Universidade Vigo

ESCOLA DE ENXEÑARÍA DE TELECOMUNICACIÓN

Ph.D. programme in Signal Theory and Communications

Ph.D. Thesis
Submitted for International Doctor Mention

DETECTION OF IMAGE RESAMPLING AND VIDEO ENCODING FOOTPRINTS FOR FORENSIC APPLICATIONS

Author: David Vázquez-Padín
Advisor: Fernando Pérez-González

2015

This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF) under projects TACTICA, COMPASS (TEC2013-47020-C2-1-R) and COMONSENS (TEC2015-69648-REDC), by the Galician Regional Government and ERDF under projects "Consolidation of Research Units" (GRC2013/009), REdTEIC (R2014/037) and AtlantTIC, and by the EU 7th Framework Programme under project NIFTy (HOME/2012/ISEC/AG/INT/4000003892).

FEDER - FONDO EUROPEO DE DESENVOLVEMENTO REXIONAL
"Unha maneira de facer Europa"

Abstract

Multimedia contents play an important role in our society. They serve as a means of communication and can be used not only as an entertainment, but also to inform or even to disseminate knowledge. The increasing relevance of multimedia contents, such as digital images, audio, or video sequences, has been tied to the development of editing software tools enabling their adjustment and enhancement, but ultimately allowing an unskilled person to easily manipulate them. As a consequence, their credibility as a source of information has been questioned and an important concern has arisen regarding their authenticity.

With the aim of recovering trust on multimedia objects, this thesis presents new techniques to detect and localize forgeries, but likewise to infer information about the processing history undergone by a multimedia content. The design of the proposed approaches is based on the theoretical analysis of characteristic traces or footprints that emerge from the application of certain processing to multimedia contents. In this thesis the derived research work encompassing multimedia forensics is divided in two parts.

The first part tackles the study of the resampling operation applied when a geometric transformation is performed to adapt a forged content to a genuine scene. The modeling of the resampling operation is addressed from different perspectives, establishing connections between this problem and other similar ones arising in distinct fields, and finally taking advantage of concepts from cyclostationarity theory, set-membership theory, or linear algebra, among others. We design different strategies for resampling factor estimation to characterize the particular transformation applied, providing estimates of the scaling factor or the rotation angle. The case of resampling detection is also considered to unveil the presence of resampling traces.

The second part of the thesis is focused on the forensic analysis of video compressed sequences. We start exposing the presence of a new footprint stemming from the double compression of video streams. By exploiting this feature, the detection of double encoding and the estimation of part of the processing history of a double compressed video are further investigated. Then, being capable of extracting information from the first compression, we move to the localization of intra-frame forgeries by applying a subsequent double quantization analysis.

Agradecementos

A elaboración desta tese supuxo un importante desafío para min, o cal me permitiu sobre todo ampliar os meus coñecementos, visitar lugares aos que probablemente nunca viaxaría e coñecer xente de diferentes países, ofrecéndome a posibilidade de entrar en contacto con distintas culturas. Isto lévame a pensar no moito que lle debo agradecer a toda a xente que me rodeou nesta etapa e que contribuíu dun xeito ou outro á consecución desta meta que tanto significa para min. Non quixera esquecerme de ninguén, mais se fose o caso pido perdón.

A primeira persoa á que lle quero agradecer profundamente a confianza depositada en min é ao meu director de tese, Fernando, quen sempre mantivo unha actitude positiva comigo, dándome valiosos consellos, poñendo todo en perspectiva e animándome continuamente a que seguise adiante. Sen a súa axuda, nunca tería chegado ata este punto e polo tanto nunca esquecerei o afortunado que fun ao pensar naquel profesor que tan ben explicaba nas clases de FCD e TDIX á hora de buscar con quen facer o proxecto fin de carreira. Aí comezou todo, gracias!

Quen participou de forma moi activa na investigación levada a cabo nesta tese foi Pedro, a quen lle agradezo moito a súa dedicación e axuda, aportando sempre ideas novas e solucionando problemas cos que eu podía levar días pelexándome. Quero agradecerlle a Carlos que supervisase os meus primeiros pasos neste mundo da investigación, indicándome por onde seguir e sempre preguntándome como ía evolucionando todo. Tamén quero darlle as gracias a Roberto e a Nuria por estar sempre dispostos a axudar e, especialmente a Carmen Touriño pola súa impecable xestión dos temas administrativos e por ser tan boa persoa.

A estancia en Florencia deixou unha importante pegada na miña vida, non só por todo o que aprendín, senón tamén pola xente que coñecín. Por iso, quero agradecerlle a Alessandro que me aceptase no seu grupo de investigación como se fose un máis dende o primeiro día e a Mauro por estar sempre pendente do noso traballo. A Tiziano e a Alessia, quero darlles as gracias por compartir comigo os momentos de cantina, explicándome cada prato que degustabamos. En especial, quero agradecerlle a Marco que me recibise como un membro máis da súa familia e que fose tan bo compañeiro á hora de traballar de forma conxunta. Espero que a gran amizade que construímos alí xunto con Teresa e fortalecemos logo en Vigo, siga adiante por moitos anos.

Quero darlle as gracias a todos os meus compañeiros de traballo do TSC-5, con quen as xornadas laborais se fan moi levadías. Gabi, empezaches sendo o meu titor de PFC, pero agora es un gran amigo con quen comparto moitas cousas persoais e por iso che estou moi agradecido. Juan, gracias tamén por ser tan bo amigo e estar sempre disposto a botar unha man tanto en asuntos técnicos como noutros máis persoais. Miguel, gracias por conseguir sacarme sempre unha gargallada, es desas persoas que transmite felicidade. Simón, gracias por compartir a túa visión da vida un pouco máis aloucada que a miña, xa que me axuda a entender moitas cuestións a priori descoñecidas. Pedrouzo, admiro e anhelo a túa serenidade, gracias por preocuparte sempre por min. Tamén quería agradecerlle a Iria, recente ausencia no TSC-5, que sementase tanta alegría e contribuíse a crear un tan bo ambiente de traballo no laboratorio. Gracias a todos os compañeiros que fixeron estadías connosco como Nahuel, Elices, Valentina, Michael, Cecilia, Serena, Omar, Reza, Benedetta, Martijn, Aurélien e Matthieu, de quen sempre aprendín algo novo.

Non podo esquecerme de antigos membros do noso grupo como Luis con quen sempre foi un pracer traballar e a quen agradezo que me ofrecese en diversas ocasións a posibilidade de colaborar en novos proxectos. Tamén quero darlle as gracias a Dani por recibirme sempre dun xeito tan afable, preocupándose constantemente por min e animándome en momentos de fraqueza. Mención especial a Gonzalo por compartir comigo boa parte do doutoramento no TSC-5 e tamén no piso xunto con César, gracias a ambos. Finalmente, quero agradecer ao resto de compañeiros que tivemos no TSC-5 como Abu, Campi, Eli, Mela, Paula, Montse e Magui, e tamén a Marta, Rocío e Roberto, con quen é un pracer compartir ceas.

Agradézolles a todos os meus amigos de Cambados: Carolina, Rubén, Fernando, Christian, Cristina, Jon, Gemma, Lito e Mónica, a súa proximidade e constante interese polo meu traballo e a situación da tese. Gracias a Jara e Juanjo por axudar a evadirme experimentando novas aventuras de escapismo por Vigo. Gracias tamén a Juanín por chamarme acotío e preguntarme que tal me vai todo.

En canto á miña familia, a elaboración desta tese non tería sido posible sen o apoio e a axuda dos meus pais, aos que lles teño que agradecer infinitamente que impregnasen en min os valores da perseveranza e do esforzo. Tamén meu irmán contribuíu en gran medida ao desenvolvemento desta tese, dándome consellos importantes e fomentando en min esa curiosidade por tratar de entender como e por que funciona todo o que nos rodea. Ao resto da familia: avós, padriños, tíos e primos, tamén lles quero mostrar o meu agradecemento por estar continuamente pendentes de como me van as cousas por Vigo.

Por último, teño que agradecer especialmente á miña moza Inés por confiar en min e apoiarme en todas as decisións que tomei, pero sobre todo por aturarme nos peores momentos de pesimismo e sempre conseguir animarme. Gracias tamén aos teus pais, á túa avoa e ás túas irmás por preocuparse tanto por min. Inés, por todo o que tiveches que aguantar, este traballo vai enteiramente dedicado a ti.

Contents

Acronyms and Abbreviations

1-D	One-dimensional
2-D	Two-dimensional
AC	Alternate Current
AR	AutoRegressive
AR(1)	1st-order AutoRegressive
AUC	Area Under Curve
CBR	Constant BitRate
CFA	Color Filter Array
CIF	Common Intermediate Format
DC	Direct Current
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DQ	Double Quantization
DTFS	Discrete-Time Fourier Series
DVD	Digital Versatile Disc
EM	Expectation/Maximization
FAR	False Alarm Rate
FIR	Finite Impulse Response
fps	frames per second
GCD	Greatest Common Divisor
GOP	Group Of Pictures
i.i.d.	independent and identically distributed
I-MB	Intra-coded MacroBlocks
JPEG	Joint Photographic Experts Group
kbps	kilo bits per second
lcm	least common multiple
MB	MacroBlock
M-JPEG	Motion JPEG
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate

MPEG	Moving Picture Experts Group
MSE	Mean Squared Error
OVE	Optimal Volume Ellipsoid
pdf	probability density function
p-map	probability map
P-MB	Predictive-coded MacroBlocks
RANSAC	RAndom SAmple Consensus
RGB	Red Green Blue
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transform
S-MB	Skipped MacroBlocks
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TIFF	Tagged Image File Format
VBR	Variable BitRate
VPF	Variation of Prediction Footprint
WLS	Weighted Least Squares

Notation

The following notational conventions will be used along the chapters of this thesis, unless otherwise stated: calligraphic letters are only used for denoting sets, e.g., $\mathcal{X}$. Common number sets, such as real numbers set or integer numbers set, are represented with double line notation, i.e., $\mathbb{R}$ and $\mathbb{Z}$, respectively.

Lowercase or uppercase letters refer to scalar variables, e.g., x or X. Boldface letters are used for representing vectors and matrices. A column vector $\mathbf{x}$ consists of N_x elements x_i, where $i \in \{0, \ldots, N_x - 1\}$, thus having $\mathbf{x} = (x_0, \ldots, x_{N_x-1})^T$. Notice that $(\cdot)^T$ stands for transposition and, similarly, when complex numbers are used $(\cdot)^H$ stands for transposition and conjugation. An $N_1 \times N_2$ matrix $\mathbf{X}$ has $N_1 N_2$ elements $X_{i,j}$ (which occasionally can be denoted by $X(i,j)$), where each index (i,j) represents an element at i-th row and j-th column with $i \in \{0, \ldots, N_1 - 1\}$ and $j \in \{0, \ldots, N_2 - 1\}$.

A time-dependent 1-D signal is denoted by $x(n)$, with n representing indistinctly a continuous index $n \in \mathbb{R}$ or a discrete-index $n \in \mathbb{Z}$. Likewise, a 2-D field is denoted by $x(\mathbf{m}) \triangleq x(m_1, m_2)$ with $\mathbf{m} \triangleq (m_1, m_2)$ (notice that this representation does not follow the above convention to represent vectors, but we only use this particular notation for denoting 2-D vectors). A time-dependent 2-D field representing, for instance, a collection of frames in a video sequence, is denoted as follows: $\underline{\mathbf{x}}(n)$, where n stands for the time index.

When dealing with stochastic processes, the mean of a process $x(n)$ is represented by $\mu_x(n) \triangleq \mathrm{E}\{x(n)\}$ and the covariance by $c_{xx}(n;\tau) \triangleq \mathrm{E}\{[x(n) - \mu_x(n))(x(n+\tau) - \mu_x(n+\tau)]\}$. We denote the cyclic correlation of a zero-mean process by $C_{xx}(\alpha;\tau)$ and the Fourier Series coefficients having period Q are denoted by $C_{xx}\left(\frac{2\pi}{Q}k;\tau\right)$, or directly by $C_{xx}(k;\tau)$, with $k \in \{0, \ldots, Q-1\}$. The Fourier Series coefficients of a process $x(n)$ are denoted by $X(k)$.

Random vectors are represented with italic bold capital letters (e.g., $\boldsymbol{X}$), their outcomes with lowercase letters (e.g., $\boldsymbol{x}$). A vector of length N starting from the n-th component, is denoted by $\boldsymbol{x}_n = (x_n, \ldots, x_{n+N-1})^T$. For a compact notation, we use $\mathrm{mod}(a,b)$ to denote the modulo operation: $a \mod b$. Floor and ceiling functions are represented by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$, respectively. On the other hand, $\lfloor \cdot \rceil$ represents the rounding function of a number to the nearest integer.

Chapter 1

Introduction

Multimedia contents—such as digital images, audio, or video—have become the most extensively used vehicle for communication during last years. The massive proliferation of these digital contents over the Internet, across the media, or through social networks has converted them into a valuable asset. As an example, with the current increase of Internet usage from mobile devices, any captured moment from an unexpected event may get the power of instantly distribute breaking information by simply sharing it in a social network.

Meanwhile, the rapid growth of editing tools that were originally devised to enhance the quality of those captured moments, enable now an unskilled person to easily manipulate them and, eventually, to create realistic synthetic contents. This state of affairs has boosted an important concern about the authenticity of multimedia objects. Moreover, due to the relative simplicity of tampering with digital images, audio, and videos, the work of the forensic investigator as a specialist in digital imagery becomes particularly relevant when a multimedia object is used as a proof of facts in a legal proceeding. In such case, it is imperative to know the origin of the multimedia object and also to trace back the processing history of its content, in order to justify whether the digital object can be admitted as a legal evidence or not.

As a means to rebuild trust in multimedia objects, a lot of techniques have arisen in the past few years to prove the authenticity or verify the integrity of multimedia contents, coping also with plausible manipulations. These techniques are commonly labeled as active or passive depending on the generation process and the role of the forensic analyst. On the one hand, active approaches require a known signal (e.g., a digital watermark) that is imperceptibly embedded in the digital content to detect forgeries later on. On the other hand, passive approaches work in the absence of any known signal and rely on the analysis of traces left by the capturing device during the acquisition process or any other operation applied after its creation, such as compression and/or edition. Given the need of special-

purpose hardware/software in the former case versus the universal applicability of the derived methods in the latter, much effort has been lately put into passive multimedia forensics. Currently, the analysis of these traces, also known as digital footprints, has been broadly investigated for images [1] and increasing attention is given to audio [2] and video [3].

