is normally measured by the mean-squared error (MSE) distortion given by

$$D = \mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = \mathbb{E} \left[ \sum_{n=0}^{N-1} (x_n - \hat{x}_n)^2 \right],$$

where  $\mathbb{E}[\cdot]$  is the expectation operator. In order to gauge the performance of the compression scheme, the distortion  $D$  is measured against the rate  $R$ , which is the expected number of bits produced by the encoder divided by the length  $N$  of  $\mathbf{x}$ . This is referred to as the rate-distortion performance. One transform code is said to be better than the other in a rate-distortion sense if, at a given  $R$ , the former can achieve a lower distortion.

### 2.2.2 Entropy code

Entropy coding is a form of reversible lossless compression, which can only be applied to discrete sources. Consider a random source  $X$  that produces a finite set of  $K$  values:  $\mathcal{A} = \{x_k\}_{1 \leq k \leq K}$ . Let  $p_k = \Pr\{X = x_k\}$  denote the probability of occurrence of  $x_k$ . A unique codeword or a binary representation  $b(x_k)$  is assigned to each value by the entropy encoder. The goal is to minimize the expected length of the binary representation of  $X$ :

$$\mathbb{E}[l(X)] = \sum_{k=1}^K p_k l_k,$$

where  $l_k$  is the length of  $b(x_k)$ .

In order to preserve the invertibility of entropy coding, it is required that no codeword can be a prefix of another. This is so that the entire sequence can be decoded without any punctuation to tell the decoder where each codeword begins and ends. An entropy code is said to be optimal if it is a prefix code that minimizes  $E[l(X)]$ . Huffman codes and arithmetic codes (see [31, 13]) are examples of optimal entropy codes. The lower bound of  $E[l(X)]$  of a prefix code is given by the Shannon entropy:

**Theorem 1** [47]: *Let  $X$  be a random source with symbols  $\{x_k\}_{1 \leq k \leq K}$  with probability  $\{p_k\}_{1 \leq k \leq K}$  where  $p_k = \Pr\{X = x_k\}$ . The expected length of a prefix code satisfies*

$$E[l(X)] \geq \mathcal{H}(X) = - \sum_{k=1}^K (p_k) \log_2(p_k),$$

where  $\mathcal{H}(X)$  is called the entropy of  $X$ . In addition, there exists an optimal entropy code such that

$$\mathcal{H}(X) \leq E[l(X)] \leq \mathcal{H}(X) + 1. \quad (2.2)$$

### 2.2.3 Scalar Quantization

Quantization is a lossy process that maps the continuous values in  $\mathbb{R}^N$  to a finite set of alphabet or a reproduction codebook  $\mathcal{C} = \{\bar{x}_k\}_{k \in \mathcal{J}} \subset \mathbb{R}^N$ , where  $\mathcal{J}$  is a finite set of indices. Usually, each component of the source  $\mathbf{x}$  is quantized individually ( $N = 1$ ) and the quantizer is called *scalar quantizer*. A more sophisticated form of quantization that operates on a group of components ( $N > 1$ ) is called vector quantization. For a detailed treatment of vector quantization, we refer to [22].

Assuming  $\mathbf{x}$  takes arbitrary real values in  $[a, b]$ , a scalar quantizer divides  $[a, b]$  into  $K$  intervals  $\{[w_{k-1}, w_k]\}_{1 \leq k \leq K}$  with  $w_0 = a$  and  $w_K = b$ . The width of each interval can be variable or fixed. We denote with  $\bar{x} = Q(x)$ , the approximation of  $x$  by a scalar quantizer  $Q$ , where  $Q(x) = x_k, \forall x \in ]w_{k-1}, w_k]$ . The width of each interval  $]w_{k-1}, w_k]$  is referred to as the *step size* denoted by  $\Delta_k$ . Given that the quantizer outputs  $K$  different values, we need  $R = \lceil \log_2(K) \rceil$  bits to represent each value with a fixed length binary code. The quantizer is then said to have rate  $R$ .

Most of today's compression schemes employ a simple uniform scalar quantizer, where the step sizes are constant i.e.  $\Delta_k = \Delta$ . The output  $\bar{x}$  is usually derived from the nearest integer multiple of  $\Delta$  that is closest to  $x$  i.e.  $\hat{x} = [x/\Delta]$ . A common alternative is to shift both the intervals and the output by half a step size.

By adding entropy codes, one can convert a fixed-rate quantizer into a variable-rate quantizer where  $R$  is given by the expected code length. Furthermore, from (2.2), we know that the performance of an optimal entropy code is bounded by the entropy. Thus, the use of a variable-rate quantizer with an optimal entropy code is also referred to as entropy-constrained quantization.

### High-resolution quantization

Consider a random source  $X$  with a probability density  $p(x)$ , the mean-square quantization error is given by

$$D = \mathbb{E} [(X - \bar{X})^2] = \int_{-\infty}^{+\infty} (x - Q(x))^2 p(x) dx.$$

A high-resolution quantizer is one where  $p(x)$  is approximately constant in each quantization bin. This is true if  $\Delta_k$  is sufficiently small relative to the rate of change of  $p(x)$ . Let us denote with  $p_k$  the probability  $\Pr\{X \in [w_{k-1}, w_k]\}$ . It then follows that  $p(x) \simeq p_k/\Delta_k$ ,  $\forall x \in [w_{k-1}, w_k]$ . The following result can be obtained under the high-resolution hypothesis:

**Theorem 2** [34]: *For a high-resolution quantizer, the distortion  $D$  measured as the MSE is minimized by setting  $x_k = (w_k + w_{k+1})/2$ , which yields*

$$D = \frac{1}{12} \sum_{k=1}^K p_k \Delta_k^2.$$

The distortion of a high-resolution uniform quantizer is, therefore, given by  $D = \frac{\Delta^2}{12}$ .

### 2.2.4 Distortion rate of scalar quantization

While the optimal design of a quantizer (see [17, 27, 26]) is beyond the scope of this thesis, we can say that an optimal quantizer is the one that minimizes the distortion  $D$  for a given rate  $R$  (or minimizes  $R$  for a given  $D$ ). The fundamental result in rate-distortion theory provides us with the lower bound on the achievable  $R$  for a given  $D$ , which is the information rate distortion function  $R(D)$ .

**Theorem 3** [13]: *The information rate distortion function  $R(D)$  for a discrete source  $\mathbf{x}$  with distortion measure  $d(x, \hat{x})$  is defined as*

$$R(D) = \min_{p(\hat{x}|x): \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(\mathbf{x}, \hat{\mathbf{x}}), \quad (2.3)$$

where  $I(\mathbf{x}, \hat{\mathbf{x}}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\hat{\mathbf{x}})$  is the mutual information and the minimization is over all conditional distribution  $p(\hat{x}|x) = p(x)p(\hat{x}|x)$  that satisfy the expected distortion constraint.

The bound given in above theorem is also known as the Shannon rate-distortion bound, which is monotonically decreasing convex function.

Unfortunately, in practice, the function  $R(D)$  is known only for a few cases. One remarkable example is given by the Gaussian source. Consider a source that produces independent and identically distributed (i.i.d.) Gaussian variables with variance  $\sigma^2$ . Its Shannon distortion-rate bound subject to the MSE is given by [13, 27]

$$D(R) = \sigma^2 2^{-2R}. \quad (2.4)$$

Let us now investigate how close the performance of a scalar quantizer is to the bound in (2.4). Using high-resolution analysis (i.e. assuming large  $R$ ) for a fixed-rate quantizer, the optimal quantizer for a Gaussian source is non-uniform with [26]

$$D(R) = \frac{\sqrt{3}\pi}{2} \sigma^2 2^{-2R}. \quad (2.5)$$

In comparison to the bound in Equation (2.4), the distortion is higher by  $\sim 4.35\text{dB}$ , equivalent to a rate loss of  $\sim 0.72$  bits per symbol.

Interestingly, high-resolution analysis of an entropy-constrained quantization shows that the optimal quantizer in this case is uniform [25]. The corresponding distortion-rate bound  $D(R)$  is

$$D(R) \approx \frac{1}{12} 2^{2\mathcal{H}_d(X)} 2^{-2R}, \quad (2.6)$$

where

$$\mathcal{H}_d(X) = - \int_{-\infty}^{+\infty} f_X(x) \log_2 f_X(x) dx$$

is the differential entropy of the source. For a Gaussian random variable, it follows that

$$D(R) = \frac{\pi e}{6} \sigma^2 2^{-2R}.$$

The distortion is now  $\sim 1.53\text{dB}$  higher than the bound and the redundancy is  $\sim 0.255$  bits per symbol, which is a significant improvement when compared to Equation (2.5).

In summary, at high-bit rates, one can conclude that the best quantization strategy is to use a uniform quantizer followed by an entropy encoder, which would result in a fairly simple lossy compression scheme whose performance is given in (2.6). In the case of Gaussian source, the achievable  $D(R)$  function is very close to that of the best possible performance bound.

### 2.2.5 Bit Allocation

In a typical transform coding structure, each transform coefficient is separately scalar quantized. Hence, the total number of bits (or the bit budget) has to be split among the coefficients in some way. Bit allocation problem refers to the question of how the bits should be allocated. Consider a set of quantizers whose  $D(R)$  functions are as follows (from Equation (2.6)):

$$D_m = c_m \sigma_m^2 2^{-2R_m} \quad \text{with} \quad R_m \in \mathcal{R}_m \quad \text{and} \quad m = 1, 2, \dots, N, \quad (2.7)$$

where  $c_m$  is a constant,  $\sigma_m \in \mathbb{R}$  and  $\mathcal{R}_m = [0, \infty[$  is a set of available rates. The aim of bit allocation is to minimize the average distortion  $D = \frac{1}{N} \sum_{m=1}^N D_m$  for a given rate  $R = \frac{1}{N} \sum_{m=1}^N R_m$ . This is a constrained optimization problem, which can be solved

using Lagrange multipliers i.e. by minimizing

$$\mathcal{L}(R, D) = D + \lambda R = \sum_{m=1}^N (D_m + \lambda R_m). \quad (2.8)$$

We refer to [38, 57, 26] for detailed treatments on bit-allocation in image and video compression.

Intuitively, the initial bit allocation is not optimal if the average distortion can be reduced by taking bits away from one coefficient and giving them to another. Therefore, from (2.7), one necessary condition for an optimal bit allocation requires

$$\frac{\partial D}{\partial R_i} = \frac{\partial D}{\partial R_j}, \quad i, j = 1, 2, \dots, N. \quad (2.9)$$

Indeed, solving (2.8) leads to the condition in (2.9). By applying (2.9) to (2.7) and ignoring the fact that all the rates must be nonnegative, the optimal bit allocation is given by [26]

$$R_m = R + \frac{1}{2} \log_2 \left( \frac{c_m}{\left( \prod_{m=1}^N c_m \right)^{1/N}} \right) + \frac{1}{2} \log_2 \left( \frac{\sigma_m^2}{\left( \prod_{m=1}^N \sigma_m^2 \right)^{1/N}} \right).$$

With the optimal bit allocation, all the distortion are equal such that  $D_m = D$ ,  $m = 1, 2, \dots, N$  and the resulting  $D(R)$  is [26]

$$D = \left( \prod_{m=1}^N c_m \right)^{1/N} \left( \prod_{m=1}^N \sigma_m^2 \right)^{1/N} 2^{-2R}. \quad (2.10)$$

Clearly, each  $R_m$  must be nonnegative for the above solution to be valid. At lower rates, the Kuhn-Tucker conditions give the components with smallest  $c_m \cdot \sigma_m^2$  no bits and the remaining components are given correspondingly higher allocations. For a uniform quantizer, the bit allocation determines the step size  $\Delta_m$  for each component. The equal-distortion property also implies that optimality can be achieved when all the step sizes are equal.



### 2.2.6 Optimal Transform for Gaussian Sources

Let us now consider the problem of compressing sources with block memory. Given a source  $\mathbf{x} \in \mathbb{R}^N$  that consists of statistically dependent samples, it would be inefficient to scalar quantize each element independently since we would not exploit the dependency of the samples. It is for this reason that a transform coder decomposes the signal over a basis  $\mathcal{B}$  of  $\mathbb{R}^N$  prior to quantization. In order to achieve the best possible decorrelating performance, the choice of the basis or the transform needs to be optimal for a given source.

Let  $\mathbf{x}$  be a jointly Gaussian zero-mean source with covariance matrix  $\Sigma_x = \mathbb{E}[\mathbf{x}\mathbf{x}^T]$  and assume that the transform  $T$  is orthogonal. Similarly, we denote the covariance matrix of the transform coefficients  $\mathbf{y}$  with  $\Sigma_y$ . Orthogonality means the Euclidean lengths are preserved, which gives  $D = \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = \mathbb{E}[\|\mathbf{y} - \bar{\mathbf{y}}\|^2]$ . A KLT is an orthogonal transform that diagonalizes  $\Sigma_x$  as follows:

$$\Sigma_y = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = T\Sigma_x T^T = \text{diag}(\lambda_0^2, \dots, \lambda_{N-1}^2).$$

Hence, the resulting transform coefficients are uncorrelated. Furthermore, the Gaussian assumption implies that the coefficients  $\{y_m\}_{0 \leq m < N}$  are independent Gaussian variables with variances  $\{\lambda_m^2\}_{0 \leq m < N}$ . We can, therefore, compress each coefficient independently. Under these assumptions, with any rate allocation, one can show that the KLT is an optimal transform [26].

From (2.6), we have that the  $m$ -th component of  $\mathbf{y}$  contributes a distortion

$$D_m(R_m) = c\lambda_m^2 2^{-2R_m}, \quad (2.11)$$

where  $R_m$  is the rate allocated to  $y_m$  and  $c$  is a constant whose values depends on the type of quantizer used. The overall  $D(R)$  is then given by

$$D(R) = \mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = \mathbb{E}[\|\mathbf{y} - \bar{\mathbf{y}}\|^2] = \frac{1}{N} \sum_{m=1}^N D_m.$$

Our aim is then to minimize  $D(R)$  subject to  $R = \frac{1}{N} \sum_{m=1}^N R_m$ . With optimal bit

allocation, it follows directly from (2.10) that the distortion simplifies to

$$D(R) = \mathbb{E} \left[ \|\mathbf{y} - \bar{\mathbf{y}}\|^2 \right] = c \left( \prod_{m=1}^N \lambda_m^2 \right)^{1/N} 2^{-2R}. \quad (2.12)$$

Hence, the optimal transform is the one that minimizes the geometric mean of  $\lambda_m^2$  given by  $\left( \prod_{m=1}^N \lambda_m^2 \right)^{1/N}$  and it is well known that the KLT minimizes this geometric mean.

**Theorem 4** [26, 34]: *Given a transform coder whose quantization error of each transform coefficient is described by Equation (2.11), there is a Karhunen-Loève basis that minimizes the geometric mean of  $\lambda_m^2$ .*

By applying Hadamard's inequality to  $\Sigma_y$ , we have that

$$\prod_{m=1}^N \lambda_m^2 \geq \det(\Sigma_y) = \det(T) \det(\Sigma_x) \det(T^T).$$

Since  $\det(T) = 1$ , the right-hand side of this inequality is invariant to the choice of  $T$ . Equality is achieved when the KLT is used. Hence, the KLT minimizes the distortion.

In the case where the source is stationary but non-Gaussian, the KLT still decorrelates the components of  $\mathbf{x}$  but does not provide independent components. The same bit allocation under high-resolution analysis can be applied as shown earlier. However, while the approach tends to give good results, we are not guaranteed that the performance is optimal.

## 2.3 Distributed Transform Coding

This section provides an overview of the recent development in distributed source coding (DSC). First, the foundations of DSC based on Slepian-Wolf [52] and Wyner-Ziv [71] theorems are studied. We then give the key results of the distributed KLT for Gaussian sources [20], which emphasizes the structural changes in the KLT under the distributed scenario. Lastly, we briefly review the Wyner-Ziv based transform coding schemes for non-Gaussian sources.

### 2.3.1 Foundations of Distributed Coding

The foundation of distributed coding dates back to the information theoretic results of Slepian and Wolf [52] and Wyner and Ziv [71] in the 1970s. First, let us consider a case where we have two discrete correlated sources  $X$  and  $Y$  with a joint probability distribution given by  $p_{X,Y}(x, y)$ . In the case of joint encoding, where both sources are available at the encoder, the minimum rate required to losslessly encode  $X$  and  $Y$  is given by the joint entropy. That is  $R = R_X + R_Y \geq \mathcal{H}(X, Y)$ , where

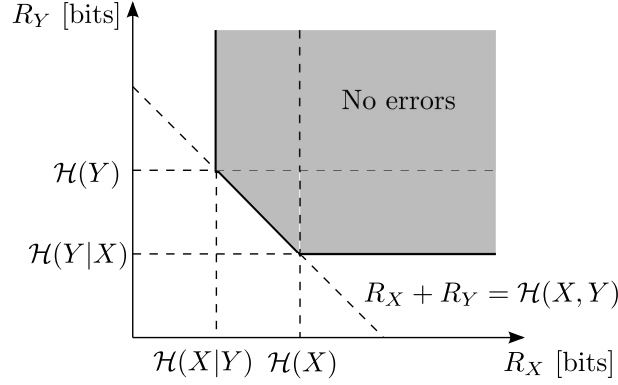
$$\mathcal{H}(X, Y) = - \sum_{n=1}^N \sum_{m=1}^M p_{X,Y}(x_n, y_m) \log_2 p_{X,Y}(x_n, y_m).$$

In the distributed source coding scenario, the two sources are separate and are encoded independently by two separate encoders but jointly decoded. Slepian and Wolf showed that one can achieve noiseless encoding with the following rates [52]:

$$\begin{aligned} R_X &\geq \mathcal{H}(X|Y), \\ R_Y &\geq \mathcal{H}(Y|X), \\ R_X + R_Y &\geq \mathcal{H}(X, Y), \end{aligned}$$

where  $\mathcal{H}(X|Y) = \mathcal{H}(X, Y) - \mathcal{H}(Y)$  is the conditional entropy of  $X$  given  $Y$ . This surprising result tells us that we can still achieve lossless encoding of  $X$  and  $Y$  at the same total rates as joint encoding even though the encoders are separated. Figure 2.2 shows the plot of the achievable rate region derived by Slepian and Wolf [52].

The counterpart to the Slepian and Wolf's theorem is the work of Wyner and Ziv on lossy source coding with side information [71]. This is a special case of Slepian-Wolf coding where the rate point is at  $(R_X, R_Y) = (\mathcal{H}(X|Y), \mathcal{H}(Y))$  i.e. the top corner of the graph in Figure 2.2. In [71], the Wyner-Ziv rate-distortion function is given for a problem of encoding  $X$  assuming that the lossless version of  $Y$  is available only at the decoder as side information. We denote with  $R_{X|Y}^{WZ}(D)$ , the lower bound of the achievable bit-rate for a given distortion and let  $R_{X|Y}(D)$  denote the rate required if  $Y$



**Figure 2.2:** Slepian-Wolf achievable rate region for distributed compression of two statistically dependent i.i.d. sources [52].

is available at the encoder as well. Unsurprisingly, it was shown that [71]

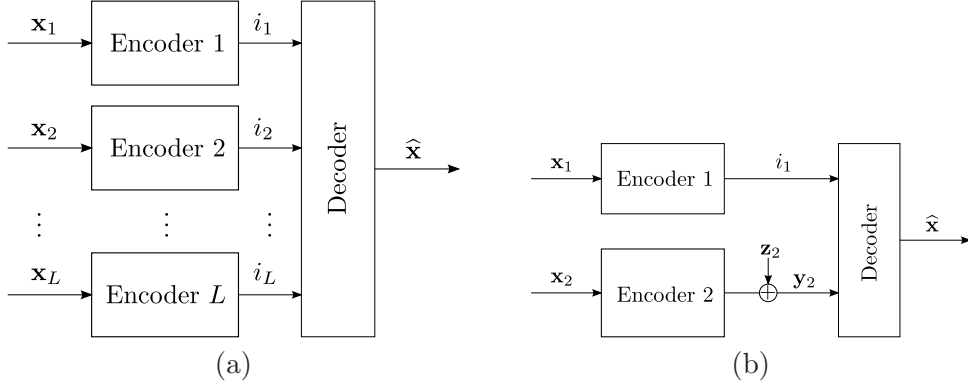
$$R_{X|Y}^{WZ}(D) - R_{X|Y}(D) \geq 0.$$

If, however,  $X$  and  $Y$  are statistically dependent Gaussian sources and if the distortion  $D$  is measured as the expected MSE such that  $D = E[\|X - \hat{X}\|^2]$  then  $R_{X|Y}^{WZ}(D) - R_{X|Y}(D) = 0$ . That is, there is no rate loss or performance loss whether the side information  $Y$  is available only at the decoder or at both the encoder and the decoder.

Lossless DSC is often referred to as Slepian-Wolf coding whereas lossy DSC with side information at the decoder is referred to as Wyner-Ziv coding. Due to the success of the transform coding strategy in the classical centralized case, it is now a common practice in today's practical DSC schemes to also apply a transform at each encoder. This is often referred to as distributed transform coding.

### 2.3.2 Distributed Karhunen-Loève Transform

In [20], Gastpar et al. considered the DSC problem where there are  $L$  independent encoders, each partially observing a jointly Gaussian source vector  $\mathbf{x}$ . This setup is depicted in Figure 2.3 (a), where the first encoder observes the first  $M_1$  components of  $\mathbf{x}$  denoted by  $\mathbf{x}_1$ , the second encoder observes the next  $M_2$  components  $\mathbf{x}_2$  and so on. The  $l$ -th encoder then produces  $k_l$ -dimensional approximation of  $\mathbf{x}_l$  by applying a  $k_l \times M_l$  local transform  $T_l$ . The central decoder receives the transform coefficients  $\mathbf{y}$



**Figure 2.3:** Distributed KLT problem setups: (a)  $L$ -terminals scenario where Encoder  $l$  has access to the subvector  $\mathbf{x}_l$  and the reconstruction is performed jointly; (b) Two-terminals scenario where Encoder 2 applies a fixed known transform  $T_2$  to the observed vector  $\mathbf{x}_2$  and the decoder receives a noisy version  $\mathbf{y}_2 = T_2\mathbf{x}_2 + \mathbf{z}_2$ .

where

$$\mathbf{y} = T\mathbf{x} = \begin{pmatrix} T_1 & 0 & \cdots & 0 \\ 0 & T_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T_L \end{pmatrix} \mathbf{x}.$$

The decoder estimates  $\mathbf{x}$  from  $\mathbf{y}$  and the estimator is given by [20]

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}] = \mathbb{E}[\mathbf{x}|T\mathbf{x}] = \Sigma_x T^T (T \Sigma_x T^T)^{-1} T \mathbf{x},$$

where  $\Sigma_x$  is the covariance matrix of  $\mathbf{x}$ . The corresponding MSE is

$$D = \mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = \text{trace} (\Sigma_x - \Sigma_x T^T (T \Sigma_x T^T)^{-1} T \Sigma_x). \quad (2.13)$$

The goal is then to find a set of local transform  $T_l$  and a quantization strategy that minimize the distortion in (2.13).

For simplicity, we consider the case presented in [20] where there are only two encoders as illustrated in Figure 2.3 (b). Encoder 1 has access to  $\mathbf{x}_1$ , which is the first  $M$  components of  $\mathbf{x}$ , and the second encoder observes  $\mathbf{x}_2$  containing the last  $N - M$  components. The covariance matrices are denoted with  $\Sigma_1 = \mathbb{E}[\mathbf{x}_1 \mathbf{x}_1^T]$ ,  $\Sigma_2 = \mathbb{E}[\mathbf{x}_2 \mathbf{x}_2^T]$  and  $\Sigma_{12} = \mathbb{E}[\mathbf{x}_1 \mathbf{x}_2^T]$ . It is assumed that the transform  $T_2$  is fixed and known at both encoders. The decoder receives a set of noisy coefficients  $\mathbf{y}_2 = T_2 \mathbf{x}_2 + \mathbf{z}_2$ , where  $\mathbf{z}_2$  is a zero-mean

jointly Gaussian vector independent of  $\mathbf{x}_2$ . The aim is to minimize  $D = \mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|^2 | \mathbf{y}_2]$  by devising an optimal local transform  $T_1$  and a compression strategy given that  $\mathbf{y}_2$  is only available as side information at the decoder.

The assumption that  $\mathbf{x}$  and  $\mathbf{z}_2$  are Gaussian means that there exists constant matrices  $A_1$  and  $A_2$  such that

$$\mathbf{x}_2 = A_1 \mathbf{x}_1 + A_2 \mathbf{y}_2 + \mathbf{v},$$

where  $A_1 \mathbf{x}_1 + A_2 \mathbf{y}_2$  is the linear approximation of  $\mathbf{x}_2$  and  $\mathbf{v}$  is a Gaussian random vector independent of  $\mathbf{x}_1$  and  $\mathbf{y}_2$ , which represents the uncertainty that cannot be estimated from  $\mathbf{x}_1$  and  $\mathbf{y}_2$ . Using the same argument, we can also write

$$\begin{pmatrix} I_M \\ A_1 \end{pmatrix} \mathbf{x}_1 = B_2 \mathbf{y}_2 + \mathbf{w},$$

where  $B_2$  is a constant matrix,  $I_M$  is the  $M$ -dimensional identity matrix and  $\mathbf{w}$  is a Gaussian random vector independent of  $\mathbf{y}_2$  with correlation matrix  $\Sigma_w$ . It was then shown in [20] that the optimal local transform  $T_1$  is given by

$$T_1 = Q^T \begin{pmatrix} I_M \\ A_1 \end{pmatrix}.$$

Here,  $Q^T$  is an  $N \times N$  matrix that diagonalizes  $\Sigma_w = Q \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_N^2) Q^T$ , where the eigenvalues  $\lambda_i^2$ ,  $i = 1, 2, \dots, N$ , are in nonincreasing order. Optimality is then achieved by keeping  $k_1$  largest coefficients related to the largest eigenvalues of  $\Sigma_w$  and the corresponding MSE can be simplified to

$$D = \mathbb{E} [\|\mathbf{x} - \hat{\mathbf{x}}\|^2 | \mathbf{y}_2] = \sum_{m=k_1+1}^N \lambda_m^2 + \mathbb{E} [\|\mathbf{v}\|^2].$$

The transform  $T_1$  is known as the *local* KLT [20]. Notice that the local KLT is now different from the centralized one.

In terms of compression, it was shown in [20] that the optimal encoding strategy is to compress each component of  $\mathbf{y}_1$  independently after applying  $T_1$ . The rate allo-

cation depends on the eigenvalues of  $\Sigma_w$ , where the components related to the largest eigenvalues get more rates. This strategy appears to be similar to the centralized case, however, there are some major differences. First, the rate allocation depends on the side information available at the decoder, which affects the matrix  $\Sigma_w$ . Second, while a standard scalar quantizer is optimal for the centralized scenario, it is not necessary the case for the distributed scenario where a more complex quantization scheme based on Wyner-Ziv coding principles may be required.

Finally, consider the case where  $\mathbf{y}_2 = \mathbf{x}_2$ . That is, the exact observation of Encoder 2 is available at the decoder so that the matrices  $A_1 = 0$  and  $\Sigma_w = \Sigma_1 - \Sigma_{12}\Sigma_2^{-1}\Sigma_{12}^T$ . The matrix  $\Sigma_w$  and the local KLT  $T_1$  have size  $M \times M$  and  $T_1$  is called the *conditional* KLT [20]. For a more detailed explanation of distributed KLT and a generalization to multi-terminal scenario, we refer to [20].

### 2.3.3 Practical Distributed Transform Coding with Side Information

In many practical situations, the assumption that the source is Gaussian may not hold and, in fact, the problem of properly modeling real-life signals such as images and videos remains largely open. Hence, the KLT is rarely used in practice and is often replaced by the DCT or the discrete wavelet transform (DWT). This section briefly describes how the concept of source coding with side information is combined with the standard model of transform coding to form a practical distributed transform coding scheme.

#### Practical Slepian-Wolf coding

The first practical coding technique for DSC with side information was the Distributed Source Coding Using Syndromes (DISCUS) [40] introduced in 1999 by Pradhan and Ramchandran. As we will see, DISCUS and other DSC schemes are highly influenced by channel coding techniques. In order to gain the intuition behind the use of channel codes in DISCUS, we first study the example presented in [40].

Let  $X$  and  $Y$  be two correlated 3-bit binary words such that the Hamming distance between them is at most one. If  $Y$  is available at both the encoder and the decoder then we can describe  $X$  using only 2 bits. This is achieved by transmitting the modulo-two

binary sum of  $X$  and  $Y$ , which can take four values i.e.  $\{000, 001, 010, 100\}$ . If, however,  $Y$  is only available at the decoder as side information; can we still transmit  $X$  using only 2 bits?

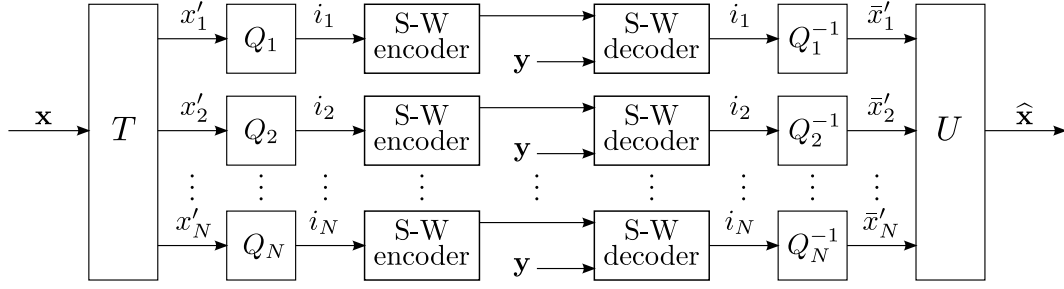
The intuition here is that it is wasteful for the encoder to allocate any bits in order to differentiate between  $X = 000$  and  $X = 111$ . This is because the Hamming distance between these two words is three and since the decoder has access to  $Y$ , it can resolve this uncertainty by picking the word that is closest to  $Y$ . We can, therefore, divide the space of all the possible values of 3-bit binary word into 4 sets such that the Hamming distance between each word in the set is 3 i.e.  $\{000, 111\}$ ,  $\{001, 110\}$ ,  $\{010, 101\}$  and  $\{100, 011\}$ . These sets are called the *cosets* of the 3-bit repetition channel code. Thus, by transmitting only the index of the coset that  $X$  belongs to, which only requires 2 bits, the decoder can retrieve  $X$  perfectly by observing  $Y$ .

This example can be generalized using  $(n, k)$  linear channel codes. In fact, the correlation between the sources can be modeled by a *virtual noisy channel* where  $X$  is the input and  $Y$  is the output of the channel with conditional distribution  $P(Y|X)$ . In the language of channel coding, each coset has a unique *syndrome* given by  $\mathbf{s}_x = H\mathbf{x}^T$ , where  $H$  is the parity-check matrix of a binary linear code. Given  $\mathbf{s}_x$ , the decoder can correct the errors (up to 1 bit in the above example) introduced by the channel and reconstruct  $X$  from  $Y$ . Thus, the problem of DSC in this case can be seen as the problem of finding a channel code that is matched to the correlation distance (or noise) between  $X$  and  $Y$  in the virtual channel model. This approach can also be extended to convolutional codes [1, 44, 40, 19, 53]. Also, [21] presented a distributed coding technique based on linear channel codes that allow a flexible allocation of transmission rates between each independent encoder.

### Practical Wyner-Ziv transform coding

A Wyner-Ziv encoder essentially consists of a quantizer followed by a Slepian-Wolf encoder. In practical distributed compression of images and videos, a Wyner-Ziv encoder is added to the standard transform coding structure. Figure 2.4 shows a typical setup for a distributed transform coding, where  $\mathbf{x} \in \mathbb{R}^N$  is a continuous random vector and





**Figure 2.4:** Practical distributed transform coding scheme. A Wyner-Ziv encoder, which consists of a quantizer followed by a Slepian-Wolf (S-W) encoder, replaces the quantizer and the entropy encoder in the standard transform coding structure.

a random vector  $\mathbf{y}$  is available at the decoder as the side information. As with the centralized case, the transform coefficients  $\mathbf{x}' = T\mathbf{x}$  are still independently quantized but a Wyner-Ziv encoder is used instead of a scalar quantizer followed by an entropy coder. The decoder then uses  $\mathbf{y}$  as side information to recover the quantized transform coefficients  $\bar{\mathbf{x}}'$  and the final estimate of the original source vector is obtained as  $\hat{\mathbf{x}} = U\bar{\mathbf{x}}'$ .

Examples of practical distributed video coding schemes are the Wyner-Ziv video codec in [24, 2, 6] and PRISM (Power-efficient, Robust, hIgh-compression, Syndrome-based Multimedia coding) [41]. Both schemes employ the standard, unmodified, block-based DCT. The quantized transform coefficients are then fed into a sophisticated channel encoder. A Rate Compatible Punctured Turbo (RCPT) coder was used in [2] whereas Syndrome-encoding with a trellis channel coder was used in [41]. It is worth noting that distributed coding can also be applied to enhance or protect the broadcast of video stream from errors introduced in the transmission channel [39, 3].

A wavelet-based distributed coding of multi-view video sequences was proposed in [18], where a network of camera observed and encoded a dynamic scene from different view points. Their scheme uses a motion compensated spatiotemporal wavelet transform (see Chapter 3) followed by a Wyner-Ziv encoder. Here, the decoder obtained the side information from one video signal, which was encoded with a conventional transform coder. Other video signals are then coded using Syndrome coding. Interestingly, it was shown that, at high rates, the motion-compensated Haar wavelet is the optimal transform. In [9], the wavelet-based Slepian-Wolf coding was used to encode hyper-spectral images, which are highly correlated within and across neighboring frequency

bands. The authors also presented a method to estimate the correlation statistics of the wavelet coefficients. We refer to [24, 28, 15] for further details and a complete overview of recent advancement in DSC.

## 2.4 Summary

This chapter has provided a review of centralized and distributed transform coding. Each building block in the classical centralized transform coding structure was discussed in Section 2.2. We also presented the rate-distortion analysis of scalar quantization with optimal bit allocation and stated that the most efficient quantization strategy is the one that uses a uniform scalar quantizer followed by an entropy coder. We also showed that the KLT is the optimal transform for Gaussian sources.

Section 2.3 then reviewed the recent development in distributed transform coding. The theoretical foundations of DSC based on Slepian-Wolf and Wyner-Ziv theorems were given. We provided an overview of the distributed KLT for Gaussian sources and emphasized the changes in the structure of the transform in comparison to the centralized case. Finally, a brief overview of practical DSC schemes was then given. In such schemes, the transform is no different from the centralized case but the quantization strategy is modified by using the Wyner-Ziv coding strategy.

# CHAPTER 3

## State-of-the-Art Compression Methods with Wavelets

### 3.1 Introduction

So far, we have been mostly concentrating on the centralized and distributed transform coding of stationary sources. We have shown that when the source is stationary the optimal transform is the KLT. In such cases, the encoding process is linear and the KLT basis is chosen a priori. Real life signals are, however, non-stationary in nature. For example, images are often modeled with non-stationary piecewise smooth functions. In this case, it is desirable for the encoding process to be adaptive and source-dependent. In this chapter, we will, therefore, depart from the compression of stationary sources and focus on the compression of deterministic piecewise regular sources.

Wavelet theory has had a profound impact on modern signal processing, particularly in the area of signal approximation and compression. Due to its high compression performance, the discrete wavelet transform (DWT) is now a predominant transform in image compression applications such as JPEG-2000 [5, 61, 64]. In addition, the inherent multiresolution property of wavelets makes the DWT a transform of choice for scalable compression schemes [46, 37, 45]. Recent studies have also clarified that the wavelet transform is the best transform for the approximation and compression of piecewise smooth signals [66, 11, 34].

In the next section, we give a brief overview of the wavelet transform. An important concept of Lipschitz regularity of a smooth function is described in Section 3.3. Section 3.4 then presents the approximation results of piecewise smooth signals. Wavelet-based compression schemes including the key distortion-rate results are then discussed in Section 3.5. A summary is then given in Section 3.6. For a more detailed treatment on wavelet theory, we refer to [67, 56, 66, 65, 34].

## 3.2 The Wavelet Transform

Consider a wavelet function  $\psi(t)$  whose set of dilated and shifted versions

$$\psi_{j,n}(t) = 2^{-j/2} \psi(2^{-j}t - n), \quad j, n \in \mathbb{Z},$$

forms a basis of  $\mathbf{L}_2(\mathbb{R})$  where  $\mathbf{L}_2(\mathbb{R})$  is the space of all square-integrable functions. The wavelet transform decomposes a finite energy continuous function  $f(t) \in \mathbf{L}_2(\mathbb{R})$  over a basis  $\{\psi_{j,n}(t)\}_{j,n \in \mathbb{Z}}$  as

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} d_{j,n} \psi_{j,n}(t) \quad (3.1)$$

and the resulting wavelet coefficients  $d_{j,n}$  are given by the inner products

$$d_{j,n} = \langle f(t), \psi_{j,n}(t) \rangle,$$

where we are assuming, for simplicity, that  $\{\psi_{j,n}(t)\}_{j,n \in \mathbb{Z}}$  is an orthogonal basis of  $\mathbf{L}_2(\mathbb{R})$ .

The multiresolution structure of the wavelet transform is reflected in (3.1), where the coefficients  $d_{j,n}$  measure the local variation of  $f(t)$  at resolution  $2^j$ . Let us denote with  $W_j$  a subspace whose basis is given by  $\{\psi_{j,n}(t)\}_{n \in \mathbb{Z}}$ . Equation (3.1) means that we can decompose  $\mathbf{L}_2(\mathbb{R})$  into mutually orthogonal subspaces:

$$\mathbf{L}_2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j.$$

It then follows that an approximation of  $f(t)$  at coarser resolution  $2^{J+1}$  is represented

by the following equation:

$$f_{J+1}(t) = \sum_{j=J+1}^{\infty} \sum_{n=-\infty}^{\infty} d_{j,n} \psi_{j,n}(t). \quad (3.2)$$

The approximation function  $f_{J+1}(t)$  resides in a coarse subspace  $V_J \subset \mathbf{L}^2(\mathbb{R})$  whose basis is formed by  $\{\varphi_{J,n}(t)\}_{n \in \mathbb{Z}}$ , which is a set of dilated and shifted versions of a different function  $\varphi(t)$  called the scaling function. We can, therefore, write

$$f_{J+1}(t) = \sum_{n=-\infty}^{\infty} c_{J,n} \varphi_{J,n}(t) \quad (3.3)$$

with  $\varphi_{J,n}(t) = 2^{-J/2} \varphi(2^{-J}t - n)$  and  $c_{J,n} = \langle f(t), \varphi_{J,n}(t) \rangle$ . The scaling coefficients  $c_{J,n}$  measure the regularity of  $f(t)$  at scale  $2^J$ . Since the basis  $\{\psi_{j,n}(t)\}_{j,n \in \mathbb{Z}}$  is complete, by adding details of  $f_{J+1}(t)$  at finer scales, the function  $f(t)$  can be recovered. Indeed, we have that

$$\mathbf{L}^2(\mathbb{R}) = \bigoplus_{j=-\infty}^J W_j \oplus V_J.$$

where  $V_{j-1} = V_j \oplus W_j$ . Thus, the wavelet transform in (3.1) can be written in terms of (3.2) and (3.3) as

$$f(t) = \sum_{n=-\infty}^{\infty} c_{J,n} \varphi_{J,n}(t) + \sum_{j=-\infty}^J \sum_{n=-\infty}^{\infty} d_{j,n} \psi_{j,n}(t), \quad (3.4)$$

where the first sum represents the coarse version of  $f(t)$  and the double sum contains the missing finer details.

The wavelet function and the scaling function are intimately linked and many properties of the wavelet function can be inferred directly from the scaling function. We say that the function  $\varphi(t)$  is an admissible scaling function of  $\mathbf{L}_2(\mathbb{R})$  if it satisfies the following three conditions [67, 56, 34, 65]:

1. Riesz basis criterion

$$A \leq \sum_{n \in \mathbb{Z}} |\Phi(\omega + 2\pi n)|^2 \leq B,$$

where  $\Phi(\omega)$  is the Fourier transform of  $\varphi(t)$ ;

## 2. Two scale relation

$$\varphi\left(\frac{t}{2}\right) = \sqrt{2} \sum_{k \in \mathbb{Z}} h[k] \varphi(t - k);$$

## 3. Partition of unity

$$\sum_{k \in \mathbb{Z}} \varphi(t - k) = 1.$$

These conditions ensure that there exists a corresponding wavelet function  $\psi(t)$ , which can be expressed as a linear combination of shifted versions of  $\varphi(t)$ :

$$\psi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g[n] \varphi(2t - n),$$

such that  $\psi(t)$  generates a basis of  $\mathbf{L}_2(\mathbb{R})$ . Here, the terms  $h[k]$  and  $g[n]$  represent the coefficients of the filters in the two-channel filterbank structure. We refer to [67, 56, 34] for a detailed treatment on wavelets and filterbanks.

