

Algorithmic Representation of Visual Information

Daby M. Sow

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

Columbia University

2000

© 2000

Daby M. Sow

All Rights Reserved

ABSTRACT

Algorithmic Representation of Visual Information

Daby M. Sow

This thesis presents new perspectives to media representation and addresses fundamental source coding problems outside the umbrella of traditional information theory, namely, the representation of finite individual objects with a finite amount of computational resources. We start by proposing a new theory, Complexity Distortion Theory, which uses programmatic descriptions to provide a mathematical framework where these problems can be addressed. The key component of this theory is the substitution of the decoder in Shannon's communication system by a computer. The mathematical framework for examining issues of efficiency is then Kolmogorov Complexity Theory. Complexity Distortion Theory extends this framework to include distortion by defining the complexity distortion function, the equivalent to the rate distortion function in this algorithmic setting. We show that this information measure predicts asymptotically the same results as the classical probabilistic information measures, for stationary and ergodic sources. These equivalences highlight the duality between Shannon and Kolmogorov's information measures. The former defines information as a set notion that requires the estimation of relative frequencies to predict asymptotic results whereas the latter is a deterministic concept defining randomness for individual objects. It allows us to formalize the universal coding problem for finite individual objects. This then closes the circle of media representation techniques, from probabilistic to deterministic approaches. It also opens new horizons outside the scope of classical source coding that we explore in the second part of this thesis. In contrast with the classical approach, computa-

tional resource bounds can be introduced naturally at the decoding end. This way, we add a new dimension to source coding theory, extending the rate distortion curve to a complexity distortion surface representing the tradeoff between rate, distortion and computational complexity. Understanding this complex tradeoff is key for the design of efficient decoders with efficient computational resource management capabilities. In the last part of this thesis, we approximate this surface in a constructive fashion yielding a new class of algorithms for the universal coding of finite objects under distortion and computational constraints. An extensive analysis of the convergence properties of these algorithms is presented together with their application to still image data.

Contents

1	Introduction	1
1.1.	Introduction	1
1.2.	Thesis Contributions	9
1.2.1	Complexity Distortion Theory	9
1.2.2	Resource Bounds in Media Representation	10
1.2.3	Universal Coding of Finite Objects	12
1.3.	Outline of Thesis	13
2	Classical Information and Rate Distortion Theories	15
2.1.	Introduction	15
2.2.	Information Sources and Notations	19
2.3.	Information Measures	24
2.3.1	Lossless Measures	24
2.3.2	Lossy Measures	27
2.4.	Source Coding Theorem and Universal Coding	32
2.5.	Conclusion	38
3	Complexity Distortion Theory	40
3.1.	Introduction	40
3.2.	Universal Turing Machines	43

3.3.	Kolmogorov Complexity	46
3.3.1	Individual Information Measure	48
3.3.2	Randomness Tests	53
3.4.	Equivalence with Information Theory	55
3.4.1	Fundamental Theorem	55
3.4.2	Proof of Fundamental Theorem	58
3.5.	Complexity Distortion Function	61
3.6.	Equivalence with Rate Distortion Theory	62
3.6.1	Extended Fundamental Theorem	62
3.6.2	Proof of Extended Fundamental Theorem	64
3.6.3	Some Remarks	72
3.7.	Conclusion	73
4	Resource Bounds in Media Representation	75
4.1.	Introduction	75
4.2.	Resource Bounded Complexity Distortion Function	78
4.3.	Universal Coding of Finite Objects with Distortion and Computa- tional Constraints	88
4.3.1	Universal Coding Revisited	90
4.3.2	The Decoder	93
4.3.3	The Encoder	94
4.4.	Convergence Analysis of Genetic Programming	97
4.4.1	Convergence	101
4.4.2	Speed of Convergence	103
4.5.	Algorithmic Representation of Images	109
4.5.1	The Decoder	109
4.5.2	The Encoder	111

4.6. Conclusion	116
5 Conclusion and future directions	119
5.1. Conclusion	119
5.2. Future directions	121
5.2.1 Channel Capacity versus System Capacity	121
5.2.2 Language Design	123
References	125
Appendix	133
A. Recursive Functions	133
B. Randomness Tests	134
C. Markov Types	137

List of Figures

2-1	Shannon's communication system. The signals transmitted on the channels represent codewords. They are typically indices to reproduction sequences on which redundancy bits have been added for error correction.	16
2-2	The separation between source and channel coding at the transmitter. In this thesis, we focus on the source coding operations.	17
2-3	The shift operator on <i>two-way</i> infinite sequences	21
2-4	A stationary information source. Initially, the initial mode choice module selects one of the stationary sources and the source stays in this mode for the rest of its operation. Hence from any long observation of the source, we can estimate the distribution of one particular stationary source. This source would be ergodic if the initial mode choice almost always select the same stationary source $i, 1 \leq i \leq m$	22
2-5	The sphere covering problem in rate distortion theory. The small spheres represent finite sequences. The probability measure associated with these sequences would be represented by the area of these small spheres. The objective is to minimize the number of big spheres needed to cover a subspace of sequences of length n with a high measure, greater than $1 - \epsilon, 0 < \epsilon < 1$	30
2-6	Universal Coding System type I.	35

2-7	Universal Coding System type II.	36
2-8	Universal Coding System type III.	36
2-9	Universal coding and the separation principle between model and data.	39
3-1	Finite Automaton: The top figure shows the transition diagram of a finite automaton. Assuming that the initial and final state are both q_0 , this automaton will recognize all sequences in which both the number of 0's and the number of 1's is even. The example shown in the bottom will leave the automaton in state q_3 at the end of the computation.	44
3-2	A Turing machine with one work or memory tape.	45
3-3	The programmable communication system. Notice that two-part codes representing algorithms and data are sent through the channel. We call this communication system Kolmogorov's communication system.	47
3-4	Duality between Shannon's entropy and Kolmogorov complexity.	54
3-5	The circle of media representation theories (the bottom-right box as well as its associated arrows, establishing its relationship with other theories, are introduced in this thesis).	64
4-1	Rate-Distortion-Complexity tradeoff.	81
4-2	Algorithmically universal coding system. This system generalizes the first three universal coding systems presented in chapter 2, section 2.4.	92
4-3	The transition matrix for the elitist strategy.	106
4-4	Hybrid Image Encoder.	112
4-5	This figure shows how the memory indexing is done, relatively to the position of the current block.	113

4-6	Programmatic representation of lenna 512x512, psnr = 29.72 dB at 1.17 bpp: (a) original, (b) gp output. In this run the language uses 4x4 blocks. Note the large errors introduced in the top of the picture because of the inability of the language to represent these blocks when the memory of the system is empty.	114
4-7	This figure compares 8x8 blocks with a significant amount of edges (according to a Sobel edge detector) with blocks that cannot be represented accurately by the GP module. Edge blocks and GP blocks with large MSE are in white: (a) edge blocks, (b) GP blocks with large error. Note the similarities between this two pictures showing that the language did not manage to represent accurately most of the edge blocks.	115
4-8	Programmatic representation of lenna 512x512 with the learning unit, psnr = 32.97 dB at 0.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 1381 DCT blocks (out of 4096), mainly at the strong edges of the image. .	116
4-9	Programmatic representation of house 256x256 with the learning unit, psnr = 35.75 dB at 0.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 425 DCT blocks (out of 1024), mainly at the strong edges of the image	117
4-10	Programmatic representation of peppers 256x256 with the learning unit, psnr = 34.02 dB at 1.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 643 DCT blocks (out of 1024), mainly at the strong edges of the image . .	117
5-1	Channel and system capacities.	122
5-2	General system channel.	123

List of Tables

4.1	Functions used to represent gray level image data.	110
4.2	Terminals used to represent gray level image data.	111

List of Abbreviations

AEP	Asymptotic Equipartition Property
CDF	Complexity Distortion Function
CDT	Complexity Distortion Theory
DCT	Discrete Cosine Transform
FA	Finite Automata
FSM	Finite State Machine
GA	Genetic Algorithm
GP	Genetic Programming
HVS	Human Visual System
IT	Information Theory
JPEG	Joint Photography Experts Group
KLT	Karhunen Loeve Transform
MAP	Maximum a posteriori
MDL	Minimum Description Length
ML	Maximum Likelihood
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
RDF	Rate Distortion Function
RDT	Rate Distortion Theory
SAOL	Structured Audio Language
TM	Turing Machine
UTM	Universal Turing Machine
VQ	Vector Quantization / Vector Quantizer

Acknowledgements

This thesis could not have been written without the help and support of numerous people that I would like to thank. I will start by expressing my gratitude to my advisor, Professor Alexandros Eleftheriadis not only for all his guidance and his encouragement to pursue my own research interests, but also for putting me in the right working environment. I am also very grateful to Professor Dimitris Anastassiou for all his help and support throughout these years. Special thanks go to Professors Shih Fu Chang and Predrag Jelenkovic for all their advises that made me a better researcher.

The quality of this work has been significantly improved by the diligent efforts of the members of my defense committee, Professor Dimitris Anastassiou, Professor Marianthi Markatou, Professor Predrag Jelenkovic, Dr Mahmoud Naghshineh whom I all warmly thank.

During the time spent at Columbia, I had the privilege to interact with a number of colleagues and friends. It would be difficult to list everybody here. I would like to give special mentions to all my colleagues in the ADVENT lab, all my officemates (past and present). I also wish to thank Professor Lazlo Toth for all his wise advises and help. I should not forget the people at Philips Research Briarcliff for giving this wonderful opportunity to work with them during the summer 1999.

Finally, I would like to thank my close family, for “everything”, Maman, Papa, Fatim, Mahmoud, Zeinab, Zhara, Oumou, Malick, Hazna, Oumar, my aunt Khadiatou and my uncle Alioune. Very special thanks go to my brother Mouhamadou for all the hard work. Also, on behalf of the entire Sow family, I would like to thank Professor Ibrahima Thioune and Professor Abdoul Salam Dia for all their help.

Last but not least, I would like to thank my wife Balkissa for all her love, support and patience.

Chapter 1

Introduction

1.1. Introduction

Current methodologies for audio-visual information representation have their roots in systems conceived and designed several decades ago. They evolved out of the desire to design optimal representations in a compression sense: minimize the average bitrate required to represent a particular source. Targeted applications involved vertical designs such as telegraphy, telephony, facsimile, videoconferencing, or even digital television. In all these cases, the desired objective was the minimization of operating costs by minimizing the required transmission bandwidth under a reproduction quality constraint or, equivalently, maximizing the quality given a bandwidth constraint.

The theoretical foundation for addressing this problem was established in 1948, when C. E. Shannon introduced Information and Rate Distortion Theories. He discovered the limits of data compression by modeling information sources with stochastic processes. According to this model, information is a measure of uncertainty called entropy. The whole theory is based on probability theory ignoring the meaning of the message which is considered “irrelevant” [70]. Pragmatic considerations (tractability) made necessary to add stationarity and ergodic assumptions on

the source. One of Shannon's key discoveries was that, for this class of stochastic sources, the negative logarithm of the probability of a typical long sequence divided by the number of symbols is a very good indicator of the amount of non redundant information conveyed in this sequence. In a lossless setting, Shannon proved in [70] that the best achievable average performance in a compression sense is close to the average negative logarithm of the probability, which is commonly called the entropy. He extended these results to the lossy case with the concept of rate distortion function taking the role of the entropy. A lot of attempts have been made since then to actually design algorithms approaching these theoretical limits and classical source coding theory attempts quite successfully to pave the way for the design of such efficient representation algorithms. In practice, as pointed by Wyner, Ziv and Wyner in [95], in a broad sense, three possible situations are commonly considered:

1. The source distribution is completely known.
2. The source distribution is completely unknown, but it belongs to a parameterized family of probability distributions.
3. The source distribution is known to be stationary and ergodic, but no other information is available.

A wide variety of efficient and practical algorithm are known for case 1, i.e. Shannon-Fano-Ellias Coding [15], Huffman Coding [41], Arithmetic Coding [65] with their extensions to lossy cases. But in practice, this situation is rare. Most of the time, the underlying probability law hidden inside the source machinery is not completely known. Hence, it becomes imperative to estimate and describe efficiently this law, prior to coding. We then fall into case 2 and 3. In case 2, modeling the source involves the estimation and description of the parameter that would index a particular distribution from the family. In case 3, a complete source distribution estimation

procedure must be performed. Clearly, these approaches to source coding raise some fundamental questions. One of them is stated in [95]: Can we find an appropriate and universal way to estimate the probability law that governs the generation of messages by the source ? Information Theory gives a lot of insight to this problem by estimating probabilities with relative frequencies from long observations of the source, under stationary and ergodic assumptions. In practice, such observations may not be available, as the length of the object to code is always inherently finite and not always large. That brings us to another question inquiring on the validity of probabilistic models for finite objects. How can we estimate a probability distribution from a finite object? This problem gives birth to a fourth leaf to the taxonomy of practical situations corresponding to the case where the object to encoded is finite and cannot be modeled with probabilities. In this case, we lose a significant amount of mathematical tractability since ergodic theory cannot be applied anymore and there is a need to address this question from a different perspective.

Although source coding has traveled a long distance since 1948, the fundamental principles are still the same. Applications of it to image/video data blossomed with the emergence of transform coding linking this field with Harmonic Analysis. The lossy coding problem as introduced by Shannon in [70] addressed extensively the coding of continuous-valued stochastic processes. Gaussian processes with the mean square error as a distortion measure were almost exclusively investigated because of their wide practical interest at that time and also because of their tractability. In fact, rate distortion tradeoffs for most other classes of continuous stochastic processes do not have known closed form solutions even today, after more than 50 years of intensive research. Information theoretic results on independently and identically distributed Gaussian sources had a tremendous impact in image compression via Harmonic Analysis and Transform Coding. Such links between harmonic analy-

sis and Shannon's Rate Distortion Theory, are presented in [19]. In general, we try to approach the performances of optimal transforms like the Karhunen-Loeve transform (KLT) which relies on statistical analysis, using less complex transforms like the discrete cosine transform (DCT) or the wavelet transform. Adopting a "maximalist" [19] position, it can be argued that there is a deep reason for the interaction of these fields of harmonic analysis and transform coding. In fact, sinusoids and wavelets have been used extensively in image/video processing and today, it is quite accurate to say that state of the art visual representation systems (MPEG-1, MPEG-2, MPEG-4, JPEG, JPEG-2000, H.263) all use these fundamental mathematical signals to represent visual information. These mathematical entities do have a special role in the field simply because of their special "optimal" role in the representation of certain stochastic processes. That brings us back to the question raised earlier in this chapter, whether stochastic processes model well finite natural and synthetic visual signals.

Today's picture of the communication world is, however, much different from what it was when the fundamental concepts behind Information Theory were introduced in the 50's and these differences might alter the way we approach the source coding problem in the future. Digital audio-visual information is no longer following the simple cycle of production, transmission, reception, and playback. The ever increasing power of modern computers has transformed them into very capable platforms for audio-visual content creation and manipulation. Users today can very easily capture compressed audio, images, or video, using a wide array of consumer electronics products (e.g., digital image and video cameras as well as PC boards that produce JPEG and MPEG-1 content directly). It is quickly realized, though, that the objective of compression may conflict with other applications requirements (ease of editing, processing, indexing and searching, etc.). Compression

is then just one of many desirable representation characteristics. Features such as object-based design, seamless access to content, editing in the compressed domain, scalability and graceful degradation for network transmission, graceful degradation with diminishing decoder capabilities, flexibility in algorithm selection, and even downloadability of new algorithms, are quickly becoming fundamental requirements for new audio-visual information representation approaches. In brief, there is an increasing need to add a significant amount of flexibility and universality. Also, it is important to realize that the back end of any audio/visual communication system is not a computer terminal or a television; it is the Human Visual System (HVS). This observation emphasizes the need for the development of communication systems able to *understand* the information content in individual messages. It is not natural to add all these new components in Shannon's framework where semantics are completely ignored.

At the same time, novel coding techniques which do not fit well in the traditional theoretical framework, have appeared. Fractals and model-based coding are characteristic examples. In the first case, the content is represented by an iterative transformation; what is transmitted is the parameters of the transform. In the latter case, the content is *synthesized* at the receiver based on a given two or three-dimensional model; what is transmitted is parameters that specify the spatio-temporal evolution of the model. We see that the areas of natural and synthetic (computer-generated) content representation are rapidly merging, due to the fact that both are now coexisting in computers. Synthetic content representation has a very rich history, with computer graphics, audio synthesis (MIDI etc.), image synthesis (ray tracing etc.). MPEG-4 [3, 64] is the first audio-visual representation standard that combines natural and synthetic content using an object-based approach.

Beyond pure representation, with the development of languages like Java, execution of platform-independent downloadable code is now commonplace on every user's desktop. Extensions of traditional programming languages towards media representation, like Flavor [24, 23] (an extension of C++/Java that incorporates bitstream representation semantics), help to bridge the gap between source coding and software application development.

All these observations show that there are a lot of reasons to believe that instead of adopting exclusively the “maximalist” position, it is quite reasonable to also make some room for the “minimalist” approach. We will not go as far as saying that Harmonic analysis has exerted an influence on data compression merely by happenstance following a real minimalist position [19]. We do not completely believe that there is no fundamental connection between, say, wavelets and sinusoids, and the structure of digitally acquired data to be compressed. In any case, it is safe to say that these techniques have received a lot of attention mainly because of their mathematical tractability and also because of the existence of fast algorithms to perform the transforms at both ends of the communication system. They have been studied extensively and applied to real data with very good performance levels but we believe that they are still far from the limits imposed by the underlying structure of typical sources of visual information, at least as it is perceived by the HVS.

On a slightly different note, note that classical source coding algorithms are optimized in two dimensions, information rate and distortion. With the proliferation of hardware programmable decoders, new dimensions of significant importance come into the picture. They are related with time and space complexity of the decoding operations. Classical source coding does not consider these problems and there is a need for a mathematical framework that would take these issues into account in order to reduce the operational cost of these complex decoding devices. Such a

novel approach to source coding would address important questions like decoding computational resource management in multithreaded decoding environment and extend the computational resource allocation problem that programmable media processors must face.

The traditional information-theoretic framework cannot address questions of optimality or even efficiency in such application environments. A fundamental change is then required in the classical communication system model proposed in [70]. Its most important shortcoming is that the structure of the receiving system is ignored. This makes it extremely hard to introduce any additional representation requirements beyond minimization of the bitrate. By forcing us to consider only the structure of the source, the only avenue available for analysis is proper probabilistic modeling and minimization of the expected bits per symbol. Note that, fundamentally, the interpretation of these symbols has not changed since the introduction of Information Theory: they are the result of a ranking procedure where the source events are ordered according to their frequency of occurrence.

To reformulate the representation problem on a more flexible basis, two fundamental changes are required. First, we need to introduce structure into the decoding system by considering it to be a programmable device, i.e., a computer modeled by a universal Turing machine (UTM). This way, current practice can be directly reflected to our mathematical model. More importantly, real implementation constraints (such as limitations in space – memory – and time) can be naturally incorporated. Second, we allow the possibility that the content itself is represented by an algorithm. In other words, instead of transmitting abstract data that, when processed by an algorithm, will generate the desired content, it is the algorithm itself that is transmitted from the source and, when executed by the receiving system, it reproduces the desired content.

In 1965, in an attempt to measure the amount of randomness in individual objects, A.N. Kolmogorov introduced another information measure based on length of descriptions [44]. Similar concepts were also presented at the same time by G. Chaitin [11] and R.J. Solomonoff [76]. In this case, entropy is a lack of compressibility. It is measured individually by the length of the shortest computer program able to generate the object to represent. Kolmogorov modified Shannon’s communication system model and replaced the decoder with a universal Turing machine. The codewords become programs written in the language of the decoder. In such a framework, the algorithm becomes itself the content rather than just a method used to process the content. The efficient representation problem becomes a modeling question where from an observation of the source, we are looking for the machinery hidden in the source. Following the Occam Razzor principle, the best source model is the simplest one. According to the work of R.J. Solomonoff [76] on inductive inference, the term “simple” can be replaced by shortest and in order to describe any object, we are looking for the shortest model-data pair, which corresponds to Kolmogorov and Chaitin’s approach. In this case, to analyze the performances of this programmable system, information must be measured using the Algorithmic or Kolmogorov¹ complexity [15, 52], a measure of length of shortest descriptions for arbitrary objects.

Since its introduction, Kolmogorov Complexity Theory has grown substantially. It has concentrated, however, on data compression, i.e., lossless representation. It has been used extensively in inference problems and in computational complexity to derive general upper bounds. In this thesis, we address information representation issues in this setting and extend the notion of complexity to include distortion. This allows the application of the complexity framework to audio-visual information

¹In this thesis, we use both terms Algorithmic and Kolmogorov Complexity.

representation, where the introduction of (ideally non-perceptible) distortion is a key mechanism for allowing non-trivial compression. The developed mathematical framework also allows us to take into account computational resource bound issues at the decoding end by taking into account the structure of the decoder inside the mathematical framework.

1.2. Thesis Contributions

The contribution of this work is mainly theoretical and can be subdivided into three themes. The first one, Complexity Distortion Theory, provides the mathematical foundations on which the other two are developed. The second theme focuses at the decoding end of the communication systems and addresses resource bound issues in media representation. The last theme switches the attention to the encoding end and discusses novel approaches to universal lossy coding in such a programmatic setting.

1.2.1 Complexity Distortion Theory

Complexity Distortion Theory is a logical extension of the Algorithmic Complexity Theory to allow lossy representations. A key question that we address in Complexity Distortion Theory is its relationship with Rate Distortion Theory. In order for the new framework to be truly unifying, it must predict identical bounds with traditional Rate Distortion Theory. It has long been known [50, 105] that Kolmogorov complexity predicts the same asymptotic bounds as entropy, for stationary ergodic sources. We prove that in a similar way, Complexity Distortion Theory predicts identical bounds with Rate Distortion Theory for stationary ergodic sources. At the heart of this equivalence is the concept of randomness developed by Kolmogorov and Chaitin and the existence of randomness tests as defined by Martin

Löff. This equivalence is central to all the main theoretical contribution of this thesis. It clearly identifies the set of sequences on which Kolmogorov's and Shannon's approaches diverge. It bridges Shannon's probabilistic world to Kolmogorov's deterministic world. From this connection interesting problems can be mapped in each world. For instance, the decoding resource management problem can be formulated naturally in Kolmogorov's setting but many properties of it are better addressed in Shannon's framework where they are much more tractable mathematically. The situation is altered when we turn our focus on the coding of finite objects. This problem is better addressed in the algorithmic framework, without any concept of probabilities. In brief, this equivalence closes the circle of traditional, probabilistic measures of information represented by Information and Rate Distortion theories, and deterministic, algorithmic measures represented by Algorithmic Complexity and Complexity Distortion theories. It also expands the duality between these two approaches to measure information. Shannon's approach grew from a desire to measure information and is based on randomness whereas Kolmogorov's approach grew from a desire to measure randomness in individual objects and is based on an individual measure of information.

1.2.2 Resource Bounds in Media Representation

Computational resource issues in source coding have received a significant amount of attention almost exclusively at the encoding end of the communication system. In this case, the goal is to find cheap encoding solutions yielding performances close to the rate distortion curve, defined in the next chapter. For instance, in vector quantization, the codebook design has pretty high computational requirements. At the other side, the decoding operations are less intensive (look-up tables). This is also true for typical transform coders where the encoders have to face complex op-

timization problems. This computational disparity explains why complexity issues have been mostly investigated at the encoding end. Things become much different when universality is added at the decoding end and the current trend in digital audio/video processing is to use general purpose processors at the decoding end of communication systems. It brings up several new exciting issues dealing with the management of computational resources at the decoding end of the system. The injection of flexibility with the use of programmable decoders degrades the overall performances of the system since dedicated (and optimized) hardware solutions are ruled out and replaced by software solutions with a general software overhead that affects the computational load of the system. It becomes imperative to improve the computational resource management process at the decoding end and understand the effects of reducing the allocated computational power of the decoding device for source decoding tasks. It becomes imperative to understand tradeoffs between information rate, distortion and computational complexity and this problem cannot be tackled naturally in Information Theory. Computational complexity issues cannot be addressed in this setting simply because the structure of the decoder is not part of the mathematical setting. Kolmogorov's setting does not suffer from this drawback. It allows the quantification of the effect of the decoding computational power on the coding efficiency. In this theme, we add a new dimension to the classical rate-distortion tradeoff which takes into account computational complexity. More specifically, we define a complexity distortion surface which extends the rate distortion curve into a multidimensional surface highlighting tradeoffs between information rate, distortion and computational complexity. We establish a key relationship between this surface and classical mutual information. This equivalence opens up the door to derive important properties of this complex tradeoff, like convexity. This provides all the necessary tools for the tuning of complex programmatic

systems for an efficient management of the resources and for a decrease of the overall representation costs.

1.2.3 Universal Coding of Finite Objects

The last theme of this thesis takes a major step towards practical considerations and addresses the universal coding problem for finite objects. Here, the goal is to find a systematic way to find short descriptions for finite objects, in a programmatic environment defined by the language of the universal decoding device, regardless of statistical considerations. We drift away from statistical modeling because of the inherent finiteness property of the object to code. This problem has a lot of similarities with Samuel’s problem introduced in 1959: How can computers program themselves ? [20]². It can be reduced to a generic machine learning issue where we seek short representations in a specified language, in order to better understand the information content. Indeed, recalling Occam’s Razor principle (stating that “if there are alternative explanations for a phenomenon, then, all things being equal, we should select the simplest one”³) together with R.J. Solomonoff’s Induction Theory [52], we realize that the universal coding problem that we are raising here is a fundamental machine learning problem and it is not surprising at all that we turn to this field to use state of the art program induction techniques to solve this problem. Our intent is to design a system capable of understanding the content of an object and expressing this knowledge in a given representation language. We believe that this is an important step to take in order to design systems able to link efficiently high level semantical concepts to low level data (at the bit level). Today, there is a lot of effort to close the gap between these two extremities of the representation

²See chap 15 by A.L. Samuel.

³This statement was made by William of Ockham 1290?-1349?.

spectrum, but we believe that the bridge between these two ends would rely on stronger foundations if programmatic representations of signals were used for low level data.

Following this approach, we have developed in software a typed genetic programming kernel, an extension of the genetic kernel described in [27], to perform program inductions for strongly typed languages and find short programmatic representations for finite objects with constraints on the decoding computational complexity and the distortion. The developed kernel allows the evolution of predefined strongly typed programming languages. The convergence properties of genetic programming techniques are not well understood in the machine learning community. There is a lack of mathematical results of this sort. In this thesis, we address certain aspect of this problem and propose formal arguments on these issues. We performed an extensive analysis of the performance of this approach, namely its convergence to optimality and its convergence speed. To illustrate the coding methodology, we then apply this technique to image data, using simple non Turing complete block-based languages.

1.3. Outline of Thesis

The thesis is organized as follows: In the next chapter, a formal introduction to the problems addressed is presented in an classical information theoretic setting. The necessary notations are introduced together with key classical source coding concepts and a discussion on universal source coding as it is conceived in the classical sense.

In chapter 3, we introduce Complexity Distortion Theory, a novel approach to media representation theory. We start with a definition of the main entities in this theory before starting a formal comparison of this algorithmic theory with rate distortion theory. This comparison revolves around two main theorems showing

equivalences between Kolmogorov and Complexity Distortion Theories with Information and Rate Distortion Theories. These equivalences are at the heart of all the significant theoretical and practical contribution of this thesis to the source coding field.

Finally, in chapter 4, we take a step towards practical considerations. We start with an extension of the rate distortion tradeoff to also include computational bounds at the decoding end. We define the complexity distortion surface and, inspired by the equivalences shown in chapter 3, we study the convexity of this surface. We then propose an algorithm for the approximation of this surface yielding a universal coding technique for finite objects. The proposed algorithm makes use of genetic programming as an optimization tool. We study its converging properties before applying it to still image data.

We end this thesis with concluding remarks in chapter 5. We also present a list of new horizons that we will explore in the future.

Chapter 2

Classical Information and Rate Distortion Theories

2.1. Introduction

In Shannon's information theory, the concept of information is presented in a communication setting. It is linked to unpredictability and signal variations. Indeed, the key observation made by Shannon in [70] is that if information is to be conveyed, it must be transmitted in signals changing unpredictably with time [69]. Accordingly, the concept of information is closely related to probabilities and in this framework, it is natural to link the amount of information sent via a signal in a fixed time interval to the rate at which this signal changes. Due to physical system limitations, this rate cannot grow to infinity. It is bounded by the bandwidth of the medium on which the transmission will occur. For instance, on local array networks millions of bits per second can be transmitted reliably. On telephone lines, the limit is lower in the order of thousands of bits per second. Upper bounds to this transmission rate for a particular medium define the channel or system capacity [70]. The fundamental problem of communication consists then of reproducing at one point either exactly or approximately a message selected at another point [70] after transmission of it

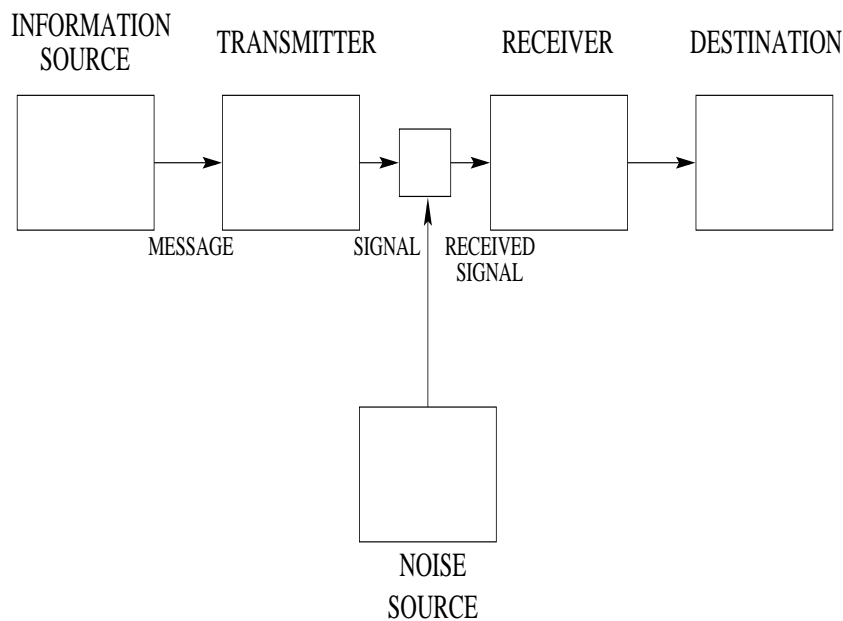


Figure 2-1: Shannon's communication system. The signals transmitted on the channels represent codewords. They are typically indices to reproduction sequences on which redundancy bits have been added for error correction.

through a medium with finite capacity.