This thesis is focused on the analysis of particular digital footprints left in multimedia contents after their processing. Our main goal is to detect and localize forgeries, but likewise to infer information about the processing history undergone by a multimedia content in a blind fashion. In the first part of the thesis, we theoretically model the resampling traces left by the application of geometric transformations to images and audio signals, while, in the second part, we reveal and further exploit a footprint that arises from the double compression of video sequences.

1.1. Motivation

Nowadays, it is rather simple to alter the information represented by a multimedia content without leaving obvious signs of manipulation. As a consequence, a forensic analyst has to deal regularly with situations where a multimedia object cannot be deemed as an undeniable proof of occurrence of a fact. For instance, in July 2010, while the British Petroleum (BP) company was struggling against the Gulf Coast oil spill, a doctored version of their command center shown in Figure 1.1(a) was published on their website by filling the blank screens with other parts of the original photo yielding the final result in Figure 1.1(b). Even if there were probably no bad intentions in retouching the genuine image, it seems that the original content of the command center could wrongly shape the public opinion of the company.

Regardless of the final intention, this kind of manipulations hampers the trustworthiness of digital images, and as it can be checked in [4], this is only one of many cases throughout History. Therefore, it is evident that there is an urgent need to develop methods and automatic tools for assuring the authenticity of multimedia contents. Furthermore, since active forensics cannot handle the analysis of arbitrary contents of unknown provenance, the need for passive forensic techniques becomes apparent.

In the context of passive forensic techniques, there does not exist a common framework to analyze multimedia contents and detect forgeries, i.e., there is not a universal tool that can explicitly determine all the modifications or transformations applied to a digital content. Instead, there is a collection of tools that exploit some of the inherent characteristics of a particular digital object (i.e., images, audio, or videos), and in doing so, try to detect the alterations such

(a) Original image (b) Tampered image

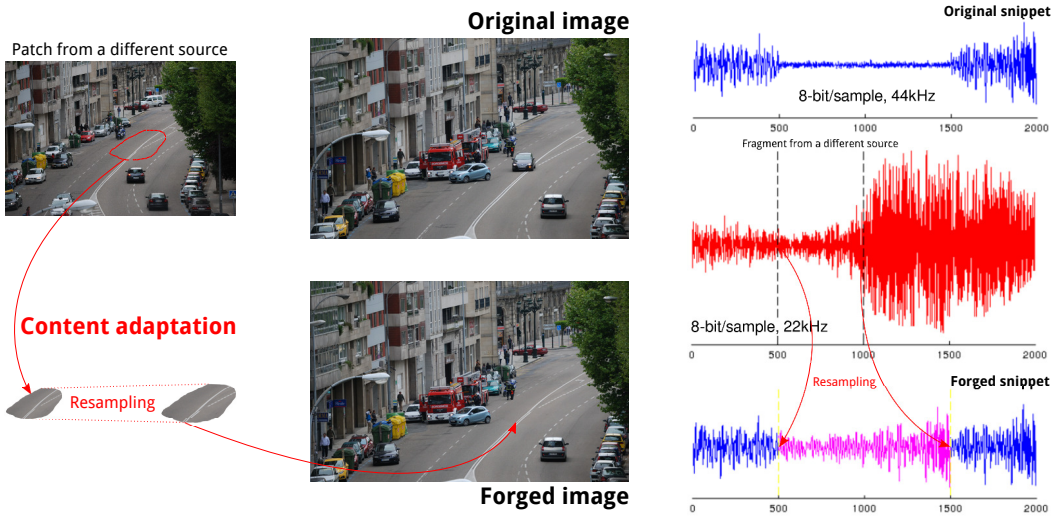
Figure 1.1: Real example of a tampered image (on the right) shown in the BP website by copying and moving parts of the original image (on the left). Courtesy of The Washington Post, July 2010.

content has been subject to. Available techniques achieve promising results but, occasionally, the lack of a theoretical model behind suggests that there is still room for improvement. Meanwhile, a lot of effort has been put into the analysis of digital images, paying less attention to video sequences, thus motivating the search for new footprints in this domain.

In this thesis we use principles of signal processing to theoretically describe digital footprints with the aim of furnishing information about the authenticity, integrity or processing history of multimedia contents. We mainly focus our analysis on images and videos, though several experiments are performed with audio signals to keep some of the proposed techniques computationally tractable. In the first part of the thesis, we deepen the understanding of the resampling traces left in a digital image (or in an audio signal, thereof) after the application of a geometric transformation. Since similar problems arise in other fields, such as Digital Communications or Automatic Control, we establish links with each field, taking advantage of concepts from cyclostationarity theory and set-membership theory, among others. In the second part of the thesis, we explore a new footprint emerging from the double compression of video sequences which allows us to infer parameters from the first compression, but also the detection of double compression and the localization of forgeries in video sequences.

1.2. Forensic Analysis of Resampled Signals

When a credible forgery is carried out, most of the time it is necessary to adapt added pieces to the original content. Such adaptation may require the use of geometric transformations that involve the use of a resampling operation which inherently leaves characteristic traces that are not typically present in a genuine



(a) Image (b) Audio

Figure 1.2: Illustrative example of how to create a forgery. In both cases, a portion from a different source is first extracted and then geometrically adapted prior to being pasted in the original content.

content. Forensic analysis of resampled signals is consequently of particular interest, since the detection of traces stemming from the resampling operation can be used as a means to unveil forgeries or to infer the processing history of a content under analysis.

1.2.1. Introduction

A digital image forgery can be accomplished throughout many different ways, but it usually involves copying a region either from the image itself or from a different one and pasting it in the original scene to add a new feature or to conceal an existing one. The adjustment of new contents to a particular scene is frequently carried out by applying geometrical transformations (e.g., scaling, rotation, or skewing), as it can be checked in the illustrative example of Figure 1.2(a). In a similar way, when two audio signals with different sampling rates are mixed, then at least the sampling rate of one of them must be adjusted in order to avoid audible distortions, as exemplified in Figure 1.2(b).

The spatial transformation of a genuine image, or a region therein, maps the intensity values at each pixel location of the original grid to a new resampled grid. This operation must be followed by the interpolation of the pixel intensity values in the intermediate locations between source pixels, which is performed through a weighted linear combination of adjacent pixels. In the case of audio signals, the same procedure is followed but across a single dimension.

These linear dependencies among neighboring samples are therefore the characteristic traces left behind by the interpolation process. Interestingly, these local dependencies rarely show up in genuine contents and they vary periodically along the resampled region, thus enabling their detection and the possible identification of the applied transformation by inferring the repetition period. This results in two different ways of tackling the forensic analysis of resampled signals: by means of resampling detection and through the estimation of the applied resampling factor. The former studies the presence or absence of resampling traces in the observed data, so that the designed detector solves the following binary hypothesis problem:

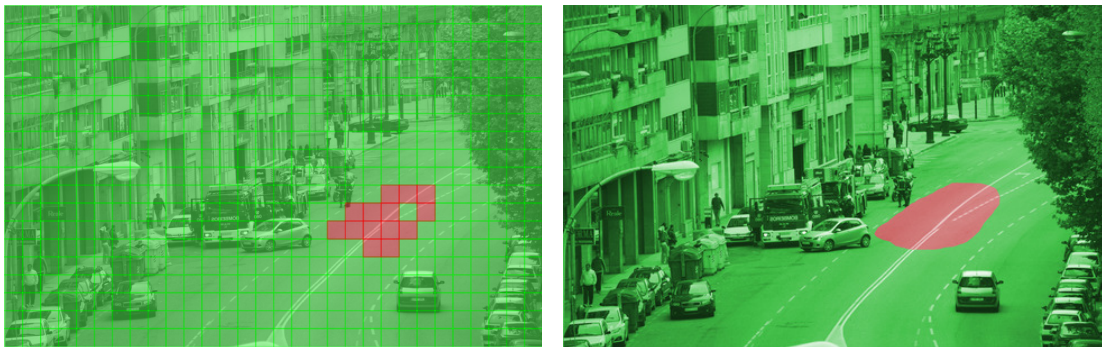
$$\mathcal{H}_0\text{: the observed data has not been resampled,}$$
$$\mathcal{H}_1\text{: the observed data has been resampled.}$$

On the other hand, when performing resampling factor estimation, the specific evolution of these resampling traces throughout the observed data is examined and an estimate of the resampling factor used in the applied spatial transformation is provided.

Although some similarities between resampling detection and estimation can be outlined, we emphasize the main difference between both approaches: resampling detection leads to a binary classification problem were the outcome is either "right" or "wrong", while resampling estimation generally does not provide an exact outcome, but an approximated value to the true resampling factor. This difference affects the manner in which the performance of each approach is evaluated. Moreover, in the last case, the particular estimation enables the identification of the applied geometric transformation, thus yielding a more accurate forensic analysis.

Finally, by performing either resampling detection or estimation, a possible form to detect forgeries lies in the analysis of inconsistencies in the resampling traces of small portions with respect to the whole content under analysis. As an example, Figure 1.3(a) shows the result of applying in a block-by-block fashion the resampling detector proposed in [5] to the forged image depicted in Figure 1.2(a). For the sake of comparison, the ground truth mask of the manipulated area is illustrated in Figure 1.3(b).

In the following, we start formulating the resampling operation for digital images in mathematical terms, then the related works on the forensic analysis of resampled signals are described, and finally, our contributions to this problem are summarized.



(a) Detected forgeries (b) Ground truth

Figure 1.3: Illustrative example of forgery detection from the composite image in Figure 1.2(a). On the left, the result of applying the resampling detector in [5] to each block of size 128×128 is depicted. On the right, the ground truth mask of the tampering is shown. Green color is used for representing non-resampled regions, while red color stands for resampled areas.

1.2.2. Resampling Process Description

Let us define a digital image with a single color channel as a $P \times Q$ matrix $\mathbf{F}$ with elements $F_{p,q}$ and indices $p \in \{0, \dots, P-1\}$ and $q \in \{0, \dots, Q-1\}$. The values of each element $F_{p,q}$ are discrete quantities whose range is determined according to the image bit depth. In practice, most digital images use 8 bits of intensity resolution per color channel, however we notice that in general $F_{p,q} \in \{0, \dots, 2^b - 1\}$, where b represents the bit depth.

The resampling operation is assumed to be linear, so each pixel value in the resampled image $\mathbf{G}$, i.e., $G_{i,j}$, is computed by linearly combining a finite set of neighboring samples coming from the original image. The process of resampling involves two main steps: the definition of the resampling grid with the new pixel locations and the computation of the intensity values in those new locations.

Regarding the first step, the mapping between the source coordinates with indices (p, q) and the resampled ones (i, j) can be expressed through an affine transformation as follows

$$\begin{pmatrix} i \\ j \end{pmatrix} = \mathbf{A} \begin{pmatrix} p \\ q \end{pmatrix} + \mathbf{b}, \tag{1.1}$$

where $\mathbf{A}$ is a matrix that embodies the linear transformation (e.g., scaling, rotation, etc.) and $\mathbf{b}$ represents the translation vector. As an example, a rotation by an angle θ counterclockwise can be written in the following matrix form

$$\mathbf{A} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{1.2}$$

Table 1.1: Impulse response and width of several interpolation kernels.

Kernel type	**Impulse response**	**Width**
Linear	$h(t) = \begin{cases} 1-\lvert t\rvert, & \text{if } \lvert t\rvert \leq \frac{k_w}{2} \\ 0, & \text{otherwise} \end{cases}$	$k_w = 2$
Catmull-Rom	$h(t) = \begin{cases} 3/2\lvert t\rvert^3 - 5/2\lvert t\rvert^2 + 1, & \text{if } \lvert t\rvert \leq \frac{k_w}{4} \\ -1/2\lvert t\rvert^3 + 5/2\lvert t\rvert^2 - 4\lvert t\rvert + 2, & \text{if } \frac{k_w}{4} < \lvert t\rvert \leq \frac{k_w}{2} \\ 0, & \text{otherwise} \end{cases}$	$k_w = 4$
B-spline	$h(t) = \begin{cases} 1/2\lvert t\rvert^3 - \lvert t\rvert^2 + 2/3, & \text{if } \lvert t\rvert \leq \frac{k_w}{4} \\ -1/6\lvert t\rvert^3 + \lvert t\rvert^2 - 2\lvert t\rvert + 4/3, & \text{if } \frac{k_w}{4} < \lvert t\rvert \leq \frac{k_w}{2} \\ 0, & \text{otherwise} \end{cases}$	$k_w = 4$
Lanczos	$h(t) = \begin{cases} \text{sinc}(t)\text{sinc}(t/3), & \text{if } \lvert t\rvert < \frac{k_w}{2} \\ 0, & \text{otherwise} \end{cases}$	$k_w = 6$

In addition, a homogeneous translation through $\mathbf{b} \triangleq (\delta, \delta)^T$ is generally applied, such that the sampling points of the resampled image are centered with respect to the grid of the original image. For the sake of brevity, in the following we will consider that the resampling operation uniformly scales each dimension of the original image by a resampling factor ξ yielding

$$\mathbf{A} = \begin{pmatrix} \xi & 0 \\ 0 & \xi \end{pmatrix}, \tag{1.3}$$

where ξ is defined as the ratio between the upsampling factor $L \in \mathbb{N}^+$ and the downsampling factor $M \in \mathbb{N}^+$, i.e., $\xi \triangleq \frac{L}{M}$ with L and M relatively prime.

The second step in the resampling process can be performed using different interpolation kernels to compute the intensity values in the new resampled grid. As previously stated, we only take into account linear interpolation strategies and, specifically, we restrict ourselves to the following types of two-dimensional separable kernels:[1] bilinear, cubic and Lanczos (i.e., truncated sinc). From the family of cubic filters described in [6] and parameterized by the pair of values (B, C), we select two well-known filters: the Catmull-Rom spline with parameters $(B, C) = (0, 0.5)$, and the cubic B-spline with $(B, C) = (1, 0)$. As Lanczos kernel, we decide to take a three-lobed Lanczos-windowed kernel following the definition in [7]. We take into consideration these interpolation kernels because they are the most commonly available in software editing tools. Note that we discard the analysis of more complex interpolation algorithms, such as adaptive or non-linear, given that their use is typically constrained to perform demosaicing and are rarely employed to resize images. Table 1.1 gathers the one-dimensional impulse response $h(t)$ with $t \in \mathbb{R}$ together with the width of each considered kernel.

