REWIND

REVerse engineering of audio-VIsual coNtent Data

Grant Agreement No. 268478

Deliverable D2.2
Second report on mathematical models

Lead partner for this deliverable: UVIGO
Version: 2.0

Dissemination level: Public

January 30, 2013

Contents

Acronyms and Abbreviations

AC	Alternating Current
AD	Adversary
AD_{FA}	Anti-Forensics Player
AD_{ST}	Steganalyzer
A/D	Analog-to-Digital
AF	Anti-Forensics
AR	Autoregressive
AVC	Advanced Video Coding
CFA	Color Filter Array
D/A	Digital-to-Analaog
DCT	Discrete Cosine Transform
DMS	Discrete Memoryless Source
EM	Expectation-Maximization
FA	Forensics Analyst
FRI	Finite Rate of Innovation
GCD	Greatest Common Divisor
GGD	Generalized Gaussian Distribution
GMM	Gaussian Mixture Model
HEVC	High Efficiency Video Coding
HP	High Pass
IDCT	Inverse Discrete Cosine Transform
i.i.d.	Independent and Identically Distributed
JPEG	Joint Photographic Experts Group
KL	Kullback-Leibler
LLL	Lenstra-Lenstra-Lovasz
LP	Low Pass
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MMF	MultiMedia Forensics
MOMS	Maximal-Order-Minimal-Support
MPEG	Moving Picture Experts Group
pdf	Probability Density Function

pmf	Probability Mass Function
QF	Quality Factor
SI	Source Identification
ST	Steganographer
TI	Transform Identification
TIFF	Tagged Image File Format
UC	Use Case
UCID	Uncompressed Colour Image Database
WP	Work Package

Introduction

This deliverable summarizes the work performed in months 13 to 21 in the scope of WP2. In this period the originally planned schedule was followed, as it is reflected in the obtained results exposed in this report. Nevertheless, special attention has been also paid to the comments made by the project reviewers. Specifically, two short chapters have been included for dealing with the similarities and differences between steganography and multimedia forensics (Chapter 1), and explaining three examples of the synergies between WP2, and WP3 and WP4 (Chapter 4). Besides these two short chapters, the main results obtained in this period of 9 months are split in a chapter dealing with theoretical general topics (Chapter 2), and other chapter studying the modeling of operator chain (Chapter 3).
Within Chapter 2, results on identification source identification game with training sequences have been included. On the other hand Chapter 3 includes results on

- JPEG quantization followed by full-frame filtering
- Interpolation estimation
- Transform coder identification based on noiseless lattice estimation
- Reacquisition modeling
- Demosaicking localization

Chapter 1

Similarities and differences between forensics and steganalysis

In the field of multimedia security one can distinguish different problems; probably watermarking, steganography, and lately forensics, are those that have received more attention. Hopefully, looking at the evolution of the other problems of multimedia security will allow us to learn some lessons about good directions to be followed by multimedia forensics. Due to the shared statistical undistinguishability constraint, it seems that forensics is more closely related to steganalysis. Therefore, the target of this chapter is to study the main parallelisms, and corresponding similarities and differences between multimedia forensics and steganography; specifically, we will focus on the similar role of forensics analyst and steganalyzer, and also on the similar target of counter-forensics player and steganographer. Based on this discussion, we will summarize what are, in our opinion, the main lessons to be learned. Finally, links to previous works in the literature will be pointed out.
For the sake of notational simplicity, we will use MMF for denoting MultiMedia Forensics, AF for Anti-Forensics, FA for Forensics Analyst, $\mathrm{AD_{FA}}$ for the anti-forensics player, ST for the steganographer, and $\mathrm{AD_{ST}}$ for the steganalyzer.

1.1 FA and $\mathrm{AD_{ST}}$

From the point of view of the FA we can consider MMF as an estimation problem (if we want to estimate the processing parameters), or a binary hypothesis problem (if we want to decide if a given content was modified or not, or in an alternative way, if it comes from source A or source B). ST counterparts to these two scenarios could be also considered, being the binary hypothesis version the classical steganographic problem, and the estimation version the so-called "quantitative steganalysis".
Therefore, the $\mathrm{AD_{ST}}$'s formal description of the steganalysis problem is nothing but a particular example of the FA's MMF formal description; this is the case for both the estimation and binary versions of the two problems. Some considerations must be taken into account :

- Kind of processing to be considered: We can consider a set of possible processing/modifications as broad as we want, although in general some constraints are imposed in that set in order to have a problem which is feasibly resolved. For example, we can focus our attention on the case of JPEG quantization+spatial filtering, cut+paste+footprint removal, information embedding+footprint removal (=steganography; in this case, the FA is indeed an $\mathrm{AD_{ST}}$).

Even smaller sets could be considered, e.g. MPSteg detection; as far as we reduce the set (or class) of possible processing, we have a more targeted scheme (following steganalysis naming). Therefore, from that point of view, **what distinguishes steganography as a particular case of MMF is the analyzed set of feasible operators**.

- If the null hypothesis does not only include the "no processing" case, but also some "light processings" (meaning that the semantics of the content are not modified) are included, then an additional multimedia security problem, namely authentication, could be also included in this framework.[1]
- Summarizing, according to this approach the $\mathrm{AD_{ST}}$ is nothing but a FA using a particular definition of the alternative hypothesis H_1, where the analyzed set of feasible operators only contains "stego-processing."

Note that in the case of universal steganalyzers, the alternative hypothesis H_1 is not well defined, since the embedding algorithm is not known. Therefore, one-class classifiers (or composite hypothesis testing) must be resorted. A similar conclusion can be derived for MMF.

1.2 $\mathbf{AD_{FA}}$ and ST

Once the similarity between FA and $\mathrm{AD_{ST}}$ has been established, one wonders if this can be extended to the similarity between $\mathrm{AD_{FA}}$ and ST. Nevertheless, in this case there is an obvious difference between the target of these two players: while in AF the goal depends on the kind of **processing** the $\mathrm{AD_{FA}}$ wants to **hide** (histogram stretching, compression, resizing, etc.), in steganography the goal is **transmitting secret information**. Deep changes in the proposed approaches are implied. Despite this difference on the target function, both $\mathrm{AD_{FA}}$ and ST share the same constraints:

- Statistical undetectability.
- Perceptual distortion. Typically, the perceptual distortion used in steganographic applications depends on the original signal (i.e., reference-based distortion measures are used). Nevertheless, in MMF the definition of such kind of measure is more involved; for example, for cut and paste attacks, one wonders how a reference-based distortion measure can be defined. However, since the original signal will be typically not accessible to the receiver, the use of reference-based distortion measures can be criticized even in the steganographic framework. If blind (no-reference) distortion measures were used (e.g., [3]), then we can define the perceptual distortion for both problems in much the same way.

  From an application point of view, in both cases one must face the constrain of the processed images looking natural to a human observer.
- The fact of the steganography problem having more players (legitimate decoder and possibly the active warden) is "just" reflected on the definition of the target function the ST tries to maximize.
- The ST strategy is defined by the embedding function, while the $\mathrm{AD_{FA}}$ strategy is the manipulation function. In both cases the feasible set of strategies is defined as that set that contains all those functions that modify the original content while verifying the previous contraints. This

[1] The reader who is not accustomed to the basics on detection and/or estimation theory is referred to [1, 2].

set is the same for both problems. The only change is the embedding/manipulation function which is chosen in each case, since the target function is different in both cases.

Nevertheless, the impact of this difference is not trivial. Indeed, this yields the overall goal to be different, as well as the kind of used techniques to differ:

- In stego an integrated approach is often used, in which the embedding function is chosen to verify the statistical imperceptibility constraint.
- On the other hand, in MMF the attacker often adopts a post processing approach: first, it applies the intended modification; then, it tries to remove traces without "spoiling" the result of the modification. This two-steps strategy should be used as long as the processing and the counterforensics steps are not carried out in different time instants, or by different players. Other scenario where it makes sense is that where several processing steps are applied, and the footprints of all of them are deleted in a single final step.

Nevertheless, probably there is not any fundamental reason for working in this case. For example, one could introduce the watermark in the content (without caring about the detectability constraint, e.g., using regular watermark embedding techniques) and then apply some post-processing for removing the traces (reducing detectability). Alternatively, the MMF attacker could devise the content modification taking in mind the detectability constraint. In general, it seems that the two steps procedure (first doing the work and then solving the problems it entails) will behave worse than the one step approach.

Consequently, it seems that the only reason for applying two-steps attacks is the complexity reduction; being the application scenario requirements (including complexity ones) for $\mathrm{AD_{FA}}$ and ST completely different, is not surprising that different approaches to both problems have been developed.

Summarizing, one can state that $\mathrm{AD_{FA}}$ and ST techniques are very different. This was not the case with the MMF FA and $\mathrm{AD_{ST}}$, which basically share the same techniques (classifiers, decision making, hypothesis testing, etc.).

1.3 Similarities

S1. The task of the $\mathrm{AD_{ST}}$ can be seen as a particular MMF task: detect the traces introduced by the ST. Indeed, its task can be interpreted as completely equivalent to detecting the traces left by any other processing tool. This makes steganalysis a particular instance of MMF.

S2. Quantitative steganalysis is a particular case of MMF, assuming that the FA does not only want to detect processing traces but also to estimate some of the parameters characterizing the processing.

S3. The constraints of the ST and the $\mathrm{AD_{FA}}$ are also virtually identical, since both of them want to modify a media (for different purposes) without leaving any kind (visible or statistical) of trace. In this sense, steganography and anti-forensics are optimization problems that share the same feasible set but differ in their target functions.

S4. Cachin's perfect steganography [4] is similar to statistics preserving AF.

S5. Steganalysis and MMF share the difficulty deriving from the lack of a good statistical model describing natural images.

1.4 Differences

D1. Despite similarity number 3, steganography and AF are quite different since their ultimate goal is very different: in AF the goal depends on the kind of processing the AD_{FA} wants to hide (histogram stretching, compression, resizing, etc.); in steganography the goal is transmitting secret information.

D2. In Steganography we have 3 players, while in MMF we usually have 2.

D3. MMF is a much wider field, encompassing issues extraneous to steganography and steganalysis: this is the case, for instance, of semantic-level MMF, like some works by Farid's group (e.g., [5]). In these cases MMF resembles more computer vision than stego.

D4. The batch steganography concept does not seem to apply to MMF.

D5. Similarities exist only with steganography by cover modification, while steganography by cover selection has nothing to share with MMF.

D6. From an application perspective stego and MMF are completely different, leading to rather different constraints (if not goals), e.g., amount of images to be analyzed, target error probabilities, etc.

1.5 Lessons to be learned

1. In the past years steganalysis has passed from classifiers based on few features, to a moderate amount of features until the very large number of features characterizing the most powerful schemes developed recently. One wonders if MMF should follow the same path.

2. In early days, steganography was focused on statistical indistinguishability, i.e., the ST was aiming at keeping some statistical quantities untouched. More recently, it seems that minimizing a properly defined distortion measure is a better choice. It is possible that AF will go through the same path.

3. Calibration has played a crucial role in steganalysis. Similar techniques can be borrowed for AF.

4. Steganalysis has switched from first order to higher order statistical analysis (and finally to classifiers with a large number of features). Again, it is possible for MMF to follow the same route.

5. Steganographers typically use one-step strategies, while most of AD_{FA}'s approaches are based on two steps. Since the two steps approach is in general suboptimal, one wonders if the one-step strategies should be also adopted in counterforensics.

1.6 Links to previous works in the literature

The similarities and differences between steganography and counter-forensics are also studied in [6]. In that work the authors point out that *both steganography and counter-forensics try to hide the very fact of a class change, and their success can be measured by the Kullback-Leibler divergence.* Nevertheless, they also defend that *steganography differs from counter-forensics in the amount and*

source of information to hide. These considerations are developed in our previous analysis, where they are stated in terms of the target function and the optimization search space constraints.
Similarly, the connections between steganalysis and the forensics analyst, as well as the difference between one-step and two-steps attacks (typically used in steganography and counter-forensics, respectively) are also mentioned.

Chapter 2

General theory

2.1 Source identification game with training sequences

In the attempt to provide a mathematical background to multimedia forensics, we introduce the source identification game with training data. The game models a scenario in which a forensic analyst has to decide whether a test sequence has been drawn from a source X or not. In turn, the adversary takes a sequence generated by a different source and modifies it in such a way to induce a classification error. The source X is known only through one or more training sequences. We derive the asymptotic Nash equilibrium of the game under the assumption that the analyst relies only on first order statistics of the test sequence. A geometric interpretation of the result is given together with a comparison with a similar version of the game with known sources. The comparison between the two versions of the games gives interesting insights into the differences and similarities of the two games.[1]

2.1.1 Introduction

Understanding the fundamental limits of multimedia forensics in an adversarial environment is a pressing need to avoid the proliferation of forensic and anti-forensic tools each focused on countering a specific action of the adversary but prone to yet another class of attacks and counter-attacks. The most natural solution to avoid entering this never-ending loop is to cast the forensic problem into a game-theoretic framework and look for the optimum strategies the players of the game (usually a forensic analyst and an adversary) should adopt. Some early attempts in this direction can be found in [9] and [10]. In [9], the authors introduce a game-theoretic framework to evaluate the effectiveness of a given attacking strategy and derive the optimal countermeasures. In [9] the attacker's strategy is fixed and the game-theoretic framework is used only to determine the optimal parameters of the forensic analysis and the attack. A more general approach is adopted in [10], where the source identification game with known statistics, namely the SI_{ks} game, is introduced. According to the framework defined in [10], given a discrete memoryless source (DMS) X with known statistics P_X, it is the goal of the Forensic Analyst (FA) to decide whether a test sequence x^n has been drawn from X or not. In doing so, he has to ensure that the false positive probability, i.e. the probability of deciding that the test sequence has not been generated by X when it actually was, stays below a predefined maximum value. The goal of the adversary (AD) is to take a sequence generated from a different and independent source $Y \simeq P_Y$ and modify it so to let the FA think that the modified

[1] The reader who is not accustomed to the basics on game theory is referred to [7, 8].

sequence has been generated by X. In doing so the AD must satisfy a distortion constraint, i.e. the distance between the original and the modified sequence must be lower than a threshold. The payoff of the AD is the false negative error probability, i.e. the probability that the FA classifies a sequence drawn from Y and further modified by the AD as a sequence drawn from X. The opposite payoff applies to the FA, thus qualifying the SI_{ks} as a zero-sum, competitive game [11]. Under the further assumption that the FA relies only on first order statistics (limited resources assumption) for his analysis and that the sources X and Y are memoryless, the asymptotic Nash equilibrium of the game can be found [10, 12], thus defining the optimum strategies for the FA and the AD when the length of the test sequence tends to infinity. A problem with the analysis carried out in [10] is the assumption that the FA and the AD know the probability mass function (pmf) of the source X. This is not the case in many practical scenarios where sources are known only through one or more training sequences. It is the goal of this chapter to reformulate the analysis carried out in [10] to address this new more realistic version of the game. As a main result, we derive the asymptotic Nash equilibrium of the new game, hereafter referred to as the SI_{tr} game, under the same limited resources assumptions used in [10]. In doing so we will discover that the optimal strategies for the FA and the AD deviate from those of the SI_{ks} game. In addition, at least in the case that the training sequences available to the FA and the AD coincide, we can show that passing from the SI_{ks} to the SI_{tr} version of the game is to the AD's advantage.
The description of the work is organized as follows. In section 2.1.2 we introduce the notation that will be used throughout the description. In section 2.1.3, we give a rigorous definition of the source identification with training data game. In section 2.1.4, we derive the asymptotic Nash equilibrium of the game. In section 2.1.5, we compare the results obtained in this work with those referring to source identification with known sources. Section 2.1.6 concludes the description with some perspective for future research.

2.1.2 Notation

In the rest of this chapter we will use capital letters to indicate discrete memoryless sources (e.g. X). Sequences of length n drawn from a source will be indicated with the corresponding lowercase letters (e.g. x^n). In the same way, we will indicate with x_i, $i = 1, n$ the $i-$th element of a sequence x^n. The alphabet of an information source will be indicated by the corresponding calligraphic capital letter (e.g. $\mathcal{X}$). Calligraphic letters will also be used to indicate classes of information sources ($\mathcal{C}$). The pmf of a discrete memoryless source X will be denoted by P_X. With a slight abuse of notation, the same symbol will be used to indicate the probability measure ruling the emission of sequences from X, so we will use the expressions $P_X(a)$ and $P_X(x^n)$ to indicate, respectively, the probability of symbol $a \in \mathcal{X}$ and the probability that the source X emits the sequence x^n. Given an event A (be it a subset of $\mathcal{X}$ or $\mathcal{X}^n$), we will use the notation $P_X(A)$ to indicate the probability of the event A under the probability measure P_X.
Our analysis relies heavily on the concepts of type and type class defined as follows (see [13] and [14] for more details). Let x^n be a sequence with elements belonging to an alphabet $\mathcal{X}$. The type P_{x^n} of x^n is the empirical pmf induced by the sequence x^n, i.e. $\forall a \in \mathcal{X}, P_{x^n}(a) = \frac{1}{n}\sum_{i=1}^{n}\delta(x_i, a)$. In the following we indicate with $\mathcal{P}_n$ the set of types with denominator n, i.e. the set of types induced by sequences of length n. Given $P \in \mathcal{P}_n$, we indicate with $T(P)$ the type class of P, i.e. the set of all the sequences in $\mathcal{X}^n$ having type P.
The Kullback-Leibler (KL) divergence between two distributions P and Q on the same finite alphabet $\mathcal{X}$ is defined as:

$$\mathcal{D}(P||Q) = \sum_{a\in\mathcal{X}} P(a)\log\frac{P(a)}{Q(a)}, \tag{2.1}$$

where, as usual, $0 \log 0 = 0$ and $p \log p/0 = \infty$ if $p > 0$. Empirical distributions can be used to calculate empirical information theoretic quantities, like, for instance, the empirical divergence between two sequences $D(P_{x^n}||P_{y^n})$.
As we said, the goal of this study is to cast the source identification problem into a game-theoretic framework, wherein identification is seen as a two-player, strategic, zero-sum game. In rigorous terms, a game is defined as a 4-tuple $G(\mathcal{S}_1, \mathcal{S}_2, u_1, u_2)$, where $\mathcal{S}_1 = \{s_{1,1} \dots s_{1,n_1}\}$ and $\mathcal{S}_2 = \{s_{2,1} \dots s_{2,n_2}\}$ are the set of strategies (actions) the first and the second player can choose from, and $u_l(s_{1,i}, s_{2,j}), l = 1, 2$ is the payoff of the game for player l, when the first player chooses the strategy $s_{1,i}$ and the second chooses $s_{2,j}$. A pair of strategies $s_{1,i}$ and $s_{2,j}$ is called a profile. In a zero-sum competitive game, the two payoff functions are strictly related to each other since for any profile we have $u_1(s_{1,i}, s_{2,j}) + u_2(s_{1,i}, s_{2,j}) = 0$. A zero-sum game, then reduces to a triplet $G(\mathcal{S}_1, \mathcal{S}_2, u)$, where we have assumed $u = u_1 = -u_2$. Note that in strategic games the players choose their strategies before starting the game so that they have no hints about the strategy actually chosen by the other player. We say that a profile (s_{1,i^*}, s_{2,j^*}) represents a Nash equilibrium if [15, 11]:

$$\begin{array}{ll} u_1((s_{1,i^*}, s_{2,j^*})) \geq u_1((s_{1,i}, s_{2,j^*})) & \forall s_{1,i} \in \mathcal{S}_1 \\ u_2((s_{1,i^*}, s_{2,j^*})) \geq u_2((s_{1,i^*}, s_{2,j})) & \forall s_{2,j} \in \mathcal{S}_2, \end{array} \tag{2.2}$$

where for a zero-sum game $-u_2 = u_1 = u$.

2.1.3 Source identification with training data

Let $\mathcal{C}$ be the class of discrete memoryless sources with alphabet $\mathcal{X}$, and let $X \simeq P_X$ be a source in $\mathcal{C}$. Given a test sequence x^n, the goal of the Forensic Analyst (FA) is to decide whether x^n was drawn from X or not[2]. As opposed to the source identification game with known sources [10], here we assume that the FA does not know P_X, and that he has to base his decision by relying on the knowledge of a training sequence t_{FA}^N drawn from X. On his side, the Adversary (AD) takes a sequence y^n emitted by another source $Y \simeq P_Y$ still belonging to $\mathcal{C}$ and tries to modify it in such a way that the FA thinks that the modified sequence was generated by X. In doing so the AD must satisfy a distortion constraint stating that the distance between the modified sequence, say z^n, and y^n must be lower than a predefined threshold. As the FA, the AD knows P_X through a training sequence t_{AD}^K, that in general may be different than t_{FA}^N. We assume that t_{FA}^N, t_{AD}^K, x^n and y^n are generated independently. With regard to P_Y, we could also assume that it is known through two training sequences, one available to the FA and one to the AD, however we will see that - at least to study the asymptotic behavior of the game - such an assumption is not necessary, and hence we take the simplifying assumption that P_Y is known neither to the FA nor to the AD. As in [10], we define the game by casting the identification problem into a hypothesis decision framework. Let then H_0 be the hypothesis that the test sequence has been generated by X (i.e. the same source that generated t_{FA}^N) and let Λ_0 be the acceptance region for H_0 (similarly we indicate with $\Lambda_1 = \Lambda_0^c$ the rejection region for H_0). We have the following:

Definition 1. *The $SI_{tr,a}(\mathcal{S}_{FA}, \mathcal{S}_{AD}, u)$ game is a zero-sum, strategic, game played by the FA and the AD, defined by the following strategies and payoff.*

- *The set of strategies the FA can choose from is the set of acceptance regions for H_0 for which the maximum false positive probability across all possible $P_X \in \mathcal{C}$ is lower than a certain threshold:*

$$\mathcal{S}_{FA} = \{\Lambda_0 : \max_{P_X \in \mathcal{C}} P_X\{(x^n, t_{FA}^N) \notin \Lambda_0\} \leq P_{fp}\}, \tag{2.3}$$

[2]With a slight abuse of notation we use the symbol x^n to indicate the test sequence even if strictly speaking it is not known whether the test sequence originated from X or Y.

where P_{fp} is a prescribed maximum false positive probability, and where $P_X\{(x^n, t_{FA}^N) \notin \Lambda_0\}$ indicates the probability that two independent sequences generated by X do not belong to Λ_0. Note that the acceptance region is defined as a union of pairs of sequences, and hence $\Lambda_0 \subset R^n \times R^N$.

- *The set of strategies the AD can choose from is formed by all the functions that map a sequence $y^n \in \mathcal{X}^n$ into a new sequence $z^n \in \mathcal{X}^n$ subject to a distortion constraint:*

$$\mathcal{S}_{AD} = \{f(y^n, t_{AD}^K) : d(y^n, f(y^n, t_{AD}^K)) \leq nD\}, \tag{2.4}$$

where $d(\cdot,\cdot)$ is a proper distance function and D is the maximum allowed per-letter distortion. Note that the function $f(\cdot)$ depends on t_{AD}^K, since when performing his attack the AD will exploit the knowledge of the training sequence.

- *The payoff function is defined in terms of the false negative error probability (P_{fn}), namely:*

$$u(\Lambda_0, f) = -P_{fn} = -\sum_{\substack{t_{FA}^N \in \mathcal{X}^N,\ t_{AD}^K \in \mathcal{X}^K \\ y^n : (f(y^n, t_{AD}^K), t_{FA}^N) \in \Lambda_0}} P_Y(y^n) P_X(t_{FA}^N) P_X(t_{AD}^K), \tag{2.5}$$

where the error probability is averaged across all possible y^n and training sequences and where we have exploited the independence of y^n, t_{FA}^N and t_{AD}^K.

Some explanations are in order with regard to the definition of the payoff function. As a matter of fact, the expression in (2.5) looks problematic, since its evaluation requires that the pmf's P_X and P_Y are known, however this is not the case in our scenario since we have assumed that P_X is known only through t_{FA}^N and t_{AD}^K, and that P_Y is not known at all. As a consequence it may seem that the players of the game are not able to compute the payoff associated to a given profile and hence have no arguments upon which they can base their choice. While this is indeed a problem in a generic setup, we will show later on that asymptotically (when n, N and K tend to infinity) the optimum strategies of the FA and the AD are uniformly optimum across all P_X and P_Y and hence the ignorance of P_X and P_Y does not represent a problem. One may wonder why we did not define the payoff under a worst case assumption (from FA's perspective) on P_X and/or P_Y. The reason is that doing so would result in a meaningless game. In fact, given that X and Y are drawn from the same class of sources $\mathcal{C}$, the worst case would always correspond to the trivial case $X = Y$ for which no meaningful forensic analysis is possible[3].
Slightly different versions of the game are obtained by assuming a different relationship between the training sequences. In certain cases we may assume that the FA has a better access to the source X than the AD. In [16], for example, the availability of a number of pictures taken from a camera X and made publicly available is exploited by the AD to take an image produced by a camera Y and modify it in such a way that the fake picture looks as if it were taken by X. The FA, exploits his better access to the source X and the knowledge of the images potentially available to the AD to distinguish the images truly generated by X and the fake images produced by the AD. In our framework, such a scenario can be quite faithfully modeled by assuming that the sequence t_{AD}^K is a subsequence of t_{FA}^N, leading to the following definition.

Definition 2. *The $SI_{tr,b}(\mathcal{S}_{FA}, \mathcal{S}_{AD}, u)$ game is a zero-sum, strategic, game played by the FA and the AD, defined as the $SI_{tr,a}$ game with the only difference that $t_{AD}^K = (t_{FA,l+1}, t_{FA,l+2} \dots t_{FA,l+K})$ with l and K known to the FA.*

[3]Alternatively, we could assume that X and Y belong to two disjoint source classes $\mathcal{C}_X$ and $\mathcal{C}_Y$. We leave this analysis for further research.

Yet another version of the game is obtained by assuming that the training sequence available to the AD corresponds to that available to the FA.

Definition 3. *The $SI_{tr,c}(\mathcal{S}_{FA}, \mathcal{S}_{AD}, u)$ game is a zero-sum, strategic, game played by the FA and the AD, defined as the $SI_{tr,a}$ game with the only difference that $K = N$ and $t_{AD}^K = t_{FA}^N$ (simply indicated as t^N in the following). The set of strategies of the FA and the AD are the same as in the $SI_{tr,a}$ game.*

In the rest of this description we will focus on the $SI_{tr,c}$ game, leaving the other versions for future research.

2.1.4 Asymptotic equilibrium for the $SI_{tr,c}$ game with limited-resources

Studying the existence of an equilibrium point for the $SI_{tr,c}$ game is a prohibitive task due to the difficulty of determining the optimum strategies for the FA and the AD, hence we consider a simplified version of the game in which the FA can only base his decision on a limited set of statistics computed on the test and training sequences. Specifically, we require that the FA relies only on the relative frequencies with which the symbols in $\mathcal{X}$ appear in x^n and t^N, i.e. P_{x^n} and P_{t^N}. Note that P_{x^n} and P_{t^N} are not sufficient statistics for the FA, since even if Y is also a memoryless source, the AD could introduce some memory within the sequence as a result of the application of $f(\cdot)$. In the same way it could introduce some dependencies between the attacked sequence z^n and t^N. It is then necessary to treat the assumption that the FA relies only on P_{x^n} and P_{t^N} as an explicit - additional - requirement. As in [10], we call this version of the game "source identification with limited-resources", and we refer to it as the $SI_{tr,*}^{lr}$ game. As a consequence of the limited resource assumption, Λ_0 can only be a union of cartesian products of pairs of type classes, i.e. if the pair of sequences (x^n, t^N) belongs to Λ_0, then any pair of sequences belonging to the cartesian product $T(P_{x^n}) \times T(P_{t^N})$ will be contained in Λ_0. Since a type class is univocally defined by the empirical pmf of the sequences contained in it, we can redefine the acceptance region Λ_0 as a union of pairs of types (P, Q) with $P \in \mathcal{P}_n$ and $Q \in \mathcal{P}_N$. In the following, we will use the two interpretations of Λ_0 (as a set of sequences or a set of types) interchangeably, the exact meaning being always clearly recoverable from the context. We are interested in studying the asymptotic behavior of the game when n and N tends to infinity. To avoid the necessity to consider two limits with n and N tending to infinity independently, we decided to express N as a function of n, and study what happens when n tends to infinity. With the above ideas in mind, we can state the following:

Definition 4. *The $SI_{tr,c}^{lr}(\mathcal{S}_{FA}, \mathcal{S}_{AD}, u)$ game is a zero-sum, strategic, game played by the FA and the AD, defined by the following strategies and payoff:*

$$\mathcal{S}_{FA} = \{\Lambda_0 \subset \mathcal{P}_n \times \mathcal{P}_{N(n)} : \max_{P_X \in \mathcal{C}} P_X\{(x^n, t^{N(n)}) \notin \Lambda_0\} \leq 2^{-\lambda n}\}, \tag{2.6}$$

$$\mathcal{S}_{AD} = \{f(y^n, t^{N(n)}) : d(y^n, f(y^n, t^{N(n)})) \leq nD\}, \tag{2.7}$$

$$u(\Lambda_0, f) = -P_{fn} = -\sum_{\substack{t^{N(n)} \in \mathcal{X}^{N(n)} \\ y^n : (f(y^n, t^{N(n)}), t^{N(n)}) \in \Lambda_0}} P_Y(y^n) P_X(t^{N(n)}). \tag{2.8}$$

Note that we ask that the false positive error probability decay exponentially fast with n, thus opening the way to the asymptotic solution of the game. Similar definitions obviously hold for the a and b versions of the game.

2.1.4.1 Optimum FA strategy

We start the study of the asymptotic equilibrium point of the $SI^{lr}_{tr,c}$ game determining the optimum decision region for the FA. In doing so we will use an analysis similar to that carried out in [17] to analyze a statistical problem with observed statistics (the main difference between our analysis and [17] is the presence of the AD, i.e. the game-theoretic nature of our problem). The derivation of the optimum strategy for the FA passes through the definition of the generalized log-likelihood ratio function $h(x^n, t^{N(n)})$. Given the test and training sequences x^n and $t^{N(n)}$, we define the generalized log-likelihood ratio function as ([17, 18])[4]:

$$h(x^n, t^N) = \mathcal{D}(P_{x^n}||P_{r^{N+n}}) + \frac{N}{n}\mathcal{D}(P_{t^N}||P_{r^{N+n}}), \tag{2.9}$$

where $P_{r^{N+n}}$ indicates the empirical pmf of the sequence r^{N+n}, obtained by concatenating t^N and x^n, i.e.

$$r^{N+n} = \begin{cases} t_i & i \leq N \\ x_{i-N} & N < i \leq n+N \end{cases} . \tag{2.10}$$

Observing that $h(x^n, t^N)$ depends on the test and the training sequences only through their empirical pmf, we can also use the notation $h(P_{x^n}, P_{t^N})$. The derivation of the Nash equilibrium for the $SI^{lr}_{tr,c}$ game passes through the following lemmas.

Lemma 1. *For any P_X we have:*

$$\begin{aligned} n\mathcal{D}(P_{x^n}||P_{r^{n+N}}) + N\mathcal{D}(P_{t^N}||P_{r^{n+N}}) \leq \\ n\mathcal{D}(P_{x^n}||P_X) + N\mathcal{D}(P_{t^N}||P_X), \end{aligned} \tag{2.11}$$

with equality holding only if $P_X = P_{r^{n+N}}$.

The proof of Lemma 1 is given in the appendix.

Lemma 2. *Let Λ_0^* be defined as follows:*

$$\Lambda_0^* = \left\{ (P_{x^n}, P_{t^N}) : h(P_{x^n}, P_{t^N}) < \lambda - |\mathcal{X}| \frac{\log(n+1)(N+1)}{n} \right\} \tag{2.12}$$

with

$$\lim_{n\to\infty} \frac{log(N(n)+1)}{n} = 0, \tag{2.13}$$

and let Λ_1^* be the corresponding rejection region. Then:

1. $\max_{P_X} P_X\{(x^n, t^{N(n)}) \notin \Lambda_0^*\} \leq 2^{-n(\lambda - \delta_n)}$, *with $\delta_n \to 0$ for $n \to \infty$,*
2. $\forall \Lambda_0 \in \mathcal{S}_{FA}$ *defined as in (2.6), we have $\Lambda_1 \subseteq \Lambda_1^*$.*

Proof. Being Λ_0^* (and Λ_1^*) a union of pairs of types (or, equivalently, unions of cartesian products of type classes), we have:

$$\begin{aligned} \max_{P_X} P_{fp} &= \max_{P_X \in \mathcal{C}} P_X\{(x^n, t^N) \notin \Lambda_0^*\} \\ &= \max_{P_X \in \mathcal{C}} \sum_{(x^n, t^N) \in \Lambda_1^*} P_X(x^n, t^N) \\ &= \max_{P_X \in \mathcal{C}} \sum_{(P_{x^n}, P_{t^N}) \in \Lambda_1^*} P_X(T(P_{x^n}) \times T(P_{t^N})). \end{aligned} \tag{2.14}$$

[4] To simplify the notation sometimes we omit the dependence of N on n.