In this setting, two fundamental problems dual with each other arise. The first one is the channel coding problem where the transmission channel is fixed and the question is to find the maximum amount of information that can be sent in this channel for all possible information sources. The second one is the main subject of this thesis. It is called the source coding problem. Here, the information source is fixed and the question is to find the smallest channel needed to transmit the source information under a distortion constraint. In this case, the optimal channel specifies ideal encoding and decoding procedures that should be performed by a codec (encoder/decoder) system. At the encoding end, a good encoder typically maps frequent source objects to very short descriptions and decodes them at the other end to get an accurate reproduction of the source signal. Although Channel and Source Coding problems were originally defined in such a communication setting,

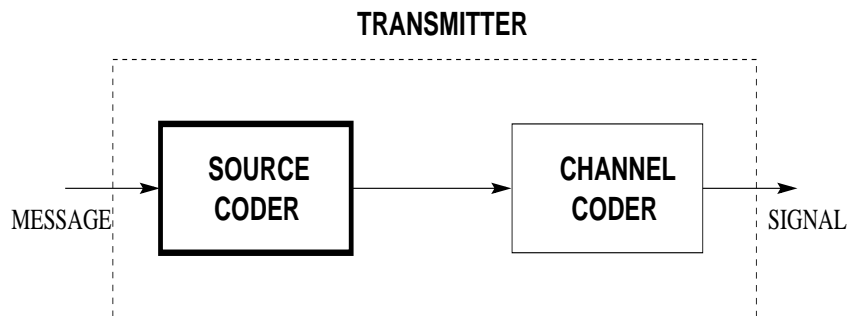


Figure 2-2: The separation between source and channel coding at the transmitter. In this thesis, we focus on the source coding operations.

they provide a general framework for the study of many real problems, ranging from the representation and transmission of signals to economics and finance including stock market predictions [15]. In all cases, the procedure relies heavily on statistical assumptions and assuming the complete knowledge of the universe of all messages at both the sending and receiving end, it is sufficient to only represent at the encoder, the selection of a particular message representing the source observation. The actual message content is *irrelevant*. In other words, the semantic aspects of communication are irrelevant in IT [70]. If the message universe is finite, then any monotonic function of its cardinal can be used to measure information. The concept of information is then an ensemble one measuring the number of possible choices to make for the selection of a message. It is commonly called the entropy and relies naturally on normalized set functions or more specifically probability measures. Accordingly, information sources are described *statistically* and modeled with *stochastic processes*. From these stochastic models, efficient coding procedures have been developed [41], [65].

This setting also highlights a very important principle that seems to be trivial but is fundamental in source coding: the use of *two-part* codes for representation [92]. Any source object can be fully represent using such two-part codes. The first part, called the model part, describes the source model. In IT, this is done

statistically by specifying the universe of all possible messages and a probability measure defined on it. The second part of the description, called the data part, describes the actual object using the assumed model represented in the first part. In IT, the data part is simply an index to a particular element of the universe of all messages. The effectiveness of a given coding technique on a source observation relies heavily on the model identification stage. Nevertheless, for simplicity reasons and mathematical tractability, it is generally assumed that source models do not change with time and that they can be estimated accurately from very long observations of the source. These two properties respectively called stationarity and ergodicity are used to justify the off-line model identification stage in most codec systems. In this case, since the model identification is static, its knowledge can be assumed at both end of the system and it does not have to be included in the representation. Such a system is not flexible but works well if the source statistics are stationary. For instance, in image compression, standards like JPEG, predefines tables for Huffman codes and quantization matrices commonly used for natural images¹. Unfortunately, it is well known that most sources of visual information are not stationary nor ergodic. Their statistics change with time. As a result, the model identification stage must become dynamic and has to be moved inside the representation process. This approach yields naturally to the concept of universal coding.

In this chapter, we review all these important information theoretic notions which are used throughout this thesis and set up the stage for the next chapters with formalization of these intuitive concepts. Although entitled “Classical Information and Rate Distortion Theories”, this chapter is not a summary of the important contribution of these fields. It is a theoretical introduction to the main problems

¹Typically, these standards allow the user to specify its own tables but there is a significant computational cost associated with these extra operations. In general, quantization tables are optimized for the incoming data but variable length coding (VLC) tables are not.

tackled in this thesis. For the representation of objects, the best place to start is inside Information and Rate Distortion Theories. Our aim is to clearly identify the scope of these theories and to what extent they can be applied in media representation theory. In Section 2.2. we begin with a presentation of the notations used throughout. Section 2.3. introduces important information theoretic concepts. It contains a discussion on information sources and provides a general setting from which we will build all the theoretical contribution of the thesis. We end this chapter with Section 2.4. with a presentation the main results of source coding theory for both the lossless and lossy settings and a discussion on Universal coding where we highlight the limitation of the information theoretic approach for the representation of finite objects.

2.2. Information Sources and Notations

As mentioned above, IT models information sources with stochastic processes. They are fully determined by an alphabet, a set of interesting and tractable events called the event space, and a probability measure defined on this event space. To be more formal, we start by introducing standard notations. Let the set of natural numbers and the set of non negative reals be respectively denoted by \mathcal{N} and \mathcal{R}^+ . The set of integers (positive and negative) is denoted by \mathcal{Z} . $|\cdot|$ denotes the absolute value when the argument is a number. It denotes the cardinal when the argument is a set. Let A_0 and \hat{A}_0 be two nonempty finite sets, respectively called the source and reproducing alphabet. We denote by \mathcal{A}_0 and $\hat{\mathcal{A}}_0$, the σ -algebras of subsets of A_0 and \hat{A}_0 . \mathcal{A}_0 and $\hat{\mathcal{A}}_0$ are event spaces for single random variables; they are closed under complementation and formation of countable unions. To extend these notions

to stochastic processes, let the measurable space (A, \mathcal{A}) be defined as:

$$(A, \mathcal{A}) = \prod_{k=-\infty}^{\infty} (A_k, \mathcal{A}_k), \quad (2.1)$$

where \prod denotes the cartesian product operator and (A_k, \mathcal{A}_k) are exemplars of the measurable space (A_0, \mathcal{A}_0) . The measurable space $(\hat{A}, \hat{\mathcal{A}})$ is defined similarly, as an infinite cartesian product of exemplars $(\hat{A}_k, \hat{\mathcal{A}}_k)$ of $(\hat{A}_0, \hat{\mathcal{A}}_0)$. An element x of A can be viewed as a two way infinite sequence of elements $x_k \in A_k$, $x = \dots, x_k, x_{k+1}, x_{k+2}, \dots$, $k \in \mathcal{N}$. By $\mu(\cdot)$ and $\nu(\cdot)$, we denote respectively probability measures on \mathcal{A} and $\hat{\mathcal{A}}$. The triplet (A, \mathcal{A}, μ) is called a time-discrete source or just a source². We also denote such a time-discrete source by $[A, \mu]$.

In practice, modeling information source with stochastic processes is made possible by two strong statistical assumptions: Stationarity and ergodicity. The former is a property shared by all information sources with statistics that do not change with time. The latter is a property shared by all information sources with statistics that can be evaluated from infinite observations of the source. To define these properties, let T be the shift transformation mapping elements of A to elements of A as follows:

$$\forall x \in A, (Tx)_k = x_{k+1} \quad (2.2)$$

For all event $E \in \mathcal{A}$, we define TE as follow:

$$TE = \{Tx : x \in E\} \quad (2.3)$$

The source is stationary if for all $E \in \mathcal{A}$, $\mu(E) = \mu(TE)$. A function g is time-invariant if $g(Tx) = g(x)$, for all $x \in A$. A set E is invariant if its indicator function

²We focus exclusively on time-discrete sources.

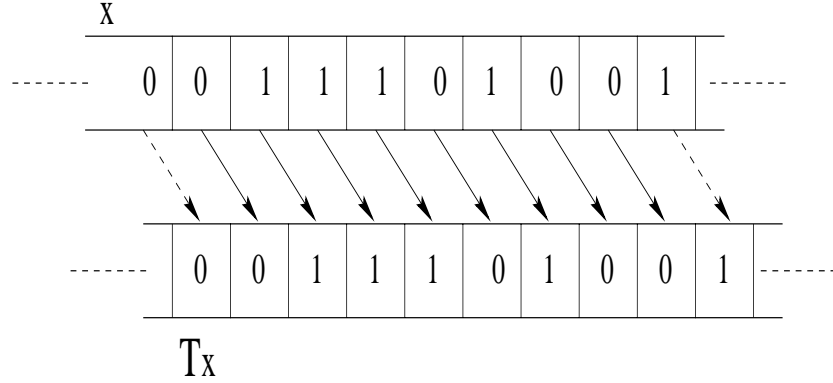


Figure 2-3: The shift operator on *two-way* infinite sequences

is invariant, meaning that for every $x \in E$, $Tx \in E$. If for every invariant set E , $\mu(E) = \mu(E) \cdot \mu(E)$, the source is ergodic. Clearly, such invariant sets either have measure 1 or 0 since the solution of $\mu(E)^2 - \mu(E) = 0$ is $\mu(E) = 0$ or 1. So, with probability 1, an ergodic source is one which has only one invariant set of measure one on which the strong law of large number can be applied since it is invariant. Hence, it is sufficient to model the source accurately in this invariant set (also called an ergodic mode). Furthermore, this modeling can be obtained from a single infinite observation if we apply the strong law of large numbers. This property known as Birkhoff's ergodic theorem is commonly used in practice where ensemble averages are approximated by time averages. The problem of modeling finite objects presents a different challenge where long observations may not be available. To handle finite sequences, for each $n \geq 1$, let the measurable space (A^n, \mathcal{A}^n) be defined by

$$(A^n, \mathcal{A}^n) = \prod_{0 \leq k \leq n-1} (A_k, \mathcal{A}_k) \quad (2.4)$$

We use A^* to denote $\bigcup_{n=0}^{\infty} A^n$. By $(A^\infty, \mathcal{A}^\infty)$ we denote $\prod_{k=0}^{\infty} (A_k, \mathcal{A}_k)$. $(\hat{A}^n, \hat{\mathcal{A}}^n)$ is defined similarly for the reproduction alphabet. On (A^n, \mathcal{A}^n) , a probability measure is obtained by taking the restriction of μ to the event space \mathcal{A}^n and is denoted by

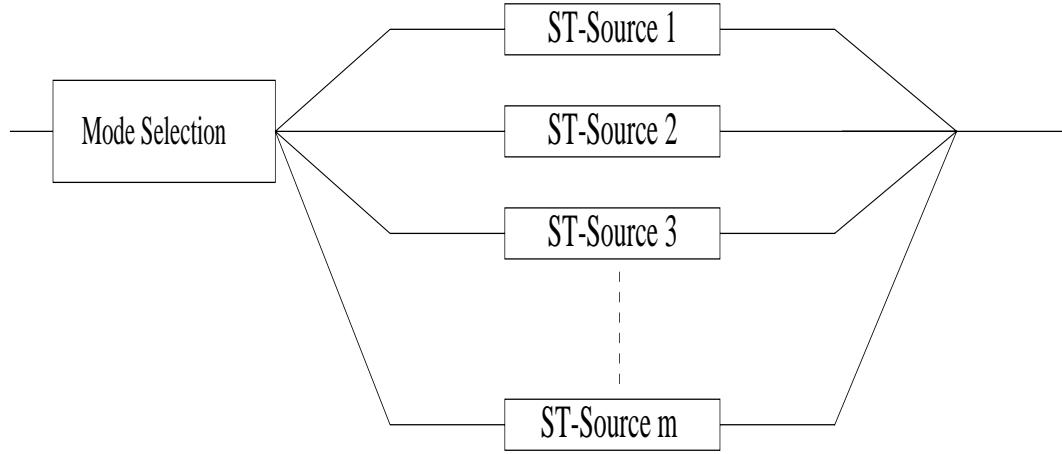


Figure 2-4: A stationary information source. Initially, the initial mode choice module selects one of the stationary sources and the source stays in this mode for the rest of its operation. Hence from any long observation of the source, we can estimate the distribution of one particular stationary source. This source would be ergodic if the initial mode choice almost always select the same stationary source $i, 1 \leq i \leq m$.

μ^n .

$$\forall E \in \mathcal{A}^n, \mu^n(E) = \mu\left(\left(\prod_{k < 0} A_k\right) \times E \times \left(\prod_{k \geq n} A_k\right)\right) \quad (2.5)$$

Elements of A^n can be viewed as finite sequences. We define l as being a function from A^n to \mathcal{N} mapping each element $x \in A^n$ to the length of the sequence, n . Let n and m be two elements of \mathcal{N} , x_n^m denotes $(x_n, x_{n+1}, \dots, x_m)$ if $m \geq n$. If $n > m$, $x_n^m = \Lambda$, the empty string. The n -fragment of x denoted by $(x)_n$ is simply the sequence composed by the first n elements of x : $(x)_n = x_1^n$. For any x, y and z belonging respectively to A^m, A^k and A^{m+k} , $z = xy$ denotes the concatenation of x and y . Clearly, $l(z) = l(x) + l(y)$. We write Γ_x to represent the set of all sequences beginning with x .

$$\Gamma_x = \left\{ w \in A^\infty : (w)_{l(x)} = x \right\} \quad (2.6)$$

Such sets are called cylinders and the measure of a finite sequence is simply the measure of the cylinder it induces. We will use the following abuse of notation: For any measure μ on \mathcal{A} , $\mu^n(x_1^n) = \mu^n(\Gamma_{x_1^n})$.

Clearly, with finite objects, it is not possible to obtain infinite time averages and the ergodic source model does not fit naturally anymore. There is not enough physical evidence to describe accurately the information source with time averages, even with the assumption that these time averages converge to the ensemble averages. More generally, it becomes difficult to attach a physical meaning to the traditional concept of probability [42]. In chapter 3, we will address this problem in a *deterministic* framework. For completeness, we close this section with more standard definitions and notations. The joint probability measure of the pair (X, Y) , with $X \in \mathcal{A}^n$ and $Y \in \hat{\mathcal{A}}^n$ is denoted by $p^n(X, Y)$. The conditional probability measure of the probability of X given Y is denoted by $q^n(X | Y)$. The conditional probability measure of the probability of Y given X is denoted by $\eta^n(Y | X)$. The product probability measure of X and Y is denoted by $\pi(X, Y) = \mu(X)\nu(Y)$. A finite and real-valued function f defined on A is said to be \mathcal{A} -measurable if $\{x : f(x) \in O\} \in \mathcal{A}$ for every open subset O of the real line. Measurable functions are also called random variables or measurements. Their inverse mapping maps σ -algebras to σ -algebras, and this guarantees that we can define a probability measure on their output space. An \mathcal{A} -measurable function f is called μ -integrable if

$$\int |f(x)| d\mu(x) < \infty \quad (2.7)$$

An \mathcal{A} -measurable function f is called μ -unit-integrable on $S \in \mathcal{A}$ if

$$\int_S f(x) d\mu(x) < 1 \quad (2.8)$$

The expected value of f with respect to measure μ is denoted by:

$$E_\mu[f] = \int f(x) d\mu(x). \quad (2.9)$$

Finally, we say that a statement holds almost surely with respect to measure μ if the set on which the statement holds has measure 1 according to μ and we abbreviate this by “ μ -a.s.”.

2.3. Information Measures

Quoting C. E. Shannon [70], the semantic aspects of communication are irrelevant to the fundamental problem of reproducing at one point either exactly or approximately a message selected at another point. The significant aspect in this setting is that the actual message is one *selected from a set* of possible messages. If the number of possible messages is finite, then this number or any monotonic function of it can be regarded as an information measure. Shannon chose the logarithm function for convenience. As mentioned in [70], this measure is close to our intuitive feeling; parameters of engineering importance such as time, bandwidth, etc., tend to vary linearly with the number of possibilities. Furthermore, it is mathematically more suitable and more tractable.

2.3.1 Lossless Measures

Let (A, \mathcal{A}, μ) be a discrete-time source of information. Information is then measured by $\log \frac{1}{\mu(\cdot)}$. This way, events with high probability contains very little information. Rare events having low probability contains a lot of information. In a sense, information is proportional to the amount of *surprise* that is contained in the message. When there is no surprise at all, the probability of the message is one and the information content is null. By $H(\mu^n)$ we denote the n^{th} order entropy or average information of the source:

$$H(\mu^n) = E_{\mu^n}[\log_2 \frac{1}{\mu^n(x)}], x \in A^n. \quad (2.10)$$

The n^{th} -order conditional entropy is defined similarly:

$$H(q^n) = E_{p^n}[\log_2 \frac{1}{q^n(x|y)}], (x, y) \in A^n \times \hat{A}^n \quad (2.11)$$

The concept of entropy-rate is used to extend the notion of entropy for random variables to stochastic processes:

$$H(\mu) = \lim_{n \rightarrow \infty} \frac{H(\mu^n)}{n} \quad (2.12)$$

when the limit exists. For a stationary source it is well known that the limit in equation 2.12 always exists and the entropy rate is well defined. If the process is also ergodic, Shannon, McMillan and Breiman have shown that the entropy rate can be estimated almost surely from a single observation of the source, as stated in the following theorem.

Theorem 1 *For a finite-valued stationary ergodic process $\{X_n\}$ with measure μ , then*

$$-\frac{1}{n} \log \mu(X_0, \dots, X_{n-1}) \rightarrow H(\mu) \quad (2.13)$$

μ -almost surely as $n \rightarrow \infty$.

Proof: See [15].

□

This theorem is commonly called the Shannon-McMillan-Breiman theorem and was proven by Breiman in the almost sure sense. A weaker version of it, where the convergence is guaranteed in probability, is often called the generalized Asymptotic Equipartition Property (AEP). The central ideas in lossless data compression revolve around this property which allows us to partition the set of all sequences of length n into a typical set and an atypical set. As its name indicates, the set of typical

sequences has a very high measure. In fact, its probability measure grows to 1 as n grows. This typical set corresponds exactly to the set of sequences verifying equation 2.13. Focusing on binary alphabet (with no loss of generality), data compression becomes possible when we observe that the cardinal of this set is in the order of $2^{nH(\mu)}$, a number smaller than 2^n , the cardinal of the set of all sequences of length n , since $H(\mu) \leq 1$. This property also shows the central role of the entropy rate in data compression. For precise mathematical definitions of typicality, the interested reader should consult [15]. It is worth mentioning here that typicality is a property of random sequences without any structural or deterministic patterns. Another important quantity in IT is the average mutual information I_n of the joint probability space $(A^n \times \hat{A}^n, \mathcal{A}^n \times \hat{\mathcal{A}}^n, p^n)$ defined by

$$I_n = \sup \sum_{i=1}^{\infty} p^n(F_i) \log_2 \frac{p^n(F_i)}{\pi^n(F_i)} \quad (2.14)$$

where the supremum is taken over all finite measurable partitions $\{F_i\}$ of $A^n \times \hat{A}^n$. Note that when A^n and \hat{A}^n are finite,

$$I_n = H(\mu^n) - H(q^n) = H(\mu^n) + H(\nu^n) - H(p^n) \quad (2.15)$$

The mutual information is a measure of the amount of information contained mutually in sources (A, \mathcal{A}, μ) and $(\hat{A}, \hat{\mathcal{A}}, \nu)$. It plays a major role in IT, specially in Rate Distortion Theory (RDT), whenever we try to analyze the effect of a channel on a message. Indeed, the mutual information between the input and output of a channel describes very well the effect of the channel on the message transmitted by measuring the amount of information that survived the transmission. In IT, the channel is defined by a conditional probability measure $\eta(y | x)$, the probability of observing an output y knowing that the input is x . Ignoring feedback, a discrete

channel is called memoryless (d.m.c) if

$$\eta(y_1^n | x_1^n) = \eta(y_1 | x_1)\eta(y_2 | x_2) \cdot \eta(y_n | x_n) \quad (2.16)$$

For such a channel, consecutive transmission of symbols are independent of one another and does not introduce any form of correlation between them. This observation is formalized in the following theorem that will be used in chapter 3.

Theorem 2 *Let $[A \times \hat{A}, p]$ be the joint process that results from passing a time discrete stationary source $[A, \mu]$ through a memoryless channel³. Then p is ergodic if μ is ergodic.*

Proof: See [6], [28].

□

In other word, a stationary ergodic source output passing through a d.m.c. keeps its properties.

2.3.2 Lossy Measures

RDT is a branch of IT concerned with problems arising when the source entropy exceeds the system capacity. In these cases, since the rate of the source has to be reduced below capacity before transmission, some form of distortion will inevitably results between the original source signal and the received signal. In this work, we focus only on single letter distortion measures. For two sequences $(x)_n$ and $(\hat{x})_n$ of

³There is an important duality between rate distortion theory and channel coding theory. In the former, the source is fixed and we are looking for the channel that minimizes the information rate whereas in the latter, the channel is fixed and we are looking for the source that maximizes the information rate.

length n it is defined by:

$$d_n(x_1^n, \hat{x}_1^n) = \frac{1}{n} \sum_{i=1}^n d(x_i, \hat{x}_i), \quad (2.17)$$

with $d(x_i, \hat{x}_i)$ being a function from $A_i \times \hat{A}_i$ to \mathcal{R}^+ . Let $B_0 = \{0, 1\}$, $B^n = B_0^n$ and $B = \bigcup_{n=0}^{\infty} B^n$, an encoder is a function from Φ_n from A^n to B . A decoder is a function Ψ_n from B to \hat{A}^n . Any set $C = \{y_i : y_i \in \hat{A}^n, 1 \leq i \leq K\}$ of reproducing words is called a code of size K and block length n for the source. The elements of C are called codewords. It is important to note that in classical rate distortion theory we make no computability assumptions⁴ on Ψ_n . A code is D -semifaithful or D -admissible if

$$E_p[d_n(x_1^n, \Psi_n(\Phi_n(x_1^n)))] \leq D, \quad D \in \mathcal{R}^+ \quad (2.18)$$

The size of a D -semifaithful code C of block length n is denoted by $K(n, D)$. To measure information in a lossy setting, consider following construction: Fix $y \in \hat{A}^n$. The set $\{x \in A^n : d_n(x, y) \leq D\}$ will be called a (D, n) -ball with center y or simply a D -ball if n is understood. It is important to note that since A_0 and \hat{A}_0 could differ, some D -balls could be empty. To avoid this, we define D_{min} as a real number equal to the infimum amount of distortion that can be obtained using any coding/decoding scheme Φ_n, Ψ_n . Obviously, if $\hat{A}_0 \supseteq A_0$, $D_{min} = 0$. But in practice it is more common to have $\hat{A}_0 \subset A_0$. From now on, we always assume that such a real number D_{min} exists and that $D \geq D_{min}$. If $A_0 = \hat{A}_0$, and if $D = 0$, the code is called noiseless or faithful. Let $G(S)$ be a union of D -balls that covers $S \subseteq A^n$. $G(S)$ is called a D -cover of S . $N(D, S)$ denotes the minimum number of D -balls needed to cover S .

⁴When the source coding theorem presented in below section 2.4. predicts the existence of a code able to approach the limits of compression, it does not guarantee that the code is recursive. See appendix A. for the definition of a recursive function.

Definition 1 *The operational rate-distortion function $R^o(D)$ is defined as:*

$$R^o(D) = \lim_{\epsilon \rightarrow 0} \lim_{n \rightarrow \infty} R_n(D, \epsilon) \quad (2.19)$$

where

$$R_n(D, \epsilon) = \min_{S: \mu(S) \geq 1-\epsilon} \frac{1}{n} \log_2 N(D, S) \quad (2.20)$$

$R_n(D, \epsilon)$ is a measure of the ratio between the number of bits needed to index the minimum number of balls required to cover S , S being a subset of A^n of measure greater than $1 - \epsilon$. The intuition here is that each element of S can be represented “ D -semifaithfully” by the index of the D -ball containing the element as shown in figure 2-5. It is well known that this definition of the operational rate-distortion function is equivalent to the definition of the information rate distortion function defined below⁵.

Definition 2 *The n th order information rate distortion function is:*

$$R^n(D) = \inf_{q^n \in Q_n(D)} \frac{I_n}{n} \quad (2.21)$$

where $Q_n(D)$ is the class of conditional probability measures q^n for which

$$E_{p^n} [d_n(x_1^n, \hat{x}_1^n)] \leq D. \quad (2.22)$$

The information rate distortion function is defined as:

$$R^I(D) = \lim_{n \rightarrow \infty} R^n(D) \quad (2.23)$$

⁵This equivalence has been shown by R. M. Gray, D. L. Neuhoff and D. S. Ornstein in [39]

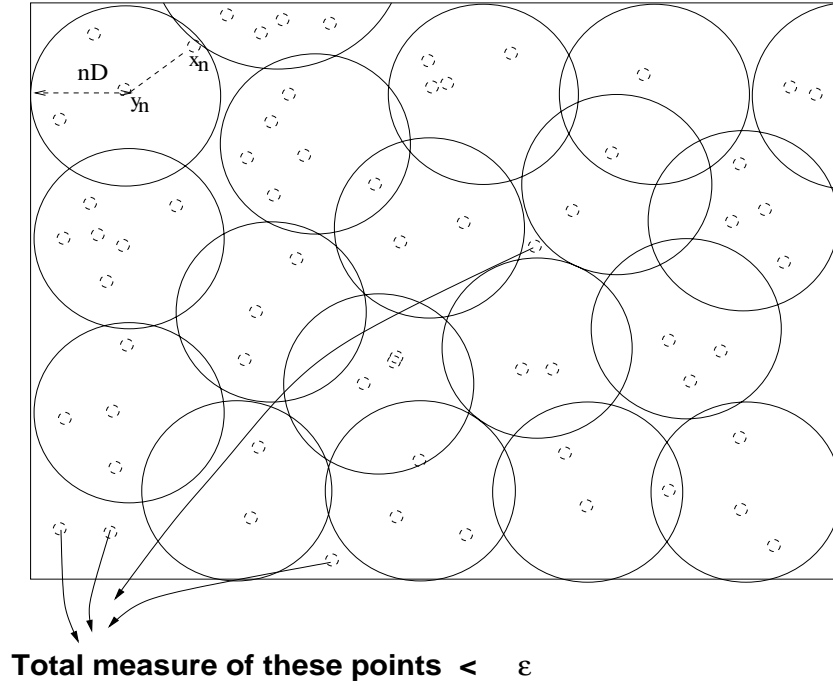


Figure 2-5: The sphere covering problem in rate distortion theory. The small spheres represent finite sequences. The probability measure associated with these sequences would be represented by the area of these small spheres. The objective is to minimize the number of big spheres needed to cover a subspace of sequences of length n with a high measure, greater than $1 - \epsilon$, $0 < \epsilon < 1$.

From now on we denote the rate distortion function by $R(D)$ and we will use both the operational and information rate distortion definitions which are equal in the almost sure sense [39]. To finish this section, we present a theorem that highlights the importance of the rate distortion function.

Theorem 3 *Let the joint source $[A \times \hat{A}, p]$ be stationary and let p^t be the restriction of p to $\mathcal{A}^t \times \hat{\mathcal{A}}^t$, Then the finiteness of the information rate*

$$R = \lim_{t \rightarrow \infty} \frac{I_t}{t} \quad (2.24)$$

is a necessary and sufficient condition for the existence of an invariant, p -integrable function $i(z)$, $z \in A^n \times \hat{A}^n$ such that

$$\lim_{t \rightarrow \infty} \frac{\log f_t(z)}{t} = i(z) \quad (2.25)$$

for p -almost-all z , where $f_t = \frac{dp^t}{d\pi^t}$ is the Radon Nikodym ⁶ derivative of p^t with respect to π^t . If $[A \times \hat{A}, p]$ is ergodic as well, then $i(z)$ is a constant, namely, the information rate R of equation 2.24.

Proof: See [63].

□

In a sense, theorem 3 is the equivalent of theorem 1 for the lossy case. The Radon Nikodym derivative is a rate that generalizes the concept of deriving a real valued function with real valued parameters to general set functions⁷. This theorem tells us that if $[A \times \hat{A}, p]$ is ergodic, then the information rate is always a constant R asymptotically equal to the Radon Nikodym derivative of the joint probability with respect to the product probability measure, removing the expectation from the mutual information on equation 2.24 and allowing us to use the ergodic property of the source and work on a single observation of the source. This will always be possible when the joint process results from passing a discrete ergodic source through a memoryless channel. See theorem 2.

⁶In fact the mutual information between general ensemble can be defined as the logarithm of the Radon Nikodym derivative of p^t with respect to π^t . The average mutual information is then $I_n = \int \log f_n(z) dp^n(z)$ if I_n is finite. See [28].

⁷See [74], [26] and [37] for detailed treatments of the Radon Nikodym derivative.

2.4. Source Coding Theorem and Universal Coding

The quantities defined in the previous section represent the effective rate at which a source generates non redundant information. Focusing on the lossy case, we have to add the requirement that the source output be reproduced with fidelity D . The next question that has to be addressed is if there exists ways to represent information sources at these prescribed rates and if there are ways to encode information below these rates even if our intuition would say no if we look at the definition of the operational rate distortion function. These questions have been addressed by Shannon and yield the following fundamental theorem in source coding theory.

Theorem 4 *Let $[A, \mu]$ be a time-discrete, stationary, ergodic source having rate distortion function $R(D)$ with respect to the single-letter fidelity criterion ρ_n , and assume that a letter y^* exists for which $E[\rho_1(x, y^*)] < \infty$. Then for any $\epsilon > 0$ and any $D \geq 0$ such that $R(D) < \infty$, there exists a $(D + \epsilon)$ -admissible code for $[A, \mu]$ with rate less than $R(D) + \epsilon$. In other words, the inequality*

$$\frac{1}{n} \log K(n, D + \epsilon) < R(D) + \epsilon \quad (2.26)$$

holds for n sufficiently large. No D -admissible source code has rate less than $R(D)$.

That is, for all n ,

$$\frac{1}{n} \log K(n, D) \geq R(D), \quad (2.27)$$

where $K(n, D)$ is the size of the code.

Proof: See [6] for a detailed proof.