[1] Two-dimensional separable kernels are those that can be applied as a product of two one-dimensional functions, evaluating each function across a single dimension.

By combining the two detailed steps in a single expression, each pixel value $G_{i,j}$ of the resampled image can be obtained as follows

$$G_{i,j} = \sum_{k=0}^{P-1} \sum_{l=0}^{Q-1} h\left(i\frac{M}{L} + \delta - k\right) h\left(j\frac{M}{L} + \delta - l\right) F_{k,l}, \tag{1.4}$$

where δ denotes the introduced shift between the two sampling grids[2] and $h(\cdot)$ represents any of the one-dimensional interpolation kernels described in Table 1.1. Given that the original image is defined at coordinates $p \in \{0, \ldots, P-1\}$ and $q \in \{0, \ldots, Q-1\}$, the resulting resampled image will take values on $i \in \{0, \ldots, (L/M)P - 1\}$ and $j \in \{0, \ldots, (L/M)Q - 1\}$.[3]

Notice that after computing all the pixels of the resampled image, its intensity values should fit the original resolution or bit depth of the input image. Therefore, as a last step, the resampled values must be quantized to the original precision, having

$$R_{i,j} = Q_\Delta\left(G_{i,j}\right),$$

where $R_{i,j}$ denotes each element of the quantized resampled image $\mathbf{R}$ and $Q_\Delta(\cdot)$ represents a uniform scalar quantizer with step size Δ.

So far, the detailed resampling operation can be applied for any resampling factor $\xi > 0$; however, if the same process is followed for resampling factors less than one, i.e., $\xi < 1$, then visual distortions might appear due to aliasing. To circumvent this distortion problem, an anti-aliasing filter must be applied prior to the resampling process to suppress the higher frequencies that may produce aliasing. Given that the anti-aliasing filter is a low-pass filter as the interpolation kernel, the typical way to implement a resampling operation avoiding aliasing is by combining the impulse response of both filters, yielding a wider version of the original kernel, i.e.,

$$h_a(t) \triangleq \tfrac{L}{M}\, h\left(\tfrac{L}{M}t\right).$$

Therefore, when $\xi < 1$, the resampled pixels are computed as in (1.4), but using the anti-aliasing version of the kernel $h_a(t)$ instead of the original $h(t)$. Note that the original length of the kernel also widens by $\xi^{-1} = \frac{M}{L}$ in such a way that the final width of $h_a(t)$ becomes $k_{w_a} \triangleq k_w \frac{M}{L}$.

1.2.3. Prior Work

The problem of resampling detection as a means to unveil forgeries has been largely investigated in recent years. Even though the resampling process can be

[2] In MATLAB's function `imresize` and also in the tool `convert` from ImageMagick's software, the shift corresponds to $\delta \triangleq \frac{1}{2}\left(1 + \frac{M}{L}\right)$.

[3] For the sake of simplicity and without loss of generality, we assume that P and Q are both multiples of M.

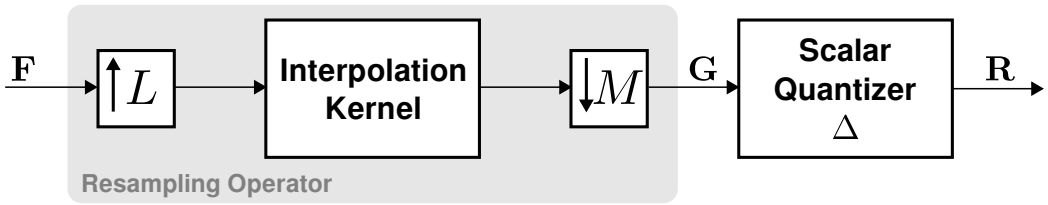


Figure 1.4: Block flow diagram of the resampling process.

modeled by a relatively simple processing chain as illustrated in Figure 1.4, many different directions have been explored to infer the presence of resampling traces from the observation of the resulting output of the processing chain.

Popescu and Farid's seminal work [8], was the first to uncover the existence of periodic correlations in **G** induced by the resampling process. By relying on a linear predictor that models the relation between each sample and its neighbors, they proposed a method to detect and quantify these periodic correlations. In particular, given any vector **z** from the quantized resampled image **R** in Figure 1.4, containing a set of $2N + 1$ adjacent samples, they use the Expectation/Maximization (EM) algorithm to estimate the predictor coefficients $\boldsymbol{\alpha}$ (with $\alpha_0 = 0$) that satisfy

$$z_i - \sum_{k=-N}^{N} \alpha_k z_{i+k} = 0.$$

Once each sample of the image under analysis has been processed through the proposed EM algorithm, a probability map (p-map) is generated comprising the probability of each sample being correlated to its neighbors. In the presence of interpolation, this p-map exhibits periodic patterns that can be captured in the frequency domain. After the generation of synthetic maps for a set of possible spatial transformations, the detector's decision is based on the similarity between the p-map of the image under analysis and each element of this set.

Despite the good results achieved by Popescu and Farid, one of the main difficulties of their approach was related to the correct initialization of some parameters for reaching the EM convergence. However, a few years later Kirchner suggested a simpler solution in [9], by focusing the analysis on the variance of the prediction residue. Following the same model above, he realized that the variance of the prediction error **e**, whose i-th sample can be computed as

$$e_i = z_i - \sum_{k=-N}^{N} \alpha_k z_{i+k},$$

also exhibits periodic artifacts. At the same time, he recognized that the formation of periodic artifacts does not depend on the actual prediction weights, thus proposing a simplified detector bypassing the EM estimation throughout the use of a prefilter with fixed symmetric coefficients (i.e., $\alpha_k = \alpha_{-k}$). Once computed

the new p-map, Kirchner discards the detection of resampling by means of the exhaustive search in [8]. Instead, the detection process is reduced to the calculation of the cumulative periodogram of the p-map under analysis, which will show a sharp transition in case of resampling. The final decision of the detector is based on the maximum absolute value of the gradient of the cumulative periodogram.

Almost in parallel with the seminal work by Popescu and Farid, Gallagher noticed in [10] that another type of prefilter yields detectable periodic artifacts in the variance of the filtered signal. In particular, he proved that the variance of the second order derivative of an interpolated signal (with i.i.d. samples coming from a Gaussian distribution) is periodic with a period equal to the resampling factor. Therefore, as a first step, the proposed method computes the second order derivative of each row from the resampled image $\mathbf{R}$. Then, the l_1-norm of each column from the resulting image is computed, generating the so-called pseudo-variance signal in [10]. As a last step, the Discrete Fourier Transform (DFT) of this variance signal is computed, ignoring the lowest frequencies of the spectrum. The detector finds resampling traces if it follows that a local peak in the magnitude of the DFT is T times greater than a local average.

Later on, Mahdian and Saic [11] extended this idea showing that under a stationary signal model, the variance of the n-th order derivative of a resampled signal is periodic with the resampling factor. Supporting this fact, Dalgaard et al. carefully analyzed in [12] the role of differentiation as a way of boosting resampling traces, showing that differentiators used as prefilters are nearly optimal. In line with this, Mahdian and Saic proposed a method maintaining the second order derivative filter from [10], but applying afterward a Radon transform to the magnitude of the filtered image. By doing this, the detection of more complex affine transformations becomes possible (a total of 180 different angles is taken into account). In this case, the search for periodicity is performed by computing the DFT of the autocovariance function of each Radon transform (previously filtered by a first order derivative filter). Finally, the proposed detector is driven by the same criterion than in [10].

All the techniques described so far work with a residue signal obtained either by a global predictor [8], a fixed linear filter [9], or a derivative filter [10, 11]. However, it was later noted by Kirchner in [13] that the specific structure of resampled images can be explicitly modeled by a series of linear predictors, whose estimated predictor coefficients describe the characteristic periodic correlations between neighboring pixels. The following model is assumed: each row/column from $\mathbf{R}$ can be written as the linear combination of their vertical/horizontal neighbors, i.e.,

$$\mathbf{r}^{(i)} = \left(\mathbf{r}^{(i-K)}, \ldots, \mathbf{r}^{(i-1)}, \mathbf{r}^{(i+1)}, \ldots, \mathbf{r}^{(i+K)} \right) \boldsymbol{\beta}^{(i)} + \boldsymbol{\epsilon}^{(i)},$$

where $\mathbf{r}^{(i)}$ denotes a column vector containing the i-th row/column of the quantized resampled image $\mathbf{R}$, $\boldsymbol{\beta}^{(i)}$ stands for the predictor coefficients, and $\boldsymbol{\epsilon}^{(i)}$ rep-

resents an error term. After using a Weighted Least Squares (WLS) procedure to estimate the coefficients $\hat{\boldsymbol{\beta}}^{(i)}$, Kirchner suggests that the differences

$$d_i = \hat{\beta}_{-1}^{(i)} - \hat{\beta}_1^{(i)}$$

are promising to detect traces of resampling. As a matter of fact, making use of a robust spectral method to reveal the periodicity in the differences d_i, the proposed detector shows very good performance especially for downscaled images.

Note that all the detailed schemes are designed to expose the presence of resampling traces, thus focusing solely on the problem of resampling detection. Although the foregoing works in [8, 9, 10, 11] provide some insights about how to estimate the resampling factor of an image, they do not evaluate the performance of the derived estimates. Following a more comprehensive analysis of the resampling estimation problem, interesting approaches have arisen in this area. For instance, different methods have been proposed for dealing with the estimation of the scaling factor ξ from (1.3) avoiding ambiguities [14, 15]. Other research works have been oriented towards the estimation of the rotation angle θ from (1.2) applied to an image, as in [16, 17]. Recently, a more general solution has been achieved in [18], where the estimation of the complete linear transformation $\mathbf{A}$ from (1.1) is performed.

In the literature, more techniques are available to expose forgeries by detecting inconsistencies in such characteristic resampling traces. We have deepened the description of the above methods mainly because they are considered as state-of-the-art techniques in resampling detection, but also because comparative results against some of them will be provided throughout this thesis. Nevertheless, interested readers may find appealing the following approaches: in [19], an example of how to use a resampling detector to unveil tampered regions is provided; in [20], the case of resampling detection in re-compressed JPEG images is investigated and further revisited in [21]; in [22], a first attempt to characterize linear dependencies through the Singular Value Decomposition (SVD) of a resampled image is proposed, resorting to a Support Vector Machine (SVM) classifier to detect resampling; finally, in [23], resampled images are detected by measuring the normalized energy density of different window sizes and feeding these values to an SVM classifier.

In this first part of the thesis we start by proposing new approaches that are also based on the frequency analysis of a residue signal, but establishing links with the cyclostationarity theory. The study of new prefilters and their design under the cyclostationarity framework has also been considered. However, given that the examination of the periodic correlations in the frequency domain presents some drawbacks such as the need for a large number of samples (to elude the windowing effect which impairs the estimator's performance), we later address the resampling estimation problem from a different perspective. In particular, we pay more attention to how the quantization applied as a last step in the diagram

of Figure 1.4 could help inferring parameters from the applied resampling operation. In this direction, we tackle the problem of the estimation of the resampling factor following the maximum likelihood criterion, from where we discover that resampling estimation can also be addressed in line with the set-membership theory. Ultimately, with the aim of characterizing the linear dependencies induced by the resampling operation, we exploit the capability of the SVD to perform resampling detection via subspace decomposition.

1.2.4. Contributions

The main contributions regarding the forensic analysis of resampled signals in this first part of the thesis can be summarized as below:

RC1. Derivation of a theoretical framework for the estimation of parameters from the applied spatial transformation to an image, establishing links between the resampling factor estimation problem and cyclostationarity theory. Within this framework, a method for estimating the actual parameters of spatially transformed images (i.e., scaling factor and rotation angle) has been derived. In addition, the design of prefilters to improve the estimation accuracy of the resampling factor has been analytically investigated.

RC2. Analysis of the resampling factor estimation following the maximum likelihood criterion. Even though the considered scenario is constrained to a piecewise linear interpolation, which unavoidably limits the scope of application of the derived estimator, important insights are provided on how to benefit from the scalar quantization applied after the resampling operation. The most distinctive contribution of the derived approach is that only a small number of samples of the resampled signal are needed to correctly estimate the employed resampling factor.

RC3. Identification of resampled signals in accordance with set-membership estimation theory. Using as starting point the foundations laid by the previous contribution, we adhere to set-membership theory to design a technique that is able to estimate the resampling factor of a one-dimensional signal. Interestingly, with this approach we can provide estimates whose singular characteristic is to be consistent with all information arising from the observed data and the a priori knowledge about the resampling process. This tool is powerful and it is often required by forensic examiners because they have to guarantee that the forensic techniques being used (in a legal proceeding, for example) are reliable, in such a way that innocent people will not be unfairly charged.

RC4. Analysis of resampling detection as a subspace decomposition problem. Delving into the linear dependencies induced by the interpolation process,

we show that upsampled images can be decomposed in two components: one of them is determined by the resampling process (in particular, by the interpolation kernel) thus belonging to a so-called signal subspace, while the other component arises from the scalar quantization applied after resampling, thus pertaining to a so-called noise subspace. From this analysis, we propose the use of the SVD for decomposing both subspaces and accordingly detect the upsampling operation.

RC5. Design and evaluation of a practical solution for exposing original an duplicated regions in a copy-move manipulation. We propose the combination of two existing methods: the first one, based on Scale Invariant Feature Transform (SIFT), is capable of finding duplicated regions; while the second one, based on a resampling estimator, allows one to identify which region is the source and which is the forged one. On account of the more comprehensive analysis that can be provided from a tampered image, this tool is valuable for a forensic analyst.

1.3. Forensic Analysis of Video Sequences

Recent advances in video compression have made possible the adoption of digital video technologies in many different fields, such as digital television broadcasting, videotelephony or Internet video streaming, among others. As happened first with digital images, today we can easily find powerful and accessible video editing software that facilitates the modification of video sequences. Consequently, in the last years, the creation of forensic tools that analyze the authenticity and integrity of digital videos has become an important field of research.

1.3.1. Introduction

Forensic analysis of video sequences, which is commonly referred to as video forensics, is an emerging discipline that strives to find information about the processing history undergone by a digital video. Since video processing is computationally more demanding than image processing, the research community started working on images first, having in mind the possible extension of the derived approaches to video streams. For instance, any working technique with JPEG compressed images could be straightforwardly adapted to the Motion JPEG (M-JPEG) video compression format.