**Vanishing moments**

One of the most well known properties of the wavelet transform is the *vanishing moments* property. The wavelet transform is said to have  $(P+1)$  vanishing moments if its analysis wavelet  $\tilde{\psi}(t)$  (the dual function of  $\psi(t)$ ) suppresses polynomials up to order  $P$ , i.e.

$$\int_{-\infty}^{\infty} t^p \tilde{\psi}(t) dt = 0, \quad \forall p \in \{0, 1, \dots, P\}. \quad (3.5)$$

There is also a direct relationship between the number of vanishing moments and the *order of approximation* of the scaling function. More precisely, we say that a scaling function  $\varphi(t)$  has a  $(P+1)$ -th order of approximation if it reproduces polynomials of maximum degree  $P$ , that is

$$\sum_{n \in \mathbb{Z}} c_n^{(p)} \varphi(t - n) = t^p, \quad \forall p \in \{0, 1, \dots, P\}, \quad (3.6)$$

for a proper choice of the coefficients  $c_n^{(p)}$ . The standard result in wavelet theory states that if the scaling function  $\varphi(t)$  has  $(P+1)$ -th order of approximation, then the corre-

sponding analysis wavelet  $\tilde{\psi}(t)$  has  $(P + 1)$  vanishing moments.

Intuitively, this is because the polynomials  $\{t^p\}_{0 \leq p \leq P}$  reside in the subspace  $V_0$  with a basis  $\{\varphi(t - n)\}_{n \in \mathbb{Z}}$ , which is perpendicular to the subspace  $\tilde{W}_0$  spanned by  $\{\tilde{\psi}(t - n)\}_{n \in \mathbb{Z}}$ . The polynomial suppression property of the wavelet transform gives a sparse representation of a smooth signal since the wavelet coefficients will essentially be zero.

### 3.3 Lipschitz Regularity

We have seen that the local regularity and singularity (i.e. smoothness and variation) of a function are measured by the scaling and wavelet coefficients respectively. This follows from the properties shown in (3.5) and (3.6). In order to characterize the singular structures of a function, the regularity is normally quantified with a Lipschitz exponent (also known as the Hölder exponent) [34].

A formal definition of uniform Lipschitz regularity is given as follows [34]:

**Definition 1** : A function  $f(t)$  is uniformly Lipschitz  $\alpha \geq 0$  over  $[a, b]$  if for all  $v \in [a, b]$  there exists  $K > 0$  and a polynomial  $p_v(t)$  of degree  $m = \lfloor \alpha \rfloor$  such that

$$\forall v \in [a, b], \quad \forall t \in [a, b], \quad f(t) = p_v(t) + \epsilon_v(t) \quad \text{with} \quad |\epsilon_v(t)| \leq K|t - v|^\alpha. \quad (3.7)$$

If  $f(t)$  is uniformly  $\alpha$ -Lipschitz then it is  $\lfloor \alpha \rfloor$  times differentiable and the polynomial  $p_v(t)$  is the Taylor expansion of  $f(t)$  at  $v$ :

$$p_v(t) = \sum_{k=0}^{\lfloor \alpha \rfloor} \frac{f^{(k)}(v)}{k!} (t - v)^k.$$

The following theorem relates the uniform Lipschitz regularity to the asymptotic decay of the Fourier transform:

**Theorem 5** [34]: A function  $f(t)$  with a Fourier transform  $F(\omega)$  is bounded and uniformly Lipschitz  $\alpha$  over  $\mathbb{R}$  if

$$\int_{-\infty}^{+\infty} |F(\omega)| (1 + |\omega|^\alpha) d\omega < \infty.$$

Thus, one can measure the minimum global regularity of a function from its Fourier transform. To locally analyze the regularity over intervals and at points, we turn to the wavelet transform since wavelets are well localized in time.

Let us define the cone of influence of a point  $v$  in the scale-space plane as a set of points  $(j, n)$  such that  $v$  is included in the support of  $\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n)$ . Suppose the wavelet  $\psi(t)$  is of compact support  $C$ , the cone of influence of  $v$  is defined by the set

$$\mathcal{I}_v = \{(j, n) \in \mathbb{Z} : |n2^j - v| \leq C2^j\}. \quad (3.8)$$

The following theorem relates the wavelet coefficients decay across scale with the uniformly Lipschitz  $\alpha$  condition:

**Theorem 6 :** *Given a function  $f(t) \in \mathbf{L}_2(\mathbb{R})$  that is uniformly  $\alpha$ -Lipschitz around  $v$  and a wavelet  $\psi(t)$  with a compact support and at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, the wavelet coefficients in the cone of influence of  $v$  satisfy*

$$|d_{j,n}| \leq A2^{j(\alpha+1/2)} \quad (3.9)$$

with a constant  $A > 0$ .

**Proof:** If  $f(t) = p(t) + \epsilon(t)$  is uniformly  $\alpha$ -Lipschitz and the wavelet function  $\psi(t)$  with a compact support has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments then, for all  $\nu \in [t_a, t_b]$ , the wavelet coefficients are as follows [34]:

$$\begin{aligned} \langle f(t), \psi_{j,n}(t) \rangle &\stackrel{(a)}{=} \underbrace{\langle p(t), \psi_{j,n}(t) \rangle}_{=0} + \langle \epsilon(t), \psi_{j,n}(t) \rangle \\ &\stackrel{(b)}{\leq} K2^{-j/2} \int_{-\infty}^{\infty} |t - \nu|^\alpha \psi(2^{-j}t - n) dt \\ &= K2^{j/2} \int_{-\infty}^{\infty} |x2^j + n2^j - \nu|^\alpha \psi(x) dx \\ &\stackrel{(c)}{\leq} KC2^{j(\alpha+1/2)} \underbrace{\int_{-\infty}^{\infty} (|x| + |C|)^\alpha \psi(x) dx}_{=B} \\ &= A2^{j(\alpha+1/2)} \end{aligned}$$

where at (a),  $\langle p(t), \psi_{j,n}(t) \rangle = 0$  due to the vanishing moments property, (b) follows from the fact that  $|\epsilon(t)| \leq K|t - \nu|^\alpha$  and (c) from the fact that the



wavelet has a compact support  $C$  and therefore  $|n2^j - \nu| \leq C2^j$ . ■

### 3.4 Linear and Nonlinear Approximation with Wavelets

Let us now present the key result of the wavelet-based linear and nonlinear approximation of piecewise smooth functions. A piecewise smooth function  $f(t)$ ,  $t \in [0, 1[$ , with  $K$  pieces is defined as

$$f(t) = \sum_{i=0}^{K-1} f_i(t) \mathbf{1}_{[t_i, t_{i+1}[}(t) \quad (3.10)$$

where  $t_0 = 0$ ,  $t_K = 1$ ,  $f_i(t)$  is uniformly  $\alpha$ -Lipschitz over  $[t_i, t_{i+1}]$  and  $\mathbf{1}_{[t_i, t_{i+1}[}(t) = 1$  for  $t \in [t_i, t_{i+1}[$  and zero elsewhere.

#### 3.4.1 Linear approximation

Given a wavelet  $\psi(t)$  and a corresponding scaling function  $\varphi(t)$ , we can decompose  $f(t) \in \mathbf{L}_2([0, 1])$  as

$$f(t) = \sum_{n=0}^{2^{-J}-1} c_{J,n} \varphi_{J,n}(t) + \sum_{j=-\infty}^J \sum_{n=0}^{2^{-j}-1} d_{j,n} \psi_{j,n}(t) \quad \text{with } J < 0. \quad (3.11)$$

Note that there are  $2^{-j}$  coefficients at scale  $2^j$ . The  $N$ -term linear approximation of  $f(t)$  can then be obtained by representing the function with only  $N$  coefficients. In linear approximation, the choice of these  $N$  coefficients is fixed a priori and is independent of  $f(t)$ . Normally, the first  $N$  coefficients are retained. If we assume that  $N$  is large and of the order  $N \sim 2^{J_N}$  then the linear approximation procedure is equivalent to keeping every coefficient in the first  $J_N$  decomposition levels (also referred to as linear multiresolution approximation). This gives us the following approximation:

$$f_N(t) = \sum_{n=0}^{2^{-J}-1} c_{J,n} \varphi_{J,n}(t) + \sum_{j=J-J_N+1}^J \sum_{n=0}^{2^{-j}-1} d_{j,n} \psi_{j,n}(t), \quad J_N \geq J \geq 1 \quad (3.12)$$

and the squared approximation error is

$$\varepsilon_l(N, f) = \|f(t) - f_N(t)\|^2 = \sum_{j=-\infty}^{J-J_N} \sum_{n=0}^{2^{-j}-1} |d_{j,n}|^2.$$

Here we are assuming for the sake of clarity that the wavelet is orthonormal. Since  $f(t) \in \mathbf{L}^2[0, 1]$ , we have that  $\lim_{N \rightarrow \infty} \varepsilon_l(N, f) = 0$ .

The rate of decay of  $\varepsilon_l(N, f)$  as  $N$  increases is dependent on the rate at which  $|d_{j,n}|$  decays across scales. The following theorems provide the decay characteristic of  $\varepsilon_l(N, f)$  for a uniformly  $\alpha$ -Lipschitz function and a piecewise smooth function:

**Theorem 7** [34]: *If  $f(t)$  is uniformly  $\alpha$ -Lipschitz over  $[0, 1]$  and the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, then*

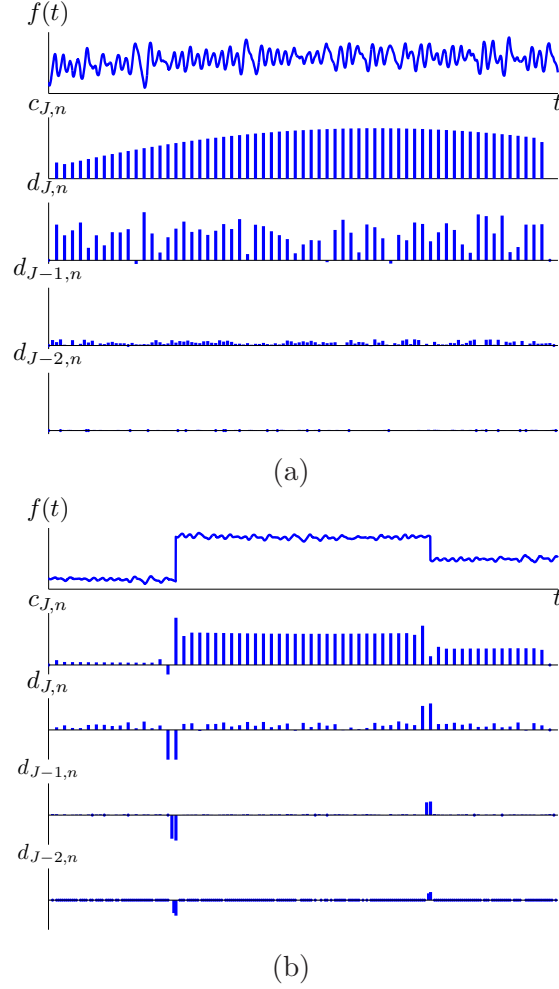
$$\varepsilon_l(N, f) = O(\|f\|_{\tilde{\mathbf{C}}_\alpha} N^{-2\alpha}). \quad (3.13)$$

**Theorem 8** [34]: *If  $f(t)$  is piecewise smooth over  $[0, 1]$  with  $K$  uniformly Lipschitz  $\alpha > 1/2$  pieces and the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, then*

$$\varepsilon_l(N, f) = O(K \|f\|_{\tilde{\mathbf{C}}_\alpha}^2 N^{-1}). \quad (3.14)$$

Here,  $\|f\|_{\tilde{\mathbf{C}}_\alpha}$  denotes the homogeneous Hölder  $\alpha$  norm, which is the infimum of the constant  $K$  that satisfies Lipschitz condition in (3.7) over the interval  $[0, 1]$  [34]. It is clear that the presence of singularity reduces the decay rate of  $\varepsilon_l(N, f)$  to  $N^{-1}$  and this decay is independent of the local regularity of  $f(t)$ .

Figure 3.1 illustrates the wavelet decomposition of a uniformly  $\alpha$ -Lipschitz smooth function and a piecewise smooth function with  $\alpha$ -Lipschitz pieces. It is clear from Figure 3.1 (b) that the wavelet coefficients in the cone of influence of discontinuities dominate the decay across scales. Thus, in comparison to a uniformly smooth function in Figure 3.1 (a), the  $N$ -term linear approximation fails to capture the larger set of coefficients. This has the effect of reducing the decay rate of the approximation error from  $N^{-2\alpha}$  to  $N^{-1}$ . An adaptive approximation method is, therefore, required in order to retain the coefficients in the cone of influence of discontinuities, which brings us to the topic of nonlinear approximation.



**Figure 3.1:** Wavelet decomposition: (a) a uniformly  $\alpha$ -Lipschitz smooth; (b) a piecewise smooth function with  $\alpha$ -Lipschitz pieces. The  $N$ -term linear approximation error decays as  $\varepsilon_l(N, f) \sim N^{-2\alpha}$  in (a) whereas  $\varepsilon_l(N, f) \sim N^{-1}$  in (b). By using an adaptive grid of nonlinear approximation and, instead, keeping the  $N$  largest coefficients, which include the wavelet coefficients in the cone of influence of singularities, we achieve a nonlinear approximation error with a decay of  $\varepsilon_n(N, f) \sim N^{-2\alpha}$ .

### 3.4.2 Nonlinear approximation

Consider the same wavelet decomposition as shown in (3.11) as illustrated in Figure 3.1 but instead of approximating  $f(t)$  by retaining the first  $N$  coefficients, we keep the  $N$  largest coefficients. These large coefficients include the scaling coefficients and the wavelet coefficients in the cone of influence of singularities with slower decays ( $\alpha < 1$ ). Hence, nonlinear approximation defines an adaptive grid with more refined approximation scale in the neighborhood of singularities.

Let  $\mathcal{J}_N$  be the index set of the  $N$  largest coefficients. The best nonlinear approxi-

mation of  $f(t)$  is then given by

$$f_{\mathcal{J}_N}(t) = \sum_{n \in \mathcal{J}_N} c_{J,n} \varphi_{J,n}(t) + \sum_{(j,n) \in \mathcal{J}_N} d_{j,n} \psi_{j,n}(t) \quad (3.15)$$

and the squared approximation error is

$$\varepsilon_n(N, f) = \|f(t) - f_{\mathcal{J}_N}(t)\|^2 = \sum_{n \notin \mathcal{J}_N} |c_{J,n}|^2 + \sum_{(j,n) \notin \mathcal{J}_N} |d_{j,n}|^2.$$

Clearly, we have that  $\varepsilon_n(N, f) \leq \varepsilon_l(N, f)$ . The following theorem gives the decay rate of  $\varepsilon_n(N, f)$  for a piecewise smooth function:

**Theorem 9** [34]: *If  $f(t)$  is piecewise smooth over  $[0, 1]$  with  $K$  uniformly Lipschitz  $\alpha > 1/2$  pieces and the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, then*

$$\varepsilon_n(N, f) = O(\|f\|_{\tilde{\mathbf{C}}_\alpha}^2 N^{-2\alpha}). \quad (3.16)$$

Therefore, from (3.14) and (3.16), we realize that nonlinear approximation is superior to linear approximation when the function is piecewise smooth.

## 3.5 Wavelet-Based Compression

We now review the key aspects of wavelet-based compression, starting from the key rate-distortion results on the compression of piecewise smooth functions. This is followed by the concept of embedded coding of wavelet coefficients. Lastly, a brief review of wavelet compression in higher dimensions (i.e. image and video compression) is given.

### 3.5.1 Distortion-rate results

Essentially, one can think of compression as a process of approximation followed by quantization. That is, the encoder only allocates bits to a certain number of coefficients that are kept. Therefore, it is not surprising that the distortion-rate performance of a wavelet-based coder is dependent on its approximation strategy and, hence, how the

approximation error decays. We refer to [11] for a detailed derivation of the  $D(R)$  functions.

Suppose that we are to compress a smooth function that satisfies a uniform  $\alpha$ -Lipschitz condition given in Definition 1. The following theorem gives the  $D(R)$  function of a compression scheme based on linear approximation (see Appendix A.1 for proof):

**Theorem 10** [11]: *If  $f(t)$  is uniformly  $\alpha$ -Lipschitz over  $[0,1]$  and the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, the distortion-rate function of a compression scheme that allocates the bits to the first  $N$  coefficients is*

$$D(R) \leq c_1 R^{-2\alpha}. \quad (3.17)$$

If  $N \sim 2^{J_N}$  such that the coefficients in the first  $J_N$  decomposition levels are kept, from Theorem 6, this is equivalent to setting the step size of a uniform quantizer to  $\Delta = A2^{(J-J_N+1)(\alpha+1/2)}$ .

Instead, if a function is piecewise smooth as described by Equation (3.10), we know from the previous section that nonlinear approximation gives a better approximation result. The following theorem highlights the difference in distortion-rate performances of linear and nonlinear approximation-based compression schemes:

**Theorem 11** [11]: *If  $f(t)$  is piecewise smooth over  $[0,1]$  with  $K$  uniformly Lipschitz  $\alpha > 1/2$  pieces and the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, the distortion-rate function of a compression scheme that allocates bits to the first  $N$  coefficients is*

$$D(R) \leq c_2 R^{-2\alpha} + c_3 R^{-1}. \quad (3.18)$$

*If, instead, a compression scheme allocates bits to the  $N$  largest coefficients, then*

$$D(R) \leq c_4 R^{-2\alpha} + c_5 \sqrt{R} 2^{-c_6 \sqrt{R}}. \quad (3.19)$$

One can easily see that the decay of the linear and nonlinear approximation errors in Theorems 8 and 9 directly translate to the decay characteristic of the  $D(R)$  functions in Theorem 11. Thus, at high rates, the distortion of a nonlinear compression scheme decays as  $R^{-2\alpha}$  whereas a scheme with a linear approximation strategy has a slower decay of  $R^{-1}$ . It is due to this difference that current state-of-the-art compression algorithms are nonlinear.

It is worth noting that the computational complexity of a nonlinear compression algorithm is much higher than that of a linear one. Furthermore, while a linear encoder only uses the bits for quantization, a nonlinear encoder also needs to allocate the bits for the indexing of the coefficients being transmitted. Therefore, in order to maximize the rate-distortion performance of a nonlinear scheme, it is also crucial for the encoder to employ an efficient indexing algorithm. In today's standards, embedded coding is the most widely used method to quantize and index wavelet coefficients.

### 3.5.2 Embedded coding

Current state-of-the-art image and video compression algorithms use embedded codes to progressively enhance the quality of the reconstructed source as the decoder receives more bits. Embedded coding algorithms are designed to organize and transmit the coefficients by their order of magnitude with the bit-plane method. Starting with the most significant bit plane, the encoder keeps going down to a less significant bit plane after each iteration. Thus, the coefficients are effectively quantized with a step size of  $2^n$  where  $n$  is iteratively decremented to progressively improve the resolution of the received coefficients. Examples of the most well known wavelet-based embedded coding algorithms are the Embedded Zerotree Wavelet (EZW) [48], Set Partitioning in Hierarchical Trees (SPIHT) [43] and Embedded Block Coding with Optimized Truncation (EBCOT) [60], which is used in JPEG-2000 standard [61].

Let us denote the 2-D transform coefficients with  $y_{m,n} = \langle f, g_{m,n} \rangle$ ,  $m, n \in \mathbb{Z}$ . We define a set of indices  $\Theta_k$ ,  $k \in \mathbb{Z}$ , where

$$\Theta_k = \{m, n : 2^k \leq |y_{m,n}| < 2^{k+1}\}$$

and a significant map  $b_k[m, n]$  such that

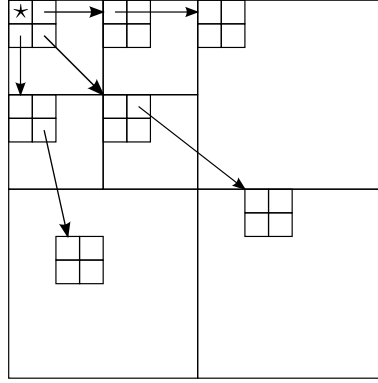
$$b_k[m, n] = \begin{cases} 0 & m, n \notin \Theta_k \\ 1 & m, n \in \Theta_k. \end{cases}$$

The different steps of an embedded coding algorithm can be summarized as follows [43]:

1. *Initialization*: Store the index  $k = \max_{(m,n)} \lfloor \log_2 |y_{m,n}| \rfloor$ ;
2. *Sorting pass*: Store the significance map  $b_k[m, n]$  for  $m, n \notin \Theta_{k+1}$  and code the sign of  $y_{m,n}$  for  $m, n \in \Theta_k$ ;
3. *Refinement pass*: Store the  $i$ -th bit of the coefficients in the sets  $\Theta_k$  for  $k > i$  whose positions are already recorded in the previous passes;
4. *Precision refinement*: Decrease  $k$  by 1 and go to Step 2.

The above algorithm can be stopped at any time. In addition, the ordering information is not transmitted. This is because both the encoder and decoder employ the same sorting algorithm. Therefore, the encoder's execution path is perfectly duplicated at the decoder. Furthermore, it is not necessary to sort every coefficient and the number of magnitude comparisons can be reduced with an appropriate sorting algorithm. For example, SPIHT uses the set partitioning sorting algorithm based on a spatial orientation tree of wavelet coefficients (see Figure 3.2), which allows it to take advantage of the dependencies across scales of the wavelet coefficients.

A detailed rate-distortion analysis of wavelet based transform coding including the impact on the performance of embedded coding can be found in [35]. As we have discussed earlier, the  $D(R)$  function for a non-Gaussian source is proportional to  $2^{-2R}$  at high bit rates. However, Mallat *et al* [35] showed that when the bit rate is low at  $R < 1$  per pixel,  $D(R)$  decays like  $CR^{1-2\gamma}$ , where  $\gamma$  is an exponent of order 1 that varies slowly as  $\log_2 R$ . They also proved that the use of embedded coding improves the performance by changing the constant  $C$  but not the exponent that determines the decay.



**Figure 3.2:** Parent-offspring dependencies in the spatial-orientation tree of wavelet coefficients found in SPIHT algorithm [43].

### 3.5.3 Wavelets in scalable video compression

Let us begin with the basic concept of interframe redundancy in video coding. When the source is an image, transforms are used to exploit the spatial or intraframe redundancy. If, however, the source is a video sequence then the coder also has to exploit the temporal or interframe redundancy in order to achieve good compression performance. The most widely used technique by today's standards is motion compensated prediction (MCP) [23]. We refer to [51, 58, 36] overviews of current video compression standards.

Let  $f_i(x, y)$  denote the  $i$ -th frame in the video sequence. In MCP, a prediction  $\tilde{f}_i(x, y)$  of  $f_i(x, y)$  is generated by the encoder, usually from the previous frame. The residual  $r_i(x, y) = f_i(x, y) - \tilde{f}_i(x, y)$  is then coded and transmitted along with the prediction parameters. Block-based MCP strategy is now the most established algorithm and is implemented in every video coding standard. First, the frame  $f_i(x, y)$  is divided into  $K$  disjoint blocks  $\{B_k\}_{1 \leq k \leq K}$ , then the prediction is formed by

$$\tilde{f}_i(x, y) = \hat{f}_{i-1}(x - u_{i,k}, y - v_{i,k}), \quad x, y \in B_k, \quad k = 1, 2, \dots, K,$$

where  $\hat{f}_{i-1}(x, y)$  is the previously decoded frame and  $[u_{i,k}, v_{i,k}]$  is the motion vector.

In recent years, a growing demand for media-rich applications over variable bandwidth channels has posed a new set of requirements in terms of spatial, temporal and signal-to-noise ratio (SNR) scalability [37, 45]. Even though standards such as MPEG-

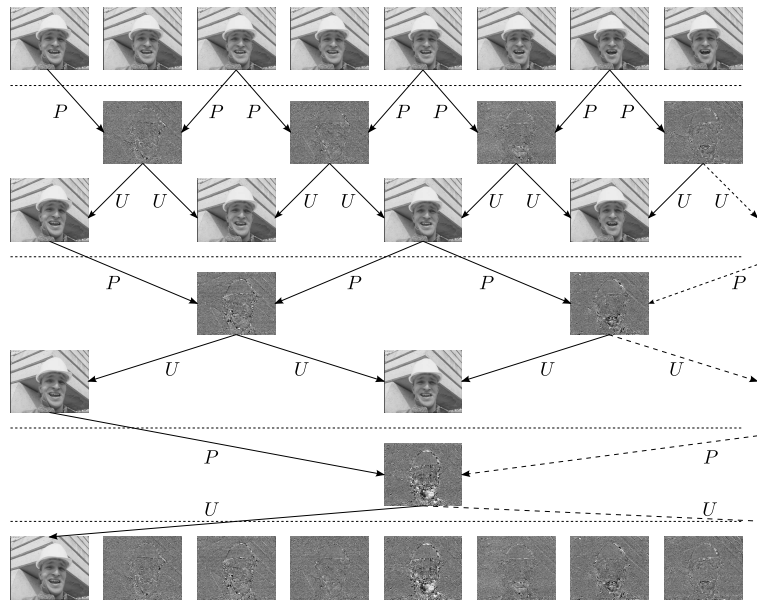


x and H.26x can achieve state-of-the-art compression performance, their closed-loop prediction structures have limited scalability. On the other hand, due to the inherent multi-resolution feature of the wavelet transform, wavelet based video coding offers a wide range of scalability while achieving high compression performance.

One of the most promising development in scalable video coding is motion compensated temporal filtering (MCTF) [10], where a wavelet transform is applied in the temporal axis as shown in Figure 3.3. In order to fully exploit the interframe redundancy, temporal filtering is performed along the direction of motion. Loosely speaking, this is equivalent to performing a block-based MCP on the video frames prior to temporal filtering. Current MCTF algorithms are based on the efficient implementation of the wavelet transform using the *lifting scheme* [59, 8]. The MCTF algorithm is now a part of the scalability extension to the current H.264/AVC standard [45]. Due to their low complexity, the most widely used wavelets for this purpose are the Haar and LeGall 5/3 [33]. Indeed, the wavelet transform can also be used to decompose the frame spatially to form a 3-D spatiotemporal wavelet transform. Examples of such video coding algorithms are the 3-D SPIHT [32], the lifting-based invertible motion adaptive transform (LIMAT) [46] and the Barbell-lifting coding scheme [72]. The rate-distortion analysis for wavelet-based scalable video coder can be found in [70].

## 3.6 Summary

This chapter focussed on the wavelet transform and the approximation and compression results for piecewise smooth functions. We stated that nonlinear approximation of piecewise smooth functions produce an error that decays as  $N^{-2\alpha}$ , which is better than a linear approximation method with a decay of  $N^{-1}$  ([11, 34]). These results translate directly to the distortion-rate performances of compression schemes that allocate the rates based on linear or nonlinear approximation strategies. We show that the corresponding  $D(R)$  functions decay as  $R^{-1}$  and  $R^{-2\alpha}$  respectively. A summary of embedded coding algorithm, which is the most widely used nonlinear approximation-based rate allocation strategy, was then presented. Finally, we gave a brief overview on the use of wavelets in scalable video compression.



**Figure 3.3:** Examples of MCTF implementation with the LeGall 5/3 lifting scheme. Three decomposition levels are shown. The video frames are divided into even and odd frames. The even frames are used to form the prediction of the odd frames by means of motion compensation. Each motion compensated prediction is multiplied with the prediction coefficient and the high-pass subband is calculated by taking the difference. Similarly, the high-pass subbands are multiplied by the update coefficient to produce the low-pass subbands. Here, the prediction steps are denoted with  $P$  and the update steps are denoted with  $U$ .

# CHAPTER 4

## Sampling of Signals with Finite Rate of Innovation

### 4.1 Introduction

Sampling theory plays a central role in the foundation of modern signal processing. Shannon's classical sampling theory states that it is possible to linearly reconstruct a bandlimited signal from its samples using a kernel with infinite support i.e. an ideal low-pass filter or a sinc function. Although the theory is very powerful, its idealization poses too restrictive constraints for real-world acquisition devices.

With the influence from the recent development in wavelet theory, the sampling process has been re-interpreted as an approximation of the original signal by projecting it onto a shift-invariant subspace of bandlimited functions. This interpretation led to the extension of the theorem to classes of non-bandlimited signal that belong to a shift-invariant subspace [63].

Later on, Vetterli *et al* [68] showed that it is possible to sample and reconstruct a class of non-bandlimited signals that do not reside in a fixed subspace. This class of signal is called signals with Finite Rate of Innovation or FRI signals. Such signals have finite degrees of freedom or rate of innovation. However the scheme still employed a sampling kernel with infinite support, which cannot be realized in practice.

Recently, in [14], Dragotti *et al.* demonstrated that many FRI signals can be sampled

and perfectly reconstructed using a wide range of kernels with finite support. The results in [14] were then extended to multidimensional FRI signals in [49]. Baboulaz *et al.* also applied the sampling theory of FRI signals to develop new techniques for image feature extraction and image super-resolution [7].

Our focus for the chapter is to review the key results presented in [14]. We start by looking at a generic 1-D sampling setup, followed by a formal definition of FRI signals. Section 4.4 describes the basic properties of sampling kernels that reproduce polynomials. Sampling schemes that sample and parametrically reconstruct non-bandlimited FRI signals are then presented in Section 4.5. Finally, Section 4.6 shows how the theory can be extended to a 2-D sampling scenario. The results from the sampling theory discussed in this chapter will be used in our original work in the following chapters of the thesis.

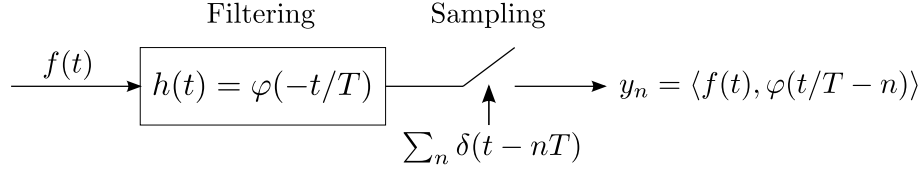
## 4.2 Sampling Setup

Let us first consider a generic 1-D sampling setup as shown in Figure 4.1. This setup represents a good abstraction of today's acquisition devices and can be extended to higher dimensions. Here, a continuous time signal  $f(t)$ ,  $t \in \mathbb{R}$ , is filtered by a (typically low-pass) filter  $h(t)$  before being uniformly sampled with a sampling period  $T$  where  $T \in \mathbb{R}^+$ . The output is a set of discrete samples  $\{y_n\}_{n \in \mathbb{Z}}$ , given by the inner product

$$y_n = \langle f(t), \varphi(t/T - n) \rangle = \int_{-\infty}^{\infty} f(t) \varphi(t/T - n) dt, \quad (4.1)$$

where the sampling kernel  $\varphi(t)$  is the scaled and time-reversed version of the impulse response  $h(t)$  of the filter.

The key problem in sampling theory is how to best reconstruct  $f(t)$  from the samples  $y_n$ . More precisely, one needs to answer the following questions: 1) What classes of signals can be reconstructed? 2) What classes of sampling kernels can be used? 3) What are the reconstruction algorithms involved? As mentioned earlier, our focus will be on a class of signals called FRI signals. The work in this thesis uses a class of sampling kernels that reproduce polynomials, which includes any valid scaling functions of the



**Figure 4.1:** Generic 1-D sampling setup: a continuous signal  $f(t)$ ,  $t \in \mathbb{R}$ , is filtered by a sampling kernel  $h(t) = \varphi(-t/T)$  and uniformly sampled by  $\sum_n \delta(t - nT)$  to obtain a set of discrete samples  $y_n = \langle f(t), \varphi(t/T - n) \rangle$ , where  $n \in \mathbb{Z}$ .

wavelet transform.

### 4.3 Signals with Finite Rate of Innovation

Let us now formally state the notion of FRI signals as given in [68]. Given a continuous signal  $f(t)$  that can be defined with a parametric representation, if the number of parameters is finite then  $f(t)$  is said to have a finite degree of freedom or a *finite rate of innovation*. That is, the signal  $f(t)$  can be described by the following form:

$$f(t) = \sum_{n \in \mathbb{Z}} \sum_{k=0}^K a_{n,k} \phi_k(t - t_n), \quad (4.2)$$

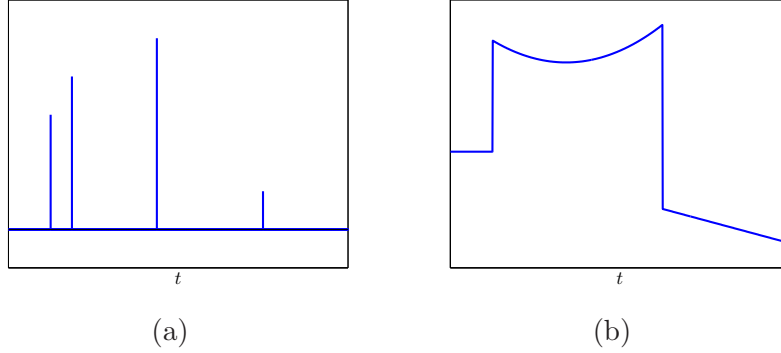
where the coefficients  $a_{n,k}$  and the shifts  $t_n$  are free parameters and the set of functions  $\{\phi_k(t)\}_{k=0,\dots,K}$  is known. Let  $C_f(t_a, t_b)$  be the counter function that counts the number of free parameters of  $f(t)$  over the interval  $[t_a, t_b]$ . The global rate of innovation is then defined as [68]

$$\rho = \lim_{\ell \rightarrow \infty} \frac{1}{\ell} C_f\left(-\frac{\ell}{2}, \frac{\ell}{2}\right). \quad (4.3)$$

The definition of an FRI signal is, therefore, given by

**Definition 2** (*Vetterli, Marziliano and Blu, [68]*): A signal with finite rate of innovation is a signal whose parametric representation is given in Equation (4.2) with a finite  $\rho$  as defined in Equation (4.3).

In some cases, however, it is more convenient to find a local rate of innovation with respect to a moving window of size  $\ell$ . The local rate of innovation at time  $t$  is defined



**Figure 4.2:** Examples of 1-D signals with finite rate of innovation: (a) streams of Diracs; (b) piecewise polynomial signals.

as [68]

$$\rho_\ell(t) = \frac{1}{\ell} C_f \left( t - \frac{\ell}{2}, t + \frac{\ell}{2} \right).$$

There is a wide range of signals that fall under Definition 2. Interestingly, these also include bandlimited signals. Let  $f_B(t)$  be a real bandlimited signal with maximum non-zero frequency  $f_{max}$ . The well known Shannon's sampling theorem states that  $f_B(t)$  can be represented with the following reconstruction formula:

$$f_B(t) = \sum_{n=-\infty}^{\infty} f_B(nT) \text{sinc} \left( \frac{t - nT}{T} \right),$$

where we are assuming that  $T = \frac{1}{f_{max}}$ . This is in fact the same representation as given in Equation (4.2) and the corresponding rate of innovation is  $\rho = 1/T = f_{max}$ .

Consider now a stream of  $K$  Diracs given by  $f(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k)$  with  $t \in [0, 1]$ . The signal is non-bandlimited but can also be classified as a FRI signal as the only free parameters are the amplitudes  $a_k$  and the locations  $t_k$ . Hence, the global rate of innovation is equal to  $2K$ . Another example of a non-bandlimited FRI signal is a piecewise polynomial signal defined on  $t \in [0, 1]$ , with  $K$  pieces of maximum degree  $R$ . Each polynomial piece can be described by  $(R+1)$  coefficients and the signal, therefore, has a finite rate of innovation  $\rho = (R+2)K$ . These two examples are illustrated in Figure 4.2.

## 4.4 Polynomial Reproducing Kernels

In [14], three classes of sampling kernels were proposed: polynomial reproducing kernels, exponential reproducing kernels and kernels with rational Fourier transform. As mentioned earlier, this thesis will only focus on kernels that reproduce polynomials.

### 4.4.1 Polynomial Reproduction Property

Polynomial reproducing kernels are any function  $\varphi(t)$  that together with its shifted versions can reproduce polynomials of maximum degree  $P$  as described by the following equation:

$$\sum_{n \in \mathbb{Z}} c_n^{(p)} \varphi(t/T - n) = t^p \quad \text{with } p = 0, 1, \dots, P \quad (4.4)$$

with a proper choice of coefficients  $\{c_n^{(p)}\}$ . A function  $\varphi(t)$  with compact support can reproduce polynomials if it satisfies the following conditions [55]:

$$\left\{ \begin{array}{ll} \Phi(0) \neq 0 & \text{and} \\ \frac{d^p \Phi(2i\pi)}{d\omega^p} = 0 & \text{for } i > 0, \quad p = 0, 1, \dots, P, \end{array} \right. \quad (4.5)$$

where  $\Phi(\omega)$  is the Fourier transform of  $\varphi(t)$ . These conditions are known as the Strang-Fix conditions of order  $(P + 1)$ .

There exists a wide range of functions satisfying Strang-Fix conditions. In terms of wavelet theory, any admissible scaling function of  $\mathbf{L}_2(\mathbb{R})$  reproduces polynomials. We recall from the previous chapter that a scaling function  $\varphi(t)$  with  $(P + 1)$ th-order of approximation reproduces polynomials of maximum degree  $P$ . One important family of scaling function is the B-spline family [62]. A B-spline function of order  $P \geq 0$  reproduces polynomial of maximum degree  $P$  and can be obtained by the  $(P + 1)$ -fold convolution of the box function  $\beta_0(t)$ :

$$\beta_P(t) = \underbrace{\beta_0(t) * \beta_0(t) \dots * \beta_0(t)}_{=P+1 \text{ times}}, \quad \text{with } B(\omega) = \frac{1 - e^{-j\omega}}{j\omega}.$$

Alternatively,  $\beta_P(t)$  can be expressed with the following formula:

$$\beta_P(t) = \frac{1}{P!} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l (t-l)_+^P,$$

where

$$(t)_+^P = \begin{cases} 0 & t < 0, \\ t^P & t \geq 0. \end{cases}$$

Interestingly, in [65], Unser and Blu reformulated the wavelet theory based on their B-spline factorization theorem and obtained the following result:

**Theorem 12** [65]:  $\varphi(t)$  is a valid scaling function with approximation order  $(P+1)$  if and only if its Fourier transform  $\Phi(\omega)$  can be factorized as  $\Phi(\omega) = B_+^P(\omega)\Phi_0(\omega) = \left(\frac{1-e^{-j\omega}}{j\omega}\right)^{P+1}\Phi_0(\omega)$ .

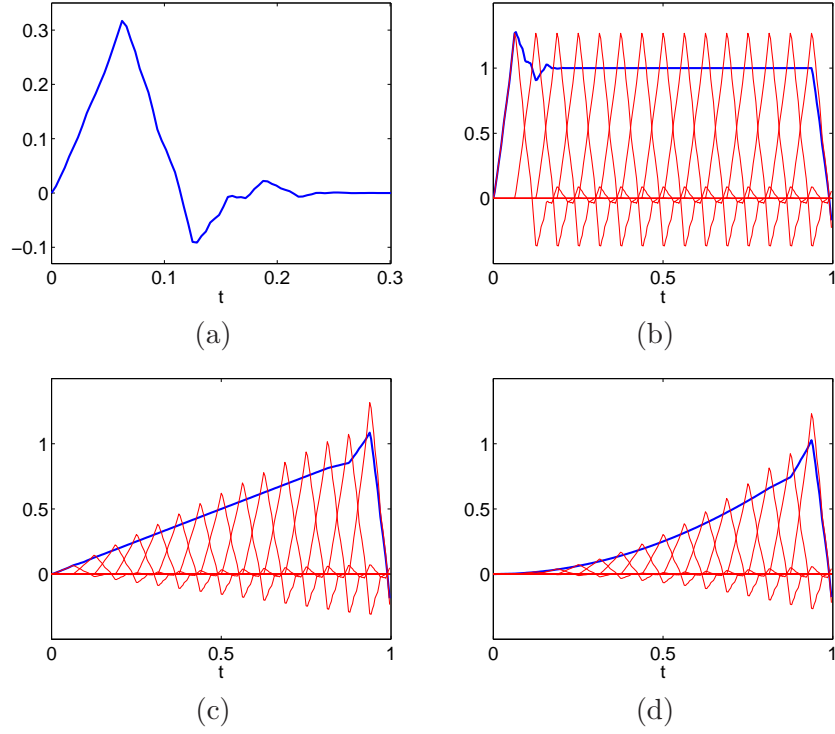
The function  $B_+^P(\omega)$  is the Fourier transform of a B-spline function of order  $P$  and  $\Phi_0(\omega)$  is a bounded function of  $\omega$ . This result is derived from the filter bank implementation of the wavelet transform. From Theorem 12, it is clear that any valid scaling function will satisfy Strang-Fix conditions in Equation (4.5). The polynomial reproduction property is illustrated with a Daubechies scaling function of order 3 in Figure 4.3. Note that this scaling function can reproduce polynomials of maximum degree two.