□

In its first part, this theorem predicts the existence of block codes with rates arbitrarily close to the rate distortion function for large block size. In its second part,

this theorem proves that the rate distortion function is in fact a lower bound for lossy compression rates by showing that the set of codes with rate less than the $R(D)$ is empty (almost surely). The importance of this result cannot be overstated. It clearly shows that the efficiency of a code can be measured by how close its rate is to the lower bound, the rate distortion function. In this sense, optimality can be reached asymptotically, as the block length increase and we call this an *asymptotically optimal* condition. What this theorem does not provide, even from its proof, is a general procedure to reach $R(D)$. But since its statement, the IT field has grown substantially and powerful techniques have been derived to obtain such asymptotically optimal codes. These techniques share the common property of relying on the knowledge of the source distribution when available. If the distribution is unknown, these techniques try to estimate it and go from there to design an optimal code. In chapter 3, we will use the following restricted version of theorem 4:

Theorem 5 *Let $[A, \mu]$ be a time-discrete, stationary, ergodic source, let $R^1(\cdot)$ be the first-order approximation to its rate distortion function and assume there exists a $y^* \in \hat{A}_0$ such that:*

$$\int d_1(x, y^*) d\mu^1(x) = d^* < \infty$$

Then, for any $\epsilon > 0$ and any $D \geq 0$, if $R^1(D)$ is finite, there exists a value of n and a code B containing K elements of \hat{A}^n such that:

$$\log_2 K \leq n(R^1(D) + \epsilon),$$

and

$$\int d_n(x | B) d\mu^n(x) \leq D + \epsilon,$$

where

$$d_n(x | B) = \min_{y \in B} d_n(x, y)$$

Proof: See [28, 6].

□

The branch of source coding theory that studies the design of codes when the source distribution is unknown, is called universal source coding. A universal code is then a code with rate higher than $R(D)$ and probability of error converging to 0 as the length of the observation increases. More formally, denote the probability of error for the code with respect to the unknown distribution μ at distortion level D by

$$P_D^{(n)} = \mu^n(X_1, X_2, \dots, X_n : d((X_1, X_2, \dots, X_n), \Phi_n(\Psi_n(X_1, X_2, \dots, X_n))) > D) \quad (2.28)$$

A rate R block code for a source will be called universal if the functions Φ_n and Ψ_n do not depend on μ and if $P_D^{(n)} \rightarrow 0$ as $n \rightarrow \infty$ if $R > R(D)$. In [101], a general taxonomy of universal lossy encoding methodologies is described. Three main groups are identified. First note that as proved in [30], for any given coding system that maps a signal vector into one of N binary words and reconstructs the approximate vector from the binary word, there exists a vector quantizer (VQ) with codebook size N that gives *exactly* the same performance, i.e., for any input vector it produces *the same reproduction as the given coding system*. Hence, it is reasonable to identify each group of this taxonomy with particular VQ structures. The first group follows the general approach described in figure 2-6. In this case, a universal codebook is obtained off-line by combining different small codebooks corresponding to different codec systems optimized for particular statistical classes of signals. Two part codes are then used to describe the source data. The first part indexes the codec

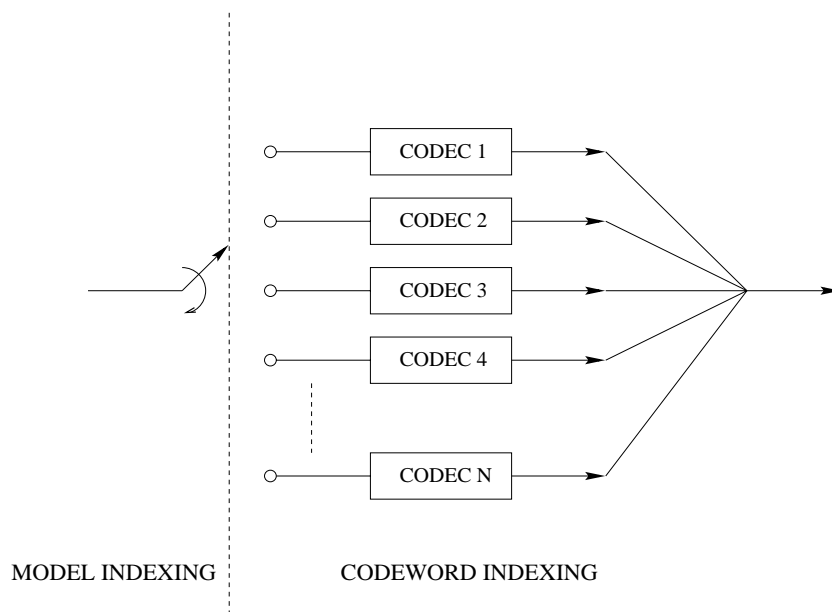


Figure 2-6: Universal Coding System type I.

or small codebook that describes best the source data, and the second part indexes the right codeword in the small codebook of the indexed codec that matches well the source data. In a sense, the first part of the description represents a model for the source object since it targets a particular class of statistical signals. The second part represents the data part of the representation, conditioning on the indexed model. This approach to universal lossy compression has been extensively studied by Effros, Chou and Gray in [22], [13].

In the second group, the universal codebook is not precomputed. It is initially empty and filled in as data comes along, as shown in figure 2. After the encoding of each symbols, the codebook is updated and the update is sent to the decoder so that it can keep track of the encoder codebook evolution and regenerate the source objects with the right codebook at any time. In this case, the codeword used for the representation also have two parts: a model part where the codebook updates are represented and a data part where the source symbols are represented. This approach to universal coding has been extensively studied by Ziv [102]. [103].

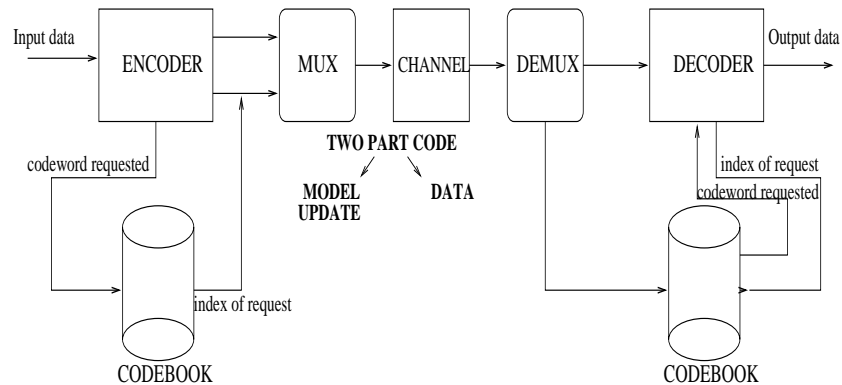


Figure 2-7: Universal Coding System type II.

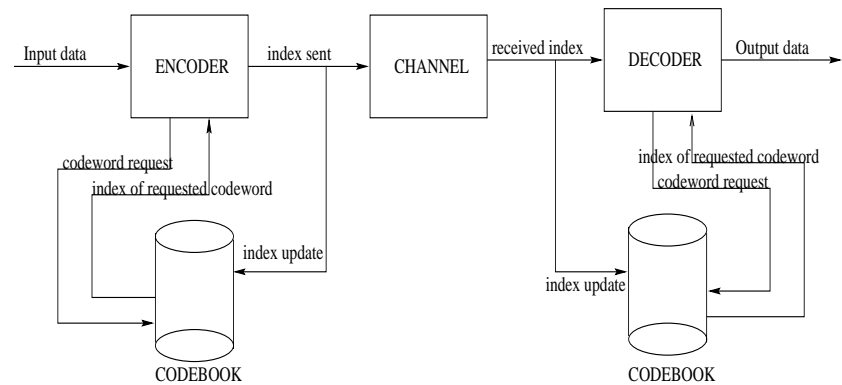


Figure 2-8: Universal Coding System type III.

The third group is similar to the second one. It forms a distinct category simply because codebook updates do not have to be sent. But it preassumes that the encoder and decoder have agreed on systematic ways to build their codebook respectively from incoming data. In this case, the codewords used for the representation have only one part, the data part. The model part is skipped because of the pre-agreement between the sender and receiver on how to build the codebook from incoming data. Extensive work on this technique has been done Zhang and Weir [101] and also Goyal [34].

Although not highlighted in the third group, an important principle comes out from all universal coding techniques: the separation principle between model and

data. The model part represents all the possible sequences that can be represented with this source. The data part is no more than an index to the sequence corresponding to the particular object to code. In a sense, the model part removes all the regularities from the data. In IT, these regularities are statistical patterns and the goal of the encoder is to minimize as much as possible the sum of the model description length and the data description length. Note that this minimization of description length can be generalized to more general problems than data compression. Indeed, in detection and estimation theory, maximum likelihood (ML) and maximum a posteriori (MAP) estimates are commonly used. In the former, the goal is to find an explanation of a phenomenon by choosing the explanation that maximizes the conditional probability of the observation conditioning on this explanation. By taking the logarithm of the inverse of this probability this maximization becomes a minimization of the description length of the observation conditioning on explanations. In the latter, the goal is to find an explanation of a phenomenon by choosing the explanation that maximizes the joint probability of the observation and the explanation. Using Bayes law and taking again the logarithm of the inverse of this joint probability, it is easy to see intuitively that this maximization is equivalent to a minimization of the description length of the explanation summed with the description length of the observation conditioning on the explanation. The explanation is commonly called the model. The observation is called the data. Hence, the goal becomes the minimization of the model description length summed with the data description with help from the model. This goal is precisely called the Minimum Description Length principle (MDL) and highlights once more the importance of the separation between model and data. In this sense, MAP estimation is closely related to universal coding as pointed by Rissanen in [66] and it seems intuitively correct to reformulate the data compression problem into a *data understanding* problem

where we are seeking the best explanation for an observation of a source. This understanding problem requires a good estimation of the machinery hidden inside the box representing the information source and a good representation of the source output, conditioning on this good source model.

Finite object modeling is a difficult challenge with such a probabilistic approach. The asymptotic nature of the definition of universal coding prevents us from addressing this problem in this setting. It limits significantly the practical domain of application of these ideas in media representation. Also, for visual information, things become much more complex when we consider the HVS at the end of the communication system. While the brain still remain a complex mysterious organ, progress in biology have proposed valid theories approximating it with a large finite state machine (or a practical computer). Indeed, electrical models for neurons have been developed to understand the transmission of information in the brain and they tend to support the finite state model. This could explain why images do not appear typical or patternless to us, although current practices in image processing treat them as typical random events, following the guidelines provided by IT that considers only typical sequences.

2.5. Conclusion

In this chapter, we have introduce fundamental concepts that will be used in this thesis. We have done a quick and certainly not exhaustive survey of the field of lossy universal source coding. We have finished it by highlighting the importance of the separation principle between model and data. In the next chapters, we will argue that this separation will always be at the heart of the design of any coding technique. It might not be strikingly visible in certain cases because of prior assumptions made on the model class that is considered by the technique. For instance, the third

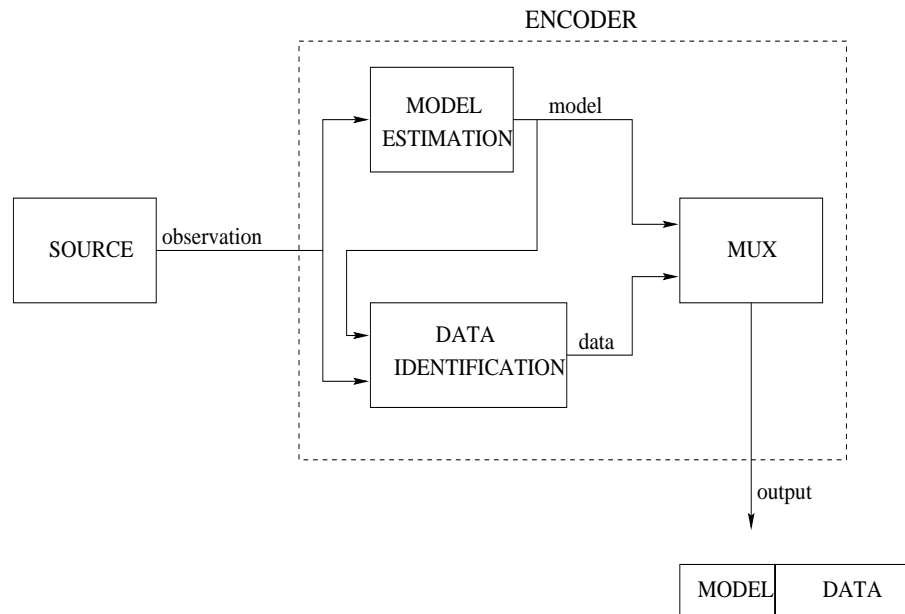


Figure 2-9: Universal coding and the separation principle between model and data.

group of universal coding technique described in the previous section restricts its model search to a narrow group by restricting the operations performed at both end of the communication system for the generation of the codebook. Nevertheless, the separation principle is inherently present in all these design techniques. In the following, we will add a new group in this taxonomy that will allow us to classify all lossy coding techniques, from traditional entropy-based methods to modern methods like model-based and fractal-based coding by formalizing and revisiting the notion of models for information sources. To do so, we will see that it will require a novel approach to the universal coding problem from a different theoretical angle that do not use entropy and rate distortion theoretic concepts. Instead, a deterministic approach to measure information that uses the Kolmogorov complexity, will be adopted and this is precisely the topic of the next chapter. This way, non typical sequences will also be considered, specially for the representation of finite visual signals.

Chapter 3

Complexity Distortion Theory

3.1. Introduction

The concept of Algorithmic or Kolmogorov complexity was introduced simultaneously by R.J. Solomonoff in [76], A.N. Kolmogorov in [44] and G. Chaitin in [11] in different settings. Solomonoff was concerned with the application of Bayes's Rule in statistics and inductive inference problems. He introduced the "Kolmogorov" complexity as an auxiliary concept to obtain a universal a priori probability [52], measuring the length of shortest effective descriptions of objects. His work provides the theoretical foundation for the Minimum Description Length (MDL) principle. MDL was introduced by Jorma Rissanen to tackle grave problems with density estimation arising from the difficulty of finding a way to give a constructive definition of probability which would allow us to recognize it in a clear cut way. All attempts to attach inherent probabilities to observed data have failed, as they indeed must [66] since they require infinite observations. This problem is even more striking when the phenomenon under investigation is finite. For instance, consider a finite sequence of zeros and ones and let's try to infer from this finite observation a description model. The use of time averages that estimate probabilities is hard to justify simply because there is not enough physical evidence to guarantee convergence to the

true distribution of the object [42]. According to the MDL principle and following Occam's Razor principle, the best explanation would be the one that describes the object with the shortest representation that includes the model description and the object description with this model.

Kolmogorov studied independently shortest effective descriptions of individual objects for its own sake and this is probably why the term Kolmogorov complexity is so deployed when we refer to the complexity of an object. His work links short descriptions to randomness and universal object distributions. It provides an interesting theoretical framework where we can discuss the modeling of finite objects. The main idea is that an object with a short description should not be considered random. For example the following sequence 01010101... has a structure that allows it to be represented in a very compressed form. Randomness becomes associate to a lack of deterministic pattern. In the following sequence, 101101100010..., it is more difficult to identify a deterministic pattern. Such a sequence cannot be compressed much and is labeled random. In contrast with Shannon's information theory which defines description length from randomness or probabilities, Kolmogorov complexity theory formalized randomness from description lengths.

Chaitin's approach grew from a different and much more ambitious motivation and yielded Algorithmic Information Theory. The result of his work is a deeper understanding of Gödel incompleteness results [31] and how they affect the search for a universal mathematical theory as conceived by David Hilbert. In an attempt to bring more light to Hilbert's dream of such a theory with no paradoxes, he came up with the need to define the amount of randomness in individual objects as did Kolmogorov. Despite being of fundamental importance for science in general, this approach is beyond the scope of this thesis but it illustrates once more the large application scope of information theoretic notions.

In all cases, the whole theory grows from the fact that, in contrast with Shannon's IT, it seems very natural to define the amount of information in relation to an individual object rather than in relation to a set of objects from which the individual objects may be selected. To formalize this individual notion of information, a description language has to be fixed. This language must be universal to allow the description of all conceivable objects. It also must be implementable, allowing us to mechanically reconstruct these objects from their descriptions. The computational model used is the universal Turing Machine and this choice is justified by the Church-Turing thesis stating that the class of algorithmically computable numerical functions (in the intuitive sense) coincides with the class of partial recursive functions, or the class of functions that can be computed on a Turing machine¹.

Kolmogorov Complexity has grown substantially since its introduction in [76, 44, 11]. It has concentrated, however, on lossless representations. In this chapter, we extend this notion of complexity to lossy representations by adding a distortion component to the mathematical picture. This extension gives rise to Complexity Distortion Theory [81, 79], a unified perspective for information representation where information – within a given distortion bound – is a deterministic notion measured by the Kolmogorov complexity. We investigate the relationship between the program-size complexity and Shannon's information measures in both the lossless and lossy context. We show that these two frameworks predict the same results for a large class of objects which are called algorithmically random objects. These objects do not have deterministic patterns. For non random objects, these two approaches diverge.

We start this chapter with a formalization of what is a programmable decoder or a universal Turing machine and discuss the limitation of this model. We then define

¹See appendix A. for precise notions on recursive functions.

the Kolmogorov complexity and the notion of randomness tests before proposing an equivalence between the Kolmogorov complexity and Shannon's entropy. We extend these results to the lossy case after introducing the concept of Complexity Distortion Function.

3.2. Universal Turing Machines

As mentioned above, the great German mathematician David Hilbert had a dream of a universal mathematical theory, cleaned from ambiguities and paradoxes. The idea was to establish the underlying consistency in all mathematics. His effort was proved impossible by Kurt Gödel [31] and later by Alan Turing [88]. In both cases, the conclusion was reached by constructing ambiguous statements in a very formal language. Gödel reached this conclusion using a language very close to LISP. Turing came up with the Turing machine, a mechanical way to generate theorems from axioms. This machine turns out to be the accepted mathematical model for computers. As pointed by Chaitin, it may not be wrong to think that the computer was invented accidentally in an attempt to shed light on Hilbert's dream.

The concept of Turing machine can be derived from simpler systems called finite automaton (FA) consisting of a finite set of states and a set of state transitions that occur on input symbols chosen from an alphabet, say $\{0, 1\}$. An example of FA is shown in figure 3-1. It is well known that the set of languages recognized by these machines corresponds to all regular expressions [40] and languages like $NMN = \{1^n 0^m 1^n \mid n, m > 0\}$ do not belong to this set. Indeed, this language cannot be recognized by a finite state system. The value of n has to be stored during the recognition process and n being unbounded, for any finite state system, there is an infinite number of large values of n for which sequences of the type $1^n 0^m 1^n$ cannot be recognized. Adding an *infinite* memory to an FA allows it to store temporary

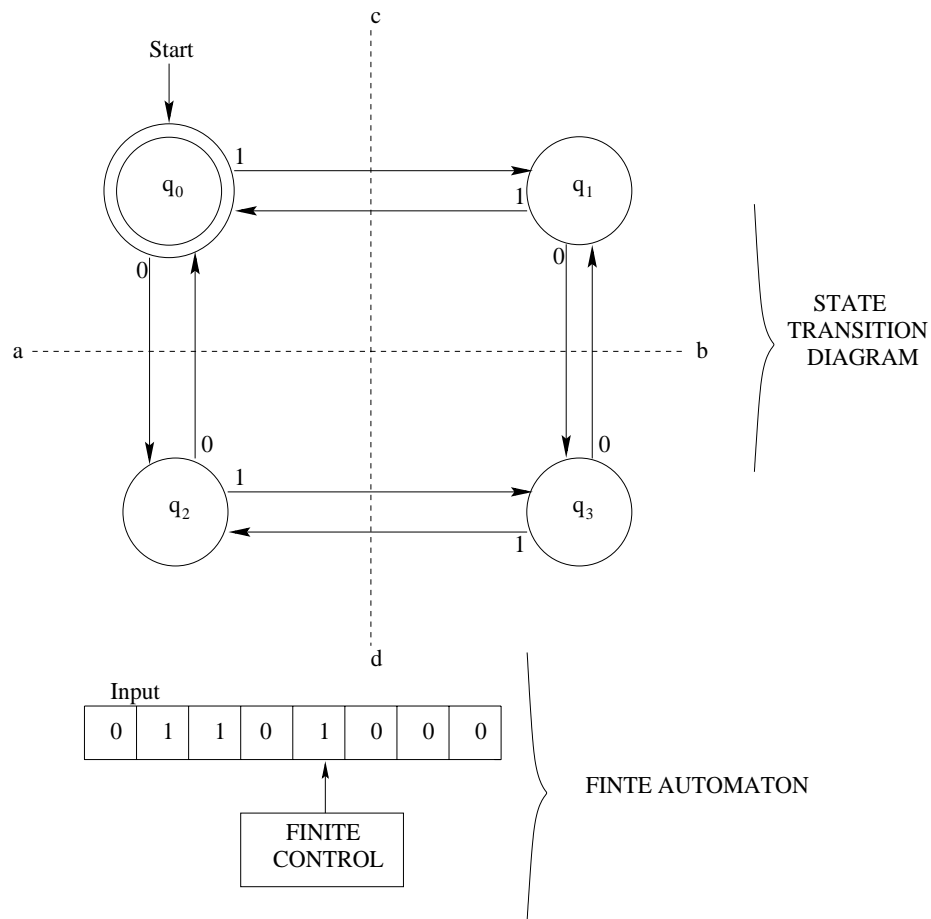


Figure 3-1: Finite Automaton: The top figure shows the transition diagram of a finite automaton. Assuming that the initial and final state are both q_0 , this automaton will recognize all sequences in which both the number of 0's and the number of 1's is even. The example shown in the bottom will leave the automaton in state q_3 at the end of the computation.

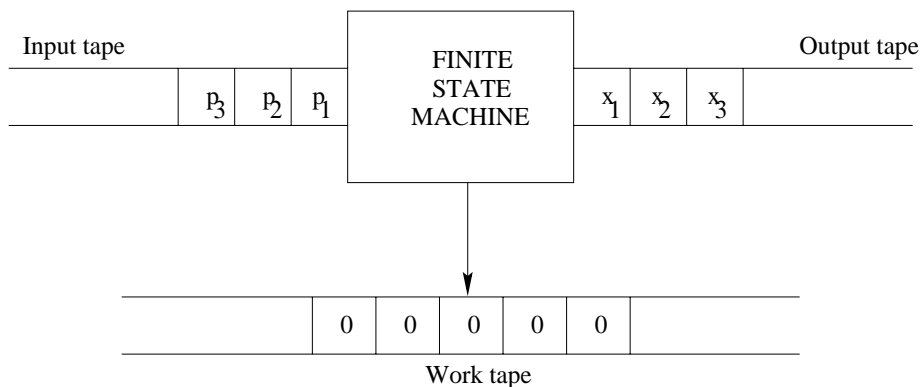


Figure 3-2: A Turing machine with one work or memory tape.

information during its computation and recognize a very large set of languages. A FA with access to an infinite memory is a Turing machine (TM) as shown in figure 3-2. This simple model of computation is very powerful and for any recursive function, there is a Turing machine able to compute it. From the Church-Turing thesis, it is largely accepted that there is a one to one mapping between recursive functions and algorithmic problems in the intuitive sense. Hence, for each algorithmic task, there is a Turing machine that can perform it.

Of further importance is the existence of a special Turing machine called the Universal Turing machine (UTM), able to simulate the actions of any other Turing machine. The intuition behind the existence of such a special machine lies behind the fact that the set of all Turing machines is *countably* infinite. Thus, we can construct our UTM by allowing it to accept as input a *two-part* program with the first part of the program indexing a particular Turing machine from the countably infinite set of all Turing machines. The second part of the program is simply the input that should be fed to the Turing machine indexed by the first part. Note that the use of two-part code here is reminiscent of universal coding procedures described in chapter 2, section 2.4.. This link is not accidental and we will use it below to come up with an extended notion of universal source coding. It is very helpful for the rest

of this thesis to also view a universal Turing machine as a universal language (also called a Turing complete language) like C or LISP, in which we can express almost all the algorithmic task that we would like to perform.

Despite its universality, it is wrong to think that this device can solve any problem. Turing showed that there is a class of problems that cannot be solved by any UTM. Of particular importance is the “Halting problem”. Turing showed it is not possible to design a Turing machine able to take in input any program and always halt either in an accepting state if the input program does halt or on in a rejecting state if the input program never halts. The consequence of the Halting problem are devastating not only in computer science but also in coding theory and mathematics as it shows that the existence of a class of problems that cannot be solved in an algorithmic or mechanical logical way. It destroys Hilbert’s dream.

3.3. Kolmogorov Complexity

Consider a traditional communication system where the decoder is replaced by a universal Turing machine as shown in figure 3-3. The codewords traveling through the channels are now two-part programs, the first indexing a particular TM or algorithm and the second part representing the data that the algorithm needs to reproduce the source message. This communication system follows our natural way of conceiving communications. Indeed, humans do communicate using languages with strict syntactic and grammatical rules. These languages differ from computer languages simply because of their domain of application and their ambiguity but the idea behind their use is the same. In such a communication setting, it seems natural to define the amount of information in an object by the length of its shortest description in the language. This approach yields directly to the notion of Kolmogorov complexity.

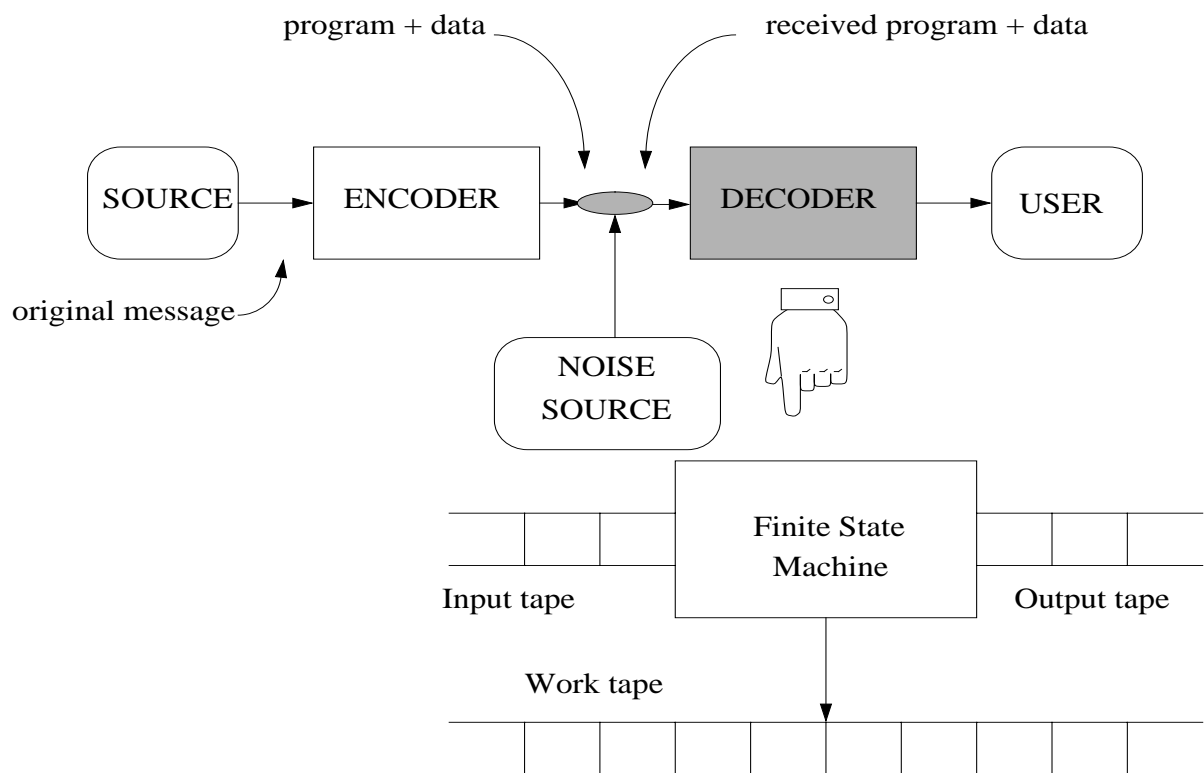


Figure 3-3: The programmable communication system. Notice that two-part codes representing algorithms and data are sent through the channel.

3.3.1 Individual Information Measure

The plain Kolmogorov complexity of sequence x_1^n is equal to the length of the shortest program written for a Turing machine (TM), that computes x_1^n and halts. Following the equivalence between TM's and recursive functions as discussed in Appendix A., we can formally define the Kolmogorov complexity as follows:

Definition 1 *Let F be an arbitrary partial recursive function. Then the complexity of the sequence $x \in A$ with respect to F is:*

$$K_F(x) = \begin{cases} \min\{l(p) : F(p) = x\}, \\ \infty \text{ if } \forall p \in A \ F(p) \neq x \end{cases}$$

The sequence p for which $F(p) = x$ is called the code or program or short description for x . Definition 1 was proposed by Kolmogorov in [44]. It is often called the plain Kolmogorov complexity. Although more suitable for information understanding than the ensemble measure proposed by Shannon, the Kolmogorov complexity as defined above is not completely satisfactory. This information measure is sub-additive only up to a logarithmic term. Also, it is not monotonic over prefixes². Intuitively the complexity of the concatenation xy of two strings x and y should always exceed the complexity of x . It is not the case with the plain complexity³. Another problem surfaces when we try to associate a probability measure with K . By defining $P(x) = \sum 2^{-l(p)}$, the sum being taken over all inputs p for which the assumed computer output x and halts, we would like $P(\cdot)$ to be a probability measure to have a better formalism for Solomonoff's induction problems. Unfortunately, this summation diverges⁴. One way to solve these problems except the non monotonicity

²See example 2.5 in [52].

³See example 2.10 in [52].

⁴See [52], p 170.

over prefixes is to use the prefix complexity measure introduced by Levin, Gacs and extended by Chaitin. In this case, programs are constrained to be prefix free and Kraft's inequality [52, 15] can be applied to associate a probability measure to K . In this thesis, we use the prefix complexity for simplicity, although the main results proposed do hold for the other variants of Kolmogorov complexity.

Definition 2 *Let F be an arbitrary partial recursive function with a prefix domain⁵. Then the prefix complexity of the sequence $x \in A$ with respect to F is:*

$$C_F(x) = \begin{cases} \min\{l(p) : F(p) = x\}, \\ \infty \text{ if } \forall p \in A \ F(p) \neq x \end{cases}$$

Theorem 1 *There exists a partial recursive function F_0 (called optimal) such that for any other partial recursive function G ,*

$$C_{F_0}(x) \leq C_G(x) + O(1) \tag{3.1}$$

Proof: See [44].