For the class of discrete memoryless sources, the number of types with denominators n and N is bounded by $(n+1)^{|X|}$ and $(N+1)^{|X|}$ respectively [13], so we can write:

$$\begin{aligned}\max_{P_X} P_{fp} &\leq \max_{P_X} \max_{(P_{x^n},P_{t^N})\in\Lambda_1^*} \\ &\quad [(n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|}P_X(T(P_{x^n})\times T(P_{t^N}))] \\ &\leq (n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|}\cdot \\ &\quad \max_{P_X} \max_{(P_{x^n},P_{t^N})\in\Lambda_1^*} 2^{-n[\mathcal{D}(P_{x^n}||P_X)+\frac{N}{n}\mathcal{D}(P_{t^N}||P_X)]},\end{aligned} \tag{2.15}$$

where for the last inequality we have exploited the independence of x^n and t^N and the property of types according to which for any sequence x^n we have $P_X(T(P_{x^n})) \leq 2^{-n\mathcal{D}(P_{x^n}||P_X)}$ (see [13]). By exploiting lemma 1, we can write:

$$\begin{aligned}\max_{P_X} P_{fp} &\leq (n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|} \\ &\quad \max_{(P_{x^n},P_{t^N})\in\Lambda_1^*} 2^{-n[\mathcal{D}(P_{x^n}||P_{r^{N+n}})+\frac{N}{n}\mathcal{D}(P_{t^N}||P_{r^{N+n}})]} \\ &\leq (n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|}\ 2^{-n(\lambda-|\mathcal{X}|\frac{\log(n+1)(N+1)}{n})} \\ &= 2^{-n(\lambda-2|\mathcal{X}|\frac{\log(n+1)(N+1)}{n})},\end{aligned} \tag{2.16}$$

where the last inequality derives from the definition of Λ_0^*. Together with (2.13), equation (2.16) proves the first part of the lemma with $\delta_n = 2|\mathcal{X}|\frac{\log(n+1)(N+1)}{n}$.
Let now (x^n, t^N) be a generic pair of sequences contained in Λ_1 (with $\Lambda_0 \in \mathcal{S}_{FA}$), due to the limited resources assumption the cartesian product between $T(P_{x^n})$ and $T(P_{t^N})$ will be entirely contained in Λ_1. Then we have:

$$\begin{aligned}2^{-\lambda n} &\geq \max_{P_X} P_X(\Lambda_1) \\ &\overset{(a)}{\geq} \max_{P_X} P_X(T(P_{x^n})\times T(P_{t^N})) \\ &\overset{(b)}{\geq} \max_{P_X} \frac{2^{-[\mathcal{D}(P_{x^n}||P_X)]+\frac{N}{n}\mathcal{D}(P_{t^N}||P_X)}}{(n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|}} \\ &\overset{(c)}{=} \frac{2^{-[\mathcal{D}(P_{x^n}||P_{r^{N+n}})]+\frac{N}{n}\mathcal{D}(P_{t^N}||P_{r^{N+n}})}}{(n+1)^{|\mathcal{X}|}(N+1)^{|\mathcal{X}|}},\end{aligned} \tag{2.17}$$

where (a) is due to the limited resources assumption, (b) follows from the independence of x^n and t^N and the lower bound on the probability of a pair of type classes [13], and (c) derives from lemma 1. By taking the logarithm of both sides we have that $(x^n, t^N) \in \Lambda_1^*$, thus completing the proof. □

The first part of the lemma shows that, at least asymptotically, Λ_0^* belongs to $\mathcal{S}_{FA}$, while the second part implies the optimality of Λ_0^*. The most important consequence of lemma 2 is that the optimum strategy of the FA is univocally determined by the false positive constraint. This solves the apparent problem that we pointed out when defining the payoff of the game, namely that the payoff depends on P_X and P_Y and hence it is not fully known to the FA. Another interesting result is that the optimum strategy of the FA does not depend on the strategy chosen by the AD, thus considerably simplifying the determination of the equilibrium point of the game. As a matter of fact, since the optimum Λ_0^* is fixed, the AD can choose his strategy by relying on the knowledge of Λ_0^*. A last consequence of lemma 2 is that Λ_0^* is the optimum FA strategy even for versions a and b of the SI_{tr}^{lr} game.

2.1.4.2 Asymptotic Nash equilibrium

To determine the Nash equilibrium of the $SI^{lr}_{tr,c}$ game, we start by deriving the optimum strategy for the AD. This is quite an easy task if we observe that the goal of the AD is to take a sequence y^n drawn from Y and modify it in such a way that:

$$h(z^n, t^N) < \lambda - |\mathcal{X}|\frac{\log(n+1)(N+1)}{n}, \tag{2.18}$$

with $d(y^n, z^n) \leq nD$. The optimum attacking strategy, then, can be expressed as a minimization problem, i.e.:

$$f^*(y^n, t^N) = \arg \min_{z^n : d(y^n, z^n) \leq nD} h(z^n, t^N). \tag{2.19}$$

Note that to implement this strategy the AD needs to know t^N, i.e. equation (2.19) determines the optimum strategy only for version c of the game.
Having determined the optimum strategies for the FA and the AD, we can state the fundamental result of this work, summarized in the following Theorem.

Theorem 1. *The profile (Λ^*_0, f^*) defined by lemma 2 and equation (2.19) is an asymptotic Nash equilibrium point for the $SI^{lr}_{tr,c}$ game.*

Proof. We have to prove that:

$$u(\Lambda^*_0, f^*) \geq u(\Lambda_0, f^*) \quad \forall \Lambda_0 \in \mathcal{S}_{FA} \tag{2.20}$$

$$-u(\Lambda^*_0, f^*) \geq u(\Lambda^*_0, f) \quad \forall f \in \mathcal{S}_{AD}. \tag{2.21}$$

The first relation holds because of lemma 2, while the second derives from the optimality of f^* when Λ^*_0 is fixed, hence proving the theorem. □

2.1.5 Discussion and comparison with the SI^{lr}_{ks} game.

In this section we give an intuitive meaning to the results proved so far. To do so we will compare the optimum strategies of the $SI^{lr}_{tr,*}$ game to those of the SI^{lr}_{ks}, i.e a version of the game in which the FA and the AD know the pmf P_X ruling the emission of symbols from the source X. In [10] it is shown that the optimum strategy for the FA relies on the divergence between the empirical pmf of the sequence x^n and P_X, i.e.:

$$\Lambda^*_{0,ks} = \left\{ P_{x^n} \in \mathcal{P}_n : \mathcal{D}(P_{x^n} || P_X) < \lambda - |\mathcal{X}|\frac{\log(n+1)}{n} \right\}. \tag{2.22}$$

One may wonder why the optimum FA strategy for the $SI^{lr}_{tr,*}$ game does not correspond to the comparison of the empirical divergence between x^n and that of the test sequence. The reason for the necessity of adopting the more complicated strategy set by lemma 2 is that in the current version of the game, the FA must ensure that the false positive probability is below the desired threshold for all possible sources in $\mathcal{C}$. To do so, he has to estimate the pmf that better *explains* the evidence provided by both x^n and t^N. In other words he has to find the pmf under which the probability of observing both the sequences x^n and t^N is maximum. This is exactly the role of $P_{r^{n+N}}$ (see equation (A3)), with the generalized log-likelihood ratio corresponding to the log of the (asymptotic) probability of observing x^n and t^N under $P_{r^{n+N}}$ (a geometrical interpretation of the decision strategies for the two versions of the game is given in Figure 2.1).

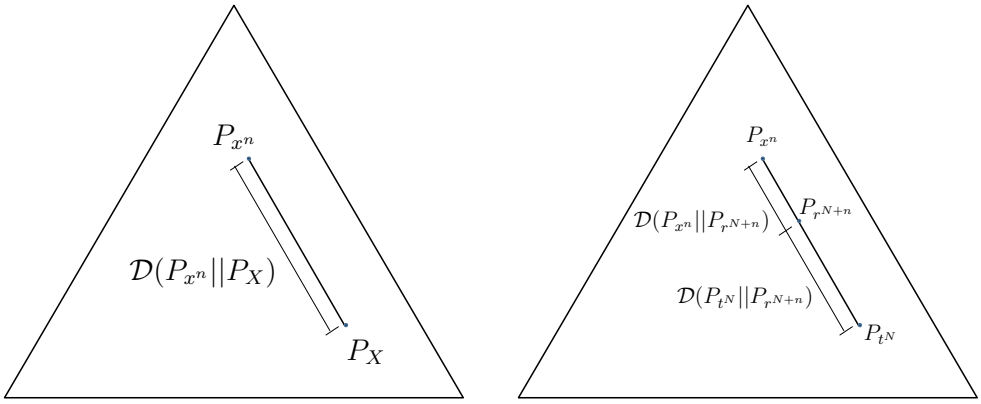


Figure 2.1: Geometric interpretation of the optimum FA strategies for the SI_{ks}^{lr} (left) and the $SI_{tr,*}^{lr}$ (right) games.

Another interesting observation regards the optimum strategy of the AD. As a matter of fact, the functions $h(P_{x^n}, P_{t^N})$ and $\mathcal{D}(P_{x^n}||P_{t^N})$ share a similar behavior: both are positive and convex functions with the absolute minimum achieved when $P_{x^n} = P_{t^N}$, so one may be tempted to think that from the AD's point of view minimizing $\mathcal{D}(P_{x^n}||P_{t^N})$ is equivalent to minimizing $h(P_{x^n}, P_{t^N})$. While this is the case in some situations, e.g. for binary sources or when the absolute minimum can be reached, in general the two minimization problems yield different solutions. It is possible, and quite easy in fact, to find two pmf's P'_{x^n} and P''_{x^n} for which $\mathcal{D}(P'_{x^n}||P_{t^N}) > \mathcal{D}(P''_{x^n}||P_{t^N})$, while $h(P'_{x^n}, P_{t^N}) < h(P''_{x^n}, P_{t^N})$.
Our final comment regards the comparison of the payoff at the equilibrium for the $SI_{tr,c}^{lr}$ and the SI_{ks}^{lr} games. Let us consider the two optimal acceptance regions, that for sake of clarity we will indicated with $\Lambda_{0,ks}^*$ and $\Lambda_{0,tr}^*$. The comparison between $\Lambda_{0,ks}^*$ and $\Lambda_{0,tr}^*$ is not straightforward since the former depends only on P_{x^n} (for a given P_X) while the latter depends both on P_{x^n} and P_{t^N}. In order to ease the comparison we assume that $P_X \in \mathcal{P}_n$ and that P_{t^N} is also fixed and equal to P_X. We can show that under this assumption, and for large n, we have $\Lambda_{0,ks}^* \subseteq \Lambda_{0,tr}^*$. To do so we note that with some algebra the log-likelihood ratio can be rewritten in the following form:

$$h(P_{x^n}, P_{t^N}) = \mathcal{D}(P_{x^n}||P_{t^N}) - \frac{N+n}{n}\mathcal{D}(P_{r^{n+N}}||P_{t^N}). \tag{2.23}$$

From the above equation we see that $h(P_{x^n}, P_{t^N}) \leq \mathcal{D}(P_{x^n}||P_{t^N})$, hence for $P_{t^N} = P_X$ and n large enough[5], the acceptance region for the game with training data contains that of the game with known sources. As a consequence, it is easier for the AD to bring a sequence y^n generated by a source Y within $\Lambda_{0,tr}^*$ and fool the FA. Version c of the SI_{tr}^{lr} game is then more favorable to the attacker than the SI_{ks}^{lr} game. While, the above argument holds only when $P_{t^N} = P_X$, we argue that this is the case even in a general setting. We leave a rigorous proof of the above property to a subsequent work.

[5]If n is large the terms $\frac{\log(n+1)}{n}$ and $\frac{\log(n+1)(N+1)}{n}$ in $\Lambda_{0,ks}^*$ and $\Lambda_{0,tr}^*$ tend to zero.

2.1.6 Conclusions

Following the definition of the SI_{ks} game, extensively treated in [10, 12], we took a further step towards the construction of a theoretical background for multimedia forensics. The source identification game with training data, in fact, is significantly closer to real applications than the game with know sources. The solution of version c of the game provided interesting insights into the optimal strategies for the FA and the AD, that somewhat differ from those that one would have obtained by simply extending the optimum strategies of the known source case. Additional, even more interesting, results are likely to derive from the solution of versions a and b of the SI_{tr} game, which will be the goal of our future work, together with the analysis of the optimal strategies and the resulting payoff for specific cases of particular interest (e.g. for Bernoulli sources). Other interesting directions for future research include the analysis of a version of the game in which the test sequence x^n may have been generated by a (limited) number of sources each known through training sequences. The extensions of the analysis to sources with memory and continuous sources is also worth attention.

2.2 Taking advantage of source correlation in forensic analysis

In a wide range of practical multimedia scenarios several correlated contents are available. The aim of this work is to quantify the gain that can be achieved in forensic applications by jointly considering those contents, instead of analyzing them separately. The used tool is the Kullback-Leibler Divergence between the distributions corresponding to different operators; the Maximum Likelihood estimator of the applied operator is also obtained, in order to illustrate how the correlation is exploited for estimation. Our detailed analysis is constrained to the Gaussian case (both for the input signal distribution and the processing randomness) and linear operators. Several practical scenarios are studied, and the relationships between the derived results are established. Finally, the links with Distributed Source Coding are highlighted.

2.2.1 Introduction

In the last decades the number of multimedia contents and their impact in our lives has dramatically increased. A paradigmatic example of both the cost reduction and ubiquity of capture devices and the growth of digital networks where those contents can be published, shared and distributed, is the wide use of mobile devices (e.g., smart phones) that jointly offer the capturing and connectivity functionalities. Multimedia contents have been converted not only in valuable evidence of our personal evolution and social life, but also in a weapon that can be used to harm the public image of individuals and organizations. In fact, simultaneously with this growth, a huge number of editing tools available in applications for non-skilled users have proliferated, thus compromising the reliability of those contents, and strongly constraining their use in some applications, for example as court evidence. As a consequence, trust on multimedia contents has steadily decreased.
In this context, multimedia forensics, an area of multimedia security, has appeared as a possible solution to the decrease of confidence on multimedia contents. The target of multimedia forensics can be summarized as assessing the processing, coding and editing steps a content has gone through. Despite the large attention that multimedia forensics has deserved during the last years (see, for instance, [19] and the references therein), most of the previous works deal with single sources, i.e., they perform the forensic analysis of video, audio or still images, but they do not consider in a joint

way several correlated instances of those media. However, examples of those correlated contents can be found in a number of practical situations, for example:

- Multimodal content: one of the most interesting cases are video files with audio tracks. For example, both the visual and audio contents provide environment information that should be coherent; otherwise, inconsistencies would indicate that at least one of the modalities was tampered with. This idea is explored in [20], where the volumetric characteristics of the capture environment are estimated both from the video and audio signals.
- Multitrack files: obviously, the left and right channels of stereo audio files are not independent; the correlation between them could be exploited for forensic purposes. The same idea is applicable to 3-D video, or multi-channel audio.

Be aware that the common characteristic of those scenarios is that a number (typically 2) of correlated sources is considered. In this work we will try to measure, by taking a theoretical approach, the advantage of jointly considering these contents for performing the forensic analysis of the total multimedia contents; specifically, we will quantify the gain that can be achieved by considering them in a joint way. Both information-theoretic and estimation tools will be used.
The rest of the paper is organized as follows: Sect. 2.2.2 introduces the used notation and the goals of the detection and estimation forensic problems. The proposed target functions and general strategies are introduced in Sect. 2.2.3, while they are particularized to the linear and Gaussian case in Sect. 2.2.4. Numerical results are introduced in Sect. 2.2.5, and conclusions are summarized in Sect. 2.2.6.

2.2.2 Notation and objectives

Random vectors will be denoted by capital bold letters (e.g., $\mathbf{Y}$), while their outcomes, and deterministic vectors in general, will use lower case bold letters (e.g., $\mathbf{y}$). $\Sigma_{\mathbf{X}}$ will be used for denoting the covariance matrix of random vector $\mathbf{X}$, and $\boldsymbol{\mu}_{\mathbf{X}}$ its mean. Subindices will be used for denoting the vector component at ith position (e.g., Y_i, or y_i); for the sake of notational simplicity $\left(\boldsymbol{\mu}_{\mathbf{X}}\right)_i = \mu_{X_i}$. The element at the ith row and jth column of a general matrix A will be denoted by $(A)_{i,j}$.
Let $X_1, X_2, \ldots, X_L$ denote L random variables, which model the correlated sources we consider; $\mathbf{X}$ will be used for denoting $(X_1, X_2, \ldots, X_L)$. Throughout this work we will assume the statistics (mean vector and covariance matrix) of $\mathbf{X}$ to be perfectly known at the detector/estimator.
We assume that each of those variables goes through a particular processing $Y_i = g_i(X_i)$,[6] where $1 \leq i \leq L$, $g_i \in \mathcal{G}$, and $\mathcal{G}$ denotes the space of memoryless proccesing operators. In general, these operators can be randomized; this randomness will be modeled by variables Z_i, $1 \leq i \leq L$, where the statistics of $\mathbf{Z}$ will be assumed to be also perfectly known at the detector/estimator. For the sake of notational simplicity, we will define every $g_i \in \mathcal{G}$ by two sets of parameters, namely, φ_i and ϕ_i, so $g_i(\cdot) = g(\cdot, \varphi_i, \phi_i)$. These two sets of parameters are used for making the distinction between those that we want to estimate/detect, and those which we do not (typically known as unwanted or nuisance parameters [1]), respectively.
For the study of the Maximum Likelihood (ML) processing operator estimator, N independent observations of $\mathbf{Z}$ will be considered, i.e., we will assume each of those L sources and the corresponding processing to be memoryless. Each of those N observations of $\mathbf{Z}$ will be denoted by $\mathbf{Z}^i$, $1 \leq i \leq N$. In the information theoretic analysis, and due to the independence among the N observations, the

[6] In one of the scenarios analyzed in the following sections, specifically, for that considered in Sect. 2.2.4.5, we have adopted a more general approach.

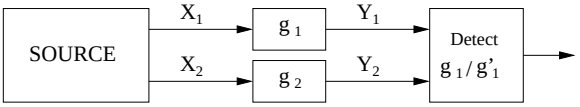


Figure 2.2: Distinguishability problem framework for $L = 2$.

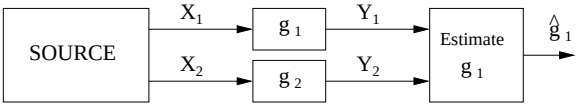


Figure 2.3: Estimation problem framework for $L = 2$.

obtained results will be proportional to N; consequently, and for the sake of notational simplicity, we will skip the superindex.
In this work we focus on two different problems:

- study of the distinguishability between g_i and h_i, $g_i \in \mathcal{G}$ and $h_i \in \mathcal{G}$. First, we analyze the case where only the marginal probability density function (pdf) of Y_i is considered, and then we compare it with its counterpart where the joint pdf of $\mathbf{Y}$ is exploited. Of course, one would expect that whenever the joint pdf is employed, the distinguishability is improved; in that sense, one of the main contributions of the current work is to consider several scenarios that model practical signal processing operations, and to quantify the improvement achieved by using the correlation between the sources (i.e., the joint pdf instead of the marginal). The block diagram of this scenario is plotted in Fig. 2.2 for the case $L = 2$.

- estimate the applied operator. Again, intuition says that the more data we consider, the better (or at least not worse) the estimation will be. We analyze how the correlation between sources is exploited by the processing operator estimation. The block diagram of this scenario is plotted in Fig. 2.3 for the case $L = 2$.

2.2.3 General case

Although already well-known in information theory, the Kullback-Leibler Divergence (KLD), also known as relative entropy, has been just recently proposed for distinguishing between different sources [10], and processing operators [21] in multimedia forensics. The KLD for continuous L-dimensional random variables is defined as

$$D(f_0||f_1) = \int_{\mathbb{R}^L} f_0(\mathbf{x}) \log\left(\frac{f_0(\mathbf{x})}{f_1(\mathbf{x})}\right) d\mathbf{x},$$

where f_0 denotes the pdf under the null hypothesis, and f_1 under the alternative one (the two hypotheses under analysis). Its use is based on its asymptotical (when the dimensionality of the problem goes to infinity) optimality, since it is asymptotically equivalent to the Neyman-Pearson criterion, which is known to be the most powerful test for the binary hypothesis problem. Indeed, Chernoff-Stein's Lemma [13] states that the false positive probability error exponent achievable for a given non-null false negative probability asymptotically converges to the KLD between the pdfs under the null and alternative hypotheses (as long as the KLD takes a finite value) when the dimensionality of the problem goes to infinity.

In the case where we only want to distinguish between the values of some of the applied signal processing operator parameters (those that we have previously denoted by φ_i), but we are not interested in distinguishing between different values of the remaining ones (i.e., ϕ_i) we will follow a worst case approach. Specifically, given that we are interested in studying the distinguishability between the processing corresponding to φ_i and φ_i', we will look for those values of ϕ_i and ϕ_i' minimizing the relative entropy, i.e., to quantify the distinguishability between $f_{g(X,\varphi_i,\cdot)}$ and $f_{g(X,\varphi_i',\cdot)}$ we compute

$$\min_{\phi_i} \min_{\phi_i'} D(f_{g(X,\varphi_i,\phi_i)} || f_{g(X,\varphi_i',\phi_i')}).$$

This approach resembles the strategy which is typically followed in the literature for statistical detection theory with unwanted parameters (c.f., [1]), since it maximizes the performance of the system (by using the optimal distinguishability measure, the KLD) for the worst case scenario, ensuring the predicted performance. This strategy is also coherent with the approach proposed in [21] for quantifying the distinguishability between different classes of processing operators.
On the other hand, the ML estimate of processing g_i requires the calculation of

$$\hat{g}_i = \arg\max_{g_i \in \mathcal{G}} f_{\mathbf{Y}}(\mathbf{y}|g_i).$$

Again, if we are interested in estimating only some of the parameters defining g_i, i.e. φ_i, then we must solve

$$\hat{\varphi}_i = \arg\max_{\varphi_i} \max_{\phi_i \in \Phi} f_{g(X,\varphi_i,\phi_i)},$$

where Φ is the feasible set of values of ϕ_i. This framework encompasses the case where ϕ_i is known to have a fixed value ϕ^*, as in such case $\Phi = \{\phi^*\}$. Note that, since in this case we are looking for the most probable operator, instead of a maxmin, a maxmax strategy will be followed; in other words, in the estimate problem it does not make sense to use a worst case approach, as one does not have to consider the probability of confusing with an alternative hypothesis.
Finally, we would like to mention that the improvement on the performance of the estimation of g_i could be also interpreted from an information-theoretic point of view. Indeed, if one considers G_i to be randomly chosen following a given distribution, then, based on fundamental properties of the entropy [13] we can bound $h(G_i|Y_i) \geq h(G_i|\mathbf{Y})$, i.e., the consideration of the output of the other processing branches will reduce (or at least not increase) the uncertainty about the processing undergone by X_i.

2.2.3.1 Links with Distributed Source Coding

In source coding, the exploitation of correlation between sources has been extensively used for improving the performance of the coding scheme in those scenarios where the coders do not share access to their input data, i.e., the Distributed Source Coding (DSC) problem [22, 23]. Indeed, this correlation is typically modeled as a virtual channel, and channel coding techniques are used for source coding purposes. Nevertheless, due to the differences in the target function between the current problem, where the processing undergone by the different sources is to be detected/estimated, and the DSC problem, where one wants to minimize the transmitted data, the traslation of the channel-coding based techniques to the forensic application seems to be unfeasible.
Another related problem is the Distributed Hypothesis Testing [24], where one wants to determine how the data should be compressed in order to minimize the transmitted information when the goal

is not the reproduction, but the inference from those data. Although in this case we have indeed a detection problem, in the forensic application we are not interested in reducing the transmitted data; consequently, the traslation of the results in [24] to the current problem appears to be very difficult.

2.2.4 Gaussian signals and linear operators

In order to provide close formulas that allow a clear comparison between the considered scenarios, we particularize the proposed framework to the case where Gaussian variables and linear processing is considered. Therefore, in this section we will consider the processing defined by $Y_i = a_i X_i + Z_i$, where a_i is a real constant, $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{X}}, \Sigma_{\mathbf{X}})$, and $Z_i \sim \mathcal{N}(\mu_{Z_i}, \sigma^2_{Z_i})$ is a Gaussian random variable independent of $\mathbf{X}$ and independent of Z_j, $1 \leq j \leq L$, $j \neq i$. Random variable Z_i might model the randomness of the processing, for example, the effect of quantizing the processed signal in a different domain, e.g., an image operator that scales the 8×8-block DCT coefficients depending on the frequency location, and then quantizes the image in the pixel domain; although the quantization error is not independent of the DCT coefficients, it is typically modeled as being so (see, for example, [25]), as a lot of different contributions are summed up when performing the DCT and IDCT.
Based on the definition of $\mathbf{Y}$, and the distributions of $\mathbf{X}$ and $\mathbf{Z}$, $\mathbf{Y}$ is also Gaussian, i.e., $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Y}}, \Sigma_{\mathbf{Y}})$, where

$$\mu_{Y_i} = a_i \mu_{X_i} + \mu_{Z_i},$$

and

$$\left(\Sigma_{\mathbf{Y}}\right)_{i,j} = a_i a_j \left(\Sigma_{\mathbf{X}}\right)_{i,j} + \sigma^2_{Z_i} \delta[i-j].$$

The main advantage of the Gaussian case, that drives us to consider this scenario with special detail, is the fact that closed formulas exist for the KLD of two Gaussian multivariate distributions. Indeed, if we consider $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Y}}, \Sigma_{\mathbf{Y}})$ and $\mathbf{Y}' \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{Y}'}, \Sigma_{\mathbf{Y}'})$, then

$$D(f_{\mathbf{Y}} || f_{\mathbf{Y}'}) = \frac{1}{2}\left[\mathrm{tr}\left(\Sigma^{-1}_{\mathbf{Y}'}\Sigma_{\mathbf{Y}}\right) + \left(\boldsymbol{\mu}_{\mathbf{Y}'} - \boldsymbol{\mu}_{\mathbf{Y}}\right)^T \Sigma^{-1}_{\mathbf{Y}'}\left(\boldsymbol{\mu}_{\mathbf{Y}'} - \boldsymbol{\mu}_{\mathbf{Y}}\right) - \log\left(\frac{|\Sigma_{\mathbf{Y}}|}{|\Sigma_{\mathbf{Y}'}|}\right) - L\right], \quad (2.24)$$

where $\mathrm{tr}(\cdot)$ is the trace operator, and $|\Sigma|$ is the determinant of matrix Σ.
Taking into account the form of Y_i considered in this section, g_i is entirely specified by a_i and $\sigma^2_{Z_i}$. In most practical scenarios we will be interested in estimating a_i, whereas $\sigma^2_{Z_i}$ is an unwanted parameter; therefore, following the notation introduced in the previous section, $\varphi_i = a_i$, and $\phi_i = \sigma^2_{Z_i}$. Consequently, the ML estimate of gain a_i requires the calculation of

$$\hat{a}_i = \arg\max_{a_i \in \mathbb{R}} \left(\max_{\sigma^2_{Z_i} \in \mathbb{R}^+} L(\mathbf{y}, a_i, \sigma^2_{Z_i}) \right),$$

where

$$L(\mathbf{y}, a_i, \sigma^2_{Z_i}) \triangleq \left(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}\right)^T \Sigma^{-1}_{\mathbf{Y}} \left(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}\right) + \log(|\Sigma_{\mathbf{Y}}|).$$

On the other hand, if the unwanted parameter is indeed known *a priori*, then that knowledge can be exploited in the estimation. Continuing with the estimate of a_i, but assuming that $\sigma^2_{Z_i}$ is known to be, say, $\left(\sigma^2_{Z_i}\right)^*$, we have that

$$\hat{a}_i = \arg\max_{a_i \in \mathbb{R}} L(\mathbf{y}, a_i, \left(\sigma^2_{Z_i}\right)^*).$$

In the following we consider 4 particular scenarios for $L = 2$ and different definitions of Y_2 (Sects. 2.2.4.2-2.2.4.5), while keeping the same definition of Y_1. For all of them, we detail the theoretical results of both the ML estimator and the KLD. The target of this analysis is to illustrate how the knowledge of Y_2 helps to estimate/detect the processing undergone by Y_1 in comparison to the scenario where only Y_1 is available (Sect. 2.2.4.1). In order to keep the mathematical tractability, we will assume $\Sigma^2_{Z_i} = \mu_{X_i} = \mu_{Z_i} = 0$, for $i = 1, 2$. The noisy case (randomized processing operators) and non-zero mean will be considered in Sect. 2.2.5 by numerical results.
The proposed scenarios can be linked with real applications in the case of audio stereo files, where each audio channel goes through an equalization filter; the samples of each channel are windowed, frequency transformed, and then each frequency coefficient is subjected to a different scaling. This effect can be roughly modeled by a frequency dependent scaling, and the differences between this model and the real processing (encompassing, for example, the windowing effect, the lack of block periodicity, and the quantization of the filtered samples in the time domain) will be modeled by Z_i. Of course the frequency coefficients do not fit the theoretical model studied in this section, but the consideration of this application scenario showcases the power of the proposed methodology. We will particularize this illustrating application for each scenario.

2.2.4.1 Scenario $Y_1 = a_1X_1 + Z_1$

Application Scenario: mono file, or only one of the stereo channels is considered for processing estimation/detection.
Under the hypotheses mentioned above,

$$\hat{a}_1 = \pm\sqrt{\frac{\sum_{i=1}^{N}\left(Y_1^i\right)^2}{N\left(\Sigma_{\mathbf{X}}\right)_{1,1}}}, \tag{2.25}$$

that is, the variance-based estimator, which in general is biased.
Concerning the KLD between $Y_1 = a_1X_1 + Z_1$ and $Y_1' = b_1X_1 + Z_1'$, one obtains

$$D(f_{Y_1}||f_{Y_1'}) = \frac{1}{2}\left(-1 + \frac{a_1^2}{b_1^2} - \log\left[\frac{a_1^2}{b_1^2}\right]\right). \tag{2.26}$$

2.2.4.2 Scenario $Y_1 = a_1X_1 + Z_1$, $Y_2 = a_2X_2 + Z_2, a_2 = a_1$

Application Scenario: stereo case, when we know that the same equalization is applied to both channels.
The ML estimator can be computed as

$$\hat{a}_1 = \pm\sqrt{\frac{\sum_{i=1}^{N}\left(Y_1^i\right)^2\left(\Sigma_{\mathbf{X}}\right)_{2,2} + \left(Y_2^i\right)^2\left(\Sigma_{\mathbf{X}}\right)_{1,1} - 2\left(Y_1^iY_2^i\right)\left(\Sigma_{\mathbf{X}}\right)_{1,2}}{2N\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - \left(\Sigma_{\mathbf{X}}\right)^2_{1,2}\right]}}.$$

Be aware that whenever X_1 and X_2 are independent, i.e., $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$, then the derived ML estimator is

$$\hat{a}_1 = \pm\sqrt{\frac{1}{2N}\left[\frac{\sum_{i=1}^{N}\left(Y_1^i\right)^2}{\left(\Sigma_{\mathbf{X}}\right)_{1,1}} + \frac{\sum_{i=1}^{N}\left(Y_2^i\right)^2}{\left(\Sigma_{\mathbf{X}}\right)_{2,2}}\right]}, \tag{2.27}$$

which is obvioulsy related to the ML estimator in (2.25).

Concerning the KLD between $(Y_1, Y_2) = (a_1X_1 + Z_1, a_2X_2 + Z_2)$ and $(Y_1', Y_2') = (b_1X_1 + Z_1', b_2X_2 + Z_2')$, we obtain

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = -1 + \frac{a_1^2}{b_1^2} - \log\left[\frac{a_1^2}{b_1^2}\right], \tag{2.28}$$

which is nothing but twice (2.26). This result makes sense, since we are considering the same processing for both channels and the noiseless case, and consequently the correlation between X_1 and X_2 does not provide any additional information; therefore, from the KLD point of view one would expect to have the same result that is achieved when two independent realizations of Y_1 are available. This result also makes sense at the light of (2.27), although in the derivation of the latter we assumed $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$.

2.2.4.3 Scenario $Y_1 = a_1X_1 + Z_1$, $Y_2 = a_2X_2 + Z_2$. a_2 is known

Application Scenario: stereo, we know the equalization applied to one channel, and want to estimate the other one.
The ML estimator is

$$\begin{aligned}\hat{a}_1 = &\left\{\sum_{i=1}^{N} -a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2} Y_1^i Y_2^i + \left[\left(\sum_{i=1}^{N} a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2} Y_1^i Y_2^i\right)^2\right.\right.\\ &\left.\left. + 4Na_2^4\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - \left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]\left(\Sigma_{\mathbf{X}}\right)_{2,2}\sum_{i=1}^{N}\left(Y_1^i\right)^2\right]^{1/2}\right\}\\ &\left[2Na_2^2\left(\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - \left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right)\right]^{-1}.\end{aligned} \tag{2.29}$$

Concerning the KLD between $(Y_1, Y_2) = (a_1X_1 + Z_1, a_2X_2 + Z_2)$ and $(Y_1', Y_2') = (b_1X_1 + Z_1', b_2X_2 + Z_2')$, we obtain

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = -1 - \frac{1}{2}\log\left(\frac{a_1^2a_2^2}{b_1^2b_2^2}\right) + \frac{\left(a_2^2b_1^2 + a_1^2b_2^2\right)\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - 2a_1a_2b_1b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2}{2b_1^2b_2^2\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - \left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]}. \tag{2.30}$$

Note that whenever $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = \tfrac{1}{2}\left[-1 + \tfrac{a_1^2}{b_1^2} - \log\left(\tfrac{a_1^2}{b_1^2}\right) - 1 + \tfrac{a_2^2}{b_2^2} - \log\left(\tfrac{a_2^2}{b_2^2}\right)\right],$$

which also follows the intuition for the KLD of multivariate Gaussian distributions of diagonal covariance matrices.
The scenario $Y_1 = a_1X_1 + Z_1$, $Y_2 = a_2X_2 + a_3X_1 + Z_2$ (so $Y_2 \neq g_2(X_2)$), where a_2 and a_3 are known, was also studied, although the obtained results are not shown here due to spatial constraints. Let only mention that it corresponds to the stereo case, where channel 2 is not only equalized, but edited by combining it with a filtered version of channel 1. Our target would be to estimate the equalizer undergone by output 1, that depends only on channel 1 input.

2.2.4.4 Scenario $Y_1 = a_1X_1 + Z_1$, $Y_2 = a_2X_2 + Z_2$. a_2 is not known

Application Scenario: stereo, we want to estimate the equalizer applied to one of the channels, but we do not know about the equalization applied to the other one.
In this framework, the value of a_2 (as a function of a_1) maximizing the ML target function is

$$\frac{-\xi+\sqrt{\xi^2+4Na_1\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]\sum_{i=1}^N a_1\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(Y_2^i\right)^2}}{2Na_1\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]},$$

where $\xi \triangleq \sum_{i=1}^N \left(\Sigma_{\mathbf{X}}\right)_{1,2} Y_1^i Y_2^i$, yielding the ML estimator

$$\hat{a}_1 = \pm\sqrt{\frac{\left(\Sigma_{\mathbf{X}}\right)_{2,2}\kappa - \sqrt{\frac{\left(\Sigma_{\mathbf{X}}\right)_{2,2}}{\left(\Sigma_{\mathbf{X}}\right)_{1,1}}\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\kappa\left[\sum_{i=1}^N Y_1^iY_2^i\right]^2}}{N\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]\sum_{i=1}^N\left(Y_2^i\right)^2}},$$

where $\kappa \triangleq \left[\sum_{i=1}^N \left(Y_1^i\right)^2\right]\left[\sum_{i=1}^N \left(Y_2^i\right)^2\right]$. Note that whenever $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$, then this estimator is equivalent to (2.25).
On the other hand, in the computation of the KLD, and given that we study the distinguishability between the processing corresponding to a_1 and b_1, we will look for those values of a_2 and b_2 minimizing the KLD. In the current scenario, the derivative of the KLD with respect to a_2 is

$$-\frac{1}{a_2} + \frac{b_1a_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2} - a_1b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2}{b_1b_2^2\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]},$$

which has roots with respect to a_2 at $\frac{b_2\left(a_1\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2+\gamma\right)}{2b_1\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}}$, where $\gamma \triangleq \sqrt{a_1^2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^4 + 4b_1^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]}$.
If one replaces (2.2.4.4) into the relative entropy, the result does not depend on b_2, so the minimization over that variable is indeed not necessary. The obtained value is

$$\begin{aligned}D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = &-\frac{1}{2}\\ &+\frac{a_1^2\left[2\left(\Sigma_{\mathbf{X}}\right)_{1,1}^2\left(\Sigma_{\mathbf{X}}\right)_{2,2}^2-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^4\right]-a_1\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\kappa}{4b_1^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}\left[\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]}\\ &-\frac{1}{2}\log\left(\frac{a_1^2\left[a_1\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2+\kappa\right]^2}{4b_1^4\left(\Sigma_{\mathbf{X}}\right)_{1,1}^2\left(\Sigma_{\mathbf{X}}\right)_{2,2}^2}\right).\end{aligned}$$

An interesting scenario, is that where $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$; under that assumption, the derivative of the KLD with respect to a_2 is equal to $-\frac{1}{a_2} + \frac{a_2}{b_2^2}$, yielding the condition $a_2^2 = b_2^2$. Indeed, in that particular framework the KLD can be written as (check the obvious relationships with (2.26))

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = \frac{1}{2}\left(-1+\frac{a_1^2}{b_1^2}-\log\left[\frac{a_1^2}{b_1^2}\right]\right)+\frac{1}{2}\left(-1+\frac{a_2^2}{b_2^2}-\log\left[\frac{a_2^2}{b_2^2}\right]\right),$$

and consequently we can minimize the KLD over $\frac{a_2^2}{b_2^2}$; straightforwardly, the achieved solution is $\frac{a_2^2}{b_2^2} = 1$, providing a null contribution to the total KLD, whose value will be

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = \frac{1}{2}\left(-1 + \frac{a_1^2}{b_1^2} - \log\left[\frac{a_1^2}{b_1^2}\right]\right), \tag{2.31}$$

i.e., the same value achieved in (2.26). The implications of this result are evident:

- Since X_1 and X_2 are independent, the consideration of Y_2 and Y_2' will not provide any knowledge that can improve the distinguishability between a_1 and b_1.
- Therefore, if we look for those values of a_2 and b_2 minimizing the KLD, we will find that $a_2^2 = b_2^2$ (due to the symmetry obtained by assuming zero-mean random variables).
- Consequently, we go back to the framework studied in Sect. 2.2.4.1.