This is one of the reasons why video forensics is still an emerging field, however more obstacles have prevented forensic investigators from addressing video forgeries. In the first place, creating realistic forgeries with videos is more laborious than tampering with images; meanwhile, video streams can be encoded

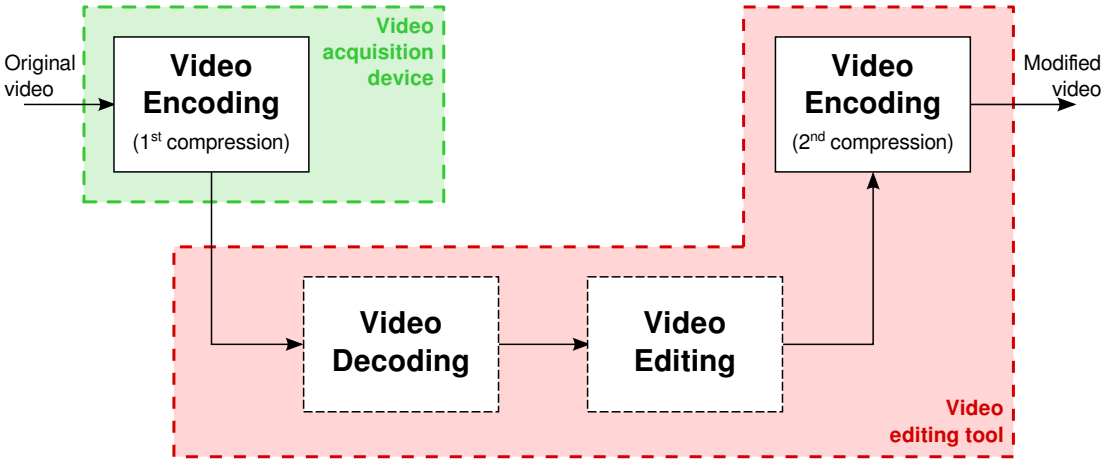


Figure 1.5: Processing chain for video manipulation.

through a large variety of encoding parameters and different compression formats, whereas digital images are usually available either in uncompressed or JPEG format. Finally, video sequences often go through stronger compressions compared to digital images, making their forensic analysis more difficult. This contrasts with the fact that nowadays digital videos are probably more used than images for security tasks (e.g., in video-surveillance systems), so their trustability must be strengthened.

Generally, existing video editing tools do not work directly on the compressed domain, but in the reconstructed spatio-temporal domain. Therefore, the process of editing a video sequence is composed of at least three main steps, as it is illustrated in Figure 1.5. At the beginning, the input video sequence is decoded, then the actual video editing task takes place, and as a last step, the edited video is re-encoded (possibly with a distinct codec or different encoding parameters).

As a consequence of such hardly avoidable but characteristic processing chain, one of the most studied tasks in video forensics is the detection of double encoding and/or transcoding. On the other hand, leveraging on the initial work on image forensics, several techniques are based on the study of the effects introduced by double quantization in DCT coefficients. Although not always applicable, these techniques are also of interest since intra coded frames (which might lead to double quantization traces in several video coding standards) are periodically generated in video streams to allow random access.

In the following, we start by covering the basics on video coding to introduce afterward the related works on video forensics. Finally, the main contributions of this thesis on the forensic analysis of video sequences are outlined.

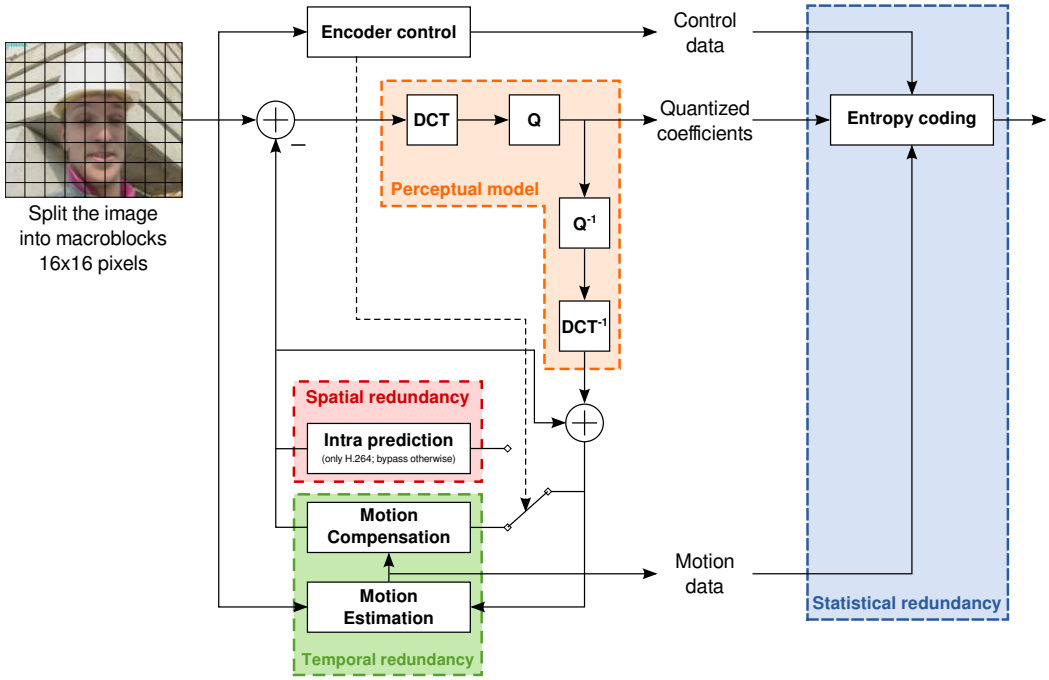


Figure 1.6: Block-based hybrid video coding scheme.

1.3.2. Video Coding Description

Over the past few decades, different video compression standards have emerged, being MPEG-2 [24], MPEG-4 Visual Part 2 [25] (we will refer to this one as MPEG-4) and H.264 [26] the three most broadly used. The oldest one, MPEG-2, is still widely used for video content storage in DVD and for broadcast television. MPEG-4, instead, has been mostly adopted in video surveillance systems and for video content sharing over the Internet. The most recent, H.264, is nowadays considered as the state-of-the-art in video compression and it is gradually replacing all its predecessors in almost all the mentioned applications, because it gives better performance than any of the preceding standards [27]. Very recently, the new standard H.265 [28] (successor of H.264) has started to be deployed, albeit its use in real systems is still scarce so we exclude its study from the forensic analysis of compressed video sequences.

Although each standard defines its own coding characteristics, their design is built over a common block-based hybrid video coding scheme which consists of motion compensated prediction and DCT-based transform quantization of the prediction error (cf. Figure 1.6), thus sharing several syntax elements. According to the block-based structure, each picture from a captured video sequence is divided into macroblocks of size 16×16 samples, which are encoded with the most suitable coding mode from each particular standard.

Different types of pictures are defined depending on the prediction process

carried out during the encoding. The three standards share the definition of intra-coded and inter- (or predictive-) coded pictures. In the former type, each macroblock is encoded without referring to other pictures within the video signal. We will identify this type of pictures as I-frames.[4] In the latter type of picture, the macroblocks can be additionally predicted from already coded and reconstructed frames (i.e., reference frames), which leads to two possible types of frames: the usually named P-frames and B-frames. The macroblocks in P-frames can only be predicted from previous reference frames, while those on B-frames can be estimated from past and/or future reference frames.

The different types of frames can be grouped into sequences, creating a Group Of Pictures (GOP). Formally, a GOP is an encoding of a sequence of frames that contains all the information that can be completely decoded within that GOP [29]. Therefore, in general, a GOP is composed of only one I-frame that indicates the beginning of the group and some combinations of P- and B-frames. We do not tackle B-frames in this thesis due to their associated complexity, so we constrain the compression to be performed according to the baseline profile for H.264 and to the equivalent simple profile for MPEG-2 and MPEG-4.

Every standard proposes its own intra/inter coding modes for each type of frame with the final goal of increasing coding efficiency. However, they all follow the same basic design: each macroblock is either coded in an intra- or an inter-coding mode, as it can be checked in Figure 1.6. Intra coding modes only exploit spatial redundancy from the captured scene, resorting to a DCT-based transform of the macroblock itself (or the residual signal obtained from an intra prediction, as in H.264). On the other hand, inter coding modes take advantage of the temporal redundancy among neighboring frames through motion compensated prediction and DCT-based transform of the achieved prediction error. Ultimately, the resulting transform coefficients in either case are quantized and then entropy-coded together with side information (e.g., particular coding modes, motion data, etc.).

1.3.3. Prior Work

Following the trail of image forensics, a growing body of literature seeking characteristic footprints left by video processing tools is now rising. Forensic researchers have been developing effective video forensic strategies intended for reconstructing the processing history of video signals under analysis and also for validating their origin. A thorough overview and taxonomy of published video forensic techniques can be found in [3].

[4] Only progressive-scan videos and full-frame encodings are considered in this thesis, thus, for the sake of clarity, the term frame is used to represent a picture or a slice independently of the standard.

As previously discussed, it is broadly accepted that double encoding is a necessary step when creating a digital video forgery, since most of the time a first encoding will occur during the acquisition process and a second one when storing the manipulated content. Pushed by such motivation, several approaches targeting this problem have been proposed, borrowing the acquired knowledge on double compression from image forensics. Accordingly, numerous works are based on the effects introduced by double quantization in the DCT coefficients of intra-coded frames.

Along these lines, the authors in [30] (that further extended the idea in [31]), unveil the artifacts left by the second compression on the distribution of the quantized DCT coefficients stemming from I-frames of a double MPEG-2 compressed video. In particular, they reveal that the histograms of two specific DCT coefficients follow a monotonically decreasing trend when the video is encoded once, while a convex shape is exhibited in presence of double encoding. A threshold-based detector is first derived to detect the convex pattern in [30], whereas a vector of features obtained from the histogram values is fed to an SVM classifier in [31]. A distinctive aspect of this approach is that it is able to work with Constant Bit Rate (CBR) encoded video streams which are more challenging than those encoded with a fixed quantization scale factor, i.e., Variable Bit Rate (VBR) encoded videos. Both detectors show promising results, however their performance drops when smaller bitrates are used in the second compression.

A similar method has been proposed in [32] to detect double compression with a different video coding standard, i.e., in H.264 video streams. Also in this case, authors take advantage of the double quantization effect and study the histogram of quantized DCT coefficients in the I-frames of the double encoded video. Nevertheless, the proposed detector which also relies on an SVM classifier only works when the second encoding is at a higher quality than the previous one. Recently, a different approach has been proposed in [33] to detect double MPEG-4 encoding. Instead of considering the histogram of DCT coefficients, authors model adjacent coefficients as a Markovian process: they evaluate the difference between adjacent coefficients obtaining a transition probability matrix. A feature is then extracted from such a matrix and used to train an SVM classifier. The method is tested on videos encoded twice in VBR mode, achieving very interesting results when the second encoding is performed at a lower quality than the first one.

Taking as reference an image forensics work by Fu et al. in [34], where the effect of double compression on JPEG images is analyzed through the Benford's law, different approaches have extended such model to the detection of double compression in video sequences [35, 36, 37]. Specifically, in [35], a straightforward extension of Fu et al.'s work is adapted to deal with MPEG videos, while in [36], the first-digit distribution of DCT coefficients from I-frames is gathered to build a feature vector to be classified within an SVM framework, indicating whether the

second encoding has been carried out at a higher or lower bitrate with respect to the first one. Finally, in [37], a set of SVM classifiers is trained with the first-digit distribution of a subset of DCT coefficients, being able to detect multiple (up to 3) compressions of the same H.264 video stream.

Setting aside the populated family of methods relying on the DCT domain, Luo et al. proposed in [38] a different approach that measures the strength of block artifacts for each frame in MPEG-2 compressed videos. Following an iterative procedure (i.e., re-encoded versions of the video under analysis are generated removing each time one frame from the beginning of the sequence), an average measure of the strength of block artifacts is calculated for each frame. For single compressed videos, this averaged measure preserves a periodic behavior, whereas for double compressed videos an irregular behavior shows up enabling its detection. Nonetheless, authors do not provide in [38] a way to automatically detect such irregular event.

Up to this point, we have only described techniques that make possible the detection of double compression maintaining the same encoder in both compressions. However, a conversion from one codec to another (i.e., a transcoding operation) might be carried out during the elaboration of the video forgery. Starting from a video that is assumed to be double encoded, Bestagini et al. put forward a way to identify the video coding standard used during the first compression [39]. Their idea is to exploit the idempotency property of common coding schemes: assuming that the original implementation of the encoders is available and that VBR mode has been used during the first encoding, the video under analysis is re-encoded with every possible encoder and every possible quantization parameter, then the similarity between the resulting sequences and the analyzed video is measured. The similarity will show a peak when any of the tested encoding settings match the one used in the first compression. To avoid the dependency with the genuine codec, this idea has been further extended in [40] by employing eigen-algorithms. However, this method is still limited to VBR video sequences.

Besides double compression detection, other works have focused on the study of tampering, such as removal or insertion of frames. Wang and Farid were the first to propose an effective method for detecting removal of frames in [41]. They discovered that when a set of frames is deleted, a de-synchronization between the GOPs in the first and second encoding takes place, which induces a periodic behavior on the prediction error of P-frames along time. Therefore, by examining the presence of such periodicity in the frequency domain, the removal of frames can be unmasked. Following a different approach, another method is presented in [42], where the different characteristics of quantization matrices employed for intra- and inter-coded pictures is taken into account to find out GOP structure inconsistencies. The main assumption lies in the fact that, when an I-frame is re-compressed as a P- or B-frame, its high-frequency DCT coefficients will be negligible, whereas the frames encoded twice as inter will not show such effect.

By measuring the energy of high-frequency DCT coefficients for each frame, a threshold-based detector is defined in order to detect a change in the GOP structure and thus unveil tampering.

With the aim of gaining more knowledge about the processing undergone by a video signal, a further step is explored trying to localize the actual forgery both in the spatial and in the time domain. Intra-frame forgery localization is probably the most challenging problem and that is why existing techniques only work under strict assumptions [3]. The first approach in this direction is the one proposed by Wang and Farid in [43], where a double quantization analysis is applied separately for each macroblock of the video under study. The underlying idea is to look for the macroblocks that show traces of double quantization against those that do not, thus pointing out a possible patch from another previously encoded sequence. The analysis is limited to the frames that have been encoded twice as intra. A recent work by Bestagini et al. in [44], is able to reveal and localize two types of forgeries. One type consists in replacing a part of the video sequence with fixed images repeated in time, and a second type also replaces a part of the video, but with a portion of the same video from a different time interval. The localization of the former type of forgery is addressed by evaluating successive differences in the pixel domain across time, thus unveiling the tampered region where zero motion is obtained. The latter type of forgery is localized by adapting the method in [45]. Authors have shown that their approach works remarkably well with realistic forged video contents coming from the Surrey University Library for Forensic Analysis (SULFA) database [46].