□

The optimal function is also called the universal function. The intuition behind this theorem is the existence of a universal computer, or UTM, able to simulate the actions of any other computer. Since Turing machines are effectively enumerable, a possible description for x is the code for x on G preceded by the index or Gödel number of G in the effective enumeration of Turing machines. The theorem follows naturally from this construction. As a result, we will drop the subscript referring to the partial recursive function and use $C(x) = C_{F_0}(x)$ as a notation for the complexity of sequence x .

⁵A prefix domain is a set of sequences where no sequence is the proper prefix of another.

Definition 3 *The conditional complexity of a sequence x given the knowledge of another sequence y is*

$$C(x | y) = \begin{cases} \min\{l(p) : F_0(p, y) = x\}, \\ \infty \text{ if } \forall p \in A \ F_0(p, y) \neq x \end{cases}$$

We have dropped the subscript referring on the partial recursive function used in the definition because theorem 1 does hold in this case for exactly the same reasons mentioned above in the unconditional case.

Theorem 2 *There is a constant c , such that for all x and y ,*

$$C(x) \leq l(x) + c \tag{3.2}$$

$$C(x | y) \leq C(x) + c \tag{3.3}$$

Proof: See [52].

□

The following theorem is a negative result known as the Noncomputability theorem [52].

Theorem 3 *C is not partial recursive.*

Proof: See [52], [105].

□

It is a negative results since it proves that any attempt to compress maximally sequences cannot be performed on a TM. It is a natural manifestation of the Halting problem. It prevents us from hoping to find a mechanical way to find shortest representations of objects. Fortunately, the next theorem states that it is possible to approximate C .

Theorem 4 *There is a total recursive function $\Phi(t, x)$, monotonic decreasing in t , such that*

$$\lim_{t \rightarrow \infty} \Phi(t, x) = C(x) \quad (3.4)$$

Proof: According to [105], this theorem is due to Kolmogorov. Its proof can be found in [105] and [52]. Since the proof hints at how computational resource bounds at the decoding end remove the Halting problem, we reproduce it here. For each value of t , it is possible to construct a prefix Turing machine that runs the reference Turing machine fixed in theorem 1 for no more than t steps on each program with length at most $l(x) + c$. We define $\Psi(t, \cdot)$ as the partial recursive function computed by this Turing machine. If for some input programs p , the computation halts with output x , then we define $\Phi(t, x) = \min\{l(p) : \Psi(t, p) = x\}$. Else, $\Phi(t, x) = l(x) + c$. Clearly, $\Phi(t, x)$ is recursive, total⁶, and non increasing with t . Also, the limit in equation 3.4 exists since for each x , there is a t such that the reference Turing machine halts and outputs x after t steps starting with input p with⁷ $l(p) = C(x)$.

□

Although we can approximate $C(x)$ from above, it does not mean that we can decide whether $\Phi(t, x) = C(x)$ or not. In more intuitive words, the approximation gets closer and closer to C but it is impossible to know how close it gets. Since $\Phi(t, x)$ is recursive, it becomes possible to introduce computational resource bounds in definitions 3.3.1 and 3 and obtain a recursive measure of information.

Definition 4 [52] *Let $\Phi^{t,s}$ be a partial recursive function computed by a Turing machine such that for any $x \in B$, the computation of $\Phi^{t,s}(x)$ requires less than*

⁶Every finite x has a value assigned to it by Φ .

⁷Since objects are finite, there is a value t^* corresponding to the number of steps the optimal description for x (with length equal to $K(x)$) takes when it is placed in input on the reference Turing machine.

$t(n)$ steps (time) and uses less than $s(n)$ tape cells (memory), $t(\cdot)$ and $s(\cdot)$ being total recursive functions. The resource-bounded Kolmogorov complexity $C_{\Phi}^{t,s}$ of x , conditional to Φ and y ($y \in B$), is defined by

$$C_{\Phi}^{t,s}(x | y) = \begin{cases} \min\{l(p) : \Phi^{t,s}(p, y) = x\}, \\ \infty \text{ if } \forall p \in A \ F(p) \neq x \end{cases}$$

When $t(\cdot)$ and $s(\cdot)$ are total recursive, $C_{\Phi}^{t,s}$ is recursive. With resource bounds, a weaker version of the invariance theorem (theorem 1) exists.

Theorem 5 *There exists a universal partial recursive function Φ_0 , such that for any other partial recursive function Φ , there is a constant c such that*

$$C_{\Phi_0}^{ct \log t, cs}(x | y) \leq C_{\Phi}^{t,s}(x | y) + c, \quad (3.5)$$

for all x and y . The constant c depends on Φ but not on x and y .

Proof: See [52].

□

Here again, this theorem allows us to drop the subscript Φ in $C_{\Phi}^{t,s}$ and write $C^{t,s}$, provided that the statement we make is not sensitive to the additive constant term in the complexity of each string, the multiplicative logarithmic factor in the time complexity, and the multiplicative constant factor in the space complexity.

In contrast with Shannon's information measure, this definition can easily include computational bounds at the decoding Turing machine. In practice the decoding time and the memory size are limited. By $C^{t,s}(x_1^n)$ we denote the shortest prefix free program that represent x_1^n , using less than $t(n)$ steps and less than $s(n)$ memory cells at the decoder. As shown in figure 3-4, this algorithmic approach to measure information allows to define the amount of randomness, or the algorithmic

probability in individual objects (See [76]), in contrast with Shannon's Information Theory (IT) which uses randomness or probability theory to define information.

3.3.2 Randomness Tests

In contrast with Shannon's ensemble information measure which is based on probabilities, the Kolmogorov complexity is an individual information measure. Another major difference between these two measures of information is in the motivation behind their introduction. Shannon's measure was introduced to tackle a communication problem and is derived from probability theory. The main motivation behind the introduction of the Kolmogorov complexity is to characterize the amount of randomness in individual objects. In this case, an information theoretic approach is used to derive a theory of randomness. As explained in [43], random sequences possess three properties:

1. being typical: This property was pointed out by Martin-Löf [54]. A random sequence is typical as opposed to be very special like an infinite sequence of zeros. The property of being typical is identical to the property of belonging to any reasonable majority. More formally, the set of random sequences should have measure 1. As a result, when we choose some object at random, we are almost sure that the object belongs to this majority and we expect to get a typical sequence. In traditional information theory, typical sequences x are the one with probability $p(x)$ close to $2^{-l(x)H}$, H being the entropy of the source (assuming that this entropy exists). The Shannon-McMillan-Breiman theorem shows that this set of sequences has measure one for stationary ergodic sources.
2. being chaotic: This property was pointed out by Kolmogorov. A random sequence is chaotic in the sense that it has no simple law governing the alternation of its terms. Traditional information theoretic notion of typicality fails



	COMPLEXITIES	PROBABILITIES
STOCHASTIC APPROACH (SHANNON)	From Probabilities to Complexities 	
INDIVIDUAL APPROACH (KOLMOGOROV)	From Complexities to Probabilities 	

Figure 3-4: Duality between Shannon's entropy and Kolmogorov complexity.

to capture this⁸. It is very easy to think of a typical sequence according to IT but with a very short mechanical description, capturing all the regularities in this sequence.

3. being stochastic: This property was pointed out by von Mises. The frequency of zeros in the beginning segments of a random sequence (assuming a Bernoulli $\frac{1}{2}$ process), must converge to $\frac{1}{2}$. And this effect must be observed not only for the entire sequence but also for any of its *properly chosen* subsequences. The term *properly chosen* is here to guarantee that there are no gambling strategy. A player betting in fixed amounts on the outcomes of any random sequence after seeing all the previous outcomes cannot obtained any gain in the long run.

⁸For example, assume a Bernoulli source with parameter $p = \frac{1}{2}$, then $x = 010101\dots$ will have an empirical probability close to 2^{-nH} for large n although this sequence is quite regular.

There have been many unsuccessful attempts to formalize these properties. In 1966, Martin-Löf [54] gave a definition of random sequences based on Kolmogorov complexity and on statistical tests verifying all three properties for infinite sequences. An interesting aspect of this work is that it also applies to finite sequences by using levels of randomness. All statistical tests are effective such that they can compute at each level of randomness which sequences should be labeled not random. This observation yields definitions of randomness for individual sequences. These precise statements are presented in appendix B.. The reader who is not familiar with these notions should consult appendix B. or [52] before going any further in this thesis.

3.4. Equivalence with Information Theory

Despite their conceptual differences, the Kolmogorov complexity and Shannon's entropy are equivalent for a very wide class of information sources. In this section, we establish these links and identify precisely the set of sequences where the equivalence does not hold. This set corresponds to non random sequences with strong deterministic or mechanical patterns.

3.4.1 Fundamental Theorem

Levin and Zvonkin were the first one to formulate an equivalence between Shannon's entropy rate and the Kolmogorov complexity. In [105] the following theorem is formulated without proof:

Theorem 1 *For a stationary ergodic source (A, \mathcal{A}, μ) with a recursive probability measure μ ,*

$$\lim_{n \rightarrow \infty} \frac{C(x_1^n)}{n} = H(\mu), \quad x_1^n \in A^n \text{ } \mu\text{-a.s.} \quad (3.6)$$

In this theorem, the ergodic decomposition of stationary process allows us to drop the ergodicity assumption if we allow the introduction of an expectation term on the Kolmogorov complexity. The importance of this result cannot be overstated as it clearly shows that despite introducing a mechanical structure at the decoding end of Shannon's classical system, we do not lose much in terms of performance. Furthermore, the proposed proof will define precisely the null set of infinite sequences where the equivalence does not hold. For finite sequences, we will derive upper bounds on the measure of that set. A formal proof of this result is presented below but first, we present the intuition behind this result.

Intuition behind Theorem 1

The equivalence between Shannon's probabilistic approach and Kolmogorov's deterministic approach holds for a large class of sequences that are called algorithmically random. Sequences from this set do not exhibit any form of regularity that could be exploited by a computer program. Hence, the best way to represent them is with a probabilistic approach. If μ is recursive, there is a mechanical way to describe it. In this case, a sequence can be described with a computer program specifying μ , followed by an efficient entropy coding technique. From such a code construction, we can show that the Kolmogorov complexity of an infinite observation of a stationary source with recursive probability measure is upper bounded by the length of this simple two-part description. Since the description of the probability measure is independent of the observation length, and since the length of the second part of the code is close to nH , H being the entropy rate of the source, we claim that

$$\limsup_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \leq H(\mu), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.7)$$

Note that in the derivation of this upper bound, we do not use Shannon's source coding theorem that predicts the existence of codes operating at rates close to the

entropy rate for stationary ergodic processes. We cannot use it here simply because Shannon's source coding theorem does not make any computational assumptions on the decoding device Ψ_n . Hence, it does not guarantee that this function is actually recursive. Instead, we use the concept of Markov types [73, 18]⁹ with a known coding procedure [73] and use the recursive assumption on the source probability measure to come up with a code that can actually be decoded on a UTM. Also, note that the two-part structure of the code which is a key characteristic of universal coding techniques. The first part of the code describes the model used, here a probabilistic model (the source distribution). The second part of the code describes the actual data, the observation, with help from the probabilistic model described in the first part. To finish the proof of the theorem 1, it remains to show that:

$$\liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \geq H(\mu), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.8)$$

This is done by proving that the set $\{x \in A^{typ} : \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} < H(\mu)\}$ is a null set, where A^{typ} is the set of sequences verifying the Shannon-McMillan-Breiman theorem. In other words, we have to prove that according to the source probability measure, it is very unlikely to have a sequence with Kolmogorov complexity below the length of its Shannon-Fano code. This conclusion is obtained directly with a universal randomness test linking Shannon-Fano code lengths with Kolmogorov complexities. Shannon-Fano code lengths are then linked to the entropy rate of the source via the Shannon-McMillan-Breiman theorem proved by Breiman in the almost sure sense.

⁹The reader who is not familiar with Markov types should consult appendix C. or [73]. For a more general a complete treatments on types, she should refer to [16].

3.4.2 Proof of Fundamental Theorem

We proceed with the proof of theorem 1. In the first part, we show the following lemma:

Lemma 1 *For a stationary ergodic source (A, \mathcal{A}, μ) with a recursive probability measure μ ,*

$$\limsup_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \leq H(\mu), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.9)$$

Proof of lemma 1:

For a lossless representation, we can represent the sequence x_1^n by the following two-part code proposed in [73]:

1. The index of the Markov k -type of x_1^n is transmitted.
2. The index of x_1^n in the type class $T_k(x_1^n)$ is transmitted.

The first part of the code requires only $m = o(n)$ if $k = \lfloor \frac{1}{2} \log_{|A_0|} n \rfloor$. The second part converges to n times the entropy rate as proved in [73]:

$$\lim_{n \rightarrow \infty} \frac{l(\Phi_n(x_1^n))}{n} = H(\mu), \quad \mu\text{-a.s.} \quad (3.10)$$

Φ_n being the encoding function described above. Since the proposed code is prefix free [73], it is a valid program for a prefix Turing machine. Furthermore, the decoding operations are clearly recursive meaning that there is a prefix Turing machine able to decode this code. This machine has to reproduce all sequences with Markov type equal to the one that has been sent and then the machine indexes the right sequence from the second part of the code. Therefore, we conclude that:

$$C(x_1^n) \leq l(\Psi_n(x_1^n)) \quad (3.11)$$

and also that:

$$\liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \leq \limsup_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \leq H(\mu), \mu\text{-a.s.} \quad (3.12)$$

proving lemma 1.

□

To prove theorem 1, what remains to be done is to prove the following lemma:

Lemma 2 *For a stationary ergodic source (A, \mathcal{A}, μ) with a recursive probability measure μ ,*

$$\liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \geq H(\mu), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.13)$$

Proof of lemma 2:

Lemma 1 in appendix B. implies that for any $x \in A^\infty$, if $\rho_0(x | \mu) < \infty$ then x is random with respect to μ . In this case, Let $\epsilon = \rho_0(x | \mu)$. For any $n \in \mathcal{N}$,

$$\frac{-C(x_1^n | \mu) - \log_2 \mu(x_1^n)}{n} \leq \frac{\epsilon}{n} \quad (3.14)$$

Since μ is assumed recursive, $C(x_1^n | \mu) \leq C(x_1^n) + O(1)$, the constant term being present to take into account the length of the description of μ .

$$\frac{-\log_2 \mu(x_1^n)}{n} \leq \frac{\epsilon}{n} + \frac{C(x_1^n | \mu)}{n} \leq \frac{\epsilon}{n} + \frac{C(x_1^n) + O(1)}{n} \quad (3.15)$$

In the limit as n grows, we get

$$\liminf_{n \rightarrow \infty} \frac{-\log_2 \mu(x_1^n)}{n} \leq \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \quad (3.16)$$

the last equation holds for any μ random sequences and this set has μ -measure 1. Therefore, the set

$$\left\{ x \in A^\infty : \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} < - \liminf_{n \rightarrow \infty} \frac{\log_2 \mu(x_1^n)}{n} \right\} \quad (3.17)$$

is a null set. Recall the Shannon-McMillan-Breiman theorem proving that for a finite-valued stationary ergodic process $\{X_n\}$ with measure μ ,

$$- \frac{1}{n} \log \mu(X_0, \dots, X_{n-1}) \rightarrow H(\mu) \quad (3.18)$$

μ -almost surely as $n \rightarrow \infty$. Since the intersection of a null set with another set is still a null set, we can use the Shannon-McMillan-Breiman theorem to conclude that for a stationary ergodic source,

$$\mu \left\{ x \in A^\infty : \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} < - \lim_{n \rightarrow \infty} \frac{\log_2 \mu(x_1^n)}{n} \right\} = \mu \left\{ x \in A^{typ} : \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} < H(\mu) \right\} = 0 \quad (3.19)$$

Consequently,

$$\liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \geq H(\mu), \quad \mu\text{-a.s.}$$

□

From lemma 1 and 2, it is trivial to show that $\lim_{n \rightarrow \infty} \frac{C(x_1^n)}{n}$ exists¹⁰ μ -almost surely and is then equal to $H(\mu)$, proving theorem 1.

□

¹⁰It is easy to construct a sequence with

$$\limsup_{n \rightarrow \infty} \frac{C(x_1^n)}{n} \neq \liminf_{n \rightarrow \infty} \frac{C(x_1^n)}{n}.$$

The limit exists at least on a set of measure one, namely the set of random sequences. See [52]

3.5. Complexity Distortion Function

In this section, we extend Kolmogorov Complexity Theory to the lossy case where distortion is allowed given rise to “Complexity Distortion Theory” (CDT). As mentioned before, Rate Distortion Theory (RDT) does not make any assumptions on the structure of the decoding function Ψ_n . The main difference between CDT and RDT is the restriction of Ψ_n to the set of partial recursive function. As a consequence, each function Ψ_n can be realized by a Turing machine. In this lossy setting, it is more natural to measure information with an extension of the Kolmogorov complexity to the semifair case. To introduce it, we contrast it with the RDF. Recall the sphere covering approach used in the definition of the operational rate distortion function in chapter 2. It was interpreted as the minimum number of balls of radius D needed to cover (almost surely) the space of all sequences of length n for very large values of n . The complexity distortion function can be introduced in the same lines. In a D -ball centered this time around x_1^n , let $\mathcal{Q}_D(x_1^n)$ be the sequence in \hat{A}^n with the smallest Kolmogorov complexity. If we have many sequences inside the ball, we pick the closest one to x_1^n , according to d_n . If we still have more than one candidate, we list them in a lexicographic order and arbitrarily pick the sequence with the smallest index in the list.

$$\mathcal{Q}_D(x_1^n) = \arg \min_{y_1^n \in \hat{A}^n : d_n(x_1^n, y_1^n) \leq D} C(y_1^n) \quad (3.20)$$

Definition 5 *The complexity distortion function is defined as:*

$$C_D(x) = \frac{C(\mathcal{Q}_D(x))}{l(x)} \quad (3.21)$$

There are two striking differences between the rate distortion function and the complexity distortion function:

1. The complexity distortion function is a deterministic quantity that does not rely on any probabilistic assumptions, in contrast with the rate distortion function relying on the probability measure associated to the source.
2. The complexity distortion function describes the best element inside a D -ball and does not ignore the information content. The rate distortion function does not describe the content of any elements of the D -ball. It simply indexes these balls following the general information theoretic belief that the actual meaning of messages is *irrelevant* to the communication problem [70]. Note that this indexing is done from the knowledge of source distribution at both end of the communication system. This knowledge describes the universe of all possible messages, a piece of information which is not required in complexity distortion theory.

This “sphere covering” approach emphasizes how CDT generalizes RDT. It has some strong connections with Vector Quantization (VQ) and sets the stage up for a new approach to universal coding that will be presented in the next chapter.

3.6. Equivalence with Rate Distortion Theory

This section extends theorem 1 to the lossy case and closes the circle of media representation techniques shown in figure 3-5, from traditional stochastic modeling approach shown on the left side of the picture, to modern deterministic modeling approach as show on the right side of the figure.

3.6.1 Extended Fundamental Theorem

The extended fundamental theorem is formulated as follows:

Theorem 2 For a stationary ergodic source (A, \mathcal{A}, μ) with a recursive probability measure μ ,

$$\lim_{n \rightarrow \infty} C_D(x_1^n) = R(D), \quad x_1^n \in A^n \quad (3.22)$$

μ -a.s.

Intuition behind Theorem 2

In the lossy case, the equivalence is obtained in the same manner as in the lossless case. A code with a recursive decoding function and with a coding rate close to the RDF is used. This code, taken from [73], also uses a two part structure and allows us to show that:

$$\limsup_{n \rightarrow \infty} C_D(x_1^n) \leq R(D), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.23)$$

The second part of the proof also uses a randomness test to show that it is not likely to have infinite sequences with a CDF strictly smaller than its RDF:

$$\liminf_{n \rightarrow \infty} C_D(x_1^n) \geq R(D), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.24)$$

The technical difficulties in this part of the proof arise when we attempt to link CDF's associated to individual sequences, with an average entity like the RDF. In the lossless case, this is done via the Shannon-McMillan-Breiman theorem. In the lossy case, this is done using theorem 3 [6] presented in chapter 2, section 2.3.2. This result was originally proved by Perez in [63].

In [98] a quantity similar to the CDF¹¹ is defined in another context, without practical considerations. In this paper, the authors did not make any computational restrictions on the source probability measure and claim that for a stationary ergodic

¹¹Note that the authors called this entity the D -distortion complexity function. We find the term "Complexity Distortion Function" more appropriate because of the existence in the literature [58] of a dual quantity that we would call the distortion complexity that is the equivalent to the distortion rate function in a complexity setting.

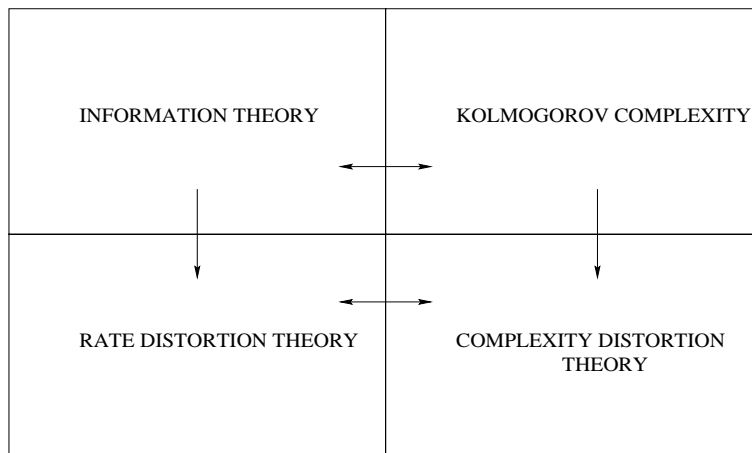


Figure 3-5: The circle of media representation theories (the bottom-right box as well as its associated arrows, establishing its relationship with other theories, are introduced in this thesis).

source, the rate distortion function and the complexity distortion function are almost surely equivalent. We present an alternative simpler proof of this result for recursive sources which is more constructive, and explains why the problem of coding finite objects, under computational resource bounds fits better in this algorithmic setting. In contrast with the proof proposed in [98] and [99] for the lossless case, the proof here does not use the variable length source coding theorem [70] nor its extension to the lossy case simply because these theorems are not constrained by any recursive assumptions¹². Furthermore, we highlight the separation principle between model and data [92] by using two-part codes. This approach provides very useful hints for the design of codecs that will be discussed in the next chapter.

3.6.2 Proof of Extended Fundamental Theorem

The proof is done in two parts. In the first part, we show the following lemma:

¹²These theorems cannot be used to compute lower bounds on Kolmogorov complexities because they predict existence of codes that may or may not be decodable by any recursive function if the source is governed by a probability law that may not be recursive, as assumed in [98, 99].

Lemma 3

$$\limsup_{n \rightarrow \infty} C_D(x_1^n) \leq R(D), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.25)$$

Proof of Lemma 3: To encode x_1^n efficiently, recall that a D -cover of a set $S \subset A^n$ is a collection $G \subset A^n$ such that every sequence in S is within D of some member of G . For a lossy representation, the sequence x_1^n can be represented by the following two part code proposed in [73]:

1. The index of the Markov k -type of x_1^n is transmitted.
2. Choose a D -cover $G_k(T_k(x_1^n))$ of least cardinality among all D -covers of $T_k(x_1^n)$.

The second part of the code is then an address of an element of $G_k(T_k(x_1^n))$ that is within D of x_1^n . Clearly, this address requires at most $\log_2 |G_k(T_k(x_1^n))|$.

We already mentioned in section 3.4.2 that the first part of the code requires only $m = o(n)$ if $k = \lfloor \frac{1}{2} \log_{|A_0|} n \rfloor$. It is shown in [73] that the ratio of the length of the second part of the code to n converges almost surely to the rate distortion function of the source.

Lemma 4

$$\lim_{n \rightarrow \infty} \frac{l(\Phi_n(x_1^n))}{n} = R(D) \quad a.s. \quad (3.26)$$

Φ_n being the encoding function corresponding to the code construction presented above.

See [73] for a proof of this result. This code is prefix free [73] and is a valid program for a prefix free Turing machine. The decoding operations are clearly recursive since the probability measure of the source is assumed recursive. As a result, we can construct a Turing Machine able to take this code as input and able to output

a sequence $y_1^n \in \hat{A}^n$ with $d_n(x_1^n, y_1^n) \leq D$. Consequently, we have shown that:

$$\limsup_{n \rightarrow \infty} C_D(x_1^n) \leq R(D), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.27)$$

□

In the second part of the proof of theorem 2, we show the following lemma:

Lemma 5

$$\liminf_{n \rightarrow \infty} C_D(x_1^n) \geq R(D), \quad x_1^n \in A^n \quad \mu\text{-a.s.} \quad (3.28)$$

Proof of lemma 5: To get this result, we start by designing a universal test for infinite sequences of the reproduction alphabet.

Lemma 6 *ν is recursive and the function*

$$\rho_1(\omega \mid \nu) = \sup_{\omega \in \Gamma_y} \{-C(y \mid \nu) - \log_2 \nu(y)\} \quad (3.29)$$

is a universal integral ν -test.

Proof of lemma 6: This result is similar to the test used in 3.4.2. The only thing that we have to show here is that ν is recursive. To do so, note that ν is a probability measure induced by the measurement $\mathcal{Q}_D(\cdot)$, from A^* to \hat{A}^* . Since μ is recursive by assumption and \mathcal{Q}_D is enumerable from theorem 4 in this chapter, then ν is enumerable and there exists a recursive function $g(k, y)$, $k \in \mathcal{N}$, $y \in \hat{A}^*$, non decreasing in k such that $\lim_{k \rightarrow \infty} g(k, y) = \nu(\Gamma_y)$. Since ν is a probability measure, we can compute an approximation ν^k of ν from below such that $\sum_{x:l(x)=l(y)} \nu^k(\Gamma_x) > 1 - \epsilon$, ϵ being arbitrary. This means that $|\nu(\Gamma_y) - \nu^k(\Gamma_y)| < \epsilon$, for all y and ν is recursive¹³. To show that ρ_1 is a universal test, we just have to invoke lemma 1,

¹³See example 4.3.2 in [52] for an identical argument. Note that the authors have a convention replacing cylinders by the actual sequences, as explain in section 4.2.

from appendix B..

□

We would like to show that $\liminf_{n \rightarrow \infty} C_D(x_1^n)$ is almost surely greater or equal to $R(D)$. To do so, we just have to prove the existence of a channel with conditional distribution belonging to $Q_n(D) = \{\eta(y_1^n | x_1^n) \mid E_p d_n(x_1^n, y_1^n) \leq D\}$ and with information rate, R , less or equal to $\liminf_{n \rightarrow \infty} C_D(x_1^n)$. We just have to show that at this rate R , we can achieve an average distortion less than or equal to D , the distortion constraint. Since the complexity distortion function is deterministic, we need a way to link infinite observations with information rates¹⁴. Consider the following deterministic channel mapping each $x_1^n \in A^n$ to $y_1^n \in \hat{A}^n$, such that $y_1^n = Q_D(x_1^n)$. Using theorem 2 and 3 of chapter 2, we would like to link its rate R with $\log_2 f^n(x_1^n, y_1^n)$ where $f^n(x_1^n, y_1^n) = \frac{dp^n(x_1^n, y_1^n)}{d\pi^n}$ and then link $\log_2 f^n(x_1^n, y_1^n)$ with $C_D(x_1^n)$. Unfortunately, the proposed channel is not memoryless. To make it memoryless we have to group source symbols into blocks of size n . We segment each source observation $x \in A$ into successive n -letter words \tilde{x}_i such that: $\tilde{x}_i = x_{nj+1}^{(n+1)j}$. Let $\tilde{x} = (\dots, \tilde{x}_{i-1}, \tilde{x}_i, \tilde{x}_{i+1}, \dots)$. Following the notation in [28, 6], the correspondence between x and \tilde{x} is denoted by $x \leftrightarrow \tilde{x}$ and for sets, $E \leftrightarrow \tilde{E}$ means that $E = \{x : x \leftrightarrow \tilde{x} \text{ for some } \tilde{x} \in \tilde{E}\}$. The σ -algebra \mathcal{A} and probability measure μ that define the source, have counterparts $\tilde{\mathcal{A}} = \{\tilde{E} : E \leftrightarrow \tilde{E} \text{ for some } E \in \mathcal{A}\}$ and $\tilde{\mu}(\tilde{E}) = \mu(E)$. The source $[\tilde{\mathcal{A}}, \tilde{\mu}]$ produces one so-called *super letter* every n time units. The shift transformation $\tilde{T} : \tilde{\mathcal{A}} \rightarrow \tilde{\mathcal{A}}$ is defined by $\tilde{T}\tilde{x} = \tilde{u}$ where $u = T^n x$ and $u \leftrightarrow \tilde{u}$. We also extend our channel and define $\tilde{Q}_D(\tilde{x}_i) = Q_D(x_{ni+1}^{(n+1)i})$. This channel is clearly memoryless on super letters but the source $[\tilde{\mathcal{A}}, \tilde{\mu}]$ may not be ergodic. Fortunately,

¹⁴This is exactly what we did in the lossless case when we invoked the Shannon-McMillan-Breiman theorem.

it can be decomposed into ergodic modes¹⁵ as follows: Define a non null invariant set $G \in \tilde{\mathcal{A}}$ as a set that both satisfies $\tilde{\mu}(G) > 0$ and $\tilde{T}(G) = G$. A non null invariant set that cannot be partitioned into two non-null invariant sets is called an ergodic mode.