#### 4.4.2 Polynomial Reproduction Coefficients

There are two main methods to retrieving the polynomial reproduction coefficients  $\{c_n^{(p)}\}$  of equation (4.4). First, the coefficients can be calculated from the dual of the scaling function  $\varphi(t)$ . The dual function  $\tilde{\varphi}(t)$  satisfies the following property:

$$\langle \varphi(t-m), \tilde{\varphi}(t-n) \rangle = \delta_{m,n} \quad (4.6)$$





**Figure 4.3:** Polynomial reproduction property of 1-D kernels as given by Equation (4.4): (a) Daubechies scaling function of order 3; (b) zeroth order polynomial ( $p = 0$ ); (c) first order polynomial ( $p = 1$ ); (d) second order polynomial ( $p = 2$ ); the blue lines in (b),(c) and (d) represent the reproduced polynomials and the red lines are the scaled and shifted versions of the db3 scaling function.

with  $\delta_{m,n}$  denoting the Kronecker's delta function. It then follows that

$$\begin{aligned}
 (1/T) \langle t^p, \tilde{\varphi}(t/T - n) \rangle &\stackrel{(a)}{=} (1/T) \left\langle \sum_k c_n^{(p)} \varphi(t/T - k), \tilde{\varphi}(t/T - n) \right\rangle \\
 &\stackrel{(b)}{=} (1/T) \sum_k c_n^{(p)} \langle \varphi(t/T - k), \tilde{\varphi}(t/T - n) \rangle \\
 &\stackrel{(c)}{=} \sum_k c_k^{(p)} \delta_{k,n} \\
 &= c_n^{(p)},
 \end{aligned} \tag{4.7}$$

where (a) and (c) follow from Equations (4.4) and (4.6) respectively and (b) comes from the linearity of the inner product. One can, therefore, directly obtain  $\{c_n^{(p)}\}$  from Equation (4.7).

The second method provides an alternative to the first, especially when  $\tilde{\varphi}(t)$  is difficult to calculate. It involves obtaining the coefficients numerically by constructing

and solving a system of equations from Equation (4.4). Let  $L_\varphi$  denote the length of the support of  $\varphi(t/T)$ , it follows that at any point in the polynomial reproduced by Equation (4.4), there exist  $N_\varphi = \lceil L_\varphi/T \rceil$  overlapping scaled and shifted versions of  $\varphi(t/T)$  that, together, make up  $t^p$ . The coefficients  $\{c_n^{(p)}\}$  can, therefore, be obtained by solving the following system of equations:

$$\begin{bmatrix} \varphi_n\left(\frac{t_1}{T}\right) & \varphi_{n+1}\left(\frac{t_1}{T}\right) & \cdots & \varphi_{n+N_\varphi-1}\left(\frac{t_1}{T}\right) \\ \varphi_n\left(\frac{t_2}{T}\right) & \varphi_{n+1}\left(\frac{t_2}{T}\right) & \cdots & \varphi_{n+N_\varphi-1}\left(\frac{t_2}{T}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_n\left(\frac{t_{N_\varphi}}{T}\right) & \varphi_{n+1}\left(\frac{t_{N_\varphi}}{T}\right) & \cdots & \varphi_{n+N_\varphi-1}\left(\frac{t_{N_\varphi}}{T}\right) \end{bmatrix} \begin{bmatrix} c_n^{(p)} \\ c_{n+1}^{(p)} \\ \vdots \\ c_{n+N_\varphi-1}^{(p)} \end{bmatrix} = \begin{bmatrix} \left(\frac{t_1}{T}\right)^p \\ \left(\frac{t_2}{T}\right)^p \\ \vdots \\ \left(\frac{t_{N_\varphi}}{T}\right)^p \end{bmatrix}, \quad (4.8)$$

where for simplicity, we denote  $\varphi(t-n)$  with  $\varphi_n(t)$ . Moreover,  $\{t_1, t_2, \dots, t_{N_\varphi}\}$  are chosen such that  $\{t_1, t_2, \dots, t_{N_\varphi}\} \in [(n + N_\varphi - 2)T, (n + N_\varphi - 1)T]$ .

#### 4.4.3 Moment-Samples Relationship

Assuming that the samples  $y_n$  of  $f(t)$  are obtained with the kernel that reproduces polynomials of maximum degree  $P$  as shown in Equation (4.4), we now show that it is possible to perfectly retrieve the exact continuous moments of  $f(t)$  up to order  $P$  from the samples  $y_n$ . We refer to this result as the *moment-samples relationship*. The continuous moment of order  $p$  of  $f(t)$  is defined as

$$M_p = \int_{-\infty}^{\infty} f(t) t^p dt. \quad (4.9)$$

It then follows that

$$\begin{aligned} M_p &= \sum_n c_n^{(p)} y_n \\ &\stackrel{(a)}{=} \sum_n c_n^{(p)} \langle f(t), \varphi(t/T - n) \rangle \\ &\stackrel{(b)}{=} \left\langle f(t), \sum_n c_n^{(p)} \varphi(t/T - n) \right\rangle \\ &\stackrel{(c)}{=} \int_{-\infty}^{\infty} f(t) t^p dt \quad \text{with } p = 0, 1, \dots, P, \end{aligned} \quad (4.10)$$

where (a) and (c) are directly from Equations (4.1) and (4.3) and (b) follows from the linearity of the inner product. As we will see in the next section, this relationship described by Equation (4.10) is at the heart of the sampling schemes presented in [14] and [49].

## 4.5 Sampling Schemes for FRI Signals

We now review in detail the sampling schemes for three types of FRI signals as presented in [14]; these include streams of Diracs, streams of differentiated Diracs and piecewise polynomial functions.

### 4.5.1 Streams of Diracs

Let  $f(t)$  be a stream of  $K$  Diracs defined by a set of parameters  $\{a_k, t_k\}_{k=0, \dots, K-1}$ , corresponding respectively to the amplitudes and locations, where

$$f(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k). \quad (4.11)$$

The function  $f(t)$  is sampled by a kernel  $\varphi(t)$  that reproduces polynomials according to Equation (4.4) to obtain  $y_n$ . The goal is then to reconstruct  $f(t)$  parametrically by retrieving  $\{a_k, t_k\}_{k=0, \dots, K-1}$  from the observed samples  $y_n$ . The reconstruction algorithm presented in [14] is non-linear and operates in three steps: first, the exact continuous moments of  $f(t)$  are retrieved; second, the locations  $\{t_k\}$  are found using the annihilating filter method and, lastly, the amplitudes  $\{a_k\}$  are obtained by solving a Vandermonde system of equations.

Clearly, the exact continuous moment  $M_p$  of order  $p$  of  $f(t)$  has the form

$$M_p = \sum_{k=0}^{K-1} a_k \int \delta(t - t_k) t^p dt = \sum_{k=0}^{K-1} a_k t_k^p. \quad (4.12)$$

We can retrieve  $M_p$  directly from  $y_n$  using the relationship shown in Equation (4.10).

More precisely, we have that

$$\begin{aligned}
 M_p &= \sum_n c_n^{(p)} y_n \\
 &= \left\langle \sum_{k=0}^{K-1} a_k \delta(t - t_k), \sum_n c_n^{(p)} \varphi(t/T - n) \right\rangle \\
 &= \int_{-\infty}^{\infty} \sum_{k=0}^{K-1} a_k \delta(t - t_k) t^p dt \\
 &= \sum_{k=0}^{K-1} a_k t_k^p \quad \text{with } p = 0, 1, \dots, P.
 \end{aligned}$$

The locations  $\{t_k\}$  can now be obtained from the retrieved moments using the annihilating filter method.

Assume that we have a filter with coefficients  $h_p$ ,  $p = 0, 1, \dots, K$ , whose  $z$  transform is given by

$$H(z) = \sum_{p=0}^K h_p z^{-p} = \prod_{k=0}^{K-1} (1 - t_k z^{-1}),$$

that is, the zeros of  $H(z)$  corresponds to a set of  $K$  locations  $\{t_k\}_{k=0, \dots, K-1}$ . By filtering a sequence of moments  $M_p$ ,  $p = 0, 1, \dots, 2K - 1$  with  $h_p$ , where  $M_p$  is given by Equation (4.12), it follows that

$$h_p * M_p = \sum_{i=0}^K h_i M_{p-i} = \sum_{i=0}^K \sum_{k=0}^{K-1} a_k h_i t_k^{p-i} = \sum_{k=0}^{K-1} a_k t_k^p \underbrace{\sum_{i=0}^K h_i t_k^{-i}}_{=0} = 0. \quad (4.13)$$

In other words, the filter  $h_p$  annihilates the sequence  $M_p$ . Hence, the filter is called annihilating filter. Since  $h_0 = 1$ , we can write (4.13) in a matrix form with  $2K$  consecutive values of  $M_p$ , which leads to the following Yule-Walker system of equations:

$$\begin{bmatrix} M_{K-1} & M_{K-2} & \cdots & M_0 \\ M_K & M_{K-1} & \cdots & M_1 \\ \vdots & \vdots & \ddots & \vdots \\ M_{N-1} & M_{N-2} & \cdots & M_{N-K} \end{bmatrix} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_K \end{pmatrix} = - \begin{pmatrix} M_K \\ M_{K+1} \\ \vdots \\ M_N \end{pmatrix}. \quad (4.14)$$

Therefore, the locations  $\{t_k\}_{k=0, \dots, K-1}$  can be obtained by solving Equation (4.14) and

finding the roots of  $h_p$ . This method requires the retrieval of the first  $2K - 1$  moments, which means that the sampling kernel  $\varphi(t)$  must be able to reproduce polynomials up to order  $2K - 1$ .

The amplitudes  $\{a_k\}_{k=0,\dots,K-1}$  can now be calculated by forming a system of equations from Equation (4.12), which gives

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_0 & t_1 & \cdots & t_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ t_0^{K-1} & t_1^{K-1} & \cdots & t_{K-1}^{K-1} \end{bmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{K-1} \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{K-1} \end{pmatrix}. \quad (4.15)$$

This is a Vandermonde system, which has a unique solution if  $t_k \neq t_l, \forall k \neq l$ . Having retrieved the amplitudes and locations  $\{a_k, t_k\}_{k=0,\dots,K-1}$ , the original Dirac function  $f(t)$  can be reconstructed from Equation (4.11).

If we now assume that  $f(t)$  is an infinite-length stream of Diracs, then one can attempt to reconstruct  $f(t)$  *locally* using the method described above. In fact, the following result was obtained in [14]:

**Theorem 13** [14]: *Given a sampling kernel  $\varphi(t)$  of length  $L_\varphi$  that can reproduce polynomials of maximum degree  $N \geq 2K - 1$ , an infinite-length stream of Diracs  $f(t) = \sum_{n \in \mathbb{Z}} a_k \delta(t - t_k)$  is uniquely determined by its samples  $y_n = \langle f(t), \varphi(t/T - n) \rangle$  if there are at most  $K$  Diracs in an interval of size  $2KL_\varphi T$ .*

#### 4.5.2 Streams of Differentiated Diracs

We now assume that  $f(t)$  is a stream of differentiated Diracs given by

$$f(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R_k-1} a_{k,r} \delta^{(r)}(t - t_k),$$

with  $K$  locations and  $\hat{K} = \sum_{k=0}^{K-1} R_k$  weights. Recall that the  $r$ -th derivative of a Dirac has the property  $\int_{-\infty}^{\infty} f(t) \delta^{(r)}(t - t_0) dt = (-1)^r f^{(r)}(t_0)$ . The exact  $p$ -th order continuous

moment  $M_p$  of  $f(t)$  is, therefore, given by

$$\begin{aligned}
 M_p = \sum_n c_n^{(p)} y_n &= \int_{-\infty}^{\infty} f(t) t^p dt \\
 &= \sum_{k=0}^{K-1} \sum_{r=0}^{R_k-1} a_{k,r} \int_{-\infty}^{\infty} \delta^{(r)}(t - t_k) t^p dt \\
 &\stackrel{(a)}{=} \sum_{k=0}^{K-1} \sum_{r=0}^{R_k-1} a_{k,r} (-1)^r \frac{p!}{(p-r)!} t_k^{p-r}, \tag{4.16}
 \end{aligned}$$

where (a) follows from the fact that  $\int_{-\infty}^{\infty} t^p \delta^{(r)}(t - t_0) dt = (-1)^r (p!/(p-r)!) t_0^{p-r}$ .

As in the previous case, Equation (4.10) is used to retrieve the exact moments  $M_p$  from the samples  $y_n$ . It was shown in [14] that a filter with  $z$ -transform

$$H(z) = \prod_{k=0}^{K-1} (1 - t_k z^{-1})^{R_k}$$

annihilates the sequence  $M_p$  where  $M_p$  is given by Equation (4.16). Similarly to the previous scheme, the locations  $\{t_k\}_{k=0,\dots,K-1}$  can then be retrieved with the annihilating filter method i.e. by solving a Yule-Walker system of equations using  $2\hat{K}$  consecutive moments of  $f(t)$ . The weights  $\{a_{k,r}\}$  can then be obtained by constructing and solving the Vandermonde system of equations from Equation (4.16). Note that we now need the kernel to reproduce polynomials up to order  $2\hat{K} - 1$ .

For an infinite-length stream of differentiated Diracs, the following result was obtained in [14]:

**Theorem 14** [14]: *Given a sampling kernel  $\varphi(t)$  of length  $L_\varphi$  that can reproduce polynomials of maximum degree  $N \geq 2\hat{K} - 1$ , an infinite-length stream of differentiated Diracs  $f(t) = \sum_{k \in \mathbb{Z}} \sum_{r=0}^{R_k-1} a_{k,r} \delta^{(r)}(t - t_k)$  is uniquely determined by its samples  $y_n = \langle f(t), \varphi(t/T - n) \rangle$  if there are at most  $K$  Diracs with  $\hat{K}$  weights in an interval of size  $2\hat{K}L_\varphi T$ .*

### 4.5.3 Piecewise Polynomial Functions

Let us now consider a piecewise polynomial function  $f(t)$  with  $K$  pieces of maximum degree  $R - 1$  ( $R > 0$ ), which can be written as

$$f(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} a_{k,r} (t - t_k)_+^r \quad (4.17)$$

with  $t_+ = \max(t, 0)$ . The  $R$ -th derivative of  $f(t)$  is a stream of differentiated Diracs where

$$f^{(R)}(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R-1} r! a_{k,r} \delta^{(R-r-1)}(t - t_k).$$

Intuitively, if we are able to obtain the samples of  $f^{(R)}(t)$  from the original samples  $y_n$  of  $f(t)$  together with the continuous moments of  $f^{(R)}(t)$ , then we can reconstruct  $f^{(R)}(t)$  from Theorem 14 and, hence,  $f(t)$ .

By taking the Fourier transform of a function  $\varphi(t) - \varphi(t - 1)$ , we have that

$$\varphi(t) - \varphi(t - 1) \iff \Phi(\omega) (1 - e^{-j\omega}) = j\omega \Phi(\omega) \frac{(1 - e^{-j\omega})}{j\omega} = j\omega \Phi(\omega) B_0(\omega),$$

where  $B_0(\omega)$  is the Fourier transform of the zero-th order B-spline function  $\beta_0(t)$ . Hence,

$$\varphi(t) - \varphi(t - 1) = \frac{d}{dt} [\varphi(t) * \beta_0(t)]. \quad (4.18)$$

Let  $z_n^{(R)}$  denote the  $R$ -th order finite difference of  $y_n$  where  $z_n^{(r)} = z_{n+1}^{(r-1)} - z_n^{(r-1)}$  and  $z_n^{(1)} = y_{n+1} - y_n$ . It was shown in [14] that

$$\begin{aligned} z_n^{(1)} = y_{n+1} - y_n &= \langle f(t), \varphi(t/T - n - 1) - \varphi(t/T - n) \rangle \\ &\stackrel{(a)}{=} \left\langle f(t), -\frac{d}{dt} [\varphi(t/T - n) * \beta_0(t/T - n)] \right\rangle \\ &\stackrel{(b)}{=} \left\langle \frac{df(t)}{dt}, \varphi(t/T - n) * \beta_0(t/T - n) \right\rangle, \end{aligned}$$

where (a) is from Equation (4.18) and (b) follows from integration by parts. This leads

to the following result:

$$z_n^{(R)} = \left\langle f^{(R)}(t), \varphi(t/T - n) * \beta_{R-1}(t/T - n) \right\rangle, \quad (4.19)$$

where  $\beta_{R-1}(t)$  denotes the  $(R - 1)$ -th order B-spline function. The  $R$ -th order finite difference  $z_n^{(R)}$  of  $y_n$ , therefore, represents the sampling of  $f^{(R)}(t)$  with the new kernel  $\varphi(t/T - n) * \beta_{R-1}(t/T - n)$ .

From Equation (4.10), we now have that

$$M_p^{(R)} = \sum_{n \in \mathbb{Z}} c_n^{(p)} z_n^{(R)}, \quad (4.20)$$

where  $M_p^{(R)}$  is the  $p$ -th order continuous moment of  $f^{(R)}(t)$ . The polynomial reproduction coefficients  $\{c_n^{(p)}\}$  of the new kernel  $\varphi(t/T - n) * \beta_{R-1}(t/T - n)$  can be obtained using the methods shown in Equations (4.7) and (4.8). The function  $f^{(R)}(t)$  and, hence,  $f(t)$  (as given in Equation (4.17)) can, therefore, be reconstructed from its samples  $y_n$  using Equations (4.17) and (4.20) together with the annihilating filter method shown earlier in this section. Note that the scaling function  $\varphi(t)$  now must be able to reproduce polynomials up to order  $2KR - 1 - R$ . We summarize the review of this sampling scheme with the following theorem from [14]:

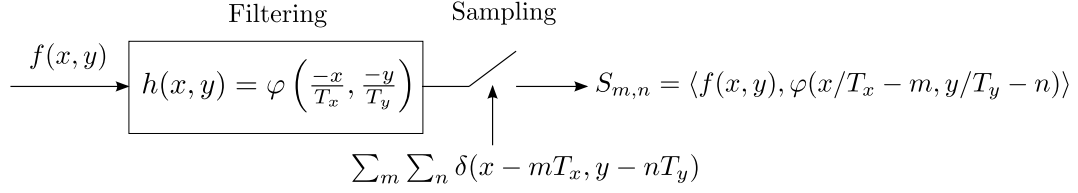
**Theorem 15** [14]: *Given a sampling kernel  $\varphi(t)$  of length  $L_\varphi$  that can reproduce polynomials of maximum degree  $P$ , an infinite-length piecewise polynomial function  $f(t)$  with pieces of maximum degree  $R$  (as given by Equation (4.17)) is uniquely determined by its samples  $y_n = \langle f(t), \varphi(t/T - n) \rangle$  if there are at most  $K$  polynomial discontinuities in an interval of size  $2K(L_\varphi + R)T$  and  $P + R \geq 2KR - 1$ .*

## 4.6 Sampling of 2-D FRI Signals

### 4.6.1 2-D Sampling Setup

Figure 4.4 shows the setup for the sampling of 2-D FRI signals. A continuous 2-D function  $f(x, y) \in \mathbb{R}^2$  with  $x, y \in \mathbb{R}$  is first convolved with a 2-D sampling kernel  $h(x, y) = \varphi(-x/T_x, -y/T_y)$  prior to being sampled with sampling periods  $T_x$  and  $T_y$





**Figure 4.4:** Generic 2-D sampling setup: a continuous signal  $f(x, y)$ ,  $x, y \in \mathbb{R}$ , is filtered by a sampling kernel  $h(x, y) = \varphi(-x/T_x, -y/T_y)$  and uniformly sampled by  $\sum_m \sum_n \delta(x - mT_x, y - nT_y)$  to obtain a set of discrete samples  $S_{m,n} = \langle f(x, y), \varphi(x/T_x - m, y/T_y - n) \rangle$ , where  $m, n \in \mathbb{Z}$  and  $T_x, T_y \in \mathbb{R}^+$ .

where  $T_x, T_y \in \mathbb{R}^+$ . The output is a set of 2-D discrete samples  $\{S_{m,n}\}_{m,n \in \mathbb{Z}}$ , given by the following inner product:

$$\begin{aligned}
 S_{m,n} &= \langle f(x, y), \varphi(x/T_x - m, y/T_y - n) \rangle \\
 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \varphi(x/T_x - m, y/T_y - n) dx dy.
 \end{aligned} \tag{4.21}$$

For simplicity, we assume that  $T_x = T_y = T$  unless explicitly specified. As with the 1-D case, we focus on a class of 2-D sampling kernels that have finite support and reproduce polynomials.

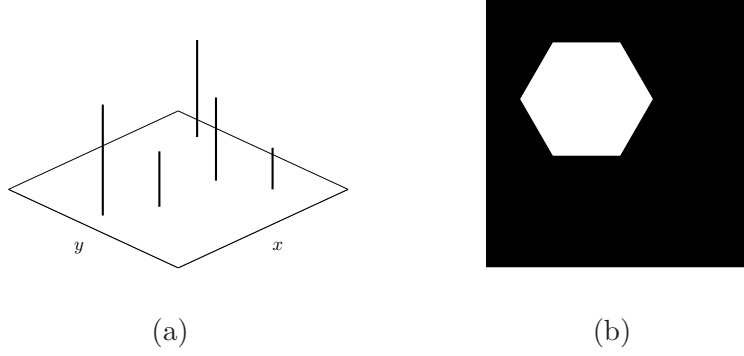
#### 4.6.2 2-D Signals with Finite Rate of Innovation

The notion of 2-D FRI signals can be easily obtained from the definition of 1-D FRI signals given in Definition 2. Consider a 2-D continuous function  $f(x, y)$  that can be represented in the following parametric form:

$$f(x, y) = \sum_{i \in \mathbb{Z}} \sum_{j \in \mathbb{Z}} \sum_{k=0}^K a_{i,j,k} \phi_k(x - x_i, y - y_j), \tag{4.22}$$

with a known set of functions  $\{\phi_k(x, y)\}_{k=0, \dots, K}$ . That is, the only free unknown parameters are the coefficients  $a_{i,j,k} \in \mathbb{R}$  and the spatial shifts  $(x_i, y_j) \in \mathbb{R}$ .

As with the 1-D case, one can introduce a function  $C_f([x_a, x_b], [y_a, y_b])$  that counts the number of free parameters over the window  $[x_a, x_b] \times [y_a, y_b]$ . The global rate of



**Figure 4.5:** Examples of 2-D signals with finite rate of innovation: (a) a set of Diracs; (b) a bilevel polygon image.

innovation is then defined as

$$\rho_{xy} = \lim_{\ell_x \rightarrow \infty, \ell_y \rightarrow \infty} \frac{1}{\ell_x \ell_y} C_f \left( \left[ -\frac{\ell_x}{2}, \frac{\ell_x}{2} \right], \left[ -\frac{\ell_y}{2}, \frac{\ell_y}{2} \right] \right). \quad (4.23)$$

A 2-D FRI signal is, therefore, a signal whose parametric representation is given by Equation (4.22) with a finite  $\rho_{xy}$  as defined in Equation (4.23). Similarly, the local rate of innovation of  $f(x, y)$  at  $(x, y)$  is given by

$$\rho_{\ell_x, \ell_y}(x, y) = \frac{1}{\ell_x \ell_y} C_f \left( \left[ x - \frac{\ell_x}{2}, x + \frac{\ell_x}{2} \right], \left[ y - \frac{\ell_y}{2}, y + \frac{\ell_y}{2} \right] \right).$$

Clearly, a set of 2-D Diracs given by  $f(x, y) = \sum_{k \in \mathbb{Z}} a_k \delta(x - x_k, y - y_k)$  is a 2-D FRI signal. Other examples of 2-D FRI signals include bilevel polygon images as illustrated in Figure 4.5. It was shown in [49] that these two types of signals can be sampled and perfectly reconstructed using variations of the methods presented in [14].

### 4.6.3 2-D Polynomial Reproducing Kernels

The polynomial reproduction property of 2-D kernels  $\varphi(x, y)$  is also a straightforward extension from the 1-D case and can be written as follows:

$$\sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} \varphi(x/T - m, y/T - n) = x^p y^q \quad (4.24)$$

with  $p = 0, 1, \dots, P$  and  $q = 0, 1, \dots, Q$ . As shown in Section 4.4.2, the polynomial reproduction coefficients  $\{c_{m,n}^{(p,q)}\}$  can be obtained from the dual function  $\tilde{\varphi}(x, y)$  of  $\varphi(x, y)$ :

$$c_{m,n}^{(p,q)} = (1/T) \langle x^p y^q, \tilde{\varphi}(x/T - m, y/T - n) \rangle. \quad (4.25)$$

The proof of Equation (4.25) is a direct extension of Equation (4.7). Alternatively, one can obtain  $\{c_{m,n}^{(p,q)}\}$  numerically by constructing and solving a system of equations from Equation (4.24) using the procedure described in Section 4.4.2.

Let us now consider a 2-D separable kernel where  $\varphi(x, y)$  is the tensor product of two 1-D functions  $\varphi_1(x)$  and  $\varphi_2(y)$ :

$$\varphi(x, y) = \varphi_1(x) \otimes \varphi_2(y).$$

Moreover, we assume that both  $\varphi_1(x)$  and  $\varphi_2(y)$  reproduce polynomials according to Equation (4.4) with corresponding sets of coefficients  $\{c_m^{(p)}\}$  and  $\{c_n^{(q)}\}$  where  $p = 0, 1, \dots, P$  and  $q = 0, 1, \dots, Q$ . It follows that

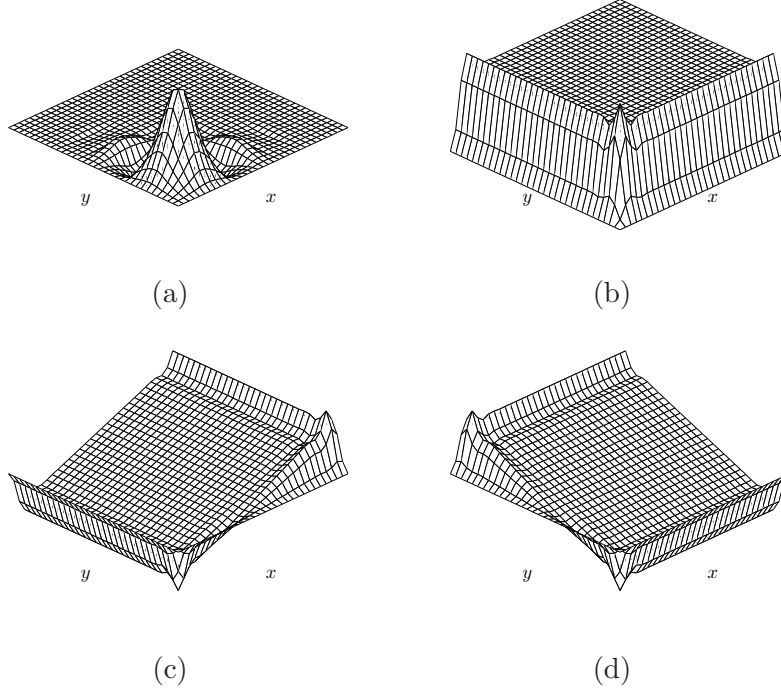
$$\sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} \varphi_1(x/T - m) \varphi_2(y/T - n) = x^p y^q \quad \text{with} \quad c_{m,n}^{(p,q)} = c_m^{(p)} c_n^{(q)}. \quad (4.26)$$

One can, therefore, easily obtain  $\{c_{m,n}^{(p,q)}\}$  from  $\{c_m^{(p)}\}$  and  $\{c_n^{(q)}\}$  in this case. Figure 4.6 illustrates the polynomial reproduction property of a 2-D Daubechies scaling function of order 4.

The 2-D continuous geometric moment of order  $(p+q)$ ,  $p, q \in \mathbb{N}$ , of  $f(x, y)$  is defined as

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy. \quad (4.27)$$

By following the proof in Equation (4.10), we arrive at the following 2-D moment-



**Figure 4.6:** Polynomial reproduction property of 2-D kernels as given by Equation (4.24): (a) 2-D Daubechies scaling function of order 4; (b) zeroth order polynomial with  $p = 0, q = 0$ ; (c) first order polynomial along  $x$ -direction with  $p = 1, q = 0$ ; (d) first order polynomial along  $y$ -direction with  $p = 0, q = 1$ . Note that the grid scale used in (a) is different from (b),(c) and (d)

samples relationship:

$$\begin{aligned}
 M_{p,q} &= \sum_m \sum_n c_{m,n}^{(p,q)} S_{m,n} \\
 &\stackrel{(a)}{=} \sum_m \sum_n c_{m,n}^{(p,q)} \langle f(x, y), \varphi(x/T_x - m, y/T_y - n) \rangle \\
 &\stackrel{(b)}{=} \left\langle f(x, y), \sum_m \sum_n c_{m,n}^{(p,q)} \varphi(x/T_x - m, y/T_y - n) \right\rangle \\
 &\stackrel{(c)}{=} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) x^p y^q dx dy,
 \end{aligned} \tag{4.28}$$

where (a) and (c) follow from Equations (4.21) and (4.24) and (b) from linearity of the inner product. Such ability to exactly retrieve geometric moments of 2-D signals from their samples is very useful. Shukla and Dragotti used the relationship in Equation (4.28) to build three reconstruction algorithms for bilevel polygon images; namely, directional-derivative-based, complex-moments-based and radon-transform-based algo-

rithms (see [49] for an in-depth coverage). Baboulaz *et al.* [7] applied the FRI principle to accurately register multiple images from their samples and this leads to a novel image super-resolution technique.

As a working example, let us consider a 2-D single Dirac function given by

$$f(x, y) = a_1 \delta(x - x_1, y - y_1).$$

Clearly, the rate of innovation equals to three as the only free parameters are the amplitude  $a_1$  and the coordinate  $(x_1, y_1)$ . We now assume that the function is sampled with a 2-D kernel  $\varphi(x, y)$  that reproduces polynomials of maximum degree  $(P + Q)$  as shown in Equation (4.24), where  $P, Q \geq 1$ . It is easy to see that the geometric moments  $M_{p,q}$  of  $f(x, y)$ , up to the first order, are as follows:

$$M_{0,0} = a_1, \quad M_{1,0} = a_1 x_1, \quad \text{and} \quad M_{0,1} = a_1 y_1.$$

We can, therefore, easily reconstruct  $f(x, y)$  from the samples  $S_{m,n}$  by retrieving the moments  $M_{0,0}$ ,  $M_{1,0}$  and  $M_{0,1}$  using Equation (4.28) and, consequently, the parameters  $a_1$  and  $(x_1, y_1)$ .

## 4.7 Summary

The basics of sampling theory for FRI signals have been reviewed in this chapter. A formal notion of FRI signals, first introduced in [68], was given in Definition 2. We exclusively focused on sampling kernels that reproduce polynomials with a particular emphasis on the moment-samples relationship (see Equation (4.10)). Under a generic sampling setup shown in Figure 4.1, we demonstrated using sampling schemes from [14] that non-bandlimited FRI signals can be sampled and perfectly reconstructed. Finally, we looked at the extension of the theory in a 2-D scenario. In the next chapter, we will present our original work on semi-parametric compression of piecewise smooth functions, which also uses the reconstruction algorithm described in Section 4.5.3.

# CHAPTER 5

## Centralized Semi-Parametric Compression of Piecewise Smooth Functions

### 5.1 Introduction

Recent mathematical analysis on the performance of wavelets in compression, which we reviewed in Chapter 3, has led to new interesting insights into the connections between compression and source approximation [11, 66, 34]. It is now well understood that under a standard transform coding structure, a compression algorithm based around a nonlinear approximation-based bit allocation strategy outperforms a linear compression algorithm.

If we have a continuous smooth function that satisfies the uniform  $\alpha$ -Lipschitz condition, we saw in Chapter 3 that the  $D(R)$  function of a wavelet-based compression scheme that allocates the bits according to a linear approximation strategy is given by

$$D(R) \leq c_1 R^{-2\alpha}. \quad (5.1)$$

If, instead, a function is piecewise smooth, Theorem 11 (Chapter 3) showed that the

$D(R)$  of the same wavelet-based compression scheme is

$$D(R) \leq c_2 R^{-2\alpha} + c_3 R^{-1}, \quad (5.2)$$

where the dominating decay rate of  $R^{-1}$  is due to the presence of discontinuities. In contrast, given the same function, a compression scheme that employs a nonlinear approximation strategy achieves the following  $D(R)$  function:

$$D(R) \leq c_4 R^{-2\alpha} + c_5 \sqrt{R} 2^{-c_6 \sqrt{R}}. \quad (5.3)$$

It is clear that at high rates, a nonlinear compression scheme gives a better distortion-rate performance with a decay of  $R^{-2\alpha}$ . Hence, conventional compression schemes are characterized by a complex nonlinear encoder and a simple decoder.

We will show in this chapter that, given a piecewise smooth function, a compression scheme whose encoder allocates the bits according to a linear approximation strategy can also achieve, in some cases, the  $D(R)$  decay of  $R^{-2\alpha}$  at high rates. This is made possible by incorporating a parametric estimation procedure into the decoder, which enables it to estimate the singular structures of a function, i.e. the locations of discontinuities, from the linearly approximated coefficients. Thus, the resulting decoding process is nonlinear. This architecture is, therefore, the *dual* of the traditional one since the computational complexity is transferred from the encoder to the decoder. There are situations such as compression in sensor network where a simple encoder is desirable.

In the next section, we recall the notion of piecewise smooth signals. We then propose our semi-parametric compression strategy in Section 5.3. A theoretical estimate of the  $D(R)$  function of the proposed compression scheme is computed in Section 5.5. Section 5.6 presents a constructive compression algorithm followed by simulation results in Section 5.7. Finally, conclusions are drawn in Section 5.8.

## 5.2 Modeling of Piecewise Smooth Functions

Throughout this chapter, we assume that the support of the continuous function is normalized to  $[0, 1[$ . We define the regularity of a function with the Lipschitz exponent. Recall from Definition 1 (Chapter 3) that a function  $f(t)$  restricted to  $[a, b]$  is said to be uniformly  $\alpha$ -Lipschitz over  $[a, b]$ , with  $\alpha \geq 0$ , if it can be written as

$$f(t) = p(t) + \epsilon(t), \quad (5.4)$$

where  $p(t)$  is a polynomial of degree  $m = \lfloor \alpha \rfloor$  and there exists a constant  $K > 0$  such that

$$\forall t \in (a, b) \text{ and } \forall \nu \in [a, b], \quad |\epsilon(t)| \leq K|t - \nu|^\alpha.$$

A piecewise smooth function  $f(t)$ ,  $t \in [0, 1[$  with  $K + 1$  pieces is then defined as

$$f(t) = \sum_{i=0}^K f_i(t) \mathbf{1}_{[t_i, t_{i+1}[}(t) \quad \text{with} \quad \mathbf{1}_{[a, b[}(t) = \begin{cases} 1 & \forall t \in [a, b[, \\ 0 & \text{otherwise.} \end{cases} \quad (5.5)$$

Here,  $t_0 = 0$ ,  $t_{K+1} = 1$  and  $f_i(t)$  is uniformly  $\alpha$ -Lipschitz over  $[t_i, t_{i+1}]$ .

Given a function  $f(t)$  as defined by (5.5), it was shown in [16] that  $f(t)$  can in fact be decomposed into two functions, namely, a piecewise polynomial function that contains the singular structure and a globally smooth function. Hence, our piecewise smooth functions can be written as follows:

$$f(t) = f_p(t) + f_\alpha(t), \quad (5.6)$$

where  $f_\alpha(t)$  is uniformly  $\alpha$ -Lipschitz over  $[0, 1[$  and  $f_p(t)$  is a piecewise polynomial function with pieces of maximum degree  $\lfloor \alpha \rfloor$ . The piecewise polynomial signal can be written as follows:

$$f_p(t) = \sum_{k=0}^K \sum_{r=0}^{\lfloor \alpha \rfloor} a_{r,k} (t - t_k)_+^r \quad (5.7)$$

with  $t_+ = \max(t, 0)$ . This signal model will be used in our analysis in the forthcoming sections.



## 5.3 Semi-Parametric Compression Strategy

### 5.3.1 Linear Approximation

First, let us recall that the wavelet decomposition of a continuous function  $f(t)$ ,  $t \in [0, 1]$ , is formally expressed as

$$f(t) = \sum_{n=0}^{L-1} c_{J,n} \varphi_{J,n}(t) + \sum_{j=-\infty}^J \sum_{n=0}^{2^{-j}-1} d_{j,n} \psi_{j,n}(t), \quad \text{with } J < 0, \quad (5.8)$$

where the basis elements of  $\mathbf{L}_2([0, 1])$  are the shifted and dilated versions of the scaling function  $\varphi(t)$  and the wavelet function  $\psi(t)$ :

$$\varphi_{J,n}(t) = 2^{-J/2} \varphi(2^{-J}t - n) \quad \text{and} \quad \psi_{j,n}(t) = 2^{-j/2} \psi(2^{-j}t - n).$$

The low-pass and high-pass coefficients,  $\{c_{J,n}\}$  and  $\{d_{j,n}\}$  respectively, are given by the following inner products:

$$c_{J,n} = \langle f(t), \tilde{\varphi}_{J,n}(t) \rangle \quad \text{and} \quad d_{j,n} = \langle f(t), \tilde{\psi}_{j,n}(t) \rangle,$$

where  $\tilde{\varphi}_{J,n}$  and  $\tilde{\psi}_{j,n}$  are the dual of  $\varphi_{J,n}$  and  $\psi_{j,n}$ . We denote the total number of low-pass coefficients with  $L$ , where

$$L = 2^{-J}.$$

The  $N$ -term linear approximation of  $f(t)$  where  $N \sim 2^{J_N}$  is then given by

$$f_N(t) = \sum_{n=0}^{L-1} c_{J,n} \varphi_{J,n}(t) + \sum_{j=J-J_N+1}^J \sum_{n=0}^{2^{-j}-1} d_{j,n} \psi_{j,n}(t), \quad \text{with } J_N \geq 1. \quad (5.9)$$

### 5.3.2 Semi-Parametric Compression Algorithm

We now introduce the concept of semi-parametric compression algorithm. Consider a piecewise smooth function  $f(t)$  given by the signal model in (5.6). Intuitively, one can recover  $f(t)$  by reconstructing the piecewise polynomial function  $f_p(t)$  and the

smooth function  $f_\alpha(t)$  separately. Since  $f_\alpha(t)$  is uniformly  $\alpha$ -Lipschitz, a compression method based on linear approximation shown in (5.9) can be used to compress  $f_\alpha(t)$  with  $D(R) \sim R^{-2\alpha}$ . On the other hand, the function  $f_p(t)$  is completely determined by a finite set of parameters. Hence, we can reconstruct  $f_p(t)$  by estimating the locations  $\{t_i\}_{0 \leq i \leq K+1}$  and the polynomial coefficients  $\{a_{r,k}\}_{0 \leq r \leq \lfloor \alpha \rfloor, 0 \leq k \leq K}$ . The reconstruction of  $f_p(t)$  can, therefore, be viewed as a parametric estimation problem.

First, we consider the wavelet decomposition of  $f(t)$  as shown in (5.8) and assume that the wavelet has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments. We denote with  $\mathcal{J}_p$  a set of indices such that

$$\mathcal{J}_p = \{(j, n) \in \mathbb{Z} : |\langle f_p(t), \psi_{j,n}(t) \rangle| > 0\}.$$

In other words, the coefficients  $\{d_{j,n}\}_{j,n \in \mathcal{J}_p}$  are in the *cone of influence* of discontinuities found in  $f_p(t)$ . On the other hand, the coefficients in  $\{d_{j,n}\}_{j,n \notin \mathcal{J}_p}$  are outside the cone of influence and the wavelet coefficients decay as  $d_{j,n} \sim 2^{j(\alpha+1/2)}$  (from Theorem 6, Chapter 3). This is illustrated in Figure 5.1.