Another asymptotical scenario, that is also particularly interesting, is that where $\left(\Sigma_{\mathbf{X}}\right)_{1,2} \rightarrow \pm\sqrt{\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}}$, i.e., if X_1 and X_2 are (almost) deterministically related. It can be checked that in that framework the KLD goes to infinity whenever $a_2 \neq \frac{a_1 b_2}{b_1}$. The intuition behind this result is also interesting: since X_1 and X_2 are related by a fixed factor, if we compute $\frac{Y_1}{Y_2} = \frac{a_1}{a_2}$ it will be trivial to distinguish that scenario from $\frac{Y_1'}{Y_2'} = \frac{b_1}{b_2}$ unless $\frac{a_1}{a_2} = \frac{b_1}{b_2}$.[7]
Therefore, in order to follow our worst case approach, we will choose a_2 to be $\frac{a_1 b_2}{b_1}$. By doing so, the resulting KLD value is

$$D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) = -1 + \frac{a_1^2}{b_1^2} - \log\left[\frac{a_1^2}{b_1^2}\right],$$

which is nothing but twice (2.26) (and therefore twice (2.31)), and exactly the same than (2.28), although in that case this value was obtained for a generic covariance matrix.
Again, this result illustrates what one would intuitively expect; the larger the correlation between the sources, the easier it will be to distinguish between the operators. Indeed, the two limit behaviors are also very enlightening:

- Whenever the considered sources are independent, the achieved KLD is equivalent to that where only Y_1 is considered. Of course in this framework Y_2 does not provide any knowledge on Y_1, and consequently we can just neglect that variable.
- Whenever the relationship between the sources is deterministic, the problem is equivalent to having two independent observations, coming from a single source.

2.2.4.5 Scenario $Y_1 = a_1X_1 + Z_1$, $Y_2 = a_2X_2 + a_3X_1 + Z_2$, so $Y_2 \neq g_2(X_2)$. a_2 and a_3 are known.

Application Scenario: stereo, channel 2 is not only equalized, but edited by combining it with a filtered version of channel 1. We want to estimate the equalizer undergone by output 1, that depends only on channel 1 input.

[7] Take into account that we assume this equality to hold before taking the limit $\left(\Sigma_{\mathbf{X}}\right)_{1,2} \rightarrow \pm\sqrt{\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}}$. In the limit the covariance matrix of $\mathbf{X}$ becomes singular (with computational problems arising when computing (2.24)), so it is important to note that by considering $\frac{a_1}{a_2} = \frac{b_1}{b_2}$ the KLD does not depend on the covariance matrix of $\mathbf{X}$.

The ML estimator is

$$\begin{aligned}\hat{a}_1 =& \frac{\sum_{i=1}^{N}\left[-\left(a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}+a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}\right)Y_1^iY_2^i\right]}{2Na_2\left[a_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-2a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{1,2}-a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]}\\ &+\left[\left(\sum_{i=1}^{N}\left(a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}+a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}\right)Y_1^iY_2^i\right)^2\right.\\ &+4Na_2\left[a_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-2a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{1,2}-a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]\\ &\left.\sum_{i=1}^{N}\left[\left(a_3^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}+a_2^2\left(\Sigma_{\mathbf{X}}\right)_{2,2}\right)Y_1^2\right]\right]\\ &\left\{2Na_2\left[a_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-2a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{1,2}-a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}^2\right]\right\}^{-1}.\end{aligned}$$

One can check that under the current analysis hypotheses, whenever $a_3 = 0$, the latter estimator is equivalent to (2.29).
Concerning the KLD,

$$\begin{aligned}D(f_{(Y_1,Y_2)}||f_{(Y_1',Y_2')}) =& -1\\ &+\frac{-2a_1b_1\left(a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}+a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}\right)\left(b_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}+b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}\right)}{2b_1^2b_2\left[b_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}\left(b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}+2b_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\right)\right]}\\ &+\frac{b_1^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(a_3^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}+a_2^2\left(\Sigma_{\mathbf{X}}\right)_{2,2}\right)+a_1^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(b_3^2\left(\Sigma_{\mathbf{X}}\right)_{1,1}+b_2^2\left(\Sigma_{\mathbf{X}}\right)_{2,2}\right)}{2b_1^2b_2\left[b_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}\left(b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}+2b_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\right)\right]}\\ &-\frac{1}{2}\log\left(\frac{a_1^2a_2\left[a_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}\left(a_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}+2a_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\right)\right]}{b_1^2b_2\left[b_2\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}-\left(\Sigma_{\mathbf{X}}\right)_{1,2}\left(b_2\left(\Sigma_{\mathbf{X}}\right)_{1,2}+2b_3\left(\Sigma_{\mathbf{X}}\right)_{1,1}\right)\right]}\right);\end{aligned}$$

again, if $a_3 = 0$ we obtain (2.30) as a particular case of the last formula.

2.2.5 Numerical results

In this section we will provide a glance at some of those Gaussian linear cases that have not been studied in the previous section due to their cumbersome mathematical expressions. First of all, we will consider the effect of the processing noise $\mathbf{Z}$. The continuous lines in Fig. 2.4 show the results obtained when $a_2 = a_1$ (correspondingly, $b_2 = b_1$), i.e., the scenario studied in Sect. 2.2.4.2. Be aware that in that framework, and as it was previously discussed, the fact of considering twice the same processing operator helps to our estimation. Nevertheless, if the observations are noisy, the closer they are, the more difficult will be to appreciate the different information that each observation provides; indeed, when they get very close (i.e., when $\left(\Sigma_{\mathbf{X}}\right)_{1,2} \to \sqrt{\left(\Sigma_{\mathbf{X}}\right)_{1,1}\left(\Sigma_{\mathbf{X}}\right)_{2,2}}$) the distinguishability will be equivalent to having a single observation (the KLD decreases to half the value we have for $\left(\Sigma_{\mathbf{X}}\right)_{1,2} = 0$). This illustrates that a very high correlation between sources is not always positive for distinguishability.
On the other hand, Fig. 2.4 also contains the results when a_2 and b_2 are not known, i.e., the scenario considered in Sect. 2.2.4.4; the curve obtained for $\sigma_Z^2 = 0$ corresponds to the results derived there. As mentioned in Sect. 2.2.3 we have decided to follow a worst case approach for this scenario. Indeed, for the noisy case the values of a_2 and b_2 minimizing the KLD are $a_2 = 0$ and $b_2 = 0$; the intuitive

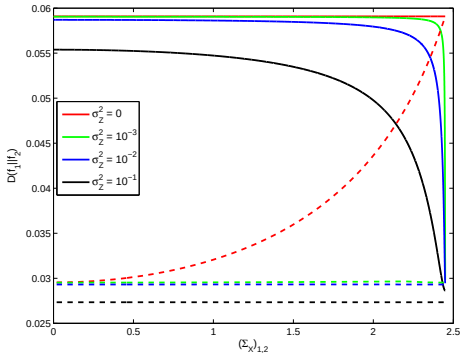


Figure 2.4: KLD when $a_2 = a_1$ and $b_2 = b_1$ (continuous lines) and when a_2 and b_2 are not known (discontinuous ones), for different values of $\sigma_{Z_1}^2 = \sigma_{Z_2}^2 = \sigma_Z^2$. $a_1 = 1$, $b_1 = 1.2$, $(\Sigma_{\mathbf{X}})_{1,1} = 2$, $(\Sigma_{\mathbf{X}})_{1,1} = 3$, $(\Sigma_{\mathbf{X}})_{1,2} \in \left[0, \sqrt{(\Sigma_{\mathbf{X}})_{1,1} (\Sigma_{\mathbf{X}})_{2,2}}\right]$, $\boldsymbol{\mu}_{\mathbf{X}} = \mathbf{0}$, $\boldsymbol{\mu}_{\mathbf{Z}} = \mathbf{0}$.

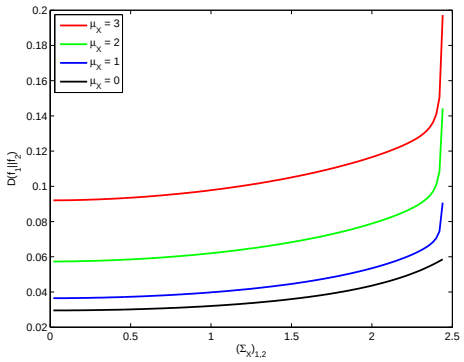


Figure 2.5: KLD when a_2 and b_2 are not known, for $\sigma_{Z_1}^2 = \sigma_{Z_2}^2 = 0$ and different values of $\mu_{X_1} = \mu_{X_2} = \mu_X$. $a_1 = 1$, $b_1 = 1.2$, $(\Sigma_{\mathbf{X}})_{1,1} = 2$, $(\Sigma_{\mathbf{X}})_{1,1} = 3$, $(\Sigma_{\mathbf{X}})_{1,2} \in \left[0, \sqrt{(\Sigma_{\mathbf{X}})_{1,1} (\Sigma_{\mathbf{X}})_{2,2}}\right]$, $\boldsymbol{\mu}_{\mathbf{Z}} = \mathbf{0}$.

idea behind this result is also clear: in the worst case we cannot trust the second observation, as it is only noise. In that case the KLD is

$$\frac{1}{2}\left[\frac{a_1^2 (\Sigma_{\mathbf{X}})_{1,1} + \sigma_Z^2}{b_1^2 (\Sigma_{\mathbf{X}})_{1,1} + \sigma_Z^2} - \log\left(\frac{a_1^2 (\Sigma_{\mathbf{X}})_{1,1} + \sigma_Z^2}{b_1^2 (\Sigma_{\mathbf{X}})_{1,1} + \sigma_Z^2}\right) - 1\right],$$

which is independent of the correlation term, as expected.
Finally, the influence of the mean of the original signal on the distinguishability is illustrated in Fig. 2.5, which clearly shows that the larger the mean of the signal, the easier will be to distinguish the considered processing operators.

2.2.6 Conclusions

In this work we have quantified the advantages of using the joint distribution of composite objects for improving the distinguishability between processing operators. Although for the sake of tractability

we have focused on the linear Gaussian case, it is evident that the principles derived here would be applicable to more general frameworks. Among these principles, we can mention the behavior of the distinguishability measures in different scenarios, and how, for example, the case where the correlated sources are known to share their processing is equivalent to having independent sources. It also interesting to note that for the case where the unwanted processing is unknown, for the noiseless case the distinguishability achieved for deterministically correlated sources is double the one obtained for independent sources, but for the noisy case the obtained value is independent of that correlation, as the second observation is considered to be pure noise. Finally, we would like to mention the *a priori* striking result showing that a larger correlation between sources does not always imply a better distinguishability between operators.

Chapter 3

Operator chain modeling

3.1 JPEG Quantization and full-frame filtering

A plethora of forensics techniques have been proposed in the literature so far, aiming at identifying specific processing operators applied to images [26], but little attention has been paid on the forensic analysis of chains of operators. In such a scenario, difficulties in the detection may arise since the statistical footprints exploited to detect a specific processing may be wiped off by the application of a second one. We propose here to continue and extend our previous work, where we analyzed the Discrete Cosine Transform (DCT) statistical properties of a JPEG image, post-processed with linear processing operators. In this scenario, some well-known statistical properties of a JPEG image are perturbed, thus complicating the application of previous forensic works (e.g. [25],[27]). We derive an accurate mathematical model that establishes a precise relationship between DCT coefficients before and after filtering. Finally, the presented analysis is exploited to build a model for the DCT distribution of JPEG images filtered with various linear kernels. We will assume in this work the quantization to be fixed and known, although future work will be devoted to remove this assumption. By mean of the χ^2 histogram distance, we measure the distinguishability between the derived models (each model depends on the applied filter kernel) and the actual distribution of a to-be-tested image, aiming at identifying the linear operator which has been applied. Other distinguishability measures may be employed. The choice of χ^2 distance was primarily due to the widespread use of the χ^2 goodness of fit test (also known as Pearsons test [28]) as one of the most accepted non-parametric statistical test for determining if an observed frequency distribution matches with a theoretical one or not. It has been successfully employed in many research areas, e.g., near duplicate image identification [29], shape and texture classification [30] and steganography [31]. Moreover, in [32] it has been shown that out of a number of non-parametric test statistics, χ^2 metric gets some of the best results in terms of accurate distance metric for probability density functions.
To the best of our knowledge, the presented work constitutes a first attempt to study the statistical perturbation introduced by linear operators on JPEG images and, although their detection is not necessarily proof of malicious tampering, the derived framework represents a valuable mean to disclose the processing history.
In Section 3.1.1 we briefly review the needed mathematical background, while in Section 3.1.2 we recall our previous work on the definition of the mathematical model characterizing the statistical properties of a JPEG linearly filtered image. Following, in Section 3.1.3 we present the experimental tests we carried out in order to verify the efficacy of the proposed method. We collect some conclusion and describe some still open issues to be further investigated in Section 3.1.4.

3.1.1 Block-wise JPEG compression and full-frame linear filtering

The JPEG standard provides a block-based compression scheme, which operates on 8×8 non-overlapping blocks of DCT coefficients. In its most commonly used format, it is well known to be a lossy scheme, i.e., some information is lost during the process, mainly due to a quantization operation.
In particular, an image I of size $(M_x \times M_y)$, here just considering the luminance channel for simplicity, is firstly partitioned into $(B_x \times B_y)$ non-overlapping blocks of size 8×8. Each block (b_x, b_y) is then independently transformed from the spatial to the frequency domain, using the DCT:

$$d^{(b_x,b_y)}(i,j) = \frac{c(i)}{2}\frac{c(j)}{2}\sum_{n_1=0}^{7}\sum_{n_2=0}^{7} o^{(b_x,b_y)}(n_1,n_2)\cos\left(\frac{2n_1+1}{16}\pi i\right)\cos\left(\frac{2n_2+1}{16}\pi j\right), \tag{3.1}$$

where $d^{(b_x b_y)}(i,j)$ is the frequency coefficient at position (i,j) in the $(b_x b_y)$-th block, with $(i,j) \in \{0,\ldots,7\}^2$, and $o^{(b_x b_y)}(n_1,n_2)$ is the pixel at position (n_1,n_2) in the $(b_x b_y)$-th block of the input image. Moreover, $c(s) = 1$ if $s > 0$, and $c(s) = 1/\sqrt{2}$ if $s = 0$.
Depending on the compression quality, a specific 8×8 quantization table is employed to quantize each DCT frequency:

$$d_q^{(b_x,b_y)}(i,j) = \text{round}\left(\frac{d^{(b_x,b_y)}(i,j)}{\Delta(i,j)}\right),$$

where $\Delta(i,j)$ are the quantization steps associated with each frequency (i,j). Such quantized coefficients are finally entropy-encoded (Huffman coding) and stored in the JPEG file format. The compressed data stream can be decompressed, applying all the steps in reverse order. Specifically, a DCT coefficient is reconstructed as

$$\hat{d}_q^{(b_x,b_y)}(i,j) \quad = \quad \Delta(i,j)\cdot\text{round}\left(\frac{d^{(b_x,b_y)}(i,j)}{\Delta(i,j)}\right), \tag{3.2}$$

and is finally transformed from the frequency to the spatial domain by applying the Inverse DCT on each 8×8 block:

$$J^{(b_x,b_y)}(i,j) = \sum_{k_1=0}^{7}\sum_{k_2=0}^{7}\frac{c(k_1)}{2}\frac{c(k_2)}{2}\hat{d}_q^{(b_x,b_y)}(k_1,k_2)\cos\left(\frac{2i+1}{16}\pi k_1\right)\cos\left(\frac{2j+1}{16}\pi k_2\right), \tag{3.3}$$

where $J^{(b_x,b_y)}(i,j)$ is the pixel value at position (i,j) in the (b_x,b_y)-th block of the JPEG image.
From (3.2), the de-quantized coefficients are mapped to multiples of the quantization step $\Delta(i,j)$, resulting in specific artifacts in the coefficient distribution. Fig. 3.1(a) shows the histogram of the DCT coefficients at frequency $(1,2)$ collected from a subset of 669 un-compressed images [33], while Fig. 3.1(b) depicts the distribution of the same data after quantization, with $\Delta(1,2) = 10$. It becomes clear that the structure of such histogram is related to the employed quantization factor. For the sake of presentation we disregard the round-off and truncation errors, in the pixel domain, introduced by the compression scheme, without affecting the conducted analysis. Previous works in the literature demonstrate that, by exploiting the described artifacts in the distribution, it is possible to discover instances of previous JPEG compression and even estimate the used quantization steps [25, 27, 34, 35]. However, it is very likely that later in its life the JPEG-compressed image will be processed by a further operator to enhance its quality. For example, it is frequent the use of full-frame operators aimed at reducing the JPEG-compression block effects. By doing so, the

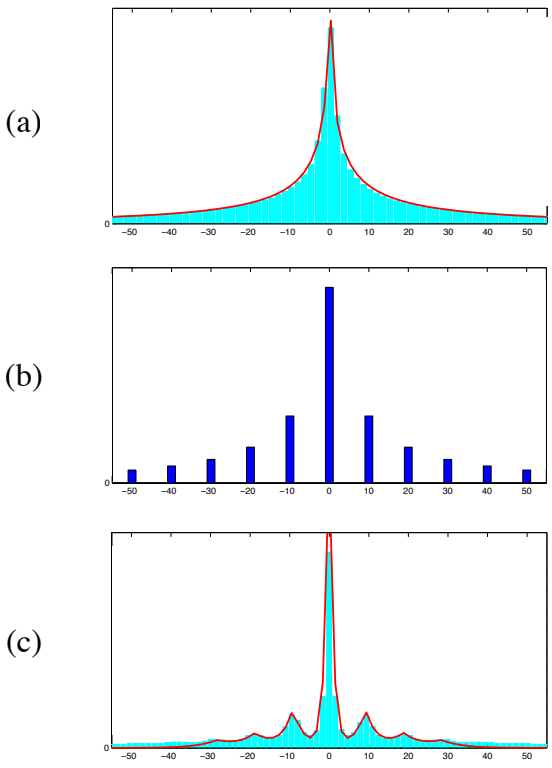


Figure 3.1: Panel (a) shows the original DCT histogram for frequency $(1, 2)$ collected from a subset of 669 uncompressed images and its curve fitting. In panel (b), the same distribution after quantization with $\Delta(1, 2) = 10$ is presented. Panel (c) shows the given distribution when a linear filtering operator (an average filter of size 3×3) has been further applied, together with the derived model for such distribution (red line).

characteristic artifacts present in the DCT distribution of JPEG images may be partially perturbed, making harder the application of those forensic algorithms. In this work, we study the combination of JPEG compression and full-frame linear filtering operations. Up to now, very little attention has been paid to study such a scenario. Specifically, linear filtering represents an interesting case study since it is a very common and useful linear tool applied for image enhancement, such as edge sharpening, noise removal illumination correcting and deblurring. It operates by convolving the original image with an appropriate filter kernel. The result of such a convolution is a filtered image, whose pixel values are a weighted sum of a certain number of neighboring pixels:

$$F(x, y) = \sum_{s_1=-N}^{N} \sum_{s_2=-N}^{N} h(s_1, s_2) J(x + s_1, y + s_2),$$

where $J(x, y) \triangleq J^{(b_x, b_y)}(i, j)$, $i = x \bmod 8$, $j = y \bmod 8$, $b_x = \lceil \frac{x}{8} \rceil$, $b_y = \lceil \frac{y}{8} \rceil$, $x \in \{0, ..., M_x - 1\}$ and $y \in \{0, ..., M_y - 1\}$, and h is the filter kernel of size $(2N + 1) \times (2N + 1)$. Fig. 3.1(c) shows the histogram of the DCT frequency coefficients of panel (b) after filtering with an Average 3×3

filter; the characteristics of the histogram of the quantized coefficients are clearly perturbed, but new patterns appear, depending both on the employed quantization factor and the filter kernel.
The aim of this research is to study such artifacts in order to identify the filter kernel a JPEG image has undergone. In order to do this, we mathematically analyze the DCT distribution of compressed and filtered images and derive an accurate model for them. Specifically, we investigate the statistical characteristics in the distribution of the DCT coefficients and, as a first result of that analysis, show that the extended assumption about the image AC DCT coefficients for different frequencies being independent, and for the same frequency being i.i.d. [36][37], does not hold; indeed, the inter- and intra-block redundancy of the quantized DCT coefficients must be taken into account. By considering those redundancies on the provided analysis, the studied processing can be accurately modeled and a general mathematical model for the distributions of DCT coefficients of JPEG filtered images can be derived, depending on the applied kernel and assuming the quantization to be fixed and known. Fig. 3.1(c) serves as an example of the derived model for the probability distribution of JPEG filtered images. Later, we build a dictionary-based database for the derived models corresponding to different applied filters and a distinguishability measure is calculated to quantify the difference between the theoretically derived models and the actual distribution of an image. In this analysis, we employ the χ^2 histogram distance, which comes from the χ^2 test [38], commonly used to compare observed frequencies with a theoretical distribution (e.g., it has been successfully used in steganography to detect if some embedding has taken place [31]). For discrete signals, the χ^2 distance is defined as follows:

$$\chi^2 = \frac{1}{2}\sum_i \frac{(P_i - Q_i)^2}{(P_i + Q_i)}, \tag{3.4}$$

where P and Q are two probability distributions to be compared. Intuitively, the χ^2 distance would tend to zero when the two distributions are very close. So, in our scheme, a lower χ^2 measure will be an evidence that will allow to identify the filter operator applied to the under-test image.

3.1.2 Mathematical model

We derive a theoretical model to describe the statistical properties of an image that has been first JPEG compressed and then linearly filtered. In order to do that, we mathematically express the deterministic relation between the quantized DCT coefficient $\hat{d}_q(x, y)$ and that of the JPEG and filtered image $d_f(x, y)$. Then, by exploiting the knowledge about the statistical properties of the distribution of $\hat{d}_q(x, y)$, we further analyze the histograms of $d_f(x, y)$, assess the dependency of the different frequencies and derive a model to theoretically characterize the probability distribution of the DCT coefficients of a JPEG image filtered with a given filter kernel.
The study case here concerned is shown in Fig. 3.2. Following the scheme backwards, we start considering the DCT coefficients $d_f(x, y)$ of a JPEG compressed and filtered image and, according to (3.1), transform them into the spatial domain $F(x, y)$. Through a linear convolution operation with the filter kernel $\mathbf{h}$, we can further express $d_f(x, y)$ as a function of the pixels of the compressed image J, which represents the Inverse DCT of $\hat{d}_q(x, y)$ (3.3). Exploiting the linearity property of both the filtering operation and the DCT transform, we are able to mathematically formulate the relation between the DCT coefficients of a quantized image $\hat{d}_q(\cdot, \cdot)$ and the DCT coefficients of the further filtered image $d_f(\cdot, \cdot)$. Specifically, given a filter kernel of size smaller than or equal to 17, the coefficients $\hat{d}_q(x, y)$ contributing in the calculation of $d_f(x, y)$ are those from the same block of $d_f(x, y)$ plus those from the 8 immediate surrounding blocks, resulting in 24×24 coefficients. It becomes clear that for filter kernels of sizes larger than 17, the number of contributing coefficients

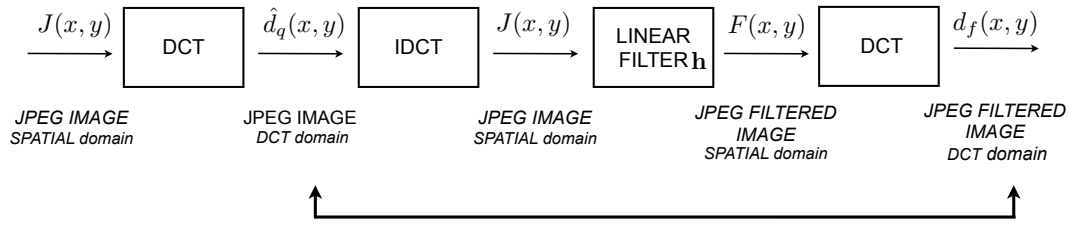


Figure 3.2: Block scheme of the considered processing operations. The aim is to establish a mathematical relationship between the quantized DCT coefficient $\hat{d}_q(x,y)$ and that of the JPEG and filtered image $d_f(x,y)$ in order to exploit the statistical properties of the distribution of $\hat{d}_q(x,y)$.

$$\begin{aligned} d_f(x,y) =& \mathrm{DCT}_{\alpha,\beta}\left(\left[\mathbf{h} * \mathrm{IDCT}_{\alpha,\beta}\left(\hat{d}_q(x,y)\right)\right]_{\lfloor\frac{x}{8}\rfloor,\lfloor\frac{y}{8}\rfloor}^{\lfloor\frac{x}{8}\rfloor+7,\lfloor\frac{y}{8}\rfloor+7}\right) \\ &+ \sum_{(k_1,k_2)\in\{-8,\ldots,15\}^2,(k_1,k_2)\neq(\alpha,\beta)} \mathrm{DCT}_{\alpha,\beta}\left(\left[\mathbf{h} * \mathrm{IDCT}_{\alpha',\beta'}\left(\hat{d}_q\left(\left\lfloor\frac{x}{8}\right\rfloor + k_1, \left\lfloor\frac{y}{8}\right\rfloor + k_2\right)\right)\right]_{\lfloor\frac{x}{8}\rfloor-\lfloor\frac{k_1}{8}\rfloor,\lfloor\frac{y}{8}\rfloor-\lfloor\frac{k_2}{8}\rfloor}^{\lfloor\frac{x}{8}\rfloor-\lfloor\frac{k_1}{8}\rfloor+7,\lfloor\frac{y}{8}\rfloor-\lfloor\frac{k_2}{8}\rfloor+7}\right). \end{aligned} \tag{3.6}$$

would increase, involving more surrounding blocks. In Section 3.1.2.1, we will show that only a subset of all the DCT coefficients is effectively relevant for the computation of each coefficient $d_f(x,y)$. Finally, we make explicit the contribution of the DCT coefficient $\hat{d}_q(x,y)$ at the same position of $d_f(x,y)$, as follows:

$$d_f(x,y) \quad = \quad \gamma \cdot \hat{d}_q(x,y) + N, \tag{3.5}$$

where $\gamma, N \in \mathbb{R}$ are a scaling factor and a noise term, respectively. These two terms can be calculated, through some math, according to (3.6), where $\alpha = x \mathrm{mod} 8$, $\beta = y \mathrm{mod} 8$, $\alpha' = k_1 \mathrm{mod} 8$, $\beta' = k_2 \mathrm{mod} 8$, $\mathrm{DCT}_{x,y}$ is the (x,y)-th DCT coefficient obtained from an 8×8 pixel block, $\mathrm{IDCT}_{x,y}$ is the 8×8 pixel block (located at $\{\lfloor\frac{x}{8}\rfloor, \ldots, \lfloor\frac{x}{8}\rfloor + 7\} \times \{\lfloor\frac{y}{8}\rfloor, \ldots, \lfloor\frac{y}{8}\rfloor + 7\}$) obtained by applying the IDCT to the (x,y) DCT coefficient, $*$ denotes the bidimensional convolution, and $[\mathbf{A}]_{a,b}^{c,d}$ denotes the submatrix of an arbitrary matrix $\mathbf{A}$ with first index taking values in $\{a, \ldots, b\}$, and second index in $\{c, \ldots, d\}$. N stands for the second term in the summation in (3.6).

3.1.2.1 Probability distribution

Once we have derived the deterministic expression in (3.5) for $d_f(x,y)$, we can exploit the knowledge about the distribution of the quantized coefficients $\hat{d}_q(x,y)$ to analyze the distribution of the DCT coefficients of the final image F.
Usually, the probability distribution of DCT coefficients in natural images is modeled as a zero-mean

Generalized Gaussian (Fig. 3.1(a)):

$$f_{GGD}(d(i,j)) = \frac{s}{2a\Gamma(1/s)}\exp -(|t|/a)^s, \tag{3.7}$$

where Γ denotes the gamma function, a and s are the scale and the shape parameter, respectively. Due to quantization, the probability distribution of each quantized DCT coefficient will be [25] (Fig. 3.1(b))

$$L_\lambda(k\Delta) \triangleq f(\hat{d}_q(i,j) = k\Delta|\Delta) = \int_{(k-\frac{1}{2})\Delta}^{(k+\frac{1}{2})\Delta} f_{GGD}(\tau)d\tau, \tag{3.8}$$

where $k \in \mathbb{Z}$ and, for the sake of notation simplicity, $\Delta = \Delta(i,j)$. Therefore the probability mass function of each frequency coefficient of a JPEG image is

$$f(\hat{d}_q(i,j) = \tau|\Delta) = \sum_k \delta(\tau - k\Delta)L_\lambda(k\Delta). \tag{3.9}$$

It becomes clear that the distribution of (3.9) presents specific artifacts, whose structure is related to the quantization step. In particular, the DCT coefficients corresponding to the (i,j)-th frequency will be located at multiples of the applied quantization step $\Delta(i,j)$, as illustrated in Fig. 3.1(b).
From probability theory [39], given two discrete independent random variables, the probability density function (pdf) of their sum is the convolution of their corresponding pdfs. Therefore, according to the derived mathematical model in (3.5), and based on the common DCT coefficients models, which typically assume the different frequency components to be independent and the coefficients in a given frequency to be i.i.d. [40], we would expect the probability distribution of the DCT coefficients $d_f(x,y)$ to be the result of a convolution between a train of impulses located at $\gamma \cdot k\Delta(i,j)$, with $\gamma \in \mathbb{R}$ and $k \in \mathbb{Z}$, and a noise component due to the contributions of all the neighboring coefficients (3.5). Moreover, we will model the noise components as a GGD distributed variable, according to (3.7), with GGD parameters depending on each centroid $\gamma \cdot k\Delta(\cdot,\cdot)$ about which such noise is centered.
However, we will illustrate that indeed the typical assumptions on the DCT coefficients distribution of natural images do not hold, thus resulting in a deviation between the classical theoretical models and the empirical data. Specifically, a scaling between the peaks in the histogram of the DCT coefficients of the compressed and filtered image and the impulse train identifying the location of the translated quantization step $\gamma \cdot k\Delta(\cdot,\cdot)$ is observed. This suggests the need for a different model for the noise component in (3.5), which cannot any longer be considered as the addition of independent variables (coefficients of different frequencies) and i.i.d. components (coefficients in the same frequency). Therefore, we analyze the mean of the noise component and verify that for real images it monotonically increases with the quantized samples value. As an example, in Fig. 3.3 the red curve represents the mean of the noise component when a Moving Average filter of size 3×3 is applied, plotted with respect to each translated quantized value $\gamma \cdot k\Delta(i,j)$; similar behaviors have been verified for different filter kernels. Moreover, for each coefficient $d_f(x,y)$, we isolate the contribution of each 24×24 coefficient and analyze their influence. In Fig. 3.3 (a)-(b) we show the specific pattern of coefficients mainly contributing to the noise, for all the AC coefficients $(1,2)$ and $(2,1)$. The red curve represents the mean of the total noise component over $\gamma \cdot k\Delta(\cdot,\cdot)$, the blue curve represents the contribution of a specific set of coefficients and the black curve corresponds to the contribution of all the remaining coefficients not specified in the previous set. This set was determined by isolating those coefficients that provided a significant noise contribution, in absolute value, over all $\gamma \cdot k\Delta(\cdot,\cdot)$ (i.e., above an empirically determined threshold.) Note that the 24×24 grid

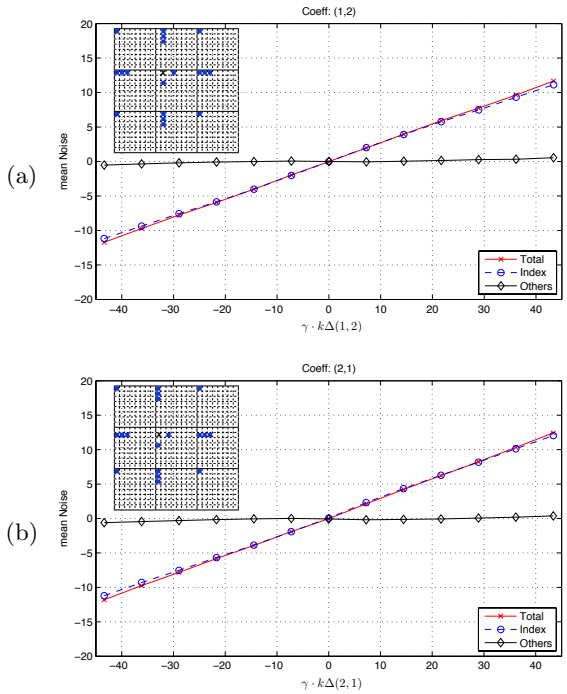


Figure 3.3: Inter- and intra-block spatial redundancy affecting to $d_f(x, y)$, over the entire image database. In each panel, corresponding to the frequencies $(1, 2)$ and $(2, 1)$, respectively (black cross in the grid in the upper left of each subplots), the total mean of the noise component (red curve) is plotted with respect to the values of the quantized filtered coefficients $\gamma \cdot k\Delta q(i, j)$. The blue curve represents the contribution of a set of coefficients (depicted in blue in the grid) and the black curve corresponds to the contribution of all the remaining coefficients not specified in the previous set.

in the upper left part of each plot corresponds to the 9 DCT blocks taken into account in (3.5), when employing a kernel filter of size smaller than or equal to 17. The black dot identifies the considered frequency and the blue dots correspond to set of coefficients which mainly influence the total noise, as verified by the curve matching. Note that due to space restrictions, we report only the behavior for the coefficients $(1, 2)$ and $(2, 1)$, but the 8 lowest frequencies in an 8×8 DCT block, discarding higher frequencies which are more likely to be quantized to zero, have been analyzed giving similar results.