Lemma 7 *Let $[A, \mu]$ be a time-discrete, stationary, ergodic source. Then the associated source $[\tilde{A}, \tilde{\mu}]$ of n -dimensional super letters can be decomposed into m ergodic modes, E'_0, \dots, E'_{m-1} , where m is a divisor of n . If $0 \leq j, \leq m - 1$ and $j \neq k$, then $\tilde{\mu}(E'_j) = \frac{1}{m}$ and $\tilde{\mu}(E'_j \cap E'_k) = 0$.*

See [6] for a proof of this result. In each of these modes, we compute the information rate that is achieved when the channel is deterministic and modeled by the function $\tilde{\mathcal{Q}}_D(\cdot)$. Consider the sources $[\tilde{A}, \tilde{\mu}_i]$, $i = 0, \dots, m - 1$, defined by

$$\tilde{\mu}_i(C') = m\tilde{\mu}(C' \cap E'_i), \quad C' \in \mathcal{A}'$$

where E'_i is the i th ergodic mode of $[\tilde{A}, \tilde{\mu}]$. It follows from lemma 7 that each of these sources are ergodic with respect to the super letter shift transformation \tilde{T} and theorem 3 can be used to get an expression for the information rate, \tilde{R}_i obtained by passing the output of these sources through the deterministic channel $\tilde{\mathcal{Q}}_D$. Let $f_i^t(\tilde{x}_1^t, \tilde{y}_1^t)$ be the Radon Nikodym derivative of \tilde{p}_i^t with respect to $\tilde{\pi}_i^t$. Then by theorem 2 and 3 in chapter 2,

$$\lim_{t \rightarrow \infty} \frac{\log_2 f_i^t(\tilde{x}_1^t, \tilde{y}_1^t)}{t} = \tilde{R}_i, \quad \tilde{p}_i\text{-a.s.} \quad (3.30)$$

¹⁵See [28] for a brief but excellent discussion on ergodic modes, in a context similar to the one that we have here.

From the definition of f_i^t , for any $s \in \mathcal{N}$, since \log_2 is continuous on \mathcal{R}^+ and right continuous at 0,

$$\log_2 f_i^s(\tilde{x}_1^s, \tilde{y}_1^s) = \log_2 \lim_{u \rightarrow \infty} \frac{p_i^s(F_u)}{\pi_i^s(F_u)} = \lim_{u \rightarrow \infty} (\log_2 p_i^s(F_u) - \log_2 \pi_i^s(F_u)) \quad (3.31)$$

where $\{F_u\}$ is a sequence of Borel sets converging regularly¹⁶ to $(\tilde{x}_1^s, \tilde{y}_1^s)$. Using De Possel's theorem [74] we can restrict the sequence $\{F_u\}$ to be nested: $F_{u+1} \subseteq F_u$, $u > 0$. Therefore, using the continuity from above property of finite measures [26],

$$\log_2 f_i^s(\tilde{x}_1^s, \tilde{y}_1^s) = \log_2 \tilde{p}_i^s(\tilde{x}_1^s, \tilde{y}_1^s) - \log_2 \tilde{\pi}_i^s(\tilde{x}_1^s, \tilde{y}_1^s), \tilde{p}_i^s\text{-a.s.},$$

and,

$$\log_2 f_i^s(\tilde{x}_1^s, \tilde{y}_1^s) = \log_2 \tilde{\mu}_i^s(\tilde{x}_1^s) + \log_2 \tilde{r}_i^s(\tilde{y}_1^s | \tilde{x}_1^s) - \log_2 \tilde{\mu}_i^s(\tilde{x}_1^s) - \log_2 \tilde{\nu}_i^s(\tilde{y}_1^s) \tilde{p}_i^s\text{-a.s.}$$

Since $\tilde{y}_1^s = \tilde{\mathcal{Q}}_D(\tilde{x}_1^s)$, $\log_2 \tilde{r}_i^s(\tilde{y}_1^s | \tilde{x}_1^s) = 0$ we obtain:

$$\log_2 f_i^s(\tilde{x}_1^s, \tilde{y}_1^s) = -\log_2 \tilde{\nu}_i^s(\tilde{y}_1^s) \tilde{p}_i^s\text{-a.s.} \quad (3.32)$$

$\tilde{\nu}_i$ is clearly recursive¹⁷. Recall lemma 6. For any \tilde{y} , if $\rho_1(\tilde{y} | \tilde{\nu}_i) < \infty$, then \tilde{y} is random with respect to $\tilde{\nu}_i$. Let $\epsilon_i = \rho_1(\tilde{y} | \tilde{\nu}_i)$. For any $s \in \mathcal{N}$,

$$\frac{-C(\tilde{y}_1^s | \tilde{\nu}_i) - \log_2 \tilde{\nu}_i^s(\tilde{y}_1^s)}{s} \leq \frac{\epsilon_i}{s}$$

¹⁶Please see [74].

¹⁷We proved that ν is recursive and know that for almost all y , $\nu(y)$ is either zero or equal to $\nu_i(y)$, if $y \in E_i$ or not. If $y \in E_i$, $\tilde{\nu}_i$ is equivalent to ν , a recursive set function.

Since $\tilde{\nu}_i$ is recursive, in the limit as $s \rightarrow \infty$, we get:

$$\liminf_{s \rightarrow \infty} \frac{-\log_2 \tilde{\nu}_i^s(\tilde{y}_1^s)}{s} \leq \liminf_{s \rightarrow \infty} \frac{C(\tilde{y}_1^s)}{s} \leq \liminf_{s \rightarrow \infty} \frac{C(y_1^{sn}) + \log_2 n + c}{s} \quad (3.33)$$

c being a constant. The last inequality holds because \tilde{y}_1^s can be computed from y_1^{sn} using a finite program of length less than $c + \log_2 n$, c being a constant. Since equation 3.33 holds for $\tilde{\nu}_i$ random sequences forming a set of $\tilde{\nu}_i$ -measure 1, the set

$$\left\{ y \in \hat{A}^\infty : \liminf_{s \rightarrow \infty} \frac{C(y_1^{sn})}{s} < -\liminf_{s \rightarrow \infty} \frac{\log_2 \tilde{\nu}_i^s(\tilde{y}_1^s)}{s} \right\}$$

is a null set according to both measures $\tilde{\nu}_i$ and \tilde{p}_i . Therefore, we conclude that:

$$-\lim_{s \rightarrow \infty} \frac{\log_2 \tilde{\nu}_i^s(\tilde{y}_1^s)}{s} \leq \liminf_{s \rightarrow \infty} \frac{C(y_1^{sn})}{s}, \tilde{p}_i\text{-a.s.} \quad (3.34)$$

Since $\tilde{y}_1^s = \tilde{Q}_D(\tilde{x}_1^s)$, $\tilde{p}_i^s(\tilde{x}_1^s, \tilde{y}_1^s) = \tilde{\mu}_i^s(\tilde{x}_1^s)$,

$$\lim_{s \rightarrow \infty} \frac{\log_2 f_i^s(\tilde{x}_1^s, \tilde{y}_1^s)}{s} \leq \liminf_{s \rightarrow \infty} n C_D(x_1^{sn}), \tilde{\mu}_i\text{-a.s.} \quad (3.35)$$

and from equation 3.30,

$$R_i = \lim_{t \rightarrow \infty} \frac{\log_2 f_i^t(\tilde{x}_1^t, \tilde{y}_1^t)}{nt} \leq \liminf_{t \rightarrow \infty} C_D(x_1^{nt}), \tilde{\mu}_i\text{-a.s.} \quad (3.36)$$

where $R_i = \frac{\tilde{R}_i}{n}$. From the subadditive ergodic theorem, it is easily verified that $\lim_{s \rightarrow \infty} C_D(x_1^s)$ is μ -almost surely a constant. Consequently, the R_i 's are also μ -almost surely upper bounded by this constant that we denote R . Note that R can be easily achieved by the source $[A, \mu]$ at distortion level D . To see this, note that $\tilde{R}_i \geq \tilde{R}_i^1(D)$, the first order rate distortion function for the source $[\tilde{A}, \tilde{\mu}_i]$, since at the rate \tilde{R}_i the distortion constraint D is respected from the definition of \tilde{Q}_D .

Hence, from theorem 5 chapter 2, for t sufficiently large, there exists a D -admissible code for $[\tilde{A}, \tilde{\mu}_i]$, of blocklength t in super letters and size

$$K_i \leq 2^{t(\tilde{R}_i^!(D)+\delta)} \leq 2^{t(\tilde{R}_{max}+\delta)} \quad (3.37)$$

where $\tilde{R}_{max} = \sup_i \tilde{R}_i$. Consider the following code obtained by combining the different m ergodic modes. Each ergodic mode can be encoded at the rate \tilde{R}_i by using $\mathcal{Q}_D(\cdot)$ yielding a code denoted B_i ¹⁸. These B_i 's can be combined into a “giant” code, B , that is D -admissible and with rate less or equal to R by taking their union over m . Clearly, the rate of B , denoted R_B , is upper bounded by:

$$R_B = \frac{\log_2 |B|}{tn} \leq \frac{\log_2(m2^{t(\tilde{R}_{max}+\delta)})}{tn} = \frac{\log_2 m + t\tilde{R}_{max} + t\delta}{tn} \quad (3.38)$$

for t large enough. Therefore since δ is arbitrary, and since $\frac{\tilde{R}_{max}}{n} = \sup_i R_i \leq R$ μ -a.s., by letting $n \rightarrow \infty$, we have shown that $R_B \leq R$, almost surely. Furthermore, since R_B is achievable with an average distortion less than D , so is R , we conclude that $R \geq R(D)$ and also that $\liminf_{m \rightarrow \infty} C_D(x_1^m) \geq R(D)$, μ -almost surely, .

□

The proof of theorem 2 follows from lemma 3 and lemma 5.

□

¹⁸This code is suggested in [6, 28] but not used in the proof of the fundamental theorem for stationary ergodic source. The reason is that the R_i used in this theorem are not constant. Therefore, a simple union of the codes B_i described below would yield upper bounds for the rate of the system and not an average rate that converges to the rate distortion function. In our case, we are not concern with the rate distortion function. We are just looking for an achievable code with rate less or equal to the complexity distortion function. It turns out that this code has a maximum rate below the complexity distortion function in each of the ergodic modes and this statement is strong enough to provide the inequality that we want.

3.6.3 Some Remarks

We end this section with two general remarks for the coding of finite objects.

Remark 1 *In the lossless case, if we remove the assumptions that μ is stationary and ergodic and just assume that μ is recursive, randomness tests can be used to make precise statements on the difference between the Shannon-Fano code length and the Kolmogorov complexity. To see this, let's compute the measure of the following set:*

$$S_n = \{x : f(x) \geq m, l(x) = n\} \quad (3.39)$$

where,

$$f(x) = -\log_2 \mu(x) - C(x | l(x)) \quad (3.40)$$

To do so, note that

$$-\log_2 \mu(x) - C(x | l(x)) \geq m$$

is equivalent to say that

$$2^{-C(x|l(x))} \geq 2^{m+\log_2 \mu(x)}$$

Since programs are assumed to be prefix free, they satisfy the Kraft inequality. By taking the summation on all $x \in S_n$ on both side of the inequality we get:

$$1 \geq \sum_{x \in S_n} 2^{-C(x|l(x))} \geq 2^m \sum_{x \in S_n} \mu(x)$$

Therefore, $\mu(S_n) \leq 2^{-m}$. This clearly shows that it is “exponentially hard” to compress below the Shannon-Fano code length even for finite sequences, if we just make the reasonable assumption that μ is recursive. This result is known as Barron's lemma and is a requirement for all randomness tests for finite sequences as stated in definition 1, appendix B.. In fact, it can be shown that $f(x)$ is a randomness test.

Remark 2 *The proposed proofs use two-part code. The first part of the code (the index of the Markov type) represents the model part. It consists of the regular part of the source observation. The second part of the code (the type index of the sequence) represents the data part and the tests proposed show that this part is random according to the source distribution. Consequently, it is enough to use typical ranking encoding techniques similar to any well known entropy coding method for the representation. The first part of the code vanishes in the rate of the code but only for infinite observations. For a finite object, this part is relevant and contains meaningful information on the structure and semantics of the source observation according to the assumed decoding language. In general, there are many ways to make the division into meaningful information and remaining random information [92]. The proposed division that uses Markov types may not be optimal for finite objects. It might be difficult to estimate an unknown source distribution from a finite observation. In the previous remark, we argue that for finite objects, the Shannon-Fano code length based on the true distribution is close to the Kolmogorov complexity with a high probability but with an empirical estimation of the source distribution, we cannot guarantee this fact unless we have access to a very long source observation. This observation is generally not available in practice, when dealing with finite individual objects.*

3.7. Conclusion

The main contribution of this chapter is the extension of the notion of Kolmogorov complexity to lossy descriptions of information using the concept of complexity distortion function. We have shown that the complexity distortion function is almost surely equal to the rate distortion function for infinite observations of stationary ergodic sources. The proposed proof revolves around the main result by Martin-Löf

stating the set of infinite random sequences has measure 1. Any property of this set, like incompressibility, holds almost surely. Also, with the type approach, we underline the separation principle between model and data. The model part can be identified as the routine used to describe the computable probability distribution. The data part is essentially the index i with maximum value corresponding to the cardinal of the type class. From a more practical point of view, there are two more interesting points to make from this proof. First, restricting the decoding function to be recursive does not reduce the performances if the source has a recursive probability measure. In fact the Church-Turing thesis guarantees that any coding algorithm belongs to the set of recursive functions, from traditional entropy coding techniques to model-based coding methods. The result is a unification of all coding algorithms under a single framework. Second, the equivalence was made possible by using limits showing that Shannon's information measure assumes the availability of an infinite amount of computational resources at the decoding end. Furthermore, due to its ensemble nature, this information measure requires infinite observations to make any accurate predictions on the performances of the communication system via stationary and ergodic assumptions. As argued in [42], since practical processes do not go on forever, identifying such processes with an infinite sequence of outputs "is a metaphysical conceit that provides no physical interpretation of probability" and information. Its dual part, the Kolmogorov complexity, is not suitable for the prediction of compression rates when computational resources are unbounded as shown in theorem 3 (where Shannon's approach works well) but in contrast with Shannon's measure, it predicts recursive compression rates for the coding of finite individual objects with a finite amount of computational resources.

Chapter 4

Resource Bounds in Media Representation

4.1. Introduction

Current media representation theories take into account information rate and distortion to measure the efficiency of coding procedures. Rate Distortion Theory defines the infimum information rate that can be achieved under a distortion constraint between the source and reproduction objects. Equivalently, CDT defines a tradeoff between information rate and distortion by defining the lowest achievable rate under a distortion constraint, in a programmable setting where the decoder is a UTM. In practice, there is another major group of constraints in source coding that is absent from these settings and that corresponds to situations where the decoding device has limited computational resources. The cost of a decoding device is related to its computational power. With more power, better performances can be achieved. With the proliferation of general purpose digital signal processors, there is a need to understand the tradeoffs between information rate, distortion and computational complexity for a better optimization of the system operations. Recently, Gormish in [33] and later Goyal in [34] stated the importance of the tradeoff

between information rate, distortion and computational complexity by introducing computational resource bounds at the encoding end with applications to transform coding [35], [36] in the lines of the work of Lengwehasatit and Ortega[49]. Assuming that the transform coder in most still image compression system has a major impact on computational complexity (time only), they measured tradeoffs between information rate, computation and distortion at the encoding end. Goyal exploited typical separable fast discrete cosine transform algorithms to find smart ways to quantize coefficients to zero and reduce the number of operations (multiplications and additions). The main idea is to get a tradeoff between distortion and computational complexity using the information rate as a parameter. From these works, interesting questions arises. For instance, it is well known that the Karhunen-Loeve transform (KLT) yields better rate distortion performances than the discrete cosine transform (DCT). If so, why are we always using the DCT in practice ? Can we have a theoretical model that would reflect directly our practical preferences ?

With the emergence of media processors, computational complexity issues arise also at the decoding end. The universal character of the UTM introduces complex optimization and computational resource allocation problems at the operating system level of the decoder. To address these issues, it is important to understand the tradeoff between rate, distortion and computational complexity.

Motivated by these observations and questions, we add in this chapter a new dimension to the classical rate distortion setting to take into account the power of the decoding device, in contrast with the work of Gormish and Goyal that considers encoding resource bounds. Clearly, this addition requires precise measures for computational power and these measures cannot be defined without assuming a particular model of computation at the decoder. Since Shannon's setting does not make any assumptions at the decoding end, it is difficult to investigate these

tradeoffs in IT. Kolmogorov's setting does not suffer from this drawback. The decoder is a UTM which provides the natural environment for the formalization of computational complexity measures. It also allows us to address the problem of representing information with only a finite amount of computational resources and to have a more realistic approach to measure information. Intuitively, two objects x and y with the same Kolmogorov complexity may not contain the same amount of information if the computational complexities of their short description are different. If more computational effort is required to generate x , it is reasonable to think of x as being more complex than y and to contain more information. Kolmogorov complexity and computational complexity have been combined under these lines and resource bounded version of the Kolmogorov complexity were introduced by Daley, Levin and Adelman¹ in the 70's. Since then, the merger of these fields has received a lot of attention in computational complexity theory. But it did not have the same impact in source coding theory and this is probably due to the non recursive nature of the plain unconstrained Kolmogorov complexity.

In this chapter, we define the tradeoff between information rate, distortion and computational complexity. The main question that we address here is if there is a encoding procedure able to respect distortion and computational complexity bounds imposed at the decoder. We start in the next section by extending the CDF before answering this question in section 4.3. where we propose a generic method to find programmatic representation of sequences respecting the decoding constraints. We then apply this methodology to still image data.

¹See [52] chap 7 for more historical notes.

4.2. Resource Bounded Complexity Distortion Function

The resource bounded complexity distortion function is a natural extension of the complexity distortion function that takes into account the limited computational capabilities of the decoder. To define it, we need precise measures of computational complexity. In practice, typical measures involve the number of multiplication and addition that can be processed by the decoding device and/or the amount of memory required. In this work, we abstract ourselves from these measures and distinguish two types of computational complexity measures: time, related to the number of operations, and space, related to memory sizes. Define $t(n)$ and $s(n)$ to be total recursive non decreasing integer functions. Let $n = l(x)$ and let T_Φ be a multitape Turing machine which computes the function Φ . If $T_\Phi(y) = x$ in less than $nt(n)$ steps and less than $ns(n)$ tape cells, we write $\Phi^{t,s}(y) = x$. In general, the superscripts t and s refer to functions of the input size $n = l(x)$. The definition can also be read with t and s corresponding to integer values bounding the amount of computational resources at the decoder. This is precisely the approach that we use here. Thus, t and s represent sharp computational bounds. We reserve the symbols $t_{\Phi,\Psi}(\cdot)$ and $s_{\Phi,\Psi}(\cdot)$ to functions mapping sequences x_1^n respectively to the number of steps Ψ need to reproduce x_1^n from its short description generated by Φ and the number of memory cells Ψ needs to reproduce x_1^n from its short description. When it is clear from the context, we drop the subscripts Φ and Ψ on $t_{\Phi,\Psi}(\cdot)$ and $s_{\Phi,\Psi}(\cdot)$. Assume as before that μ is a recursive probability measure. In RDT, the average distortion for the conditional distribution η (which corresponds to an encoding/decoding procedure

in IT^2) was defined in chapter 2

$$D(\eta) = E_p[d_n(x_1^n, \Psi_n(\Phi_n(x_1^n)))] = \int d_n(x_1^n, \Psi_n(\Phi_n(x_1^n)))dp$$

Similarly, let $T_{\Phi, \Psi}(x_1^n) = \frac{t(x_1^n)}{n}$ and $S_{\Phi, \Psi}(x_1^n) = \frac{s(x_1^n)}{n}$ be respectively the time and space complexity of the decoding of x_1^n on Ψ . We introduce the average complexity for Φ, Ψ as being:

$$T(\eta) = E_p[T_{\Phi, \Psi}(x_1^n)] = \int T_{\Phi, \Psi}(x_1^n)dp \quad (4.1)$$

$$S(\eta) = E_p[S_{\Phi, \Psi}(x_1^n)] = \int S_{\Phi, \Psi}(x_1^n)dp \quad (4.2)$$

The main problem raised in this section is to understand the tradeoffs between Kolmogorov complexity (or information rate), time complexity, space complexity and distortion. Following the RDT setting, a rate distortion time-complexity, space-complexity quadruplet (R, D, t, s) is said to be achievable if there exists an encoding function Φ and a decoding function Ψ such that:

$$\begin{aligned} T(\eta) &= E_p[T_{\Phi, \Psi}(x_1^n)] = \int T_{\Phi, \Psi}(x_1^n)dp \leq t, \\ S(\eta) &= E_p[S_{\Phi, \Psi}(x_1^n)] = \int S_{\Phi, \Psi}(x_1^n)dp \leq s, \\ D(\eta) &= E_p[d_n(x_1^n, \Psi(\Phi(x_1^n)))] = \int d_n(x_1^n, \Psi(\Phi(x_1^n)))dp \leq D \end{aligned} \quad (4.3)$$

as $n \rightarrow \infty$. With distortion and decoding resource bounds, information is measured with the operational resource bounded complexity distortion function (RBCDF) defined as follows:

$$C_D^{t,s}(x) = \frac{C_D^{t,s}(\mathcal{Q}_D^{t,s}(x))}{l(x)} \quad (4.4)$$

²Note in this section that the encoding procedure could be random. In this case, it could be represented by the distribution η , like in source coding theory. For deterministic encoding procedures, η is always 1.

where

$$Q_D^{t,s}(x_1^n) = \arg \min_{y_1^n \in \hat{A}^n: d_n(x_1^n, y_1^n) \leq D} C^{t,s}(y_1^n)$$

It defines the minimum compression rate that can be achieved in a programmable system with distortion and computational bounds. Let

$$\bar{C}_D^{t,s} = \liminf_{n \rightarrow \infty} \int C_D^{t,s}(x_1^n) dp \quad (4.5)$$

From the definition of $C_D^{t,s}(\cdot)$, it is straight forward to show that this function delimits the space of all rate distortion complexity quadruplets into an achievable and non achievable region. The “hyper” surface of all the points of $C_D^{t,s}(\cdot)$ belongs to the achievable subspace since there exists recursive coding/decoding pairs operating at these points. In the second part of this chapter we will propose efficient algorithmic procedures for the evaluation of this surface. Its points can be recursively obtained (recall theorem 4 in 3). As $t, s \rightarrow \infty$, the halting problem gets in our way and it becomes harder to evaluate the surface, even when n is finite. $\bar{C}_D^t = \lim_{s \rightarrow \infty} \bar{C}_D^{t,s}$ is represented in figure 4-1. This surface assumes an infinite amount of decoding computational space.

Remark 1 *It is important to remark here that introducing resource bounds on a UTM limits significantly its computational power. In fact, such a UTM is formally an FSM. The key here is that a TM may have infinite tapes but at each time of the computation, only a finite amount of memory and time is used. The amount of computational resources is unbounded but finite at any time. In fact any real computer has only a finite amount of computational power. Indeed, as mentioned in [94], “we can even go so far as to compute a finite bound on the maximum amount of memory constructible out of the material composing the known universe and be tempted to claim that, for all practical purposes, FSM’s serve as models of effective procedures.*

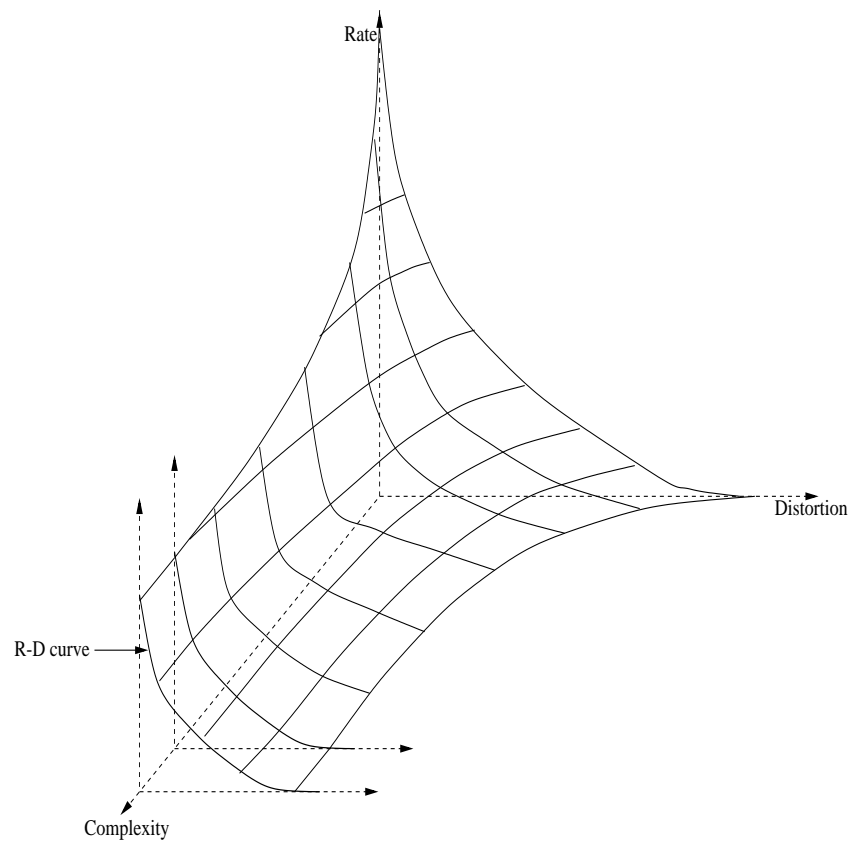


Figure 4-1: Rate-Distortion-Complexity tradeoff.

Unfortunately, the finiteness of an FSM is not a mathematically useful concept. The finiteness constraints can often get in the way of a concise, understandable description of an algorithm. We often, for example, write programs in which we assume that all our intermediate results will fit in their respective variables. Even though the assumption may not always be strictly justified, by making it, we greatly reduce the amount of detail we have to handle and this is certainly desirable". From a pure compression point of view, the FSM model is sufficient, under stationary and ergodic assumptions. In this case, traditional compression algorithms (like Shannon-Fano, Huffman, or even universal extensions of arithmetic coding and Lempel-Ziv coding) yields asymptotically optimal representations. This fact is clearly shown in [70] where encoders and decoders are modeled by transducers which are FSM's with a finite amount of memory. But when the understanding problem is considered, the content and the clarity of the descriptions become important factors that forces us to extend the FSM model to the TM model with infinite memory, as we commonly do with real computers which are physically transducers with a large but finite amount of memory and time.

The deterministic framework provided by the Kolmogorov complexity allows the definition of the RBCDF but it does not have the mathematical elegance of Shannon's probabilistic setting. Indeed, it is quite difficult to make precise statements on the properties of this complex rate-distortion-complexity relationship with the tools that we have developed so far simply because we do not rely on a strong and well known mathematical framework like probability theory. The most important properties of the classical rate distortion function were derived from the connections made with the mutual information. Similarly, we established connections with the probabilistic setting and connect the RBCDF and mutual information by extending the equivalences proposed in the previous chapter. The result of this is the following

theorem.

Theorem 1 *Let $[A, \mu]$ be a recursive stationary ergodic source. Let*

$$P_D^{t,s} = \left\{ \eta \mid \nu \in P_n^{t,s}, \int d_n(x_1^n, y_1^n) dp \leq D \right\} \quad (4.6)$$

where $P_n^{t,s}$ is the class of probability measures for objects of length n , with cumulative densities arithmetically computable in less than t steps and using less than s memory cells. Define

$$R_n(D, t, s) = \min_{P_D^{t,s}} \frac{I_n}{n} \quad (4.7)$$

where I_n is the mutual information as it is defined in chapter 2 and

$$R(D, t, s) = \lim_{n \rightarrow \infty} R_n(D, t, s) \quad (4.8)$$

then,

$$\lim_{n \rightarrow \infty} \bar{C}_D^{t,s} = R(D, t, s) \quad (4.9)$$

for all $t \geq T_{min}$ and $s \geq S_{min}$, where

$$T_{min} = \inf \left\{ t \mid P_D^{t,s} \neq \phi \right\},$$

$$S_{min} = \inf \left\{ s \mid P_D^{t,s} \neq \phi \right\}$$

ϕ representing here the empty set.

Before we go on with the proof, note that T_{min} and S_{min} are well defined simply because of the existence of recursive coding procedures operating at any level D . Also, by arithmetically computable in less than t steps and using less than s memory cells, we simply mean that the cumulative density can be encoded by an arithmetic coder using less than t steps and less than s memory cells.