We have restricted ourselves to the description of the foregoing methods since they are closely related to the work carried out in the second part of this thesis. However, readers can still widen their perspective on video forensics by referring to the following works: the localization of frame removal and insertion in compressed videos has been extended to the three main codecs, i.e., MPEG-2, MPEG-4, and H.264, in [47]; the first anti-forensic technique capable of hiding evidence of frame deletion or addition in MPEG video sequences has been derived in [48] and further extended in [49]; an appealing approach linking video tampering detection with resampling detection to expose video splicing with different frame rates has been proposed in [50]; given the facility to cover digital footprints through video recapture (i.e., recapture is used as an anti-forensic technique), its detection has gained attention and has been tackled in several works [51, 52, 53]; finally, a systematic analysis of popular video file formats from a forensic point of view has been recently addressed in [54].

In the second part of this thesis, we mainly focus on the detection of double encoding and the localization of intra-frame forgeries in video sequences. We first disclose a new characteristic footprint, caused by double compression of a video signal, exploiting it to detect double encoding and also to provide valuable information from the first compression such as the size of the employed GOP.

Then, combining this with a double quantization analysis, intra-frame forgeries are also exposed.

1.3.4. Contributions

The main contributions concerning the forensic analysis of video sequences in this second part of the thesis are summarized below:

VC1. Discovery of a new digital footprint that is left behind when a video sequence is encoded twice. Such footprint reflects an unexpected change in the macroblock prediction types of re-encoded P-frames. By performing an analysis of the periodicity of this footprint across time, a threshold-based detector is designed to reveal the presence of double encoded videos. Surprisingly, the characteristic footprint is detectable on CBR videos and can endure relatively strong second compressions.

VC2. Estimation of part of the processing history of a video sequence under analysis. Specifically, a blind estimation of the length of the GOP in the first compression is derived by processing the periodicity of the extracted footprint over time. Given that most of the cameras work with a fixed and distinct GOP size, the estimation of the GOP in the first compression is valuable for a forensic analyst since, for instance, it might help to link a forged video with a specific type of camera.

VC3. Design and evaluation of a novel and practical solution for localizing forgeries in MPEG-2 video sequences. This solution uses the above GOP size estimation as a means to expose originally coded I-frames and combines it with a double quantization analysis on the resulting double coded I-frames to provide a probability map of tampering for each frame under evaluation.

1.4. Structure of the Thesis

The content of this thesis is structured in 9 chapters, divided into two parts. Part I includes Chapters 2 to 6 and describes the contributions on forensic analysis of resampled signals. Part II includes Chapters 7 and 8 and is composed of the contributions on forensic analysis of video sequences. Finally, Chapter 9 elaborates the conclusions drawn from the ideas introduced in this thesis and provides possible future lines of work.

1.5. Publications

In the first part of this thesis, Chapters 2 to 6 comprehend the research work which led to the following publications:

R1 David Vázquez-Padín, Carlos Mosquera, and Fernando Pérez-González. *Two-Dimensional Statistical Test for the Presence of Almost Cyclostationarity on Images.* In IEEE International Conference on Image Processing (ICIP'2010), Hong Kong, China, September 2010.

R2 David Vázquez-Padín and Fernando Pérez-González. *Exposing Original and Duplicated Regions Using SIFT Features and Resampling Traces.* In 10th International Workshop on Digital Forensics and Watermarking (IWDW'2011), Atlantic City, NY, USA, October 2011.

R3 David Vázquez-Padín and Fernando Pérez-González. *Prefilter Design for Forensic Resampling Estimation.* In IEEE International Workshop on Information Forensics and Security (WIFS'2011), Foz do Iguaçu, Brazil, December 2011.

R4 David Vázquez-Padín and Pedro Comesaña. *ML Estimation of the Resampling Factor.* In IEEE International Workshop on Information Forensics and Security (WIFS'2012), Tenerife, Spain, December 2012.

R5 David Vázquez-Padín, Pedro Comesaña, and Fernando Pérez-González. *Set-Membership Identification of Resampled Signals.* In IEEE International Workshop on Information Forensics and Security (WIFS'2013), Guangzhou, China, November 2013.

R6 David Vázquez-Padín, Pedro Comesaña, and Fernando Pérez-González. *An SVD Approach to Forensic Image Resampling Detection.* In European Signal Processing Conference (EUSIPCO'2015), Nice, France, September 2015.

In the second part of this thesis, Chapters 7 and 8 comprise the research work which led to the following publications:

V1 David Vázquez-Padín, Marco Fontani, Tiziano Bianchi, Pedro Comesaña, Alessandro Piva and Mauro Barni. *Detection of Video Double Encoding with GOP Size Estimation.* In IEEE International Workshop on Information Forensics and Security (WIFS'2012), Tenerife, Spain, December 2012.

V2 Daniele Labartino, Tiziano Bianchi, Alessia De Rosa, Marco Fontani, David Vázquez-Padín, Alessandro Piva, and Mauro Barni. *Localization of Forgeries in MPEG-2 Video through GOP size and DQ Analysis.* In IEEE International Workshop on Multimedia Signal Processing (MMSP'2013), Pula (Sardinia), Italy, October 2013. Top 10% Award.

Table 1.2: Summary of chapters, contributions, and publications.

Parts	**Chapters**	**Contributions**	**Publications**
I	Chapter 2	RC1, RC5	R1, R2
I	Chapter 3	RC1	R3
I	Chapter 4	RC2	R4
I	Chapter 5	RC3	R5
I	Chapter 6	RC4	R6
II	Chapter 7	VC1, VC2	V1
II	Chapter 8	VC3	V2

In addition to these publications, the following patent application has been derived as a result of other parallel works in active video forensics:

P1 *Title*: METHOD AND SYSTEM FOR EMBEDDING INFORMATION AND AUTHENTICATING A H.264 VIDEO USING A DIGITAL WATERMARK
International Application No.: PCT/EP2013/068067
Filling date: 02/09/2013
Inventors: L. Pérez-Freire (ES), G. Domínguez-Conde (ES), D. Vázquez-Padín (ES), L. Z. Dzianach (PL)
Applicant: Centum Research & Technology S.L.U.

Finally, the relation between the different chapters, contributions and published papers is summarized in Table 1.2.

Part I

Forensic Analysis of Resampled Signals

Chapter 2

Study of the Presence of Almost Cyclostationarity on Images

In this chapter, we first study the presence of almost cyclostationary fields in images for the detection and estimation of digital forgeries. The almost periodically correlated fields in the two-dimensional space are introduced by the necessary resampling operation associated to the applied spatial transformation. In this theoretical context, we extend to the two-dimensional space a statistical time-domain test for unveiling the presence of cyclostationarity. The proposed method allows us to estimate the scaling factor and the rotation angle of resized and rotated images, respectively. Examples of the output of the derived method are shown and comparative results are presented to evaluate the performance of the two-dimensional extension.

In the last part of the chapter, we address a common type of digital image forgery, known as copy-move image splicing, consisting in the duplication of a region from the image itself to conceal or duplicate some portion of the captured scene. Combining the aforementioned resampling-based method with an existing detector of copy-move manipulations, we provide a practical solution to point out and differentiate which is the original region and which is the tampered one by analyzing the resampling factor of each area. Comparative results are also presented in this case to evaluate the performance of the combination of both approaches.

2.1. Introduction

Throughout the first part of the previous chapter we have seen that a lot of powerful and intuitive software editing tools are nowadays available, facilitating the manipulation and alteration of digital images. With the aim of identifying

traces of possible forgeries, we have also highlighted several passive techniques working in the absence of any known signal. From this, we have noticed that one of the main problems addressed in this field is the detection of geometric transformations—such as scaling, rotation, or skewing—since they are usually employed when an image forgery is carried out.

The detection of these spatial transformations has been studied following different approaches as pointed out in Section 1.2.3, but in this chapter we will mainly focus on the work done by Mahdian and Saic in [11]. Extending the idea proposed by Gallagher in [10], they suggest to filter the image under analysis with a second-order derivative filter, apply a Radon transform at specific angles and then study the covariance of the resulting signal in the frequency domain. By doing so, they provide a blind and very fast method capable of detecting traces of spatial transformations. However, the proposed method presents some weaknesses in the estimation of the scaling factor and the rotation angle, due to the projection onto a single dimension of the Radon transform.

Motivated by these shortcomings and the need of a theoretical framework to explain why interpolated images present periodically correlated fields, we propose to use the cyclostationarity theory for resampling factor estimation. The derived method is a two-dimensional extension of a statistical time-domain test proposed by Dandawaté and Giannakis in [55], allowing us to estimate the resampling factor of a spatially transformed image, specifically the scaling factor and the rotation angle.

In the next section, we first synthesize the model for the spatial transformation of images (which has been thoroughly described in Section 1.2.2), and then we introduce the cyclostationarity theory needed for the estimation of the resampling factor. In Section 2.3, the two-dimensional extension of the time-domain test for revealing the presence of cyclostationarity is carried out. Section 2.4 presents the results obtained by our method, drawing a comparison with those obtained by Mahdian and Saic's approach [11]. Reaching the final part of the chapter, in Section 2.5, the novel and practical solution enabling the distinction of original and duplicated regions is described. Finally, Section 2.6 provides the conclusions.

2.2. Preliminaries and Problem Statement

Throughout this chapter we will consider an original image as the output provided by an acquisition system after the operations of sampling and quantization. The resulting digital image $\mathbf{F}$ is a matrix of integer values defined on a discrete grid of size $P \times Q$, where each element $F_{p,q}$ represents a gray level. The convention used for the source coordinates with indices (p, q) is that $p \in \{0, \ldots, P-1\}$ represents the vertical axis and $q \in \{0, \ldots, Q-1\}$ the horizontal one.

2.2.1. Spatial Transformations

The spatial transformation of an original image $\mathbf{F}$ maps the intensity value at each pixel location (p, q) to another location (i, j) in the new resampled image $\mathbf{G}$, whose elements are denoted by $G_{i,j}$. The most commonly used transformation is the affine one that combines several linear operations like translation, rotation, scaling, skewing, etc. The mapping can be expressed as:

$$\begin{pmatrix} i \\ j \end{pmatrix} = \mathbf{A} \begin{pmatrix} p \\ q \end{pmatrix} + \mathbf{b},$$

where $\mathbf{A}$ is the matrix that defines the linear transformation and $\mathbf{b}$ represents the translation vector. In general, the pixels in the resulting image will not map to exact integer coordinates on the source image, but rather to intermediate locations between source pixels. Therefore, when any of the mentioned spatial transformations is performed, it is necessary to apply a pixel interpolation algorithm. The interpolation of a spatial transformed image by a generic resampling factor $\boldsymbol{\xi} \triangleq (\xi_1, \xi_2) \triangleq (L_1/M_1, L_2/M_2)$ can be modeled by the following expression:

$$G_{i,j} = \sum_{k=0}^{P-1} \sum_{l=0}^{Q-1} h\left(i\frac{M_1}{L_1} - k\right) h\left(j\frac{M_2}{L_2} - l\right) F_{k,l}, \tag{2.1}$$

where $h(\cdot)$ represents the one-dimensional impulse response of any interpolation kernel, such as those gathered in Table 1.1.[1] Many different interpolation filters are available with different characteristics, but as shown in Table 1.1, the most common are: linear, cubic, and truncated-sinc kernels.

Finally, to fit the original resolution, the resampled values must be quantized to the original precision, having $R_{i,j} = Q_\Delta(G_{i,j})$, where $R_{i,j}$ denotes each element of the quantized resampled image $\mathbf{R}$, and $Q_\Delta(\cdot)$ stands for a uniform scalar quantizer with step size Δ.

2.2.2. Cyclostationary Approach

Once we have mathematically described the resampling process, we can observe from (2.1) that an interpolated image can be seen as a random field (i.e., the original image) that is periodically filtered with the same kernel. As a consequence, the resampled image will exhibit periodically correlated fields (cf. [56]) with a period equal to the resampling factor $\boldsymbol{\xi} = (L_1/M_1, L_2/M_2)$. Equivalently, the output image is cyclostationary with period $\boldsymbol{\xi}$.

[1] For the sake of simplicity, but without loss of generality, we refrain from explicitly adding the shift δ between the two sampling grids as in (1.4).

In the one-dimensional case, Sathe and Vaidyanathan showed in [57] that the output of a multirate system that performs sampling rate conversion by a factor $\xi \triangleq L/M$, produces a cyclostationary signal with period $L/\mathrm{GCD}(L, M)$ provided that the input signal is wide-sense stationary (the output becomes wide-sense stationary only when the interpolation filter is ideal). They only take into account pure cyclostationary processes, i.e., with an integer cyclic period; nevertheless, for the estimation of the resampling factor it is more convenient to consider that the output can be an almost cyclostationary process.

This idea regarding multirate systems can be extended to the spatial domain with two dimensions, but before we have to extend the concept of almost cyclostationarity to the two-dimensional space. As it is mentioned in [58], those time series that have an "almost integer" period accept generalized (or limiting) Fourier expansions, so following the definition in [56] of periodically correlated fields with an integer period, we introduce the concept of almost cyclostationary random fields.

Definition 1 *Let $x(\mathbf{m}) \triangleq x(m_1, m_2)$ be a real random field with mean $\mu_x(\mathbf{m}) \triangleq E\{x(\mathbf{m})\}$ and covariance $c_{xx}(\mathbf{m}; \boldsymbol{\tau}) \triangleq E\{[x(\mathbf{m}) - \mu_x(\mathbf{m})][x(\mathbf{m}+\boldsymbol{\tau}) - \mu_x(\mathbf{m}+\boldsymbol{\tau})]\}$, where $\mathbf{m} \triangleq (m_1, m_2) \in \mathbb{Z}^2$ and $\boldsymbol{\tau} \triangleq (\tau_1, \tau_2) \in \mathbb{Z}^2$. The random field $x(m_1, m_2)$ is strongly almost periodically correlated (equivalently, almost cyclostationary) with period $\mathbf{T} \triangleq (T_1, T_2)$, if and only if its mean and covariance functions satisfy*

$$\mu_x(m_1, m_2) = \mu_x(m_1 + kT_1, m_2 + lT_2),$$
$$c_{xx}(m_1, m_2; \boldsymbol{\tau}) = c_{xx}(m_1 + kT_1, m_2 + lT_2; \boldsymbol{\tau}),$$

for all integers $m_1, m_2, \tau_1, \tau_2, k, l$ and rational numbers T_1, T_2.