Therefore, an accurate model for the distribution of the DCT coefficients of a filtered JPEG image can be derived, taking into account the scaling inferred by the noise component. In Fig. 3.4 it is shown how the translated impulses, now centered in $\gamma \cdot k\Delta(\cdot, \cdot)$ plus the mean of the noise component, match with the peaks of the histogram.

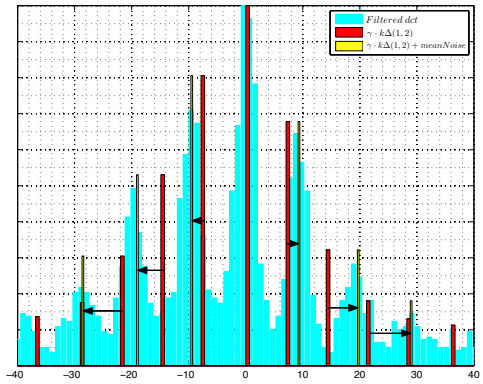


Figure 3.4: Probability distribution, at frequency $(1, 2)$, quantized with a step $q(1, 2) = 10$ and filtered with a 3×3 averaging filter. The red impulses represent the location of $\gamma \cdot k\Delta(1, 2)$ while the yellow ones are translated by the mean of the Noise component. The latter perfectly match with location of the peaks in the histogram.

3.1.3 Proposed forensic approach

The main idea of the proposed forensic approach is to estimate the filter operator an image has been gone through. We note that linear filtering is a very powerful tool employed for image enhancement. As a first attempt in this forensic case study, we assume the quantization applied to the image during compression to be known a priori. Future work will release this assumption. The main idea is to model the probability distributions of the DCT coefficients of an image being filtered with one of the linear filters present in a predefined dictionary.
To build a generalized model associated with each filter in the dictionary, we proceed as follows:

- We collect the $(1, 2)$, $(2, 1)$ and $(2, 2)$ DCT frequency coefficients from a random half of the 1338 images present in the UCID- Uncompressed Image Database [33]. We then fit, for each frequency, a Generalized Gaussian distribution, as in (3.7). This fitting is as shown in Fig. 3.1(a).
- We evaluate the distribution of the quantized coefficients, according to (3.8). In this work we simulated a uniform scalar quantizer with step size $\Delta = 10$ (Fig. 3.1(b)).
- We select a set of linear filters to be part of the dictionary, among which, Moving Average, Gaussian, Laplacian, with different settings for the window size, the variance σ^2 or the scale parameter α, as reported in Tab. 3.1.
- For each filter and AC DCT coefficient, we calculate γ in (3.5), according to (3.6).
- For each $\gamma \cdot k\Delta(\cdot, \cdot)$ the corresponding noise component is modeled as a GGD.
- The distribution of DCT coefficients, quantized and filtered with a given kernel, will be the sum of many GGDs, each of them centered in $\gamma \cdot k\Delta(\cdot, \cdot)$ translated by the mean of the noise component and with amplitude depending on the distribution of the quantized and not filtered coefficients, i.e., (3.8).

An example of the generalized model for the average filter with window size 3×3 is reported in Fig. 3.1(c).

1. LP Average [3 × 3]	3. LP Gaussian [3 × 3], σ^2= 0.5	9. LP Laplacian, σ^2= 0.2
2. LP Average [5 × 5]	5. LP Gaussian [3 × 3], σ^2= 0.5	10. LP Laplacian, σ^2= 0.7
4. LP Gaussian [3 × 3], σ^2= 1	7. HP Laplacian, σ^2= 0.2	11. HP Average [3 × 3]
6. LP Gaussian [5 × 5], σ^2= 1	8. HP Laplacian, σ^2= 0.7	12. HP Average [5 × 5]

Table 3.1: Filters selected to be part of the dictionary-based filter database, grouped according to the similarity of their frequency response.

To verify the distinguishability among the derived models for the considered filters, we calculate the χ^2 distance, as defined in (3.4). Following [32], we combine the independently calculated comparisons for each analyzed frequency coefficient by summing their values, according to Minkowski norm:

$$\chi^2_{tot} = \sum_r (\chi^2_r(P,Q))^p$$

where χ^2_r is the distance, as in (3.4), between distributions P and Q calculated for the r-th frequency coefficient and p has been empirically shown to be optimal if set equal to 1. As a result, a good distinguishability has been observed, even if very low χ^2 values may mislead the correct classification of the filter, as in the case of χ^2 among filters (1)-(2)-(4)-(6). This is due to the similarity of the frequency response of those filters for the analyzed DCT coefficients, and consequently, of their corresponding models. This issue is not specific of the presented framework, but a general constraint. Tab. 3.1 shows groups based on the similarity of the frequency response of the filters selected to be part of the dictionary.

Given the models for all the distributions corresponding to a specific filter, the performance of the proposed algorithm is verified in terms of percentage of correct classification over the image database. To each image in the database, not previously used to build the models, we apply different compressions with quality factors $QF \in \{40, 50, 60, 70, 80, 90\}$ and post-process them with each of the filter kernels present in the dictionary. We then compare the obtained DCT histogram, for each of the selected frequency, with all the corresponding DCT coefficient pdfs derived in the steps described above. As a preliminary study we selected some frequency coefficients to be a representative set of low, medium and high frequencies. As a future work, we plan to further explore the effect of different frequency coefficients.

The estimated applied filter is that providing the minimum χ^2 distance. Correct classification results are reported in Tab. 3.2. Accuracy is reported for each of the 12 filters in the dictionary, when a given quality factor QF is applied. For each compression, a set of frequency coefficients is specified, indicating the best combination of the considered coefficients which better help identifying the applied filter in terms of the highest accuracy. Finally, the average accuracies for each quality factor are reported. These results are very promising and show the efficacy of the proposed technique.

Intuitively, lower frequencies will be more significant when dealing with low-pass filters, while higher frequencies will be needed to correctly identify high-pass filters. Based on this idea, we performed classification by building two separated filters database, composed by low-pass and high-pass filters, respectively. Results are reported in Tabs. 3.3-3.4, where an improved accuracy for each set of filters is proven.

QF	40 [%]	50 [%]	60 [%]	70 [%]	80 [%]	90 [%]
Filter \ Coeffs	(1,2), (2,1) (6,3)	(1,2), (2,1) (6,4), (6,1)	(1,2), (2,1) (6,4), (6,1)	(1,2), (2,1) (5,4),(6,4),(6,1)	(2,1) , (6,4) (6,1)	(1,2), (5,4) (6,3),(6,4),(6,1)
1	81.8	71.7	82.8	68.2	71.7	82.4
2	96.3	98.1	99.1	92.8	92.4	94.9
3	73.6	91.9	86.7	91.5	89.8	82.5
4	82.2	59.2	70.4	66.5	68.3	77.1
5	73.5	92.1	86.7	91.5	89.8	82.7
6	93.6	85.1	93.4	79.8	80.7	84.6
7	97.6	96.7	94.8	83.0	64.0	34.8
8	93.9	93.8	94.0	82.2	63.8	52.8
9	98.9	98.8	99.0	98.8	99.2	95.5
10	98.9	98.7	99.0	98.4	99.1	93.4
11	68.0	68.5	61.4	58.9	71.2	68.8
12	98.4	96.6	94.17	92.4	88.3	77.9
Mean	88.05	87.6	88.5	83.7	81.5	77.3

Tab. 3.2: Percentages of correct classification when each filter is applied to each image in the database, compressed with different quality factors QF.

3.1.4 Conclusion

We have presented a mathematical model to characterize the DCT coefficients distributions of a full-frame linearly-filtered JPEG image. We explicitly express the theoretical relationship between the DCT coefficients before and after filtering and as a first result we show that, in the considered scenario, AC DCT coefficients for different frequencies cannot be any longer considered independent, nor those for the same frequency be i.i.d. By considering the inter- and intra-block redundancy of the quantized DCT coefficients, we have accurately analyzed the effect of the considered processing. The derived theoretical model allows building a dictionary-based database of theoretical distributions of quantized images being filtered with a given kernel. We then exploit such dictionary for estimating the filter given the quantization, by using the χ^2 histogram distance as target function. The presented framework represents a first attempt to analyze the effects of full-frame linear operations on block-based compressed images. Future work will be devoted to enlarge the dictionary for the employed filter kernels and to eliminate the assumption on the knowledge of the quantization, so that eventually this framework may be regarded as a forensically helpful means to jointly disclose the applied compression factors and the filter kernel. Moreover, we plan to develop tools enabling to decide if the considered filter is low-pass or high-pass.

QF	40 [%]	50 [%]	60 [%]	70 [%]	80 [%]	90 [%]
Coeffs / Filter	(1,2), (2,1) (2,2) (3,7)	(1,2),(2,2),(5,4) (6,4), (6,1)	(1,2), (2,2) (2,8)	(1,2) , (5,4) (6,4),(6,1)	(2,1) , (5,4) (6,4) , (6,1)	(6,3) , (6,4) (6,1)
1	90.3	82.9	91.8	78.8	80.0	93.4
2	98.8	98.6	98.95	99.1	98.2	86.1
3	95.2	88.9	94.8	94.9	90.1	91.0
4	80.4	69.5	85.8	73.8	77.6	87.0
5	65.2	88.9	94.8	95.1	90.3	91.0
6	95.4	90.6	96.4	84.75	85.8	91.9
9	99.6	98.9	99.9	97.9	99.1	92.4
10	99.4	98.5	99.7	96.6	97.9	88.5
Mean	94.3	89.7	95.3	90.1	89.8	90.2

Tab. 3.3: Percentages of correct classification when only low pass filters are considered.

QF	40 [%]	50 [%]	60 [%]	70 [%]	80 [%]	90 [%]
Coeffs / Filter	(8,1),(6,1) (1,8)	(8,1),(6,1) (2,8)	(8,1), (6,2) (2,8)	(2,1) , (8,2), (7,1) (6,1),(6,2) , (2,8)	(8,1) , (7,2) (3,8)	(1,2),(8,1),(8,2) (7,2),(6,1),(3,8)
7	97.6	97.3	98.1	99.3	97.6	98.4
8	96.4	96.3	96.9	98.8	96.7	97.5
11	99.7	98.9	99.3	98.6	99.9	98.5
12	100	99.9	100	99.8	98.7	96.9
Mean	98.4	98.1	98.5	99.2	98.2	97.8

Tab. 3.4: Percentages of correct classification when only high pass filters are considered.

3.2 Interpolation estimation

The problem of resampling factor estimation for tampering detection is addressed following the maximum likelihood criterion in this section. By relying on the rounding operation applied after resampling, an approximation of the likelihood function of the quantized resampled signal is obtained. From the underlying statistical model, the maximum likelihood estimate is derived for one-dimensional signals and a piecewise linear interpolation. The performance of the obtained estimator is evaluated, showing that it outperforms state-of-the-art methods.

3.2.1 Introduction

A well-known problem in this research area is the detection of resampling traces as a means to unveil the application of a geometric transformation and the estimation of the resampling factor for specifying the parameters of the applied transformation.
Seminal works addressing this topic [41, 42, 43], were focused on the detection of the particular correlation introduced between neighboring pixels by the resampling operation inherently present when a spatial transformation (e.g., scaling or rotation) has been performed.
Since the resampling operation can be modeled as a time-varying filtering that induces periodic correlations, links between this problem and the cyclostationarity theory have been established

in [44] and [45], providing a theoretical framework for the estimation of the parameters of the transformation. Within this framework, two different approaches have been proposed for finding the optimum prefilter that might be applied to a resampled image for achieving the best performance in the estimation of the resampling factor [46, 47].
At some point, all the mentioned approaches perform an analysis in the frequency domain for the detection or estimation of this periodic behavior, by looking at spectral peaks corresponding to underlying periodicities. Nevertheless, the frequency analysis presents some drawbacks: 1) a considerably large number of samples is needed to obtain reliable results; 2) the presence of periodic patterns in the content of the image usually misleads the detector and the estimator; and 3) the windowing effect impairs the performance of the mentioned methods when slight spatial transformations are employed (i.e., with a resampling factor near 1).
With these shortcomings in mind, in this section we will address the estimation of the resampling factor following the Maximum Likelihood (ML) criterion. The approximation of the likelihood function of the resampled signal will rely on the rounding operation applied after the resampling. Therefore, by correctly modeling the relationship between the distribution of the quantization noise and the quantized resampled signal, an optimum estimator of the resampling factor will be provided. The proposed approach will only consider one-dimensional (1-D) signals, but the idea can easily be extended to the two-dimensional case, to be applied to images. The three discussed drawbacks of the previous methods will be sorted out with the proposed estimator.

3.2.2 Preliminaries and problem formulation

A digital image forgery can be done in many different ways, but it usually involves cropping some region from a particular image and pasting it into a different one. The adjustment of this new content to a specific scene is commonly carried out by applying geometrical transformations (e.g., rotation or scaling) that inherently need to perform a resampling operation. Since the tampering should not introduce visible distortions, only slight transformations will be applied, thus requiring that the resampling estimator should achieve good performance for resampling factors near 1. This work just studies the case where the resampling factor is larger than 1. Of course, the use of resampling factors smaller than 1 are commonly used; however, the analysis is formally quite different, so we leave the study of such case for a future work.
The problem of resampling estimation is addressed for 1-D signals because the derivation of the Maximum Likelihood Estimate (MLE) of the resampling factor is more tractable and affordable than considering directly the two-dimensional (2-D) case. However, we will see in Section 3.2.3 that the obtained method following the ML criterion can be easily extended to the 2-D case. The same holds for the considered interpolation filter. The use of a piecewise linear interpolation scheme is a clear limitation of our work, which should be considered in this regard as a first attempt to introduce MLE principles in the resampling estimation problem. We notice that the methodology here introduced can be extended to include more general filters.

3.2.2.1 Notation

A time-dependent 1-D signal will be represented as $x(n)$. Random variables will be denoted by capital letters (e.g., X) and their realizations by lowercase letters (e.g., x). Random vectors will be represented with bold capital letters (e.g., $\boldsymbol{X}$), their outcomes with lowercase letters (e.g., $\boldsymbol{x}$) and each ith component will be denoted as x_i. The length of a vector $\boldsymbol{x}$ will be expressed as $L_x \in \mathbb{N}^+$ and, for convenience, the index i to identify each component of the vector will satisfy $i \in \{0, \ldots, L_x - 1\}$. A vector of length N starting from the nth component, will be denoted by $\boldsymbol{x}_n = (x_n, \ldots, x_{n+N-1})^T$.

Floor and ceiling functions will be represented by $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$, respectively. To denote the set of all integer numbers multiple of a given integer value n, we will use the notation $n\mathbb{Z}$. For a compact notation, we will use $\mathrm{mod}(a, b)$ to denote the modulo operation: a mod b.

3.2.2.2 Problem formulation

In the following, we will mathematically describe all the steps involved in the change of the sampling rate of a 1-D signal $x(n)$, by a resampling factor denoted by ξ. This description will allow us to set out an approach based on the ML criterion in Section 3.2.3, for the estimation of the applied resampling factor.

Let us start by defining the resampling factor ξ as the ratio between the applied upsampling factor L and downsampling factor M, i.e., $\xi \triangleq \frac{L}{M}$ with $L \in \mathbb{N}^+$ and $M \in \mathbb{N}^+$. To ensure a unique representation of ξ, we will consider that L and M are coprime, but note that this is not a limitation. As it was stated above, the possible range of values for the resampling factor will be $\xi > 1$. For this range of resampling factors, the general expression for a resampled signal $y(n)$ is given by the following relation with the original signal $x(n)$:

$$y(n) = \sum_k x(k) h\left(n\frac{M}{L} - k\right),$$

where $h(t)$ with $t \in \mathbb{R}$ represents the interpolation filter. As it was previously indicated, the interpolation filter used during the resampling process will be assumed linear, with the following impulse response

$$h(t) = \begin{cases} 1 - |t|, & \text{if } |t| \leq 1 \\ 0, & \text{otherwise} \end{cases}.$$

Therefore, considering this interpolation filter, each component of the resampled vector can be computed as the linear combination of at most two samples from the original signal,

$$y(n) = \begin{cases} x\left(\lfloor n\frac{M}{L}\rfloor\right)\left(1 - \mathrm{mod}\left(n\frac{M}{L}, 1\right)\right) + x\left(\lfloor n\frac{M}{L}\rfloor + 1\right)\mathrm{mod}\left(n\frac{M}{L}, 1\right) & , \quad \text{if} \quad n \notin L\mathbb{Z} \\ x\left(n\frac{M}{L}\right), & \text{if} \quad n \in L\mathbb{Z} \end{cases}.$$

Regarding the set of values that the original signal can take, we will consider that all the samples $x(n)$ have already been quantized by a uniform scalar quantizer with step size Δ, in order to fit into a finite precision representation. Even though the interpolated values $y(n)$ will be generally represented with more bits, a requantization to the original precision is often done prior to saving the resulting signal. This quantized version of the resampled signal, denoted by $z(n)$, will be expressed as

$$z(n) = \begin{cases} Q_\Delta\left(x\left(\lfloor n\frac{M}{L}\rfloor\right)\left(1 - \mathrm{mod}\left(n\frac{M}{L}, 1\right)\right) + x\left(\lfloor n\frac{M}{L}\rfloor + 1\right)\mathrm{mod}\left(n\frac{M}{L}, 1\right)\right) & , \quad \text{if } n \notin L\mathbb{Z} \\ x\left(n\frac{M}{L}\right), & \text{if } n \in L\mathbb{Z} \end{cases}, \tag{3.10}$$

where $Q_\Delta(\cdot)$ represents a uniform scalar quantization with step size Δ (i.e., the same one used for the original signal).

From the second condition in (3.10), it is evident that some of the original samples are "visible" in the quantized resampled version. On the other hand, the remaining values of the resampled signal are the combination of "visible" and "non-visible" samples from the original signal that are later quantized. This fact will help to define the likelihood function of the quantized resampled signal.

3.2.3 ML approach to resampling estimation

For the definition of the MLE of ξ, the original signal will be represented by the vector $\boldsymbol{x}$ with L_x samples and the corresponding quantized resampled signal by the vector $\boldsymbol{z}$ with L_z samples. For convenience, we will assume that the length of the original signal is $L_x = N + 1$ with N a multiple of M, and so, the corresponding length of the resampled signal will be $L_z = \xi N + 1$. We will find it convenient to model vectors $\boldsymbol{x}$ and $\boldsymbol{z}$ as outcomes of random vectors $\boldsymbol{X}$ and $\boldsymbol{Z}$, respectively.
Based on the above analysis, the estimation of the resampling factor $\hat{\xi}$ following the ML criterion relies on finding the value of ξ that makes the observed values of the quantized resampled vector $\boldsymbol{z}$ most likely. Nevertheless, given a vector of observations, their components z_i could be misaligned with the periodic structure of the resampled signal in (3.10). Hence, a parameter ϕ must be considered to shift the components of the vector, in order to align the periodic structure of z_i with $z(n)$. The possible values of ϕ lie in the range $0 \leq \phi \leq L - 1$. Therefore, the MLE of ξ becomes

$$\hat{\xi} = \arg\max_{\xi>1} \max_{0\leq\phi\leq L-1} f_{\boldsymbol{Z}|\Xi,\Phi}(\boldsymbol{z}|\xi,\phi).$$

Note that we are not considering a set of possible parameters for the interpolation filter because in the case of a piecewise linear interpolation, once we fix the resampling factor, then the filter is automatically determined (cf. Eq. (3.10)). On the other hand, given that the shift ϕ is not a determining factor for the derivation of the target function, for the sake of simplicity, we will assume that the vector of observations is correctly aligned and, thus, the MLE can be written as

$$\hat{\xi} = \arg\max_{\xi>1} f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi).$$

For the calculation of that joint probability density function (pdf) we will exploit the fact that some samples of the interpolated signal exactly match the original (cf. Eq. (3.10)), and also the linear relation established between the remaining samples.

3.2.3.1 Derivation of $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$

Along the derivation of the joint pdf $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$, for the sake of notational simplicity, we will refer to this one as $f_{\boldsymbol{Z}}(\boldsymbol{z})$, considering implicitly that we are assuming a particular resampling factor ξ. From the dependence between the quantized resampled signal and the original one, the joint pdf can be written in a general way as

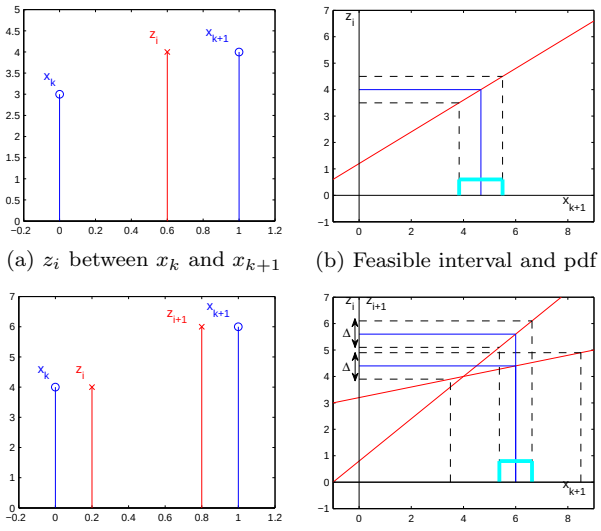
$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \int_{\mathbb{R}^{N+1}} f_{\boldsymbol{Z}|\boldsymbol{X}}(\boldsymbol{z}|\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}.$$

We assume that no a priori knowledge on the distribution of the input signal is available. This is equivalent to considering that $f_{\boldsymbol{X}}(\boldsymbol{x})$ is uniform and, consequently, the joint pdf can be approximated by the following relation

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) \approx \int_{\mathbb{R}^{N+1}} f_{\boldsymbol{Z}|\boldsymbol{X}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{x}.$$

Equation (3.10), indicates that every L samples of the observed vector $\boldsymbol{z}$, we have a visible sample from the original signal. This implies that the random variable Z_i, given X_k, is deterministic whenever $i \in L\mathbb{Z}$ and $k \in M\mathbb{Z}$. For this reason, the previous joint pdf can be obtained by processing $(L_z - 1)/L = N/M$ distinct and disjoint blocks, i.e.,

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) \approx \prod_{j=0}^{N/M-1} \int_{\mathbb{R}^M} f_{\boldsymbol{Z}_{Lj}|\boldsymbol{X}_{Mj}}(\boldsymbol{z}_{Lj}|\boldsymbol{x}_{Mj}) d\boldsymbol{x}_{Mj}, \tag{3.11}$$



(a) z_i between x_k and x_{k+1} (b) Feasible interval and pdf

(c) z_i, z_{i+1} between x_k and x_{k+1} (d) Feasible interval and pdf

Figure 3.5: Illustrative example, showing the last two possible cases for z_i. Pdfs obtained are shown graphically. Note that $\Delta = 1$.

where $\boldsymbol{Z}_{Lj}$ and $\boldsymbol{X}_{Mj}$ (and also their corresponding outcomes) are vectors of size L and M, respectively.
The calculation of the contribution of each block of L samples from the vector of observations $\boldsymbol{z}_{Lj}$ in (3.11), will depend on its relation with the corresponding M samples of the vector of the original signal, i.e., $\boldsymbol{x}_{Mj}$. This relation is determined by the assumed resampling factor ξ.
Therefore, considering an arbitrary sample z_i that will be linearly related with at most two original samples x_k and x_{k+1}, with $k \triangleq \left\lfloor i\frac{M}{L} \right\rfloor$ (cf. Eq. (3.10)), three cases are possible:

- z_i is a visible sample, thus deterministic. Consequently

$$f_{Z_i|X_k}(z_i|x_k) = \delta(z_i - x_k),$$

  where $\delta(\cdot)$ represents the Dirac delta.

- z_i is the only sample between two original ones as it is shown in Fig. 3.5(a). In this case, if the variance of the original signal is large enough with respect to the variance of the quantization noise, then the quantization error can be considered uniform (we will call this the "fine-quantization assumption"), and the obtained pdf is

$$f_{Z_i|X_k,X_{k+1}}(z_i|x_k, x_{k+1}) = \Pi\left(\frac{a_i x_k + b_i x_{k+1} - z_i}{\Delta}\right),$$

  where $\Pi(t)$ denotes a rectangular pulse that is 1 if $t \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ and 0 otherwise. In this case, for the sake of clarity, we have used $a_i \triangleq (1 - \mathrm{mod}(i\frac{M}{L}, 1))$ and $b_i \triangleq \mathrm{mod}(i\frac{M}{L}, 1)$, obtained from (3.10). A graphical representation, depicted in Fig. 3.5(b), shows how the rectangular pdf is derived from z_i.

- z_i is one of several resampled values between two original samples, as it is shown in Fig. 3.5(c). As before, the following pdf is valid if the fine-quantization assumption holds, hence

$$f_{\boldsymbol{Z}_i|X_k,X_{k+1}}(\boldsymbol{z}_i|x_k,x_{k+1}) = \prod_m \Pi\left(\frac{a_m x_k + b_m x_{k+1} - z_m}{\Delta}\right),$$

  where m will increase from i to the number of resampled values located between the two original samples. Fig. 3.5(d) shows the resulting pdf for the considered example.

Each time we obtain the pdf for a particular z_i (or a group of them), the corresponding integral in (3.11) must be evaluated with respect to the corresponding original sample x_k. Intuitively, we can observe that the calculation of (3.11) will finally be the convolution of several rectangular functions, leading to a feasible and easy implementation. Note that those uniform distributions are obtained only if the fine-quantization assumption holds. Given the importance of this assumption, its effect on the performance of the MLE will be analyzed in Section 3.2.4.

3.2.3.2 Method description

For a better understanding on how the obtained MLE can be easily implemented, we will exemplify the calculation of the target function $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$ when a particular resampling factor ξ_t is tested. In this illustrative example we will consider a vector of observations $\boldsymbol{z}$ (already aligned), corresponding to a signal that has been resampled by a factor $\xi = \frac{5}{3}$. In Fig. 3.6(a), an example of this vector of observations is shown, along with the corresponding vector of original samples $\boldsymbol{x}$. In the mentioned figure, solid lines are used for representing the resampled values (consequently, also the original samples that are visible), while dashed lines are used for representing the non-visible samples of the original signal.
Since the calculation of the target function $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$ can be split by processing blocks of L samples of the observed vector, in this example, we will show how to process a single block. For the calculation of the remaining blocks, the same process should be repeated. Assuming that the resampling factor under test is $\xi_t = \frac{5}{3}$, these are the followed steps:

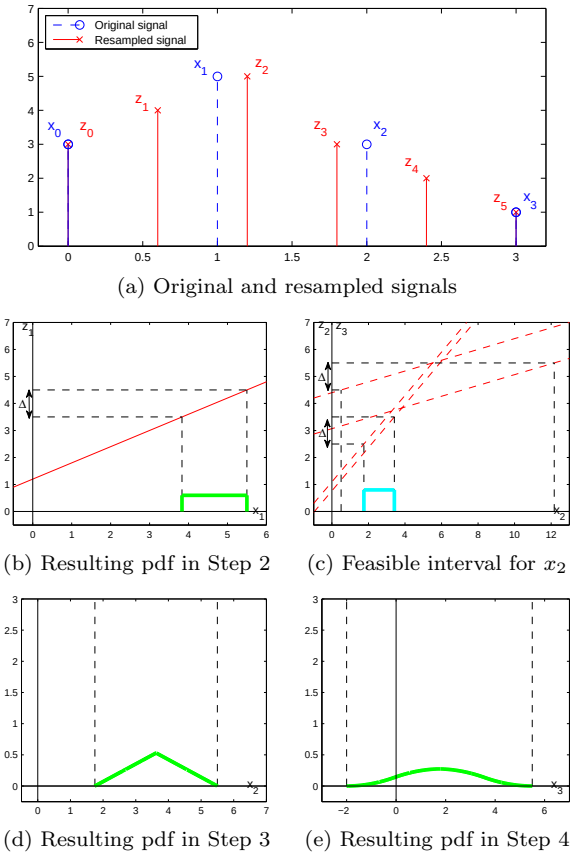
1. The first sample z_0 is a visible one, then we know that $z_0 = x_0$ and, thus, $f_{Z_0|X_0,\Xi}(z_0|x_0,\xi_t) = \delta(z_0 - x_0)$.

2. The second sample z_1 is located between two original samples, i.e., the visible x_0 and the non-visible x_1. Hence, we have $f_{Z_1|X_0,X_1,\Xi}(z_0|x_0,x_1,\xi_t) = \Pi\left(\frac{a_1 x_0 + b_1 x_1 - z_1}{\Delta}\right)$.

   Fig. 3.6(b) shows with a red line the linear relation between the interpolated value and the original ones $y_1 = a_1 x_0 + b_1 x_1$, with the value of x_0 fixed, i.e., from the previous step $x_0 = z_0$. From the value of z_1 we obtain the feasible interval of x_1 (represented with dashed black lines). Finally, the resulting pdf after the convolution of the rectangular function with the delta obtained in Step 1 is plotted in green.

3. The third and fourth samples, z_2 and z_3, are located between the two original samples x_1 and x_2. In this case, we have seen that $f_{\boldsymbol{Z}_2|X_1,X_2,\Xi}(\boldsymbol{z}_2|x_1,x_2,\xi_t) = \Pi\left(\frac{a_2 x_1 + b_2 x_2 - z_2}{\Delta}\right)\Pi\left(\frac{a_2 x_1 + b_2 x_2 - z_3}{\Delta}\right)$.

   Fig. 3.6(c) shows in this case the corresponding two linear relations for $y_2 = a_2 x_1 + b_2 x_2$ and $y_3 = a_3 x_1 + b_3 x_2$. Be aware that in this case x_1 can take any value in the range obtained in Step 2, and that is the reason why the dashed red lines are plotted. From the product of the two rectangular pdfs, we obtain the feasible interval for x_2 (whose pdf is represented in cyan).



(a) Original and resampled signals

(b) Resulting pdf in Step 2

(c) Feasible interval for x_2

(d) Resulting pdf in Step 3

(e) Resulting pdf in Step 4

Figure 3.6: Graphical representation of the method description. Note that $\Delta = 1$.

At this point, it is important to note that when the resampling factor under test does not match the true one, the previous product of rectangular pdfs could lead to an empty feasible set for x_2. If this happened, then we would automatically infer the infeasibility of the tested resampling factor, so the estimation algorithm would move to the next resampling factor in the candidate set.

If the factor cannot be discarded, then we must compute the convolution of the uniform pdf here obtained with the one resulting from Step 2. The result is plotted in green in Fig. 3.6(d).

4. The fifth sample z_4 is processed in the same way as in Step 2, but considering that now the linear relation $y_4 = a_4 x_2 + b_4 x_3$ must be evaluated with the set of possible values of x_2. Proceeding this way, we obtain the feasible interval for x_3 and the corresponding pdf. Both are shown in Fig. 3.6(e).

5. At this point, we have finished processing the L samples in the block and we have the resulting pdf as a function of x_3. Since the next sample is visible, i.e., $z_5 = x_3$, to determine the contribution of these L samples to the target function $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi_t)$, we evaluate the resulting pdf taking into account the actual value of z_5.

As before, if the value of z_5 falls outside the possible range of x_3, then the resampling factor under test is discarded.

Following this procedure, the maximization of the target function $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$ is performed over the set of candidate resampling factors $\xi > 1$ that have not been discarded, achieving the MLE $\hat{\xi}$. After this qualitative explanation, it is clear that the 2-D extension of this method is straightforward.

3.2.4 Experimental results

The experimental validation of the obtained MLE is divided in two parts. In the first one, the performance of the estimator is evaluated by using synthetic signals and its behavior in terms of the fine-quantization assumption is analyzed. In the second part, natural 1-D signals from the audio database in [48] (which contains different music styles) are used to test the estimator in a more realistic scenario. To confirm that the described method is able to sort out the drawbacks pointed out in the Introduction, comparative results with a 1-D version of the resampling detector proposed by Popescu and Farid in [41] are also provided.

3.2.4.1 Performance analysis with synthetic signals

In this case, we consider as synthetic signal a first-order autoregressive (AR) process, parameterized by a single correlation coefficient ρ. The AR(1) model is commonly used for characterizing the correlation between samples of natural signals, where the value of ρ adjusts the model. Typically, close to 1 values are considered for modeling natural signals, as it is done with images [49]; hence, $\rho = 0.95$ will be used in the following simulations. The AR(1) process has the following form

$$u(n) = w(n) + \rho u(n-1),$$

where $w(n)$ is a Gaussian process with zero mean and variance σ_W^2. Note that in this case, the process $w(n)$ is actually the innovation from one sample to another of the AR(1) process, so results will be drawn as a function of σ_W^2 to evaluate the validity of the fine-quantization assumption.
To reproduce the conditions of the considered model, the original signal $x(n)$ is obtained by quantizing the generated AR(1) process, i.e., $x(n) = Q_\Delta(u(n))$ with $\Delta = 1$. Regarding the set of considered resampling factors, for the sake of simplicity, we use a finite discrete set, obtained by sampling the interval $(1, 5]$ with step size 0.05 (from 1.05 to 2) and 0.5 (from 2 to 5). Be aware that we use the same set for the true resampling factor ξ and the values tested by the ML estimator, ξ_t. We consider that the estimation of the resampling factor is correct if $\hat{\xi} = \xi$, i.e., if the estimated value is indeed the one used for resampling the original signal, up to the precision used when griding ξ and ξ_t. For all the experiments, the length of the vector of observations is $L_z = 400$.
Fig. 3.7 shows the percentage of correct estimation for some of the resampling factors in the set as a function of σ_W^2. From this plot, we can observe that the performance of the estimator strongly depends on the mentioned variance of innovation, as well as on the true resampling factor used. For instance, by resampling the AR process with $\xi = 5$, a very small value for the variance of innovation ($\sigma_W^2 = 0.5$), is required to correctly estimate the resampling factor for all the experiments; nevertheless, for $\xi = 2$, almost a value of $\sigma_W^2 = 50$ will be necessary for getting the same estimation performance. In general, and in accordance with the assumptions backing the analysis introduced in the previous section, the higher σ_W^2, the better the estimation will be.
Although ML-based estimators are frequently thought to be computationally demanding, if the fine-quantization assumption holds, then the estimation proposed in the previous section is very cheap and only a few samples are required for correctly estimating the actual resampling factor. Remember

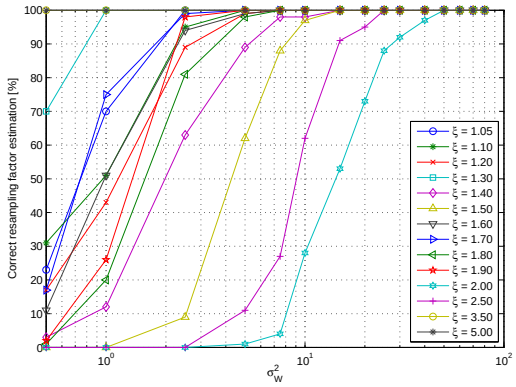


Figure 3.7: Correct resampling factor estimation percentage for different resampling factors as a function of σ^2_W. $\rho = 0.95$, and 500 Monte Carlo realizations are considered.