Proof: This theorem is proved by using equivalences between information theory and complexity distortion theory presented in chapter 3. Stationarity is assumed here to make sure that the limit $\lim_{n \rightarrow \infty} \frac{I_n}{n}$ exists. To find this equivalent form, recall the definition of $P_D^{t,s}$:

$$P_D^{t,s} = \left\{ \eta \mid \nu \in P_n^{t,s}, \int d_n(x_1^n, y_1^n) dp \leq D \right\} \quad (4.10)$$

where $P_n^{t,s}$ is the class of probability measures for objects of length n , with cumulative densities arithmetically computable in less than t steps and using less than s memory cells. Clearly, all probability measures belonging to $P_n^{t,s}$ are recursive. Also recall that:

$$R_n(D, t, s) = \min_{P_D^{t,s}} \frac{I_n}{n} \quad (4.11)$$

Note that from the equivalence between CDT and RDT, for a stationary ergodic source,

$$\lim_{n,t,s \rightarrow \infty} R_n(D, t, s) = \lim_{n,t,s \rightarrow \infty} \bar{C}_D^{t,s}(x_1^n) = \lim_{n \rightarrow \infty} C_D(x_1^n) \mu\text{-a.s.} \quad (4.12)$$

We show that this equivalence still holds without the limits on computational resource bounds. To see this, consider the optimal conditional probability distribution $\hat{\eta} \in P_n^{t,s}$ that minimizes I_n . Let $\hat{\nu}$ be the induced probability distribution on the reproduction space. It is easy to realize that

$$C^{t,s}(y_1^n) \leq -\log \hat{\nu}(y_1^n) + c_{\hat{\nu}} \quad (4.13)$$

since $\hat{\nu}$ is recursive and from a program computing $\hat{\nu}$, we can represent y_1^n with less than t steps and less than s memory cells using the procedure used in arithmetic coding. $c_{\hat{\nu}}$ is just a constant independent of y_1^n . Furthermore, from theorem 4,

chapter 3,

$$C(y_1^n) \leq C^{t,s}(y_1^n) \leq -\log \hat{\nu}(y_1^n) + c_{\hat{\nu}} \quad (4.14)$$

From theorem 8.1.1 in [52],

$$0 \leq \left(\int C(y_1^n) d\hat{\nu} - H(\hat{\nu}) \right) \leq c_{\hat{\nu}} \quad (4.15)$$

where $c_{\hat{\nu}}$ is again a constant that does not depend on y_1^n . As a consequence, if we take expectations on equation 4.13, and we combine this with equation 4.15, we obtain:

$$H(\hat{\nu}) \leq \int C(y_1^n) d\hat{\nu} \leq \int C^{t,s}(y_1^n) d\hat{\nu} \leq H(\hat{\nu}) + \max\{c_{\hat{\nu}}, c_{\hat{\nu}}'\} \quad (4.16)$$

As n grows to infinity and dividing everything by n , we get:

$$\lim_{n \rightarrow \infty} \frac{H(\hat{\nu})}{n} \leq \liminf_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} \leq \limsup_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} \leq \lim_{n \rightarrow \infty} \frac{H(\hat{\nu}) + \max\{c_{\hat{\nu}}, c_{\hat{\nu}}'\}}{n} \quad (4.17)$$

Since the entropy rate converges for stationary sources, $\lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu}$ exists also and we conclude that

$$\lim_{n \rightarrow \infty} \frac{H(\hat{\nu})}{n} = \lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} \quad (4.18)$$

Using randomness tests, we have also shown in the previous chapter that

$$\lim_{n \rightarrow \infty} \frac{-\log \hat{\nu}(y_1^n)}{n} = \lim_{n \rightarrow \infty} \frac{C(y_1^n)}{n} \text{ almost surely}$$

Since we assume $\hat{\nu} \in P_n^{t,s}$, it follows that

$$\lim_{n \rightarrow \infty} \frac{-\log \hat{\nu}(y_1^n)}{n} = \lim_{n \rightarrow \infty} \frac{C^{t,s}(y_1^n)}{n} \text{ almost surely}$$

Hence,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \mathcal{Q}_D^{t,s}(x_1^n) = \lim_{n \rightarrow \infty} \arg \min_{d_n(x_1^n, y_1^n) \leq D} C^{t,s}(y_1^n) = \lim_{n \rightarrow \infty} \arg \min_{d_n(x_1^n, y_1^n) \leq D} -\log \hat{\nu}(y_1^n) \text{ almost surely}$$

Clearly, $\mathcal{Q}_D^{t,s}$ defines then a channel mapping that minimizes $R(D, t, s)$ almost surely and corresponds to $\hat{\eta}$. And using this deterministic channel mapping, $\lim_{n \rightarrow \infty} \frac{H(\hat{\nu})}{n} = I_n = \lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu}$ and we get:

$$R(D, t, s) = \lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} \quad (4.19)$$

It remains to show that $\lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} = \lim_{n \rightarrow \infty} \bar{C}_D^{t,s}$. This is trivial since $y_1^n = \mathcal{Q}_D^{t,s}(x_1^n)$ and $\hat{\nu}(y_1^n) = \sum_{\mathcal{Q}_D^{t,s^{-1}}(y_1^n)} \mu x_1^n$, where $\mathcal{Q}_D^{t,s^{-1}}(y_1^n)$ represents the set of all sequences x_1^n with image y_1^n using $\mathcal{Q}_D^{t,s}(\cdot)$. Hence,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{y_1^n} \sum_{x_1^n \in \mathcal{Q}_D^{t,s^{-1}}(y_1^n)} = \lim_{n \rightarrow \infty} \frac{1}{n} \int C^{t,s}(y_1^n) d\hat{\nu}$$

and we have shown that

$$\lim_{n \rightarrow \infty} \bar{C}_D^{t,s} = R(D, t, s) \quad (4.20)$$

□

From this result, the most important property of this surface, convexity, can be derived as stated in the next theorem:

Theorem 2 *The average resource bounded complexity distortion function is convex in the region where $t \geq T_{min}$ and $s \geq S_{min}$, where*

$$T_{min} = \inf \{t \mid P_D^{t,s} \neq \phi\},$$

and,

$$S_{min} = \inf \{s \mid P_D^{t,s} \neq \phi\}$$

Proof:

The convexity property is derived using theorem 1 and the properties of the mutual information. First, note that the average distortion and the average complexity are linear functions of the conditional distribution . To see this, choose two distinct conditional distributions q_1 and q_2 operating at two distinct points on the average resource bounded complexity distortion surface. At these points, let R_1 and R_2 be the corresponding information rates. Let (Φ_1, Ψ_1) and (Φ_2, Ψ_2) be the corresponding encoder/decoder pairs inducing q_1 and q_2 . Let $0 \leq \lambda \leq 1$, then

$$D(\lambda q_1 + (1 - \lambda)q_2) = \lambda D(q_1) + (1 - \lambda)D(q_2)$$

and, if $(\Phi_\lambda, \Psi_\lambda)$ is defined as the encoding/decoding pair inducing the conditional distribution $\lambda q_1 + (1 - \lambda)q_2$, then it is easy to realize that $(\Phi_\lambda, \Psi_\lambda)$ operates at a distortion level equal to $D(\lambda q_1 + (1 - \lambda)q_2)$ and with average time and space complexity given by:

$$T(\lambda q_1 + (1 - \lambda)q_2) = \lambda T(q_1) + (1 - \lambda)T(q_2)$$

and

$$S(\lambda q_1 + (1 - \lambda)q_2) = \lambda S(q_1) + (1 - \lambda)S(q_2)$$

From equation 4.20 the average resource bounded complexity distortion function can be expressed as a mutual information which is a convex function of the conditional distribution [15]. Denote R_λ the information rate at $\lambda q_1 + (1 - \lambda)q_2$. Then, $R_\lambda \leq \lambda R_1 + (1 - \lambda)R_2$, proving theorem 2.

□

Convexity is a key property to understand. It justifies the usage of standard optimization tools in order to design efficient computational resource management algorithms in such programmable environments. Note that the theorem holds with a limit and also with stationary assumptions on the source. In general, convexity is not true without this limit simply because the set of non random objects have a non zero measure without this limit. But in general, Barron's lemma (that has been further extended to the lossy case in [46]) provides an intuitive argument that would guarantee convexity most of the time. Precise statements on this point still remain to be established.

4.3. Universal Coding of Finite Objects with Distortion and Computational Constraints

The convergence between complexity distortion function and rate distortion function is due to the existence of types or relative frequencies. Classical coding techniques use this property and are only asymptotically optimal for stationary ergodic sources. The question that we address in this section is the design of efficient codec systems for finite objects with a limited amount of computational resources. The goal is to find a representation with performances as close as possible to the complexity distortion function. To do so, we adopt a novel approach to universality that we present in section 4.3.1. The problem becomes a mixture of Solomonoff's time (resource) limited optimization problem and McCarthy's inversion problem. In the time limited optimization problem, we have a machine M , whose inputs are finite strings and whose outputs are numbers. We are given a fixed time T^3 . The problem

³More generally we are given a fixed amount of computational resources.

is to find within time T an input string, s , such that $M(s)$ is as large as possible. In McCarthy's problem [55], we are given a partial function $f_m(n)$ computed by the m^{th} TM⁴ and we would like to design a TM which, when confronted by the number pair (m, r) , computes as efficiently as possible a function $g(m, r)$ such that $f_m(g(m, r)) = r$. Here, efficiency hints towards computational resource bounds at the encoding end. Clearly, g is not recursive, but following a construction identical to the one used in the proof of theorem 4, chapter 3, we can approximate g . Unfortunately, this procedure is highly inefficient and not practical. Levin in [51] approached this problem and proposed the LSEARCH (as it is called by Solomonoff [77]) algorithm⁵. This algorithm has been used and extended by Solomonoff and Levin to include time limited optimization problems. "The most general kind of induction can be formulated as the problem of finding short descriptions of data, which is a time limited optimization problem, and therefore amenable to LSEARCH" [77]. To tackle problems efficiently, heuristics were added in the search to reduce the size of the search space. The probabilistic analog of a heuristic is a function that assigns low probabilities to areas of the search space that would be cut out by the heuristic. In the machine learning theory this is called the "inductive bias". Furthermore it would be necessary to have to machine modify the probability distribution over the search space as a result of its own experience in problem solving. Evolutionary search techniques provide us a natural way to achieve these goals. In section 4.3.3, we focus on Genetic Programming. Following the novel approach to universal coding (section 4.3.1), we proposed a coding system for finite objects yielding performances arbitrarily close to the resource bounded complexity distortion function. After a

⁴ m is just the Gödel number or the index of this TM in a standard enumeration of TM's. See [52] for a discussion on standard enumeration of TM's.

⁵See [51], [52] chap 7, pp 503-505 for a good presentation of LSEARCH. A brief summary of its extensions can be found in [77].

presentation of the decoding end of the communication system, we present the encoder and provide an analysis of the time complexity of the proposed algorithm.

4.3.1 Universal Coding Revisited

Current universal coding techniques approach optimality asymptotically, as the length of the observation increases to infinity. By definition, a codec system is universal if its coding rate approaches asymptotically the rate distortion function. With stationary and ergodic assumptions, such systems have been designed and categorized in three classes as discussed in chapter 2. Efforts to break the barrier imposed by these non realistic statistical assumptions are presented in [22, 13], with optimality still achieved asymptotically. Universal coding of finite objects does not fit naturally in this framework and in this section we address this problem by showing how deterministic source models can also be used to model efficiently finite objects. We believe that this extension is necessary in image processing simply because most natural images do not appear random or patternless to us because of the structure and limiting capabilities of the HVS. Consequently, probabilistic models should not be forced on such data just for tractability reasons. Deterministic models should also be considered and the actual information content, which is irrelevant for infinite objects, should not be ignored. Under these lines, we formalize the universal coding problem for finite objects and add a fourth class into the taxonomy of lossy universal coding techniques. For practical consideration, we restrict our attention to recursive decoding functions and also consider decoding resource bounds in the setting.

Definition 1 *Let the probability of error $P_e^{(n)}$ be defined as follow:*

$$P_e^{(n,t_e,s_e)} = \mu\{x_1^n \mid d_n(x_1^n, \Phi_n^{t_d,s_d}(\Psi_n^{t_e,s_e}(x_1^n))) \geq D\} \quad (4.21)$$

We say that a rate R block code corresponding to the encoding/decoding pair (Φ_n, Ψ_n) is algorithmically universal if the functions Φ_n and Ψ_n are recursive and if

$$\lim_{s_e, t_e \rightarrow \infty} P_e^{(n, t_e, s_e)} = 0 \quad (4.22)$$

whenever $R \geq C_D^{t_d, s_d}(x_1^n)$

Theorem 1 For a fixed computational constraint (t_d, s_d) and distortion constraint D , algorithmically universal codes exist.

Proof:

With computational resource bounds at the decoding end, recursive encoders operating arbitrarily close to $C_D^{t_d, s_d}(x_1^n)$ do exist, proving the theorem.

□

In this setting, the decoder being a universal Turing machine, we also have a two stage coding process, the first one describing an algorithm (or a TM) and the second one describing the data for this algorithm (the program for that TM). The universality of this machine is precisely what we try to exploit to extend the classical notion of universal coding and include all coding techniques, from traditional approach like Huffman, Arithmetic or even Lempel-Ziv techniques to modern approach like model-based and fractal-based coding techniques. In fact, with the Church-Turing thesis, we consider all representation techniques from an algorithmic angle and unify them under the same framework. The general structure of this encoder is shown in figure 4-2. With such a universal decoder, what we are really doing here is allowing more complex languages than the one used in traditional VQ systems, at the decoding end. The design of such VQ decoders is a simple language design problem. The “grammar” of such languages is very simple and it allows only one instruction, the READ instruction that is used to access different locations of the codebook. The

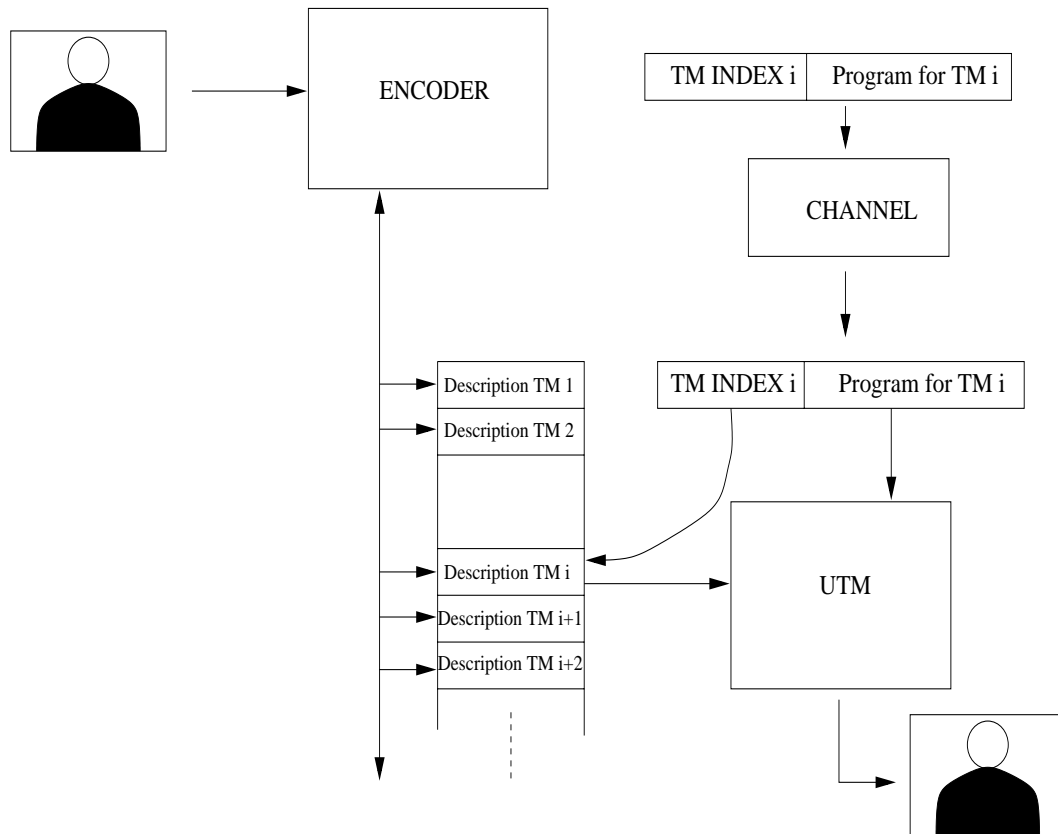


Figure 4-2: Algorithmically universal coding system. This system generalizes the first three universal coding systems presented in chapter 2, section 2.4.

“vocabulary” of the language is contained in these codebooks. With a universal Turing machine at the decoding end, we are allowing more complex grammars to be used to represent the source data. Ideally, we would like to have a high level language at the decoder that would help us to extend the traditional pixel oriented way of representing visual information and reduce the gap between high level and low level semantics of the source data. This language design step is equivalent to the probability estimation step in classical IT. In practice, this design should depend on the prior knowledge on the application at hand. This situation is similar to the one we have to face when we design software systems. We would rather use LISP than C for artificial intelligence applications, although both languages are Turing complete. Similarly, there is an urgent need for a Turing complete language for the representation of images. This problem is outside the scope of this thesis but it is important to remember that good language design choices limit the software overhead imposed by the first stage of the representation. In theory, this model overhead becomes negligible for infinite sequences but in practice, it must be taken into account. Such a Turing complete language is used in MPEG-4 Audio. It is called Structured Audio Orchestra Language [21].

4.3.2 The Decoder

The key component of the complexity distortion approach is the substitution of the decoder in Shannon’s classical communication system by a Turing machine. To simplify the discussion, we assume that our decoding TM, Ψ_n is a multitape TM with a one way read only input tape called the program tape. Our input alphabet contains four symbols: $B = \{0, 1, \beta, \phi\}$. β is the typical blank symbol. We add an extra symbol ϕ that will be used as a special character representing the “no operation” instruction. This will greatly simplify the theoretical analysis of the convergence of

the algorithms. When the decoder scans this symbol on its program tape, it stays in the same state and simply moves the program head one cell to the right. Its meaning is equivalent to the use of introns and selector operators in biology and in [53]. TM programs are isomorphically equivalent to LISP statements or symbolic expressions. In fact compilers use this fact to translate any piece of code into their equivalent abstract syntax tree representation. Introns⁶ refer to parts of a program that do not affect its computation. Selector operators are simple function nodes in abstract syntax trees that selects one and only one outgoing edge for traversal. This way they link potential interesting parts of the programs in a non ambiguous way, just like the ϕ operation does on TM program strings. Designing the language of the decoder delimits the space of possible representations that we call the program search space. With a Turing complete language, optimal representations yielding compression rates close to the complexity distortion function belongs to the search space. Also, there is a constant c , such that for any $x \in A$, $C(x_1^n) \leq n + c$ [52]. Hence, by a trivial counting argument, we see that the problem of encoding x_1^n can be reduced to a search problem in a space containing $2^{n+c+1} - 1$ programs if we ignore introns.

4.3.3 The Encoder

With the program space specified by the structure of the decoder, the encoder has to explore this space and find an efficient representation for the source object to encode. The idea behind genetic programming is to perform a beam search which is a compromise between exhaustive and hill climbing techniques[4]. In contrast with genetic algorithms, programs are not restricted to have the same length. An evaluation metric, commonly called a fitness measure, is used to measure the efficiency

⁶The terminology is borrowed from biology where it is also referred as “junk DNA”.

of each point in the program space. This number is a measure of how well the corresponding program represents the source object. When lossy compression with resource bounds at the decoder is the main problem, the fitness has to be a function of the amount of distortion introduced by the representation, the length of the representation and its computational complexity. The fitness is used to select out a certain number of the most promising solutions for further transformation. It allows us to incorporate heuristics probabilistically in the search and give more weight to short and accurate representations. The genetic programming method starts by generating randomly an initial population of programs, also called a generation. These programs are then run for less than t steps and using less than s memory cells to determine their fitness. Using these fitness numbers for this population, and following the Darwin principle of survival of the fittest, a new generation of programs is obtained by performing genetic operations. The most common operations are the crossover, the mutation and the reproduction. In the crossover operation, two parent programs belonging to the initial generation are chosen. Subtrees of these programs are randomly chosen and swap to give birth to two offsprings in the new generation. The mutation operation simply changes randomly some nodes in the abstract syntax trees of individuals of the new generation. The reproduction copies good programs in the new generation. Details of these operations can be found in [4]. What is interesting here is that under general conditions (to be mentioned below), when this process is repeated, the probability to have an element with maximum fitness in the population converges to 1 [91]. To see this, note that the dynamic of this algorithm can be modeled by a Markov chain. Populations have fixed size, each possible one corresponding to a state in the Markov chain. Since the object that has to be coded is finite in length, the number of possible states in

this process is finite⁷. The convergence of the genetic algorithms (with fixed length representations) depends on the structure of the transition matrix Q of this Markov chain. As shown in [91], optimality can be reached almost surely in polynomial time if the following two points are satisfied:

1. The second largest eigenvalue⁸ of Q , denoted λ_* , is suitably bounded away from 1 so that the Markov chain is rapidly mixing.
2. The stationary distribution π gives probability greater than ϵ , where $\frac{1}{\epsilon}$ is polynomial in the problem parameter, to the set of states that contains individuals of best fitness.

The first property requires that Q is irreducible with non negative entries which will always be the case if we have a strictly positive mutation probability forcing ergodicity. The second property is more difficult to satisfy. It can be ensured by a good design of the decoder. Assuming that it also holds, the following algorithm can be used at the encoder:

1. From a start state, evolve through a polynomial number of generations;
2. From the final population vector, select the fittest individual.
3. Repeat step 1 and 2 a polynomial number of times.

The third step of the algorithm is necessary to boost the convergence probability. Almost surely discovery of an individual with optimal fitness is guaranteed as long as property 1 and 2 are verified. Unfortunately, to our knowledge, it is still unknown whether property 2 holds for GP systems where the representation length is not

⁷Formally, there is a constant c such that for all $x_1^n \in B_0^n$, $C(x_1^n) \leq n + c$. Therefore, the cardinal of the program space is bounded.

⁸The largest eigenvalue of Q is 1 if the chain is irreducible. Its associated left eigenvector is then π , the stationary distribution.

fixed. The problem comes from the variable length representation of GP which prevents the derivation of closed form expression for the transition matrix.

4.4. Convergence Analysis of Genetic Programming

In contrast with GA, the behavior of GP is not well understood, and the theoretical analysis of the GA model cannot be directly applied to GP. Fortunately, for problems where an upper bound on the representation length is known, it is possible to modify a GP search into a GA search. The idea here is taken from [53]. We stuff the symbol ϕ in programs in order to fix their length to the maximum $n + c$. We then perform a GA search on those extended strings before cleaning the optimal solution by removing the introns ϕ . Fix $L = n + c$. By M we denote the population size of each generation. A program p is a string of length L , $p \in B^L$. From now on, by $l(p)$, we denote the length of the program p without the introns ϕ that were artificially introduced to fix the length of the program strings. Since B is finite, we can define a lexicographic order on elements of B^L and represent them with unsigned integers identifying their place in the lexicographic list. We assume that the ordering is done in the following manner:

1. All programs with outputs yielding strings inside the D -ball centered at x_1^n are ranked based on their length, the shortest ones having a lower lexicographic index. These programs respect the distortion constraint D .
2. All programs with outputs yielding strings outside the D -ball centered at x_1^n are ranked based on the amount of distortion they introduce when representing x_1^n .
3. All programs with outputs yielding strings inside the D -ball centered at x_1^n have lexicographic index lower than any program yielding output strings out-

side the D -ball centered at x_1^n .

From this convention, it is easy to see that the best individuals have very low lexicographic indices. It remains to find an efficient way to identify these individuals in polynomial time. This lexicographic order allows each population k to be represented by a vector $(Z(0, k), Z(1, k), \dots, Z(\alpha^L - 1, k))$ whose entries $Z(i, k)$ denotes the occurrences of individual i in population k , $0 \leq i \leq \alpha^L - 1$, where⁹ $\alpha = |B| - 1 = 3$. . It is easy to see that the total number of different population is given by [60]:

$$N = \binom{M + \alpha^L - 1}{M} \quad (4.23)$$

Let f be the fitness function that assigns a real number to the lexicographic index of each program. For a program p with index i_p , $f(i_p)$ is a function of its length and the amount of distortion introduced by the representation on the decoder Ψ_n after execution in less than t steps and using less than s memory cells. Define f as follows:

$$\begin{aligned} f(i_p) = & 1(D(p) > D) \frac{D_{max} + \beta - D(p)}{D_{max} + \beta} \\ & + 1(D(p) \leq D)(n + c - l(p) + 1) \end{aligned} \quad (4.24)$$

where β is a strictly positive real number, c an integer constant such that $\forall x_n$, $C(x_1^n) \leq n + c$. D_{max} is an upper bound on the amount of distortion that can be introduced at the encoder. $D_{max} = \sup_{(x_1^n, y_1^n) \in A^n \times \hat{A}^n} \{d_n(x_1^n, y_1^n)\}$. Finally, $D(p) = d_n(x_1^n, \Psi(p))$.

Lemma 1 *The function f defined in equation 4.24 respects program ranks i.e. f is monotonic decreasing on its domain.*

⁹The blank symbol is not part of the representation. That's why $\alpha = |B| - 1$.

Proof of lemma 1: Let p_1 and p_2 be two elements of B^L . Assume without loss of generality that $i_{p_1} \leq i_{p_2}$ then following the definition of the lexicographic order, either of the following must be true:

1. $D(p_1) \leq D, D(p_2) \leq D$ and $l(p_1) < l(p_2)$
2. $D(p_1) > D, D(p_2) > D$ and $D(p_1) \leq D(p_2)$
3. $D(p_1) \leq D$ and $D(p_2) > D$

Case 1: Since $D(p_1) \leq D$ and $D(p_2) \leq D$,

$$f(i_{p_1}) = n + c - l(p_1) + 1$$

and

$$f(i_{p_2}) = n + c - l(p_2) + 1$$

Since $l(p_1) \leq l(p_2)$, $f(p_1) \geq f(p_2)$. Hence, the lemma holds for case 1.

Case 2: Since $D(p_1) > D$ and $D(p_2) > D$,

$$f(i_{p_1}) = \frac{D_{max} + \beta - D(p_1)}{D_{max} + \beta}$$

$$f(i_{p_2}) = \frac{D_{max} + \beta - D(p_2)}{D_{max} + \beta}$$

Since $D(p_1) \leq D(p_2)$, $f(i_{p_1}) \geq f(i_{p_2})$. Hence, the lemma holds for case 2.

Case 3: Since $D(p_1) \leq D$ and $D(p_2) > D$,

$$f(i_{p_1}) = n + c - l(p_1) + 1 \geq 1 \geq f(i_{p_2}) = \frac{D_{max} + \beta - D(p_2)}{D_{max} + \beta}$$

. Hence, $f(i_{p_1}) \geq f(i_{p_2})$ and the lemma holds for case 3.

The lemma is proved.

□

Note that when the distortion is smaller than the constraint D , the fitness criterion takes into account only the length of the representation. In practice, this should not matter as long as the distortion constraint is not violated.

The GA search is identical to the GP search. It starts with a random generation of M different programs constituting the first generation. Then, the following three steps are used to generate new good programs:

1. *Selection*: Compute the following probabilities:

$$p(i, k) = \frac{f(i)Z(i, k)}{\sum_{h=0}^{\alpha^L-1} f(h)Z(h, k)} \quad (4.25)$$

$p(i, k)$ is just the probability of having program i at generation k . With these probabilities, randomly select two programs.

2. *Crossover*: Generate two new individuals by exchanging the $1 \leq l \leq L - 1$ right most symbols of the two individuals obtained by the selection phase. The number of exchanged symbols l is chosen uniformly at random¹⁰, from $[1, L - 1]$.
3. *Mutation*: Invert, with probability μ the actual bits (symbols equal to 0 or 1) in the program of the two new individuals obtained from the crossover generation. Ignore the introns.

A new generation is then obtained by reserving the $M - 2m, m \in \{0, 1, \dots, M/2\}$ programs with the highest fitness in the previous generation and applying the three genetic steps, selection/crossover/mutation, to get $\beta = 2m$ new individuals. Clearly, this process can be modeled by a Markov chain with state space S corresponding

¹⁰To simplify the discussion, we consider here the one-point crossover.

to the set of all N different generations. Consider a lexicographic ordering of these N generations; the $N \times N$ transition matrix $Q = (Q_{k,v})$ of this process has been computed by Vose et al. in [60]. $Q_{k,v}$ is just the conditional probability to move to generation v from generation k . When the number of generated individuals equals exactly $\beta = 2m = M$, M even, it is shown in [60] that:

$$Q_{k,v} = M! \prod_{j=0}^{\alpha^L-1} \frac{1}{Z(j,v)!} r(j,k)^{Z(j,v)} \quad (4.26)$$

where $r(j,k)$ is the probability that individual j occurs in population k . This can be proved by noting that $Z(j,k)$ is generated according to the multinomial distribution based on $r(j,k), j = 0, 1, \dots, \alpha^L - 1$. See [60] for a derivation of $r(j,k)$ for a binary alphabet.

4.4.1 Convergence

Let $q^{(0)} = (q_1^{(0)}, q_2^{(0)}, \dots, q_N^{(0)})$ be the initial distribution of the Markov chain. The distribution at generation n is given by $q^{(n)} = (q_1^{(n)}, q_2^{(n)}, \dots, q_N^{(n)}) = q^{(0)}Q^n$, Q^n being the n -th power of the transition matrix Q . The objective of the algorithm is to converge as fast as possible to a population that includes an optimal program. Let κ be the set of populations which include the individual with the highest fitness value. In this section, we investigate how closely $\sum_{k \in \kappa} q_k^\infty$ is to one. In [67], it is shown that the transition matrix of the simple GA with $\beta = 2m = M$ is primitive (irreducible aperiodic) and this implies that $\sum_{k \in \kappa} q_k^\infty$ does not converge to one since the other states $k \notin \kappa$ are also recurrent non null. It does converge to a stationary distribution and Monte Carlo sampling can be used to almost surely discover the best individual [91]. In [85], almost surely convergence is obtained in another fashion by preventing the genetic operators from reducing the performance of the best individual during the evolution. In that paper, a spot is always reserved in the next generation for the

best individual of the current generation. In this modified GA algorithm, also called the elitist strategy, β is set at $2m = M - 1$, M obviously odd. As a consequence, the transition matrix becomes:

$$Q_{k,v} = (M - 1)! \prod_{j=0}^{\alpha^L - 1} \frac{1}{Y(j,v)!} r(j,k)^{Y(j,v)} \quad (4.27)$$

for $i^*(k) \geq i^*(v)$, and 0 for $i^*(k) < i^*(v)$, where $i^*(j)$ is the index of the best program in generation j , and where

$$Y(j,k) = \begin{cases} Z(j,k) & \text{if } j \neq i^*(k) \\ Z(j,k) - 1 & \text{if } j = i^*(k) \end{cases}$$

In this case, the transition matrix Q is indecomposable (reducible with only one aperiodic recurrent class). In fact, it is shown in [85] that Q has α^L sub-matrices $Q(i)$ of size $N(i) \times N(i)$, $i = 0, 1, \dots, \alpha^L - 1$, as diagonal elements and all the components to the upper right of these sub-matrices are zeros. $N(i)$ is the number of populations k in which $i = i^*(k)$. It is shown to be:

$$N(i) = \begin{pmatrix} M - 1 + \alpha^L - i \\ M - 1 \end{pmatrix} \quad (4.28)$$

It is shown in [85] that for this modified GA algorithm, there is a constant C such that

$$\sum_{k \in \kappa} q_k^{(n)} \geq 1 - C |\lambda_*|^n, \quad (4.29)$$

where

$$|\lambda_*| = \max_{1 \leq i \leq 2^L - 1} \max_{1 \leq j \leq N(i)} |\lambda_{i,j}| \quad (4.30)$$

and $\lambda_{i,j}$, $j = 1, 2, \dots, N(i)$ denotes the $N(i)$ eigenvalues of the sub-matrix $Q(i)$, $i = 0, 1, \dots, 2^L - 1$ which are identical to the eigenvalues of Q . Since λ_* is less

than 1, convergence is guaranteed. The convergence to one can also be guaranteed by decreasing gradually the mutation probability in a simulated annealing fashion. In fact, the analogy with simulated annealing can be taken further. The mutation operator can be interpreted as a temperature parameter controlling the stability of the system. It also guarantees convergence in the almost sure sense to globally optimal solutions. With a proper “cool down” schedule [29, 86, 84], convergence is guaranteed [29, 86].

All these results prove the existence of a family of probabilistic algorithms converging to the best individual with probability 1. Note that this convergence to 1 although necessary, is not sufficient for the design of an efficient coding system. After all, if this was the only criterion for efficiency, we would have used a full search method. This is precisely why we analyze the speed of the convergence of these techniques in the next section.

4.4.2 Speed of Convergence

Ideally, we would like to reach any reasonable level of convergence (in terms of probability to find optimal representation) in polynomial time with a reasonable polynomial exponent. To formalize this, we follow the concept of rapidly mixing Markov Chain as described in [75].