### Linear approximation-based quantization

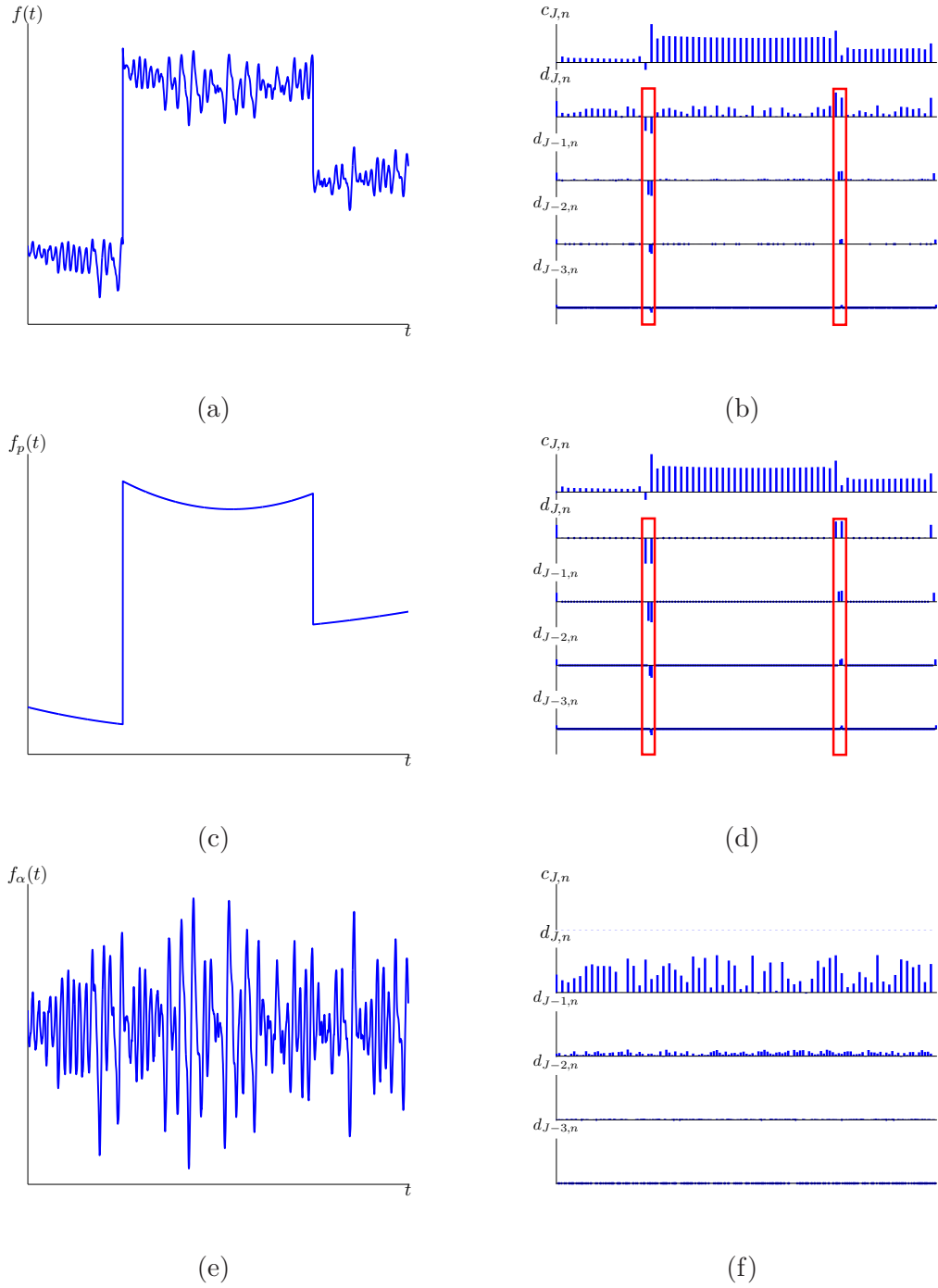
Let us now address the linear approximation-based quantization strategy in details. With linear approximation shown in (5.9), the function  $f_\alpha(t)$  is approximated by transmitting only the coefficients in decomposition level  $j = J - J_N + 1, \dots, J$ . Since the wavelet coefficients of  $f_\alpha(t)$  decays as  $d_{j,n} \leq A2^{j(\alpha+1/2)}$  (from Theorem 6), this is equivalent to setting the quantizer step size  $\Delta$  at

$$\Delta = A2^{(J-J_N+1)(\alpha+1/2)},$$

where  $A = \max_{j,n} |d_{j,n}|$ . Therefore, the number of bits allocated to each coefficient at resolution  $2^{-j}$  is given by

$$R_{j,\alpha} = \left\lceil \log_2 \left( \frac{A2^{j(\alpha+1/2)}}{\Delta} \right) \right\rceil + 1, \quad j = J - J_N + 1, \dots, J, \quad (5.10)$$

where an extra bit is needed to code the sign.



**Figure 5.1:** Illustration of a piecewise smooth function. (a) A piecewise smooth function  $f(t) = f_p(t) + f_\alpha(t)$ ; (b) coefficients of  $f(t)$ , the high-pass coefficients in the boxes are in the cone of influence of discontinuities represented by the index set  $\mathcal{J}_p$  and the coefficients outside the boxes are in the set  $\mathcal{J}_\alpha$ ; (c) a piecewise polynomial function  $f_p(t)$ ; (d) coefficients of  $f_p(t)$ , the high-pass coefficients in the boxes correspond to the index set  $\mathcal{J}_p$ , which can be predicted using parametric estimation; (e) a smooth  $\alpha$ -Lipschitz function  $f_\alpha(t)$ ; (f) coefficients of  $f_\alpha(t)$ .

In reality, the encoder does not have a direct access to the smooth component  $f_\alpha(t)$  of  $f(t)$ . This is not a problem for the coefficients  $\{d_{j,n}\}_{(j,n) \notin \mathcal{J}_p}$  outside the cone of influence of discontinuities as

$$d_{j,n} = \langle f(t), \psi_{j,n}(t) \rangle = \langle f_\alpha(t), \psi_{j,n}(t) \rangle + \underbrace{\langle f_p(t), \psi_{j,n}(t) \rangle}_{=0} = \langle f_\alpha(t), \psi_{j,n}(t) \rangle, \quad (j, n) \notin \mathcal{J}_p.$$

Hence, the above quantization strategy can be applied directly. For the coefficients  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  in the cone of influence of discontinuities, it follows that

$$d_{j,n} = \langle f(t), \psi_{j,n}(t) \rangle = \langle f_\alpha(t), \psi_{j,n}(t) \rangle + \langle f_p(t), \psi_{j,n}(t) \rangle, \quad (j, n) \in \mathcal{J}_p.$$

Since the coefficients  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  do not decay as  $2^{j(\alpha+1/2)}$  across scales, the values of  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  are outside the range of the quantizer i.e. for the same  $\Delta$ , more than  $R_{j,\alpha}$  bits are required to code the coefficients in the set  $\mathcal{J}_p$ . However, it is true that the information of the wavelet coefficients of  $f_\alpha(t)$  is fully contained within the first  $R_{j,\alpha}$  least significant bits (LSB). Therefore, for the coefficients  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$ , only the first  $R_{j,\alpha}$  LSBs are transmitted to the decoder and the rest of the bits can be discarded.

#### Decoding with Parametric Estimation

As mentioned earlier, the reconstruction of the piecewise polynomial component  $f_p(t)$  can be done parametrically. We recall that the wavelet coefficients of  $f(t)$  is given by  $d_{j,n} = \langle f_\alpha(t), \psi_{j,n}(t) \rangle + \langle f_p(t), \psi_{j,n}(t) \rangle$ ,  $(j, n) \in \mathcal{J}_p$ . Let us denote with  $\hat{f}_p(t)$ , the reconstructed version of  $f_p(t)$ , which is obtained from parametric estimation, and let  $\hat{d}_{j,n} = \langle \hat{f}_p(t), \psi_{j,n}(t) \rangle$  be the corresponding wavelet coefficients. In addition, by following the above quantization strategy, the decoder receives the first  $R_{j,\alpha}$  LSBs of  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  (see (5.10)). We denote with  $\bar{d}_{j,n}$ , the quantized version of  $d_{j,n}$ . We will now show that  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  can be decoded using the concept of error correction code. Let us demonstrate this concept with the following example.

Consider the case where  $R_{J,\alpha} = 2$  bits for a given  $J$ . Suppose that the binary representation of  $\bar{d}_{J,n}$  is 1111 and that, at the decoder, parametric estimation gives  $\hat{d}_{J,n}$  that is represented with 1110. The encoder sends the first  $R_{J,\alpha}$  LSBs of  $\bar{d}_{J,n}$ , which is

11, to the decoder. Since  $d_{J,n} = \langle f_\alpha(t), \psi_{j,n}(t) \rangle + \langle f_p(t), \psi_{j,n}(t) \rangle$ , this means that the quantized version of  $\langle f_\alpha(t), \psi_{j,n}(t) \rangle$  can either take a positive value of 1 or a negative value of 111 (this is without the sign bit attached), which would lead to  $\bar{d}_{J,n} = 1111$  or  $\bar{d}_{J,n} = 111$  respectively. However, since the decoder knows that the magnitude of  $\langle f_\alpha(t), \psi_{j,n}(t) \rangle$  is bounded by  $R_{J,\alpha} - 1 = 1$  bit, it can select  $\bar{d}_{J,n} = 1111$  as the correct decoded value. This is, in spirit, similar to the use channel coding technique in Wyner-Ziv coding reviewed in Chapter 2, where the first  $R_{J,\alpha}$  bits is the equivalent of the coset.

The proposed ‘semi-parametric’ compression algorithm for a piecewise smooth function can now be outlined as follows:

**Algorithm 1** : *Semi-parametric compression algorithm.*

Encoding process:

1. ***N-term linear approximation***: the encoder approximates  $f(t)$  as shown in (5.9);
2. ***Quantization***: the coefficients  $\{c_{J,n}\}$  and  $\{d_{j,n}\}_{J-J_N+1 \leq j \leq J}$  are quantized using a linear approximation-based quantization strategy as discussed in this section to obtain the quantized coefficients  $\{\bar{c}_{J,n}\}$  and  $\{\bar{d}_{j,n}\}_{J-J_N+1 \leq j \leq J}$ .

Decoding process:

1. ***Parametric estimation***: the decoder approximates  $f_p(t)$  by estimating the locations  $\{t_i\}_{0 \leq i \leq K+1}$  and the polynomial coefficients  $\{a_{r,k}\}_{0 \leq r \leq \lfloor \alpha \rfloor, 0 \leq k \leq K}$  of  $f_p(t)$  from the received quantized coefficients  $\{\bar{c}_{J,n}\}$  and  $\{\bar{d}_{j,n}\}_{J-J_N+1 \leq j \leq J}$  to obtain  $\hat{f}_p(t)$ ;
2. ***Cone of influence prediction***: the coefficients  $\{d_{j,n}\}_{-\infty < j \leq J, (j,n) \in \mathcal{I}_p}$  in the cone of influence are predicted as follows:

$$\hat{d}_{j,n} = \langle \hat{f}_p(t), \psi_{j,n}(t) \rangle, \quad j = -\infty, \dots, J - J_N, \quad (j, n) \in \mathcal{I}_p;$$

3. ***Error correction decoding***: the decoder uses the received  $R_{j,\alpha}$  LSBs of quantized coefficients  $\bar{d}_{j,n}$  together with the predicted coefficients  $\hat{d}_{j,n}$  to decode  $\{d_{j,n}\}_{J-J_N+1 \leq j \leq J, (j,n) \in \mathcal{I}_p}$ , where we denote with  $\tilde{d}_{j,n}$ , the decoded version of  $d_{j,n}$ ;
4. ***Final reconstruction***:  $f(t)$  is reconstructed from the inverse wavelet transform of the following set of coefficients:

$$\{\bar{c}_{J,n}\}, \quad \{\bar{d}_{j,n}\}_{J-J_N+1 \leq j \leq J, (j,n) \notin \mathcal{I}_p}, \quad \{\tilde{d}_{j,n}\}_{J-J_N+1 \leq j \leq J, (j,n) \in \mathcal{I}_p} \quad \text{and} \quad \{\hat{d}_{j,n}\}_{-\infty < j \leq J - J_N}.$$

By using parametric estimation, the decoder is able to *predict* the coefficients in the cone of influence from the reconstructed function  $\hat{f}_p(t)$  (see Figure 5.1 (c) and (d)). Moreover, the encoder in Algorithm 1 is low in complexity as it is based on linear wavelet approximation. Thus, there is no need to employ any sorting algorithm or transmit the locations of the coefficients.

## 5.4 Cramér-Rao Bound of Parametric Estimation

In this section, we assess the efficiency of using scaling and wavelet coefficients in parametric estimation, which is a core element of the decoding in Algorithm 1. In particular, we study the Cramér-Rao Bound (CRB), which is the bound on the error of parametric estimation.

### 5.4.1 Derivation of Cramér-Rao Bound

Let us begin by deriving the CRB for our problem. Given a function  $f(\Theta, t)$  where  $\Theta = (\theta_1, \theta_2, \dots, \theta_K)^T$  is a vector of  $K$  deterministic parameters, the CRB provides us the lower bound on the variance of any unbiased estimator i.e.

$$CRB(\Theta) \leq E \left[ \left( \hat{\Theta} - \Theta \right) \left( \hat{\Theta} - \Theta \right)^T \right], \quad (5.11)$$

where  $\hat{\Theta}$  is obtained from any unbiased estimation procedure. The CRB can be calculated from the inverse of the Fisher Information Matrix  $I(\Theta)$  as

$$CRB(\Theta) = I^{-1}(\Theta) = \left( E \left[ \nabla l(\Theta) \nabla l(\Theta)^T \right] \right)^{-1},$$

where  $l(\Theta)$  is the log-likelihood function and  $\nabla = \left( \frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \dots, \frac{\partial}{\partial \theta_K} \right)$ .

We now consider the problem of estimating  $\Theta$  from a set of noisy transform coefficients. Let  $\mathcal{J}_L$  denote the index set of the coefficients that are received by the estimator. The following result can be obtained [12].

**Theorem 16 :** *Consider a function  $f(\Theta, t) \in \mathbf{L}_2([0, 1])$  and a set of noisy measurement*

$\{\hat{y}_n\}_{n \in \mathcal{J}_L}$  given by

$$\hat{y}_n = y_n + \epsilon_n = y_n(f(\Theta, t)) + \epsilon_n, \quad n \in \mathcal{J}_L, \quad (5.12)$$

where  $\epsilon_n$  is i.i.d. additive Gaussian noise with zero mean and variance  $\sigma_\epsilon^2$ . The Cramér-Rao lower bound  $CRB(\Theta)$  of any unbiased estimator that uses  $\{\hat{y}_n\}_{n \in \mathcal{J}_L}$  to estimate  $\Theta$  is as follows:

$$CRB(\Theta) = \sigma_\epsilon^2 \left( \sum_{n \in \mathcal{J}_L} \nabla y_n \nabla y_n^T \right)^{-1}, \quad (5.13)$$

where  $\nabla y_n = \left( \frac{\partial y_n}{\partial \theta_1}, \frac{\partial y_n}{\partial \theta_2}, \dots, \frac{\partial y_n}{\partial \theta_K} \right)$ .

**Proof:** Since the coefficients  $\{y_n\}_{n \in \mathcal{J}_L}$  are also dependent on  $\Theta$ , we can apply the formula for change of variable to obtain

$$p_{\hat{y}}(\hat{y}_n | \Theta) = p_\epsilon(\hat{y}_n - y_n). \quad (5.14)$$

Using the fact that  $\epsilon_n$  is i.i.d. Gaussian noise together with (5.14), we can express the log-likelihood function  $l(\Theta)$  as:

$$l(\Theta) = \ln P(\{\hat{y}_n\}_{n \in \mathcal{J}_L} | \Theta) = \ln \prod_{n \in \mathcal{J}_L} p_{\hat{y}}(\hat{y}_n | \Theta) = \sum_{n \in \mathcal{J}_L} \ln p_\epsilon(\hat{y}_n - y_n). \quad (5.15)$$

It then follows that the partial derivative of  $l(\Theta)$  with respect to  $\theta_i$  is given by

$$\begin{aligned} \frac{\partial l(\Theta)}{\partial \theta_i} &= \frac{\partial l(\Theta)}{\partial (\hat{y}_n - y_n)} \frac{\partial (\hat{y}_n - y_n)}{\partial y_n} \frac{\partial y_n}{\partial \theta_i} \\ &\stackrel{(a)}{=} - \sum_{n \in \mathcal{J}_L} \frac{p'_\epsilon(\hat{y}_n - y_n)}{p_\epsilon(\hat{y}_n - y_n)} \frac{\partial y_n}{\partial \theta_i} \\ &\stackrel{(b)}{=} \sigma_\epsilon^{-2} \sum_{n \in \mathcal{J}_L} \epsilon_n \frac{\partial y_n}{\partial \theta_i}, \end{aligned}$$

where (a) follows from (5.15) and (b) from the fact that

$$p_\epsilon(\epsilon_n) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp\left(-\frac{\epsilon_n^2}{2\sigma_\epsilon^2}\right).$$

Hence, we have that  $\nabla l(\Theta) = \sigma_\epsilon^{-2} \sum_{n \in \mathcal{J}_L} \epsilon_n \nabla y_n$ . The Fisher Information

Matrix can now be determined as

$$\begin{aligned}
 I(\Theta) &= E [\nabla l(\Theta) \nabla l(\Theta)^T] \\
 &= \sigma_\epsilon^{-4} E [\sum_n \sum_m \epsilon_n \epsilon_m \nabla y_n \nabla y_m^T] \\
 &= \sigma_\epsilon^{-4} \sum_n \sum_m E [\epsilon_n \epsilon_m] \nabla y_n \nabla y_m^T \\
 &= \sigma_\epsilon^{-2} \sum_n \sum_m \delta_{n,m} \nabla y_n \nabla y_m^T \\
 &= \sigma_\epsilon^{-2} \sum_n \nabla y_n \nabla y_n^T.
 \end{aligned} \tag{5.16}$$

By finding the inverse of  $I(\Theta)$ , we obtain the expression for the  $CRB(\Theta)$  in (5.13), which concludes the proof.  $\blacksquare$

#### 5.4.2 Evaluation of Cramér-Rao Bound

In order to gain some intuition, we consider the following simplified estimation problem. The parametric function of interest  $s(t)$  is assumed to be piecewise constant with a single discontinuity:

$$s(t) = \begin{cases} 0 & t < t_0, \\ A & t_0 \leq t < 1. \end{cases} \tag{5.17}$$

The estimator then has to retrieve  $\Theta = (t_0, A)^T$  from a set of noisy coefficients of  $s(t)$ . For simplicity, we assume that the estimator only receives noisy versions of the  $L$  scaling coefficients of  $s(t)$ . More precisely, the estimator receives

$$\hat{y}_n = y_n + \epsilon_n = \langle s(t), \varphi_{J,n}(t) \rangle + \epsilon_n, \quad n = 0, 1, \dots, L-1.$$

By applying the formula in (5.13), we have that  $CRB(\Theta) = \sigma_\epsilon^2 J_{t_0,A}$ , where  $J_{t_0,A}$  is

$$J_{t_0,A} = \begin{bmatrix} \sum_{n=0}^{L-1} \left( \frac{\partial y_n}{\partial t_0} \right)^2 & \sum_{n=0}^{L-1} \frac{\partial y_n}{\partial t_0} \frac{\partial y_n}{\partial A} \\ \sum_{n=0}^{L-1} \frac{\partial y_n}{\partial A} \frac{\partial y_n}{\partial t_0} & \sum_{n=0}^{L-1} \left( \frac{\partial y_n}{\partial A} \right)^2 \end{bmatrix}^{-1}. \tag{5.18}$$

It, therefore, follows that the CRBs for the estimation of the location  $t_0$  and the amplitude  $A$  of  $s(t)$  are given by

$$CRB(t_0) = \sigma_\epsilon^2 (J_{t_0,A})_{11} \quad \text{and} \quad CRB(A) = \sigma_\epsilon^2 (J_{t_0,A})_{22} \tag{5.19}$$



where  $(J_{t_0,A})_{ij}$  denotes the entry in the  $i$ -th row and  $j$ -th column of the matrix  $J_{t_0,A}$ .

### B-spline scaling functions and parametric estimation

Let us now examine the above CRBs for an important family of scaling function, which is the B-spline family. A B-spline function of order  $P \geq 0$  is given by

$$\beta(t) = \frac{1}{P!} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l (t-l)_+^P, \quad (5.20)$$

where

$$(t)_+^P = \begin{cases} 0 & t < 0, \\ t^P & t \geq 0. \end{cases}$$

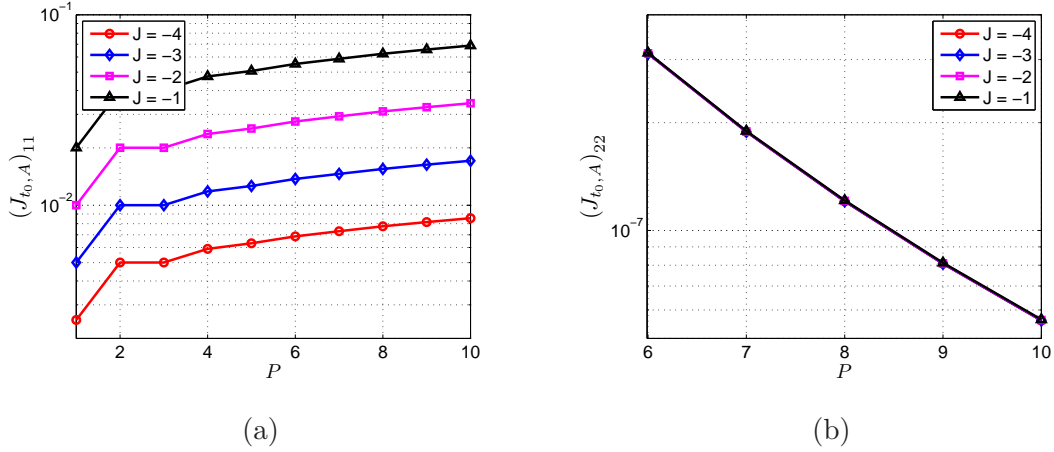
The coefficients are, therefore, given by

$$y_n = \langle s(t), \beta_{J,n}(t) \rangle, \quad J < 0, \quad (5.21)$$

where  $\beta_{J,n}(t) = 2^{-J/2} \beta(2^{-J}t - n)$ . Note that, in reality, the wavelet transforms with B-spline scaling function of order one or higher are not orthogonal and the scaling coefficients would be given by  $\langle s(t), \tilde{\beta}_{J,n}(t) \rangle$  where  $\tilde{\beta}$  is the dual of  $\beta$ . Nevertheless, the results shown in this section can still give us a good understanding of the estimation of the step function.

It is relatively straight forward to evaluate equations (5.18) and (5.19). The full evaluation can be found in the Appendix A.2. Figure 5.2 shows the plots of the values of  $(J_{t_0,A})_{11}$  and  $(J_{t_0,A})_{22}$  (from (5.18)) against the order of the B-spline function at resolution  $2^J$ . Note that higher  $(J_{t_0,A})_{11}$  and  $(J_{t_0,A})_{22}$  translate to larger errors in the estimation of  $t_0$  and  $A$  respectively.

It is clear that the estimation of  $t_0$  improves with decreasing order of B-spline function. Intuitively, this is because reducing the order of a B-spline by one decreases its support size by  $2^{-J}$ . Thus, the discontinuity is captured more accurately with fewer number of coefficients. For the estimation of  $A$ , however, the opposite is true as the accuracy improves with increasing order of B-spline function since larger support allows a single coefficient to capture a longer observation of  $A$ . Finally, we can see that by lower-



**Figure 5.2:** Evaluation of  $J_{t_0,A}$  in (5.18) using B-spline scaling functions of order  $P$ , where  $0 < P \leq 10$ . (a) Plots of  $(J_{t_0,A})_{11}$  against  $P$ . (b) Plots of  $(J_{t_0,A})_{22}$  against  $P$ .

ing  $J$ , which has the effect of increasing the resolution and the total number of low-pass coefficients ( $L = 2^{-J}$ ), we obtain better estimates of both parameters. Interestingly, this improvement is much less visible for the estimation of  $A$ .

## 5.5 Distortion-Rate Analysis

We now give the derivation of the distortion-rate bound  $D(R)$  of our proposed semi-parametric compression scheme outlined in Algorithm 1. For simplicity, in the analysis that follows, we assume that the function to be compressed  $f(t) \in \mathbf{L}_2([0, 1])$  is piecewise smooth with one discontinuity and that each piece is  $\alpha$ -Lipschitz with  $0 \leq \alpha < 1$ . From our signal model shown in (5.6), it follows that the piecewise polynomial component  $f_p(t)$  is in fact a piecewise constant function. We can, therefore, write this simplified piecewise smooth function  $f(t)$  as

$$f(t) = s(t) + f_\alpha(t), \quad (5.22)$$

where  $s(t)$  is the step function as shown in (5.17). In addition, we also assume the decoder uses  $L$  low-pass coefficients to estimate  $t_0$  and  $A$ . These assumptions, however, do not change the overall behavior of the distortion-rate bound in terms of the decay rate.

The following analysis uses the fact that the decoder described in Algorithm 1 essentially reconstructs  $s(t)$  and  $f_\alpha(t)$  separately. Therefore, the total distortion  $D$  can be written with the following sum:

$$D = D_\alpha + D_s, \quad (5.23)$$

where  $D_\alpha$  and  $D_s$  are the distortion from the reconstruction of the smooth function  $f_\alpha(t)$  and the step function  $s(t)$  respectively. Since  $f_\alpha(t)$  is uniformly  $\alpha$ -Lipschitz smooth and given that the wavelet basis has at least  $\lfloor \alpha + 1 \rfloor$  vanishing moments, an encoder in Algorithm 1 whose rate allocation follows the  $N$ -term wavelet linear approximation strategy achieves

$$D_\alpha(R_1) \leq c_7 R_1^{-2\alpha}, \quad (5.24)$$

where  $R_1$  is the total rate (in bits) allocated for the compression of  $f_\alpha(t)$ . The quantization strategy has previously been discussed in details in Section 5.3.2. For the coefficients in the cone of influence of discontinuities, it is true that the information of the wavelet coefficients of  $f_\alpha(t)$  is fully contained within the first  $R_{j,\alpha}$  least significant bits. Therefore, any additional distortion has an effect on  $\langle f_p(t), \psi_{j,n}(t) \rangle$  only and is included in  $D_s$ . We can, therefore, conclude that (5.24) holds inside the cone of influence as well. Our next step then is to derive  $D_s$ .

### Quantized coefficients representation

The low-pass coefficients of  $f(t)$  can be written as follows:

$$\begin{aligned} y_n &= \langle f(t), \varphi_{J,n}(t) \rangle \\ &\stackrel{(a)}{=} \langle s(t), \varphi_{J,n}(t) \rangle + \langle f_\alpha(t), \varphi_{J,n}(t) \rangle \\ &= y_n^s + y_n^\alpha, \end{aligned}$$

where (a) follows from (5.6) and the linearity of the inner product. We can then write the quantized coefficients as

$$\bar{y}_n = y_n + \epsilon_n^q = y_n^s + y_n^\alpha + \epsilon_n^q = y_n^s + \epsilon_n^s, \quad (5.25)$$

where  $\epsilon_n^q$  represents the quantization noise, which is assumed to be additive and Gaussian. Thus, we have written the quantized coefficients  $\bar{y}_n$  as the sum of the coefficients of the step function  $y_n^s$  and the *noise* term  $\epsilon_n^s = \epsilon_n^q + y_n^\alpha$ .

Suppose that a uniform scalar quantizer is used, at high rates, it follows that the variance  $\sigma_q^2$  of the quantization noise  $\{\epsilon_n^q\}_{0 \leq n \leq L-1}$  is given by

$$\sigma_q^2 = C 2^{-2 \frac{R_2}{L}}, \quad (5.26)$$

where  $C$  is a constant and  $R_2$  is the total rate allocated to represent  $\{y_n\}_{0 \leq n \leq L-1}$ .

Let us also make the following assumptions for the computation of  $D_s$ :

- the probability density function (PDF) of  $y_n^\alpha$  is zero-mean Gaussian<sup>1</sup> with variance  $\sigma_\alpha^2$ ,
- both  $\epsilon_n^q$  and  $y_n^\alpha$  are independent, which implies  $\epsilon_n^s$  is Gaussian distributed with zero mean and variance  $\sigma_\epsilon^2$ , where

$$\sigma_\epsilon^2 = (\sigma_q^2 + \sigma_\alpha^2), \quad (5.27)$$

- finally, we assume that the estimators of  $t_0$  and  $A$  are unbiased minimum variance estimators that achieve the CRBs.

### Distortion from parametric estimation

We can now derive the distortion  $D_s$  of the reconstructed step function. The reconstructed step function  $\hat{s}(t)$  can be written as

$$\hat{s}(t) = \begin{cases} 0 & t < t_0 + \epsilon_t, \\ A + \epsilon_A & t_0 + \epsilon_t \leq t \leq 1. \end{cases}$$

<sup>1</sup>The PDF of  $y_n^\alpha$  and  $\epsilon_n^q$  are arbitrarily assumed to be zero-mean Gaussian as this allows us to use the analytical expression of the CRB, given by (5.13). The derived distortion-rate bound is then verified with simulations.

Here, the errors in the estimation of  $t_0$  and  $A$  are represented by  $\epsilon_t$  and  $\epsilon_A$ . It then follows that the average distortion  $D_s = \text{MSE}(s(t) - \hat{s}(t))$  is given by

$$D_s = \mathbb{E} \left[ \int (s(t) - \hat{s}(t))^2 dt \right] = \mathbb{E} \left[ \int_{t_0}^{t_0+|\epsilon_t|} A^2 dt + \int_{t_0+\epsilon_t}^1 \epsilon_A^2 dt \right] = \mathbb{E} [A^2|\epsilon_t| + c_\tau \epsilon_A^2 - \epsilon_A^2 \epsilon_t]$$

with a constant  $0 \leq c_\tau \leq 1$ . By assuming that  $\epsilon_t$  and  $\epsilon_A$  are independent, we have that

$$D_s = \mathbb{E} [A^2|\epsilon_t|] + \mathbb{E} [c_\tau \epsilon_A^2] - \underbrace{\mathbb{E} [\epsilon_A^2] \mathbb{E} [\epsilon_t]}_{=0} = A^2 \mathbb{E} [|\epsilon_t|] + c_\tau \mathbb{E} [\epsilon_A^2],$$

where  $\mathbb{E} [|\epsilon_t|]$  is the mean absolute deviation of  $\epsilon_t$ .

We now denote with  $\sigma_t^2$  and  $\sigma_A^2$  the variances of  $\epsilon_t$  and  $\epsilon_A$  respectively. Our assumption that the estimators are unbiased minimum variance estimators means that both  $\sigma_t^2$  and  $\sigma_A^2$  are given by their respective CRBs shown in (5.19) where, from (5.25),  $\sigma_t^2$  is given by (5.27). Using Jensen's inequality for concave functions<sup>1</sup>, we have that

$$\mathbb{E} [|\epsilon_t|] = \mathbb{E} \left[ \sqrt{(\epsilon_t - \mathbb{E}[\epsilon_t])^2} \right] \leq \sqrt{\mathbb{E} [(\epsilon_t - \mathbb{E}[\epsilon_t])^2]} = \sigma_t = \sqrt{\text{CRB}(t_0)}$$

as  $\mathbb{E}[\epsilon_t] = 0$ . Clearly,  $\mathbb{E} [\epsilon_A^2] = \sigma_A^2 = \text{CRB}(A)$ . Therefore, the expected distortion can be written as

$$D_s \leq A^2 \sigma_t + \sigma_A^2 = A^2 \sqrt{\text{CRB}(t_0)} + \text{CRB}(A).$$

By using the expression for the CRBs in (5.19) together with the relationship given in (5.25), we obtain the following distortion-rate bound for the estimation of the step function:

$$\begin{aligned} D_s(R_2) &\leq A^2 \sigma_\epsilon (J_{t_0,A})_{11}^{\frac{1}{2}} + \sigma_\epsilon^2 (J_{t_0,A})_{22} \\ &\stackrel{(a)}{=} c_8 (\sigma_q^2 + \sigma_\alpha^2)^{\frac{1}{2}} + c_9 (\sigma_q^2 + \sigma_\alpha^2) \\ &\stackrel{(b)}{=} c_8 \left( c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2 \right)^{\frac{1}{2}} + c_9 \left( c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2 \right), \end{aligned} \tag{5.28}$$

where (a) and (b) follow from substituting in (5.27) and (5.26) respectively. The expression for the total distortion-rate bound can now be obtained by substituting (5.24)

<sup>1</sup>Jensen's inequality: given a random variable  $X$  and a concave function  $f(x)$ , it follows that  $\mathbb{E}[f(X)] \leq f(\mathbb{E}[X])$ .

and (5.28) into (5.23), which gives

$$D(R) \leq c_7 R_1^{-2\alpha} + c_8 \left( c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2 \right)^{\frac{1}{2}} + c_9 \left( c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2 \right), \quad (5.29)$$

where the total rate  $R$  is equal to

$$R = R_1 + R_2. \quad (5.30)$$

### Bit allocation problem

Given the total rate  $R$ , we now need to allocate the bits among  $R_1$  and  $R_2$  so that the distortion in (5.29) is minimized. This is a well known constrained optimization problem, which can be solved using a Lagrange multiplier method. One necessary condition for the optimal bit allocation is that the derivatives of the distortion  $D$  with respect to  $R_1$  and  $R_2$  must be equal i.e.

$$\frac{\partial D}{\partial R_1} = \frac{\partial D}{\partial R_2}. \quad (5.31)$$

First, let us consider the case where the variance of  $\{y_n^\alpha\}$  is negligible i.e.  $\sigma_\alpha^2 \approx 0$ . The distortion-rate function now becomes

$$D(R) \leq c_7 R_1^{-2\alpha} + c_8 \sqrt{c_{10}} 2^{\frac{-R_2}{L}} + c_9 c_{10} 2^{\frac{-2R_2}{L}}. \quad (5.32)$$

By applying the condition given in (5.31) to (5.32) and assuming a high-rate regime, we have that the bits can be approximately allocated as

$$R_2 \approx L(2\alpha + 1) \log_2 R_1 + C' \quad (5.33)$$

and the total rate is then given by

$$R = R_1 + L(2\alpha + 1) \log_2 R_1 + C' \approx R_1. \quad (5.34)$$

From the substitution of (5.33) into (5.32), together with the approximation in (5.34),

we have that the total distortion is

$$D(R) \leq c_7 R^{-2\alpha} + c_{11} R^{-(2\alpha+1)} + c_{12} R^{-2(2\alpha+1)}. \quad (5.35)$$

From (5.35), we can see that both terms  $c_{11} R^{-(2\alpha+1)}$  and  $c_{12} R^{-2(2\alpha+1)}$  represent the distortion due to the discontinuity, which decay faster than  $c_7 R^{-2\alpha}$ . Therefore, given that  $\sigma_\alpha^2 \approx 0$ , the distortion-rate curve of our proposed scheme follows  $D(R) \sim R^{-2\alpha}$  at high  $R$ .

If we now consider the case where  $\sigma_\alpha^2 > 0$  and assume that  $c_{10} 2^{\frac{-2R_2}{L}} < \sigma_\alpha^2$ , the distortion given in (5.29) can then be approximated with a Taylor series expansion of the square root term  $\left(c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2\right)^{\frac{1}{2}}$  to obtain

$$D(R) \leq c_7 R_1^{-2\alpha} + c_8 \left( \frac{c_{10} 2^{\frac{-2R_2}{L}}}{2\sigma_\alpha} + \sigma_\alpha \right) + c_9 \left( c_{10} 2^{\frac{-2R_2}{L}} + \sigma_\alpha^2 \right). \quad (5.36)$$

By solving the equal gradient condition in (5.31), where  $D$  is approximately given by (5.36), we obtain the following rate allocation:

$$R_2 = \frac{L}{2} (2\alpha + 1) \log_2 R_1 + C' \quad (5.37)$$

with a constant  $C'$ . In the high-rate regime (high  $R$ ), the total rate  $R$  can be approximated by

$$R = R_1 + \frac{L}{2} (2\alpha + 1) \log_2 R_1 + C' \approx R_1. \quad (5.38)$$

Therefore, by substituting (5.37) into (5.36) and using the approximation shown in (5.38), the overall distortion-rate function of our semi-parametric compression scheme is

$$D(R) \leq c_7 R^{-2\alpha} + c_{11} R^{-(2\alpha+1)} + (c_8 \sigma_\alpha + c_9 \sigma_\alpha^2). \quad (5.39)$$

Note that the term  $c_{13} R^{-(2\alpha+1)}$  now represents the distortion caused by the discontinuity, which still decays faster than the distortion from the encoding of the smooth function in our scheme. The results of this distortion-rate analysis can be summarized as follows:

**Summary 1** Consider a piecewise smooth function  $f(t)$  with a single discontinuity and two  $\alpha$ -Lipschitz pieces, where  $0 \leq \alpha < 1$ . The semi-parametric compression of  $f(t)$  in Algorithm 1, which employs a linear approximation strategy at the encoder and reconstructs the piecewise constant component  $f_p(t)$  and the uniformly smooth component  $f_\alpha(t)$  separately, achieves the following distortion-rate function:

$$D(R) \leq c_7 R^{-2\alpha} + c_{11} R^{-(2\alpha+1)} + c_{12} R^{-2(2\alpha+1)}$$

when the variance of the coefficients of  $f_\alpha(t)$  is close to zero. Otherwise, the achievable distortion-rate function is

$$D(R) \leq c_7 R^{-2\alpha} + c_{11} R^{-(2\alpha+1)} + (c_8 \sigma_\alpha + c_9 \sigma_\alpha^2).$$

Therefore, given that  $\sigma_\alpha^2$  is sufficiently small, the proposed scheme can achieve the dominating decay rate of  $R^{-2\alpha}$  for a wide range of rates. Such performance is comparable to that of a compression scheme based on nonlinear approximation as shown in (5.3). Moreover, the  $D(R)$  curve of a compression scheme based on a linear encoder and a linear decoder is characterized by the much slower decay of  $R^{-1}$  (see (5.2)).

## 5.6 Constructive Compression Algorithms

### 5.6.1 FRI-Based Parametric Estimation Algorithm

This section introduces a practical parametric estimation technique inspired by the recently developed concept of sampling of signals with finite rate of innovation (FRI) (see Chapter 4). It is easy to see that a piecewise polynomial signal with pieces of maximum degree  $\lfloor \alpha + 1 \rfloor$  (as shown in (5.7)) also belongs to this class of functions as there are a finite number of discontinuities and each polynomial piece can be described by at most  $\lfloor \alpha \rfloor$  polynomial coefficients. The sampling scheme for the reconstruction of piecewise polynomial functions from its low-pass coefficients is given in Chapter 4, Section 4.5.3.

Consider a set of noisy quantized low-pass coefficients  $\{\bar{c}_{M,n}\}$  of a piecewise poly-



nomial function  $f_p(t) = \sum_{k=0}^1 \sum_{r=0}^{R-1} a_{k,r}(t - t_k)_+^r$  where  $t_0 = 0$ ,  $t_1 \in ]0, 1[$  and  $\bar{c}_{J,n} = \langle f_p(t), \varphi_{J,n}(t) \rangle + \epsilon_n$ . Assuming that  $\varphi_{J,n}(t)$  reproduces polynomials of maximum order  $P \geq R - 1$ , we now present a FRI-based parametric estimation algorithm for the estimation of  $f_p(t)$ :

**Algorithm 2** : *FRI-based parametric estimation algorithm.*

1. **Finite difference**: the  $R$ -order finite difference of  $\{\bar{c}_{M,n}\}$  is obtained by

$$\bar{z}_n^{(R)} = \bar{z}_{n+1}^{(R-1)} - \bar{z}_n^{(R-1)} \quad \text{with} \quad \bar{z}_n^{(1)} = \bar{c}_{J,n+1} - \bar{c}_{J,n}.$$

2. **Thresholding**: in order to reduce the effect of noise, thresholding is applied as

$$\tilde{z}_n^{(R)} = \begin{cases} \bar{z}_n^{(R)} & \bar{z}_n^{(R)} \geq z_{th} \\ 0 & \text{otherwise.} \end{cases}$$

where  $z_{th}$  is the threshold.