Such random fields accept generalized Fourier expansions and assuming that $x(m_1, m_2)$ has zero mean, the generalized Fourier series pair for every $\boldsymbol{\tau}$ is:

$$c_{xx}(m_1, m_2; \boldsymbol{\tau}) = \sum_{(\alpha_1, \alpha_2) \in \mathcal{A}_{xx}} C_{xx}(\alpha_1, \alpha_2; \boldsymbol{\tau}) e^{j(\alpha_1 m_1 + \alpha_2 m_2)},$$

$$C_{xx}(\alpha_1, \alpha_2; \boldsymbol{\tau}) = \lim_{M_1, M_2 \to \infty} \frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} c_{xx}(m_1, m_2; \boldsymbol{\tau}) e^{-j(\alpha_1 m_1 + \alpha_2 m_2)}, \quad (2.2)$$

where (α_1, α_2) represents each frequency pair in the cyclic domain. The set of cyclic frequencies $\mathcal{A}_{xx} \triangleq \{\boldsymbol{\alpha} \triangleq (\alpha_1, \alpha_2) : C_{xx}(\boldsymbol{\alpha}; \boldsymbol{\tau}) \neq 0, -\pi < \alpha_1, \alpha_2 \leq \pi\}$ must be countable and we assume that the limit exists in the mean-square sense. To express those random fields in terms of Fourier Transforms, we define the cyclic spectrum.

Definition 2 *The cyclic spectrum for random fields $x(m_1, m_2)$, is defined as:*

$$S_{xx}(\boldsymbol{\alpha}; \omega_1, \omega_2) \triangleq \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} C_{xx}(\boldsymbol{\alpha}; \tau_1, \tau_2) e^{-j(\omega_1 \tau_1 + \omega_2 \tau_2)},$$

where $\boldsymbol{\omega} \triangleq (\omega_1, \omega_2)$ *represents each frequency pair in the frequency domain.*

In order to show the presence of almost cyclostationary fields in a resampled image by a rational factor $\boldsymbol{\xi}$, we consider the single case when the original image is an infinite-length white-noise random field with zero mean and variance equal to one. In this situation the cyclic correlation becomes

$$c_{xx}(m_1, m_2; \boldsymbol{\tau}) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h\left(m_1 \frac{M_1}{L_1} - i\right) h\left(m_2 \frac{M_2}{L_2} - j\right) \times h\left((m_1 + \tau_1)\frac{M_1}{L_1} - i\right) h\left((m_2 + \tau_2)\frac{M_2}{L_2} - j\right),$$

and it is easy to see that

$$c_{xx}(m_1, m_2; \boldsymbol{\tau}) = c_{xx}\left(m_1 + k\frac{L_1}{M_1}, m_2 + l\frac{L_2}{M_2}; \boldsymbol{\tau}\right),$$

with $k, l \in \mathbb{Z}$. Hence, unless the kernel used is ideal, the output is almost cyclostationary with period $\mathbf{T} = (L_1/M_1, L_2/M_2)$. The same reasoning can be applied to real images, albeit with an unknown distribution which makes more difficult the estimation of the cyclic period. For this reason, we choose to extend the time-domain test proposed by Dandawaté and Giannakis in [55] that allows the detection of almost periodicities without considering a specific distribution on the data.

2.3. Extension of the Time-Domain Test

The calculation of the scaling factor $\boldsymbol{\xi}$ or the rotation angle θ of a spatially transformed image can be achieved through the estimation of the cyclic frequencies $\boldsymbol{\alpha}$, as we will see at the end of this section. Assuming that an image block of size $N \times N$ can be modeled through a real random field $z(m_1, m_2)$ with zero mean, the detection of the set of cyclic frequency pairs in (2.2) can be made through the estimation of the cyclic correlation:

$$\begin{aligned}\hat{C}_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau}) &= \hat{C}_{zz}(\alpha_1, \alpha_2; \tau_1, \tau_2) \\ &= \frac{1}{N^2} \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} z(m_1, m_2) z(m_1 + \tau_1, m_2 + \tau_2) e^{-j(\alpha_1 m_1 + \alpha_2 m_2)}. \end{aligned} \quad (2.3)$$

This estimate $\hat{C}_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau})$ is asymptotically unbiased according to Definition 1. Thus, if we represent $e_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau})$ as the estimation error and $C_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau})$ as the ideal covariance, the estimation provides:

$$\hat{C}_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau}) = C_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau}) + e_{zz}(\boldsymbol{\alpha}; \boldsymbol{\tau}),$$

where $e_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau})$ vanishes asymptotically as $N \to \infty$. To make a decision about the presence or absence of a given cyclic frequency in the image block, we build up a vector from $\hat{C}_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau})$ evaluated in a set of K lags $\{\boldsymbol{\tau}_k\}_{k=1}^{K} = \{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_K : \boldsymbol{\tau}_k \in \mathbb{Z}^2\}$:

$$\hat{\mathbf{c}}_{zz} \triangleq \frac{1}{\sqrt{2}}\left(\hat{C}_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau}_1), \ldots, \hat{C}_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau}_K), \hat{C}^{*}_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau}_1), \ldots, \hat{C}^{*}_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau}_K)\right)^{T},$$

and we consider the following hypothesis testing problem:

$$\begin{aligned} &\mathcal{H}_0 : \boldsymbol{\alpha} \notin \mathcal{A}_{zz}, \forall\{\boldsymbol{\tau}_k\}_{k=1}^{K} && \Rightarrow && \hat{\mathbf{c}}_{zz} = \mathbf{e}_{zz}, \\ &\mathcal{H}_1 : \boldsymbol{\alpha} \in \mathcal{A}_{zz}, \text{for some } \{\boldsymbol{\tau}_k\}_{k=1}^{K} && \Rightarrow && \hat{\mathbf{c}}_{zz} = \mathbf{c}_{zz} + \mathbf{e}_{zz}, \end{aligned} \tag{2.4}$$

where $\mathcal{A}_{zz} \triangleq \{\boldsymbol{\alpha} = (\alpha_1, \alpha_2) : C_{zz}(\boldsymbol{\alpha};\boldsymbol{\tau}) \neq 0, -\pi < \alpha_1, \alpha_2 \leq \pi\}$. Note that $\mathbf{c}_{zz}$ is the corresponding true value of the cyclic correlation vector and $\mathbf{e}_{zz}$ is the estimation error vector. From (2.4), if we know the distribution of the estimation error $\mathbf{e}_{zz}$, we can seek a threshold to detect the cyclic frequency pairs (α_1, α_2) given that $\mathbf{c}_{zz}$ is deterministic. Dandawaté and Giannakis use the asymptotic properties of the cyclic correlation estimator to infer the asymptotic distribution of the estimation error. In our case, considering that the extension to the spatial domain of the mixing conditions (**A1** in [55]) is fulfilled, then the cyclic correlation estimator in (2.3) is asymptotically normal and thus the error estimation converges in distribution to a multivariate normal, i.e.,

$$\lim_{N\to\infty} N\mathbf{e}_{zz} \overset{\mathcal{D}}{=} \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{zz}),$$

where $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{zz})$ represents a multivariate normal distribution with zero-mean vector and asymptotic covariance matrix $\boldsymbol{\Sigma}_{zz}$, which is defined as follows:

$$\boldsymbol{\Sigma}_{zz} \triangleq \lim_{N\to\infty} N^2 \mathrm{cov}\{\hat{\mathbf{c}}_{zz}, \hat{\mathbf{c}}^{H}_{zz}\} = \frac{1}{2}\begin{bmatrix} \mathbf{S}^{(*)}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(\mathbf{0}; -\boldsymbol{\alpha}) & \mathbf{S}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(2\boldsymbol{\alpha}; \boldsymbol{\alpha}) \\ (\mathbf{S}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(2\boldsymbol{\alpha}; \boldsymbol{\alpha}))^{*} & (\mathbf{S}^{(*)}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(\mathbf{0}; -\boldsymbol{\alpha}))^{*} \end{bmatrix}.$$

In the above expression, $\mathbf{S}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(\boldsymbol{\alpha},\boldsymbol{\omega})$ is a $K \times K$ matrix whose (k,l)-th entries are given by the cyclic cross-spectrum of $z_{\boldsymbol{\tau}_k}(\mathbf{m}) \triangleq z(\mathbf{m})z(\mathbf{m}+\boldsymbol{\tau}_k)$ and $z_{\boldsymbol{\tau}_l}(\mathbf{m}) \triangleq z(\mathbf{m})z(\mathbf{m}+\boldsymbol{\tau}_l)$ for the different K lags and, similarly, matrix $\mathbf{S}^{(*)}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(\boldsymbol{\alpha},\boldsymbol{\omega})$ is obtained from the cyclic cross-spectrum of $z_{\boldsymbol{\tau}_k}(\mathbf{m})$ and $z^{*}_{\boldsymbol{\tau}_l}(\mathbf{m})$ at the different lags. Hence, for N large enough, the vector $\hat{\mathbf{c}}_{zz}$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ differs only in the mean. In order to solve this detection problem, we use (in the same way as in [55]) the norm of a weighted version of the cyclic correlation estimation vector ($\boldsymbol{\gamma} = N\hat{\mathbf{c}}^{H}_{zz}\hat{\boldsymbol{\Sigma}}^{-1/2}_{zz}$), so the statistic and then the likelihood ratio test with a threshold Γ correspond to:

$$\mathcal{T}_{zz} = \|\boldsymbol{\gamma}\|_2^2 = N^2\hat{\mathbf{c}}^{H}_{zz}\hat{\boldsymbol{\Sigma}}^{-1}_{zz}\hat{\mathbf{c}}_{zz} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \Gamma,$$

where $\|\cdot\|_2$ stands for the Euclidean norm and $\hat{\mathbf{\Sigma}}_{zz}$ is an estimate of the asymptotic covariance matrix. From Theorem 2 in [55], the statistic $\mathcal{T}_{zz}$ has the following asymptotic distribution under $\mathcal{H}_0$

$$\lim_{N\to\infty} \mathcal{T}_{zz} \overset{\mathcal{D}}{=} \chi^2_{2K},$$

where χ^2_{2K} represents a chi-square distribution with $2K$ degrees of freedom. Under $\mathcal{H}_1$ and for N large enough, the asymptotic distribution is approximately Gaussian

$$\mathcal{T}_{zz} \sim \mathcal{N}(N^2 \hat{\mathbf{c}}_{zz}^H \hat{\mathbf{\Sigma}}_{zz}^{-1} \hat{\mathbf{c}}_{zz}, 4N^2 \hat{\mathbf{c}}_{zz}^H \hat{\mathbf{\Sigma}}_{zz}^{-1} \hat{\mathbf{c}}_{zz}).$$

Once we know the asymptotic distribution of the statistic $\mathcal{T}_{zz}$ under the two hypotheses, we can set the threshold Γ for a fixed probability of false alarm $P_F = \Pr(\mathcal{T}_{zz} \geq \Gamma|\mathcal{H}_0) = \Pr(\chi^2_{2K} \geq \Gamma)$ and then estimate the set of cyclic frequencies $\mathcal{A}_{zz}$. Below, the fundamental steps for the implementation of our method are presented.

As a first step, the image block under analysis $\mathbf{Z}$ of size $N \times N$ is selected from the quantized resampled image $\mathbf{R}$. Then, the mean from $\mathbf{Z}$ is removed yielding a zero-mean random field $z(m_1, m_2)$ with $m_1, m_2 \in \{0, \ldots, N-1\}$. Finally, the following algorithm for each frequency pair $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ defined in the Discrete Fourier Transform (DFT) grid is applied:

1. From the data $z(m_1, m_2)$ and using (2.3), we compute the vector $\hat{\mathbf{c}}_{zz}$ for a fixed set of K lags $\{\boldsymbol{\tau}_k\}_{k=1}^K$.

2. We estimate the asymptotic covariance matrix $\mathbf{\Sigma}_{zz}$ using the cyclic spectrum estimator. From the two options available for cyclic spectral estimation [58], we use the smoothed periodogram with a frequency domain window $W(\omega_1, \omega_2)$ of size $P \times P$ (with P odd). So, defining

$$I_{\boldsymbol{\tau}}(\omega_1, \omega_2) \triangleq \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} z(m_1, m_2) z(m_1 + \tau_1, m_2 + \tau_2) e^{-j(\omega_1 m_1 + \omega_2 m_2)},$$

we calculate the elements of the matrix $\hat{\mathbf{\Sigma}}_{zz}$ as

$$\hat{\mathbf{S}}^{(*)}_{\boldsymbol{\tau}_k, \boldsymbol{\tau}_l}(\mathbf{0}; -\boldsymbol{\alpha}) = \frac{1}{(NP)^2} \sum_{r=-(P-1)/2}^{(P-1)/2} \sum_{s=-(P-1)/2}^{(P-1)/2} W(r, s) \times I_{\boldsymbol{\tau}_k}\left(\alpha_1 + \frac{2\pi r}{N}, \alpha_2 + \frac{2\pi s}{N}\right) I^*_{\boldsymbol{\tau}_l}\left(\alpha_1 + \frac{2\pi r}{N}, \alpha_2 + \frac{2\pi s}{N}\right),$$

and for $\hat{\mathbf{S}}_{\boldsymbol{\tau}_k, \boldsymbol{\tau}_l}(2\boldsymbol{\alpha}; \boldsymbol{\alpha})$ we take the same expression used for $\hat{\mathbf{S}}^{(*)}_{\boldsymbol{\tau}_k, \boldsymbol{\tau}_l}(\mathbf{0}; -\boldsymbol{\alpha})$,

but adopting $I_{\boldsymbol{\tau}_l}(\boldsymbol{\omega})$ instead of $I^*_{\boldsymbol{\tau}_l}(\boldsymbol{\omega})$, i.e.,

$$\hat{\mathbf{S}}_{\boldsymbol{\tau}_k,\boldsymbol{\tau}_l}(2\boldsymbol{\alpha};\boldsymbol{\alpha}) = \frac{1}{(NP)^2}\sum_{r=-(P-1)/2}^{(P-1)/2}\sum_{s=-(P-1)/2}^{(P-1)/2} W(r,s) \times I_{\boldsymbol{\tau}_k}\left(\alpha_1+\frac{2\pi r}{N},\alpha_2+\frac{2\pi s}{N}\right) I_{\boldsymbol{\tau}_l}\left(\alpha_1+\frac{2\pi r}{N},\alpha_2+\frac{2\pi s}{N}\right).$$

3. Once $\hat{\boldsymbol{\Sigma}}_{zz}$ is obtained, we calculate the test statistic $\mathcal{T}_{zz} = N^2\hat{\mathbf{c}}_{zz}^H\hat{\boldsymbol{\Sigma}}_{zz}^{-1}\hat{\mathbf{c}}_{zz}$.
4. For a given probability of false alarm P_F, we set Γ.
5. We declare the frequency pair $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ as cyclic if $\mathcal{T}_{zz} \geq \Gamma$.