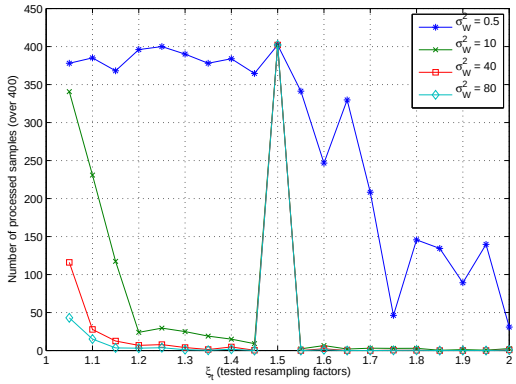


Figure 3.8: Number of discarded samples for different values of σ^2_W, as a function of ξ_t. The true resampling factor is $\xi = \frac{3}{2}$. 500 Monte Carlo realizations were performed.

that when a resampling factor under test does not match the true one, then it can be discarded when an empty set is obtained for a non-visible sample or when a visible sample falls outside the obtained interval (cf. Steps 3 and 5 in Section 3.2.3.2).
This is illustrated at Fig. 3.8, where the number of samples required for discarding the candidate resampling factor is shown for different values of σ^2_W, when $\xi = \frac{3}{2}$. As it can be checked in that figure, whenever the $\xi_t = \xi$, the tested resampling factor will not be discarded, even when the full vector of observations is considered, as it should be expected. It is also important to point out that the larger the value of σ^2_W, i.e., the more accurate the fine-quantization assumption is, the smaller number of samples is required for discarding a wrong ξ_t.

3.2.4.2 Performance analysis with real audio signals

For the evaluation of the estimator in a real scenario, we consider the "Music Genres" audio database [48], composed of 1000 uncompressed audio files with 10 different music styles (for instance some

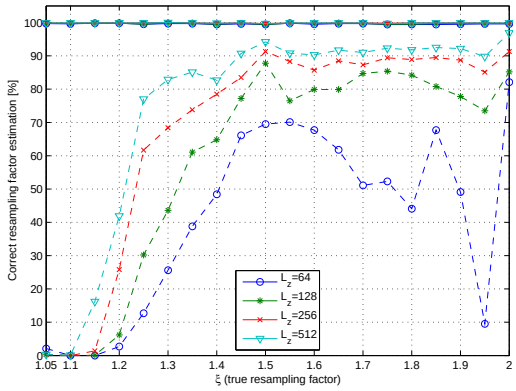


Figure 3.9: Comparison of the correct estimation percentage of the proposed MLE versus the method proposed in [41]. Solid lines represent the obtained results with the MLE, while dashed lines are used for the method [41].

of them are blues, country, jazz, pop or rock). The performance of the proposed estimator will be checked by fixing the number of available samples, and looking for inconsistencies in the resampled signal with respect to the tested resampling factor. For comparison, the same tests will be performed with a state-of-the-art resampling detector, i.e., the one proposed by Popescu and Farid in [41].[1]
The set of resampling factors that we will consider in this case will be in the interval $(1, 2]$ (sampled with a step size of 0.05). Since we are interested in comparing the performance with different sizes for the vector of observations, we perform the experiments with the following set of values $L_z \in \{64, 128, 256, 512\}$.
The results obtained with both methods are shown in Fig. 3.9. As we can observe, the method proposed by Popescu and Farid is highly dependent on the number of available samples, whereas our proposed MLE is essentially independent of this parameter. In the same way, the performance achieved by their method is poor when the applied resampling factor is close to 1, which is neither an issue for our estimator. These two limitations of Popescu and Farid's method come from the frequency analysis performed (once the pmap has been computed) for the detection of the resampling factor, as we pointed out in the Introduction. From these results, it is clear that the MLE method becomes very useful for estimating the resampling factor when a small number of samples are available, thus leading to a very practical forensic tool.
Although the performance of the MLE is very good, if we consider a noisy vector of observations then the method of Popescu and Farid is expected to be more robust than the proposed MLE. The reason is that in their model for the EM algorithm, they assume Gaussian noise, and in our case, we are only assuming the presence of uniformly distributed noise, due to the quantization. We note, however, that it is possible to extend our model to the case of Gaussian noise. Such extension is left for future research.

[1] The neighborhood of the predictor is set to $N = 3$, yielding a window of length 7.

3.2.5 Conclusions

The problem of resampling factor estimation following the ML criterion has been investigated in this section for the 1-D case. The derived MLE from this analysis has been tested with audio signals showing very good performance. The most distinctive characteristic of the proposed approach is that only a few number of samples of the resampled signal is needed to correctly estimate the used resampling factor.
Since the scenario where the proposed resampling factor estimator can be employed is quite limited, future work will focus on improving this aspect. As a first step, the 2-D extension of the obtained method will be explicitly derived. Introduction of new parameters in the model such as general interpolation filters, noisy observations or resampling factors smaller than one will be studied. Possible links between this work and set membership algorithms will also be considered.

3.3 Transform coder identification based on quantization footprints and lattice theory

Transform coding is routinely used for lossy compression of discrete sources with memory. The input signal is divided into N-dimensional vectors, which are transformed by means of a linear mapping. Then, transform coefficients are quantized and entropy coded. In this section we consider the problem of identifying the transform matrix as well as the quantization step sizes. We study the challenging case in which the only available information is a set of P transform decoded vectors. We formulate the problem in terms of finding the lattice with the largest determinant that contains all observed vectors. We propose an algorithm that is able to find the optimal solution and we formally study its convergence properties. Our analysis shows that it is possible to identify successfully both the transform and the quantization step sizes when $P \geq N + \delta$ where δ is a small integer, and the probability of failure decreases exponentially to zero as $P - N$ increases.

3.3.1 Introduction

Due to its centrality to any type of multimedia data, transform coding theory is now extensively used in a new range of applications that rely on the possibility of reverse-engineering complex chains of operators starting from the available output signals. Indeed, the lifespan of a multimedia signal is virtually unbounded. This is due to the ability of creating copies and the availability of inexpensive storage options. However, signals seldom remain identical to their original version. As they pass through processing chains, some operators, including transform coding, are bound to leave subtle characteristic footprints on the signals, which can be identified in order to uncover their processing history. This insight might be extremely useful in a wide range of scenarios in the field of multimedia signal processing at large including, e.g.,: i) forensics, in order to address tasks such as source device identification [50] or tampering detection [51][35]; ii) quality assessment, to enable no-reference methods that rely solely on the received signals [52][53]; iii) digital restoration, which requires prior knowledge about the operations that affected a digital signal [54].
In this context, several works have exploited the footprints left by transform coding. In [55], a method was proposed to infer the implementation-dependent quantization matrix template used in a JPEG-compressed image. Double JPEG compression introduces characteristic peaks in the histogram of DCT coefficients, which can be detected and used, e.g, for tampering localization [56][35]. More recently, similar techniques were applied to video signals for the cases of MPEG-2 [57][58], MPEG-4 [59][60] and H.264/AVC [61].

All the aforementioned works require prior knowledge of the type of standard being considered. This implies that the specific transform in use is assumed to be known, whereas the quantization step sizes need to be estimated. In practice, it might be useful to be able to infer which transform was used in order to understand, for example, whether an image was compressed using the DCT-based JPEG or the wavelet-based JPEG 2000 and, in the latter case, which wavelet transform was used. Similarly, it would be good to be able to infer if a video sequence was compressed using MPEG-2, MPEG-4 or H.264/AVC. Some efforts in this direction can be found in [62].

Most of the above methods focus only on a specific type of multimedia signal (e.g., only images or only videos) and are to some extent heuristic. It is therefore natural to try and develop a universal theory of transform coder identification that is independent of the specific application at hand. To this end, here we consider a general model of transform coding that can be tailored to describe a large variety of practical implementations that are found in lossy coding systems, including those adopted in multimedia communication. Specifically, a 1-dimensional input signal is encoded by partitioning it into non-overlapping N-dimensional vectors, which are then transformed by means of a linear mapping. Then, transform coefficients are quantized and entropy coded. At the decoder, quantization symbols are entropy decoded and mapped to reconstruction levels. Then, the inverse transform is applied to obtain an approximation of the signal in its original domain.

Given the output produced by a specific transform coding chain, we investigate the problem of identifying its parameters. To this end, we assume both the size and the alignment of the transform to be known, as they can be estimated with methods available in the literature [58][55]. We propose an algorithm that receives as input a set of P transform decoded vectors embedded in a N-dimensional vector space and produces as output an estimation of the transform adopted, as well as the quantization step sizes, whenever these can be unambiguously determined. We leverage the intrinsic discrete nature of the problem, by observing the fact that these vectors are bound to belong to a N-dimensional lattice. Hence, the problem is formulated in terms of finding a lattice that contains all observed vectors. We propose an algorithm that is able to solve the problem and we formally study its convergence properties. Our analysis shows that it is possible to successfully identify both the transform and the quantization step sizes with high probability when $P > N$. In the experiments we found that an excess of approximately 6-7 observed vectors beyond the dimension N of the space is generally sufficient to ensure successful convergence. In addition, the complexity of the algorithm is shown to grow linearly with N.

It is important to mention that the method used to solve the problem addressed in this work is related to Euclid's algorithm, which is used to find the greatest common divisor (GCD) in a set of integers. Indeed, when $N = 1$ and $P = 2$, the proposed method coincides with Euclid's algorithm. However, in this case the problem reduces to estimating the quantization step size, as the transform is trivially defined. Note that, lattice theory has been widely used for source and channel coding (e.g., [63, 64, 65]). However, to the best of the authors' knowledge, this theory has not been employed to address the problem of identifying a linear mapping using the footprint left by quantization. Only [25] uses similar principles but their goal is to investigate the color compression history, i.e., the colorspace used in JPEG compression. Therefore, the solution proposed is tailored to work in a 3-dimensional vector space, thus avoiding the challenges that arise in higher dimensional spaces.

Also, it is important not to confuse the problem addressed here with the classical problem of lattice reduction [65]. In the latter case, given a basis for a lattice, one seeks an equivalent basis matrix with favorable properties. Usually, such a basis consists of vectors that are short and with improved orthogonality. There are several definitions of lattice reduction with corresponding reduction criteria, each meeting a different tradeoff between quality of the reduced basis and the computational effort required for finding it. The most popular one is the Lenstra-Lenstra-Lovasz (LLL) reduction [66],

which can be interpreted as an extension of the Gauss reduction to lattices of rank greater than 2. The rest of this section is organized as follows. Section 3.3.2 introduces the necessary notation and formulates the transform identification problem and Section 3.3.3 provides the background on lattice theory. The proposed method is described in Section 3.3.4. Then, a theoretical analysis of the convergence properties is presented in Section 3.3.5. The performance of the transform identification algorithm is evaluated empirically in Section 3.3.6. Finally, Section 3.3.7 indicates the open issues and stimulating further investigations.

3.3.2 Problem statement

The symbols x, $\mathbf{x}$ and $\mathbf{X}$ denote, respectively, a scalar, a column vector and a matrix. A $M \times N$ matrix $\mathbf{X}$ can be written either in terms of its columns or rows. Specifically,

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{array} \right] = \left[\begin{array}{c} \bar{\mathbf{x}}_1^T \\ \bar{\mathbf{x}}_2^T \\ \dots \\ \bar{\mathbf{x}}_M^T \end{array} \right]. \tag{3.12}$$

Let $\mathbf{x}$ denote a N-dimensional vector and $\mathbf{W}$ a transform matrix, whose rows represent the transform basis functions.
Transform coding is performed by applying scalar quantization to the transform coefficients $\mathbf{y} = \mathbf{W}\mathbf{x}$. Let $\mathcal{Q}_i(\cdot)$ denote the quantizer associated to the i-th transform coefficient. We assume that $\mathcal{Q}_i(\cdot)$ is a scalar uniform quantizer with step size Δ_i, $i = 1, \dots, N$. Therefore, the reconstructed quantized coefficients can be written as $\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N]^T$, with

$$\tilde{y}_i = \mathcal{Q}_i(y_i) = \Delta_i \cdot \text{round}\left[\frac{y_i}{\Delta_i}\right], \quad i = 1, \dots, N. \tag{3.13}$$

The reconstructed block in the original domain is given by $\tilde{\mathbf{x}} = \mathbf{W}^{-1}\tilde{\mathbf{y}}$.
Let $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$ denote a set of P observed N-dimensional vectors, which are the output of a transform coder. Due to quantization, the unobserved vectors representing quantized transform coefficients $\{\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_P\}$ are constrained to belong to a lattice $\mathcal{L}_y$ described by the following basis:

$$\mathbf{B}_y = \left[\begin{array}{cccc} \Delta_1 & 0 & \dots & 0 \\ 0 & \Delta_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Delta_N \end{array} \right] \tag{3.14}$$

Therefore, the observed vectors $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$ belong to a lattice $\mathcal{L}_x$ described by the basis:

$$\mathbf{B}_x = [\mathbf{b}_{x,1}, \dots, \mathbf{b}_{x,N}] = \mathbf{W}^{-1}\mathbf{B}_y, \tag{3.15}$$

with $\mathbf{b}_{x,i} = \Delta_i \hat{\mathbf{w}}_i$, $i = 1, \dots, N$, $\mathbf{W}^{-1} = [\hat{\mathbf{w}}_1, \dots, \hat{\mathbf{w}}_N]$.
In this work we study the problem of determining $\mathbf{B}_x$ from a finite set of $P \geq N$ distinct vectors $\{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_P\}$. That is, we seek to determine the parameters of a transform coder based on the footprints left on its output. We propose an algorithm to solve this problem and we study its convergence properties. In addition, we show that the probability of correctly determining $\mathbf{B}_x$ (or, equivalently, another basis for the lattice $\mathcal{L}_x$) is monotonically increasing in the number of observations P, and rapidly approaching one when $P > N$.

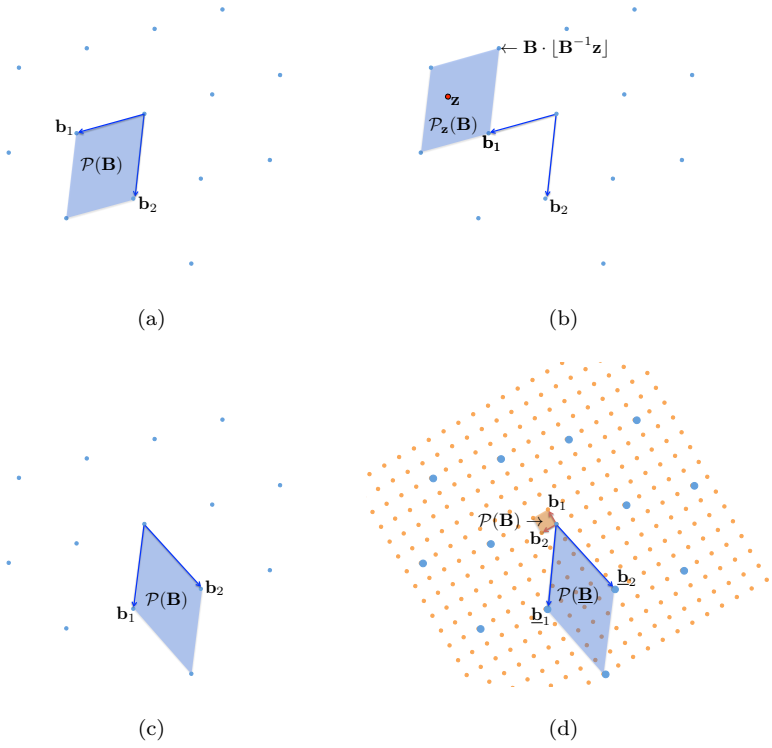


Figure 3.10: Examples of lattices. (a) The fundamental parallelotope of a lattice defined by a basis $\mathbf{B}$. (b) Parallelotope enclosing an arbitrary vector $\mathbf{z}$. (c) Another (equivalent) basis for the lattice in (a). (d) An example of a sub-lattice of the lattice $\mathcal{L}(\mathbf{B})$.

Note that when determining $\mathbf{B}_x$, the proposed method does not make any assumption on the structure of the transform matrix $\mathbf{W}$. In the general case, given $\mathbf{B}_x$, it is not possible to uniquely determine $\mathbf{W}$ and the quantization step sizes Δ_i, $i = 1, \ldots, N$. Indeed, the length of each basis vector $\mathbf{b}_{x,i}$ can be factored out as $\|\mathbf{b}_{x,i}\|_2 = \Delta_i \|\hat{\mathbf{w}}_i\|_2$. However, in the important case in which $\mathbf{W}$ represents an orthonormal transform, the quantization step sizes Δ_i, $i = 1, \ldots, N$, and the transform matrix $\mathbf{W}$ can be immediately obtained from $\mathbf{B}_x$. Indeed, $\mathbf{W}^{-1} = \mathbf{W}^T$, $\hat{\mathbf{w}}_i = \bar{\mathbf{w}}_i$, $i = 1, \ldots, N$, with $\|\bar{\mathbf{w}}_i\|_2 = 1$. Therefore:

$$\Delta_i = \|\mathbf{b}_{x,i}\|_2, \quad i = 1, \ldots, N, \tag{3.16}$$

$$\bar{\mathbf{w}}_i = \mathbf{b}_{x,i} / \|\mathbf{b}_{x,i}\|_2 \quad i = 1, \ldots, N. \tag{3.17}$$

3.3.3 Background on lattice theory

In this section we provide the necessary background on lattice theory. Further details can be found, e.g., in [67][68][65]. Let $\mathcal{L}$ denote a lattice of rank K embedded in $\mathbb{R}^N$. Let $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_K]$ denote a basis for the lattice $\mathcal{L}$. That is,

$$\mathcal{L} = \{\mathbf{x} | a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \ldots + a_K\mathbf{b}_K, a_i \in \mathbb{Z}\}. \tag{3.18}$$

In order to make the mapping between a basis and the corresponding lattice explicit, the latter can be expressed as $\mathcal{L}(\mathbf{B})$.
Any lattice basis also describes a fundamental parallelotope according to

$$\mathcal{P}(\mathbf{B}) = \left\{ \mathbf{x} | \mathbf{x} = \sum_{i=1}^{K} \theta_i \mathbf{b}_i, 0 \leq \theta_i < 1 \right\}. \tag{3.19}$$

When $K = 2, 3$, $\mathcal{P}(\mathbf{B})$ is, respectively, a parallelogram or a parallelepiped. As an example, Figure 3.10(a) shows the fundamental parallelotope corresponding to a lattice basis $\mathbf{B}$ when $K = 2$. Given a point $\mathbf{z} \in \mathbb{R}^K$, let $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ denote the parallelotope enclosing $\mathbf{z}$. $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ is obtained by translating $\mathcal{P}(\mathbf{B})$ so that its origin coincides with one of the lattice points. More specifically,

$$\mathcal{P}_z(\mathbf{B}) = \left\{ \mathbf{x} | \mathbf{x} = \mathbf{B} \cdot \left\lfloor \mathbf{B}^{-1}\mathbf{z} \right\rfloor + \sum_{i=1}^{K} \theta_i \mathbf{b}_i, 0 \leq \theta_i < 1 \right\}. \tag{3.20}$$

Figure 3.10(b) illustrates $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$ for an arbitrary vector $\mathbf{z}$.
Different bases for the same lattice lead to different fundamental parallelotopes. For example, Figure 3.10(a) and Figure 3.10(c) depict two different bases for the same lattice, together with the corresponding fundamental parallelotopes. However, the volume of $\mathcal{P}(\mathbf{B})$ is the same for all bases of a given lattice. This volume equals the so-called *lattice determinant*, which is a lattice invariant defined as

$$|\mathcal{L}| = \sqrt{\det(\mathbf{B}^T\mathbf{B})}. \tag{3.21}$$

If the lattice is full rank, i.e., $K = N$, the lattice determinant equals the determinant of the matrix $\mathbf{B}$, $|\mathcal{L}| = |\det(\mathbf{B})|$.
Let $\underline{\mathcal{L}}$ denote a sub-lattice of $\mathcal{L}$. That is, for any vector $\mathbf{x} \in \underline{\mathcal{L}}$, then $\mathbf{x} \in \mathcal{L}$. A basis $\underline{\mathbf{B}}$ for $\underline{\mathcal{L}}$ can be expressed in terms of $\mathbf{B}$ as

$$\underline{\mathbf{B}} = \mathbf{B}\mathbf{U}, \tag{3.22}$$

where $\mathbf{U}$ is such that $u_{ij} \in \mathbb{Z}$. Moreover, let $\det(\mathbf{U}) = \pm m$, then

$$\frac{|\underline{\mathcal{L}}|}{|\mathcal{L}|} = |\det(\mathbf{U})| = m \tag{3.23}$$

and we say that $\underline{\mathcal{L}}$ is a sub-lattice of $\mathcal{L}$ of index m. For example, Figure 3.10(d) shows two lattices $\underline{\mathcal{L}}$ and $\mathcal{L}$, such that $\underline{\mathcal{L}} \subset \mathcal{L}$. In this case, the matrix $\mathbf{U}$ is equal to

$$\mathbf{U} = \begin{bmatrix} -4 & -5 \\ 3 & -1 \end{bmatrix}, \tag{3.24}$$

and $\underline{\mathcal{L}}$ is a sub-lattice of index $m = 19$.

3.3.4 An algorithm for transform identification

In this section we propose an algorithm that is able to determine the parameters of a transform coder from its output, i.e., a set of observed vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$. This is accomplished by finding a suitable lattice $\mathcal{L}^*$ such that $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\} \subset \mathcal{L}^*$. In Section 3.3.5.3 we will show that, with probability approaching one, $\mathcal{L}^* \equiv \mathcal{L}_x$, provided that $P - N > 0$.
The problem of determining a basis for the lattice $\mathcal{L}_x$ is complicated by the fact that we typically observe a finite (and possibly small) number of vectors P embedded in a possibly large dimensional

space. More precisely, $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$ belong to a bounded lattice, in virtue of the fact that each transform coefficient y_i is quantized with a finite number of bits R_i, to one of 2^{R_i} reconstruction levels. Let $\bar{R}$ denote the average number of bits allocated to transform coefficients. The number of potential lattice points is equal to

$$\prod_{i=1}^{N} 2^{R_i} = 2^{\sum_{i=1}^{N} R_i} = 2^{N\bar{R}}, \tag{3.25}$$

and only P of them are covered by observed vectors. Thus, we note that, given $\bar{R}$, the number of lattice points increases exponentially with the dimension N and that in most cases of practical relevance $P \ll 2^{N\bar{R}}$.

Another issue arises from the fact that, for a set of vectors $\{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$, there are infinitely many lattices that include all of them. Indeed, any lattice $\bar{\mathcal{L}}$ such that $\mathcal{L}_x \subset \bar{\mathcal{L}}$ is compatible with the observed set of vectors. Note that any basis of the form $\mathbf{B} = \mathbf{B}_x\mathbf{U}^{-1}$, with $\det(\mathbf{U}) = \pm m$, with m an integer greater than one defines a compatible lattice $\bar{\mathcal{L}}$. A simple example is obtained setting $\mathbf{U} = a\mathbf{I}$, $a \in \mathbb{N}, a > 1$.

In order to resolve this ambiguity, we seek the lattice $\mathcal{L}^*$ that maximizes the lattice determinant $|\mathcal{L}|$, within this infinite set of compatible lattices. That is,

$$\begin{array}{ll} \underset{\mathcal{L}(\mathbf{B})}{\text{maximize}} & |\mathcal{L}(\mathbf{B})| \\ \text{subject to} & \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\} \subset \mathcal{L}(\mathbf{B}). \end{array} \tag{3.26}$$

For example, for the set of observed points $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3\}$ depicted in Figure 3.11(a), Figure 3.11(g) illustrates a basis for the lattice that is the optimal solution of (3.26). In contrast, the lattice in Figure 3.11(h) is a feasible solution of (3.26), but it is not optimal, since it is characterized by a lower value of the lattice determinant.

The proposed method used to solve the problem above is detailed in Algorithm 1. The method constructs an initial basis for an N-dimensional lattice (line 1). This is accomplished by considering the vectors in $\mathcal{O}$ until N linearly independent vectors are found. These vectors are used as columns of the starting estimate $\mathbf{B}^{(0)}$ and to populate the initial set of visited vectors $\mathcal{S}$. We denote with $\mathcal{U}$ the set of vectors in $\mathcal{O}$ that have not been visited yet. Then, the solution of (3.26) is constructed iteratively, by considering the remaining vectors in $\mathcal{U}$ one by one. At each iteration, the function `recurseTI` returns a basis for a lattice that solves (3.26), in which the constraint is imposed only on the subset of visited vectors $\mathcal{S}$, that is, $\mathcal{S} \subset \mathcal{L}(\mathbf{B})$. As such, the algorithm starts finding the solution of an under-constrained problem and additional constraints are added as more vectors are visited.

Figure 3.11 shows an illustrative example when $N = 2$ and three vectors $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3\}$ are observed (Figure 3.11(b)). The initial basis (line 1) is constructed using $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{x}}_2$, since they are linearly independent (Figure 3.11(b)). Then, the point $\tilde{\mathbf{x}}_3$ is selected (line 6 and Figure 3.11(c)) and the function `recurseTI` (line 9) returns a basis that solves (3.26), i.e., a basis with the largest lattice determinant that includes all observed vectors. Figure 3.11(f) illustrates such a basis, and Figure 3.11(g) shows an equivalent basis obtained after lattice reduction.

The core of the method is the recursive function `recurseTI`. When describing this function, we keep a clear distinction between algorithm template and algorithm instance, as it is customary in computer science. We start describing the template in Algorithm 2, which does not specify the function entirely. Then, a concrete instance of the template is detailed in Algorithm 3. The rationale of maintaining this distinction is motivated by the fact that the correctness of the method is a property that descends from the template alone, as further discussed in Section 3.3.5.1. Conversely, the rate of convergence depends on the specific algorithm instance, as explained in Section 3.3.5.2.

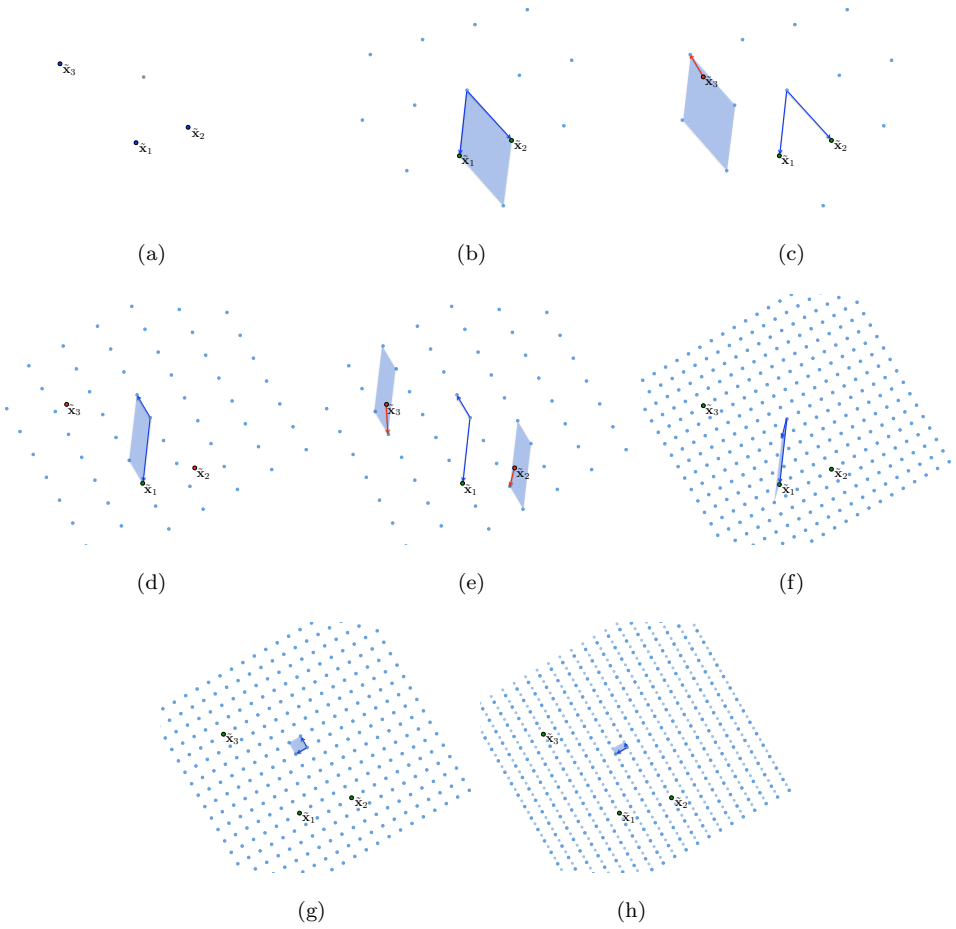


Figure 3.11: An example of transform identification. A set of three observed vectors is given in (a). Then, (b)-(h) show, step-by-step, how the solution to problem (3.26) is sought by Algorithm 1.

3.3.4.1 An algorithm template for `recurseTI`

The function `recurseTI` receives as input a set of visited vectors $\mathcal{S}$ and the current estimate of a basis $\mathbf{B}$ for the lattice $\mathcal{L}(\mathbf{B})$. If $\mathcal{S} \subset \mathcal{L}$, i.e., all the vectors in $\mathcal{S}$ belong to the lattice defined by $\mathbf{B}$, the recursion is terminated (line 1 in Algorithm 2). Otherwise, one of the vectors $\tilde{\mathbf{x}}$ that does not belong to $\mathcal{L}$ is selected (line 4) and the parallelotope which encloses it is identified (line 5). Then, a vector $\mathbf{d}$ is computed as the difference between $\tilde{\mathbf{x}}$ and one of the vertices of the parallelotope (line 6). The intuition here is to capture a short vector that cannot be represented by the current lattice, and to modify the current basis in such a way that (upon convergence) it can be represented. Hence, the updated basis is constructed by replacing one of the columns of $\mathbf{B}$ with $\mathbf{d}$ (line 8). Among the N possible cases, any choice such that $\mathbf{B}_i$ is non-singular represents a valid selection (line 10).
In the example in Figure 3.11, two recursive steps are performed before terminating `recurseTI`. In the first call, it is verified that $\tilde{\mathbf{x}}_3$ does not belong to the lattice defined by the current basis

ALGORITHM 1: `TI` algorithm

Input: *Set of observed vectors* $\mathcal{O} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_P\}$
Output: *A basis* $\mathbf{B}$ *of the lattice solution of* (3.26)

1. $\mathbf{B}^{(0)} = \texttt{initBasis}(\mathcal{O})$;
2. $\mathcal{S} = \{\mathbf{b}_1, \ldots, \mathbf{b}_N\}$;
3. $\mathcal{U} = \mathcal{O} \setminus \mathcal{S}$;
4. $r = 0$
5. **while** card$\{\mathcal{U}\} > 0$;
6. Pick $\tilde{\mathbf{x}}$ in $\mathcal{U}$;
7. $\mathcal{U} = \mathcal{U} \setminus \{\tilde{\mathbf{x}}\}$;
8. $\mathcal{S} = \mathcal{S} \cup \tilde{\mathbf{x}}$;
9. $\mathbf{B}^{(r+1)} = \texttt{recurseTI}(\mathbf{B}^{(r)}, \mathcal{S})$;
10. $r = r + 1$
11. **end**

ALGORITHM 2: `recurseTI`$(\mathbf{B}, \mathcal{S})$

Input: *Set of vectors* $\mathcal{S} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_S\}$*, a basis* $\mathbf{B}$ *of a lattice.*
Output: *A basis of a lattice* $\mathcal{L}$ *with maximum determinant* $|\mathcal{L}|$*, such that* $\mathcal{S} \subset \mathcal{L}$

1. **if** $\mathcal{S} \subset \mathcal{L}(\mathbf{B})$
2. **return B**
3. **else**
4. Pick $\mathbf{z} \in \mathcal{S} \setminus \mathcal{L}(\mathbf{B})$.
5. Determine $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$.
6. Pick a vertex $\mathbf{v}$ of $\mathcal{P}_{\mathbf{z}}(\mathbf{B})$.
7. Compute $\mathbf{d} = \mathbf{z} - \mathbf{v}$.
8. Compute $\mathbf{B}_i$, replacing the i-th column of $\mathbf{B}$ with $\mathbf{d}$.
9. Pick an index l, such that $\det(\mathbf{B}_l) \neq 0$.
10. `recurseTI`$(\mathbf{B}_l, \mathcal{S})$;
11. **end**

(Figure 3.11(c)), and the updated basis is constructed (Figure 3.11(d)) by replacing one of the two basis vectors with the difference vector between $\tilde{\mathbf{x}}_3$ and one of the vertices of $\mathcal{P}_{\tilde{\mathbf{x}}_3}(\mathbf{B})$. In the second call it is verified that neither $\tilde{\mathbf{x}}_3$ nor $\tilde{\mathbf{x}}_2$ belong to the updated lattice. Therefore, one of the two difference vectors (e.g., the one representing the difference between $\tilde{\mathbf{x}}_2$ and one of the vertices of $\mathcal{P}_{\tilde{\mathbf{x}}_2}(\mathbf{B})$) is used to replace one of the two basis vectors. In the third call the recursion is terminated, because all points in $\mathcal{S}$ belong to the lattice.
In Section 3.3.5.1, it is shown that the recursion always terminates in a finite number of steps and leads to the optimal solution of (3.26). The solution the algorithm converges to, though, might be a sub-lattice of the underlying lattice $\mathcal{L}_x$, i.e., $\mathcal{L}^* \subset \mathcal{L}_x$. Fortunately, this is a very unlikely event, even when the number of observed points P is only slightly larger than N, as further discussed in Section 3.3.5.3.

3.3.4.2 An algorithm instance for `recurseTI`

A practical instantiation of the template presented in Algorithm 2 requires to specify how to perform the choices at line 4, 6 and 9, which were left undefined. Note that these choices are arbitrary and have no effect on the correctness of the method, although they might affect the number of recursive steps needed to achieve convergence.