3. **Moments estimation**: the continuous moments of  $f_p^{(R)}(t)$  are estimated as:

$$\tilde{M}_p^{(R)} = \sum_n c_n^{(p)} \tilde{z}_n^{(R)} \quad p = 0, 1, \dots, 2R - 1.$$

4. **Annihilating filter method**: the locations  $t_k$  of the discontinuities are estimated from  $\{\tilde{M}_p^{(R)}\}_{p=0, \dots, 2R-1}$  with the annihilating filter method as shown in (4.13).

5. **Solving Vandermonde system**: the amplitudes of the stream of differentiated Diracs  $f_p^{(R)}(t)$  are estimated by solving the Vandermonde system of equations of  $\{\tilde{M}_p^{(R)}\}_{p=0, \dots, 2R-1}$ , which is derived from the identity in (4.20).

6. **Integration**: the function  $f_p(t)$  is estimated by integrating the reconstructed  $R$ -th derivative of  $f_p(t)$ .

Note that the above algorithm can also be applied to functions with more than two pieces by locally reconstructing  $f_p(t)$  at one discontinuity point at a time. Consequently, a practical semi-parametric compression scheme with a low-complexity encoder can now be constructed from Algorithm 1, where the parametric estimation step is implemented with Algorithm 2. This allows the decoder to approximate the piecewise polynomial function from the quantized low-pass coefficients  $\{\bar{c}_{J,n}\}$ .

### 5.6.2 FRI-based Semi-Parametric Compression with Nonlinear Encoder

We now present an alternative semi-parametric compression scheme that includes a high-complexity encoder, which uses both a parametric estimation algorithm and a non-linear approximation based coding strategy, as shown below. Such scheme can be adopted when the signal of interest does not exactly fit the model described in (5.6) but is still well approximated by it. Hence, the use of a parametric estimation algorithm at the encoder exploits the fact that the high-pass coefficients in the cone of influence can be estimated from the low-pass coefficients. The parametric estimation step can be considered as an additional step to the existing coding strategy. Simulation results of the algorithm below can be found in Section 5.7.

**Algorithm 3** : *Semi-parametric compression with nonlinear encoder.*

#### Encoding

1. **Nonlinear encoding:** the encoder quantizes the  $N$  largest coefficients to obtain

$$\{\bar{c}_{J,n}\}_{n \in \mathcal{J}_N} \quad \text{and} \quad \{\bar{d}_{j,n}\}_{(j,n) \in \mathcal{J}_N}$$

with  $\mathcal{J}_N$  denoting a set of indices of the  $N$  largest coefficients.

2. **Parametric estimation:** the encoder estimates  $f_p(t)$  with  $\{\bar{c}_{J,n}\}_{n \in \mathcal{J}_N}$  using Algorithm 2 to obtain  $\hat{f}_p(t)$ .
3. **Cone of influence prediction:** the coefficients in the cone of influence  $\{d_{j,n}\}_{(j,n) \in \mathcal{J}_p}$  are predicted as follows:

$$\hat{d}_{j,n} = \langle \hat{f}_p(t), \psi_{j,n}(t) \rangle, \quad (j,n) \in \mathcal{J}_p.$$

4. **Residual calculation:** the residual is calculated as

$$\tilde{d}_{j,n} = d_{j,n} - \hat{d}_{j,n}, \quad (j,n) \in \mathcal{J}_p$$

5. **Encoding mode decision:** if  $\sum_{(j,n) \in \mathcal{J}_p} |\tilde{d}_{j,n}|^2 < \sum_{(j,n) \in \mathcal{J}_p} |d_{j,n}|^2$ , then the encoder creates a new set of  $N$ -largest coefficients,  $\mathcal{J}_N^*$ , from  $\{c_{J,n}\}$ ,  $\{d_{j,n}\}_{(j,n) \notin \mathcal{J}_p}$  and  $\{\tilde{d}_{j,n}\}_{(j,n) \in \mathcal{J}_p}$ , which are re-quantized and transmitted. Else, the quantized coefficients in step 1 are transmitted.

#### Decoding

**Synchronized decoding:** if parametric estimation is required, the decoder repeats step 2 and step 3, and reconstructs  $f(t)$  as

$$\{\bar{c}_{J,n}\}_{n \in \mathcal{I}_N^*}, \quad \{\bar{d}_{j,n}\}_{(j,n) \in \mathcal{I}_N^* - \mathcal{I}_p} \quad \text{and} \quad \{\bar{\bar{d}}_{j,n} + \hat{d}_{j,n}\}_{(j,n) \in \mathcal{I}_p}.$$

Else, a standard procedure is used without the parametric estimation steps.

## 5.7 Simulation Results

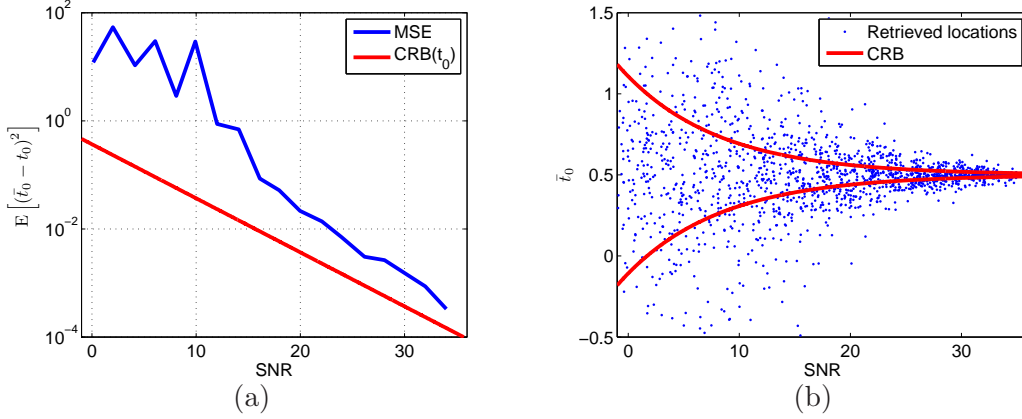
### 5.7.1 Parametric Estimation Algorithm

Let us start by comparing the variance of an estimator that uses Algorithm 2, which is based on the concept of sampling of FRI signals, against the CRB described in Section 5.4.2. In this simulation, the coefficients  $\{y_n\}$  of the step function  $s(t)$  given by (5.17) are obtained as  $y_n = \{\langle s(t), \varphi_{J,n}(t) \rangle\}$  with  $J = 6$ , where  $\varphi(t)$  is the first order B-spline scaling function. Gaussian noise with variance  $\sigma_\epsilon^2$  is then added to  $\{y_n\}$ . The values of the amplitude  $A$  and the location  $t_0$  are estimated using Algorithm 2.

Figure 5.3 shows the plot of the MSE of the estimation  $E[(\bar{t}_0 - t_0)^2]$  in comparison with the corresponding CRB (see Section 5.4.2) together with the plot of the retrieved locations. Note that the signal-to-noise ratio (SNR) is calculated as  $10 \log_{10} \left( \frac{\text{var}[y_n]}{\sigma_\epsilon^2} \right)$  where  $\text{var}[y_n]$  is the variance of  $\{y_n\}$ . We can see that our proposed algorithm exhibits the same decay as the CRB when the SNR reaches approximately 15dB even though the estimator does not achieve the lower bound. The plots for the estimation of the amplitude  $A$  are shown in Figure 5.4. Note that the MSE in the estimation of  $A$  follows the CRB even at low SNR.

### 5.7.2 Semi-Parametric Compression Algorithm

In this section, we present the simulation results of the semi-parametric compression scheme described in Algorithm 1 where the parametric estimation step is implemented with Algorithm 2. The simulation results of the simplified signal model in (5.22) are presented first, where the function  $f(t)$ ,  $t \in [0, 1]$ , is made of a step function and a smooth  $\alpha$ -Lipschitz function with  $\alpha = 1$ . A ten-level wavelet transform with a first order B-spline scaling function was used to decompose  $f(t)$ . Note that in all our simulation,

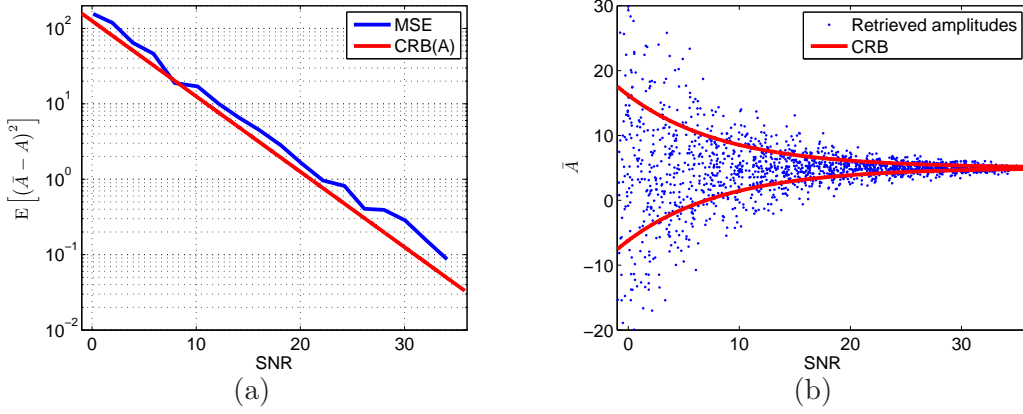


**Figure 5.3:** Error in the estimation of  $t_0$ . (a) MSE in the estimation of  $t_0$  using Algorithm 2 in comparison with the corresponding CRB. (b) retrieved locations  $\bar{t}_0$  using Algorithm 2 and the corresponding CRB.

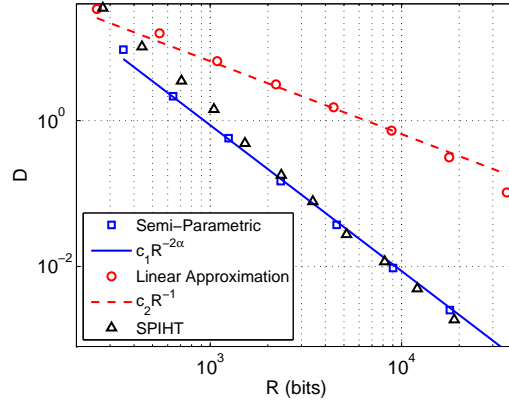
the function  $f(t)$  is generated by adding the piecewise polynomial component  $f_p(t)$  directly to the smooth component  $f_\alpha(t)$ . The smooth function  $f_\alpha(t)$  is generated in the wavelet domain as follows: first, a set of coefficients  $\{d_{J,n}\}$  at the coarsest decomposition level is generated with a random number generator; the rest of the coefficients at  $j$ -th decomposition level are then created by scaling the maximum value of the random number generator as  $d_{j,n} \sim A2^{j(\alpha+1/2)}$  where  $A = \max_n |d_{J,n}|$ .

Figure 5.5 shows the distortion-rate plot of our proposed semi-parametric compression scheme in comparison with the  $D(R)$  curves of a linear approximation-based scheme and a non-linear approximation-based scheme, which is a 1-D version of the SPIHT algorithm [43]. The tree structure of the SPIHT algorithm can simply be adjusted to work with 1-D signals. Our scheme achieves a decay rate of  $R^{-2\alpha} = R^{-2}$ , which is in line with our analysis of Section 5.5, and the performance is comparable with SPIHT. In contrast, the distortion of a linear approximation based scheme decays as  $R^{-1}$ . The reconstructed functions from the two compression schemes are illustrated in Figure 5.6. Note that, in this simulation, the term  $\sigma_\alpha^2$  is insignificant.

Let us now consider the case where  $\sigma_\alpha^2$  is not negligible and its effect is visible in the  $D(R)$  curve. Figure 5.7 shows the  $D(R)$  plots of our semi-parametric compression scheme for a piecewise smooth function, which consists of a step function and a uniformly smooth part with  $\alpha = 0.95$ . The variance  $\sigma_\alpha^2$  of  $c_{J,n} = \langle f_\alpha(t), \varphi_{J,n}(t) \rangle$  causes the



**Figure 5.4:** Error in the estimation of  $A$ . (a) MSE in the estimation of  $A$  using Algorithm 2 in comparison with the corresponding CRB. (b) retrieved amplitudes  $\bar{A}$  using Algorithm 2 and the corresponding CRB.



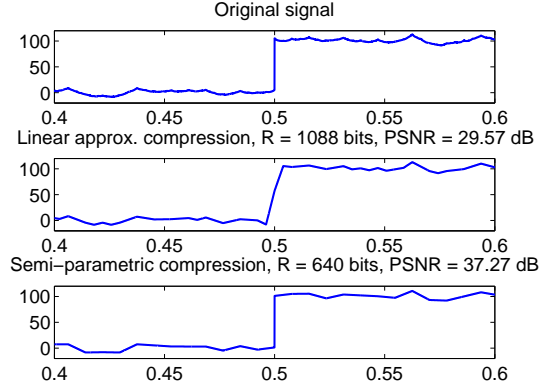
**Figure 5.5:** Distortion-rate plots (log scale) for the compression of piecewise smooth function with one discontinuity and  $\alpha = 1$ . The proposed semi-parametric compression scheme achieves the decay rate of  $R^{-2\alpha}$  whereas the distortion of a linear approximation based scheme decayed as  $R^{-1}$ .

behavior of the  $D(R)$  curve to change from  $D(R) \sim R^{-2\alpha}$  to

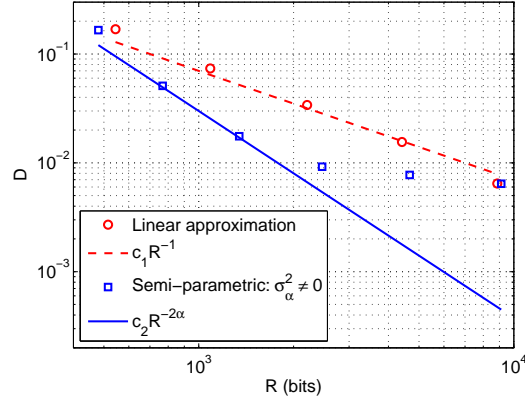
$$D(R) \sim C_1 \sigma_\alpha + C_2 \sigma_\alpha^2 \quad (5.40)$$

after a certain rate point. The value of  $C_1$  and  $C_2$  in (5.40) depend largely on the performance of the parametric estimation algorithm. Thus, a better parametric estimation algorithm allows a wider range of  $R$  where  $D(R) \sim R^{-2\alpha}$ .

The simulation results for the compression of a piecewise smooth function  $f(t)$ ,  $t \in [0, 1]$  that was generated with the signal model described by (5.6) are given next. Here,  $f_p(t)$  is a piecewise quadratic function with three pieces and  $f_\alpha(t)$  has the degree



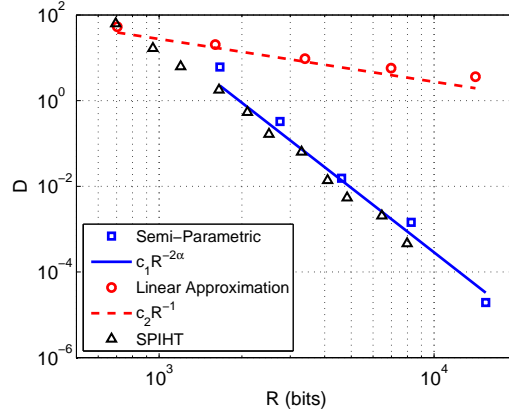
**Figure 5.6:** Plots of the original signal, the reconstructed signals with a linear approximation based scheme and a semi-parametric scheme. The original function is piecewise smooth with one discontinuity and  $\alpha = 1$



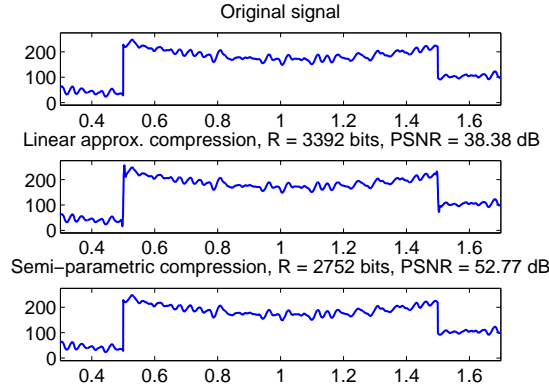
**Figure 5.7:** Distortion-rate plots (log scale) for the compression of piecewise smooth function with one discontinuity and  $\alpha = 0.95$ . The variance  $\sigma_\alpha^2$  of  $c_{J,n} = \langle f_\alpha(t), \varphi_{J,n}(t) \rangle$  causes the behavior of the  $D(R)$  curve to change from  $D(R) \sim R^{-2\alpha}$  to  $D(R) \sim C\sigma_\alpha^2$  after a certain rate point.

of smoothness set to  $\alpha = 2.5$ . We used a six-level wavelet decomposition with a second order B-spline scaling function. The parametric estimation for the two discontinuities in the function are done locally. The  $D(R)$  curve of our proposed scheme are shown together with the plots from a pure linear approximation-based scheme and SPIHT in Figure 5.8. The proposed scheme also achieves a decay rate of  $R^{-2\alpha} = R^{-5}$  in this case with a comparable performance to that of the modified SPIHT algorithm. The reconstructed functions are shown in Figure 5.9. The term  $\sigma_\alpha^2$  is also insignificant in this simulation.

In summary, the simulation results show that the proposed semi-parametric compression scheme is able to achieve the same decay in the  $D(R)$  curve as a compression scheme that employs a non-linear approximation strategy for a wide range of rates.



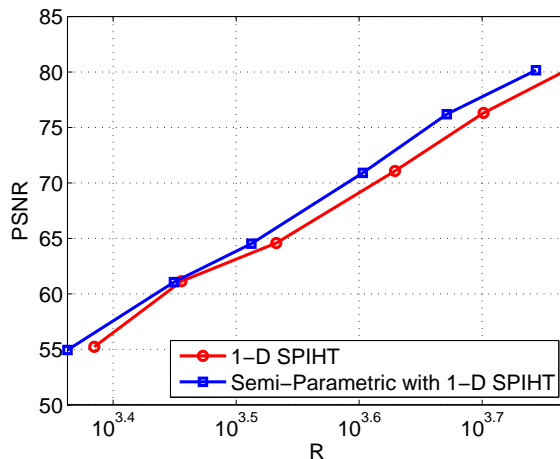
**Figure 5.8:** Distortion-rate plots (log scale) for the compression of piecewise smooth function with two discontinuities and  $\alpha = 2.5$ . The proposed semi-parametric compression scheme achieves the decay rate of  $R^{-2\alpha}$ .



**Figure 5.9:** Plots of the original signal, the reconstructed signals with a linear approximation based scheme and a semi-parametric scheme. The original function is piecewise smooth with two discontinuities and  $\alpha = 2.5$ .

### 5.7.3 Semi-Parametric Compression Algorithm with Nonlinear Encoder

The simulation results of a semi-parametric compression scheme with high-complexity, nonlinear encoder described in Algorithm 3 are presented in this section. In our simulation, we use the 1-D version of SPIHT discussed in Section 5.7.2 to perform nonlinear encoding. A piecewise smooth function  $f(t)$ ,  $t \in [0, 1[$  was then generated with a piecewise quadratic function with four pieces and a smooth  $\alpha$ -Lipschitz function with  $\alpha = 2.75$ . As with the previous simulation, we also use a second order B-spline scaling function with six decomposition levels and the parametric estimation for the three discontinuities are done locally. Figure 5.10 shows the PSNR plots of the proposed semi-parametric compression scheme in comparison with a pure SPIHT encoder. At high rates, our combined scheme achieves a gain of approximately 1-3 dB in the PSNR



**Figure 5.10:** PSNR plots of semi-parametric compression algorithm with nonlinear encoder; at high rates, the combined semi-parametric compression with 1-D SPIHT achieved a gain of approximately 1-3 dB.

over a scheme that uses only the SPIHT encoder.

## 5.8 Conclusion

This chapter introduces a new semi-parametric compression algorithm for piecewise smooth functions. We take advantage of the fact that a piecewise smooth function can be decomposed into two components: a uniformly smooth function and a piecewise polynomial function. The encoder of the proposed algorithm allocates the rate using a linear approximation strategy. The decoding process is instead nonlinear as it reconstructs the piecewise polynomial component of the function with parametric estimation. This allows the decoder to predict the wavelet coefficients in the cone of influence of the discontinuities. This new structure reflects the shift of the computational complexity from the encoder to the decoder. We then showed that the distortion-rate function  $D(R)$  of our proposed scheme achieves a decay of  $R^{-2\alpha}$  at high rates, which is comparable to that of an optimal compression scheme based on nonlinear approximation.

In addition, we proposed a practical method based on FRI theory to estimate the piecewise smooth function from the scaling coefficients of the wavelet decomposition. Another practical semi-parametric compression scheme that employs a nonlinear encoder was also presented. Finally, our simulation results confirm that the proposed



scheme achieves the  $D(R)$  function with a decay of  $R^{-2\alpha}$  at high rates. The simulation of a semi-parametric nonlinear compression also shows that the additional parametric estimation step can improve the overall performance of existing nonlinear compression scheme.

# CHAPTER 6

## Distributed Semi-Parametric Compression of Piecewise Smooth Functions

### 6.1 Introduction

Distributed transform coding of correlated sources has gained much research interest in recent years (see Chapter 2 for a review). A distributed coding scenario requires two or more sources to be independently encoded but jointly decoded. In contrast to the centralized case, joint decoding means that the computational complexity is shifted from the encoder to the decoder. In order to gain new insights into distributed transform coding and its applications, researchers have investigated a number of different setups (see [24, 18, 20, 28, 15]) such as, for example, the distributed KLT for Gaussian sources [20]. With non-Gaussian sources, practical distributed compression schemes change the way transform coefficients are quantized by applying channel coding techniques to the existing centralized transform coding structure [24, 18, 28]. In such cases, more attention is paid to the statistical modeling of the transform coefficients rather than the modeling of the source and the resulting distributed compression schemes are more heuristic in nature.

In this chapter, we investigate the new problem of distributed transform coding of

piecewise smooth functions. This class of functions is commonly used to model, for example, a scan line of an image [66, 11]. Our aim here is to provide precise answers to the following questions: what compression strategies would be suitable in this new scenario; in an ideal case, what are the differences in distortion-rate performance between the distributed, joint and independent compression algorithms. The centralized semi-parametric compression presented in Chapter 5 has already given us some new insights into the distortion-rate performance of a wavelet-based scheme whose computational complexity is shifted from the encoder to the decoder. It was shown that the proposed scheme can still achieve the same decay in the  $D(R)$  function as that of a conventional scheme.

In the next section, we present our model of the signal disparity. The distributed semi-parametric compression strategies based on the disparity models are then proposed in Section 6.3. We then conduct the distortion-rate analysis of the proposed scheme in Section 6.4. Section 6.5 shows the simulation results and, finally, conclusions are drawn in Section 6.6.

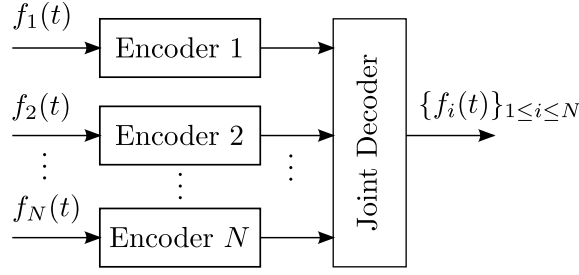
## 6.2 Signal and Disparity Models

In this chapter, we focus on the distributed compression scenario where  $N$  piecewise smooth signals are independently encoded but are decoded jointly as depicted in Figure 6.1. We denote a set of  $N$  functions with  $\{f_i(t)\}_{1 \leq i \leq N}$ , where the subscript  $i$  indicates the signal that is observed by Encoder  $i$ . Here, we also use the same model of piecewise smooth functions given in (5.6). Hence, each piecewise smooth signal  $f_i(t) \in \mathbf{L}_2([0, 1])$  can be written as

$$f_i(t) = f_{i_p}(t) + f_{i_\alpha}(t), \quad i = 1, \dots, N, \quad (6.1)$$

where  $f_{i_p}(t)$  and  $f_{i_\alpha}(t)$  denote a piecewise polynomial function with pieces of maximum degree  $\lfloor \alpha \rfloor$  and a uniformly smooth  $\alpha$ -Lipschitz function.

In the analysis that follows in this chapter, we consider a disparity model where the main difference between the two observed signals,  $f_i(t)$  and  $f_j(t)$ , is described by a



**Figure 6.1:** Distributed compression scenario. A set of  $N$  functions  $\{f_i(t)\}_{1 \leq i \leq N}$  are individually encoded but are jointly decoded.

global shift (or a translation). This disparity model can, therefore, be defined as

$$f_i(t) = f_j(t - \tau_{ij}) + \epsilon_{ij\alpha}(t), \quad i \neq j, \quad i, j \in (1, \dots, N) \quad \text{and} \quad |\tau_{ij}| < 1. \quad (6.2)$$

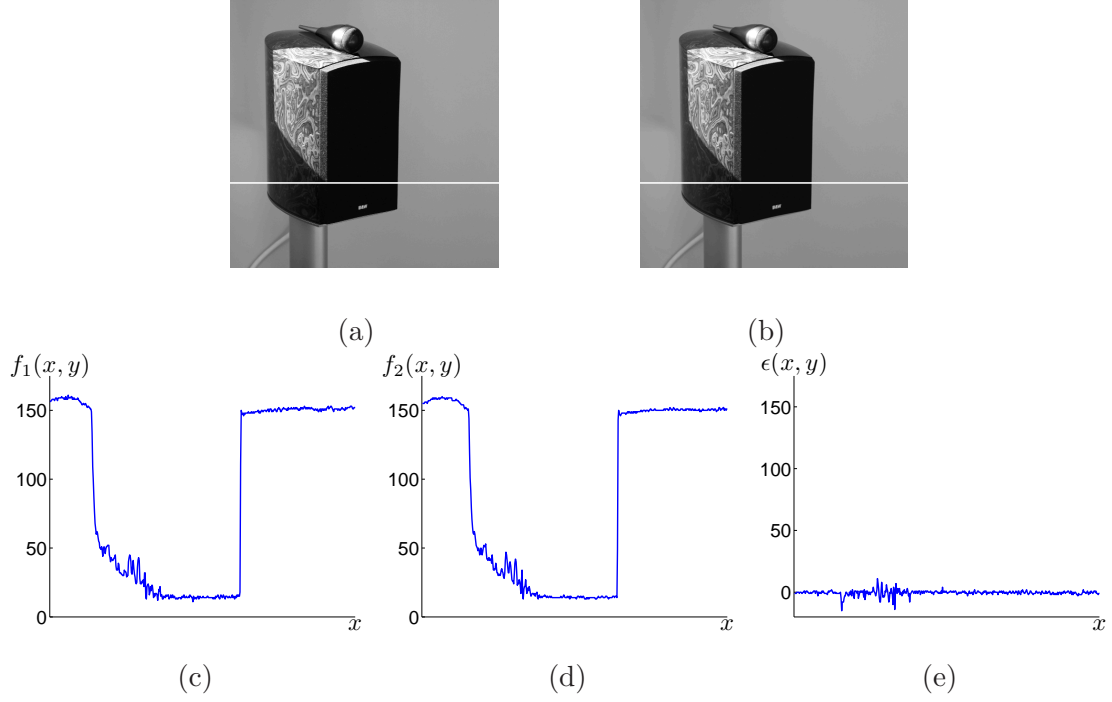
The term  $\epsilon_{ij\alpha}(t)$  represents the prediction error (or the residual), where it is assumed for simplicity that  $\epsilon_{ij\alpha}(t)$  is uniformly  $\alpha$ -Lipschitz. The construction of this model is inspired by applications such as the block-based prediction found in many video compression algorithms, where a set of translation vectors are estimated and transmitted along with the residual. Another example is a set of images captured by an array of cameras, where the disparity between each image is well approximated by a shift parameter. Our aim is then to devise a distributed compression strategy for this disparity model. Figure 6.2 shows an example of the scan lines taken from stereo images and the corresponding prediction error. Note that some regions of the residual shown in Figure 6.2 (c) are not smooth because the image contains texture, which is not well captured by this simplified signal model.

### 6.3 Distributed Semi-Parametric Compression Strategies

Let us now consider a distributed compression strategy for the signal model in (6.2). For simplicity, we write each function  $f_i(t)$  as

$$f_i(t) = f_1(t - \tau_i) + \epsilon_{i\alpha}(t), \quad i = 2, \dots, N, \quad (6.3)$$

where  $\epsilon_{i\alpha}(t)$  is uniformly  $\alpha$ -Lipschitz smooth. Moreover, we assume that a reconstructed version of  $f_1(t)$  is available at the decoder by means of conventional wavelet nonlinear



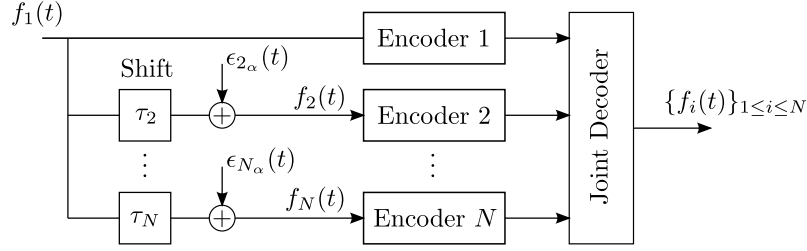
**Figure 6.2:** Examples of the scan lines taken from stereo images shown in (a) and (b). (c) A scan line of the first image  $f_1(x, y')$ ; (d) a scan line from the second image  $f_2(x, y')$ ; (e) the prediction error given by  $f_2(x, y') - f_1(x - \tau_x, y')$ , where  $\tau_x$  denotes the shift parameter.

approximation-based compression strategy. Given that the disparity model between  $f_i(t)$  and  $f_j(t)$  is known a priori,  $f_i(t)$  can be reconstructed by first estimating the shift parameter  $\tau_i$  and then the residual  $\epsilon_{i_\alpha}(t)$  as follows:

$$\hat{f}_i(t) = \hat{f}_1(t - \hat{\tau}_i) + \hat{\epsilon}_{i_\alpha}(t),$$

where  $\hat{f}_i(t)$  and  $\hat{\epsilon}_{i_\alpha}(t)$  denote the reconstructed versions, and  $\hat{\tau}_i$  is the estimated shift parameter. This setup is depicted in Figure 6.3. One of the challenges here is that the Encoder 2 to  $N$  have no access to  $f_1(t)$  and, hence, the prediction error  $\epsilon_{i_\alpha}(t)$  cannot be directly calculated and transmitted. Our goal is then to devise a suitable compression strategy for Encoder 2 to  $N$ .

From the piecewise smooth function model in (6.1), let us define the locations of the discontinuities in  $f_{i_p}(t)$  with  $\{t_{i_k}\}_{1 \leq k \leq K}$  such that  $t_{i_k} = t_{1_k} - \tau_i$ . Therefore, by



**Figure 6.3:** Distributed compression problem setup using the disparity-by-translation model with prediction error. Each function is piecewise smooth and  $f_i(t) = f_1(t - \tau_i) + \epsilon_{i_\alpha}(t)$ ,  $i = 2, \dots, N$ .

retrieving  $\{t_{i_k}\}_{1 \leq k \leq K}$ , the shift parameters  $\tau_i$  can be estimated by taking the average:

$$\hat{\tau}_i = \frac{1}{K} \sum_{k=1}^K (\hat{t}_{1_k} - \hat{t}_{i_k}), \quad i = 2, \dots, N. \quad (6.4)$$

In the following analysis, we will assume that the decoder is able to retrieve the locations  $\{t_{1_k}\}_{1 \leq k \leq K}$ . Hence, the problem of estimating  $\tau_i$  becomes the problem of estimating the locations  $\{t_{i_k}\}_{1 \leq k \leq K}$ ,  $i = 2, \dots, N$ . In addition, we will assume that the decoder only uses  $L$  low-pass coefficients  $c_{i,J,n} = \langle f_i(t), \varphi_{J,n}(t) \rangle$ ,  $n = 0, \dots, L-1$ , of  $f_i(t)$  to estimate  $\{t_{i_k}\}_{1 \leq k \leq K}$ .

Since the reconstructed version of  $f_1(t)$  is available at the decoder, the prediction of  $f_i(t)$  can be formed by  $\tilde{f}_i(t) = \hat{f}_1(t - \hat{\tau}_i)$ . Assuming that the range of the amplitude of  $\epsilon_{i_\alpha}(t)$  can be estimated a priori, we can adopt a similar quantization strategy as the one discussed in the previous chapter, where the encoder only transmits the required least significant bits to the decoder. We, therefore, propose the following semi-parametric distributed compression algorithm:

**Algorithm 4** *A distributed semi-parametric compression scheme.*

Encoding and Decoding of  $f_1(t)$

1. **Nonlinear approximation-based compression:**  $f_1(t)$  is encoded and decoded with a conventional wavelet nonlinear approximation-based compression scheme;
2. **Extracting locations of discontinuities:** the locations  $\{t_{1_k}\}_{k=0, \dots, K}$  are extracted from  $\hat{f}_1(t)$ .

Encoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. ***N-term linear approximation:*** the encoder approximates  $f_i(t)$  as shown in (5.9);
2. ***Quantization:*** the coefficients  $\{c_{i,j,n}\}$  and  $\{d_{i,j,n}\}_{J-J_N+1 \leq j \leq J}$  are quantized using a linear approximation-based quantization strategy as discussed in Section 5.3.2 to obtain  $\{\bar{c}_{i,j,n}\}$  and  $\{\bar{d}_{i,j,n}\}_{J-J_N+1 \leq j \leq J}$ . For  $\{\bar{d}_{i,j,n}\}$ , only the required LSBs are transmitted. The analysis that determines the number of required LSBs will be given in the next section.

Joint Decoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. ***Parametric estimation:*** the decoder estimates the locations  $\{t_{i_k}\}_{k=0,\dots,K}$  from the  $L$  quantized low-pass coefficients  $\{\bar{c}_{i,j,n}\}$  and the shift parameter  $\tau_i$  is calculated using (6.4);
2. ***Prediction by translation:*** a predicted version of  $f_i(t)$  is formed by  $\tilde{f}_i(t) = \hat{f}_1(t - \hat{\tau})$  and the coefficients  $\{\tilde{d}_{i,j,n} = \langle \tilde{f}_i(t), \psi_{j,n}(t) \rangle\}_{J-J_N+1 \leq j \leq J}$  are obtained;
3. ***Error correction decoding:*** the error correction decoding technique discussed in Section 5.3.2 can also be applied at this stage where the received LSBs of quantized coefficients  $\{\bar{d}_{i,j,n}\}_{J-J_N+1 \leq j \leq J}$  are used together with the predicted coefficients  $\{\tilde{d}_{i,j,n}\}_{J-J_N+1 \leq j \leq J}$  to decode  $d_{i,j,n}$ . The decoded coefficient is denoted with  $\hat{d}_{i,j,n}$ ;
4. ***Final reconstruction:*** the signal  $f_i(t)$  is reconstructed by taking the inverse wavelet transform of the following set of coefficients:

$$\{\bar{c}_{i,j,n}\}, \quad \{\hat{d}_{i,j,n}\}_{J-J_N+1 \leq j \leq J} \quad \text{and} \quad \{\tilde{d}_{i,j,n}\}_{-\infty < j \leq J-J_N}.$$

Note that the encoding strategy of  $f_i(t)$  is based on the proposed compression scheme in Algorithm 1 in the previous chapter.

## 6.4 Distortion-Rate Analysis

This section presents the distortion-rate analysis of the proposed distributed compression algorithm. As with the centralized case, we consider a simplified model of the piecewise smooth signal, which consists of a step function and a uniformly smooth  $\alpha$ -Lipschitz function. For the sake of clarity in the following analysis, we will first assume

that there are only two functions  $f_1(t)$  and  $f_2(t)$  where

$$f_i(t) = s_i(t) + f_{i_\alpha}(t) \quad \text{with} \quad s_i(t) = \begin{cases} 0 & t < t_i, \\ A & t_i \leq t < 1 \end{cases} \quad \text{for } i = 1, 2.$$

The generalization of the analysis to the case of  $N$  signals will be given at the end of this section.

#### 6.4.1 Disparity by Translation

Let us begin by assuming that there is no prediction error (6.2) i.e.  $\epsilon_{2_\alpha}(t) = 0$ . Hence, the function  $f_2(t)$  can be written as

$$f_2(t) = f_1(t - \tau) = s_1(t - \tau) + f_{1_\alpha}(t - \tau).$$

We have that the total distortion is the sum of two distortion terms:

$$D = D_1 + D_2, \tag{6.5}$$

where  $D_1$  and  $D_2$  are the distortion due to the reconstruction of  $f_1(t)$  and  $f_2(t)$  respectively. Since a conventional wavelet nonlinear approximation-based compression scheme is used to encode  $f_1(t)$  in the proposed algorithm, from Theorem 11,  $D_1$  is given by

$$D_1(R_1) \leq c_1 R_1^{-2\alpha}, \tag{6.6}$$

where  $R_1$  is the total number of bits allocated to compress  $f_1(t)$ . Our next task is then to derive  $D_2$ .

Since the decoder reconstructs  $f_2(t)$  by estimating the shift parameter  $\tau = t_1 - t_2$ , the reconstructed function can be written as

$$\hat{f}_2(t) = \hat{f}_1(t - \hat{\tau}) = \hat{s}_1(t - \hat{\tau}) + \hat{f}_{1_\alpha}(t - \hat{\tau}),$$

where  $\hat{\tau}$  is the estimated shift parameter. Let  $\epsilon_{f_1}(t)$  denote the compression error of



$f_1(t)$ :  $f_1(t) = \hat{f}_1(t) + \epsilon_{f_1}(t)$ . It then follows that  $D_2$ , which is measured as the MSE, is given by

$$\begin{aligned}
 D_2 &= \mathbb{E} \left[ \int \left( f_2(t) - \hat{f}_2(t) \right)^2 dt \right] \\
 &= \mathbb{E} \left[ \int \left( f_1(t - \tau) - \hat{f}_1(t - \hat{\tau}) \right)^2 dt \right] \\
 &= \mathbb{E} \left[ \int \left( f_1(t - \tau) - (f_1(t - \hat{\tau}) + \epsilon_{f_1}(t - \hat{\tau})) \right)^2 dt \right] \\
 &= \mathbb{E} \left[ \int (s_1(t - \tau) - s_1(t - \hat{\tau}))^2 dt \right] \\
 &\quad + \mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau}))^2 dt \right] \\
 &\quad + \mathbb{E} \left[ \int \epsilon_{f_1}(t - \hat{\tau})^2 dt \right] \\
 &\quad + 2\mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau})) (s_1(t - \tau) - s_1(t - \hat{\tau})) dt \right] \\
 &\quad + 2\mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau})) \epsilon_{f_1}(t - \hat{\tau}) dt \right] \\
 &\quad + 2\mathbb{E} \left[ \int (s_1(t - \tau) - s_1(t - \hat{\tau})) \epsilon_{f_1}(t - \hat{\tau}) dt \right].
 \end{aligned}$$

Let  $\epsilon_\tau$  denote an error in the estimation of  $\tau$  i.e.  $\epsilon_\tau = \tau - \hat{\tau}$ . Given that  $\epsilon_\tau$  is small, the above expression can be simplified with the following approximation:

$$\begin{aligned}
 \mathbb{E} \left[ \int (s_1(t - \tau) - s_1(t - \hat{\tau}))^2 dt \right] &\gg \mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau}))^2 dt \right] \\
 &\text{and} \\
 \mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau})) (s_1(t - \tau) - s_1(t - \hat{\tau})) dt \right] &\approx 0.
 \end{aligned}$$

In addition, if  $R_1$  is high, we further assume that

$$\begin{aligned}
 \mathbb{E} \left[ \int (f_{1_\alpha}(t - \tau) - f_{1_\alpha}(t - \hat{\tau})) \epsilon_{f_1}(t - \hat{\tau}) dt \right] &\approx 0 \\
 &\text{and} \\
 \mathbb{E} \left[ \int (s_1(t - \tau) - s_1(t - \hat{\tau})) \epsilon_{f_1}(t - \hat{\tau}) dt \right] &\approx 0.
 \end{aligned}$$

The distortion  $D_2$  can, therefore, be approximated as

$$D_2 \approx \mathbb{E} \left[ \int (s_1(t - \tau) - s_1(t - \hat{\tau}))^2 dt \right] + \mathbb{E} \left[ \int \epsilon_{f_1}^2(t - \hat{\tau}) dt \right] = A^2 \mathbb{E} [|\epsilon_\tau|] + D_1.$$

It is not surprising that the term  $D_1$  appears in the above expression as the algorithm uses  $\hat{f}_1(t)$  to form the prediction of  $f_2(t)$ . Furthermore, given that  $|A| > \sup_{t \in [0,1]} |f_{1_\alpha}(t)|$ , it is clear that the distortion due to the reconstruction of the step function dominates

in  $D_2$ .