After the application of the method, we obtain the resampling factor $\boldsymbol{\xi} = (\xi_1, \xi_2)$ from the detected cyclic frequencies (α_1, α_2), due to the relation between these and the cyclic periods (T_1, T_2), i.e., $\alpha_i = 2\pi/T_i = 2\pi/\xi_i$ with $i \in \{1, 2\}$. However, because of aliasing and for any $\xi_i > 1$, we have the same cyclic frequencies for the scaling factors ξ_i and $\frac{\xi_i}{\xi_i - 1}$. So despite this unavoidable ambiguity, the estimated value of the resampling factor $\hat{\boldsymbol{\xi}}$ can be computed as follows:

$$\hat{\xi}_i = \begin{cases} \dfrac{2\pi}{2\pi - |\alpha_i|}, & -\pi \leq \alpha_i \leq \pi \quad (1 < \hat{\xi}_i \leq 2) \\ \dfrac{2\pi}{|\alpha_i|}, & -\pi \leq \alpha_i \leq \pi \quad (\hat{\xi}_i \geq 2) \end{cases}, \tag{2.5}$$

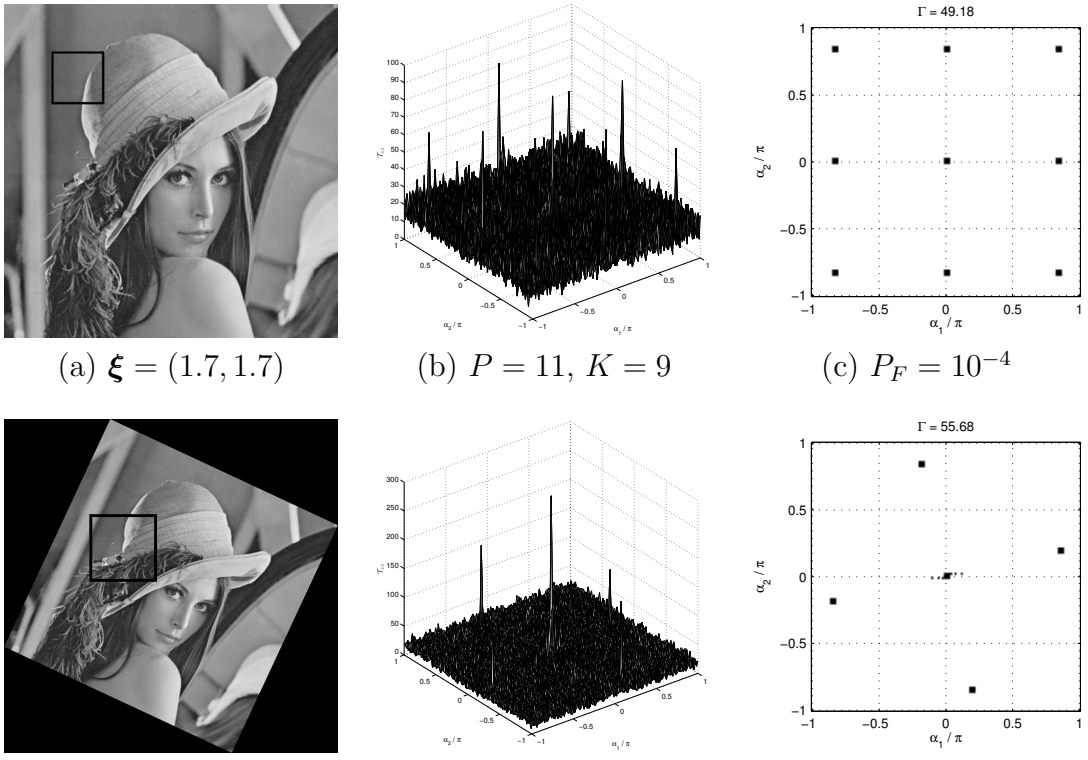
for $i \in \{1, 2\}$. On the other hand, if we consider that θ is the angle of rotation of the image in a counterclockwise direction around its center point, its estimation from the detected cyclic frequencies $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ can be reached through the following relation:

$$\phi = \arctan\left(\frac{\alpha_2}{\alpha_1}\right) \bmod \frac{\pi}{2},$$

where mod represents the modulo operation, and finally, the estimated angle is obtained by

$$\hat{\theta} = \begin{cases} -2\phi, & \text{if } \ 0 \leq \phi \leq \frac{\pi}{12} \\ -\arccos(\kappa), & \text{if } \frac{\pi}{12} < \phi \leq \frac{5\pi}{12} \\ \frac{\pi}{2} - 2\phi, & \text{if } \frac{5\pi}{12} < \phi \leq \frac{\pi}{2} \end{cases}, \tag{2.6}$$

where $\kappa \triangleq \cos^2(\phi)(\sqrt{2\tan(\phi)} - \tan(\phi) + \tan^2(\phi))$. From the above definition of the estimate, it is clear that our method will not be able to distinguish angles separated by 90°, i.e., the same estimation will be obtained for any $\theta + n\frac{\pi}{2}$ with $n \in \mathbb{Z}$. Note, however, that this ambiguity is also common to other resampling-based methods [8, 11]. Moreover, because of the DFT symmetry, the cyclic frequencies for the angles $\theta = -\frac{\pi}{6}$ and $\theta = -\frac{\pi}{3}$ are the same, thus yielding an ambiguity when estimating these precise angles.



(a) $\boldsymbol{\xi} = (1.7, 1.7)$ (b) $P = 11$, $K = 9$ (c) $P_F = 10^{-4}$

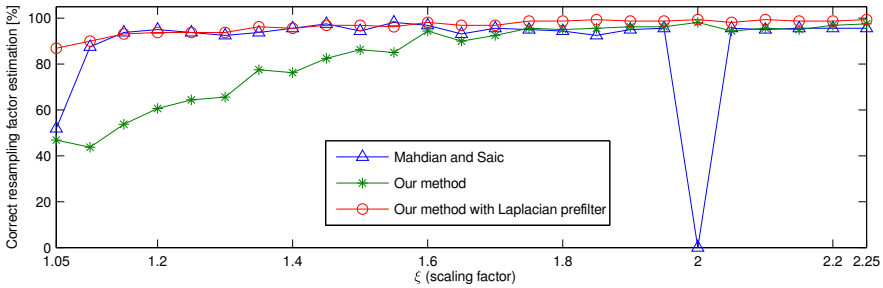
(d) $\theta = -25°$ (e) $P = 21$, $K = 9$ (f) $P_F = 10^{-5}$

Figure 2.1: Graphical results obtained with the proposed method for two different spatial transformations.

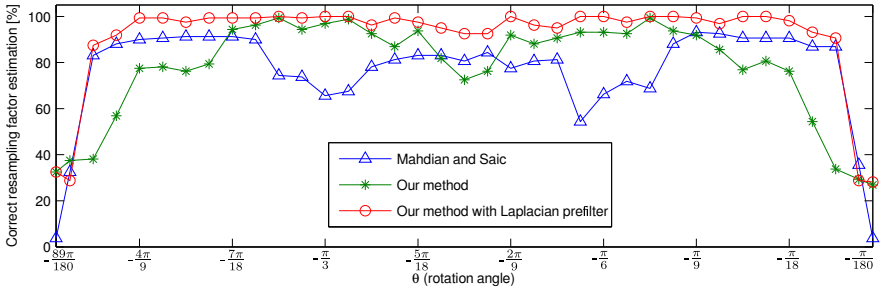
2.4. Experimental Results

With the aim of showing how is the output of our method, we present in Figure 2.1 the results obtained for two different spatial transformations. Figures 2.1(a) and 2.1(d) depict the analyzed block of size 128×128 pixels in each spatially transformed image. The statistic $\mathcal{T}_{zz}$ is plotted in Figures 2.1(b) and 2.1(e), where the peaks indicating the presence of possible cyclic frequencies can clearly be distinguished. In both cases, the spectral window used is a two-dimensional Kaiser window of parameter $\beta = 1$ with size $P \times P$. After applying the threshold Γ, we represent in Figures 2.1(c) and 2.1(f) the detected cyclic frequencies that make possible the identification of the applied transformation.

For the evaluation of our method, we use 40 TIFF format images from the Miscellaneous volume of the USC-SIPI image database (discarding the 4 test pattern images) and we perform two different experiments. In order to evaluate the performance of our method, we compare our results with those obtained using the technique proposed by Mahdian and Saic in [11]. Since our main objective is to detect forgeries in a relatively small region of the image, we use an image block $\mathbf{Z}$ of size 128×128 pixels for both approaches (i.e., $N = 128$). The sizes of



(a) $P = 11$, $K = 9$, $P_F = 10^{-4}$ and $\Gamma = 49.18$



(b) $P = 21$, $K = 9$, $P_F = 10^{-5}$ and $\Gamma = 55.68$

Figure 2.2: Comparative results obtained with both methods for different scaling factors and rotation angles.

the tested images are of 256×256, 512×512 or 1024×1024 pixels, so whenever possible we apply both methods to four blocks and take the average of the results obtained for each image.

In the first experiment, we study the estimation accuracy when all the images from the database are uniformly scaled by a factor ξ, i.e., using the same factor for each dimension, such that $\boldsymbol{\xi} = (\xi, \xi)$. The set of tested scaling factors is defined in the interval $[1.05, 2.25]$ discretized with step size 0.05 (i.e., $\xi \in \{1.05, 1.1, \ldots, 2.2, 2.25\}$), and the used interpolation kernel is Lanczos (cf. Table 1.1). We decide that the estimation is correct when the estimated scaling factor $\hat{\xi}_i$, obtained through (2.5), satisfies $|\hat{\xi}_i - \xi| < 0.05$ for any $i \in \{1, 2\}$.

In Figure 2.2(a) we plot the average percentage of successful resampling factor estimates for both methods. We also represent the estimation accuracy of our method applying first a Laplacian operator to the whole image. As we can see, the performance of our method is worse if we do not use the Laplacian prefilter, mainly for scaling factors close to 1. The application of a high-pass filter like the Laplacian operator eliminates low-frequency components (belonging to the image content) that are near the spectral peaks (corresponding to the cyclic frequencies associated to these scaling factors), thus improving the estimation results. It can also be observed that the method of Mahdian and Saic cannot detect the resampling factor $\xi = 2$, which is not an issue for ours.

In the second experiment, we analyze the performance of our method when all the images from the database are rotated by a discrete set of angles in the range $-\frac{\pi}{2} < \theta < 0$ sampled with a step size of $\frac{\pi}{72}$, and fixing the lowest value of the interval at $-\frac{89\pi}{180}$ and the highest at $-\frac{\pi}{180}$. In this case, we use the Catmull-Rom interpolation kernel (cf. Table 1.1) and we determine that the estimation of the angle is correct for our method when the estimated angle following (2.6) satisfies $|\hat{\theta} - \theta| < \frac{\pi}{72}$. For the method of Mahdian and Saic we use other criterion because we can only assess the angle from the position of the corresponding spectral peak, denoted by ω_θ, so in this case we decide that the angle is correct if $|\hat{\omega}_\theta - \omega_\theta| < 0.022$. The threshold used in both cases is equivalent because it corresponds to the minimum distance between the theoretical values for the defined set of angles.

Figure 2.2(b) shows the comparative results for the two approaches. The best results are obtained when our method is combined with the use of the Laplacian prefilter. We have to notice that the output of the method of Mahdian and Saic presents the spectral peaks at the same positions for any angle $\theta + n\frac{\pi}{4}$ with $n \in \mathbb{Z}$, so $\omega_\theta = \omega_{(\theta \mod -\frac{\pi}{4})}$ for $-\frac{\pi}{2} \leq \theta \leq -\frac{\pi}{4}$. Hence, their method shows more ambiguities than ours, which just fails at discerning $\theta = -\frac{\pi}{6}$ and $\theta = -\frac{\pi}{3}$ within the interval $-\frac{\pi}{2} < \theta < 0$. Despite of this, the shown results are presented without taking these errors into account.

As a conclusion, the proposed method performs better than the one described by Mahdian and Saic in [11] for estimating the parameters of spatially transformed images. As a counterpart, our method is more time consuming, but the processing in the two-dimensional space provides more information. For instance, we avoid some ambiguities caused by indistinguishable periodic patterns in the one-dimensional case. Note that all the experiments, including the resampling operations, were carried out in MATLAB.

2.5. Practical Solution: Exposing Original and Duplicated Regions

As previously pointed out, a characteristic type of digital image forgery is the duplication of a region in the same image to hide or duplicate some portion of the captured scene. The detection of region duplication forgeries has been recently addressed using methods based on SIFT features that provide points of the regions involved in the tampering and also the parameters of the geometric transformation between both regions. However, examining this output, there is no sufficient information about which of the two regions is the original and which is the duplicate. A reliable image forensic analysis must supply this information. Therefore, in this section, we outline how to use the above resampling-based

(a) Original image (b) Tampered image

Figure 2.3: Real example of a tampered image (on the right) shown in the BP website by copying and moving parts of the original image (on the left). Courtesy of The Washington Post, July 2010.

method for accurately distinguishing between the original and the tampered region by analyzing the resampling factor of each area.

2.5.1. Introduction

At the beginning of Chapter 1 we have discussed a motivating and representative case of how easy the alteration of a digital image can be. Specifically, we have seen that during the BP oil crisis, the image shown in Figure 2.3(a) was doctored on the BP website by filling the blank screens with other parts of the same picture yielding the forged image in Figure 2.3(b). The resulting image is a perfect example of a realistic copy-move manipulation.