ALGORITHM 3: `recurseTI`$(\mathbf{B}, \mathcal{S})$

Input: *Set of vectors* $\mathcal{S} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_S\}$*, a basis* $\mathbf{B}$ *of a lattice.*
Output: *A basis of a lattice* $\mathcal{L}$ *with maximum determinant* $|\mathcal{L}|$*, such that* $\mathcal{S} \subset \mathcal{L}$

1. **if** condnum$(\mathbf{B}) > T$
2. $\quad \mathbf{B} = \texttt{LLL}(\mathbf{B})$
3. **end**
4. $\hat{\mathbf{x}}_i = \mathbf{B} \cdot \text{round}(\mathbf{B}^{-1}\tilde{\mathbf{x}}_i)$, $i = 1, \ldots, S$;
5. **if** $(\max_{j=1,\ldots,S} \|\tilde{\mathbf{x}}_j - \hat{\mathbf{x}}_j\|_2) = 0$
6. $\quad$ **return B**
7. **else**
8. $\quad f = \arg\min_{j \in \{l | \|\tilde{\mathbf{x}}_l - \hat{\mathbf{x}}_l\|_2 > 0\}} \|\tilde{\mathbf{x}}_j - \hat{\mathbf{x}}_j\|_2$;
9. $\quad \mathbf{d} = \tilde{\mathbf{x}}_f - \hat{\mathbf{x}}_f$;
10. $\quad \boldsymbol{\theta} = \mathbf{B}^{-1}\mathbf{d}$;
11. $\quad l = \arg\min_{j \in \{p | \theta_p \neq 0\}} |\theta_j|$;
12. $\quad \texttt{recurseTI}(\mathbf{B}_l, \mathcal{S})$;
13. **end**

In our specific implementation, the selection of the vector $\tilde{\mathbf{x}} \in \mathcal{S} \setminus \mathcal{L}(\mathbf{B})$ (line 4 in Algorithm 2), the vertex of the parallelotope (line 6) and the column to be replaced (line 9) are carried out as detailed in Algorithm 3. The rationale is to construct a new basis related to a lattice with the smallest lattice determinant $|\mathcal{L}(\mathbf{B})|$, so as to tighten the upper bound on the value of the optimal solution, i.e., $|\mathcal{L}^*| \leq |\mathcal{L}(\mathbf{B})|$.
Specifically, given a basis $\mathbf{B}$ as input, we compute the vector $\hat{\mathbf{x}} = \mathbf{B} \cdot \text{round}(\mathbf{B}^{-1}\tilde{\mathbf{x}})$, which represents one of the vertices of the parallelotope enclosing $\tilde{\mathbf{x}}$ (line 4 in Algorithm 3). In order to prevent numerical instability induced by the inversion of the matrix $\mathbf{B}$, we perform basis reduction according to the LLL algorithm (line 2) and we find a nearly orthogonal basis which is equivalent to $\mathbf{B}$, but has a smaller orthogonality defect. In our implementation, we perform basis reduction only when the condition number is greater than a threshold T, which was set equal to 10^4 (line 1).
Then, the selected point $\mathbf{z} = \tilde{\mathbf{x}}_f$ is the one that minimizes the distance from the corresponding vertex (line 8). That is,

$$f = \arg \min_{j \in \{l | \|\tilde{\mathbf{x}}_l - \hat{\mathbf{x}}_l\|_2 > 0\}} \|\tilde{\mathbf{x}}_j - \hat{\mathbf{x}}_j\|_2, \tag{3.27}$$

so as to minimize the length of the new basis vector $\mathbf{d}$. Similarly, the choice of the new basis among the set of (up to) N candidate bases $\mathbf{B}_i$ (line 11) is to select the one that leads to the smallest lattice determinant, after excluding those that do not have rank N. From Cramer's rule, it follows that $\det(\mathbf{B}_i) = \theta_i \det(\mathbf{B})$, where $\boldsymbol{\theta} = \mathbf{B}^{-1}\mathbf{d}$ is the expansion of $\mathbf{d}$ in the basis $\mathbf{B}$. Hence, we replace the l-th column of $\mathbf{B}$, which is the one corresponding to the entry of $\boldsymbol{\theta}$ with the least strictly positive absolute value. That is,

$$l = \arg \min_{j \in \{p | \theta_p \neq 0\}} |\theta_j|. \tag{3.28}$$

3.3.5 Analysis

3.3.5.1 Convergence

In this section, we prove that the proposed algorithm converges in a finite number of recursive steps to the solution $\mathcal{L}^*$ of (3.26). To this end, we rely on the specifications of the algorithm template in Algorithm 2.

Let $\mathbf{B}^{(0)}$ denote the initial estimate of a basis of the lattice, which is constructed, for example, by selecting as its columns a subset of N linearly independent vectors in $\mathcal{O}$ (Algorithm 1, line 1). Hence, each vector of the initial basis $\mathbf{B}^{(0)}$ can be expressed as a linear combination with integer coefficients of the columns of $\mathbf{B}_x$. Thus, we can write $\mathbf{B}^{(0)} = \mathbf{B}_x\mathbf{A}$, with $\det(\mathbf{A}) = m$ and $m \in \mathbb{Z} \setminus \{0\}$. From this, it follows that $|\mathcal{L}(\mathbf{B}^{(0)})| = m \cdot |\mathcal{L}_x|$ and $|\mathcal{L}_x| \leq |\mathcal{L}(\mathbf{B}^{(0)})|$
Let $\mathbf{B}^{(r)}$ denote the estimate obtained after the r-th call of the recursive function `recurseTI`. It is possible to prove the following lemma:

Lemma 3. *$|\mathcal{L}(\mathbf{B}^{(r+1)})| \leq |\mathcal{L}(\mathbf{B}^{(r)})|$, with equality if and only if $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(r)}) = \mathcal{L}(\mathbf{B}^{(r+1)})$*

Proof. If $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(r)})$, then $\mathbf{B}^{(r+1)} = \mathbf{B}^{(r)}$ and the recursion terminates. Otherwise, let $\mathbf{z} \in \mathcal{S} \setminus \mathcal{L}(\mathbf{B}^{(r)})$ be any of the points which does not belong to the lattice defined by $\mathbf{B}^{(r)}$, $\mathbf{v}$ any of the vertices of $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$ and $\mathbf{d} = \mathbf{z} - \mathbf{v}$. The vector $\mathbf{d}$ can be expressed in terms of the basis $\mathbf{B}^{(r)}$ as

$$\mathbf{d} = \mathbf{B}^{(r)}\boldsymbol{\theta}. \tag{3.29}$$

By definition, the vector $\mathbf{z}$ belongs to $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$, hence $-1 \leq \theta_i \leq 1$. Since $\mathbf{z} \notin \mathcal{L}(\mathbf{B}^{(r)})$, $\mathbf{z}$ does not belong to the vertices of $\mathcal{P}_{\mathbf{z}}(\mathbf{B}^{(r)})$. It follows that there is at least one coefficient θ_l in the basis expansion of $\mathbf{d}$, such that $0 < |\theta_l| < 1$.
The vector $\mathbf{d}$ replaces the i-th column of $\mathbf{B}^{(r)}$ to obtain $\mathbf{B}_i^{(r)}$. From Cramer's rule,

$$\det(\mathbf{B}_i^{(r)}) = \theta_i \det(\mathbf{B}^{(r)}) \tag{3.30}$$

Therefore, if we select l, such that $0 < |\theta_l| < 1$,

$$|\mathcal{L}(\mathbf{B}^{(r+1)})| = |\det(\mathbf{B}^{(r+1)})| = |\det(\mathbf{B}_l^{(r)})| = |\theta_l||\det(\mathbf{B}^{(r)})| < |\det(\mathbf{B}^{(r)})| = |\mathcal{L}(\mathbf{B}^{(r)})| \tag{3.31}$$

Note that there must be at least one such an index l, as indicated above.

□

We construct the sequence of integer numbers

$$s_r = |\mathcal{L}(\mathbf{B}^{(r)})|, \quad r = 0, 1, \ldots, R. \tag{3.32}$$

Let R denote the smallest integer such that $|\mathcal{L}(\mathbf{B}^{(R)})| = |\mathcal{L}(\mathbf{B}^{(R+1)})|$. That is, R is the number of steps needed to achieve convergence. It is possible to prove the following theorem:

Theorem 2. *Algorithm 1 converges to the solution of* (3.26).

Proof. Let $\mathcal{L}^*$ denote the solution of (3.26), i.e., the lattice with maximum volume that includes all observed vectors $\mathcal{S}$. We need to prove that $\mathcal{L}(\mathbf{B}^{(R)}) = \mathcal{L}^*$.
First, we prove that $|\mathcal{L}(\mathbf{B}^{(R)})|$ cannot decrease beyond $|\mathcal{L}^*|$, i.e., $|\mathcal{L}^*| \leq |\mathcal{L}(\mathbf{B}^{(R)})|$. To this end, let $\mathcal{L}(\mathbf{B}^{(R-1)})$ denote the lattice obtained at the iteration just before convergence. Hence, there is at least one observed vector $\tilde{\mathbf{x}} \in \mathcal{L}^*$ such that $\tilde{\mathbf{x}} \notin \mathcal{L}(\mathbf{B}^{(R-1)})$. Lemma 3 establishes that $|\mathcal{L}(\mathbf{B}^{(R)})| < |\mathcal{L}(\mathbf{B}^{(R-1)})|$.
Let $\mathbf{d}$ denote the difference vector as in line 7 of Algorithm 2. By construction, $\mathbf{d} \in \mathcal{L}^*$. Let $\mathbf{B}^*$ denote a basis for $\mathcal{L}^*$. Then, it is possible to write $\mathbf{d} = \mathbf{B}^*\boldsymbol{\theta}^*$, $\theta_i^* \in \mathbb{Z}$. $\mathcal{L}(\mathbf{B}^{(R-1)})$ is a sublattice of $\mathcal{L}^*$. Hence, $\mathbf{B}^{(R-1)} = \mathbf{B}^*\mathbf{A}$, where $\mathbf{A}$ is a matrix of integer elements such that $\det(\mathbf{A}) = m$, with $m \in \mathbb{Z} \setminus \{0\}$, and $|\mathcal{L}(\mathbf{B}^{(R-1)})|/|\mathcal{L}^*| = m$.
It is possible to express $\mathbf{d}$ in the basis expansion of $\mathbf{B}^{(R-1)}$. That is,

$$\boldsymbol{\theta} = (\mathbf{B}^{(R-1)})^{-1}\mathbf{d} = (\mathbf{B}^*\mathbf{A})^{-1}\mathbf{B}^*\boldsymbol{\theta}^* = \mathbf{A}^{-1}\boldsymbol{\theta}^* = \frac{1}{\det(\mathbf{A})}\text{cofactor}(\mathbf{A})\boldsymbol{\theta}^*. \tag{3.33}$$

Note that both the cofactor matrix cofactor($\mathbf{A}$) and $\boldsymbol{\theta}^*$ have integer elements. Hence, the vector cofactor($\mathbf{A}$)$\boldsymbol{\theta}^*$ has integer elements. Any nonzero element of $\boldsymbol{\theta}$ is an integer multiple of $1/\det(\mathbf{A}) = 1/m$. Therefore, if $\theta_i \neq 0$, $|\theta_i| \geq 1/m$.
From the proof of Lemma 3, we know that

$$|\mathcal{L}(\mathbf{B}^{(R)})| = |\theta_l||\mathcal{L}(\mathbf{B}^{(R-1)})| \geq \frac{1}{m}|\mathcal{L}(\mathbf{B}^{(R-1)})| = |\mathcal{L}^*|, \tag{3.34}$$

where θ_l is one of the nonzero elements of $\boldsymbol{\theta}$.
To prove that $|\mathcal{L}(\mathbf{B}^{(R)})| = |\mathcal{L}^*|$, it remains to be shown that cannot be $|\mathcal{L}(\mathbf{B}^{(R)})| > |\mathcal{L}^*|$. Indeed, if this were the case, $\mathcal{L}(\mathbf{B}^{(R)})$ would be the optimal solution of (3.26), since it includes all observed points $\mathcal{S}$ and has volume larger than $|\mathcal{L}^*|$. □

Note that $R < \infty$, i.e., convergence is achieved in a finite number of steps. Indeed, $\{s_r\}$ is a sequence of integer values. The sequence is monotonically decreasing due to Lemma 3, until convergence is achieved and $\mathcal{S} \subset \mathcal{L}(\mathbf{B}^{(R)})$. In addition, it is bounded from below by $|\mathcal{L}_x|$. Therefore, convergence is achieved in up to $|\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x|$ number of steps. In the following section we show that with a specific instantiation of Algorithm 2 given in Algorithm 3 it is possible to ensure a significantly faster convergence rate.

3.3.5.2 Rate of convergence

It is possible to prove that the proposed method implemented according to the instance presented in Algorithm 3 converges in a number of steps that is upper bounded by $\lceil \log_2(|\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x|) \rceil$. To show this, it suffices to demonstrate that the value of the lattice determinant is (at least) halved between two consecutive calls of `recurseTI`, as stated by the following theorem.

Theorem 3. *If* $\mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)})$*, then* $\frac{|\mathcal{L}(\mathbf{B}^{(r+1)})|}{|\mathcal{L}(\mathbf{B}^{(r)})|} \leq \frac{1}{2}$

Proof. Since $\mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)})$, then $\max_{j=1,\dots,S} \|\tilde{\mathbf{x}}_j - \hat{\mathbf{x}}_j\|_2 > 0$, and the recursion is not terminated. Consider the vector $\mathbf{d} = \tilde{\mathbf{x}}_f - \hat{\mathbf{x}}_f$, which can be expressed in the basis $\mathbf{B}^{(r)}$ as $\mathbf{d} = \mathbf{B}^{(r)}\boldsymbol{\theta}$. Dropping the superscript $^{(r)}$, it is possible to write

$$\begin{aligned}
\boldsymbol{\theta} &= \mathbf{B}^{-1}\mathbf{d} = \mathbf{B}^{-1}(\tilde{\mathbf{x}}_f - \hat{\mathbf{x}}_f) && (3.35)\\
&= \mathbf{B}^{-1}\tilde{\mathbf{x}}_f - \mathbf{B}^{-1}(\mathbf{B} \cdot \text{round}(\mathbf{B}^{-1}\hat{\mathbf{x}}_f)) && (3.36)\\
&= \mathbf{B}^{-1}\tilde{\mathbf{x}}_f - \text{round}(\mathbf{B}^{-1}\hat{\mathbf{x}}_f) = \mathbf{a} - \text{round}(\mathbf{a}), && (3.37)
\end{aligned}$$

where we set $\mathbf{a} = \mathbf{B}^{-1}\hat{\mathbf{x}}_f$. Due to the properties of rounding, $-1/2 \leq \theta_i < 1/2$. Thus, replacing any of the columns of $\mathbf{B}^{(r)}$ such that $\theta_l \neq 0$, we obtain, using Cramer's rule,

$$\frac{|\mathcal{L}(\mathbf{B}^{(r+1)})|}{|\mathcal{L}(\mathbf{B}^{(r)})|} = |\theta_l| < \frac{1}{2} \tag{3.38}$$

□

Based on Theorem 3,

$$|\mathcal{L}(\mathbf{B}^{(r)})| \leq \left(\frac{1}{2}\right)^r |\mathcal{L}(\mathbf{B}^{(0)})|, \quad \forall r > 0, \mathcal{S} \not\subset \mathcal{L}(\mathbf{B}^{(r)}) \tag{3.39}$$

Hence, convergence is achieved in up to

$$\left\lceil \log_2 \frac{|\mathcal{L}(\mathbf{B}^{(0)})|}{|\mathcal{L}_x|} \right\rceil \tag{3.40}$$

number of steps.
Note that this upper bound on the convergence rate is guaranteed solely on the basis of the way the vertex of the parallelotope is selected, whereas it does not depend neither on which point is selected, nor on which column is replaced. However, the heuristics applied in Algorithm 3 are based on the rationale of reducing the ratio $\frac{|\mathcal{L}(\mathbf{B}^{(r+1)})|}{|\mathcal{L}(\mathbf{B}^{(r)})|}$ as much as possible.

3.3.5.3 Probability of success

In Section 3.3.5.1, we showed that the proposed method converges to the optimal solution $\mathcal{L}^*$ of (3.26). In this section, we show that it converges to the correct (and unique) lattice $\mathcal{L}_x$ (i.e., $\mathcal{L}^* \equiv \mathcal{L}_x$) with high probability, provided that the number of observed vectors P is greater than N. Given a lattice $\mathcal{L}_x$ of rank N embedded in $\mathbb{R}^N$, there is more than one sub-lattice $\underline{\mathcal{L}}$ of $\mathcal{L}$ of index m. It can be shown that the number of sub-lattices is equal to [69]

$$f_N(m) = \prod_{i=1}^{q} \prod_{j=1}^{N-1} \frac{p_i^{t_i+j} - 1}{p_i^j - 1} = \prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j-1} - 1}{p_i^j - 1}, \tag{3.41}$$

where $m = p_1^{t_1} \cdots p_q^{t_q}$ is the prime factorization of m. That is, $p_1, \ldots, p_q$ are the prime factors of m, and t_s is the multiplicity of the factor p_s.
For example, when $N = 2$ and $m = 2$, $f_2(2) = 3$. Given the basis $\mathbf{B} = \mathbf{I}$, the corresponding sub-lattices of $\mathcal{L}(\mathbf{B})$ are generated by, e.g, the following bases

$$\mathbf{B}_1 = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B}_3 = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix}. \tag{3.42}$$

In order to determine analytically a lower bound on the probability of converging to the correct solution, we need to prove the following lemma, which provides bounds on the number of sub-lattices.

Lemma 4. *Given a lattice $\mathcal{L}_x$ of rank N embedded in $\mathbb{R}^N$, the number $f_N(m)$ of sub-lattices of index m is bounded by*

$$m^{N-1} < f_N(m) < m^N. \tag{3.43}$$

Proof. It is possible to derive both an upper and a lower bound on the number of sub-lattices that are independent from the prime factorisation of m starting from (3.41). Since for all cases of interest $N > 1$, we have:

$$\frac{p_i^{N+j-1} - 1}{p_i^j - 1} > \frac{p_i^{N+j-1}}{p_i^j}. \tag{3.44}$$

Substituting in (3.41), we have a function $\underline{f_N}(m)$ that is guaranteed to yield values below $f_N(m)$:

$$\underline{f_N}(m) = \prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j-1}}{p_i^j}. \tag{3.45}$$

This can be simplified to:

$$\underline{f_N}(m) = \prod_{i=1}^{q} p_i^{t_i(N-1)}. \tag{3.46}$$

This is equivalent to the $(N-1)^{\text{th}}$ power of the product of the prime factors of m. That is, the lower bound of $f_N(m)$ can be expressed as:

$$\underline{f_N}(m) = m^{N-1}. \tag{3.47}$$

In terms of the upper bound of $f_N(m)$, we proceed similarly by starting with the observation that:

$$\frac{p_i^{N+j-1} - 1}{p_i^j - 1} < \frac{p_i^{N+j}}{p_i^j}. \tag{3.48}$$

By substituting back into (3.41), we can observe that:

$$\prod_{i=1}^{q} \prod_{j=1}^{t_i} \frac{p_i^{N+j}}{p_i^j} = m\underline{f_N}(m). \tag{3.49}$$

Hence, it is easy to see that the upper bound on $f_N(m)$ can be expressed as:

$$\overline{f_N}(m) = m^N. \tag{3.50}$$

Therefore, since $\underline{f_N}(m) < f_N(m) < \overline{f_N}(m)$, we have:

$$m^{N-1} < f_N(m) < m^N. \tag{3.51}$$

□

Now, consider a specific sub-lattice $\underline{\mathcal{L}} \subset \mathcal{L}_x$ of index m and a set of P vectors from the original lattice $\mathcal{L}_x$. In the case of uniformly distributed vectors, the probability that one vector belong to the sub-lattice $\underline{\mathcal{L}}$ is equal to $(1/m)$. Thus, the probability that all P vectors belong to the same sub-lattice $\underline{\mathcal{L}}$ is equal to $(1/m)^P$, assuming statistical independence among the set of vectors.
Let $p_{\text{fail}}(N, P)$ denote the probability of failing to detect the underlying lattice $\mathcal{L}_x$ of rank N, when P points are observed. Then, $p_{\text{succ}}(N, P) = 1 - p_{\text{fail}}(N, P)$. A failure occurs whenever all P vectors fall in any of the sub-lattices of index m. Hence, we can write

$$p_{\text{fail}}(N, P) < \sum_{m=2}^{\infty} f_N(m) \left(\frac{1}{m}\right)^P < \sum_{m=2}^{\infty} m^N \left(\frac{1}{m}\right)^P = \sum_{m=2}^{\infty} \frac{1}{m^{P-N}} = \zeta(P-N) - 1 \tag{3.52}$$

The first inequality is a union bound, i.e., the probability of failure is upper bounded by the sum of the probabilities of observing all P vectors in a given sub-lattice. The second inequality follows from the upper bound given by Lemma 4. The last expression contains $\zeta(\cdot)$, which is the Riemann's zeta function. That is,

$$\zeta(s) = \sum_{m=1}^{\infty} \frac{1}{m^s}. \tag{3.53}$$

Note that the infinite series converges when the real part of the argument s is greater than 1. In our case, this requires $P - N > 1$ or $P > N + 1$. Then, the probability of success is lower bounded by

$$p_{\text{succ}}(N, P) > 2 - \zeta(P - N). \tag{3.54}$$

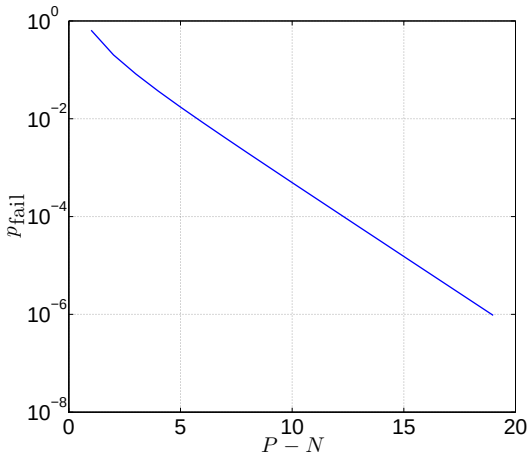
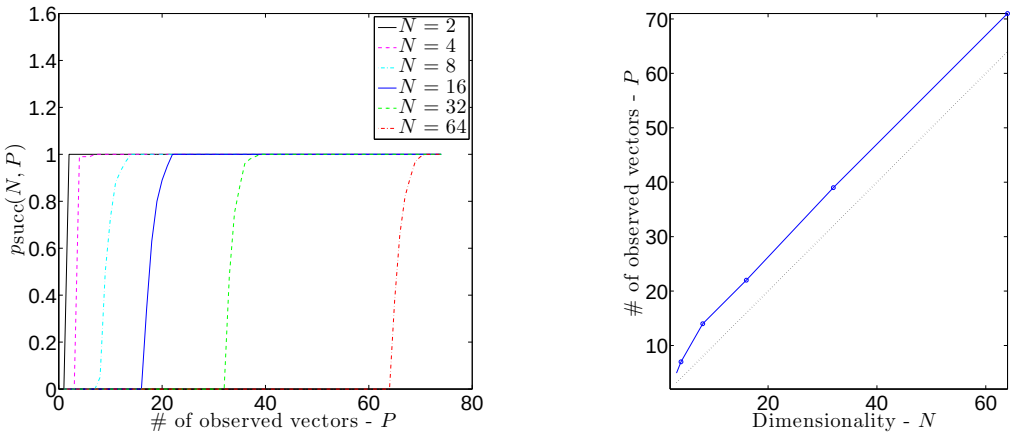


Figure 3.12: Upper bound on the probability of failure $p_{\text{fail}}(P, N)$.

It is interesting to observe that the probability of failure/success depend solely on the difference $P - N$. Hence, the number P of observed vectors needed to correctly identify the underlying lattice grows linearly with the dimensionality N of the embedding vector space, despite the number of potential lattice points grows exponentially with N, as indicated in Section 3.3.4.
Figure 3.12 shows that the upper bound on the probability of failure rapidly decreases to zero even for modest values of $P - N$.

3.3.6 Experiments

Section 3.3.5 provided a lower bound on the probability of successfully identifying the transform and the quantization step sizes. In this section, this aspect is evaluated experimentally. In addition, we provide further insight on the complexity of the algorithm, expressed in terms of the number of recursive steps needed to compute the sought solution.
To this end, we generated data sets of N-dimensional vectors, whose elements are sampled from a Gaussian random variable $\mathcal{N}(0, \sigma^2)$. We considered the adverse case in which the elements are independent and identically distributed. Therefore, the distribution of the vectors is isotropic and no clue could be obtained from a statistical analysis of the distribution. Without loss of generality, we set $\sigma = 2$, $\mathbf{W} = \mathbf{I}$ and $\Delta_i = 1$, $i = 1, \ldots, N$. The same results were obtained using different transform matrices and quantization step sizes.
Figure 3.13(a) shows the empirical probability of success when $N = 2, 4, 8, 16, 32, 64$, and the number of observed vectors P is varied, averaged over 100 realizations. As expected $p_{\text{succ}}(N, P) = 0$ when the number of vectors P does not exceed the dimensionality of the embedding vector space, i.e., $P \leq N$. Then, as soon as $P > N$, $p_{\text{succ}}(N, P)$ grows rapidly to one, when just a few additional vectors are visited. More specifically, Figure 3.13(b) illustrates the number of observed vectors P needed to achieve $p_{\text{succ}}(N, P) > 1 - \epsilon$, where ϵ was set equal to 10^{-15}. It is possible to observe that, when $N > 2$, the number of observed vectors needs to exceed by 6-7 units the dimensionality, and such a difference is independent from N, as expected based on the analysis in Section 3.3.5. Note that the results shown in Figure 3.13 are completely oblivious of the specific implementation of Algorithm 2.



(a) (b)

Figure 3.13: (a) Empirical probability of success of Algorithm 1 in identifying the transform and the quantization step sizes as a function of the number of observed vectors P and the dimensionality of the embedding vector space N. (b) Number of observed vectors P needed to achieve $p_{\text{succ}}(N, P) > 1 - \epsilon$, with $\epsilon = 10^{-15}$.

At the same time, it is interesting to evaluate the complexity when the specific instance of Algorithm 2, namely Algorithm 3, is adopted. Figure 3.14 shows the total number of recursive calls needed to converge to the solution of (3.26). Note that when a large enough number P of vectors is observed, the algorithm converges to the correct lattice $\mathcal{L}_x$. Thus, visiting additional vectors does not increase the number of recursive calls, since the base step of the recursion is always met. Figure 3.14 shows two cases, that differ in the way the set of observed vectors is visited, i.e., randomly, or sorted in ascending order of distance from the origin of the vector space. In both cases, the number of recursive calls grows linearly with N. This is aligned with the analysis in Section 3.3.5.2, which shows that convergence proceeds at a rate such that the number of recursive steps is upper bounded by $\lceil \log_2 |\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x| \rceil$. A (loose) bound on the lattice determinant is given by

$$|\mathcal{L}(\mathbf{B}^{(0)})| = |\det(\mathbf{B}^{(0)})| \leq \|\mathbf{b}_1^{(0)}\|_2 \|\mathbf{b}_2^{(0)}\|_2 \cdot \|\mathbf{b}_N^{(0)}\|_2 \leq \|\mathbf{b}_{\max}^{(0)}\|_2^N, \tag{3.55}$$

where the first inequality stems from Hadamard inequality and $\mathbf{b}_{\max}^{(0)}$ is the column of $\mathbf{B}^{(0)}$ with the largest norm. Therefore,

$$\lceil \log_2 |\mathcal{L}(\mathbf{B}^{(0)})|/|\mathcal{L}_x| \rceil \leq \lceil N \log_2(\|\mathbf{b}_{\max}^{(0)}\|_2)/|\mathcal{L}_x| \rceil \tag{3.56}$$

This explain the dependency on N, as well as the fact that sorting the vectors so as to initialize $\mathbf{B}^{(0)}$ with shorter vectors reduces the number of recursive calls.

3.3.7 Conclusions

In this section we proposed a method which is able to identify the parameters of a transform coder from a set of P transform decoded vectors embedded in a N-dimensional space. We proved that it is possible to successfully identify the transform and the quantization step sizes when $P > N$ and this

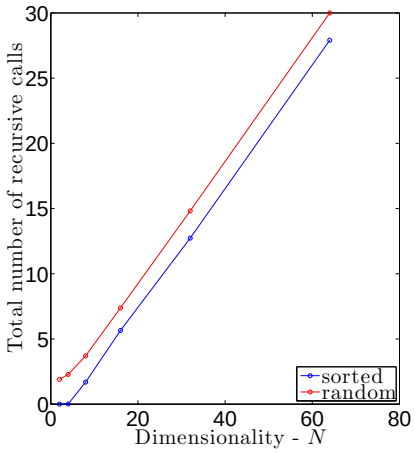


Figure 3.14: Total number of recursive calls to `recurseTI` as a function of the dimensionality of the space N and the strategy adopted to visit the observed vectors.

despite of the huge number of potential quantization bins, which grows exponentially with N for a target bitrate. In addition, we proved that the probability of failure decreases exponentially to zero when $P - N$ increases. In our experiments we found that an excess of approximately 6-7 observed vectors beyond the dimension N of the space is generally sufficient to ensure successful convergence. In this work, we focused on a noiseless scenario, in which we observe directly the output of the decoder. In some cases, though, signals are processed in complex chains, in which multiple transform coders are cascaded, thus introducing noise in the observed vectors. Consequently, the observed vectors do not lie exactly on lattice points. Extending the proposed method to this new scenario represents an interesting research avenue to be investigated.

3.4 Modeling reacquisition

In this section, we focus on modelling chains of multiple A/D and D/A conversion. Specifically, we focus on the case of single recapture, according to the pipeline shown in Fig. 3.15. Throughout the study, the input signal considered is modeled as a step function since in 2-D this feature would correspond to a straight edge - a feature abundant in natural images - and study under which conditions we can detect recapture and identify parameters of the original A/D and D/A operators. Ultimately, we aim to provide a complete end-to-end analytical method for reverse engineering of acquisition chains.

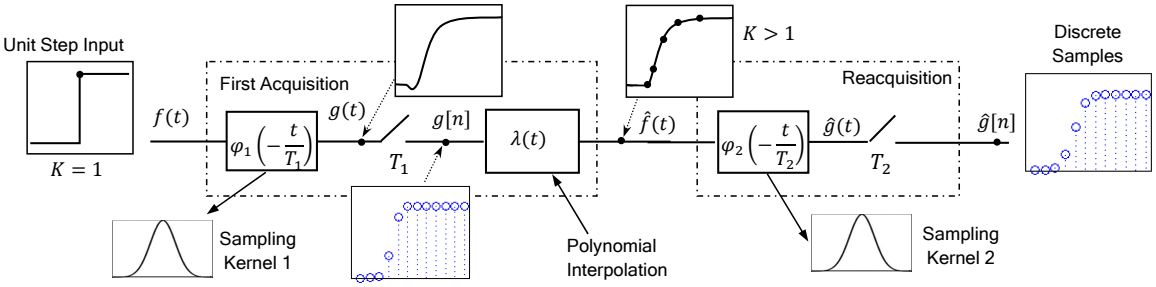


Figure 3.15: Problem statement diagram for signals with FRI in the chain of signal acquisition.

3.4.1 Preliminaries and Problem Setup

The process of sampling and acquisition of a signal is depicted in Figure 3.16, where the signal $x(t)$ is filtered before being uniformly sampled. This leads to the measurements $y[n] = \langle x(t), \varphi(t/T - n)\rangle$, where the sampling kernel is the time reversed and scaled version of the filter's impulse response $h(t)$. Reconstruction is achieved using the linear filter $\lambda(t)$ which yields $\hat{x}(t) = \sum_{n\in Z} y[n]\lambda(t/T - n)$. We assume $\lambda(t)$ is a polynomial spline or a MOMS function [70] of order R, therefore $\hat{x}(t)$ is a piecewise polynomial function of maximum order R. We also assume $\hat{x}(t) \neq x(t)$.

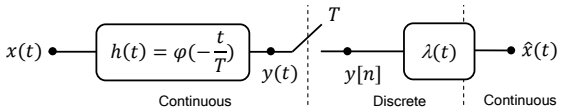


Figure 3.16: Classical signal acquisition and reconstruction model.

The sampling and reconstruction operations are then put in series as shown in Fig. 3.15. The parameters $\varphi_1(t)$, T_1, $\varphi_2(t)$, and T_2, are the sampling kernels and sampling periods of the first and the second acquisition devices respectively. With this chain structure, the problem conditions are set as follows:

1. The input signal is fixed as a box function $f(t) = u(t - t_1) - u(t - t_2)$, where t_1 and t_2 are the unknown locations of the unit step functions and $t_1 < t_2$. First we consider $t_2 \to \infty$, thus the input $f(t)$ can be approximated by a step $u(t - t_s)$. The role of t_2 will be discussed in Section 3.4.3.

2. The type of interpolation is polynomial interpolation with maximum degree R.

3. The second sampling kernel $\varphi_2(t)$ is one of those introduced in [71] and has the special properties that it can reproduce polynomials or exponentials.

Given access to only a query digital signal with an edge $q[n]$, the key questions for the problem of reverse engineering the acquisition chain are:

- What stages in the chain are the samples $q[n]$ from? That is, was $q[n]$ obtained by acquiring $f(t)$ directly with $\varphi_2(t)$ or was $q[n]$ the reacquired signal $\hat{g}[n]$ in Figure 3.15.
- In the case of reacquisition, how can we retrieve the following important parameters: i) maximum order of polynomial used for interpolation (R) ii) sampling period T_1, and iii) sampling kernel $\varphi_1(t)$?
- Under which condition on φ_2, T_2 can we solve (b)?