Given that the location  $t_1$  can be retrieved by the decoder, the error in the parametric estimation then becomes

$$\epsilon_\tau = t_2 - \hat{t}_2 = \epsilon_{t_2}.$$

By assuming that the estimator of  $t_2$  is minimum-variance unbiased estimator, from the analysis of the centralized case in Chapter 5, this leads to the distortion of the form:

$$D_2 \approx A^2 \mathbb{E} [|\epsilon_{t_2}|] + D_1 = A^2 \sqrt{CRB(t_2)} + D_1,$$

where  $CRB(t_2)$  is the Cramér-Rao lower bound in the estimation of  $t_2$ . Using the same set of assumptions as our analysis in Chapter 5, it follows that

$$\sqrt{CRB(t_2)} = c_2 (\sigma_q^2 + \sigma_\alpha^2)^{\frac{1}{2}} \quad \text{with} \quad \sigma_q^2 = C 2^{-\frac{2R_2}{L}}.$$

Here,  $\sigma_q^2$  is the variance of the quantization noise and  $\sigma_\alpha^2$  represents the variance of the term  $\langle f_{2_\alpha}(t), \varphi_{J,n}(t) \rangle$ . Hence, we have that

$$D_2 \approx c_2 \left( c_3 2^{-\frac{2R_2}{L}} + \sigma_\alpha^2 \right)^{\frac{1}{2}} + D_1. \quad (6.7)$$

Therefore, the total distortion can be approximated by substituting (6.6) and (6.7) into (6.5), which gives

$$D \approx 2c_1 R_1^{-2\alpha} + c_2 \left( c_3 2^{-\frac{2R_2}{L}} + \sigma_\alpha^2 \right)^{\frac{1}{2}}. \quad (6.8)$$

Assuming a high rate regime, where  $c_3 2^{-\frac{2R_2}{L}} < \sigma_\alpha^2$ , the distortion given in (6.8) can be approximated with a Taylor series expansion of the square root function as follows:

$$D(R) \approx 2c_1 R_1^{-2\alpha} + \frac{c_2 c_3}{2\sigma_\alpha} 2^{-\frac{2R_2}{L}} + c_2 \sigma_\alpha. \quad (6.9)$$

By solving the Lagrange multiplier method, we have that the optimal rate allocation for  $R_2$  is

$$R_2 = \frac{L}{2} (2\alpha + 1) \log_2 R_1 + C \quad (6.10)$$

with a constant  $C$ . The total rate  $R$  is thus given by

$$R = R_1 + \frac{L}{2}(2\alpha + 1)\log_2 R_1 + C \approx R_1. \quad (6.11)$$

at high  $R_1$ . Therefore, by substituting (6.10) into (6.9) together with the approximation in (6.11), the overall  $D(R)$  function of the proposed scheme is as follows:

$$D(R) \leq 2c_1 R^{-2\alpha} + c_4 R^{-(2\alpha+1)} + c_2 \sigma_\alpha. \quad (6.12)$$

If the term  $\sigma_\alpha$  is sufficiently small, then  $D(R)$  decays as  $R^{-2\alpha}$  for a wide range of rates. Finally, we note here that even though Encoder 2 employs a linear compression strategy, the overall  $D(R)$  function has a decay characteristic of a nonlinear scheme for a wide range of rates.

#### 6.4.2 Disparity by Translation with Prediction Error

We can now add the prediction error  $\epsilon_\alpha(t)$  to the distortion-rate analysis or

$$f_2(t) = f_1(t - \tau) + \epsilon_\alpha(t) = s_1(t - \tau) + f_{1_\alpha}(t - \tau) + \epsilon_\alpha(t).$$

It follows that the total distortion of the proposed scheme is now given by

$$D = D_1 + D_2 = D_1 + D_\tau + D_\epsilon,$$

where  $D_\tau$  is due to the reconstruction of the prediction  $\tilde{f}_2(t) = \hat{f}_1(t - \hat{\tau})$  and  $D_\epsilon$  is due to the reconstruction of the prediction error. Our next step is to determine  $D_\epsilon$ .

#### Linear approximation of the prediction error

Since the function  $\epsilon_\alpha(t)$  is uniformly  $\alpha$ -Lipschitz, the wavelet linear approximation based compression gives the following distortion:

$$D_\epsilon(R_\epsilon) = \beta D_1(R_\epsilon) = \beta c_1 R_\epsilon^{-2\alpha} \quad \text{with } \beta \geq 0, \quad (6.13)$$

where  $R_\epsilon$  is the total rates allocated to represent  $\epsilon_\alpha(t)$ . The constant  $\beta$  is used to relate the energy of the prediction error to  $f_1(t)$ . In terms of the wavelet coefficients, (6.13) implies

$$\max_{j,n \in \mathbb{Z}} |d_{\epsilon_{j,n}}| \leq \sqrt{\beta} \max_{j,n \in \mathbb{Z}} |d_{1_{j,n}}|, \quad (6.14)$$

where  $\{d_{\epsilon_{j,n}}\}$  and  $\{d_{1_{j,n}}\}$  denote the wavelet coefficients of  $\epsilon_\alpha(t)$  and  $f_1(t)$  respectively.

If we are to compress  $\epsilon_\alpha(t)$  directly, a linear approximation-based compression would achieve the  $D(R)$  given in (6.13). Since  $d_{\epsilon_{j,n}}$  decays as  $2^{j(\alpha+1/2)}$  across scales, from (6.14), the linear approximation based compression strategy that keeps the coefficients in decomposition level  $J - J_N + 1 \leq j \leq J$  sets the quantizer step size to

$$\Delta_\epsilon \leq \sqrt{\beta} \max_{n \in \mathbb{Z}} (|d_{1_{J,n}}|) 2^{-J_N(\alpha+1/2)}. \quad (6.15)$$

Here, we assume that  $\max_{n \in \mathbb{Z}} (|d_{1_{J,n}}|) \approx \max_{n \in \mathbb{Z}} (|d_{2_{J,n}}|)$ . Let  $R_\epsilon(j)$  be the number of bits per coefficient required to directly quantize  $d_{\epsilon_{j,n}}$ , it follows that

$$\begin{aligned} R_\epsilon(j) &= \left\lceil \log_2 \left( \frac{\sqrt{\beta} \sup_{n \in \mathbb{Z}} (|d_{2_{J,n}}|) 2^{-(J-j)(\alpha+1/2)}}{\Delta_\epsilon} \right) \right\rceil + 1 \\ &= \lceil (J_N - J + j)(\alpha + 1/2) \rceil + 1, \quad \text{with } J_N > 0, \quad J < 0 \quad \text{and} \quad j \leq J. \end{aligned} \quad (6.16)$$

Note that one extra bit has been included for the sign.

In our setup, however, Encoder 2 does not have access to  $\epsilon_\alpha(t)$ . Intuitively, one can still quantize the coefficients of  $f_2(t)$  with a step size  $\Delta_\epsilon$  as given in (6.15). If we assume that the best possible prediction, which is  $f_1(t - \tau)$ , is available at the decoder, then the information of the wavelet coefficients of  $\epsilon_\alpha(t)$  is fully contained within the first  $R_\epsilon(j)$  LSBs. Thus, with the same argument as the one given in Section 5.5 of the previous chapter, the quantized coefficients of the prediction error can be retrieved from the first  $R_\epsilon(j)$  LSBs of each coefficient and any additional distortion is due to the error in the prediction. In the case where the best possible prediction can be obtained, the compression performance is then equivalent to that of the joint encoding case where  $\epsilon_\alpha(t)$  is accessible.

From (6.13), it is clear that  $\beta$  is the key parameter that relates the prediction error

$\epsilon_\alpha(t)$  to the side information  $f_1(t)$ . In reality, however,  $\beta$  has to be estimated prior to compression as  $\epsilon_\alpha(t)$  is not accessible. We denote with  $\beta^*$ , the value of  $\beta$  estimated by Encoder 2. Thus, the actual quantizer step size used by Encoder 2 is given by

$$\Delta_\epsilon^* = \sqrt{\beta^*} \sup_{n \in \mathbb{Z}} (|d_{2J,n}|) 2^{-J_N(\alpha+1/2)}. \quad (6.17)$$

Let us consider an ideal case where the best possible prediction of  $f_2(t)$ , which is given by  $f_1(t - \tau)$ , is available at the decoder. If  $\beta^* = \beta$ , the proposed scheme achieves the  $D(R)$  equivalent to that of a joint encoding scheme. Instead, if  $\beta^* > \beta$ , then the step size  $\Delta_\epsilon^*$  will be too large and the added redundancy will result in an inferior compression performance. On the other hand, if the quality of the prediction  $\tilde{f}_2(t) = \hat{f}_1(t - \hat{\tau})$  deteriorates, a larger  $\beta^*$  allows a wider range of error to be recovered. Hence, the value of  $\beta^*$  represents a trade-off between the compression performance and the robustness of the distributed compression algorithm. Lastly, if  $\beta^* < \beta$  then the encoder underestimates the energy of the prediction error, which means that not enough bits will be transmitted for  $\epsilon_\alpha(t)$  to be correctly decoded. In the analysis that follows, we will assume that  $\beta^* \geq \beta$ .

### Total distortion with prediction error

With the above quantization strategy, following from the discussion in Section 5.5 of Chapter 5, we assume that  $D_\epsilon$  decays as

$$D_\epsilon \leq \beta^* c_1 R_{2_\epsilon}^{-2\alpha}$$

with a linear approximation-based compression strategy both inside and outside the cone of influence of discontinuities. This is because the encoder transmits enough bits to carry the information of  $\epsilon_\alpha(t)$  and any additional error is due to the parametric estimation of the prediction  $\tilde{f}_2(t)$ . Here,  $R_{2_\epsilon}$  denotes the total number of bits allocated to represent  $\epsilon_\alpha(t)$ . Note also that we have used the estimated  $\beta^*$  instead of  $\beta$ . From the analysis in the previous setup, where the prediction error was absent, it directly follows

from (6.7) and (6.9) that  $D_\tau$  is given by

$$D_\tau(R_{2_\tau}) \approx c_1 R_1^{-2\alpha} + \frac{c_2 c_3}{2\sigma_\alpha} 2^{\frac{-2R_{2_\tau}}{L}} + c_2 \sigma_\alpha,$$

where  $R_{2_\tau}$  is the number of bits allocated to the  $L$  low-pass coefficients, which are used to estimate  $\tilde{f}_2(t)$ .

We can now express the total distortion as follows:

$$D(R) \leq 2c_1 R_1^{-2\alpha} + \frac{c_2 c_3}{2\sigma_\alpha} 2^{\frac{-2R_{2_\tau}}{L}} + c_2 \sigma_\alpha + \beta^* c_1 R_{2_\epsilon}^{-2\alpha} \quad (6.18)$$

with  $R = R_1 + R_{2_\tau} + R_{2_\epsilon}$ . The optimal bit allocation can then be obtained using the Lagrange multiplier method, which gives

$$\begin{aligned} R_{2_\tau} &= \frac{L}{2}(2\alpha + 1) \log_2 R_1 + C, \\ R_{2_\epsilon} &= \left(\frac{\beta^*}{2}\right)^{\frac{1}{(2\alpha+1)}} R_1. \end{aligned} \quad (6.19)$$

The total rate  $R$  can then be approximated as

$$\begin{aligned} R &= R_1 + \frac{L}{2}(2\alpha + 1) \log_2 R_1 + C + \left(\frac{\beta^*}{2}\right)^{\frac{1}{(2\alpha+1)}} R_1 \\ &\approx \left(1 + \left(\frac{\beta^*}{2}\right)^{\frac{1}{(2\alpha+1)}}\right) R_1 \end{aligned} \quad (6.20)$$

at high  $R$ . By substituting (6.19) and (6.20) into (6.18), we have that the  $D(R)$  curve for the proposed distributed semi-parametric compression is given by

$$D(R) \leq \left(1 + \left(\frac{\beta^*}{2}\right)^{\frac{1}{(2\alpha+1)}}\right)^{2\alpha+1} \left(2c_1 R^{-2\alpha} + c_4 R^{-(2\alpha+1)}\right) + c_2 \sigma_\alpha. \quad (6.21)$$

If we assume that  $R$  is high and that the term  $\sigma_\alpha$  is negligible, then the distortion-rate behavior at high rates follows

$$D(R) \leq 2 \left(1 + \left(\frac{\beta^*}{2}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1} c_1 R^{-2\alpha}. \quad (6.22)$$

### 6.4.3 Comparison with Independent and Joint Compression

#### Independent compression

Let us now compare the distortion-rate performance of the proposed compression scheme with a compression scheme that encodes and decodes  $f_1(t)$  and  $f_2(t)$  independently using a wavelet nonlinear approximation-based strategy. We denote the distortion-rate curve of such independent-encoder scheme with  $D_{ind}(R)$ , which can be written as

$$D_{ind}(R) \leq c_1 R_1^{-2\alpha} + c_1 R_2^{-2\alpha} \quad \text{with} \quad R = R_1 + R_2.$$

Clearly, the optimal rate allocation is given by  $R_1 = R_2$ . This gives the following distortion-rate function:

$$D_{ind}(R) \leq 2c_1 \left(\frac{R}{2}\right)^{-2\alpha} = 2^{2\alpha+1} c_1 R^{-2\alpha}. \quad (6.23)$$

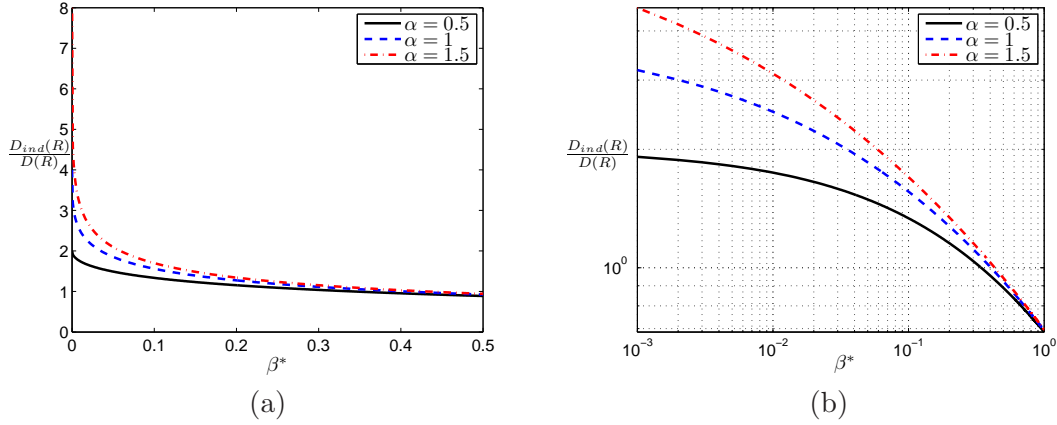
In comparison to the proposed distributed scheme, assuming that  $\sigma_\alpha$  is sufficiently small, the distortion of the independent scheme is higher by a factor of

$$\frac{D_{ind}(R)}{D(R)} = \frac{2^{2\alpha}}{\left(1 + \left(\frac{\beta^*}{2}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1}}. \quad (6.24)$$

Figure 6.4 (a) shows the plots of (6.24). Interestingly, the gain in the compression performance increases with the smoothness of the function. The log-log plot in Figure 6.4 (b) reveals that the performance gain over independent coding scheme exhibits a super-exponential decay with increasing  $\beta^*$ .

#### Centralized compression

As a benchmark for the best-case scenario, the  $D(R)$  of the proposed scheme is gauged against that of a compression scheme with a joint encoder and a joint decoder. In this setting, the encoder has access to both functions  $f_1(t)$  and  $f_2(t)$ . That is, the encoder knows the true value of  $\beta$ . This allows the residual function  $\epsilon_\alpha(t)$  to be encoded with optimal bit allocation and transmitted directly to the decoder along with the quantized



**Figure 6.4:** Plots of the function in (6.24). The gain in compression performance over an independent scheme increases with the degree of smoothness of the function. From the log-log plot in (b), we can see that the performance gain exhibits a super-exponential decay with increasing  $\beta^*$ .

shift parameter  $\tau$ . The decoder can then reconstruct  $f_2(t)$  as  $\hat{f}_2(t) = \bar{f}_1(t - \bar{\tau}) + \bar{\epsilon}_\alpha(t)$ .

Let  $\epsilon_\tau^q$  be the quantization error of  $\tau$ , assuming that a uniform quantizer is used, the distortion in the reconstruction of the prediction  $\tilde{f}_2(t) = f_1(t - \tau)$  can be approximated by

$$D_\tau(R_\tau) \approx A^2 E[|\epsilon_\tau^q|] \leq \frac{A^2}{\sqrt{12}} 2^{-R_\tau},$$

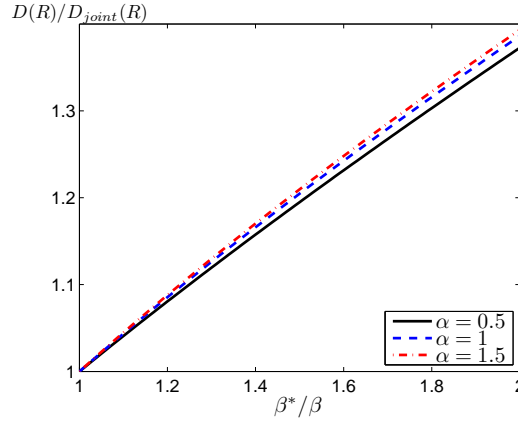
where we have used the Jensen's inequality and  $R_\tau$  denotes the number of bits allocated to quantize  $\tau$ . By following a similar analysis as shown in the distributed case, the total distortion can be shown to be

$$D_{joint}(R) \leq 2c_1 R_1^{-2\alpha} + \frac{A^2}{\sqrt{12}} 2^{-R_\tau} + \beta c_1 R_\epsilon^{-2\alpha},$$

where  $R_\epsilon$  is the total bits allocated to the compression of  $\epsilon_\alpha(t)$ . It then follows that the optimal bit allocation is given by

$$\begin{aligned} R_\tau &= (2\alpha + 1) \log_2 R_1 + C, \\ R_\epsilon &= \left(\frac{\beta}{2}\right)^{\frac{1}{(2\alpha+1)}} R_1. \end{aligned}$$





**Figure 6.5:** The plot of  $\frac{D(R)}{D_{joint}(R)}$  against  $\frac{\beta^*}{\beta}$ . The redundancy increases linearly with the ratio  $\frac{\beta^*}{\beta}$ .

In a high-rate regime, we can approximate the total rate  $R$  with

$$R = R_1 + R_\tau + R_\epsilon \approx \left( 1 + \left( \frac{\beta}{2} \right)^{\frac{1}{(2\alpha+1)}} \right) R_1.$$

It then follows that the joint encoding scheme achieves the following  $D(R)$  at high rates:

$$D(R) \leq \left( 1 + \left( \frac{\beta}{2} \right)^{\frac{1}{2\alpha+1}} \right)^{2\alpha+1} \left( 2c_1 R^{-2\alpha} + c_9 R^{-(2\alpha+1)} \right), \quad (6.25)$$

which has the same form as the distributed case given in (6.22). The only difference is in the values of  $\beta^*$  and  $\beta$ . This means that the closer the value of  $\beta^*$  can be to the actual  $\beta$  (i.e. the better the quality of the prediction of  $f_2(t)$ ), the closer the performance of the distributed compression scheme is to that of an ideal joint encoding scenario. The plot in Figure 6.5 indicates that the redundancy measured as  $\frac{D(R)}{D_{joint}(R)}$  increases linearly as  $\frac{\beta^*}{\beta}$  increases.

#### 6.4.4 Extension to $N$ Signals

We now extend the distortion-rate analysis for the setup in (6.3) to  $N$  signals. It is assumed here that  $\sigma_{i_\alpha} = \sigma_\alpha$  for  $i = 2, \dots, N$ . In addition, we assume that Encoder 2 to  $N$  use the same  $\beta^*$ . From (6.18), it follows that the total distortion is now given by

$$D(R) \leq N c_1 R_1^{-2\alpha} + \frac{c_2 c_3}{2\sigma_\alpha} \sum_{i=2}^N 2^{\frac{-2R_{i_\tau}}{L}} + (N-1) c_2 \sigma_\alpha + \beta^* c_1 \sum_{i=2}^N R_{i_\epsilon}^{-2\alpha},$$

which gives the following optimal rate allocation:

$$R_{i_\tau} = \frac{L}{2}(2\alpha + 1) \log_2 R_1 + C \quad \text{and} \quad R_{i_\epsilon} = \left(\frac{\beta^*}{N}\right)^{\frac{1}{2\alpha+1}} R_1, \quad i = 2, \dots, N.$$

The total rate can then be approximated as  $R \approx \left(1 + (N-1) \left(\frac{\beta^*}{N}\right)^{\frac{1}{2\alpha+1}}\right) R_1$ . Finally, the resulting  $D(R)$  bound obtained is given as follows:

$$D(R) \leq \left(1 + (N-1) \left(\frac{\beta^*}{N}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1} \left(Nc_1 R^{-2\alpha} + (N-1)c_{10} R^{-(2\alpha+1)}\right) + (N-1)c_2 \sigma_\alpha.$$

Similarly, one can easily show that the independent compression scheme for  $N$  signals achieves

$$D_{ind}(R) = 2c_1 N^{2\alpha} R^{-2\alpha}.$$

The same approach can be applied to a scheme with a joint encoder and the corresponding  $D(R)$  bound is given by

$$D_{joint}(R) \leq \left(1 + (N-1) \left(\frac{\beta}{N}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1} \left(Nc_1 R^{-2\alpha} + (N-1)c_{11} R^{-(2\alpha+1)}\right).$$

We summarize the findings of this section as follows:

**Summary 2** Consider a set of  $N$  piecewise smooth functions,  $\{f_i(t)\}_{1 \leq i \leq N} \in \mathbf{L}_2([0, 1])$ , where each consists of a step function and a uniformly  $\alpha$ -Lipschitz function and  $f_i(t) = f_1(t - \tau) + \epsilon_{i_\alpha}(t)$ . The function  $\epsilon_{i_\alpha}(t)$  is uniformly  $\alpha$ -Lipschitz. Given that the  $D(R)$  function corresponding to a linear compression of  $\epsilon_{i_\alpha}(t)$  follows  $D_{i_\epsilon}(R_{i_\epsilon}) = \beta D_1(R_{i_\epsilon})$ , at high rates, the semi-parametric distributed compression scheme presented in Algorithm 4 achieves

$$D(R) \leq \left(1 + (N-1) \left(\frac{\beta^*}{N}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1} \left(Nc_1 R^{-2\alpha} + (N-1)c_{10} R^{-(2\alpha+1)}\right) + (N-1)c_2 \sigma_\alpha.$$

where  $\beta^*$  is the estimate of  $\beta$ . At high rates, provided that  $\sigma_\alpha$  is sufficiently small, if  $\beta^* = \beta$  then the achieved  $D(R)$  performance is comparable to that of a joint encoding

scheme and is better by a factor of

$$\frac{2N^{2\alpha-1}}{\left(1 + (N-1) \left(\frac{\beta^*}{N}\right)^{\frac{1}{2\alpha+1}}\right)^{2\alpha+1}}$$

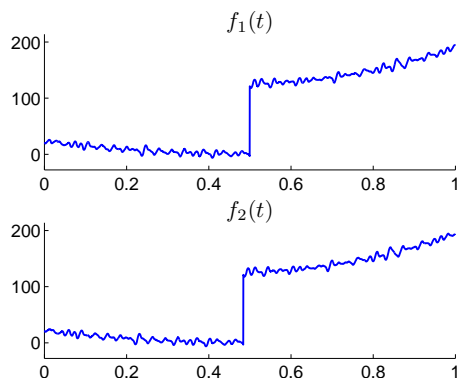
when compared to an independent compression scheme.

## 6.5 Simulation results

The simulation results of the proposed distributed semi-parametric compression schemes are presented in this section. We use a 1-D SPIHT [43] as discussed in Chapter 5 to perform the nonlinear compression of  $f_1(t)$ . Two piecewise smooth functions,  $f_1(t)$  and  $f_2(t)$ , are generated, where  $f_2(t) = f_1(t - \tau)$  and  $t \in [0, 1[$ . Both functions contain two smooth pieces with  $\alpha = 2.6$ . In the simulations, FRI-based algorithms presented in Chapter 5 are used to perform the parametric estimation of the location  $t_2$ . The wavelet transform uses a second order B-spline scaling function to decompose the signals up to six decomposition levels. Note that the term  $\sigma_\alpha$  is negligible in our simulations.

In the first simulation, the disparity between the two functions is completely described by the shift  $\tau$  with no prediction error as shown in Figure 6.6. Figure 6.7(a) shows the  $D(R)$  plots of the proposed scheme of Algorithm 4. Here, only the low-pass coefficients were encoded. The gain in performance over the independent encoding scheme is approximately  $\frac{D_{ind}(R)}{D(R)} \approx 28$  and the predicted value is  $2^{2\alpha} = 36.76$ . The plots in decibel are shown in Figure 6.7 (b). At high rates, our scheme outperforms the independent scheme by approximately 15 dB. The gain predicted is  $10 \log_{10}(2^{2\alpha}) = 15.65\text{dB}$ . In comparison to the joint encoding scheme, the compression performance of the proposed scheme is outperformed by approximately 2 dB.

The second simulation includes the prediction error into the setup, where a smooth function  $\epsilon_\alpha(t)$  is added to  $f_2(t)$  such that  $\sup_{t \in [0, 1[} |\epsilon_\alpha(t)| \leq \sup_{t \in [0, 1[} \sqrt{\beta} |f_1(t)|$  with  $\beta = 0.04$ . Encoder 2 uses  $\beta^* = 0.06$  in our simulation. Note that the distributed compression scheme requires  $\beta^* \geq \beta$  in order to decode the error correctly. The plots of the two functions  $f_1(t)$  and  $f_2(t)$  are shown in Figure 6.8. Figure 6.9(a) shows  $D(R)$



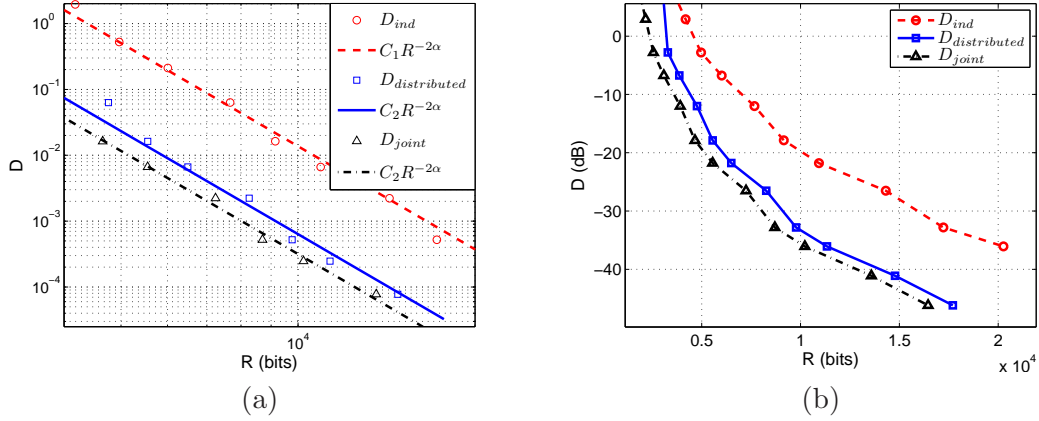
**Figure 6.6:** Plots of the two original piecewise smooth functions  $f_1(t)$  and  $f_2(t)$  used in the first simulation, where  $f_2(t) = f_1(t - \tau)$ .

plots of the proposed scheme of Algorithm 4. The distortion of the proposed distributed scheme is approximately 2.33 times lower when compared to the independent coding scheme, which is in line with the predicted gain of 2.26 times given by (6.24). The result also shows that the achieved distortion is very close to that of the joint encoding scheme. This is because  $\beta^*$  is well calibrated to be close to  $\beta$ . Figure 6.9 (b) shows the same plots in decibel scale. At high rates, our scheme outperforms the independent scheme by approximately 3 dB (the gain predicted in our analysis is 3.5 dB).

## 6.6 Conclusion

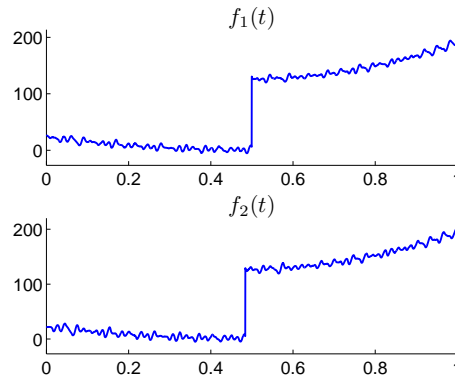
This chapter has presented the distributed semi-parametric compression schemes of piecewise smooth functions. The signal disparity model described the difference between each signal with a translation and a prediction error. Our problem setup assumes that one of the two functions is available at the decoder via conventional compression method. The distributed compression algorithm is then built based on the proposed centralized semi-parametric algorithms presented in the previous chapter, where the encoder only employs a standard linear approximation-based compression strategy.

The distortion-rate analysis shows that the proposed scheme can achieve a compression performance comparable to that of a centralized joint encoding scheme for a wide range of rates. We have also calculated the gain in performance relative to the independent encoding and decoding scheme. When the prediction error is presented in the

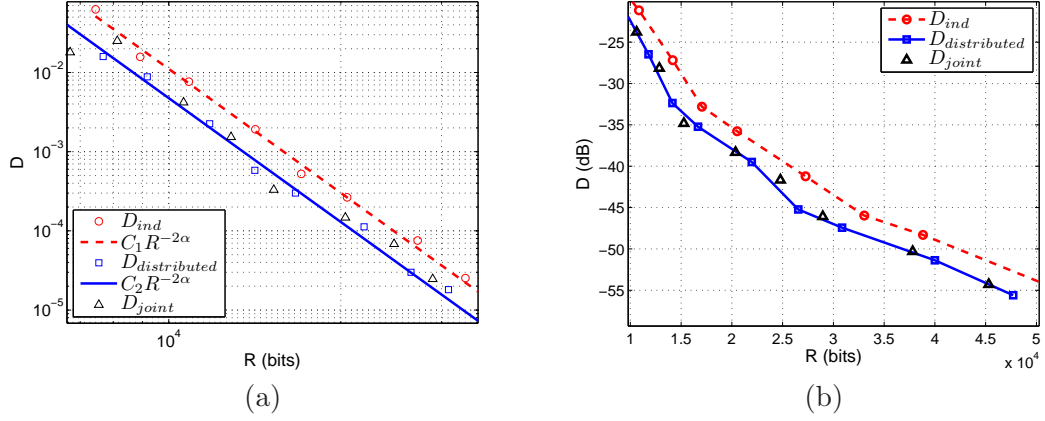


**Figure 6.7:** Distortion-rate plots of compression schemes based on disparity-by-translation model (log-log scale in (a) and decibel scale in (b)). The proposed distributed semi-parametric compression scheme in Algorithm 4 outperforms the independent compression scheme by a factor of  $\frac{D_{ind}(R)}{D(R)} \approx 28$  ( $\approx 15$  dB), where the predicted factor is  $2^{2\alpha} = 36.76$  (15.65 dB). The joint encoding scheme outperforms the proposed scheme by approximately 2 dB

form of a smooth function, the encoder has to estimate the power of the error (i.e.  $\beta$ ) a priori. In contrast to the standard standard channel coding approaches, which always assume that the wavelet coefficients of the prediction error do not decay across scales, the proposed scheme can allocate the bits more precisely with linear approximation-based strategy in this case. Here, the value of the estimated  $\beta^*$  has an impact on the overall performance. Given that  $\beta^* \approx \beta$  and that the decoder can form the prediction of the function  $f_2(t)$  that is close to that of a joint encoder, the proposed scheme can achieve the  $D(R)$  that is comparable to that of the joint encoding algorithm. With



**Figure 6.8:** Plots of the two original piecewise smooth functions  $f_1(t)$  and  $f_2(t)$  used in the second simulation, where  $f_2(t) = f_1(t - \tau) + \epsilon_\alpha(t)$ .



**Figure 6.9:** Distortion-rate plots of compression schemes based on disparity-by-translation model with prediction error (log-log scale in (a) and decibel scale in (b)). The proposed distributed semi-parametric compression scheme in Algorithm 4 outperforms the independent compression scheme by a factor of  $\frac{D_{ind}(R)}{D(R)} \approx 2.33$ , which is close to the predicted value of 2.26 (from (6.24)). The achieved  $D(R)$  is comparable to that of the joint encoder scheme.

the proposed compression algorithm, there is no change to the structure of the wavelet transform. Instead, the quantization strategy and the bit allocation are different from the centralized case, where both depend on the value of the parameter  $\beta$ . Here, we can see that the use of an error correction code in the proposed scheme is similar to the use of coset in channel coding technique, which is used in the Wyner-Ziv problem reviewed in Chapter 2.

# CHAPTER 7

## Distributed Semi-Parametric Compression of 2-D Signals

### 7.1 Introduction

This chapter focuses on the applications of the semi-parametric compression schemes discussed in Chapter 5 and 6 in the context of distributed coding of images and video. In particular, we show that by using the transform coefficients to perform parametric estimation, the complex process of motion estimation can be shifted to the decoder side. This allows a reduction of the complexity of the encoding process and, by performing joint decoding, the overall performances are improved.

In the next section, we briefly review the notion of the wavelet transform and approximation in 2-D. Section 7.3 presents a case study of the FRI-based distributed compression scheme for a toy model of a video sequence, which consists of a translating bi-level polygon. This concept is then extended, in Section 7.4, to a real object whose motion can be described by an affine transform. The applications of the proposed coding schemes to more realistic signals are then presented in Section 7.5. Finally, conclusions are then given in Section 7.6.

## 7.2 Wavelet Approximation in 2-D

Let us first formally state the notion of wavelet linear and nonlinear approximation of 2-D continuous functions, which is a straightforward extension of the 1-D case reviewed in Chapter 3. Given a continuous 2-D function  $f(x, y)$ ,  $x, y \in \mathbf{L}_2([0, 1])$ , its wavelet decomposition is given as follows:

$$f(x, y) = \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^J \varphi_{m,n}^J(x, y) + \sum_{j=-\infty}^J \sum_{m=0}^{2^{-j}-1} \sum_{n=0}^{2^{-j}-1} d_{m,n}^j \psi_{m,n}^j(x, y) \quad \text{with } J < 0. \quad (7.1)$$

Here, we have that  $L_m = L_n = 2^{-J}$ . In this chapter, we consider the case where the scaling function  $\varphi(x, y)$  and the wavelet  $\psi(x, y)$  are separable. Hence,  $\varphi(x, y)$  and  $\psi(x, y)$  are obtained by the tensor product of two 1-D functions:  $\varphi(x, y) = \varphi_1(x) \otimes \varphi_2(y)$  and  $\psi(x, y) = \psi_1(x) \otimes \psi_2(y)$ . We assume that  $\varphi_1 = \varphi_2 = \varphi$  and  $\psi_1 = \psi_2 = \psi$ . It, therefore, follows that  $\psi_{m,n}^j(x, y) = \psi_{j,m}(x) \otimes \psi_{j,n}(y)$ , where  $\psi_{j,m}(x) = 2^{-j/2} \psi(2^{-j}x - m)$  and  $\psi_{j,n}(y) = 2^{-j/2} \psi(2^{-j}y - n)$ ,  $m, n \in \mathbb{Z}$ , and similarly for  $\varphi_{m,n}^J(x, y)$ . The low-pass and high-pass coefficients,  $\{c_{m,n}^J\}$  and  $\{d_{m,n}^j\}$ , are then given by the following inner products:

$$c_{m,n}^J = \langle f(x, y), \tilde{\varphi}_{m,n}^J(x, y) \rangle \quad \text{and} \quad d_{m,n}^j = \langle f(x, y), \tilde{\psi}_{m,n}^j(x, y) \rangle,$$

where  $\tilde{\varphi}_{m,n}^J(x, y)$  and  $\tilde{\psi}_{m,n}^j(x, y)$  are the dual of  $\varphi_{m,n}^J(x, y)$  and  $\psi_{m,n}^j(x, y)$  respectively.

We, therefore, have that the  $N$ -term linear approximation of  $f(x, y)$ , where  $N \sim 2^{2j}$ , is given by

$$f_N(x, y) = \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^J \varphi_{m,n}^J(x, y) + \sum_{j=J-J_N+1}^J \sum_{m=0}^{2^{-j}-1} \sum_{n=0}^{2^{-j}-1} d_{m,n}^j \psi_{m,n}^j(x, y),$$

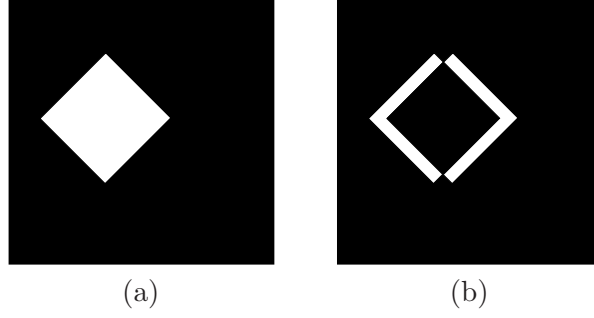
which is equivalent to keeping every coefficient in the first  $J_N$  decomposition levels.

Finally, the best nonlinear approximation of  $f(x, y)$  is given by

$$f_{J_N}(x, y) = \sum_{(m,n) \in \mathcal{J}_N} c_{m,n}^J \varphi_{m,n}^J(x, y) + \sum_{(j,m,n) \in \mathcal{J}_N} d_{m,n}^j \psi_{m,n}^j(x, y),$$

where  $\mathcal{J}_N$  denotes the index set of the  $N$  largest coefficients.





**Figure 7.1:** Examples of equilateral bilevel polygon signals: (a) a polygon with four corner points; (b) illustration of the distortion due to the error in the retrieved translation vector.

## 7.3 Bi-level Polygon Case Study

### 7.3.1 Signal Model

In order to gain some intuition, we start by considering a simple toy model of a video sequence (or an array of images) that consists of a single bi-level equilateral polygon in a uniform background. The polygon is uniquely defined by a set of locations of its  $K$  corner points  $\{(x_k, y_k)\}_{k=1, \dots, K}$ . We assume that the disparity between each frame is described by a translation vector  $\boldsymbol{\tau}_i = [\tau_{x_i}, \tau_{y_i}]$ . Let us define a set of  $N$  frames with  $f_i(x, y)$ ,  $i = 1, 2, \dots, N$ ,  $x, y \in \mathbf{L}_2([0, 1])$  and let  $f_1(x, y)$  be the reference or the key frame. Thus, the relationship between each frame can be written as follows:

$$f_i(x, y) = f_1(x - \tau_{x_i}, y - \tau_{y_i}), \quad i = 2, 3, \dots, N.$$

In the analysis that follows, we will assume for simplicity that

$$\iint f_i(x, y) dx dy = 1, \quad i = 1, 2, \dots, N.$$

We denote the length of each side of the polygon with  $\ell$  and the amplitude of the polygon with  $A$ . The uniform background has the amplitude set to zero. The illustration of this signal model is shown in Figure 7.1 (a).

### 7.3.2 Distributed Compression Strategy

#### Compression algorithm

We now construct a distributed compression scheme based on the approach presented in the previous chapter. Here, we assume that both the encoder and the decoder know that the sequence contains a single bi-level polygon and that the disparity between each frame is due to translation only. The proposed compression algorithm is given as follows:

**Algorithm 5 :** *Distributed semi-parametric compression algorithm for the translating equilateral bi-level polygon sequence.*

Encoding and Decoding of the Key Frame  $f_1(x, y)$

1. **Quantization of corner points:** the locations of the corner points  $\{(x_k, y_k)\}_{k=1, \dots, K}$  of the polygon in  $f_1(t)$  are quantized with a uniform scalar quantizer and transmitted by the encoder;
2. **Reconstruction of the key frame:** the decoder reconstructs the key frame to obtain  $\hat{f}_1(x, y)$  from the received quantized locations of the corner points  $\{(\bar{x}_k, \bar{y}_k)\}_{k=1, \dots, K}$ .