Definition 1 Consider a family of Markov Chains $\mathcal{MC}(x)$ parameterized on strings $x \in A$, with a finite set of states S , transition matrix $Q = (Q_{ij})$, stationary distribution (when it exists) $\pi = (\pi_i)$ and relative pointwise distance (r.p.d.) for each $x \in A$, over $U \subseteq S$ after t steps,

$$\Delta_U^{(x)}(t) = \max_{i \in S, j \in U} \frac{|Q_{ij}^{(t)} - \pi_j|}{\pi_j}$$

where $Q_{ij}^{(t)}$ is the (i, j) entry of Q^t . The family of chains is rapidly mixing iff there exists a polynomially bounded function $q : \mathcal{N} \times \mathcal{R}^+ \rightarrow \mathcal{N}$ such that

$$\tau_U^{(x)}(\epsilon) \leq q(l(x), \log_2 \epsilon^{-1})$$

where

$$\tau_U^{(x)}(\epsilon) = \min \{t \in \mathcal{N} : \Delta_U^{(x)}(t) \leq \epsilon \forall t' \geq t\} \quad (4.31)$$

for all $x \in A$ and $0 < \epsilon \leq 1$.

Note that this concept was introduced for ergodic chains. Here, we slightly extend it and this extension is valid as long as the set of states U contains only recurrent non null states. $\tau_U^{(x)}(\epsilon)$ measures how quickly the states in U reach stationarity after a start in any state $i \in S$. The rapid mixing condition guarantees that the convergence to stationarity is not too slow and that any acceptable level of convergence (quantified by ϵ) can be reached in polynomial time. For our problem, the parameter x is just the sequence that we would like to encode. We also fix U to the set of states corresponding to the submatrix $Q(0)$ if we adopt the elitist strategy [85]. In general, U is the set of states containing optimal elements. For the elitist strategy, more intuition behind this convergence lies in the structure of the transition matrix Q [85] as shown in figure 4-3. The states of the chain have been ordered according to lexicographic order i_{p_j} . The submatrices $Q(i)$ have been defined above and the matrix $Q(0)$ contains exclusively states (or populations) with optimal individuals (programs). By copying the best individual of the generation in the next generation, we guarantee that the chain is not ergodic and converges to the submatrix $Q(0)$. The speed of this convergence clearly depends on the structure of Q . For example, if the first column of Q is $[1, 1, \dots, 1]^T$, it would almost surely take only 1 generation for convergence to the best individual. Intuitively, the more the energy is compacted

in the lower diagonal of Q , the faster the process will converge. The second largest eigenvalue λ_* of Q dictates the speed at which the process converges to optimality. Clearly, the lower $|\lambda_*|$ is, the faster the convergence is. Recall also from elementary algebra Gerschgorin's theorem that states that the module of each eigenvalue of Q is upper-bounded by $\max_k \sum_v Q_{k,v}$. This theorem also applies to each of the $Q(i)$ submatrices, especially to the one having λ_* for eigenvalue. As a result, we see that the magnitude of this eigenvalue is upper bounded by the amount of energy in the most "energetic" row of one of the $Q(i)$'s. Note that transitions within the submatrices $Q(i)$ are not desired because these transitions corresponds to events where the new generation does not discover anything new. The fitness of the best individual does not increase in these cases. The magnitude of the second largest eigenvalue is a very good indicator of the probability of having such transitions, as being upper bounded by the maximum probability of not evolving. The larger this value is, the slower the convergence is since most of the energy in Q will then be in the submatrices $Q(i)$ and not below the diagonal that these submatrices form. These intuitive observations have a lot in common with the concept of conductance discussed in [75] and can be generalized for the other genetic algorithms approaches. In the rest of this section, we compute an upper bound for the r.p.d. of each chain in the family defined by the encoding algorithm. Following the algebraic approach proposed in [75], we derive general conditions on the second largest eigenvalue of Q to guarantee the rapidly mixing property and show that it is possible to solve our universal coding problem in polynomial time.

To ensure the rapidly mixing property, the following result is used: As mentioned in [75], this definition of rapid mixing "is rather a strict one because it is based on the relative pointwise distance, which is a severe measure for two reasons. Firstly, it demands that the distribution of the chain be close to the stationary distribution

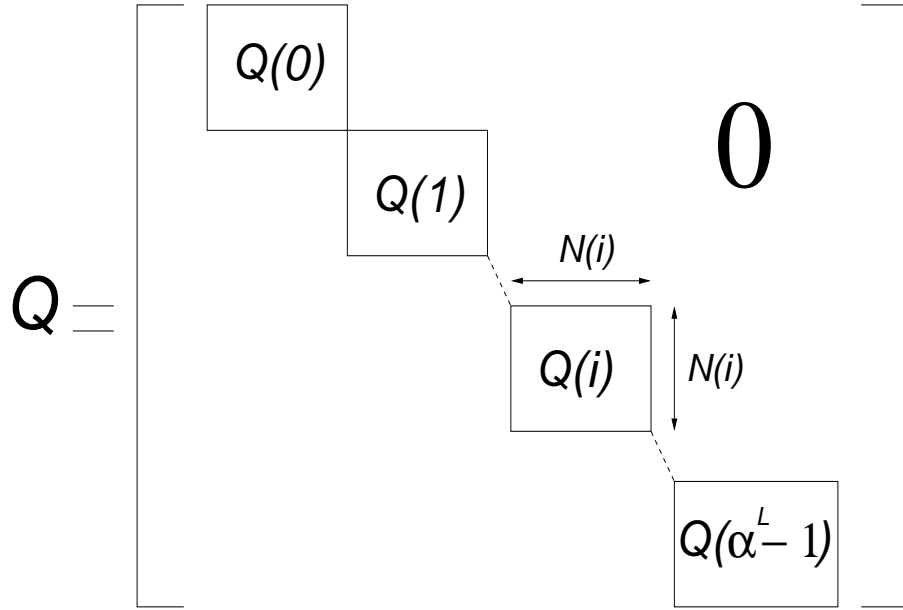


Figure 4-3: The transition matrix for the elitist strategy.

at every point (in U for our problem) and secondly, it demands that convergence be rapid from every initial state”.

Lemma 2

$$\Delta_U^{(x)}(t) \leq \frac{|\lambda_*^t|}{\min_{j \in U} \pi_j}$$

For all t even,

$$\Delta_U^{(x)}(t) \geq \lambda_*^t$$

Moreover, if all eigenvalues of P are non-negative, the lower bound holds for all $t \in \mathcal{N}$

Proof: See [75]

□

Despite imposing restrictions in lemma 2, the sign of the eigenvalues never present an obstacle to rapid mixing because any chain can be modified to have all eigenvalues positive, as explained in [75]. Hence, from this lemma, it is clear that the rapid

mixing property will be respected *if and only if* the second largest eigenvalue is suitably bounded away from 1. We sum up this fact in the following theorem:

Theorem 3 *The rapid mixing condition is guaranteed if and only if*

$$\frac{1}{\log_2 \frac{1}{|\lambda_*|}} = p(l(x), \log_2 \epsilon^{-1}) \quad (4.32)$$

$p(l(x), \log_2 \epsilon^{-1})$ being a positive polynomial function in $l(x)$, the size of the input.

Proof: Assume that the chain is rapid mixing. Then by definition, for each $0 < \epsilon < 1$, there is a polynomially bounded function $q : \mathcal{N} \times \mathcal{R}^+ \rightarrow \mathcal{N}$ such that:

$$\tau_U^{(x)}(\epsilon) \leq q(l(x), \log_2 \epsilon^{-1})$$

where

$$\tau_U^{(x)}(\epsilon) = \min \{ t \in \mathcal{N} : \Delta_U^{(x)}(t') \leq \epsilon \ \forall t' \geq t \} \quad (4.33)$$

Hence,

$$\forall t \geq \tau_U^{(x)}(\epsilon), \Delta_U^{(x)}(t) \leq \epsilon$$

By lemma 2,

$$|\lambda_*|^t \leq \Delta_U^{(x)}(t) \leq \epsilon$$

This implies that:

$$t \log_2 |\lambda_*| \leq \log_2 \epsilon$$

And this can be rewritten as:

$$t \log_2 \frac{1}{|\lambda_*|} \geq \log_2 \epsilon^{-1}$$

Therefore,

$$\frac{\log_2 \epsilon^{-1}}{\log_2 \frac{1}{|\lambda_*|}} \leq t, \forall t \geq \tau_U^{(x)}(\epsilon)$$

Since $\tau_U^{(x)}(\epsilon) \leq q(l(x), \log_2 \epsilon^{-1})$, we conclude that

$$\frac{\log_2 \epsilon^{-1}}{\log_2 \frac{1}{|\lambda_*|}} = p(l(x), \log_2 \epsilon^{-1})$$

with $l(x) = n + c$ for the universal coding problem.

Assume that

$$\frac{1}{\log_2 \frac{1}{|\lambda_*|}} = p(l(x), \log_2 \epsilon^{-1})$$

From the definition of $\tau_U^{(x)}(\epsilon)$,

$$\forall t \geq \tau_U^{(x)}(\epsilon), \Delta_U^{(x)}(t) \leq \epsilon$$

By lemma 2, if $\frac{|\lambda_*|^t}{\pi_{min}} \leq \epsilon$ then $\Delta_U^{(x)}(t) \leq \epsilon$ where $\pi_{min} = \min_{j \in U} \pi_j$. Saying that $\frac{|\lambda_*|^t}{\pi_{min}} \leq \epsilon$ is equivalent to say that

$$t \log_2 |\lambda_*| \leq \log_2 \pi_{min} \epsilon$$

and this can be rewritten as:

$$t \geq \frac{\log_2(\pi_{min} \epsilon)^{-1}}{\frac{1}{|\lambda_*|}}$$

The last equation holds for all $t \geq \tau_U^{(x)}(\epsilon)$. Therefore, if $\frac{\log_2(\pi_{min} \epsilon)^{-1}}{\frac{1}{|\lambda_*|}}$ is polynomial in $l(x)$ and $\log_2 \epsilon^{-1}$, so is $\tau_U^{(x)}(\epsilon)$ and the chain is rapid mixing. And this will hold if $\log_2 \pi_{min}^{-1}$ is polynomial in $l(x)$, a result that can be easily derived for the genetic problem by taking a closer look at the mutation operator. In this case, note that

there is at least a probability equal to μ^L , ($L = n + c$) to change from any state of the chain to any other state. Therefore, $\pi_{min} \geq \mu^L$ and $\log_2 \frac{1}{\pi_{min}} \leq L \log_2 \frac{1}{\mu}$. We conclude that the chain is rapid mixing and we have proved theorem 3.

□

In general, closed form expressions for λ_* are difficult to compute but for one point crossover systems it is shown in [97] that $\tau_U^{(x)}(\epsilon) \leq l(x) \ln(l(x)) + \ln \epsilon^{-1}$. This provides a bound for $|\lambda_*|$ and shows that the proposed universal coding algorithm is fast.

4.5. Algorithmic Representation of Images

In this section, we apply the algorithm described in the previous section to still image data. Our aim is to model visual information deterministically, with no probabilistic assumption. Similar experiments were performed by Nordin [61] on images and sound and Koza [47] on images with primitive language that looked like machine code with no theoretical analysis and little success in terms of distortion. We take these experiments one step further and significantly improve the quality of the representation to approach acceptable levels in terms of distortion and rate, despite using a very simple language also. We first describe the decoding operations before presenting the encoding steps with a few experimental results.

4.5.1 The Decoder

The decoding device is completely specified by its language. Tables 4.1 and 4.2 show a complete description of the instruction used for these tests. As shown there, the language used is block-based. We restrict ourselves to blocks to keep the implementation simple. This language contains two types of functions and terminals, a block type (blk) and an integer type (int). Function nodes involving the blk type

Instruction	Return Type	Argument 1	Argument 2	Argument 3	Action
Rotate	blk	(blk) block	(int) angle	–	Rotation
Scale	blk	(blk) block	(int) angle	–	Scaling
TFilter	blk	(blk) block	(int) angle	–	Filter
LFilter	blk	(blk) block	(int) angle	–	Filter
Min	blk	(blk) block	(blk) input-block-2	–	Minimization
Max	blk	(blk) block	(blk) input-block-2	–	Maximization
Trans	blk	(blk) block	(int) arg1	(int) arg2	Translation
Read	blk	(int) x-index	(int) y-index	–	Memory Read
Write	blk	(int) x-index	(int) y-index	(blk) block	Memory Write
*	int	(int) arg1	(int) arg2	–	Multiplication
/	int	(int) arg1	(int) arg2	–	Division
+	int	(int) arg1	(int) arg2	–	Addition
-	int	(int) arg1	(int) arg2	–	Subtraction

Table 4.1: Functions used to represent gray level image data.

generally perform spatial operations (rotations, scale, translation etc) except for the Read and Write function nodes. These two functions are used to access a two dimensional memory where blocks of data can be stored and retrieved during the computation. There are also 2 terminal nodes of type blk that can be used in the computation. Most of the function nodes of type blk require integer parameter to control their execution. The standard arithmetic operations $+$, $-$, $*$, $/$ are used for this purpose, together with integer terminals from 1 to 9.

This language is extremely simple. It is not even Turing complete but it contains enough nodes to illustrate how the genetic programming approach described in the previous section can be used to develop deterministic models for images. As a result, we do not attempt to compete with state of the art image compression systems [59, 68]. For this, we would need a much more complex language developed from all the expertise that the image processing community has gained over the years as discussed in chapter 5, section 5.2.2.

Instruction	Return Type	Action
Dark	blk	Terminal (blk)
Light	blk	Terminal (blk)
1	int	Terminal (int)
2	int	Terminal (int)
3	int	Terminal (int)
4	int	Terminal (int)
5	int	Terminal (int)
6	int	Terminal (int)
7	int	Terminal (int)
8	int	Terminal (int)
9	int	Terminal (int)

Table 4.2: Terminals used to represent gray level image data.

4.5.2 The Encoder

The general encoder architecture is shown in figure 4-4. At the heart of this system is the GP module performing a GP search in the program search space defined by the decoding language and working in tandem with the UTM. The GP module uses the genetic operations described in section 4.3.3. To do so, we have developed a Typed GP kernel that allows us to specify decoding programs with different node types. This system is an extension of the public GP system described in [27] which like most GP kernel does not allow the use of nodes with different return types in the syntax of the decoding language.

Every time an incoming block is given to this module, a best program for the representation of this block is found and tested. The resulting block is then stored in the 2D memory and is now available for GP encoding of the future blocks. As explained in figure 4-5, the 2D memory is indexed from the location of the current block to encoded in the following manner: assume that we would like to encode block (x, y) , x representing the column location in the image and y the row location. Then memory indices $(0, 0)$ would correspond exactly to position (x, y) . Memory indices (i, j) would correspond to absolute locations $(x - i, y - j)$ in the memory. This way,

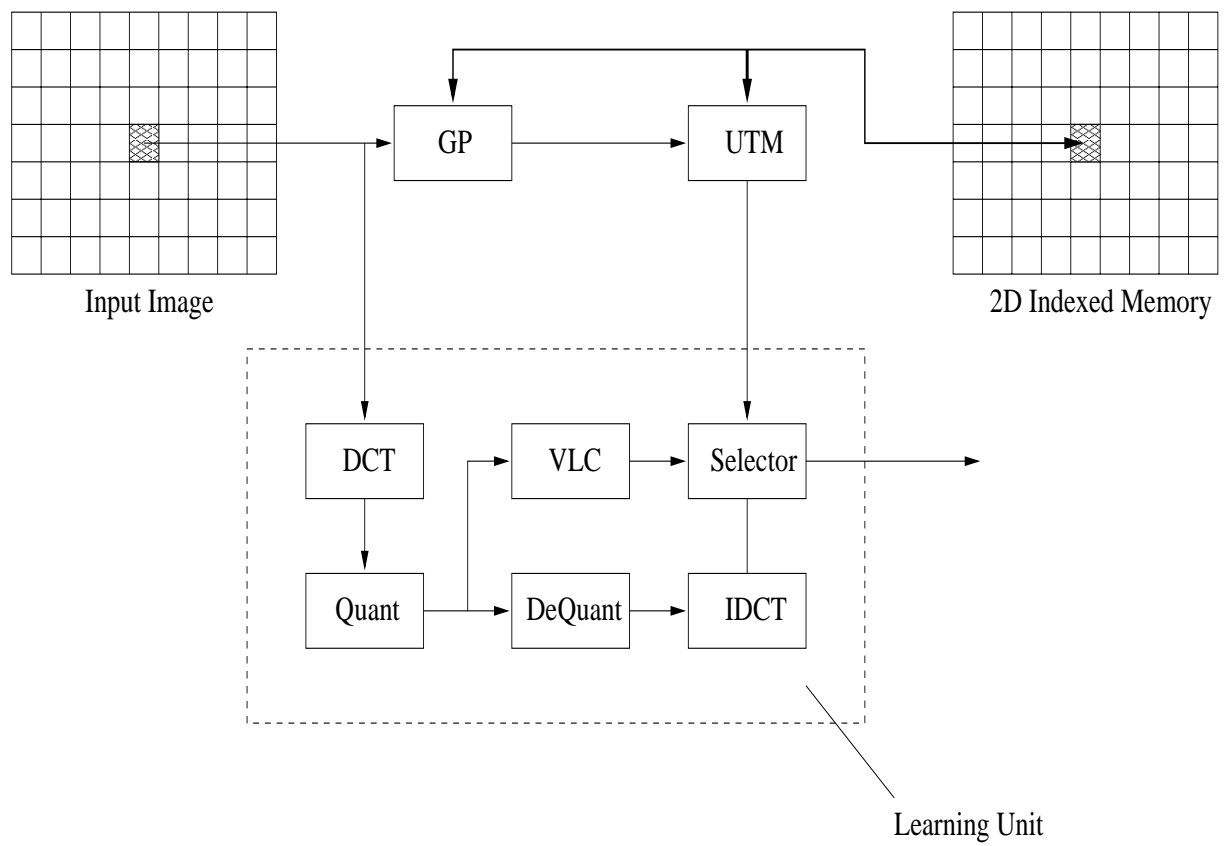


Figure 4-4: Hybrid Image Encoder.

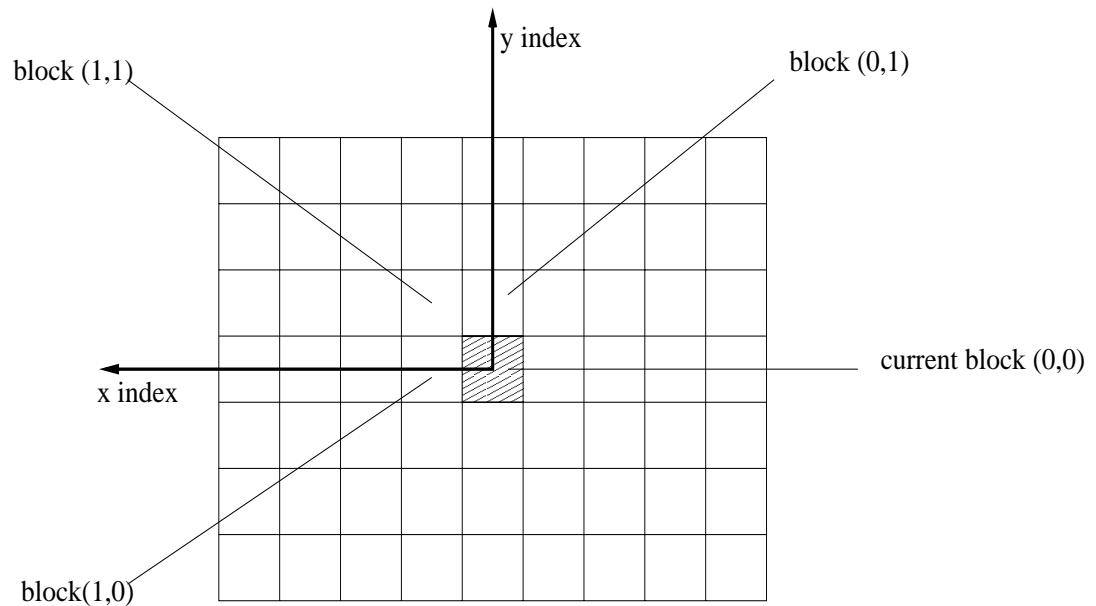


Figure 4-5: This figure shows how the memory indexing is done, relatively to the position of the current block.

we exploit the Markovian structure of images and index neighboring blocks with small indices requiring less computation and less function nodes (arithmetic) in the programs.

Due to the limitation imposed by the program space, the search is not always successful in finding a best program meeting the distortion constraint incorporated inside the fitness function. To resolve this problem, we have introduced a learning unit in the system, that would teach the GP module how to represent complex blocks. As result, the system can operate in two distinct modes. In a pure GP mode, the learning unit is disabled. The systems sends to the decoder the best program identified by the GP module. In a hybrid mode, we allow the codec to bypass the GP output and send a traditional DCT based representation of the block if the GP module fails to find a good representation, based on distortion alone.

In figure 4-6 we show the results of a pure GP evolution for lenna 512x512, using 4x4 blocks. To have an idea of well GP performs note that these results in terms



(a)



(b)

Figure 4-6: Programmatic representation of lenna 512x512, psnr = 29.72 dB at 1.17 bpp: (a) original, (b) gp output. In this run the language uses 4x4 blocks. Note the large errors introduced in the top of the picture because of the inability of the language to represent these blocks when the memory of the system is empty.

of psnr and number of bits per pixel (bpp) of the pure GP mode are still far from what a DCT codec system would give using 8x8 blocks but this is mainly due to the difference in block sizes and the limitations of the decoding language.

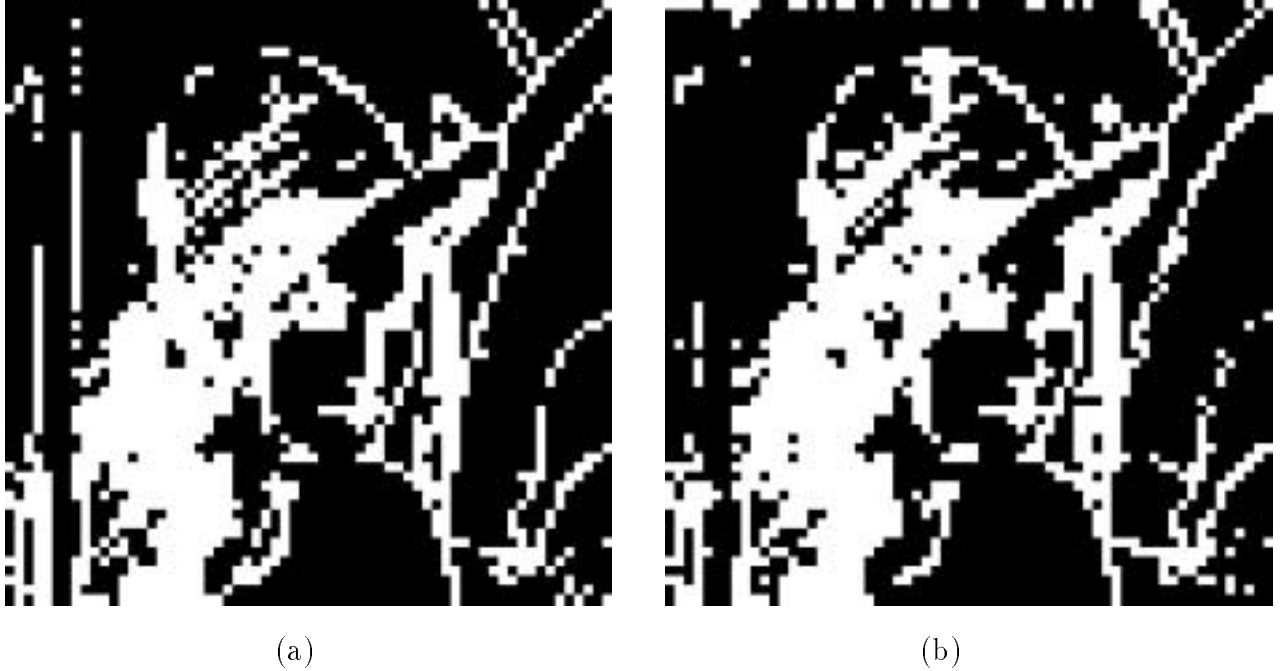


Figure 4-7: This figure compares 8x8 blocks with a significant amount of edges (according to a Sobel edge detector) with blocks that cannot be represented accurately by the GP module. Edge blocks and GP blocks with large MSE are in white: (a) edge blocks, (b) GP blocks with large error. Note the similarities between this two pictures showing that the language did not manage to represent accurately most of the edge blocks.

Like VQ techniques, this pure GP systems does not perform well for blocks bigger than 4x4 without help from the DCT learning unit. This is mainly due to the use of the mean square error (MSE) in the fitness function as a distortion measure. It is well known [30] that the MSE does a poor averaging job in VQ type systems especially around the sharp edges of the image. These edges become blurred. In figure 4-7, we illustrate this point by showing the locations of the blocks that could not be represented accurately by a pure GP system using 8x8 blocks. These locations corresponds to the edges of the source images. With the learning

unit, the gp system performs closer to a standard DCT systems with variable length coding. Experimental results are shown in figures 4-8,4-9 and 4-10. In all cases, the system was not able to encode the strong edges.



Figure 4-8: Programmatic representation of lenna 512x512 with the learning unit, psnr = 32.97 dB at 0.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 1381 DCT blocks (out of 4096), mainly at the strong edges of the image.

4.6. Conclusion

In this chapter, we have introduced computational resource constraints to source coding. The result is an extension of the rate distortion function into a surface showing the tradeoff between information rate, distortion and computational complexity, that we called the complexity distortion surface. The convexity of this surface has been established in the first part of this chapter by exploiting equivalences between complexity distortion functions and mutual information. In the second part of the chapter, we proposed a novel approach to universal data compression with an algorithm able to perform at rates arbitrarily close to the complexity distortion surface. We studied the converging properties of the proposed method before illustrating it



(a)



(b)

Figure 4-9: Programmatic representation of house 256x256 with the learning unit, psnr = 35.75 dB at 0.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 425 DCT blocks (out of 1024), mainly at the strong edges of the image



(a)



(b)

Figure 4-10: Programmatic representation of peppers 256x256 with the learning unit, psnr = 34.02 dB at 1.9 bpp: (a) original, (b) gp output. In this run the language uses 8x8 blocks. The learning unit introduced 643 DCT blocks (out of 1024), mainly at the strong edges of the image

on gray level still image data. The proposed technique does not rely on any probabilistic assumptions and does not require infinite source objects for convergence to optimality.

Chapter 5

Conclusion and future directions

5.1. Conclusion

We have presented a novel approach to source coding by extending Kolmogorov complexity theory, which replaces the decoder by a universal Turing machine in Shannon's classical communication system, to the lossy case yielding complexity distortion theory. The motivation behind this work was initially behind the current trend in media representation that requires a unified approach for all the spectrum of coding techniques, from traditional entropy coding to modern approaches like fractal and model-based coding. This current trend also sees the emergence of new problems generalizing the classical compression rate/distortion optimization into more complex problems which requires a better understanding of the information content, i.e., object-based design, seamless access to content, editing in the compressed domain, scalability and graceful degradation for network transmission, graceful degradation with diminishing decoder capabilities, flexibility in algorithm selection, and even downloadability of new algorithms. With this in mind, we have defined the complexity distortion function and proved that this function is equal with probability one, to the rate distortion function for all stationary and ergodic process with recursive probability measure. The mathematical derivation of this

equivalence is at the heart of this thesis. It clearly highlights two major points of fundamental importance for the design requirements of today's communication systems.

First, this result serves as theoretical bridge from Kolmogorov's deterministic setting to Shannon's probabilistic setting enabling us to tackle very important issues that are beyond the scope of traditional source coding, namely how to introduce decoding computational resource considerations in information theory. These issues can be defined clearly in Complexity Distortion Theory and then bridged to classical information theoretic entities to benefit from the well understood properties of the main information theoretic concepts. Following this approach, we have defined the tradeoff between computational complexity, information rate and distortion and via links to the mutual information, we derived conditions for the convexity of this surface, a key property for the design of efficient resource management algorithms at the decoding end of such programmable communication systems. This introduction of resource bounds considerations also brings up several interesting open problems related to our understanding of the capacity of a channel. These issues are discussed next in section 5.2.1.

Second, the proposed proof for the equivalence between CDF and RDF identifies clearly the null set where these two entities are not equivalent. This set corresponds to the set of non random sequences with strong deterministic patterns. Because of its low measure, this set is neglected in the design of efficient representation algorithms. Typical algorithms like Lempel-Ziv, Huffman Coding and Arithmetic Coding ignore completely this set and focus on its much larger complement. This complement does not have measure one for the coding of finite objects but Barron's lemma and randomness tests clearly show that it has a very high measure even for finite sequences. Hence from a mathematical perspective, this set does not seem to

have a significant importance. Problems arise when we observe audio-visual (synthetic and natural) objects around us and how they are perceived by the human visual system. They do have a lot of important structural properties that seems to escape from our classical mathematical model. We believe that these objects have strong components that belongs to this very small set of non random sequences. Assuming the contrary would assume that these signals are completely patternless and would prevent us from analyzing them with our human visual system. Motivated by these observations, we have developed a novel approach to lossy universal source coding for finite length objects attempting to encode objects at rate closed to the complexity distortion surface. Under these lines, we have proposed an efficient universal coding technique based on genetic programming. The technique was then illustrated on image data, using a simple block-based language. These simple experiments highlight the importance of designing good languages for media representation as discussed in the following, in section 5.2.2.