3.4.2 Sampling Theory for Signals with Finite Rate of Innovation

In this section we provide the preliminary mathematical background in FRI theory for our analysis in the next section. A signal with Finite Rate of Innovation (FRI) is defined as a signal which has a finite number of degrees of freedom per unit of time. Given a finite number of shifts t_k and amplitudes $\alpha_{k,r}$, a signal with FRI $x(t)$ can be described by known functions $\{f_r(t)\}_{r=0}^{R-1}$ as follows:

$$x(t) = \sum_{k\in\mathbb{Z}} \sum_{r=0}^{R-1} \alpha_{k,r} f_r(t - t_k). \tag{3.57}$$

Examples of signals with FRI include a stream of Diracs, a stream of differentiated Diracs, and piecewise polynomial functions. Calling the classical sampling diagram in Figure 3.16, now let us consider a sampling scheme for signals with FRI. Let the input signal $x(t)$ be a train of K Diracs which is described by K pairs of free parameters: the locations t_k and the amplitudes a_k as follows:

$$x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k). \tag{3.58}$$

The input is then sampled with sampling kernel $\varphi(t)$ with period T before discrete samples $y[n]$ are obtained. In this work, we assume that the sampling kernel used is a function that can reproduce polynomials as described in [71]. For polynomial reproducing kernels there exists a set of coefficients $c_{n,p}$ such that:

$$\sum_{n\in\mathbb{Z}} c_{n,p} \varphi(\frac{t}{T} - n) = t^p \ ; \ p = 0, 1, 2, ..., P. \tag{3.59}$$

Next, the moments τ_p of order p of the signal can be computed as follows:

$$\begin{aligned} \tau_p &= \sum_n c_{n,p} y[n] \overset{(a)}{=} \langle x(t), \sum_n c_{n,p} \varphi(t/T - n) \rangle \\ &\overset{(b)}{=} \langle \sum_{k=0}^{K-1} a_k \delta(t - t_k), \sum_n c_{n,p} \varphi(t/T - n) \rangle \\ &\overset{(c)}{=} \sum_{k=0}^{K-1} a_k t_k^p \ ; \ p = 0, 1, 2, ..., P, \end{aligned} \tag{3.60}$$

where (a) follows from the linearity of inner product while (b) and (c) are from equations (3.58) and (3.59) respectively. Once the moments τ_p; $p = 0, 1, ..., P$ and $P \geq 2K$ have been computed, the following Toeplitz matrix is constructed:

$$S = \begin{bmatrix} \tau_K & \tau_{K-1} & \cdots & \tau_0 \\ \tau_{K+1} & \tau_K & \cdots & \tau_1 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_P & \tau_{P-1} & \cdots & \tau_{P-K} \end{bmatrix}. \tag{3.61}$$

Note that, one can show [71] that S has always rank K (number of Diracs in $x(t)$) and that $x(t)$ is determined from the knowledge of the null space of S. Next let us consider an input signal which is a piecewise polynomial signal with K pieces of maximum degree $R \geqslant 0$, that is

$$x(t) = \sum_{k=1}^{K} \sum_{r=0}^{R} a_{k,r}(t - t_k)^r. \tag{3.62}$$

Clearly the $(R + 1)$ order derivative $x^{(R+1)}(t) = \frac{d^{(R+1)}x(t)}{dt^{(R+1)}}$ is given by a train of differentiated Diracs at the locations t_k as follows:

$$x^{(R+1)}(t) = \sum_{k=0}^{K-1} \sum_{r=0}^{R} r! a_{k,r} \delta^{(R-r)}(t - t_k). \tag{3.63}$$

We observe that $x^{(R+1)}(t)$ is a FRI signal with non-zero values $a_{k,r}$ at the locations t_k of discontinuities of the input $x(t)$. We note that the finite difference $z^{(1)}[n]$ satisfies [71]:

$$\begin{aligned} z^{(1)}[n] &= y[n+1] - y[n] \\ &= \langle x(t), \varphi(t/T - n - 1) - \varphi(t/T - n) \rangle \\ &= \langle \frac{dx(t)}{dt}, \varphi(t/T - n) * \beta_0(t/T - n) \rangle. \end{aligned} \tag{3.64}$$

Therefore, the moments of the derivative of $x(t)$ are given by $\tau_p = \sum_n c^{(1)}_{n,p} z^{(1)}[n]$, where $c^{(1)}_{n,p}$ are the polynomial reproduction coefficients of (3.59) for the new kernel $\varphi(t) * \beta_0(t)$. The moments of the $R + 1$ derivative of $x(t)$ can be obtained similarly. Finally, it is again possible to show that the Toeplitz matrix S of the moments of $x^{(R+1)}(t)$ has rank proportional to the degrees of freedom of $x^{(R+1)}(t)$.

3.4.3 Reacquisition Detection and the Retrieval of Chain Parameters

We are given a query digital signal $q[n]$ representing a 2D edge, and we would like to understand whether this is the result of a single capture of a unit step function $f(t) = u(t - t_s)$ with $\varphi_2(t)$ or whether this is the result of reacquisition. An illustrative example of the two possible shapes of $q[n]$ is shown in Figure 3.17. In Figure 3.17 (c) we show the case of a single acquisition of $f(t)$ shown in Figure 3.17 (a), whereas Figure 3.17 (d) shows a reacquired signal obtained after linear interpolation of (c) to yield 3(b) and sampling of 3(b) with $\varphi_2(t)$. We note that $g[n]$ and $\hat{g}[n]$ are hardly distinguishable yet they still contain all the information necessary to reverse engineering the acquisition chain as shown next.

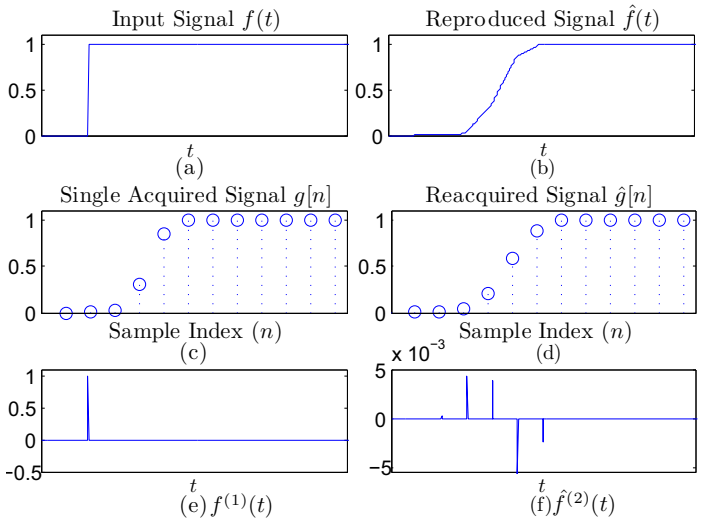


Figure 3.17: Comparative plots of continuous input signals (a) step input $f(t)$ (b) reconstructed signal $\hat{f}(t)$ with corresponding discrete samples (c) $g[n]$ and (d) $\hat{g}[n]$ and reconstructed locations using FRI sampling theory (e) and (f) respectively.

The input signal $f(t)$ is a unit step function which is described by only one free parameter - the location of step t_s. When the input signal $f(t)$ is acquired, the observed samples $g[n]$ are distorted by the sampling kernel. All possible $g[n]$, however, are still determined by one free parameter. In contrast, $\hat{f}(t)$ is obtained from polynomial interpolation and is a polynomial function with discontinuities at locations multiple of period T_1. The signal is a special case of FRI signals in (3.62).

We thereby use this principle to create an algorithm for reacquisition detection. We first aim to detect whether the query $q[n]$ was the result of single or double acquisition. Since a step function is a piecewise polynomial of maximum degree $R = 0$, the moments are computed using a first order finite difference of the query as $\tau_p = \sum_n c_{n,p} q^{(1)}[n]$. The moments are then used to construct the Toeplitz matrix S. The matrix S of size 2x2 is sufficient for reacquisition detection. Essentially the matrix is always rank-deficient with rank = 1 if $q[n]$ is acquired from a step input. On the other hand, if S is full rank, it means $q[n]$ stems from a recapture.

If the query signal is determined to be recaptured, the interesting question is how we can retrieve some important parameters of the chain including the sampling period T_1, interpolation function $\lambda(t)$, and the first sampling kernel φ_1.

Firstly, the maximum order R of polynomial interpolation function $\lambda(t)$ can be retrieved from the properties of FRI reconstructed signals. According to Section 3.4.2, piecewise polynomial functions of maximum order R are fully suppressed by differentiation of order $R + 1$. If we measure the number of degrees of freedom using Toeplitz matrix S, the matrix will be full rank until the finite difference of order $r \geq R + 1$ is applied to a query samples $q[n]$. When $r = R + 1$, the matrix will be rank deficient with rank K, equal to the number of K pieces of piecewise polynomial function. Figure 3.18 summarizes the retrieval algorithm for the order R using iterative finite difference and rank measurement until S is rank deficient.

Next, all the locations of discontinuities t_k and the continuous function $\hat{f}(t)$ can be retrieved using the annihilating filter method as discussed in [71]. Each t_k represents the location of samples $g[n]$ used

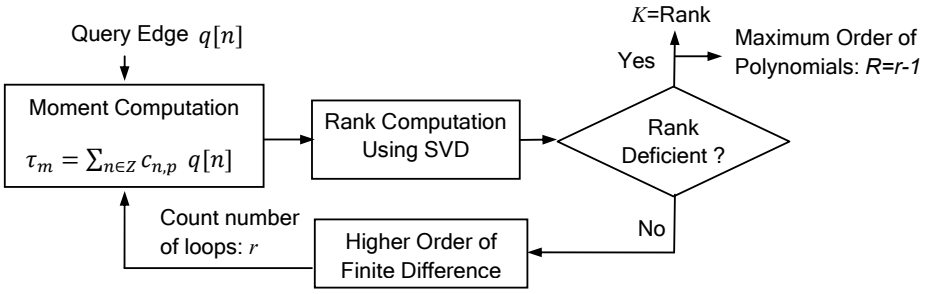


Figure 3.18: Iterative algorithm for the retrieval of maximum order R of polynomials used in interpolation function $\lambda(t)$.

in interpolation. From the retrieval results in Figure 3.17 (f), the distances between differentiated Diracs $\hat{f}^{(2)}(t)$ are uniform and the sampling period T_1 can be estimated from the average of the distances.

The retrieved $\hat{f}(t)$ and T_1 then can be used to estimate the samples $g[n]$ through the reverse sampling. Finally, the retrieval of φ_1 can be further achieved using the best matching between the samples and all possible dictionary elements as proposed in [72], which was previously included as part of D3.2. Instead we focus on providing the sufficient conditions on φ_2 and T_2 that allow us to retrieve the chain.

First, the maximum degree P of polynomials which the second kernel φ_2 can reproduce must be sufficiently large. From [71], the kernel must be able to reproduce polynomials of maximum degree $P \geqslant 2(R+1)K - R - 2$ in order to achieve perfect reconstruction of a piecewise polynomial of maximum degree R with K discontinuities. In our case, the unit step input signal is sampled with uniform sampling period T_1 and the samples are then interpolated to continuous domain again. The number of discontinuities can be computed as $K \leqslant \frac{L_1}{T_1} + 1$, where L_1 is the support of the first sampling kernel φ_1. Therefore, the order P which provides the precise retrieval results is given by $P \geqslant 2(R+1)\frac{L_1}{T_1} + R$.

Second, we consider the role of t_2 which is now the constant and $t_2 > t_1$. Consequently the input is a rectangular pulse $f(t) = u(t - t_2) - u(t - t_1)$. It is then acquired and reproduced by the chain. Since signal reconstruction creates a new group of K piecewise polynomials from samples of a unit step input, one needs to assure that two groups of piecewise polynomials are sufficiently distant in order to avoid the overlap. The minimum interval required is greater than $2KT_1$. From[71], a piecewise polynomial function with two groups of K pieces of maximum degree R can influence an interval of size $2K(L_2 + R + 1)T_2$. One therefore can calculate the bound $T_1 > (L_2 + R + 1)T_2$, which imposes a constrain on the maximum sampling period T_2. Here L_2 is the support of $\varphi_2(t)$.

When sampling signals satisfying the above requirements, one-to-one mapping between discrete samples and chains structures is guaranteed. We conclude by providing a counter example to show that signals obtained from different acquisition chains can be indistinguishable when the sufficient conditions are violated. Let $q_a[n]$ and $q_b[n]$ are query discrete samples acquired from different chain structures. The signal $q_a[n]$ is obtained from single acquisition of the step input $f_a(t) = u(t - (T_2 + \frac{T_2}{2}))$ using a box spline kernel [73] and $T_1 = T_2$ or $\varphi_{1a}(t) = \beta_0(\frac{t}{T_2})$. On the other hand, $q_b[n]$ is from reacquisition. Given that $f_b(t) = u(t - (T_2 + \frac{T_2}{4}))$ is the initial input, the signal is sampled using $\varphi_{1b}(t) = \beta_0(\frac{2t}{T_2})$ before the samples are reproduced to $\hat{f}_b(t)$ again by linear interpolation. From Figure 3.19, one can compute $q[n] = \langle f(t), \varphi_2(t/T_2 - n)\rangle$ and we have $q_a[n] =$

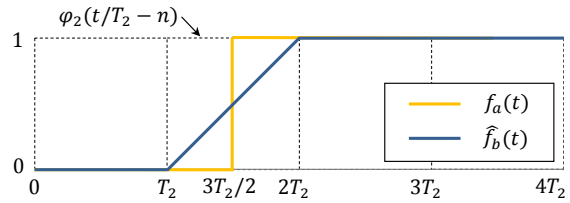


Figure 3.19: Counter examples when sampling $f_a(t)$ and $\hat{f}_b(t)$ with a sampling kernel $\varphi(t) = \beta_0(\frac{t}{T_2})$

$q_b[n] = [0\ \frac{1}{2}\ 1\ \frac{1}{2}\ 0]$. The signals from different chains become indistinguishable because the kernels used can reproduce polynomials up to degree P=0 which violates the condition. Thus one-to-one mapping is not guaranteed and the proposed algorithm cannot retrieve unique chain solution.

3.4.4 Conclusions

We have presented a theoretical scheme for the retrieval of a signal acquisition chain. With the theory of sampling signals with FRI, we are able to classify discrete samples from different stages in the chain model and create reacquisition detection algorithm. In addition, the method allows us to develop a retrieval algorithm for important parameters in the reacquisition chain. We finally have discussed the sufficient conditions to the retrieval of the parameters of the acquisition chain.

3.5 Demosaicking localization

An image coming from a digital camera, in the absence of any successive processing, will show specific traces due to the demosaicking algorithm applied on color filter array (CFA) elements. The analysis of such traces at the local level may be a powerful instrument to verify the history of a digital image, since demosaicking inconsistencies between different parts of the image will put image integrity in doubt. In this section, we will introduce a statistical model that can be used to characterize demosaicking artifacts at a local level. Namely, we propose a new feature that is related to the presence/absence of these artifacts on a patch as small as a single CFA element, and we will show that such a feature can be modeled by introducing a simple Gaussian mixture model (GMM).

3.5.1 CFA interpolation

During the CFA interpolation process, the estimation of the values in the new lattice based on the known values can be locally approximated as a filtering process through an interpolation kernel periodically applied to the original image to achieve the resulting image. Thus, the identification of artifacts due to CFA demosaicing can be seen as a particular case of the detection of interpolation artifacts.
In [43], Kirchner demonstrated that for a resampled stationary and non-constant signal $s(x)$, with $x \in \mathbb{Z}$, the variance of the residue of a linear predictor $\mathrm{Var}[e(x)]$ is periodic with a period equal to the original sampling rate. Hence, if we consider the signal resampled according to an integer interpolation factor r, we have $\mathrm{Var}[e(x)] = \mathrm{Var}[e(x+r)]$, since the original sampling period corresponds to r samples of the resampled signal.
For the case of CFA demosaicking, if we consider a single dimension, the general result presented in [43] turns into $\mathrm{Var}[e(x)] = \mathrm{Var}[e(x+2)]$, that is the variance of the prediction error assumes only two possible values, one for the odd positions and another one for the even positions. In more detail, considering for example the interpolation of the green color channel $G(x)$ in a particular row of the image, the acquired signal $s_A(x)$ is

$$s_A(x) = \begin{cases} G(x) & x \text{ even} \\ 0 & x \text{ odd} \end{cases} \tag{3.65}$$

If we consider a simplified demosaicking model, the resulting signal $s_R(x)$, composed by the acquired component $s_A(x)$ and by the interpolated component, takes values:

$$s_R(x) = \begin{cases} s_A(x) = G(x) & x \text{ even} \\ \sum_u h_u s_A(x+u) & x \text{ odd} \end{cases} \tag{3.66}$$

where h_u represents the interpolation kernel. In the above model, we assume that each color channel is independently interpolated using a linear filter and that original sensor samples are not modified by the interpolation process. In practice, since only odd values of u contribute to the above summation, we will restrict our attention to the case $h_u = 0$ for u odd. The prediction error is then defined as $e(x) = s_R(x) - s_P(x)$, with:

$$s_P(x) = \sum_u k_u s_R(x+u) \tag{3.67}$$

the predicted signal, and k_u the prediction kernel. Hence:

$$e(x) = \begin{cases} G(x) - \sum_u k_u s_R(x+u) & x \text{ even} \\ \sum_u h_u s_A(x+u) - \sum_u k_u s_R(x+u) & x \text{ odd} \end{cases} \tag{3.68}$$

By assuming to use the same kernel for the interpolation and the prediction (i.e. $h_u = k_u$), the prediction error in odd positions is identically zero, while in the even positions takes values different from zero. Hence, in such an ideal case, var$[e(x)]$ is expected to be zero in the positions corresponding to the demosaicked signal, and different from zero in the positions corresponding to the acquired signal.
In general, the exact interpolation coefficients may not be known, however we can assume that $k_u = 0$ for u odd. Moreover, we can also assume $\sum_u k_u = \sum_u h_u = 1$, which usually holds for common interpolation kernels. In this case, equation (3.68) above can be rewritten as

$$e(x) = \begin{cases} G(x) - \sum_u k_u \sum_v h_v G(x+u+v) & x \text{ even} \\ \sum_u (h_u - k_u) G(x+u) & x \text{ odd} \end{cases} \tag{3.69}$$

By assuming the acquired signal samples to be independent and identically distributed (i.i.d.) with mean μ_G and variance σ_G^2, the mean of the prediction error can be evaluated as

$$E[e(x)] = \begin{cases} \mu_G - \mu_G \sum_u k_u \sum_v h_v = 0 & x \text{ even} \\ \mu_G \left(\sum_u h_u - \sum_u k_u\right) = 0 & x \text{ odd} \end{cases} \tag{3.70}$$

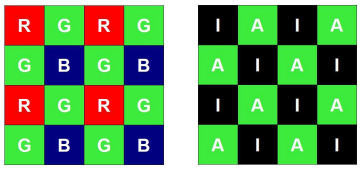
whereas the variance of the prediction error is

$$\begin{aligned} \mathrm{Var}[e(x)] =& \mathrm{Var}\Big[(1 - \sum_u k_u h_{-u}) G(x) \\ &+ \sum_{t \neq 0} \Big(\sum_u k_u h_{t-u} \Big) G(x+t) \Big] \\ =& \sigma_G^2 \Big[(1 - \sum_u k_u h_{-u})^2 + \sum_{t \neq 0} (\sum_u k_u h_{t-u})^2 \Big] \end{aligned} \tag{3.71}$$

for x even and

$$\mathrm{Var}[e(x)] = \mathrm{Var}\Big[\sum_u (h_u - k_u) G(x+u) \Big] = \sigma_G^2 \sum_u (h_u - k_u)^2 \tag{3.72}$$

for x odd. According to the above model, the prediction error has zero mean and variance proportional to the variance of the acquired signal. However, when the prediction kernel is close to the interpolation kernel, the variance of prediction error will be much higher at the positions of the acquired pixels than at the positions of interpolated pixels.
In order to extend the previous analysis to the bidimensional case, without loss of generality we will consider as specific CFA the most frequently used Bayer's filter mosaic, a 2×2 array having red and green filters for one row and green and blue filters for the other (see Fig. 3.20(a)). Furthermore, we will consider only the green channel; since the green channel is upsampled by a factor 2, for a generic square block we have the same number of samples (and the same estimation reliability) for both classes of pixels (either acquired or interpolated).
By focusing on the green channel, the even/odd positions (i.e. acquired/interpolated samples) of the one-dimensional case turn into the quincunx lattice $\mathcal{A}$ for the acquired green values and the complementary quincunx lattice $\mathcal{I}$ for the interpolated green values (see Fig. 3.20(b)). Similar to the one-dimensional case, we assume that in the presence of CFA interpolation the variance of the prediction error on lattice $\mathcal{A}$ is higher than the variance of the prediction error on lattice $\mathcal{I}$, and in both cases it is content dependent. On the contrary, when no demosaicking has been applied, the variance of the prediction error assumes similar values on the two lattices.



(a) (b)

Figure 3.20: (a) the Bayer's filter mosaic; (b) the quincunx lattice $\mathcal{A}$ for the acquired green channels and the complementary quincunx lattice $\mathcal{I}$ for the interpolated green channels.

3.5.2 Proposed model

The exact modeling of the prediction error would be in general tricky, since it depends on many parameters like the content of the image and the actual demosaicking filter. However, it is possible to consider a local function of the prediction error that can be modeled in a very simple and effective way. Let us suppose that $s(x, y)$, with $(x, y) \in \mathbb{Z}^2$, is an observed image. The prediction error can be obtained as:

$$e(x, y) = s(x, y) - \sum_{u,v \neq 0} k_{u,v} s(x + u, y + v) \tag{3.73}$$

where $k_{u,v}$ is a bidimensional prediction filter. In the ideal case, $k_{u,v} = h_{u,v}$ $\forall (u, v)$ where $h_{u,v}$ is the interpolation kernel of the demosaicking algorithm. In general, we can assume that $k_{u,v} \neq h_{u,v}$, since the in-camera demosaicking algorithm is usually unknown.
Because of the local stationarity of the residue, we consider a local estimation of the variance of the prediction error from a neighborhood of either interpolated ($\mathcal{I}$) or acquired ($\mathcal{A}$) pixels, according to the pixel location. By assuming that the local stationarity of prediction error is valid in a $(2K + 1) \times (2K + 1)$ window, it is possible to define the local weighted variance of the prediction error as:

$$\sigma_e^2(x, y) = \frac{1}{c}\Big[\Big(\sum_{i,j=-K}^{K} \alpha_{ij}\, e^2(x + i, y + j)\Big) - (\mu_e)^2\Big] \tag{3.74}$$

where α_{ij} are suitable weights, $\mu_e = \sum_{i,j=-K}^{K} \alpha_{ij}\ e(x + i, y + j)$ is a local weighted mean of the prediction error and $c = 1 - \sum_{i,j=-K}^{K} \alpha_{ij}^2$ is a scale factor that makes the estimator unbiased, i.e., $E[\sigma_e^2(x, y)] = \text{var}[e(x, y)]$, for each pixel class. The weights α_{ij} are obtained as $\alpha_{ij} = \alpha'_{ij} / \sum_{i,j} \alpha'_{ij}$ where

$$\alpha'_{ij} = \begin{cases} W(i, j) & \text{if } e(x + i, x + j) \text{ belongs to} \\ & \text{the same class of } e(x, y) \\ 0 & \text{otherwise} \end{cases}$$

and $W(i, j)$ is a $(2K + 1) \times (2K + 1)$ Gaussian window with standard deviation $K/2$.
Given a $N \times N$ image, we analyze it by considering $B \times B$ non-overlapping blocks, where B is related to the period of Bayer's filter mosaic: the smallest period (and block dimension) is $(2, 2)$, but also multiples can be adopted. The generic block in position (k, l) is denoted as $\mathcal{B}_{k,l}$ with $k, l = 0, \ldots, \frac{N}{B} - 1$. Each block is composed by disjoint sets of acquired and interpolated pixels, indicated as $\mathcal{B}_{A_{k,l}}$ and $\mathcal{B}_{I_{k,l}}$, respectively. We then define the feature $\mathbf{L}$:

$$\mathbf{L}(k, l) = \log\left[\frac{GM_A(k, l)}{GM_I(k, l)}\right] \tag{3.75}$$

where $GM_A(k,l)$ is the *geometric mean* of the variance of prediction errors at acquired pixel positions, defined as:

$$GM_A(k,l) = \left[\prod_{i,j \in \mathcal{B}_{A_{k,l}}} \sigma_e^2(i,j) \right]^{\frac{1}{|\mathcal{B}_{A_{k,l}}|}} \tag{3.76}$$

whereas $GM_I(k,l)$ is similarly defined for the interpolated pixels.
The proposed feature $\mathbf{L}$ allows us to evaluate the imbalance between the local variance of prediction errors when an image is demosaicked: indeed, in this case the local variance of the prediction error of acquired pixels is higher than that of interpolated pixels and thus the expected value of $\mathbf{L}(k,l)$ is a nonzero positive amount. On the other hand, if an image is not demosaicked, this difference between the variance of prediction errors of acquired an interpolated pixels disappears, since the content can be assumed to present locally the same statistical properties, and the expected value of $\mathbf{L}(k,l)$ is zero.
Let M_1 and M_2 be the hypotheses of presence and absence of CFA artifacts, respectively. In order to have a simple and tractable model, we assume that $\mathbf{L}(k,l)$ is Gaussian distributed under both hypotheses and for any possible size B of the blocks $\mathcal{B}_{k,l}$. For a fixed B, we can characterize our feature using the following *conditional probability density functions*:

$$Pr\{\mathbf{L}(k,l)|M_1\} = \mathcal{N}(\mu_1, \sigma_1^2) \tag{3.77}$$

with $\mu_1 > 0$, and

$$Pr\{\mathbf{L}(k,l)|M_2\} = \mathcal{N}(0, \sigma_2^2). \tag{3.78}$$

The above densities hold $\forall k,l = 0,\ldots,\frac{N}{B}-1$, i.e., we assume that the parameters of the two conditional pdfs do not change over the considered image, such that they can be globally estimated. If a demosaicked image contains some tampered regions in which CFA artifacts have been destroyed (as it may occur in a common splicing operation), both hypotheses M_1 and M_2 are present, therefore $\mathbf{L}(k,l)$ can be modeled as a mixture of Gaussian distributions. The first component, with $\mu_1 > 0$, is due to regions in which CFA artifacts are present, whereas the second component, with $\mu_2 = 0$, is due to tampered regions in which CFA artifacts have been removed. In order to estimate simultaneously the parameters of the proposed GMM, we employ the *Expectation-Maximization (EM) algorithm* [74]. This is a standard iterative algorithm that estimates the mean and the variance of the component distributions by maximizing the expected value of a *complete log-likelihood function* with respect to the distribution parameters. In our case, the EM algorithm is used to estimate only μ_1, σ_1, and σ_2, since we assume $\mu_2 = 0$.

3.5.3 Validation

The results presented in this section have been obtained on a dataset consisting of 400 original color images, in TIFF uncompressed format, coming from 4 different cameras (100 images for each camera): Canon EOS 450D, Nikon D50, Nikon D90, Nikon D7000. All cameras are equipped with a Bayer CFA, thus respecting our requirement that authentic images come from a camera leaving demosaicking traces, but the in-camera demosaicking algorithms of such devices are unknown. Each image was cropped to 512×512 pixels, maintaining the original Bayer pattern, which has been verified by inspecting the technical specifications of the raw image format. We will refer to such a dataset as the *original dataset*.
The first step was to verify the assumption of Gaussian distribution on $\mathbf{L}(k,l)$, both in the presence and in the absence of CFA artifacts. To this end, starting from 100 images selected from the *original dataset*, we have created two datasets satisfying the M_1 (presence of CFA) and M_2 (absence of CFA)

	bilinear	bicubic	gradient-based	median
No CFA	1.589	1.558	1.672	1.812
Ideal	2.168	2.134	2.049	2.016
Canon EOS	2.001	1.908	1.897	1.962
Nikon D50	1.736	1.797	1.834	1.814
Nikon D7000	2.206	2.066	1.729	1.899
Nikon D90	1.998	1.924	1.667	1.927

Table 3.5: Median value of the GGD shape parameters estimated from the distribution of the feature $\mathbf{L}(k,l)$ for each image, considering different predictors on different datasets.

hypotheses. To create the dataset corresponding to M_1, the original images have been sampled according to the Bayer CFA pattern and then re-interpolated using four possible demosaicking algorithms, namely bilinear, bicubic, gradient-based and median (see [75] for more details on such interpolation algorithms). This allowed us to know the interpolation kernel on the whole image, and then to exactly predict the interpolated values with the four different predictors (we refer to these cases as 'ideal'). To create the dataset corresponding to M_2, each color channel of the original images has been upsampled by a factor two, blurred with a 7×7 median filter, and downsampled by a factor two, thus removing all CFA artifacts. Features are then computed using again the four predictors as before.
Moving towards realistic conditions, we also computed the value of $\mathbf{L}(k,l)$ under the M_1 hypothesis on the *original dataset* of 400 TIFF uncompressed images interpolated using their unknown in-camera demosaicking algorithms, and applying bilinear, bicubic, gradient-based and median predictors.
We verified the approximate Gaussian distribution of the features for all the classes described so far, i.e.: absence of CFA, presence of CFA with known interpolation kernel, and the four sets of cameras with unknown CFA demosaicking algorithms; for each of these six classes, the features have been computed with the four different interpolation algorithms (bilinear, bicubic, gradient-based, median) setting $B = 8$. The approximately Gaussian behavior of the features has been verified by fitting them with a generalized Gaussian distribution (GGD), given by

$$p(\mathbf{L}) = \frac{1}{Z} e^{-(|\mathbf{L}-\mu|/\eta)^{\nu}} \tag{3.79}$$

where μ is a location parameter (mean), η is a scale parameter, ν is a shape parameter, and Z is a normalization factor so that $p(\mathbf{L})$ integrates to one. The Gaussian distribution is a particular case of the GGD for $\nu = 2$. Since our conjecture is that the Gaussian assumption holds for a single image, but not necessarily over the whole dataset, the shape parameter has been independently estimated for each image using the Mallat's method [76]. In Table 3.5 we report the median value of the estimated shape parameters for the six classes and the four interpolation algorithms. The values indicate a reasonable fit of the proposed model. Interestingly, the model appears more fitting in the presence of CFA artifacts, and when the predictor is matched to the actual interpolation algorithm.

Furthermore, we plot the mean value of the features in order to verify how features in M_1 hypothesis can be discriminated by features in M_2 hypothesis, both in ideal and in realistic cases. In Fig. 3.21, we show the results for the ideal case in absence of CFA (first row) and presence of known CFA (second row). In Fig. 3.22, we show the 16 histograms of the mean values of $\mathbf{L}(k,l)$: along each row we have histograms referring to the same camera, from top to bottom, Canon EOS 450D, Nikon D50,

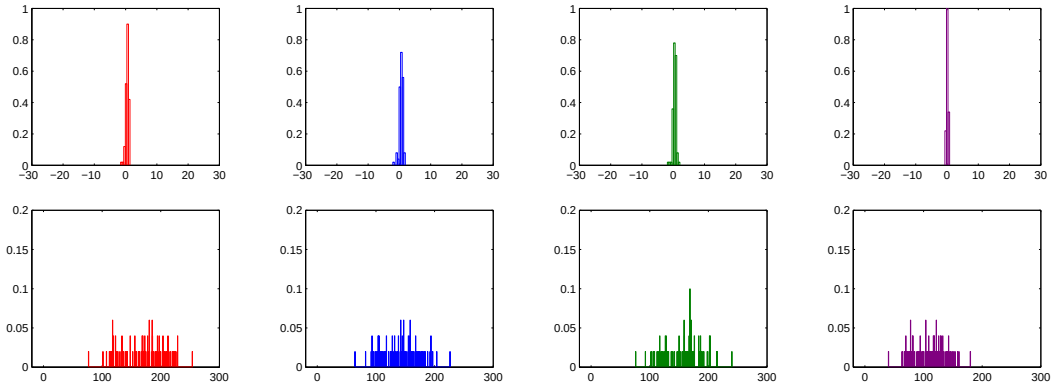


Figure 3.21: Distribution of the average value of $\mathbf{L}(k,l)$ on an image, feature evaluated on 8×8 blocks, in the absence of CFA artifacts (top row) and when the predictor is the same as the demosaicking algorithm (bottom row), using different predictors: from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).

Nikon D90, Nikon D7000. For both the Figures along each column we have histograms referring to the same predictor, from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).
Globally, the above results confirm that the proposed features has zero mean under the M_2 hypothesis and mean greater than zero under the M_1 hypothesis. The histograms also highlight that the four predictors have different behaviors. The median predictor does not seem well suited to detect CFA artifacts, since it produces values of $\mathbf{L}(k,l)$ closer to zero than the other predictors, irrespective of the camera.

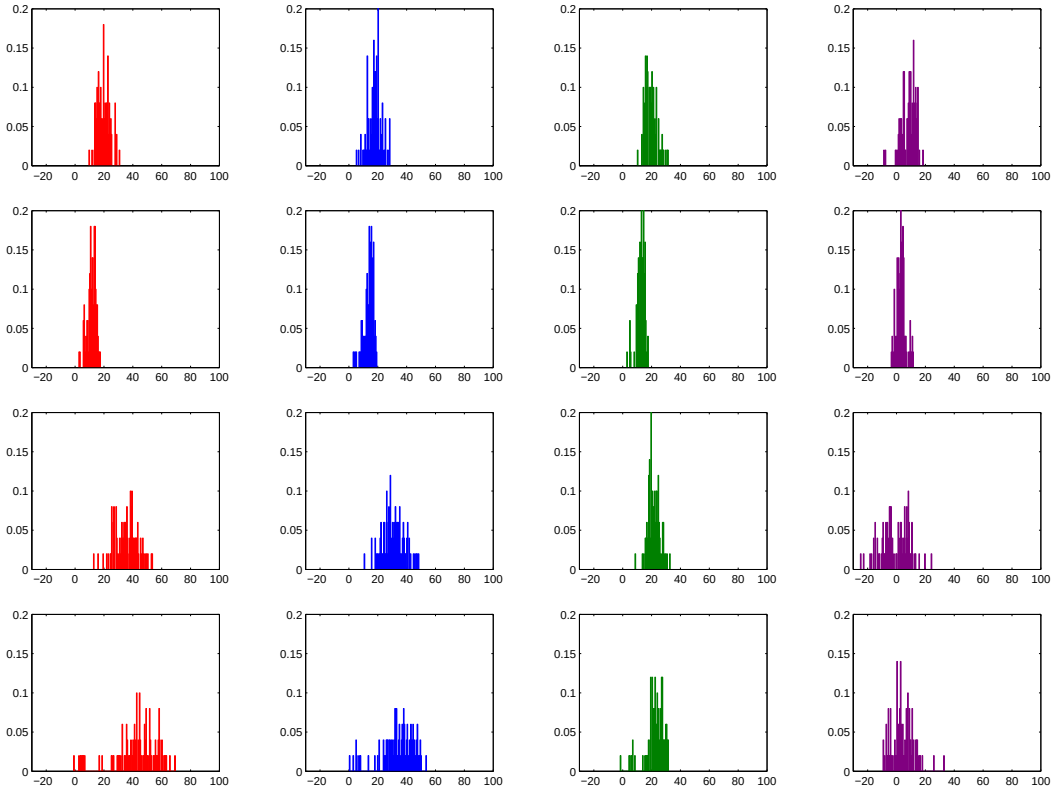


Figure 3.22: Distribution of the average value of $\mathbf{L}(k,l)$ on an image, feature evaluated on 8×8 blocks, with unknown in-camera demosaicking algorithms and using different predictors: along each row we have histograms referring to the same camera, from top to bottom, Canon EOS 450D, Nikon D50, Nikon D7000, Nikon D90; along each column we have histograms referring to the same predictor, from left to right, bilinear (red), bicubic (blue), gradient-based (green), median (violet).