Encoding of  $f_i(x, y)$ ,  $i = 2, \dots, N$

1. **Uniform quantization of low-pass coefficients:**  $L_m L_n$  low-pass coefficients  $\{c_{m,n}^J\}_{0 \leq m < L_m, 0 \leq n < L_n}$ , of  $f_i(t)$  are uniformly quantized and transmitted.

Joint Decoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. **Parametric estimation:** the decoder estimates the translation vectors  $\{\tau_i\}_{2 \leq i \leq N}$ , from the received quantized low-pass coefficients  $\{\bar{c}_{m,n}^J\}$ ;
2. **Reconstruction by translation:**  $f_i(x, y)$  is reconstructed as  $\hat{f}_i(x, y) = \hat{f}_1(x - \hat{\tau}_{x_i}, y - \hat{\tau}_{y_i})$ .

#### Parametric estimation with FRI principle

In this section, we show that the translation vectors can be estimated from the continuous geometric moments of the function  $f_i(x, y)$ . We first recall that a continuous

geometric moment of order  $(p + q)$  is defined as

$$M_{p,q} = \iint f(x, y) x^p y^q dx dy.$$

Let us denote the geometric moment of  $f_i(x, y)$  with  $M_{p,q}^i$ . It then follows that the first order moment of the  $i$ -th frame is given by

$$\begin{aligned} M_{1,0}^i &= \iint f_1(x - \tau_{x_i}, y - \tau_{y_i}) x dx dy \\ &\stackrel{(a)}{=} \iint f_1(x', y') (x' + \tau_{x_i}) dx' dy' \\ &= M_{1,0}^1 + \tau_{x_i} M_{0,0}^1 \\ &\stackrel{(b)}{=} M_{1,0}^1 + \tau_{x_i}, \end{aligned}$$

where (a) follows from the substitution  $x' = x - \tau_{x_i}$  and (b) from the assumption that  $M_{0,0}^i = \iint f_i(x, y) dx dy = 1$ . A similar expression for  $\tau_{y_i}$  is obtained by evaluating  $M_{0,1}^i$ . Therefore, the translation vector  $\tau_i$ ,  $i = 2, \dots, N$ , can be calculated from the first order moments as follows:

$$\tau_{x_i} = M_{1,0}^i - M_{1,0}^1 \quad \text{and} \quad \tau_{y_i} = M_{0,1}^i - M_{0,1}^1. \quad (7.2)$$

It is now clear that  $\tau_i$  can be retrieved from the moments of  $f_i(x, y)$ . Therefore, in order to perform parametric estimation during joint decoding, the decoder of Algorithm 5 can apply the 2-D moment-samples relationship shown in (4.28) in Chapter 4 to estimate  $M_{p,q}^i$ ,  $p, q = 0, 1$ , from the low-pass coefficients as follows:

$$\widehat{M}_{p,q}^i = \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^{(p,q)} \bar{c}_{m,n}^J, \quad p, q = 0, 1 \quad (7.3)$$

with a proper choice of coefficients  $c_{m,n}^{(p,q)}$ . We denote the estimated moments with  $\widehat{M}_{p,q}^i$ .

The translation vector  $\tau_i$  can then be estimated as

$$\hat{\tau}_{x_i} = \widehat{M}_{1,0}^i - \widehat{M}_{1,0}^1 \quad \text{and} \quad \hat{\tau}_{y_i} = \widehat{M}_{0,1}^i - \widehat{M}_{0,1}^1. \quad (7.4)$$

### 7.3.3 Quantization Error in Moment-Samples Relationship

Before moving on to the distortion-rate analysis of the above compression algorithm, we first assess the effect of quantization error on the retrieved moments in (7.3). Let us write  $\bar{c}_{m,n}^J = c_{m,n}^J + \epsilon_{m,n}^q$ , where  $\epsilon_{m,n}^q$  denotes the quantization noise. From (7.3), it follows that

$$\widehat{M}_{p,q}^i = \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^{(p,q)} c_{m,n}^J + \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^{(p,q)} \epsilon_{m,n}^q = M_{p,q}^i + w_{p,q}^i, \quad (7.5)$$

where  $w_{p,q}$  represents the error in the retrieved moments due to quantization.

Since a uniform scalar quantizer with a step size  $\Delta$  is used, at high rates, the PDF of each term in the sum  $\sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} c_{m,n}^{(p,q)} \epsilon_{m,n}^q$  is given by

$$p(c_{m,n}^{(p,q)} \epsilon_{m,n}^q) = \begin{cases} \frac{1}{c_{m,n}^{(p,q)} \Delta} & -\frac{c_{m,n}^{(p,q)} \Delta}{2} \leq c_{m,n}^{(p,q)} \epsilon_{m,n}^q \leq \frac{c_{m,n}^{(p,q)} \Delta}{2}, \\ 0 & \text{otherwise.} \end{cases} \quad (7.6)$$

Moreover, we assume that  $\epsilon_{m,n}^q$  are independent so that the PDF of  $w_{p,q}^i$  is the  $(L_m L_n - 1)$  times convolution:

$$p(w_{p,q}) = \underbrace{p(c_{0,0}^{(p,q)} \epsilon_{0,0}^q) * p(c_{0,1}^{(p,q)} \epsilon_{0,1}^q) * \dots * p(c_{L_m-1, L_n-1}^{(p,q)} \epsilon_{L_m-1, L_n-1}^q)}_{(L_m L_n - 1) \text{ times}}.$$

Given that  $L_m L_n$  is large,  $p(w_{p,q}^i)$  is approximately a Gaussian function. From (7.6), we have that the variance of the term  $c_{m,n}^{(p,q)} \epsilon_{m,n}^q$  is as follows:

$$\text{var}(c_{m,n}^{(p,q)} \epsilon_{m,n}^q) = \frac{\left(c_{m,n}^{(p,q)} \Delta\right)^2}{12}.$$

Let  $R_c$  be the number of bits allocated to represent one low-pass coefficient. We have that

$$\Delta = 2^{-R_c} \left( \max_{0 \leq m < L_m, 0 \leq n < L_n} |c_{m,n}^J| \right).$$

Note that the sign bit is excluded for simplicity here. Therefore, we have that the

variance of  $w_{p,q}$ , which is denoted by  $\sigma_w^2$ , is given by

$$\sigma_w^2 = \frac{\Delta^2}{12} \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} \left( c_{m,n}^{(p,q)} \right)^2 = C_1 2^{-2R_c} \quad (7.7)$$

with a constant  $C_1 = \frac{1}{12} \left( \max_{0 \leq m < L_m, 0 \leq n < L_n} |c_{m,n}^J| \right)^2 \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} \left( c_{m,n}^{(p,q)} \right)^2$ .

### 7.3.4 Distortion-Rate Analysis

#### Distributed semi-parametric compression scheme

We now derive the  $D(R)$  bound of the scheme proposed in Algorithm 5. The total distortion due to the reconstruction of  $f_i(x, y)$ ,  $i = 1, \dots, N$ , is given by  $D_{distr} = \sum_{i=1}^N D_i$ , where  $D_i = \mathbb{E} \left[ \|f_i(x, y) - \hat{f}_i(x, y)\|^2 \right]$ . Consider first, the distortion of the key frame  $D_1$ , which arises from the quantization of the corner points  $\{(x_k, y_k)\}_{k=1, \dots, K}$ . Assuming that a uniform quantizer is used with a step size  $\Delta_1 = 1/2^{R_1}$  where  $R_1$  is the number of bits allocated to each component of the corner point coordinate, at high rates, the PDFs of the quantization error of the  $x$  and  $y$  coordinates are

$$p(\epsilon_y) = p(\epsilon_x) = p(\epsilon_{x,y}) = \begin{cases} \frac{1}{\Delta_1} & -\frac{\Delta_1}{2} \leq \epsilon_{x,y} \leq \frac{\Delta_1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, we assume that the distortion due to quantization of the  $x$  and  $y$  coordinates are independent and, thus, additive. It then follows that  $D_1$  is bounded by

$$\begin{aligned} D_1 &\leq \mathbb{E} [KA^2 \ell |\epsilon_x|] + \mathbb{E} [KA^2 \ell |\epsilon_y|] \\ &= 2KA^2 \ell \mathbb{E} [|\epsilon_{x,y}|] \\ &\stackrel{(a)}{\leq} 2KA^2 \ell \frac{\Delta_1}{\sqrt{12}} \\ &= \frac{1}{\sqrt{3}} KA^2 \ell 2^{-R_1}, \end{aligned} \quad (7.8)$$

where in (a) we have applied Jensen's inequality for concave functions.

Let us now derive the bounds for the distortion  $D_i$ ,  $i = 2, 3, \dots, N$ . Since the encoding method is the same for each of the non-key frames, it is clear that  $D_i = D_j = D_c$  for  $i \neq j$  and  $i, j = 2, \dots, N$ . Intuitively, the distortion  $D_1$  reappears in  $D_c$  because the

decoder in Algorithm 5 reconstructs  $f_i(x, y)$ ,  $i = 2, \dots, N$ , by shifting  $\hat{f}_1(x, y)$ . Hence, we have that

$$D_c = D_1 + D_\tau,$$

where  $D_\tau$  is due to the error in the retrieved translation vector. Figure 7.1 (b) shows a visualization of the squared error in the reconstruction due to the error in the retrieved shift parameter  $\tau_i$ . As with before, we assume that the distortion from the error in the  $x$  and  $y$  coordinates of  $\tau_i$  are independent and additive. Let  $\epsilon_{\tau_x}$  and  $\epsilon_{\tau_y}$  denote the error in the retrieved  $\tau_i$  such that  $\hat{\tau}_i = [\hat{\tau}_{x_i}, \hat{\tau}_{y_i}] = [\tau_{x_i} + \epsilon_{\tau_x}, \tau_{y_i} + \epsilon_{\tau_y}]$ . From (7.4) and (7.5), we have that

$$\hat{\tau}_{x_i} = \widehat{M}_{1,0}^i - \widehat{M}_{1,0}^1 = M_{1,0}^i - M_{1,0}^1 + w_{1,0}^i - \epsilon_{M_{1,0}^1},$$

where  $\epsilon_{M_{1,0}^1}$  is the error in the retrieved moment of the key frame. At high rates, we assume that  $w_{1,0}^i \gg \epsilon_{M_{1,0}^1}$  and that  $\epsilon_{M_{1,0}^1} \approx 0$ . Therefore, we have that

$$\epsilon_{\tau_x} = w_{1,0}^i \quad \text{and} \quad \epsilon_{\tau_y} = w_{0,1}^i.$$

It then follows that  $D_\tau$  is bounded by

$$\begin{aligned} D_\tau &\leq \mathbb{E} [KA^2\ell|\epsilon_{\tau_x}|] + \mathbb{E} [KA^2\ell|\epsilon_{\tau_y}|] \\ &= KA^2\ell (\mathbb{E} [|w_{1,0}^i|] + \mathbb{E} [|w_{0,1}^i|]) \\ &\stackrel{(a)}{\leq} 2KA^2\ell\sigma_w \\ &\stackrel{(b)}{=} 2KA^2\ell\sqrt{C_1}2^{-R_c}, \end{aligned} \tag{7.9}$$

where we have used Jensen's inequality at (a) and (b) follows from (7.7). Here we assumed that  $\varphi_{m,n}^J$  is symmetrical so that  $\sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} \left(c_{m,n}^{(1,0)}\right)^2 = \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} \left(c_{m,n}^{(0,1)}\right)^2$ .

We can now write the total distortion as

$$\begin{aligned} D_{distr} &= \sum_{i=1}^N D_i \\ &= ND_1 + (N-1)D_\tau \\ &\leq N\frac{1}{\sqrt{3}}KA^2\ell 2^{-R_1} + (N-1)2KA^2\ell\sqrt{C_1}2^{-R_c} \end{aligned} \tag{7.10}$$

and the total rate is given by

$$R = 2KR_1 + (N - 1)L_m L_n R_c. \quad (7.11)$$

By solving the Lagrange multiplier method, we obtain the following optimal rate allocation:

$$R_c = R_1 + \log_2 \left( \frac{\sqrt{C_2}(N - 1)}{N} \right), \quad (7.12)$$

where  $C_2 = \left( \max_{0 \leq m < L_m, 0 \leq n < L_n} |c_{m,n}^J| \right)^2 \sum_{m=0}^{L_m-1} \sum_{n=0}^{L_n-1} \left( c_{m,n}^{(1,0)} \right)^2$ . By substituting (7.12) into (7.11) and (7.10), we have that the  $D(R)$  bound for the compression scheme in Algorithm 5 is given by

$$D_{distr}(R) \leq NKA^2\ell \frac{2}{\sqrt{3}} \left( \frac{\sqrt{C_2}(N - 1)}{N} \right)^{\left( \frac{N-1}{2K+(N-1)L_m L_n} \right)} 2^{-\frac{R}{2(K+(N-1)L_m L_n/2)}}. \quad (7.13)$$

### Comparison with independent and joint compression schemes

In order to gauge the performance of the proposed scheme, we compare the  $D(R)$  bound in (7.13) to that of the ideal independent and joint compression schemes. For the independent scheme, the coordinates of the corner points in each frame are uniformly quantized and directly transmitted to the decoder. It is easy to extend the result in (7.8) to derive the distortion for the independent scheme. It then follows that

$$D_{ind} = \sum_{i=1}^N D_i = \sum_{i=1}^N \frac{1}{\sqrt{3}} KA^2\ell 2^{-R_i}.$$

Clearly, the optimal rate allocation is  $R_i = R_j = R_k$ ,  $i \neq j$ , and the total rate is given by  $R = 2NKR_k$ . This gives us the following  $D(R)$  bound

$$D_{ind}(R) = NKA^2\ell \frac{1}{\sqrt{3}} 2^{-\frac{R}{2NK}}. \quad (7.14)$$

For the joint compression scheme, the coordinates of the corner points in the key frame  $f_1(x, y)$  are quantized and transmitted to the decoder along with the translation vectors  $\tau_i$ ,  $i = 2, \dots, N$ . Using the same argument as the analysis of the distributed case,

we have that  $D = ND_1 + (N - 1)D_\tau$ , where  $D_\tau$  is the distortion due to the quantization of  $\tau_i$ . Given a uniform quantizer with a step size  $\Delta_\tau = \frac{1}{2^{R_\tau}}$ , where  $R_\tau$  is the rate allocated to each component of the translation vector, with the assumption that the distortion of the  $x$  and  $y$  coordinates are additive, we have that

$$\begin{aligned} D_\tau &\leq \mathbb{E} [KA^2\ell|\epsilon_{\tau_x}|] + \mathbb{E} [KA^2\ell|\epsilon_{\tau_y}|] \\ &\stackrel{(a)}{\leq} 2KA^2\ell\frac{\Delta_\tau}{\sqrt{12}} \\ &= \frac{1}{\sqrt{3}}KA^2\ell 2^{-R_\tau}, \end{aligned}$$

where Jensen's inequality is applied at (a). The total distortion is, therefore, given by

$$D \leq \frac{1}{\sqrt{3}}KA^2\ell (N2^{-R_1} + (N - 1)2^{-R_\tau})$$

with the total rate  $R = 2KR_1 + 2(N - 1)R_\tau$ . This gives us the optimal rate allocation of  $R_\tau = R_1 + \log_2 \left( \frac{N-1}{N} \right)$  and the corresponding  $D(R)$  bound is

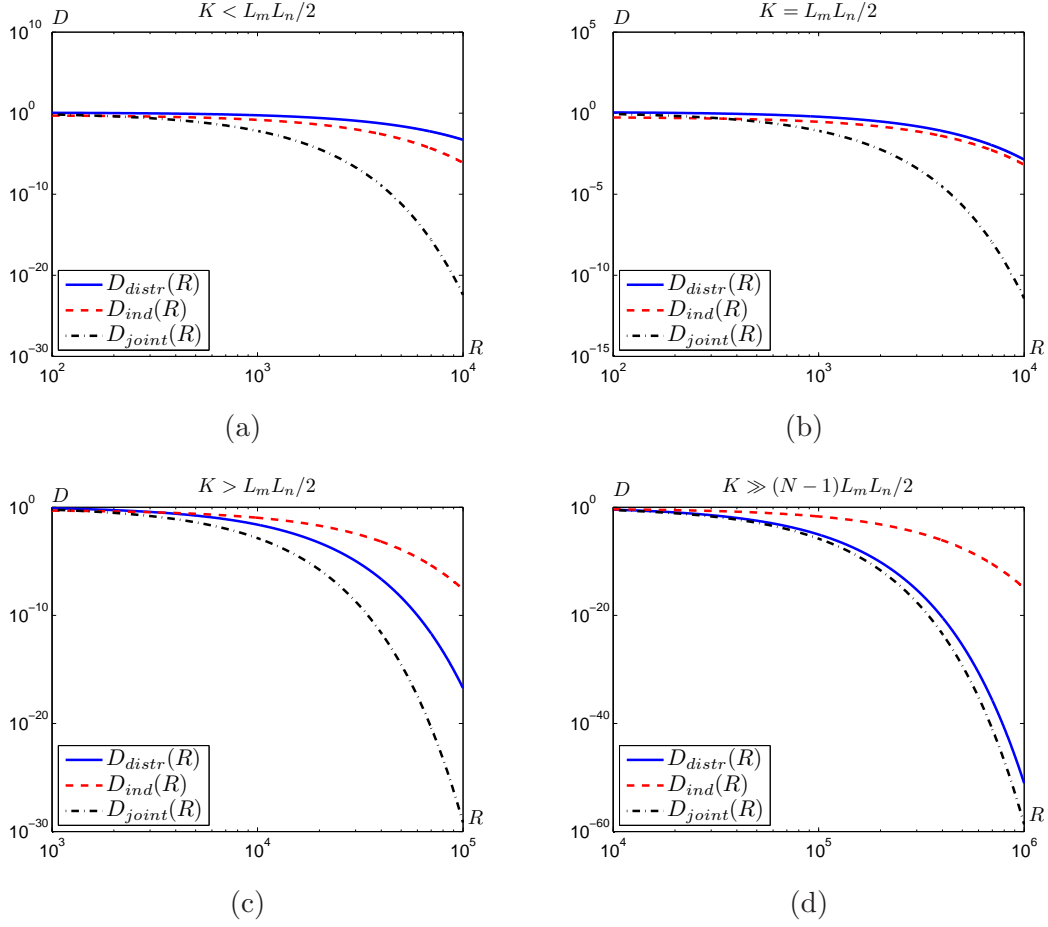
$$D_{joint}(R) = NKA^2\ell \frac{2}{\sqrt{3}} \left( \frac{N-1}{N} \right)^{\frac{N-1}{K+N-1}} 2^{-\frac{R}{2(K+N-1)}}. \quad (7.15)$$

### D(R) performance comparison

From (7.13) and (7.15), it is clear that the distortion of the proposed distributed scheme in Algorithm 5 always decays at a slower rate than the ideal joint compression scheme as  $\frac{R}{2(K+(N-1)L_mL_n/2)} < \frac{R}{2(K+N-1)}$ . By reducing the quantity  $L_mL_n$ , which translates to lowering the resolution of the low-pass coefficients, the  $D(R)$  of the proposed scheme approaches that of the joint compression scheme. Note that the FRI principle allows the moments to be retrieved even at lower resolution. Moreover, as the complexity of the polygon increases such that  $K \gg (N - 1)L_mL_n/2$ , the achieved  $D(R)$  becomes very close to that of the ideal joint compression scheme.

In comparison to the independent scheme whose  $D(R)$  decays as  $2^{-\frac{R}{2NK}}$ , if  $K > L_mL_n/2$  then the distortion of the proposed distributed scheme decays at a faster rate. One can, therefore, conclude that if the number of corner points of the polygon exceeds  $L_mL_n/2$ , which can be interpreted as a threshold of the degree of complexity of an





**Figure 7.2:** Plots of  $D(R)$  functions (in log scale) of the proposed distributed semi-parametric, independent and joint compression schemes for four sequences of translating bi-level polygons with different number of corner points  $K$ : (a)  $K < L_m L_n / 2$ ; (b)  $K = L_m L_n / 2$ ; (c)  $K > L_m L_n / 2$  and (d)  $K \gg (N-1)L_m L_n / 2$ . As the complexity of the polygon increases, i.e. as  $K$  increases, the bound  $D_{distr}(R)$  approaches  $D_{joint}(R)$ .

image, then one can achieve a superior  $D(R)$  performance by employing the distributed compression scheme. On the other hand, given a simpler image with smaller  $K$ , an independent scheme gives a better compression result. Figure 7.2 illustrates this finding with the plots of  $D_{ind}(R)$ ,  $D_{distr}(R)$  and  $D_{joint}(R)$  for sequences of a polygon with different level of complexity  $K$ .

## 7.4 Real Objects with Affine Transform Disparity Model

### 7.4.1 Signal Model

Following the analysis of the simplified bi-level polygon model presented in the previous section, let us now consider a sequence that contains a real object in a uniform background whose frame-to-frame disparity can be described by an affine transform as illustrated in Figures 7.5 (a) and 7.6 (a). Let  $\mathbf{x}_i$  denote the coordinate points in the frame  $f_i(x, y)$  and similarly for  $\mathbf{x}_j$ . With the affine transform assumption, the disparity between the  $i^{th}$  frame and the  $j^{th}$  frame is given by:

$$f_j(\mathbf{x}_j) = f_i(A_{ij}\mathbf{x}_i + \boldsymbol{\tau}_{ij}), \quad i \neq j \quad i, j = 1, 2, \dots, N, \quad (7.16)$$

where  $A_{ij}$  is a non-singular affine transformation matrix and  $\boldsymbol{\tau}_{ij}$  is a translation vector. Hence, this disparity model has six parameters where the matrix  $A_{ij}$  can accommodate rotation, shearing and scaling.

### 7.4.2 Estimating Affine Parameters from Moments

In [29], the method to retrieve the matrix  $A_{ij}$  using second and higher order moments was presented. Heikkilä showed that, by using the whitening transform, the estimation of  $A_{ij}$  can be reduced to a problem of finding a rotational matrix  $R$ . We now briefly show the derivation of the result given in [29].

#### Central and complex moments

Let us begin with the definitions of the central moments  $\mu_{p,q}$  and complex moments  $\kappa_{p,q}$  of order  $(p + q)$ . Given a continuous function  $f(x, y)$ . The central moments  $\mu_{p,q}$  are

defined about the barycenter  $(\bar{x}, \bar{y}) = \left(\frac{M_{1,0}}{M_{0,0}}, \frac{M_{0,1}}{M_{0,0}}\right)$  of  $f(x, y)$  as [7]

$$\begin{aligned}
 \mu_{p,q} &= \int \int f(x, y)(x - \bar{x})^p (y - \bar{y})^q dx dy \\
 &= \int \int f(x, y) \left( \sum_{k=0}^p \binom{p}{k} (-\bar{x})^{p-k} x^k \right) \left( \sum_{l=0}^q \binom{q}{l} (-\bar{y})^{q-l} y^l \right) dx dy \\
 &= \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} (-\bar{x})^{p-k} (-\bar{y})^{q-l} M_{k,l}.
 \end{aligned} \tag{7.17}$$

The complex moments  $\kappa_{p,q}$  are defined on the complex plane  $z = x + jy$ ,  $j = \sqrt{-1}$  as [7]

$$\begin{aligned}
 \kappa_{p,q} &= \int \int f(x, y)(x + jy)^p (x - jy)^q dx dy \\
 &= \int \int f(x, y) \left( \sum_{k=0}^p \binom{p}{k} (jy)^{p-k} x^k \right) \left( \sum_{l=0}^q \binom{q}{l} (-jy)^{q-l} x^l \right) dx dy \\
 &= \sum_{k=0}^p \sum_{l=0}^q \binom{p}{k} \binom{q}{l} j^{p-k+q+l} (-1)^{q-l} M_{k+l, p-k+q-l}.
 \end{aligned} \tag{7.18}$$

Therefore, both the central and complex moments can be obtained from the combination of geometric moments.

### Estimating affine transform matrix

Consider a covariance matrix defined by:

$$\Sigma = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{bmatrix},$$

from the affine transform equation given in (7.16), it can be shown that the covariance matrix of  $f_j$ , denoted by  $\Sigma_j$ , can be written as  $\Sigma_j = A_{ij} \Sigma_i A_{ij}^T$ . It has been shown that any two point sets can be matched under an affine transformation if their canonical forms can be matched under rotation alone [29]. A point set is said to be in a canonical form if its covariance is an identity matrix. We can convert a point set into the canonical form using the whitening transform. The method presented in [29] is the Cholesky

factorization:

$$\Sigma = FF^T, \quad F = \begin{bmatrix} \sqrt{\mu_{2,0}} & 0 \\ \frac{\mu_{1,1}}{\sqrt{\mu_{2,0}}} & \sqrt{\mu_{0,2} - \frac{\mu_{1,1}^2}{\mu_{2,0}}} \end{bmatrix}. \quad (7.19)$$

The whitening transform can then be expressed as

$$\mathbf{y} = F^{-1}\bar{\mathbf{x}}, \quad (7.20)$$

where  $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{E}[\mathbf{x}]$  and the covariance matrix of the vector  $\mathbf{y}$  is now an identity matrix i.e.  $\mathbf{E}[\mathbf{y}\mathbf{y}^T] = I$ . By substituting (7.20) into (7.16) we have that

$$\begin{aligned} F_j \mathbf{y}_j &= A_{ij} F_i \mathbf{y}_i \\ F_j F_j^T &\stackrel{(a)}{=} A_{ij} F_i F_i^T A_{ij}^T, \end{aligned} \quad (7.21)$$

where (a) represents the quadratic form. The solution to this quadratic form  $TT^T = SS^T$  is given by  $T = SR$  where  $R$  is an orthogonal matrix [29], which yields

$$A_{ij} = F_j R F_i^{-1}. \quad (7.22)$$

Thus, the problem of finding  $A_{ij}$  can be reduced to a problem of finding the matrix  $R$  and since  $R$  is a  $2 \times 2$  orthogonal matrix, we have that

$$R = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}. \quad (7.23)$$

Heikkilä showed that the matrix  $R$  can be estimated from higher order complex moments of  $\mathbf{y}$ . By substituting (7.22) into (7.21), we have that  $\mathbf{y}_j = R\mathbf{y}_i$ . The complex moments  $\eta_{p,q}$  of order  $(p+q) = 3$  of  $\mathbf{y}$  in polar coordinates is given by

$$\eta_{p,q} = \mathbf{E}[d^{p+q} e^{j(p-q)\theta}], \quad (7.24)$$

where  $d$  is the magnitude of  $\mathbf{y}$  and  $\theta$  is the phase of  $\mathbf{y}$ . Since  $\mathbf{y}_j$  is the rotated version

of  $\mathbf{y}_i$ , we have that

$$\eta_{p,q}^j = \eta_{p,q}^i e^{j(p-q)\alpha}. \quad (7.25)$$

Therefore, the angle of rotation  $\alpha$  can be solved using the complex moments of  $\mathbf{y}$  as follows:

$$\alpha \bmod \left( \frac{2\pi}{p-q} \right) = \left( \frac{\arg(\eta_{p,q}^j) - \arg(\eta_{p,q}^i)}{p-q} \right) \bmod \left( \frac{2\pi}{p-q} \right). \quad (7.26)$$

It was then shown in [29] that  $\eta_{p,q}$  can be calculated from a combination of central moments  $\mu_{p,q}$  using the following formulae:

$$\begin{aligned} \operatorname{Re}\{\eta_{2,1}\} &= (\mu_{3,0}\mu_{0,2} - 2\mu_{2,1}\mu_{1,1} + \mu_{2,0}\mu_{1,2}) \mu_{2,0}^{\frac{1}{2}} \kappa^{-1}; \\ \operatorname{Im}\{\eta_{2,1}\} &= (-\mu_{3,0}\mu_{1,1}\mu_{0,2} + \mu_{2,1}\mu_{2,0}\mu_{0,2} + 2\mu_{1,1}^2\mu_{2,1} - 3\mu_{1,1}\mu_{2,0}\mu_{1,2} + \mu_{2,0}^2\mu_{0,3}) \mu_{2,0}^{\frac{1}{2}} \kappa^{\frac{3}{2}}; \\ \operatorname{Re}\{\eta_{3,0}\} &= (\mu_{3,0}\mu_{0,2}\mu_{2,0} - 4\mu_{3,0}\mu_{1,1}^2 + 6\mu_{2,1}\mu_{1,1}\mu_{2,0} - 3\mu_{2,0}^2\mu_{1,2}) \mu_{2,0}^{-\frac{3}{2}} \kappa^{-1}; \\ \operatorname{Im}\{\eta_{3,0}\} &= (-3\mu_{3,0}\mu_{1,1}\mu_{0,2}\mu_{2,0} + 4\mu_{3,0}\mu_{1,1}^2 + 3\mu_{2,1}\mu_{2,0}^2\mu_{0,2} - 6\mu_{1,1}^2\mu_{2,1}\mu_{2,0} \\ &\quad + 3\mu_{1,1}\mu_{2,0}^2\mu_{1,2} - \mu_{2,0}^3\mu_{0,3}) \mu_{2,0}^{-\frac{3}{2}} \kappa^{-\frac{3}{2}} \end{aligned} \quad (7.27)$$

with  $\kappa = \mu_{2,0}\mu_{0,2} - \mu_{1,1}^2$ . Here,  $\operatorname{Re}\{\cdot\}$  and  $\operatorname{Im}\{\cdot\}$  refer to the real and imaginary part of the complex moment.

In summary, the affine transformation matrix can be retrieved from a set of geometric moments with the following algorithm:

**Algorithm 6 :** *Calculation of the affine transformation matrix from geometric moments.*

1. **Geometric moments:** the geometric moments  $M_{p,q}$ ,  $(p+q) = 0, \dots, 3$ , of  $f_i(\mathbf{x}_i)$  and  $f_j(\mathbf{x}_j)$  are obtained;
2. **Central moments:** the central moments  $\mu_{p,q}$ ,  $(p+q) = 0, \dots, 3$ , of  $f_i(\mathbf{x}_i)$  and  $f_j(\mathbf{x}_j)$  are calculated using (7.17);
3. **Complex moments:** the corresponding complex moments  $\eta_{p,q}^i$  and  $\eta_{p,q}^j$ ,  $(p+q) = 3$ , are then retrieved with the set of formulae given in (7.27);
4. **Rotation matrix:** the rotation matrix  $R$  can then be obtained using (7.26) and (7.23);
5. **Affine transform matrix:** the affine transformation matrix  $A_{ij}$  is retrieved from (7.19) and (7.22).

A distributed semi-parametric compression scheme based on the above algorithm is presented next.

### 7.4.3 Distributed Compression Scheme

We can now devise a distributed compression scheme for a sequence with the affine transform disparity model. It is also assumed here that the decoder only uses the low-pass coefficients to estimate the affine transform matrix. In order to control the number of bits allocated to the low-pass coefficients  $c_{m,n}^J$  in relation to the high-pass coefficients  $d_{m,n}^j$ , we introduce a scaling parameter  $\lambda \in \mathbb{Z}$  such that  $c_{m,n}^J$  is scaled by

$$c_{m,n}^{J*} = c_{m,n}^J 2^\lambda$$

before being transmitted by the encoder. Here, a positive  $\lambda$  represents the bit shift to the left. The proposed algorithm is as follows:

**Algorithm 7 :** *Distributed semi-parametric compression algorithm for the sequence with a real object undergoing affine transform.*

Encoding and Decoding of the Key Frame  $f_1(x, y)$

1. **Nonlinear approximation-based compression:**  $f_1(x, y)$  is encoded and decoded with a conventional wavelet nonlinear approximation-based compression scheme.

Encoding of  $f_i(x, y)$ ,  $i = 2, \dots, N$

1. **Scaling of low-pass coefficients:** the low-pass coefficients  $\{c_{m,n}^J\}$  of  $f_i(x, y)$  are scaled as follows:

$$c_{m,n}^{J*} = c_{m,n}^J 2^\lambda, \quad \lambda \in \mathbb{Z},$$

and the parameter  $\lambda$  is transmitted to the decoder;

2. **Nonlinear approximation-based compression:** the coefficients  $\{c_{m,n}^{J*}\}_{(m,n) \in \mathcal{I}_N}$  and  $\{d_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}$  of  $f_i(x, y)$  are then encoded and with a conventional wavelet nonlinear approximation-based compression scheme.

Joint Decoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. **Parametric estimation:** the decoder estimates the affine transform matrix  $A_i$  and the translation vector  $\tau_i$ , from the received quantized low-pass coefficients  $\{\bar{c}_{m,n}^J = \bar{c}_{m,n}^{J*}/2^\lambda\}$ ;

2. **Prediction with affine transform:** the prediction of  $f_i(x, y)$  is formed by

$$\tilde{f}_i(\mathbf{x}) = \hat{f}_1(\hat{A}_i \mathbf{x} - \hat{\tau}_i)$$

and the predicted coefficients  $\tilde{c}_{m,n}^J = \langle \tilde{f}_i(x, y), \varphi_{m,n}^J(x, y) \rangle$  and  $\tilde{d}_{m,n}^j = \langle \tilde{f}_i(x, y), \psi_{m,n}^j(x, y) \rangle$  are obtained;

3. **Final reconstruction:** the decoder reconstructs  $f_i(x, y)$  by taking the inverse wavelet transform of the following set of coefficients:

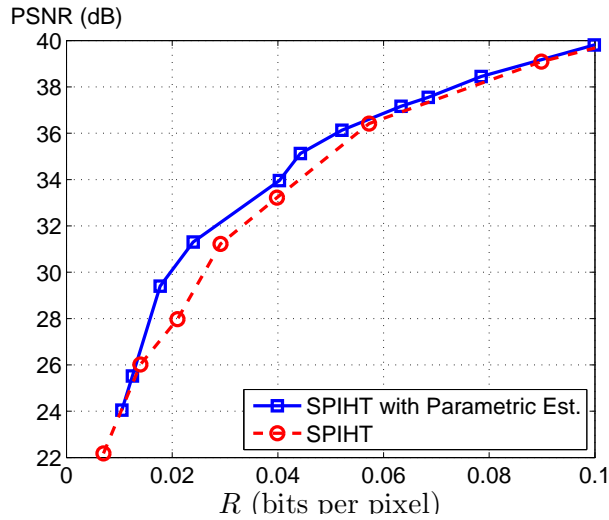
$$\{\tilde{c}_{m,n}^J\}_{(m,n) \in \mathcal{I}_N}, \{\tilde{d}_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}, \{\tilde{c}_{m,n}^J\}_{(m,n) \notin \mathcal{I}_N} \text{ and } \{\tilde{d}_{m,n}^j\}_{(j,m,n) \notin \mathcal{I}_N}.$$

The parametric estimation step of the above compression scheme can be done with Algorithm 6. This allows the complex task of motion estimation to be implemented at the decoder. As with before, we can use the moment-samples relationship from the sampling theory of FRI signals to estimate the geometric moments from the quantized low-pass coefficients as shown in (7.3). Note that, in our work, the bit allocation strategy between the key frame and the non-key frames is done using a greedy strategy, meaning that an additional bit is given to the frame that improves the overall PSNR the most.

#### 7.4.4 Simulation Results

We now present the simulation results of the proposed scheme, where the parametric estimation step is implemented with Algorithm 6. The SPIHT algorithm [43] is used to perform the nonlinear compression of each frame. We use the wavelet transform with a Daubechies 4 scaling function to decompose each frame up to four decomposition levels. In order to gauge the performance, we compare both the PSNR plot and the visual quality obtained from our scheme against an independent scheme where each frame is encoded and decoded with SPIHT.

Figure 7.5 (a) shows the first sequence used in our simulation, which consists of four images. We find that the effective range of the low-pass scaling parameter is  $-2 \leq \lambda \leq 1$ . The plot of the PSNR against the total rate  $R$  in bits per pixel (bpp) is given in Figure 7.3. From the plot, the proposed scheme outperforms the independent scheme by approximately 1 to 2 dB at lower rates. Consider a point at 0.04 bpp, the proposed distributed scheme achieves the PSNR of 33.96 dB  $\lambda = 1$ , whereas the

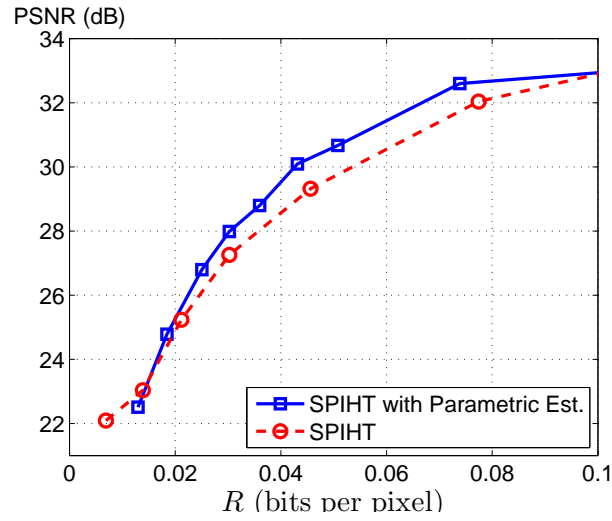


**Figure 7.3:** The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 7. The input sequence shown is shown in Figure 7.5 (a).

independent scheme achieves 33.22 dB at the same rate. The reconstructed sequences by the two schemes at these two rate points are displayed in Figure 7.5 (b) and (c). It is clear that the images compressed by the proposed scheme are sharper. In fact, the difference lies in the way the bits are allocated. With our scheme, the key frame  $f_1(x, y)$  is allocated 20,285 bits and  $\{f_i(x, y)\}_{i=2,3,4}$  take, on average, 7,320 bits per frame. Joint decoding then allows the details of the first frame to be used for the prediction of the coefficients in other frames. This in turn improves the overall visual quality of the reconstructed sequence. Independent encoding, on the other hand, allocates the bits to each frame equally.

The second testing sequence is shown in Figure 7.6 (a). At lower rates, setting  $-2 \leq \lambda \leq 0$  gives a better performance. Since this sequence contains more details when compared to the previous one, it is beneficial to allocate more bits to the high-pass coefficients while letting the low-pass coefficients be predicted from the key frame. The corresponding PSNR plot is given in Figure 7.4. As with the previous simulation, the proposed scheme outperforms the independent scheme by approximately 1 to 2 dB at lower rates. At 0.043 bpp, the proposed distributed scheme achieves the PSNR of 30.1 dB with  $\lambda = -2$ , whereas the independent scheme achieves 29.3 dB at 0.045 bpp. The reconstructed sequences are shown in Figure 7.6 (b) and (c). It is also worth noting that for this sequence, at the rate point shown, the key frame receives 8,176 bits while





**Figure 7.4:** The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 7. The input sequence shown is shown in Figure 7.6 (a).

other frames receive 9,265 bits per frame on average. Even though the rate allocation is almost equally distributed amongst the frames, by using joint decoding and prioritizing the transmitted coefficients differently (i.e. giving more bits to the low-pass in the key frame while focusing on the high-pass in other frames) the overall performance can be improved.

1<sup>st</sup> Frame



4<sup>th</sup> Frame



(a) original sequence;



(b) distributed semi-parametric compression;



(c) independent SPIHT algorithm.

**Figure 7.5:** Illustration of the compression of the first sequence with a real object whose disparity is described by the affine transform: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 7 at 0.04 bpp with the PSNR of 333.96 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.04 bpp with the PSNR of 33.22 dB.

1<sup>st</sup> Frame



4<sup>th</sup> Frame



(a) original sequence;



(b) distributed semi-parametric compression;



(c) independent SPIHT algorithm.

**Figure 7.6:** Illustration of the compression of the second sequence with a real object whose disparity is described by the affine transform: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 7 at 0.043 bpp with the PSNR of 30.1 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.045 bpp with the PSNR of 29.3 dB.

## 7.5 Applications to Real Signals

In this section, we show the potential applications of the distributed semi-parametric compression schemes presented in the previous sections. We consider two types of signals: a set of images obtained from an array of two cameras and a video sequence with a fixed background.

### 7.5.1 Array of Images

Let us first consider a simple set of  $N$  images  $f_i(x, y)$ ,  $i = 1, 2, \dots, N$ , captured by an array of cameras where the disparity between each image can be approximated with

$$f_i(\mathbf{x}) \approx f_1(\mathbf{x} - \boldsymbol{\tau}_i),$$

with  $\tau_i$  denoting a translation vector. Examples of these images are shown in Figures 7.9 (a) and 7.10 (a).

### Compression algorithm

We propose the following semi-parametric compression algorithm:

**Algorithm 8 :** *Distributed semi-parametric compression algorithm for an array of images.*

Encoding and Decoding of the Key Image  $f_1(x, y)$

1. **Nonlinear approximation-based compression:**  $f_1(x, y)$  is encoded and decoded with a conventional wavelet nonlinear approximation-based compression scheme.