5.2. Future directions

5.2.1 Channel Capacity versus System Capacity

Source coding with resource bounds brings up several interesting open questions that have not receive much attention in information theory. Indeed, as mentioned in [69], traditional information theory does not differentiate the channel capacity from the system capacity. The former defines the maximum amount of information that can be transmitted by a channel defined probabilistically with a conditional distribution. The latter defines the maximum amount of information that can be reproduced at the decoding end of the communication system. Clearly, with no decoding computational constraints, these two quantities are equal since the only bottleneck in the entire system is due to the reliability of the channel. With lim-

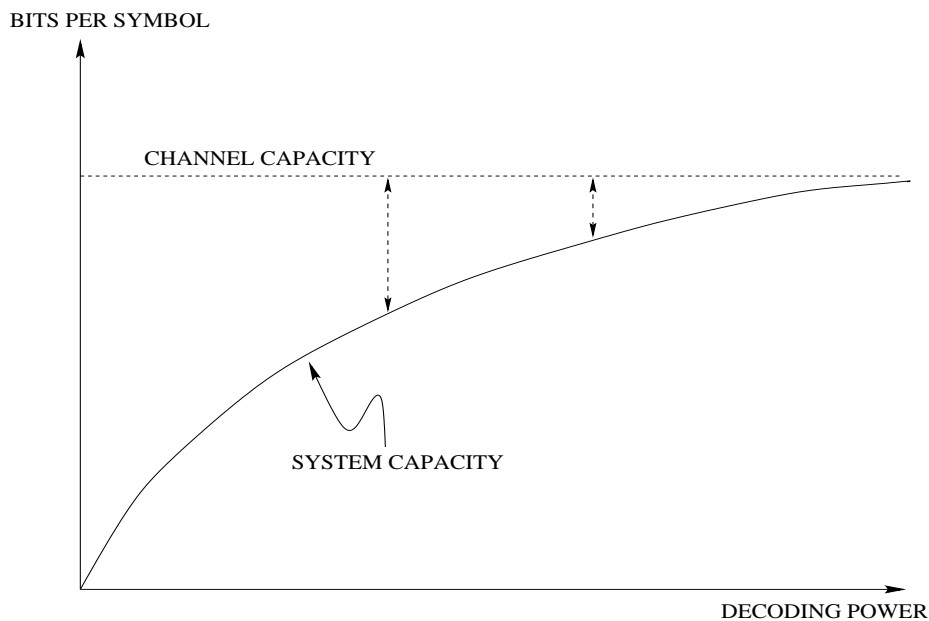


Figure 5-1: Channel and system capacities.

As the decoding power increases, the system capacity converges to the channel capacity. In fact, the channel capacity is always larger than the system capacity simply because the channel is part of the system. As the computational power of the decoder increases, the system capacity gets closer and closer to the channel capacity (see figure 5-1) and there is an urgent need to understand this convergence. Defining such a system capacity is not natural in the classical IT framework. Once more, CDT provides an interesting setting where this issue can be addressed. To do this, we could conceptually group together the channel and both the source and channel decoder¹, as shown in figure 5-2. We call the resulting block *the system channel*. We allow the decoding operations to be lossy in order to meet the space and time requirements and the system capacity takes into account this decoding loss of information. This situation is common in practice when a decoding general purpose digital signal processor (DSP) has to meet various hard timing constraints imposed by the application and

¹We group both the source and channel decoder together because computational bounds on the source and channel decoder also affect the system capacity.

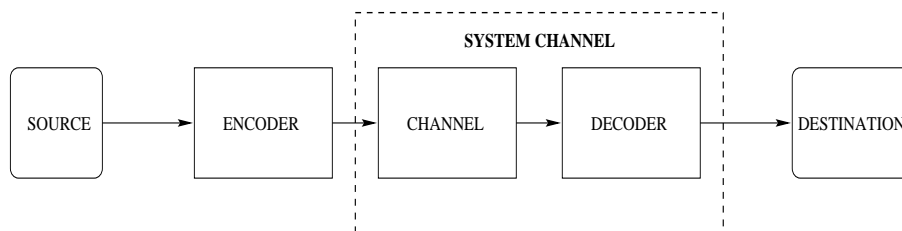


Figure 5-2: General system channel.

must drop significant information to respect these constraints. More work on this issue will be done in the future.

5.2.2 Language Design

Efficient representation languages for visual representation have yet to be formalized in media representation. In audio processing, MPEG-4 introduced a Turing complete language called Structured Audio Orchestra Language (SAOL). Under the same lines, there is a need for the equivalent in image/video processing to improve current state of art representation techniques beyond compression considerations. Such a language should benefit from the prior work in this field on which a significant amount of new features could be added for Turing completeness and true universality. We believe that this step is necessary for the design of visual information system at high semantic levels. Currently, there is a significant gap between low level data at the bit level and high level data like it is perceived by the human visual systems. A good representation language would significantly close this gap and enable the design of more powerful visual information systems. There is an interesting analogy with general computing systems that also process information. At the bit or even assembly language level, we can only dream about designing complex applications like word processors or web browsers. With the development of modern operating systems and modern computer languages like C/C++, these dreams became reality. This addition of software layers on top of the raw hardware

enabled us to communicate in a more adequate way with the hardware. Similarly, good representation languages would allow us to have better connections between the human visual system and digital image/video data.

References

- [1] E. Allender. *Time Bounded Kolmogorov Complexity in Complexity Theory in the Book Kolmogorov Complexity and Computational Complexity*, chapter 2. Springer-Verlag, 1992.
- [2] M. Anthony and N. Biggs. *Computational Learning Theory, An Introduction*. Cambridge University Press, 1992.
- [3] O. Avaro, P. Chou, A. Eleftheriadis, C. Herpel, and C. Reader. The MPEG-4 System and Description Languages: A Way Ahead in Audio Visual Information Representation. *Signal Processing: Image Communication*, 1997.
- [4] W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Fracone. *Genetic Programming: An Introduction*. Morgan Kaufmann, 1998.
- [5] M.F. Barnsley. *Fractals Everywhere 2nd ed.* Academic Press Professional, 1993.
- [6] T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Inc., 1971.
- [7] T. Berger and J.D. Gibson. Lossy source coding. *IEEE Transactions on Information Theory*, 44.
- [8] R.V. Book. *On Sets with Small Information Content in the book Kolmogorov Complexity and Computational Complexity*, chapter 3. Springer-Verlag, 1992.
- [9] C. Calude. *Theories of Computational Complexity*. North-Holland, 1988.
- [10] C. Calude. *Information and Randomness An Algorithmic Perspective*. Springer-Verlag, 1994.
- [11] G. J. Chaitin. On the Length of Programs for Computing Finite Binary Sequences. *Journal of The Association for Computing Machinery*, 13, 1966.
- [12] G.J. Chaitin. A Theory of Program Size Formally Identical to Information Theory. *Journal of The Association for Computing Machinery*, 22, 1975.

- [13] P.A. Chou, M. Effros, and R.M. Gray. A Vector Quantization Approach to Universal Noiseless Coding and Quantization. *IEEE Transactions on Information Theory*, 42, 1996.
- [14] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1995.
- [15] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Son, 1991.
- [16] I. Csiszar. The method of types. *IEEE Transactions on Information Theory*, 44.
- [17] M. Davis. *Computability and Unsolvability*. Dover Publications, Inc, New York, 1982.
- [18] L.D. Davisson, G. Longo, and A. Sgarro. The Error Exponent for the Noiseless Encoding of Finite Ergodic Markov Sources. *IEEE Trans. Info. Th.*, 27, 1981.
- [19] D.L. Donoho, M. Vetterli, R.A. DeVore, and I. Daubechies. Data compression and harmonic analysis. *IEEE Transactions on Information Theory*, 44.
- [20] G.F. Luger ed. *Computation and Intelligence*. The MIT Press, 1995.
- [21] R. Vaananen E.D. Scheirer and J. Huopaniemi. AudioBIFS: Describing Audio Scenes with MPEG-4 Multimedia Standard. *IEEE Transactions on Multimedia*, 1:3:237–250, 1999.
- [22] M. Effros, P.A. Chou, and R.M. Gray. Weighted Universal Image Compression. *IEEE Transactions on Image Processing*, 10, 1999.
- [23] A. Eleftheriadis. Flavor: A Language for Media Representation. In *Proceedings, ACM Multimedia 97 Conference*, Seattle, WA, 1997.
- [24] Y. Fang and A. Eleftheriadis. A Syntactic Framework for Bitstream-Level Representation of Audio-Visual Objects. In *Proceedings, 3rd IEEE Int'l Conf. on Image Processing*, pages II.421–II.424, Lausanne, Switzerland, September 1996.
- [25] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 2. John Wiley & Sons, 2nd edition, 1957.
- [26] G. B. Folland. *Real Analysis, Modern Techniques and their Applications*. John Wiley & Sons, Inc, 1984.

- [27] A. Fraser and T. Weinbrenner. The genetic programming kernel version 0.5.2. <http://www.emk.e-technik.th-darmstadt.de/thomasw/gp.html>, 1997.
- [28] R. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc, 1968.
- [29] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- [30] A. Gersho and R.M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, 1992.
- [31] K. Gödel. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. Dover Publications Inc., 1992.
- [32] A. Goldberg and M. Sipser. Compression and ranking. *SIAM J. Comput.*, 20:524–536, 1991.
- [33] M. Gormish. *Source Coding with Channel, Distortion and Complexity Constraints*. PhD Dissertation, Dept. Elec. Eng. Stanford University, 1994.
- [34] V. Goyal. *Beyond Transform Coding*. PhD Dissertation, Dept. Elec. Eng. and Comp. Sciences, U. C. Berkeley, 1998.
- [35] V. Goyal and M. Vetterli. Computation-distortion characteristics of block transform coding. *Proc. IEEE Int. Conf. Acoustics, Speech, & Sig. Proc.*, 4:2729.
- [36] V. Goyal and M. Vetterli. Computation-distortion characteristics of jpeg encoding and decoding. *Proc. 31st Asilomar Conf. on Signals, Systems, & Computers 1997*, 1:229.
- [37] R.M. Gray. *Probability, Random Processes, and Ergodic Theory*. Springer-Verlag, 1988.
- [38] R.M. Gray. *Source Coding Theory*. Kluwer Academic Publishers, 1990.
- [39] R.M. Gray, D.L. Neuhoff, and D.S. Ornstein. Non-block source coding with a fidelity criterion. *Annals of Probability*, 3:478–491, 1975.
- [40] J.E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory Languages, and Computation*. Addison Wesley, 1979.
- [41] D.A. Huffman. A method for the construction of minimum redundancy codes. *Proc. IRE*, 1952.

- [42] R.C. Jeffrey. *Basic Problems in Methodology and Linguistics*, chapter Mises Redux, pages 213–222. 1977.
- [43] A. Kolmogorov and V. Uspenskii. Algorithms and Randomness. *SIAM Journal Theory Probab. Appl.*, 32:3:389–412, 1987.
- [44] A. N. Kolmogorov. Three Approaches to the Quantitative Definition of Information. *Problems Inform. Transmission*, 1(1):1–7, 1965.
- [45] A.N. Kolmogorov. Logical basis for information theory and probability theory. *Trans. Inform. Theory*, 1968.
- [46] Y. Kontoyiannis. Pointwise redundancy in lossy data compression and universal lossy data compression. *IEEE Transactions on Information Theory*, 46:136–152, 2000.
- [47] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press., 1992.
- [48] ed L. Torres and ed M. Kunt. *Video Coding: The Second Generation Approach*. Kluwer Academic, 1996.
- [49] K. Lengwehasatit and A. Ortega. Distortion/decoding time tradeoffs in software dct-based image coding. *Proc. IEEE Int. Conf. Acoustics, Speech, & Sig. Proc.*, 4:2725.
- [50] Sik K. Leung-Yan-Cheong and Thomas M. Cover. Some Equivalence Between Shannon Entropy and Kolmogorov Complexity. *IEEE Transactions on Information Theory*, 24:331–339, 1978.
- [51] L.A. Levin. Universal search problems. *Problems Inform. Transmission*, 9:265–266, 1973.
- [52] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Text and monographs in Computer Science Springer-Verlag, 2 edition, 1997.
- [53] Wineberg M and F. Oppacher. *Parallel Problem Solving from Nature III, Vol 866 of Lec. Notes in Computer Science*, chapter A Representation Scheme to Perform Program Induction in a Canonical Genetic Algorithm. Springer-Verlag, 1994.
- [54] P. Martin-Löf. On the Concept of a Random Sequence. *Theory Prob. Appl.*, 11, 1966.

- [55] J. McCarthy. *Automata Studies, Annals of Mathematical Studies*, chapter The Inversion of Functions Defined by Turing Machines, pages 177–181. Number 34. Princeton University Press, 1956.
- [56] T.M. Mitchell. *Machine Learning*. Mc Graw-Hill, 1997.
- [57] J. Muramatsu and F. Kanaya. Distortion Complexity and Rate-Distortion Function. *IEICE Trans. Fundamentals*, E77-A, 1994.
- [58] J. Muramatsu and F. Kanaya. Dual Quantity of the Distortion-Complexity and a Universal Data-Base for Fixed-Rate Data Compression with Distortion. *IEICE Trans. Fundamentals*, E79-A, 1996.
- [59] A.N. Netravali and B. G. Haskell. *Digital Pictures: Representation, Compression, and Standards, 2nd ed.* Plenum Press, 1995.
- [60] A. Nix and M.D. Vose. Modeling genetic algorithms with markov chains. *Annals of Mathematics and Artificial Intelligence*, 5:27–34, 1991.
- [61] P. Nordin and W. Banzhaf. Programmatic compression of images and sound. *Proceedings of First International Conference on genetic Programming*, 1996.
- [62] C. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.
- [63] A. Perez. Extensions of Shannon-McMillan’s Limit Theorem to More General Stochastic Processes. *Trans. Third Prague Conf. on Inform. Theory, Statist. Decision Functions, and Random Processes*, 1964.
- [64] A. Puri and A. Eleftheriadis. *Visual Communication and Image Processing*, chapter MPEG-4: An Object-Based Multimedia Coding Standard. Marcel Dekker, 1998.
- [65] J. Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM J. Res. Devel.*, 20.
- [66] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, 1989.
- [67] G. Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Trans. Neural Networks, special issue on Evolutionary Computing*, 5:96–101, Jan. 1994.
- [68] 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*, 6:3:243–250, 1996.

- [69] M. Schwartz. *Information Transmission Modulation and Noise*. McGraw-Hill, 4th edition, 1990.
- [70] C.E Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27, 1948.
- [71] C.E. Shannon. Communication in presence of noise. *Proc. IRE*, 1949.
- [72] C.E. Shannon. Coding theorems for a discrete source with fidelity criterion. *IRE Nat. Conv. Rec.*, 1959.
- [73] P.C. Shields. Universal Almost Sure Data Compression using Markov Types. *Problems of Control and Information Theory*, 19, 1990.
- [74] G. E. Shilov and B. L. Gurevich. *Integral Measure and Derivative: A Unified Approach*. Prentice-Hall, 1966.
- [75] A. Sinclair. *Algorithms for Random Generation and Counting*. Birkhäuser, 1993.
- [76] R. J. Solomonoff. A Formal Theory of Inductive Inference, Part 1 and Part 2. *Inform. Contr.*, 7, 1964.
- [77] R.J. Solomonoff. *Computational Learning Theory; Proc. 2nd European Conf. Vol. 904 of Lec. Notes Artificial Intelligence*,, chapter The Discovery of Algorithmic Probability, pages 1–18. Springer-Verlag, 1995.
- [78] D. Sow and A. Eleftheriadis. On weighted universal image compression. *submitted to IEEE Trans. on Image Proc.*, December 1999.
- [79] D. Sow and A. Eleftheriadis. Complexity distortion theory. *submitted to IEEE Trans. on Inf. Theory*, June 2000.
- [80] D. Sow and A. Eleftheriadis. Approximation of the Resource Bounded Complexity Distortion Function. *Proceedings 2000 IEEE International Symposium on Information Theory*, pages —, June 25 - July 1 2000.
- [81] D. Sow and A. Eleftheriadis. Complexity Distortion Theory. *Proceedings 1997 IEEE International Symposium on Information Theory*, page 188, June 29 - July 4 1997.
- [82] D. Sow and A. Eleftheriadis. Representing Information with Computational Resource Bounds. *Proceedings 32nd Asilomar Conference on Signals, Systems and Computers*, November 1998.

- [83] D. Sow and A. Eleftheriadis. Algorithmic Representation of Visual Information. *Proceedings 1997 IEEE International Conference on Image Processing*, October 1997.
- [84] R. Sun, J. Dayhoff, and W. Weigand. A population-based search from genetic algorithms through thermodynamic operation. *Technical Research Report, Institute for Systems Research*, T.R.94-78, 1994.
- [85] J. Suzuki. A markov chain analysis on simple genetic algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 25:4:655–659, April 1995.
- [86] J. Suzuki. A further result on the markov chain model of gas and their application to sa-like strategy. *IEEE Trans. on Systems, Man, and Cybernetics*, pages 95–102, Feb. 1998.
- [87] H.L. Van Trees. *Detection Estimation, and Modulation Theory*. John Wiley and Sons, 1968.
- [88] A. Turing. On computable numbers with an application to the Entscheidungsproblem.
- [89] V.A. Uspensky and A. Shen. Relations between varieties of kolmogorov complexities. *Mathematical Systems Theory*, 29, 1996.
- [90] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice-Hall, 1995.
- [91] P. Vitanyi. A discipline of evolutionary programming. *Theoretical Computer Science*, to appear.
- [92] P. Vitanyi and M. Li. Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity. *IEEE Transactions on Information Theory*, 46:446.
- [93] O. Watanabe. *Kolmogorov Complexity and Computational Complexity*. Springer-Verlag, 1992.
- [94] W. Wulf, M. Shaw, and P. N. Hilfinger. *Fundamental Structures of Computer Science*. Addison Wesley, 1981.
- [95] A.D. Wyner, J. Ziv, and A.J. Wyner. On the role of pattern matching in information theory. *IEEE Transactions on Information Theory*, 44.
- [96] ed Y. Fisher. *Fractal Image Compression*. Springer Verlag, 1995.

- [97] A. Sinclair Y. Rabani, Y. Rabinovich. A computational view of population genetics. *Proc, 27th ACM Symp. Theor. Comput.*, pages 83–92, 1995.
- [98] E. Yang and S. Shen. Distortion Program-size Complexity with Respect to a Fidelity Criterion and Rate Distortion Function. *IEEE Transactions on Information Theory*, 39:288–292, 1993.
- [99] E.H. Yang. The Proof of Levin’s Conjecture. *Chinese Sci. Bull.*, 34, 1989.
- [100] ed Y.S. Abu-Mostafa. *Complexity in Information Theory*. Springer-Verlag, 1993.
- [101] Z. Zhang and V.K. Wei. An on-line universal lossy data compression algorithm via continuous codebook refinement-Part I, 1996.
- [102] J. Ziv. Coding for sources with unknown statistics: Part I. Probability of error, 1972.
- [103] J. Ziv. Coding for sources with unknown statistics: Part II. Distortion relative to a fidelity criterion, 1972.
- [104] J. Ziv. Back from Infinity: A Constrained Resources Approach to Information Theory. *Proceedings 1997 IEEE International Symposium on Information Theory*, page 4, 1997.
- [105] A. K. Zvonkin and L. A. Levin. The Complexity of Finite Objects and The Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms. *Russian Math. Surveys*, 25:6:83–124, 1970.

Appendix

A. Recursive Functions

One of the main results of Computability Theory is the equivalence between the formal conception of what an algorithm is and the intuitive notion of an effective procedure. This equivalence is known as the Church-Turing thesis. A formal definition of what an algorithm is can be found in [17] using the concept of recursive function. Following the procedure proposed in [105], we can map finite sequences, belonging to A^* , to \mathcal{N} . Functions on A^* can then be seen as functions on \mathcal{N} . By F^i , we denote a function with i arguments, $F(x_1, \dots, x_i), x_j \in A^*, 1 \leq j \leq i$. When the number of arguments is clear from the context, we drop the superscript. Let C^1, O^n, I_m^n be functions taking the following values: $C^1(x) = x + 1, O^n(x_1, \dots, x_n) = 0, I_m^n(x_1, \dots, x_n) = x_m$.

Definition 1 *The operation of minimalization associates with each total function $f^{n+1}(y, x_1, \dots, x_n)$ the function $h^n(x_1, \dots, x_n)$ whose value is the least value y , if one such exists, for which*

$$f^{n+1}(y, x_1, \dots, x_n) = 0$$

and which is undefined if no such y exists. We write:

$$h^n(x_1, \dots, x_n) = \min_y [f^{n+1}(y, x_1, \dots, x_n)]. \quad (5.1)$$

Definition 2 *The function f^{n+1} is said to originate from the function g^n and from the function h^{n+2} by a primitive recursion if for all natural numbers x_1, \dots, x_n, y we have:*

$$f^{n+1}(x_1, \cdot, x_n, 0) = g(x_1, \cdot, x_n), \quad (5.2)$$

$$f^{n+1}(x_1, \cdot, x_n, y + 1) = h^{n+2}(x_1, \cdot, x_n, y, f^{n+1}(x_1, \cdot, x_n, y)). \quad (5.3)$$

Definition 3 *A function F is called partial recursive if it can be obtained from the functions C^1 , O^n , I_m^n by a finite number of operations of substitution (superposition), of primitive recursion and of minimalization².*

Note that there are many equivalent definitions for recursive functions using different primitive functions spanning the same space of functions. In all cases, an algorithm is just a recursive function that can be expressed by the construction mentioned in definition 3. In 1936 Alan Turing³ introduced the Turing machine (TM), a device able to compute any recursive function. This device is a finite state machine (FSM) with access to a finite number of infinite memory tapes on which the device can store information. This device, although not practical, is widely accepted as the mathematical model for computation because of the one to one mapping between recursive functions and TM's. It is not a practical model because of the infinite size of its memory tape(s). Without memory, the device can only recognize regular languages. In order to accept context sensitive grammars, an infinite amount of memory is required but at any time during the computation, the TM uses a finite amount of memory⁴. The Church-Turing-Thesis stating that anything that can be done algorithmically can be performed by a Turing machine, justifies the use of the TM as a model for computation. It clearly justifies the substitution of the decoder in Shannon's classical communication system, by a Turing machine. It also makes a step forward towards a unification of all coding techniques, from traditional entropy methods which estimate relative frequencies, to novel approaches like fractal coding or even model-based coding techniques. Another advantage is the potential design of computer languages for representations. The use of such languages allows semantical descriptions of information which is necessary to understand the information content. In this case, it seems more natural to use an algorithmic measure of information.

B. Randomness Tests

In this section, we follow the presentation made in [52] on randomness of individual objects, finite and infinite.

²As mentioned in [105], partial recursive functions constructed without the minimalization operation are defined everywhere. Only the operation of minimalization can lead to functions that are not defined everywhere because this operation consisting of successive verification of the validity of equation 5.1 might never stop.

³See [17] for detail description of Turing machines.

⁴It is more accurate to say that the memory tapes are unbounded in size.

Definition 1 Let P be a recursive probability measure on the sample space \mathcal{N} . A total function $\delta : \mathcal{N} \rightarrow \mathcal{N}$ is a P -test (Martin-Löf test for randomness) if:

1. δ is enumerable meaning that the set $V = \{(m, x) : \delta(x) \geq m\}$ is recursively enumerable;
2. $\sum \{P(x) : \delta(x) \geq m, l(x) = n\} \leq 2^{-m}$, for all n .

When applied to a sequence x , if $\delta(x) \geq m$, then from the second condition in the previous definition, with probability 2^{-m} , the sequence x is not random because it belongs to a small set $V_m = \{x : \delta(x) \geq m\}$, for $m \geq 1$ with small measure and we recall that a random sequence must belong to a reasonable majority. Note that it is assumed that P is recursive. It is hard to imagine what practical use it would have to allow P not to be recursive. The critical regions associated with the common statistical tests are present in the form of the sequence $V_1 \supseteq V_2 \supseteq \dots$, with V_m defined above.. Nesting is assured since $\delta(x) \geq m + 1$ implies $\delta(x) \geq m$. Also, note that each V_m is recursively enumerable from item 1 in definition 1. This means that randomness testing at level m gives a certificate of non randomness for a sequence when $\delta(x) \geq m$. When $\delta(x) \leq m$, the result of the test is unknown. The Kolmogorov complexity is an indicator for randomness. Sequences with short descriptions contain a good amount of regularities and should not be classified random. The machinery developed so far is still too weak to identify almost all regular sequences. To do so, we need the notion of universal test for randomness

Definition 2 A universal Martin-Löf test for randomness with respect to measure P (universal P -test) is a test $\delta_0(\cdot | P)$ such that for each P -test δ , there is a constant c , such that for all x , we have $\delta_0(x | P) \geq \delta(x) - c$.

A major result from this theory is the existence of a universal test. Any other randomness test cannot discover more than a constant amount of randomness. This fact also plays a central role in this thesis. It allows us to bridge Kolmogorov complexities with entropies. To make accurate statements about infinite sequences, we need the following definitions.

Definition 3 Let μ be a recursive probability measure on the space of infinite sequences, A^∞ . A total function $\delta : A^\infty \rightarrow \mathcal{N} \cup \{\infty\}$ is a sequential μ -test if:

1. $\delta(w) = \sup_{n \in \mathcal{N}} \{\gamma(x_1^n)\}$, where $\gamma : \mathcal{N} \rightarrow \mathcal{N}$ is a total enumerable function meaning that $V = \{(m, y) : \gamma(y) \geq m\}$ is a recursively enumerable set;
2. $\mu\{x : \delta(x) \geq m\} \leq 2^{-m}$, for each $m \geq 0$.

For a sequential μ -test for randomness δ , if $\delta(x) = \infty$, the sequence fails δ and from item 2 of definition 3, the set of x failing δ has measure 0 respecting the typicality property of random sequences.

Definition 4 Let \mathcal{V} be the set of all sequential μ -tests. An infinite binary sequence x , or the binary represented real number $0.x$, is called μ -random if it passes all sequential μ -tests:

$$x \notin \bigcup_{V \in \mathcal{V}} \bigcap_{m=1}^{\infty} V_m \quad (5.4)$$

where $V_m = \bigcup \{\Gamma_y : (m, y) \in V\}$ and V being defined in definition 3

The sets V_m are also called critical regions. By construction, they are nested: $V_m \supseteq V_{m+1}$, $m = 1, 2, \dots$ From definition 3 we see that $\mu(\bigcap_{m=1}^{\infty} V_m) = 0$ and since \mathcal{V} is countably infinite, $\mu(\bigcup_{V \in \mathcal{V}} \bigcap_{m=1}^{\infty} V_m) = 0$. The sets $\bigcap_{m=1}^{\infty} V_m$ and $\bigcup_{V \in \mathcal{V}} \bigcap_{m=1}^{\infty} V_m$ are respectively called *constructive μ -null set* and *maximal constructive μ -null set*. The definition of the universal sequential μ -test is a direct consequence of this observation.

Definition 5 A universal sequential μ -test f is a sequential μ -test such that for each sequential μ -test δ_i , there is constant $c \geq 0$ and for all infinite sequence x , we have $f(x) \geq \delta_i(x) - c$

Definition 6 Let μ be a probability measure on A^∞ and let $\delta_0(\cdot | \mu)$ be a universal sequential μ -test. An infinite sequence x is μ -random in the sense of Martin-Löf, if $\delta_0(\cdot | \mu) < \infty$.

As mentioned earlier, the delimitation between random and non random sequences is clearer for the infinite case. We simply need a universal test and verify if the value of the test is finite or not. There are other ways to test sequences for randomness. The following type of test is used extensively to establish links between Shannon's entropy and the Kolmogorov complexity.

Definition 7 Let f be a unit integrable function over A^∞ with respect to μ assumed recursive. A function δ is an integral- μ -test iff $\delta(\omega) = \log f(\omega)$. It is a universal integral μ -test if it additively dominates all integral μ -tests.

Integral tests are equivalent to sequential tests. If f is enumerable unit integrable over A^∞ with respect to μ , then it can be shown that $\log f(\cdot)$ is a sequential test. Also, if δ is a sequential μ -test, then the function f defined by $\log f(\omega) = \delta(\omega) - 2 \log \delta(\omega) - c$ is an enumerable unit integrable function⁵. Of particular interest is the following integral test which provides a link between Shannon's entropy and Kolmogorov complexity.

⁵See [52] pp 287, lemma 4.5.7

Lemma 1 *Let μ be a recursive measure. Let $\omega \in A^\infty$. The function*

$$\rho_0(\omega \mid \mu) = \sup_{\omega \in \Gamma_x} \{-C(x \mid \mu) - \log_2 \mu(x)\} \quad (5.5)$$

is a universal integral μ -test⁶.

Proof: See [52]. □

Just like in the case of finite sequences, this lemma shows that the difference between the Kolmogorov complexity and the Shannon-Fano code length is an indicator of randomness. This result is at the heart of all equivalence results derived in chapter 3 and 4.

C. Markov Types

The Markov k -type is defined by sliding a window of length $k + 1$ along x_1^n and counting frequencies. These relative frequencies are then used to define empirical transition probabilities.

Definition 3 *Let \bar{x} be the following infinite sequence defined by:*

$$\bar{x}_{tn+i} = x_i, \quad i = 1, 2, \dots, n; \quad t = 0, 1, 2, \dots \quad (5.6)$$

For each integer $0 \leq k < n$ and for each $a_1^k \in A^k$, define

$$\hat{p}_k(a_1^k) = \lim_{L \rightarrow \infty} \frac{1}{L} \mid \{i : \bar{x}_{i+1}^{i+k} = a_1^k, 0 \leq i < L\} \mid \quad (5.7)$$

The periodicity of \bar{x} implies that

$$\hat{p}_k(a_1^k) = \frac{1}{n} \mid \{i : \bar{x}_{i+1}^{i+k} = a_1^k, 0 \leq i < n\} \mid \quad (5.8)$$

and

$$\hat{p}_{k-1}(a_1^{k-1}) = \sum_{a_k \in A_0} \hat{p}_k(a_1^k), \quad a_1^{k-1} \in A^{k-1} \quad (5.9)$$

⁶The x 's in this definition are finite sequences. They represent prefixes of ω . The value of the test is simply the supremum of the difference between the complexity of those prefix and the logarithm of their measure. This difference is simply the randomness deficiency for finite sequences.

Definition 4 The Markov k -type of x_1^n is the Markov measure $\hat{\mu}^k$ with state space A^k , stationary probabilities $\hat{p}_k(a_1^k)$, and transition probabilities $p(\cdot | \cdot)$ defined by:

$$p(a_{k+1} | a_1^k) = \frac{\hat{p}_{k+1}(a_1^{k+1})}{\hat{p}_k(a_1^k)} \quad (5.10)$$

The entropy \hat{H}^k of the k -type $\hat{\mu}^k$ is given by:

$$\hat{H}^k = - \sum_{a_1^{k+1}} \hat{p}_{k+1}(a_1^{k+1}) \log_2 \frac{\hat{p}_{k+1}(a_1^{k+1})}{\hat{p}_k(a_1^k)} \quad (5.11)$$

Definition 5 The type class $T_k(x_1^n)$ is defined as the set of all sequences of length n with Markov type equal to the Markov type of x_1^n .