Chapter 4

Synergies with WP3 and WP4

4.1 JPEG Quantization and full-frame filtering

As introduced in Sec.3.1, we studied the DCT coefficients of a full-frame linearly filtered JPEG image and derived an accurate mathematical analysis establishing the theoretical relationship between the DCT coefficients before and after filtering. Such theoretical framework allows to fully characterize the probabilistic distributions of the DCT coefficients of the quantized and filtered image, given different kernels. When testing the mathematical model we faced an inconsistency between the theoretically derived models and the actual distributions of DCT coefficients of compressed filtered images, affecting the goodness of the fit. By further investigating this incongruence, we showed that the DCT coefficients corresponding to different frequencies cannot be considered independent, and those of the same frequency cannot be assumed to be i.i.d. distributed. Although in the literature this is a typical assumption (e.g., in [36][37][40], where it appears to be convenient and reasonable), such approximation would compromise here the conducted analysis. By considering those inter- and intra-block redundancies of the DCT coefficients in the provided analysis, we were able to redefine the proposed theoretical model and provide an accurate mathematical framework of the considered processing chain.
Following we briefly recall the derived mathematical model, as described in Section 3.1 and describe in details the steps taken to redefine the theoretical models taking into account the inter- and intra-block redundancy.

4.1.1 Theoretical model for the DCT coefficients of a JPEG filtered image.

We derive a theoretical model to describe the statistical properties of an image that has been first JPEG compressed and then linearly filtered. In order to do that, we mathematically express the deterministic relation between the quantized DCT coefficient $\hat{d}_q(x,y)$ and those of the JPEG and filtered image $d_f(x,y)$. Recalling Sec. 3.1.2, we can express such relationship as follows:

$$d_f(x,y) \quad = \quad \gamma \cdot \hat{d}_q(x,y) + N, \tag{4.1}$$

where $\gamma, N \in \mathbb{R}$ are a scaling factor and a noise term, respectively. These two terms can be calculated, through some math, according to (4.2), where $\alpha = x \text{mod} 8$, $\beta = y \text{mod} 8$, $\alpha' = k_1 \text{mod} 8$, $\beta' = k_2 \text{mod} 8$, $\text{DCT}_{x,y}$ is the (x,y)-th DCT coefficient obtained from an 8×8 pixel block, $\text{IDCT}_{x,y}$ is the 8×8 pixel

$$d_f(x,y) = \mathrm{DCT}_{\alpha,\beta}\left(\left[\mathbf{h} * \mathrm{IDCT}_{\alpha,\beta}\left(\hat{d}_q(x,y)\right)\right]_{\lfloor\frac{x}{8}\rfloor,\lfloor\frac{y}{8}\rfloor}^{\lfloor\frac{x}{8}\rfloor+7,\lfloor\frac{y}{8}\rfloor+7}\right) + \sum_{\substack{(k_1,k_2)\in\{-8,\dots,15\}^2 \\ (k_1,k_2)\neq(\alpha,\beta)}} \mathrm{DCT}_{\alpha,\beta}\left(\left[\mathbf{h} * \mathrm{IDCT}_{\alpha',\beta'}\left(\hat{d}_q\left(\left\lfloor\frac{x}{8}\right\rfloor+k_1, \left\lfloor\frac{y}{8}\right\rfloor+k_2\right)\right)\right]_{\lfloor\frac{x}{8}\rfloor-\lfloor\frac{k_1}{8}\rfloor,\lfloor\frac{y}{8}\rfloor-\lfloor\frac{k_2}{8}\rfloor}^{\lfloor\frac{x}{8}\rfloor-\lfloor\frac{k_1}{8}\rfloor+7,\lfloor\frac{y}{8}\rfloor-\lfloor\frac{k_2}{8}\rfloor+7}\right). \tag{4.2}$$

block (located at $\{\lfloor\frac{x}{8}\rfloor,\dots,\lfloor\frac{x}{8}\rfloor+7\}\times\{\lfloor\frac{y}{8}\rfloor,\dots,\lfloor\frac{y}{8}\rfloor+7\}$) obtained by applying the IDCT to the (x,y) DCT coefficient, $*$ denotes the bidimensional convolution, and $[\mathbf{A}]_{a,b}^{c,d}$ denotes the submatrix of an arbitrary matrix $\mathbf{A}$ with first index taking values in $\{a,\dots,b\}$, and second index in $\{c,\dots,d\}$. N stands for the second term in the summation in (4.2).
Then, by exploiting the knowledge about the statistical properties of the distribution of $\hat{d}_q(x,y)$, we further analyze the histograms of $d_f(x,y)$ and assess the dependency of the different frequencies.

4.1.1.1 Probability distribution

From probability theory [39], given two discrete independent random variables, the probability density function (pdf) of their sum is the convolution of their corresponding pdfs. Recalling Sec. 3.1.2.1, it is known that the probability mass function of each frequency coefficient of a JPEG image can be expressed as an impulse train, with each impulse located at multiples of the applied quantization step $\Delta(i,j)$. Therefore, according to the derived mathematical model in (4.2), and based on the common DCT coefficients models, which typically assume the different frequency components to be independent and the coefficients in a given frequency to be i.i.d. [40], we would expect the probability distribution of the DCT coefficients $d_f(x,y)$ to be the result of a convolution between a train of impulses located at $\gamma\cdot k\Delta(i,j)$, with $\gamma\in\mathbb{R}$, and a noise component due to the contributions of all the neighboring coefficients (4.2). Moreover, according to the Central Limit theorem, we can model the noise component as a GGD distributed variable, with GGD parameters depending on each centroid $\gamma\cdot k\Delta(\cdot,\cdot)$ about which such noise is centered. However, we experienced a divergence between the classical theoretical models and the empirical data, showing that indeed the typical assumptions on the DCT coefficients distribution of natural images do not hold.
This suggests the need for a different model for the noise component in (4.2), which cannot any longer be considered as the addition of independent variables (coefficients of different frequencies) and i.i.d. components (coefficients in the same frequency).

4.1.1.2 Redefinition of the theoretical model

In Fig. 4.1 it is shown an example of DCT histogram where we can observe a scaling between the peaks in the histogram of the DCT coefficients of the compressed and filtered image and the impulse train identifying the location of the translated quantization step $\gamma\cdot k\Delta(\cdot,\cdot)$ (red bars). In order to justify this inconsistency between theoretical and empirical data we analyze the mean of the noise component and verify that for real images it monotonically increases with the quantized samples value. As an example, in Fig. 4.2 the red curve represents the mean of the noise component when a Moving Average filter of size 3×3 is applied, plotted with respect to each translated quantized value

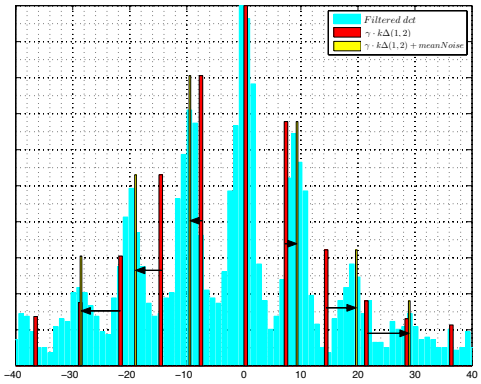
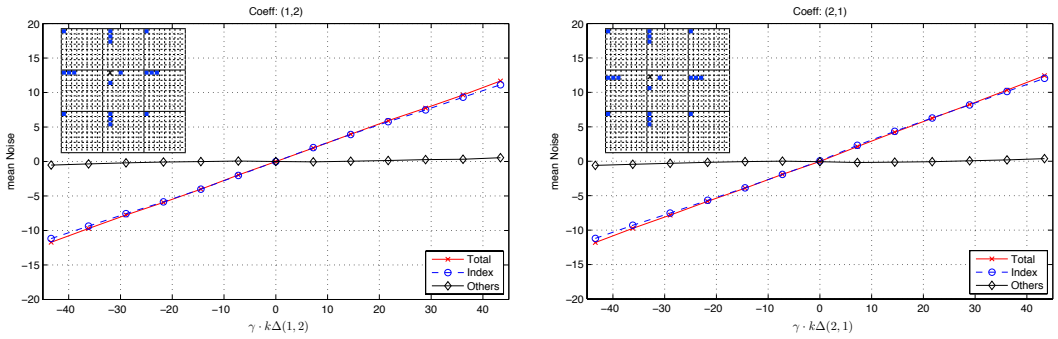


Figure 4.1: Probability distribution, at frequency $(1, 2)$, quantized with a step $q(1, 2) = 10$ and filtered with a 3×3 averaging filter. The red impulses represent the location of $\gamma \cdot k\Delta(1, 2)$ while the yellow ones are translated by the mean of the Noise component. The latter perfectly match with location of the peaks in the histogram, thus allowing an accrue redefinition of the theoretical model fitting perfectly the empirical data.

$\gamma \cdot k\Delta(i, j)$; similar behaviors have been verified for different filter kernels. Moreover, we investigate the verified divergence between the behavior in natural images and the theoretically modeled one, by determining the main causes of the noise mean component. For each coefficient $d_f(x, y)$, we isolate the contribution of each 24×24 coefficient and analyze their influence. In Fig. 4.2 (a)-(b) we show the specific pattern of coefficients mainly contributing to the noise, for all the AC coefficients $(1, 2)$ and $(2, 1)$. The red curve represents the mean of the total noise component over $\gamma \cdot k\Delta(\cdot, \cdot)$, the blue curve represents the contribution of a specific set of coefficients and the black curve corresponds to the contribution of all the remaining coefficients not specified in the previous set. This set was determined by isolating those coefficients that provided a significant noise contribution, in absolute value, over all $\gamma \cdot k\Delta(\cdot, \cdot)$ (i.e., above an empirically determined threshold.) Note that the 24×24 grid in the upper left part of each plot corresponds to the 9 DCT blocks taken into account in (4.1), when employing a kernel filter of size smaller than or equal to 17. The black dot identifies the considered frequency and the blue dots correspond to set of coefficients which mainly influence the total noise, as verified by the curve matching. We report only the behavior for the coefficients $(1, 2)$ and $(2, 1)$, but similar results are given for different frequencies.
Therefore, an accurate model for the distribution of the DCT coefficients of a filtered JPEG image can be redefined, taking into account the scaling inferred by the noise component. In particular, the distribution of DCT coefficients, quantized and filtered with a given kernel, will be the sum of many GGDs, each of them centered in $\gamma \cdot k\Delta(\cdot, \cdot)$ translated by the mean of the noise component and with amplitude depending on the distribution of the quantized and not filtered coefficients.
In Fig. 4.1 it is shown how the translated impulses, now centered in $\gamma \cdot k\Delta(\cdot, \cdot)$ plus the mean of the noise component, match with the peaks of the histogram.



(a) (b)

Figure 4.2: Inter- and intra-block spatial redundancy affecting to $d_f(x, y)$, over the entire image database. In each panel, corresponding to the frequencies $(1, 2)$ and $(2, 1)$, respectively (black cross in the grid in the upper left of each subplots), the total mean of the noise component (red curve) is plotted with respect to the values of the quantized filtered coefficients $\gamma \cdot k\Delta q(i, j)$. The blue curve represents the contribution of a set of coefficients (depicted in blue in the grid) and the black curve corresponds to the contribution of all the remaining coefficients not specified in the previous set.

4.2 Transform coder identification based on quantization footprints and lattice theory

Transform coding has emerged over the years as the dominating compression strategy. Transform coding is adopted in virtually all multimedia compression standards including image compression standards such as JPEG [77] and JPEG 2000 [78, 79] and video compression standards such as, for example, H.264/AVC [80] and HEVC [81]. This is due to the fact that transform coders are very effective and yet computationally inexpensive since the encoding operation is divided into three relatively simple steps: the computation of a linear transformation of the data, scalar quantization of each coefficient, and entropy coding.

Due to its centrality to any type of multimedia data, transform coding theory is now extensively used in a new range of applications that rely on the possibility of reverse-engineering complex chains of operators starting from the available output signals. This has stimulated a rich set of activities in WP3, which were focused on exploiting the footprints left by single, double and multiple compression, for the case of image, video and audio signals. All these works required prior knowledge of the type of standard being considered. This implies that the specific transform in use is assumed to be known, whereas the quantization step sizes need to be estimated. Although earlier standards (e.g., JPEG, MPEG-2 and MPEG-4) adopted the Discrete Cosine Transform (DCT) on 8×8 blocks, more recent coding architectures addressed by the application scenarios identified in WP5 are more diversified in terms of both the type of transform being used and the block size. For example, JPEG2000 is based on a wavelet transform that can be applied to tiles whose size can extend to the whole image [82]. H.264/AVC adopts an approximation of the DCT transform on 4×4 blocks, which can be implemented using integer arithmetic [83]. In addition, in the high-profile, H.264/AVC enables the adaptive choice between 4×4 and 8×8 transform block sizes. The recent HEVC standard under development goes even further, supporting four different transform block sizes in the range from 4×4 to 32×32 [81]. The core transforms used for 4×4 and 8×8 are the same as in H.264/AVC,

while for the two larger block sizes, integer transforms based on Chen's fast algorithm for the DCT are used [84]. In addition, the adoption of hybrid transforms, which can be obtained by means of a separable combination of DCT and DST (Discrete Sine Transform) is being investigated in [85]. All this indicates that the identification of the actual transform being used might give important clues about the processing history of a digital signal.

Most of the methods investigated in WP3 focused only on a specific type of multimedia signal (e.g., only images or only videos) and are to some extent ad-hoc. It is therefore natural to try and develop a universal theory of transform coder identification that is independent of the specific application at hand. To this end, in WP2 we considered a general model of transform coding that can be tailored to describe a large variety of practical implementations that are found in lossy coding systems, including those adopted in multimedia communication. This has led to a transform coder identification algorithm which leverages quantization footprints and lattice theory. The proposed method is able to successfully determine the parameters of the transform coder when a very small number of vectors are observed as output of a transform coder.

However, a comparison of the model adopted in WP2 with the application scenarios in WP5 (e.g., UC-04, UC-10, etc.), reveals that, in many circumstances, signals are compressed multiple times. This provided a feedback to WP2, thus suggesting a more general model, which consists of cascading two transform coders, with the goal of identifying the parameters of both of them. With respect to the scenario addressed in Section 3.3 of this deliverable, this requires to handle noisy observation, where noise might be introduced by the quantizer of the second transform coder. Although not described in this deliverable, we have obtained very promising results also along this direction, which stimulated further investigations to be continued during the third year of the project, despite of the fact that WP2 will be officially ended.

4.3 Modeling reacquisition chains

As part of the theoretical analysis carried out in WP2 and detailed in Section 3.4 in this deliverable, we constructed a model characterising image reacquisition, as shown in Figure 4.3.

Our chain model has been characterised under specific conditions. Given the assumptions of an input belonging to the class Finite Rate of Innovation (FRI) signals, and exponential-reproducing sampling kernels, we have shown how to recover several important parameters relative to the first unknown acquisition stage. In particular, we have shown how under these assumptions it is possible to recover the sampling period and order of the interpolator for chain structure identification, recapture detection and the parameter retrieval, all within a completely deterministic framework.

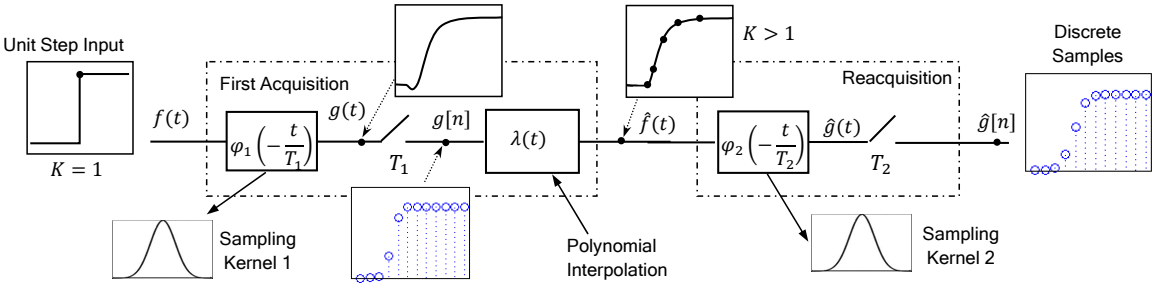


Figure 4.3: Image recapture model.

The model developed was then applied to the practical problem of camera identification and recapture detection of natural images. The work carried out as part of WP2 created the theoretical foundations upon which the practical footprint detector was then developed.
However, rather than a direct implementation of the model developed, the footprint detector presented in D3.2 relaxed several of the assumptions made in order to be able to operate with natural images. In this case, the input from natural images does not always belong to the class of FRI signals and the sampling kernels can be of arbitrary shape. We have therefore extended the framework described in this deliverable by relaxing the assumption on sampling kernels which are now extracted from a dictionary of acquisition devices. The method proposed in D3.2 finds the dictionary elements which can best represent edge signals in a query image. While the kernel shape assumption was relaxed during the transition from theoretical model to practical footprint detector, the FRI signal theory was still central to the detector developed, and the kernel dictionary was built from edges observed in natural images that well approximate the properties of FRI signals.
Finally, the information flow is not unidirectional, as the theoretical model was adjusted and expanded following implementation of the footprint detector. In Figure 4.4, the connections between the various components of the theoretical framework and footprint detector are highlighted. The backconnections from WP3 to WP2 represent the analysis of sufficient conditions for acquisition chain retrieval whenever the input signal is an image edge, as well as the identification of the first sampling kernel using the dictionary method outlined in D3.2.

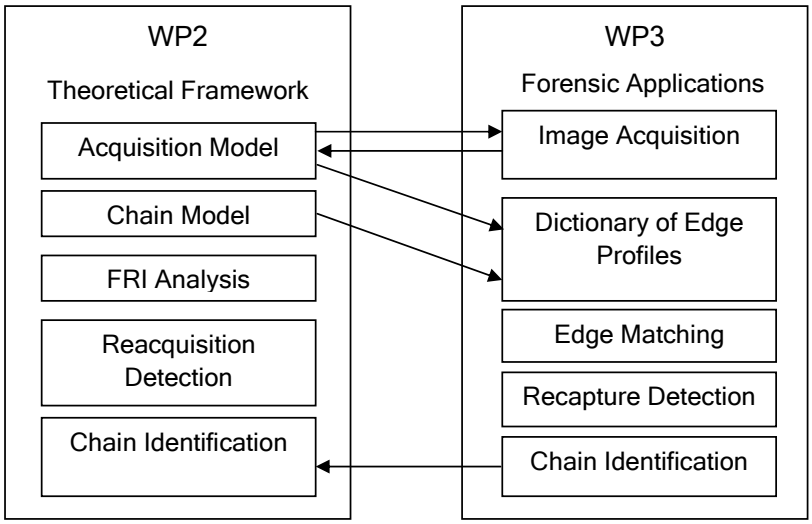


Figure 4.4: Flow of information between WP2 and WP3 contributions.

4.4 Demosaicking localization

The results obtained with the tool for the localization of color filter array (CFA) artifacts based on the above described mathematical model indicate that the proposed method is less effective in the presence of either almost flat areas or sharp edges. This suggests some possible directions for the extension of the above mentioned model. In the case of flat areas, we can observe that the prediction error is almost zero irrespective of the presence of CFA artifacts. This fact appears as an intrinsic

limitation of the proposed model. A more accurate model could assume that the distribution of the proposed feature is in general a mixture of two (or more) Gaussian components even in the absence of an explicit manipulation of the CFA structure. Such components should be associated with areas of the image exhibiting similar statistical patterns. For example, different regions of the image could be segmented according to the variance of the content and a specific Gaussian component could be used to model each class. Since flat areas will be likely associated to components with a mean very close to zero, the localization of forgeries should look for local inconsistencies in the modeling of the proposed features that do not correspond to analogous inconsistencies in the local variance of the content. In the current implementation of the tool, such an analysis is based on human interpretation of the forgery maps: an ambitious goal is that of devising a model permitting an automatic evaluation of such clues. Further research will be devoted to extending the current models according to the above insights.
On the other hand, the low performances observed in the presence of sharp edges can be ascribed to the signal adaptive and possibly non-linear behavior of realistic in-camera demosaicking algorithms. At least in theory, such effects could be eliminated by using some prior knowledge regarding in-camera CFA interpolation, which should yield results very close to the ideal behavior foreseen by the theoretical model. An alternative approach could be that of reverse engineering the CFA interpolation algorithm, for example using methods such as those described in [86] to take into account a signal adaptive behavior. However, in the presence of heavily manipulated images this approach is likely to produce a biased estimate and must be handled with care.
A final observation regards the fact that detection performance is strongly affected by JPEG compression, limiting the applicability of the present model to scenarios in which the image under test is either uncompressed or compressed with high quality factors. Unfortunately, in this case the results of WP3 do not offer sufficient clues about possible improvements of the current model. Further research should be devoted to assessing whether there is a fundamental limit regarding the detectability of CFA artifacts in the presence of JPEG compression.

4.5 Double JPEG Compression Models

The experimental evaluation of the tools for the localization of image forgeries based on double JPEG compression statistics has offered some interesting insights on the accuracy of the related mathematical models. Usually, the likelihood maps produced according to the above mentioned models show some false alarms in image regions with either low intensity variance, like a very uniform blue sky, or saturated values, which can occur in the presence of overexposition. As in the case of CFA artifacts, the above behaviour suggests that the model should also take into account the local statistics of the image. However, in this case the extension of the current models does not appear as straightforward as in the CFA case.
A very simple observation is that in the presence of flat areas we will have most of the AC DCT coefficients quantized to zero, irrespective of the occurence of either single or double compression. This suggests to include in the model a reliability factor depending on the number of DCT coefficients having value equal to zero. A similar correction has been already applied in the case of the simplified model described in Section 2.2.3 of the Deliverable 2.1 and the results confirms that this is indeed beneficial. Nevertheless, further research is needed to devise suitable strategies for incorporating local image statistics in the current models.
The results obtained with the developed tools also show that it is usually very difficult to separate the distributions of singly compressed and doubly compressed DCT coefficients when the quality of the second compression is lower than the quality of the first one. A possible solution to overcome

this problem could be that of jointly modeling the DCT coefficients of several DCT frequencies, however at the cost of a much more complex model. Moreover, as in the case of CFA artifacts, an interesting question is whether there is some fundamental limit regarding the detectability of double JPEG compression. The development of such a theory will be a future research topic.

Appendix A

Proof of Lemma 1

We start by remembering that for a memoryless source we have [13]:

$$n\mathcal{D}(P_{x^n}||P_X) = -\log(P_X(x^n)) - nH(P_{x^n}). \tag{A1}$$

By applying the above property to the right-hand side of equation (2.11), we obtain:

$$\begin{aligned} &n\mathcal{D}(P_{x^n}||P_X) + N\mathcal{D}(P_{t^N}||P_X) = \\ &- nH(P_{x^n}) - NH(P_{t^N}) - \log P_X(r^{n+N}), \end{aligned} \tag{A2}$$

where we have used the memoryless nature of P_X due to which $P_X(r^{n+N}) = P_X(t^N) \cdot P_X(x^n)$. For any $P_X \in \mathcal{C}$ we also have[1]:

$$P_X(r^{n+N}) \leq \prod_{a\in\mathcal{X}} P_{r^{n+N}}(a)^{N_{r^{n+N}}(a)}, \tag{A3}$$

where $N_{r^{n+N}}(a)$ indicates the number of times that symbol a appears in r^{n+N}, and where equality holds if $P_X(a) = P_{r^{n+N}}(a)$ for all a. By applying the log function we have:

$$\begin{aligned} \log P_X(r^{n+N}) &\leq \log \prod_{a\in\mathcal{X}} P_{r^{n+N}}(a)^{N_{r^{n+N}}(a)} \\ &= \log \prod_{a\in\mathcal{X}} P_{r^{n+N}}(a)^{(N_{x^n}(a)+N_{t^N}(a))} \\ &= \sum_{a\in\mathcal{X}} N_{x^n}(a) \log P_{r^{n+N}}(a) + \\ &\quad \sum_{a\in\mathcal{X}} N_{t^N}(a) \log P_{r^{n+N}}(a). \end{aligned} \tag{A4}$$

By inserting the above inequality in (A2), and by using the definition of empirical KL divergence we obtain:

$$\begin{aligned} &n\mathcal{D}(P_{x^n}||P_X) + N\mathcal{D}(P_{t^N}||P_X) \\ &\geq \sum_{a\in\mathcal{X}} N_{x^n}(a) \log \frac{P_{x^n}(a)}{P_{r^{n+N}}(a)} + \sum_{a\in\mathcal{X}} N_{t^N}(a) \log \frac{P_{t^N}(a)}{P_{r^{n+N}}(a)} \\ &= n\mathcal{D}(P_{x^n}||P_{r^{n+N}}) + N\mathcal{D}(P_{t^N}||P_{r^{n+N}}), \end{aligned} \tag{A5}$$

where the equality holds if $P_X = P_{r^{n+N}}$, thus completing the proof.

[1] Relationship (A3) can be easily proved by resorting to Jensen's inequality.

Bibliography

[1] Harry L. van Trees, *Detection, Estimation, and Modulation Theory*, Wiley-Interscience, 2001.

[2] H. Vincent Poor, *An Introduction to Signal Detection and Estimation*, Springer, 1994.

[3] Michele A. Saad, Alan C. Bovik, and Christophe Charrier, "Blind image quality assesment: A natural scene statistics approach in the dct domain," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3339–3352, August 2012.

[4] C. Cachin, "An information-theoretic model for steganography," *Information Hiding Internation Workshop*, vol. 1525, pp. 306–318, 1998.

[5] J. O'Brien and H. Farid, "Exposing photo manipulation with inconsistent reflections," *ACM Transactions on Graphics*, vol. 1, no. 31, pp. 1–11, 2012.

[6] Rainer Bohme and Matthias Kirchner, *Digital Image Forensics*, chapter Counter-Forensics: Attacking Image Forensics, Springer, 2012.

[7] Martin J. Osborne and Ariel Rubinstein, *A Course in Game Theory*, The MIT Press, 1994.

[8] Drew Fudenberg and Jean Tirole, *Game Theory*, The MIT Press, 1991.

[9] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Forensics vs. anti-forensics: a decision and game theoretic framework," in *ICASSP 2012, IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 25-30 March 2012.

[10] M. Barni, "A game theoretic approach to source identification with known statistics," in *ICASSP 2012, IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 25-30 March 2012.

[11] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[12] M. Barni and B. Tondi, "The source identification game: an information-theoretic perspective," *submitted to IEEE Transactions on Information Forensics and Security*, 2012.

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, 1991.

[14] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems. 2nd edition*, Cambridge University Press, 2011.

[15] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[16] M. Goljan, J. Fridrich, and M. Chen, "Sensor noise camera identification: countering counter forensics," in *SPIE Conference on Media Forensics and Security, San Jose, CA*, 2010.

[17] M. Gutman, "Asymptotically optimal classification for multiple tests with empirically observed statistics," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 401–408, March 1989.

[18] M. Kendall and S. Stuart, *The Advanced Theory of Statistics, vol. 2, 4th edition*, MacMillan, New York, 1979.

[19] "Special issue on digital forensics," *IEEE Signal Processing Magazing*, vol. 26, no. 2, March 2009.

[20] Simone Milani, Pier Francesco Piazza, Marco Tagliasacchi, and Stefano Tubaro, "An audio-video compatibility test for multimedia tampering detection," Submitted to ACM Information Hiding and Multimedia Security 2013.

[21] Pedro Comesaña, "Detection and information theoretic measures for quantifying the distinguishability between multimedia operator chains," in *Proc. of IEEE WIFS*, Tenerife, Spain, December 2012, pp. 211–216.

[22] David Slepian and Jack W. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[23] Aaron D. Wyner and Jacob Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, January 1976.

[24] Md. Saifur Rahman and Aaron b. Wagner, "On the optimality of binning for distributed hypothesis testing," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6282–6303, October 2012.

[25] R. Neelamani, R. de Queiroz, Z. Fan, S. Dash, and R.G. Baraniuk, "JPEG compression history estimation for color images," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1365–1378, June 2006.

[26] J. Rhedi, W. Taktak, and J-L. Dugelay, "Digital image forensics: a booklet for beginners.," *Multimedia Application Tools*, vol. 51, pp. 133–162, 2011.

[27] W. Luo, J. Huang, and G. Qiu;, "JPEG error analysis and its application to digital image forensics," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 480–491, 2010.

[28] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.," *Philosophical Magazine*, vol. 50, pp. 157–175, 1900.

[29] D. Xu, T. Cham, S. Yan, L. Duan, and S. Chang, "Near duplicate identification with spatially aligned pyramid matching.," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 20, pp. 1068–1079, 2010.

[30] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study.," *International Journal of Computer Vision*, vol. 73, pp. 213–238, 2007.

[31] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems - breaking the steganographic utilities ezstego, jsteg, steganos, and s-toolsand some lessons learned.," *Lectures Notes on Computer Science, Springer-Verlag*, vol. 1768, pp. 61–75, 2000.

[32] J. Puzicha, T. Hofmann, and J. Buhmann, "Non- parametric similarity measures for unsupervised texture segmen- tation and image retrieval.," *IEEE Computer Society Conference on Computer Vision and Patter Recognition (CVPR)*, pp. 267–272, 1997.

[33] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, pp. 472–480, 2004.

[34] T. Bianchi, A. De Rosa, and A. Piva, "Improved dct coefficient analysis for forgery localization in jpeg images," in *ICASSP*, 2011, pp. 2444–2447.

[35] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, June 2012.

[36] Ingemar J. Cox, *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, 2008.

[37] M. Barni and F. Bartolini, *Watermarking systems engineering - Enabling digital assets security and other applications*, CRC Press, 2004.

[38] G.W. Corder and D.I. Foreman, *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*, Wiley, 2009.

[39] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill International Edition, 1991.

[40] Pierre Moulin and M. Kvanç Mihçak, "A framework for evaluating the data-hiding capacity of image sources," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1029–1042, 2002.

[41] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, feb. 2005.

[42] B. Mahdian and S. Saic, "Blind authentication using periodic properties of interpolation," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 529–538, sept. 2008.

[43] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear prediction residue," in *10th ACM Multimedia and Security Workshop (MM&Sec '08)*, 2008, pp. 11–20.

[44] B. Mahdian and S. Saic, "A cyclostationarity analysis applied to image forensics," in *2009 Workshop on Applications of Computer Vision (WACV)*, dec. 2009, pp. 1–6.

[45] D. Vázquez-Padín, C. Mosquera, and F. Pérez-González, "Two-dimensional statistical test for the presence of almost cyclostationarity on images," in *2010 17th IEEE Int. Conference on Image Processing (ICIP)*, sept. 2010, pp. 1745–1748.

[46] N. Dalgaard, C. Mosquera, and F. Pérez-González, "On the role of differentiation for resampling detection," in *2010 17th IEEE Int. Conference on Image Processing (ICIP)*, sept. 2010, pp. 1753–1756.

[47] D. Vázquez-Padín and F. Pérez-González, "Prefilter design for forensic resampling estimation," in *2011 IEEE Int. Workshop on Information Forensics and Security (WIFS)*, dec. 2011, pp. 1–6.

[48] "Audio database," `http://opihi.cs.uvic.ca/sound/genres.tar.gz`.

[49] Anil K. Jain, *Fundamentals of digital image processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[50] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[51] M. Chen, J. J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008.

[52] G. Valenzise, S. Magni, M. Tagliasacchi, and S. Tubaro, "No-reference pixel video quality monitoring of channel-induced distortion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 605–618, 2012.

[53] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 932–946, 2009.

[54] M.R. Banham and A.K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Magazine*, vol. 14, no. 2, pp. 24–41, 1997.

[55] Z. Fan and R. L. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.

[56] J. Lukás and J. Fridrich, "Estimation of primary quantization matrix in double compressed JPEG images," in *Proc. of DFRWS*, 2003.

[57] Y. Chen, K. S. Challapali, and M. Balakrishnan, "Extracting coding parameters from pre-coded MPEG-2 video," in *IEEE International Conference on Image Processing (ICIP)*, 1998, pp. 360–364.

[58] H. Li and S. Forchhammer, "MPEG2 video parameter and no reference PSNR estimation," in *Picture Coding Symposium (PCS)*, 2009, pp. 1–4.

[59] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2008, vol. 6819.

[60] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proceedings of the 11th ACM workshop on Multimedia and security*, New York, NY, USA, 2009, MM&Sec, pp. 39–48, ACM.

[61] M. Tagliasacchi and S. Tubaro, "Blind estimation of the QP parameter in H.264/AVC decoded video," in *International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 2010*, 2010, pp. 1–4.

[62] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, "Video codec identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 2257–2260.

[63] R. M. Gray and D.L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, oct 1998.

[64] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice coes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, june 2002.

[65] D. Wübben, D. Seethaler, J. Jaldn, and G. Matz, "Lattice reduction: A survey with applications in wireless communications," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 70–91, May 2011.

[66] A. K. Lenstra, H. W. Lenstra, and L. Lovsz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, 1982, 10.1007/BF01457454.

[67] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*, Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[68] H. Cohen, *A Course in Computational Algebraic Number Theory*, vol. 138 of *Graduate Texts in Mathematics*, Springer, 1993.

[69] B. Gruber, "Alternative formulae for the number of sublattices," *Acta Crystallographica Section A*, vol. 53, pp. 807–808, 1997.

[70] T. Blu, P. Thévenaz, and M. Unser, "MOMS: Maximal-order interpolation of minimal support," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1069–1080, July 2001.

[71] P.L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," vol. 55, no. 5, pp. 1741–1757, May 2007.

[72] T. Thongkamwitoon, H. Muammar, and P.L. Dragotti, "Identification of image acquisition chains using a dictionary of edge profiles," pp. 1757–1761, 2012.

[73] M. Unser, A. Aldroubi, and M. Eden, "B-Spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, February 1993.

[74] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B 39*, pp. 1–38, 1977.

[75] Alin C. Popescu and Hany Farid, "Exposing digital forgeries in color filter array interpolated image," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–1959, 2005.

[76] S.G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674 –693, July 1989.

[77] G.K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, feb 1992.

[78] D.S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards, and Practice*, Springer-Verlag, 2002.

[79] D.T. Lee, "JPEG 2000: Retrospective and new developments," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 32–41, jan 2005.

[80] G.J. Sullivan and T. Wiegand, "Video compression - from concepts to the H.264/AVC standard," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 18–31, jan 2005.

[81] M. Winken, P. Helle, D. Marpe, H. Schwarz, and T. Wiegand, "Transform coding in the HEVC test model," in *IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 3693–3696.

[82] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[83] H.S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, July 2003.

[84] W. Chen, C. Smith, and S. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Transactions on Communications*, vol. 25, no. 9, pp. 1004–1009, Sept. 1977.

[85] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 545–554, Apr. 2012.

[86] Ashwin Swaminathan, Min Wu, and K. J. Ray Liu, "Nonintrusive component forensics of visual sensors using output images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, March 2007.