Encoding of  $f_i(x, y)$ ,  $i = 2, \dots, N$

1. **Scaling of low-pass coefficients:** the low-pass coefficients  $\{c_{m,n}^J\}$  of  $f_i(x, y)$  are scaled as follows:

$$c_{m,n}^{J*} = c_{m,n}^J 2^\lambda, \quad \lambda \in \mathbb{Z},$$

and the parameter  $\lambda$  is transmitted to the decoder. Here we assume that the value of  $\lambda$  is selected a priori based on empirical results that give the lowest distortion-rate performance;

2. **Nonlinear approximation-based compression:** the coefficients  $\{c_{m,n}^{J*}\}_{(m,n) \in \mathcal{I}_N}$  and  $\{d_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}$  of  $f_i(x, y)$  are then encoded and with a conventional wavelet nonlinear approximation-based compression scheme.

Joint Decoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. **Parametric estimation:** the decoder estimates the translation vector  $\tau_i$  from the received quantized coefficients  $\{\bar{c}_{m,n}^J = \bar{c}_{m,n}^{J*}/2^\lambda\}$  and  $\{\bar{d}_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}$ ;
2. **Prediction with translation:** the prediction of  $f_i(x, y)$  is formed by

$$\tilde{f}_i(\mathbf{x}) = \hat{f}_1(\mathbf{x} - \hat{\tau}_i)$$

and the predicted coefficients  $\tilde{c}_{m,n}^J = \langle \tilde{f}_i(x, y), \varphi_{m,n}^J(x, y) \rangle$  and  $\tilde{d}_{m,n}^j = \langle \tilde{f}_i(x, y), \psi_{m,n}^j(x, y) \rangle$  are obtained;

3. **Final reconstruction:** the decoder reconstructs  $f_i(x, y)$  by taking the inverse wavelet transform of the following set of coefficients:

$$\{\bar{c}_{m,n}^J\}_{(m,n) \in \mathcal{I}_N}, \quad \{\bar{d}_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}, \quad \{\tilde{c}_{m,n}^J\}_{(m,n) \notin \mathcal{I}_N} \quad \text{and} \quad \{\tilde{d}_{m,n}^j\}_{(j,m,n) \notin \mathcal{I}_N}.$$

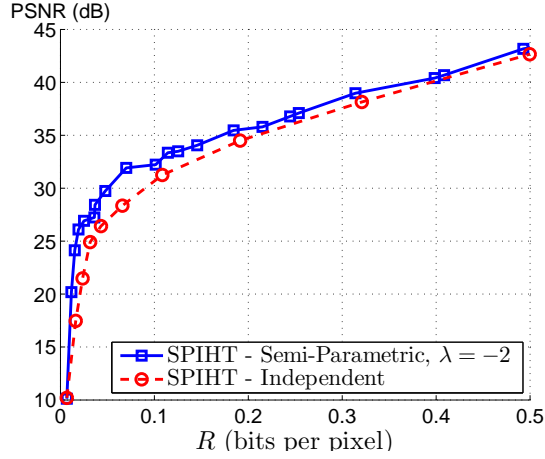
As with before, the scaling of low-pass coefficients allows the encoder to control the rate allocation. In our work, the parametric estimation of the translation vector is implemented with a block-based approach. First, the decoder obtains the function  $\bar{f}_i(x, y)$  by taking the inverse wavelet transform of the received coefficients  $\bar{c}_{m,n}^J$  and  $\bar{d}_{m,n}^j$ . The resulting image  $\bar{f}_i(x, y)$  is then divided into  $K$  disjoint blocks  $\{B_k\}_{1 \leq k \leq K}$  and the decoder searches for the translation vector  $\hat{\tau}_{i,k}$  to minimize the following:

$$\min_{\hat{\tau}_{i,k} \in S_k} \left\| \hat{f}_1(\mathbf{x}) - \bar{f}_i(\mathbf{x} + \hat{\tau}_{i,k}) \right\|^2, \quad \mathbf{x} \in B_k, \quad k = 1, 2, \dots, K,$$

where  $S_k$  is the search range. The vector  $\hat{\tau}_i$  can then be obtained by taking the average of  $\hat{\tau}_{i,k}$ . Clearly, one can also form the prediction  $\tilde{f}_i(\mathbf{x})$  directly by shifting the blocks.

### Simulation results

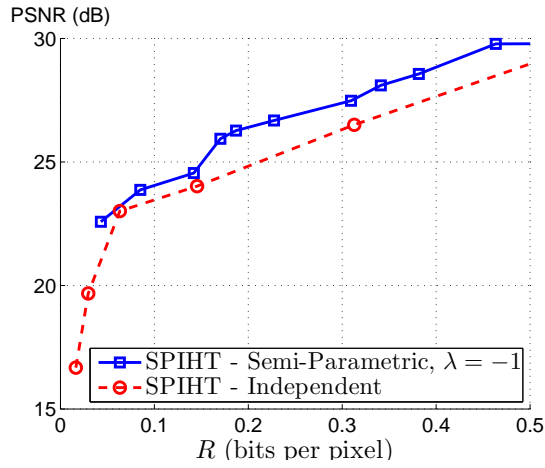
Let us now present the simulation results of the proposed scheme. Here, we consider a set of two images. We apply the biorthogonal2.2 wavelet transform up to four decomposition levels and encode each frame with the SPIHT algorithm [43]. An independent encoding



**Figure 7.7:** The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 8. The input sequence shown is shown in Figure 7.9 (a).

scheme where each frame is encoded and decoded with SPIHT is used to compare the performance of the proposed scheme.

Figure 7.9 (a) shows the first set of images used in our simulation. For the estimation of the translation vector using the block-based approach described above, we found that by allocating more bits to the high pass coefficients i.e. setting  $\lambda < 0$ , a more accurate prediction is obtained. This is because block-matching method works more efficiently when the information about the edges of the image is available. The plot of the PSNR against the total rate  $R$  is given in Figure 7.7. At lower bit rates, our proposed scheme outperforms the independent scheme by approximately up to 3 dB with  $\lambda = -2$ . At 0.07 bpp, the proposed distributed scheme achieves the PSNR of 31.9 dB and the independent scheme achieves 28.35 dB at 0.066 bits. Figure 7.9 (b) and (c) show the reconstructed images by the two schemes at these two rate points. We can see that the reconstructed images by the proposed scheme are sharper. At this rate point, the first image receives 0.11 bpp and the second image gets 0.029 bpp.



**Figure 7.8:** The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 8. The input sequence shown is shown in Figure 7.10 (a).

Figure 7.8 shows the PSNR plots for the second set of images as illustrated in Figure 7.10 where we set  $\lambda = -1$ . A similar gain in performance is also observed here. At 0.34 bpp, the proposed scheme achieves the PSNR of 28.1 dB. Here the first and second images are encoded at 0.63 bpp and 0.055 bpp respectively. In contrast, the independent scheme achieves the PSNR of 26.5 dB at 0.31 bpp with equally distributed rate allocation.

1<sup>st</sup> Image



2<sup>nd</sup> Image



(a) original set of images;



(b) distributed semi-parametric compression;



(c) independent SPIHT algorithm.

**Figure 7.9:** The first illustration of the compression of an array of images whose disparity is described by a translation vector. (a) the original images; (b) the reconstructed images using the proposed distributed semi-parametric compression scheme at 0.07 bpp with the PSNR of 31.9 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.066 bpp with the PSNR of 28.35 dB.



1<sup>st</sup> Image

2<sup>nd</sup> Image


(a) original set of images;



(b) distributed semi-parametric compression;



(c) independent SPIHT algorithm.

**Figure 7.10:** The second illustration of the compression of an array of images whose disparity is described by a translation vector. (a) the original images; (b) the reconstructed images using the proposed distributed semi-parametric compression scheme at 0.34 bpp with the PSNR of 28.1 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.31 bpp with the PSNR of 26.5 dB.

### 7.5.2 Real Video Sequence with Fixed Background

Our aim here is to apply the framework presented in Section 7.4 to a more realistic video sequence with a fixed background. We used the ‘highway sequence’<sup>1</sup> in our experiment as shown in Figure 7.13 (a), where each frame is  $512 \times 512$  in resolution. In this sequence, the motion of each object or each car can be modeled with a simplified affine transform that involves only translation and scaling i.e. the affine transform matrix is diagonal. Let us denote with  $\mathcal{O}_{i,k}$ ,  $k = 1, 2, \dots, N$ , a set of coordinates of each object in  $f_i(x, y)$ . We can, therefore, obtain the following approximation:

$$f_i(\mathbf{x}_i) \approx f_1(A_{i,k}\mathbf{x}_1 - \boldsymbol{\tau}_{i,k}), \quad \forall \mathbf{x}_i \in \mathcal{O}_{i,k} \quad \text{and} \quad \forall \mathbf{x}_1 \in \mathcal{O}_{1,k} \quad \text{with} \quad k = 1, 2, \dots, N$$

where  $A_{i,k}$  is the affine transform matrix and  $\boldsymbol{\tau}_{i,k}$  is the translation vector that describes the disparity of the  $k$ -th object between the first frame and the  $i$ -th frame.

#### Compression algorithm

In our algorithm, we divide the sequence into groups of pictures (GOP), where each GOP has  $N$  frames, which are denoted by  $f_i(x, y)$ ,  $i = 1, 2, \dots, N$ . The first frame of the GOP  $f_1(x, y)$  is referred to as the key frame. It is also assumed here that the decoder only uses the low-pass coefficients to perform the parametric estimation. We now propose the following distributed semi-parametric compression scheme:

**Algorithm 9 :** *Distributed semi-parametric compression algorithm for each GOP in the highway sequence.*

Encoding and Decoding of the Key Frame  $f_1(x, y)$

1. **Nonlinear approximation-based compression:**  $f_1(x, y)$  is encoded and decoded with a conventional wavelet nonlinear approximation-based compression scheme.

Encoding of  $f_i(x, y)$ ,  $i = 2, \dots, N$

1. **Nonlinear approximation-based compression:** the coefficients  $\{c_{m,n}^J\}_{(m,n) \in \mathcal{I}_N}$  and  $\{d_{m,n}^j\}_{(j,m,n) \in \mathcal{I}_N}$  of  $f_i(x, y)$  are encoded with a conventional wavelet nonlinear approximation-based compression scheme.

---

<sup>1</sup>Courtesy of the ACTS Project AC304 MODEST.

Joint Decoding of  $f_i(t)$ ,  $i = 2, \dots, N$

1. **Background extraction:** the decoder extracts the background  $g(x, y)$  from previously decoded key frames;
2. **Object segmentation:** with the extracted background, the decoder apply segmentation to the key frame to obtain a set of objects  $\{f_1(x, y)\}_{x, y \in \mathcal{O}_{1,k}}$ ,  $k = 1, 2, \dots, K$ ;
3. **Low-pass coefficients segmentation:** by using the low-pass coefficients of extracted background  $\langle g(x, y), \varphi_{m,n}^J(x, y) \rangle$ , the decoder segments the received low-pass coefficients  $\bar{c}_{m,n}^J$  to form a set of low-pass coefficients of each object  $\{\bar{c}_{m,n}^J\}_{m,n \in \mathcal{O}_{i,k}^J}$ ,  $k = 1, 2, \dots, K$ , where  $\mathcal{O}_{i,k}^J$  denotes a set of coordinates of the low-pass coefficients of the  $k$ -th object in  $f_i(x, y)$ ;
4. **Parametric estimation:** for each object, the decoder estimates the affine transform matrix  $A_{i,k}$  and the translation vector  $\tau_{i,k}$  from  $f_1(x, y)$  with  $(x, y) \in \mathcal{O}_{1,k}$  and  $\bar{c}_{m,n}^J$  with  $(m, n) \in \mathcal{O}_{i,k}^J$ ,  $k = 1, 2, \dots, K$ ;
5. **Prediction with affine transform:** the prediction of  $f_i(x, y)$  is formed as

$$\tilde{f}_i(\mathbf{x}) = \hat{f}_1(\hat{A}_{i,k}\mathbf{x} - \hat{\tau}_{i,k}), \quad \forall \mathbf{x}_i \in \mathcal{O}_{i,k} \quad \text{and} \quad \forall \mathbf{x}_1 \in \mathcal{O}_{1,k}, \quad k = 1, 2, \dots, K$$

and

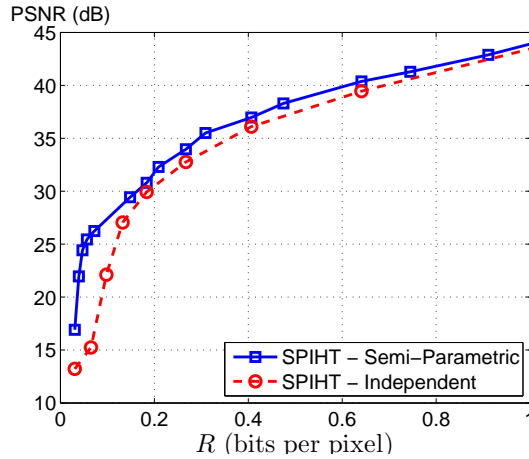
$$\tilde{f}_i(x, y) = g(x, y), \quad \forall (x, y) \notin \mathcal{O}_{i,k}, \quad k = 1, 2, \dots, K.$$

The predicted coefficients  $\{\bar{c}_{m,n}^J\}$  and  $\{\tilde{d}_{m,n}^J\}$  of  $\tilde{f}_i(x, y)$  are then obtained;

6. **Final reconstruction:** the decoder reconstructs  $f_i(x, y)$  by taking the inverse wavelet transform of the following set of coefficients:

$$\{\bar{c}_{m,n}^J\}_{(m,n) \in \mathcal{J}_N}, \quad \{\tilde{d}_{m,n}^J\}_{(j,m,n) \in \mathcal{J}_N}, \quad \{\tilde{c}_{m,n}^J\}_{(m,n) \notin \mathcal{J}_N} \quad \text{and} \quad \{\tilde{d}_{m,n}^J\}_{(j,m,n) \notin \mathcal{J}_N}.$$

The estimation of the affine transform matrix can be implemented with Algorithm 6 and the translation vector can be obtained from (7.2). In this case, however, a simpler estimation method of the matrix  $A_{i,k}$  can also be used if we assume that the decoder has prior knowledge about the nature of the disparity between each objects. More specifically, as the object moves higher along the  $y$ -axis, its size decreases. Therefore, the decoder can be trained to calculate the scaling factor from the relative position of the object and the retrieved translation vector.



**Figure 7.11:** The plot of the PSNR against the bit rate (in bits per pixel) for the proposed compression scheme in Algorithm 9. The input sequence is shown in Figure 7.13 (a).

### Simulation results

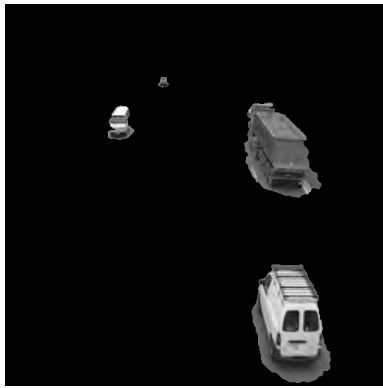
We now show the simulation results of the proposed distributed semi-parametric compression scheme above. The GOP size is set to four frames and we compress a set of images extracted from the first GOP. Once again, we use the SPIHT algorithm [43] to perform the nonlinear compression. The wavelet transform with a Daubechies 4 scaling function is used to decompose each frame up to three decomposition levels. A greedy bit allocation strategy is also used here. We compare the PSNR and the visual quality obtained from our scheme with an independent scheme where each frame is encoded and decoded with SPIHT.

Figure 7.11 shows the plot of the PSNR against the total rate  $R$ . The proposed scheme outperforms the independent scheme by approximately 1 to 2 dB. At 0.2 bpp, the proposed scheme achieves the PSNR of 32.3 dB, whereas the independent scheme achieves 29.9 dB at 0.18 bpp. The extracted background as well as the segmentation of the objects and their low-pass coefficients are shown in Figure 7.12. Figure 7.13 (b) and (c) show the reconstructed sequence by the proposed scheme and an independent compression scheme. Notice that the objects in the reconstructed images obtained from our scheme are sharper. This is due to the use of parametric estimation during joint decoding, which allows the decoder to predict the wavelet coefficients of each object from the key frame. Note that the overall PSNR also improves because of the use of the extracted background. At this rate point, the encoder allocates 0.3 bpp for the key

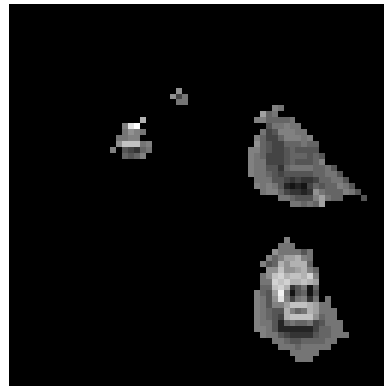
frame and 0.18 bpp for the other frames.



(a)



(b)



(c)

**Figure 7.12:** Illustration of the segmentation results of the key frame and the corresponding low-pass coefficients: (a) the extracted background; (b) the segmentation of the key frame; (c) the segmentation of the low-pass coefficients in the second frame.

1<sup>st</sup> Frame



2<sup>nd</sup> Frame



(a) original sequence;



(b) distributed semi-parametric compression;



(c) independent SPIHT algorithm.

**Figure 7.13:** Illustration of the compression of the highway sequence: (a) the original sequence; (b) the reconstructed sequence using the proposed distributed semi-parametric compression scheme in Algorithm 9 at 0.2 bpp with the PSNR of 32.3 dB; (c) the reconstructed sequence using the independent SPIHT algorithm at 0.18 bpp with the PSNR of 29.9 dB.

## 7.6 Conclusion

In this chapter, we have presented a new approach to distributed compression based on the concept of joint decoding with parametric estimation. We first considered a synthetic video sequence, which consists of a translating bi-level polygon. The compression algorithm inspired by the sampling theory of FRI signals was presented, where we also analyzed the effect of quantization of the coefficients on the retrieved moments. Our analysis shows that as the complexity of the polygon increases, the gap between the performance of the proposed scheme and the ideal joint encoding scheme narrows. On the other hand, for a simple polygon, an independent encoding-decoding scheme can give a better performance.

We then considered a sequence of real object undergoing an affine transformation in a uniform background. Based on the work of [29], we described a way to estimate the affine transform matrix from the low-pass coefficients. A distributed compression scheme based on this result was then proposed. Our simulation results show that improvements in terms of the PSNR and the visual quality are observed with the proposed scheme when compared to a scheme with independent decoding.

Lastly, we presented two compression algorithms for a set of real images taken from an array of camera and a video sequence with a fixed background. Both compression schemes outperform the independent encoding-decoding schemes in terms of the PSNR and visual quality. The differences are more pronounced at lower bit rates. This is because the process of parametric estimation allows the decoder to better predict the wavelet coefficients using the information from the decoded key frame.

# CHAPTER 8

## Conclusions

### 8.1 Thesis Summary

The objective of the research presented in this thesis is to develop a new approach to centralized and distributed compression using wavelets. We first approach the problem from a theoretical point of view by using a piecewise smooth function as the signal model. Our goal is to provide a precise set of answers to the following questions: given that the complexity is shifted from the encoder to the decoder, can the same distortion-rate performance be achieved? When the piecewise smooth function is partially observed by a number of independent encoders, can we still obtain the distortion-rate performance that is comparable to that of a joint encoding scheme? Finally, how can we apply this new framework in the context of distributed image and video compression?

#### **Centralized semi-parametric compression: a scheme with a linear encoder and a nonlinear decoder**

A new semi-parametric compression algorithm for piecewise smooth functions has been proposed. This algorithm reflects the shift of the computational complexity from the encoder to the decoder. The encoder of the proposed algorithm uses a wavelet-based linear approximation strategy. The decoder is, instead, nonlinear and employs a parametric estimation technique to reconstruct the singular structure of the observed function.



This enables it to predict the wavelet coefficients in the cone of influence of the discontinuities. Our analysis shows that the distortion-rate function of the proposed scheme achieves a dominating decay rate of  $R^{-2\alpha}$  for a wide range of rates, which is comparable to that of a conventional compression scheme with a nonlinear encoder and a linear decoder. A practical parametric estimation algorithm, which applies the results of the new sampling theory of FRI signals, has also been presented.

### **Distributed semi-parametric compression: a scheme with independent encoders and a joint decoder**

We have extended the concept of semi-parametric compression to devise a new distributed compression scheme. We modeled the disparity between each observed signal with a shift and a prediction error. The decoder of the proposed scheme uses a parametric estimation technique to retrieve the locations of discontinuities in each signal and, in turn, calculate the shift parameter. We also show that the prediction error can be transmitted to the decoder where the number of bits required depends on the quality of the prediction formed by the decoder. The distortion-rate analysis shows that the proposed scheme can achieve a compression performance comparable to that of a centralized joint encoding scheme for a wide range of rates. The gain in performance relative to the independent encoding and decoding scheme has also been determined. In the proposed scheme, the rate allocation depends on the power of the prediction error, which has to be estimated a priori. Lastly, there is no change to the structure of the wavelet transform in our scheme. Instead, the quantization strategy and the bit allocation are different from the centralized case, where both depend on the power of the prediction error.

### **Applications of semi-parametric compression: a new approach to distributed image and video compression**

The proposed distributed semi-parametric compression framework has been applied in the context of distributed compression of images and videos. First, a toy model of a video sequence has been constructed and the semi-parametric compression algorithm

inspired by the FRI sampling theory was presented. Our distortion-rate analysis shows that it is more advantageous to employ joint decoding as the complexity of the polygon increases. The second model consists of a real object in a uniform background whose motion is described by an affine transform. For this model, we devise a compression scheme whose decoder is able to estimate the affine transform matrix from the received scaling coefficients by using the results from FRI theory and the moments-based registration technique presented in [29]. Finally, we present two practical compression algorithms for a set of real images taken from an array of cameras and a video sequence with a fixed background. The simulation results show that all of our proposed schemes can outperform the independent encoding-decoding schemes, both, in terms of the PSNR and the visual quality, where differences are more pronounced at lower bit rates. The reason behind such improvement lies in the use of parametric estimation, which allows the decoder to predict the unknown wavelet coefficients from the previously decoded reference images. In addition, the bit allocation of our schemes is generally non-symmetric.

## 8.2 Future Work

The concept of semi-parametric compression presented in this thesis leads to a rather different approach to compression. To this end, there are still many problems that remain largely open for future research.

### **Parametric estimation from scaling and wavelet coefficients**

One of the key features of the proposed compression scheme is the use of parametric estimation at the decoder. In this thesis, we have presented a practical parametric estimation algorithm based on the sampling theory of FRI signals. It was shown in Chapter 5, however, that the proposed algorithm does not achieve the CRB and is, therefore, not optimal. Therefore, one can improve the compression performance of the semi-parametric compression scheme further by developing a better parametric estimation algorithm. In addition, because the information of the discontinuities is well captured in the wavelet coefficients, an algorithm that also uses both the scaling and wavelet coefficients is more likely to produce a more accurate result. Thus, the development of

such algorithm will be an important step in the future research of this topic.

### **Rate allocation and quantization strategies**

In the practical compression schemes presented in Chapter 7, the scaling and wavelet coefficients are used in the parametric estimation step during the decoding procedure. Thus, the fact that the largest coefficients should receive more bits may no longer hold. In Chapter 7, we carry out a preliminary test by introducing the scaling parameter  $\lambda$ , which scales the low-pass coefficients by  $2^\lambda$  prior to being coded with a standard algorithm such as SPIHT. Generally speaking, in addition to the energy contained within the coefficients, the bit allocation should also consider their use in the parametric estimation, which allows for the prediction of other absent coefficients. Therefore, a standard encoding algorithm such as SPIHT may not be the most suitable algorithm for our approach. Hence, the future work will also focus on the development of a more suitable encoding algorithm.

### **The directional transforms**

The decoding process proposed in this thesis is not limited to the use of the wavelet transform. It is now well-known that the wavelet transform is not the most efficient transform to represent the discontinuities along the curves found in natural images. Thus, other transforms that take into account the directionality of the discontinuities have been proposed. An example of such transforms is the curvelet transform [54]. It is, therefore, interesting to see whether one can improve the performance of a compression scheme based on such directional transforms by integrating parametric estimation into the decoding procedure.

### **Beyond the piecewise smooth model**

The theoretical results presented in this thesis is based on the piecewise smooth model. In reality, the increase in applications with high-definition images means that textures will become a vital part in compression problems. The discontinuity caused by the boundary of an object, however, remains an important feature in images. Moreover,

the locations of such discontinuities can still be approximated with a parametric representation. Hence, for our future work, we look to apply the concept of joint decoding with parametric estimation in a distributed compression of signals that are not piecewise smooth but still contain some discontinuities. In this case, it is also interesting to investigate how the prediction of different textures from previously decoded images can be achieved.

# Appendix

## A.1 Proof of Theorem 10

We prove here that, given a function  $f(t)$ ,  $t \in [0, 1[$ , whose wavelet coefficients decay as  $d_{j,n} \sim 2^{j(\alpha+1/2)}$  across scales, the distortion of a wavelet-based compression scheme that uses a linear approximation strategy is given by

$$D(R) \leq c_1 R^{-2\alpha}. \quad (\text{A.1.1})$$

**Proof:** Assume that the compression algorithm keeps all the coefficients from decomposition level  $J_N$  onwards with  $J_N < 0$ . Since  $d_{j,n} \sim 2^{j(\alpha+1/2)}$ , this is equivalent to setting the step size of the quantizer  $\Delta$  to be

$$\Delta = c_{11} 2^{J_N(\alpha+1/2)}. \quad (\text{A.1.2})$$

The distortion  $D$ , measured by MSE, has two components:

$$D = D_1 + D_2,$$

where  $D_1$  is due to the discarding of coefficients and  $D_2$  is due to quantization of the retained coefficients. Let  $N_j$  denotes total number of the wavelet coefficients at level  $j$ . Since the wavelet function  $\psi(t)$  and the corresponding

scaling function  $\varphi(t)$  form the basis of  $\mathbf{L}_2[0, 1]$ , then we have that

$$N_j = 2^{-j}. \quad (\text{A.1.3})$$

It then follows that

$$\begin{aligned} D_1 &= \sum_{j=-\infty}^{J_N-1} \sum_{n=0}^{N_j} \langle f(t), \psi_{j,n}(t) \rangle^2 \\ &\stackrel{(a)}{\leq} \sum_{j=-\infty}^{J_N-1} \sum_{n=0}^{N_j} C_1 2^{2j(\alpha+1/2)} \\ &\stackrel{(b)}{=} C_1 \sum_{j=-\infty}^{J_N-1} 2^{-j} 2^{2j(\alpha+1/2)} \\ &= C_1 \sum_{j=-\infty}^{J_N-1} 2^{j^{2\alpha}} \\ &\approx C_1 2^{(J_N-1)^{2\alpha}} \\ &\stackrel{(c)}{=} C_1 (C' R^{-1})^{2\alpha} \\ &= C_2 R^{-2\alpha}, \end{aligned}$$

where (a) follows the fact that  $d_{j,n} \sim 2^{j(\alpha+1/2)}$ , (b) from (A.1.3) and (c) from the fact that the total rates  $R$  is proportional to the total number of coefficients  $N_{J_N}$ :  $R = C(N_{J_N}) = 2C(N_{J_N-1}) = C'' 2^{-(J_N-1)}$ . The distortion  $D_2$  is given by the sum of the variance of the quantization noise:

$$\begin{aligned} D_2 &= \sum_{j=J_N}^{\infty} \sum_{n=0}^{N_j} \frac{\Delta^2}{12} \\ &\stackrel{(a)}{=} C 2^{2J_N(\alpha+1/2)} \sum_{j=J_N}^{\infty} N_j \\ &\stackrel{(b)}{=} C 2^{2J_N(\alpha+1/2)} \sum_{j=J_N}^{\infty} 2^{-j} \\ &\approx C 2^{2J_N(\alpha+1/2)} 2^{-J_N} \\ &= C' 2^{J_N^{2\alpha}} \\ &\stackrel{(c)}{=} C' (C'' R^{-1})^{2\alpha} \\ &= C_3 R^{-2\alpha}, \end{aligned}$$

where (a) follows from (A.1.2), (b) from (A.1.3) and (c) from the fact that  $R = C(N_{J_N}) = C 2^{-J_N}$ . Therefore, we have that

$$D = C_2 R^{-2\alpha} + C_3 R^{-2\alpha} = c_1 R^{-2\alpha},$$

which proves the equation (A.1.1). ■

## A.2 Evaluation of Cramér-Rao Bound with B-Spline Scaling Functions

This section presents the evaluation of the CRBs in (5.19). We use a family of B-spline scaling functions to generate the coefficients used in the estimation of the location  $t_0$  and amplitude  $A$  of a step function  $s(t)$  where

$$s(t) = \begin{cases} 0 & t < t_0, \\ A & t \geq t_0. \end{cases}$$

More specifically, we show the calculation of the matrix  $J_{t_0,A}$  in (5.18). The results are plotted in Figure 5.2.

Recall that a B-spline function of order  $P \geq 0$  is given by

$$\beta(t) = \frac{1}{P!} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l (t-l)_+^P \quad \text{with} \quad (t)_+^P = \begin{cases} 0 & t < 0, \\ t^P & t \geq 0. \end{cases}$$

The coefficients are then given by

$$y_n = \langle s(t), \beta_{J,n}(t) \rangle, \quad J < 0, \tag{A.2.1}$$

where  $\beta_{J,n}(t) = 2^{-J/2} \beta(2^{-J}t - n)$ . Let us denote the scaling factor with  $T = 2^J$ .

Evaluating (A.2.1) gives

$$\begin{aligned} y_n &= \frac{1}{P! \sqrt{T}} \int_{nT}^{(n+P+1)T} s(t) \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l \left(\frac{t}{T} - n - l\right)_+^P dt \\ &= \frac{1}{P! \sqrt{T}} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l \int_{nT}^{(n+P+1)T} s(t) \left(\frac{t}{T} - n - l\right)_+^P dt \\ &= \begin{cases} 0 & n < \lfloor \frac{t_0}{T} \rfloor - P, \\ \frac{A}{P! \sqrt{T}} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l \int_{t_0}^{(n+P+1)T} \left(\frac{t}{T} - n - l\right)_+^P dt & \lfloor \frac{t_0}{T} \rfloor - P \leq n \leq \lfloor \frac{t_0}{T} \rfloor, \\ \frac{A}{P! \sqrt{T}} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^l \int_{nT}^{(n+P+1)T} \left(\frac{t}{T} - n - l\right)_+^P dt & n > \lfloor \frac{t_0}{T} \rfloor. \end{cases} \end{aligned}$$

It then follows that the partial derivative of  $y_n$  with respect to  $t_0$  is

$$\frac{\partial y_n}{\partial t_0} = \begin{cases} \frac{A}{P!\sqrt{T}} \sum_{l=0}^{P+1} \binom{P+1}{l} (-1)^{l+1} \left(\frac{t_0}{T} - n - l\right)_+^P & \lfloor \frac{t_0}{T} \rfloor - P \leq n \leq \lfloor \frac{t_0}{T} \rfloor, \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.2.2})$$

and, similarly, the partial derivative with respect to  $A$  is

$$\frac{\partial y_n}{\partial A} = \begin{cases} \frac{\sqrt{T}}{P!} \sum_{l=0}^{P+1} \binom{P}{l} (-1)^l \left( (P+1-l)^{(P+1)} - \left(\frac{t_0}{T} - n - l\right)_+^{P+1} \right) & \lfloor \frac{t_0}{T} \rfloor - P \leq n \leq \lfloor \frac{t_0}{T} \rfloor, \\ \frac{\sqrt{T}}{P!} \sum_{l=0}^{P+1} \binom{P}{l} (-1)^l (P+1-l)^{(P+1)} & n > \lfloor \frac{t_0}{T} \rfloor, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A.2.3})$$

By substituting (A.2.2) and (A.2.3) into (5.19), we obtain the CRBs.



# Bibliography

- [1] A. Aaron and B. Girod, “Compression with side information using turbo codes,” in *Proceedings of Data Compression Conference*, 2002, pp. 252–261.
- [2] A. Aaron, S. Rane, E. Setton, and B. Girod, “Transform-domain Wyner-Ziv codec for video,” in *Proc. Visual Communications and Image Processing*, San Jose, CA, January 2004.
- [3] A. Aaron, S. Rane, R. Zhang, and B. Girod, “Wyner-Ziv coding for video: applications to compression and error resilience,” in *Proceedings of Data Compression Conference*, 2003, pp. 93–102.
- [4] N. Ahmed, T. Natarajan, and K. Rao, “Discrete cosine transform,” *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.
- [5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.
- [6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and O. M., “The DISCOVER codec: Architecture, techniques and evaluation,” in *Picture Coding Symposium (PCS)*, Lisboa, Portugal, November 2007.
- [7] L. Baboulaz and P. Dragotti, “Exact feature extraction using finite rate of innovation principles with an application to image super-resolution,” *IEEE Transactions on Image Processing*, vol. 18, no. 2, pp. 281–298, February 2009.

- [8] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, “Wavelet transforms that map integers to integers,” *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332–369, 1998.
- [9] N.-M. Cheung, C. Tang, A. Ortega, and C. S. Raghavendra, “Efficient wavelet-based predictive Slepian-Wolf coding for hyperspectral imagery,” *EURASIP Journal on Signal Processing*, vol. 86, no. 11, pp. 3180–3195, 2006.
- [10] S.-J. Choi and J. Woods, “Motion-compensated 3-d subband coding of video,” *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 155–167, February 1999.
- [11] A. Cohen, I. Daubechies, O. Guleryuz, and M. Orchard, “On the importance of combining wavelet-based nonlinear approximation with coding strategies,” *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1895–1921, July 2002.
- [12] L. Coulot, M. Vetterli, T. Blu, and P. L. Dragotti, “Sampling signals with finite rate of innovation in presence of noise,” *École Polytechnique Fédérale de Lausanne (EPFL) Technical Report*, 2007.
- [13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.
- [14] P. L. Dragotti, M. Vetterli, and T. Blu, “Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix,” *IEEE Transactions on Signal Processing*, vol. 55(5), pp. 1741–1757, May 2007.
- [15] P. Dragotti and M. Gastpar, *Distributed Source Coding: Theory, Algorithms and Applications*. Academic Press, January 2009.
- [16] P. Dragotti and M. Vetterli, “Wavelet footprints: theory, algorithms, and applications,” *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1306–1323, May 2003.
- [17] N. Farvardin and J. Modestino, “Optimum quantizer performance for a class of non-gaussian memoryless sources,” *IEEE Transactions on Information Theory*, vol. 30, no. 3, pp. 485 – 497, May 1984.

- [18] M. Flierl and P. Vandergheynst, “Distributed coding of highly correlated image sequences with motion-compensated temporal wavelets,” *Eurasip Journal Applied Signal Processing*, vol. 2006, September 2006.
- [19] J. Garcia-Frias and F. Cabarcas, “Approaching the slepian-wolf boundary using practical channel codes,” *ACM Signal Process.*, vol. 86, no. 11, pp. 3096–3101, 2006.
- [20] M. Gastpar, P. Dragotti, and M. Vetterli, “The distributed KarhunenLove transform,” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5177–5196, December 2006.
- [21] N. Gehrig and P. Dragotti, “Symmetric and asymmetric Slepian-Wolf codes with systematic and nonsystematic linear codes,” *IEEE Communications Letters*, vol. 9, no. 1, pp. 61–63, 2005.
- [22] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.
- [23] B. Girod, “Motion-compensating prediction with fractional-pel accuracy,” *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604–612, April 1993.
- [24] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.
- [25] H. Gish and J. Pierce, “Asymptotically efficient quantizing,” *IEEE Transactions on Information Theory*, vol. 14, no. 5, pp. 676–683, September 1968.
- [26] V. K. Goyal, “Theoretical foundations of transform coding,” *IEEE Signal Processing Magazine*, vol. 18(5), pp. 9–21, September 2001.
- [27] R. Gray and D. Neuhoff, “Quantization,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.
- [28] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, “Distributed monoview and multiview video coding,” *Signal Processing Magazine, IEEE*, vol. 24, no. 5, pp. 67–76, 2007.

- [29] J. Heikkilä, “Pattern matching with affine moment descriptors,” *Pattern Recognition*, vol. 37, no. 9, pp. 1825–1834, 2004.
- [30] J. Huang and P. Schultheiss, “Block quantization of correlated gaussian random variables,” *IEEE Transactions on Communications Systems*, vol. 11, no. 3, pp. 289–296, September 1963.
- [31] D. Huffman, “A method for the construction of minimum-redundancy codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, September 1952.
- [32] B.-J. Kim, Z. Xiong, and W. Pearlman, “Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT),” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1374–1387, December 2000.
- [33] D. Le Gall and A. Tabatabai, “Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques,” in *International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 761–764 vol.2.
- [34] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, December 2008.
- [35] S. Mallat and F. Falzon, “Analysis of low bit rate image transform coding,” *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1027–1042, April 1998.
- [36] D. Marpe, T. Wiegand, and G. Sullivan, “The H.264/MPEG4 advanced video coding standard and its applications,” *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, August 2006.
- [37] J.-R. Ohm, “Advances in scalable video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, January 2005.
- [38] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

- [39] S. Pradhan and K. Ramchandran, “Enhancing analog image transmission systems using digital side information: a new wavelet-based image coding paradigm,” in *Data Compression Conference, 2001. Proceedings. DCC 2001.*, 2001, pp. 63–72.
- [40] —, “Distributed source coding using syndromes (DISCUS): design and construction,” *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.
- [41] R. Puri and K. Ramchandran, “PRISM: a video coding architecture based on distributed compression principles,” in *40th Allerton Conference on Communication, Control and Computing*, Allerton, IL, October 2002.
- [42] D. Rebollo-Monedero, S. Rane, A. Aaron, and B. Girod, “High-rate quantization and transform coding with side information at the decoder,” *EURASIP Journal on Signal Processing*, vol. 86 (11), pp. 3160–3179, November 2006.
- [43] A. Said and W. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.
- [44] D. Schonberg, S. S. Pradhan, and K. Ramchandran, “LDPC codes can approach the slepian-wolf bound for general binary sources,” in *40th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.
- [45] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.
- [46] A. Secker and D. Taubman, “Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression,” *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, 2003.
- [47] C. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, January 1949.
- [48] J. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

- [49] P. Shukla and P. Dragotti, “Sampling schemes for multidimensional signals with finite rate of innovation,” *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3670–3686, July 2007.
- [50] T. Sikora, “MPEG digital video-coding standards,” *IEEE Signal Processing Magazine*, vol. 14, no. 5, pp. 82–100, 1997.
- [51] —, “Trends and perspectives in image and video coding,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 6–17, 2005.
- [52] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [53] V. Stankovic, A. Liveris, Z. Xiong, and C. Georgiades, “On code design for the slepian-wolf problem and lossless multiterminal networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1495–1507, April 2006.
- [54] J.-L. Starck, E. Candes, and D. Donoho, “The curvelet transform for image denoising,” *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, June 2002.
- [55] G. Strang and G. Fix, “A Fourier analysis of the finite element variational method,” in *Constructive Aspect of Functional Analysis*, Rome, Italy, 1971, pp. 796–830.
- [56] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [57] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.
- [58] —, “Video compression - from concepts to the H.264/AVC standard,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, 2005.
- [59] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

- [60] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.
- [61] D. Taubman and M. Marcellin., *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2001.
- [62] M. Unser, “Ten good reasons for using spline wavelets,” in *Proceedings of the SPIE Conference on Mathematical Imaging: Wavelet Applications in Signal and Image Processing V*, vol. 3169, San Diego CA, USA, July 30-August 1, 1997, pp. 422–431.
- [63] —, “Sampling-50 years after Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.
- [64] M. Unser and T. Blu, “Mathematical properties of the JPEG2000 wavelet filters,” *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1080–1090, 2003.
- [65] —, “Wavelet theory demystified,” *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 470–483, 2003.
- [66] M. Vetterli, “Wavelets, approximation, and compression,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 59–73, 2001.
- [67] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice Hall PTR, April 1995.
- [68] M. Vetterli, P. Marziliano, and T. Blu, “Sampling signals with finite rate of innovation,” *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.
- [69] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [70] M. Wang and M. van der Schaar, “Operational rate-distortion modeling for wavelet video coders,” *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3505–3517, 2006.

- [71] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- [72] R. Xiong, J. Xu, F. Wu, and S. Li, “Barbell-lifting based 3-d wavelet coding scheme,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1256 –1269, sept. 2007.