Currently, in the context of passive forensic techniques there are several methods that are capable of detecting duplicated regions (cf. Section 5.1.1 in [1]), providing a set of matched regions, but being unable to determine which belong to the genuine scene and which are clones. We will tackle this problem by estimating the resampling factor in the matched regions. In particular, the proposed practical solution combines these two different and complementary forensic tools to reach a more accurate forensic analysis of tampered images. The main idea is to mitigate the drawbacks of each technique by using the characteristics of the other.

As a consequence, in the next section we start discussing in more detail the advantages and disadvantages of the selected types of techniques. In Section 2.5.3, the applied model is described focusing on the combination of both techniques as a means of improving performance. Finally, experimental results carried out with this image forensic scheme are summarized in Section 2.5.4.

(a) Region duplication detection (b) Resampling analysis

Figure 2.4: Examples of drawbacks of each technique. On the left, the detected regions are highlighted and tagged with **1** and **2**. On the right, the tampered region is highlighted and each analyzed block is denoted by a white border box.

2.5.2. Advantages and Disadvantages of each Technique

The complementary behavior of both techniques can be established from the analysis of advantages and drawbacks of each, as it is summarized below.

Advantages/drawbacks of region duplication detectors: First works in this area were based on an exhaustive search and analysis of correlation properties of the image [59]. However, more efficient approaches have been recently proposed, such as those in [60] and [61]. These two works address copy-move detection by searching for similar SIFT descriptors extracted from the image under analysis (more details on these descriptors will be given in Section 2.5.3.1).

Figure 2.4(a) illustrates the typical output from any of these copy-move detectors, where the matched regions are highlighted and tagged with numbers **1** and **2**. Even though these methods are capable of estimating the geometric relation between these two regions, they cannot distinguish the original region (i.e., **1**) from the cloned patch (i.e., **2**). Moreover, an important limitation from the SIFT-based methods comes also from the difficulty to extract reliable descriptors from less textured regions of the image, thus hindering detection performance.

Taking into account the advantages of the region duplication detectors, these methods are able to detect copy-move forgeries even when no geometric transformations are applied to the pasted regions. Furthermore, the recently proposed methods based on SIFT (i.e., [60] and [61]), allow for a very fast analysis of an entire image, in terms of computation time.

Advantages/drawbacks of resampling detectors: The detection of resampling traces and the estimation of the applied resampling factor have been

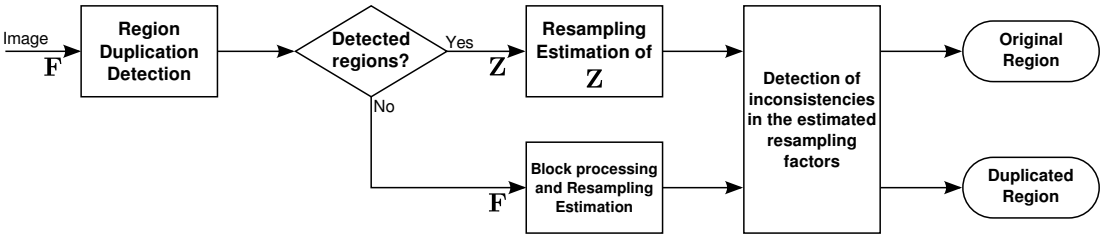


Figure 2.5: Block diagram of the proposed image forensic analysis tool.

studied in several works, as covered in Section 1.2.3. These methods are able to accurately estimate the scaling factor and the rotation angle of spatially transformed images, basing their analysis on periodic linear correlations introduced by the applied spatial transformation.

Although these methods provide good results in controlled scenarios, when they are evaluated in more realistic situations, their performance gets worse [62, 63]. For instance, as shown in Figure 2.4(b), it is very likely that the tampered region will not be aligned with the analysis grid, thus failing in the localization of the forgery. Notice that a non-overlapping block-based analysis is generally carried out to minimize the computation burden. Moreover, an important handicap of these methods is the impossibility to detect basic copy-move forgeries (without content adaptation), since the resampling factor of the whole image remains constant.

After highlighting the advantages and disadvantages of both approaches, it can be expected that the combination of them will provide better performance and also a more complete and accurate forensic analysis of tampered images.

2.5.3. Model Description

In order to overcome the problem related to the distinction of the original regions from the tampered ones using a region duplication detector, but also to avoid the aforementioned misdetections of the resampling detectors, the proposed approach uses a combination of both techniques.

In Figure 2.5 we represent in block diagram form the steps involved in the proposed forensic analysis of an image. As a first step, we use a region duplication detector to extract the original and the cloned regions. When the method is not able to find any duplicate, it is necessary to analyze the entire image following a block-based procedure and looking for inconsistencies in the resampling factor of each block. However, if the region duplication detector is capable of finding the duplicated regions, then the resampling-based method is just applied to estimate the resampling factor of each area. Finally, according to the results obtained in

the previous stages, the system determines and differentiates the original regions from the tampered ones.

Next, we describe the specific methods that are fit together to build a practical implementation of the proposed forensic analysis tool.

2.5.3.1. A SIFT-based method for region duplication detection

Nowadays, we can find several approaches based on the matching of image features and keypoints (e.g., [60] and [61]) which provide very good results for the detection of duplicated regions. In this case, we rely on the method proposed by Amerini et al. in [60].

Following the steps described in [60] we analyze a digital image with a single color channel $\mathbf{F}$ of size $P \times Q$ with elements $F_{p,q}$ and indices $p \in \{0, \ldots, P-1\}$ and $q \in \{0, \ldots, Q-1\}$. We apply the algorithm proposed by Lowe in [64] to produce a set $\mathcal{X}$ of N keypoints:

$$\mathcal{X} = \{\mathbf{x}_i \in \mathbb{Z}^2 : i = 0, \ldots, N-1\},$$

with their respective SIFT descriptors:

$$\mathcal{D} = \{\mathbf{d}_i \in \mathbb{R}^{128} : i = 0, \ldots, N-1\},$$

where each descriptor is a 128-dimensional vector. Since the descriptors of a duplicated region will look like those of the original area, we want to identify the nearest neighbor of each descriptor to find a possible match of similar keypoints. To that end, for each descriptor $\mathbf{d}_i$, the Euclidean distance between each pair of descriptors is computed and gathered in a set $\mathcal{S}_i$, obtaining

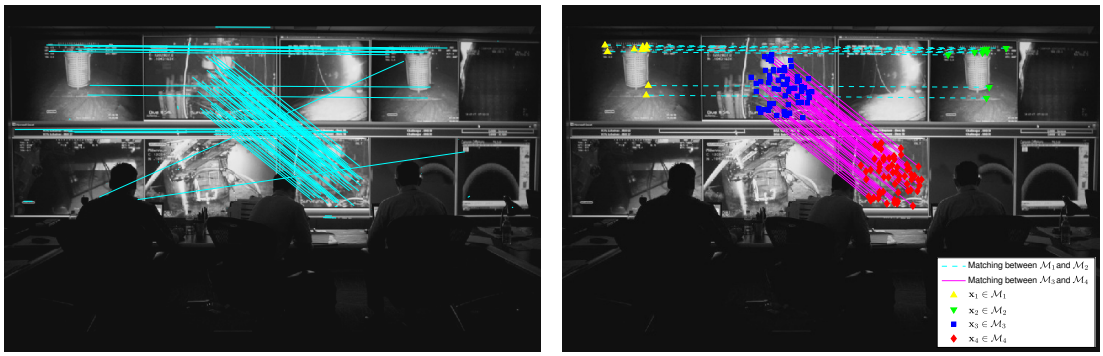
$$\mathcal{S}_i = \{\|\mathbf{d}_i - \mathbf{d}_j\|_2 : j = 0, \ldots, N-1,\ j \neq i\},$$

whose elements will be sorted in ascending order, for convenience. The matching between a keypoint $\mathbf{x}_i$ and any other keypoint $\mathbf{x}_j$ (with $j \neq i$) is satisfied when the ratio between the distance of the closest neighbor in $\mathcal{S}_i$, i.e., $s_0^{(i)}$, and that of the second-closest one, i.e., $s_1^{(i)}$, is less than a threshold Υ:

$$\frac{s_0^{(i)}}{s_1^{(i)}} < \Upsilon.$$

As an example, employing a threshold $\Upsilon = 0.6$ and applying this procedure to the BP tampered image shown in Figure 2.3(b), we get the result depicted in Figure 2.6(a).

Once the subset of matched keypoints, i.e., $\mathcal{X}_m$, from $\mathcal{X}$ is obtained, it is necessary to cluster these data so as to enable the distinction between the different



(a) Matched keypoints (b) Clustered and detected regions

Figure 2.6: Followed steps for the detection of cloned areas. On the left side, solid lines represent the matching between keypoints, while on the right side, different markers are used to identify the clustered data and solid/dashed lines link the related regions.

matched regions. For clustering on the spatial location of the matched points, an agglomerative hierarchical clustering is used as proposed in [60]. Assuming that we have at least two matched areas, the result of this process provides $T \geq 2$ different sets of matched points $\mathcal{M}_t$ with $t = 1, \ldots, T$, so $\mathcal{X}_m = \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_T$, and this serves to define the different duplicated regions. Continuing with the BP doctored image, we illustrate in Figure 2.6(b) the four sets of points that determine the two different tampered regions linked by solid and dashed lines, respectively. Note that some outliers have been removed after the clustering process.

From the points in a certain region $\mathbf{x}_q \in \mathcal{M}_q$ and the corresponding matched points $\mathbf{x}_r \in \mathcal{M}_r$, we can estimate the geometric transformation applied between the two matched areas:

$$\begin{pmatrix} \mathbf{x}_q^T \\ 1 \end{pmatrix} = \mathbf{H}_{qr} \begin{pmatrix} \mathbf{x}_r^T \\ 1 \end{pmatrix},$$

where $\mathbf{H}_{qr}$ represents an affine homography. By using the RAndom SAmple Consensus (RANSAC) algorithm, a maximum likelihood estimation of the affine homography $\mathbf{H}_{qr}$ can be carried out.

Now, suppose that from the SIFT-based method we have obtained $T = 2$ matching regions (denoted by the sets of points $\mathcal{M}_1$ and $\mathcal{M}_2$) and also an estimate of the relation between both $\hat{\mathbf{H}}_{12}$. Then, using this information, we still cannot demonstrate whether the points in $\mathcal{M}_1$ correspond to the original area and those on $\mathcal{M}_2$ to the duplicated one, or vice-versa. However, the method explained below will help to answer this question.

2.5.3.2. A resampling-based method to reveal tampered regions

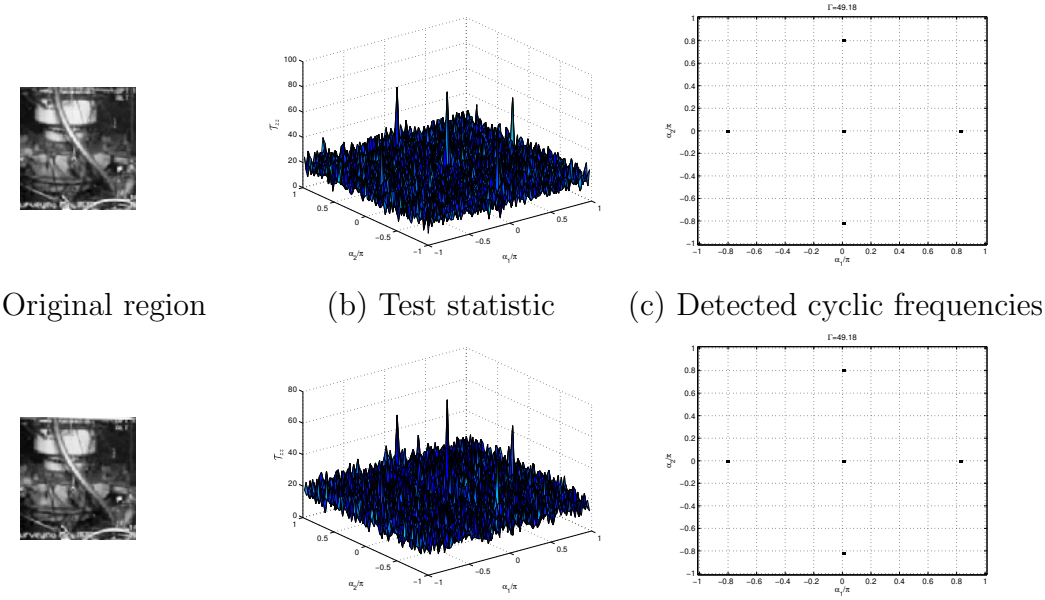
An appropriate way to determine whether a matched region is the source or the duplicate is to use a resampling estimator that gives a measure of the existing linear correlations among the pixels of such region. If the SIFT-based method is not capable of finding a duplicated region, then we can use any of the proposed methods in Section 1.2.3 to make an exhaustive analysis, processing all the blocks of the image and looking for inconsistencies in the resampling traces.

However, we are more interested in the case where the SIFT-based method does provide the detected cloned regions. So, assuming two matching regions, described by the sets of points $\mathcal{M}_1$ and $\mathcal{M}_2$, we will use the resampling-based method described in Section 2.3, for the identification of the original region and the duplicate. As previously indicated, this method takes a block $\mathbf{Z}$ of an image and applies a statistical test for the evaluation of the presence of almost cyclostationarity. The main steps of this test are summarized at the end of Section 2.3.

Given that the resulting regions from the SIFT-based method are generally non-square and the resampling-based method only works with square blocks, we have to adapt the detected regions to a square support. A simple way to do that is to take a square block $\mathbf{Z}$ that gathers all the pixels included within the contour of a set of points, i.e., $\mathcal{M}_1$ or $\mathcal{M}_2$, and pad with zeros the remaining elements of matrix $\mathbf{Z}$. The zero-padding approach is probably a suboptimal solution, but doing this with each set of points we can estimate the resampling factor for each region. One of the objectives of this work is also to study the performance of the resampling-based method in such scenario.

As we have stated before, a resampling detector cannot differentiate the original source from its duplicated versions if a copy-move forgery is performed without content adjustment. That is exactly what happens with the tampered regions, labeled as $\mathcal{M}_2$ and $\mathcal{M}_4$ in Figure 2.6(b). In fact, applying the statistical test to the matched regions $\mathcal{M}_3$ and $\mathcal{M}_4$, we obtain the same resampling factor ($\hat{\xi} \approx 5/3$) in both cases, as we can see in Figure 2.7. Thus, in this particular scenario, the resampling-based method only identifies the scaling factor applied to the whole image without distinguishing the source region from the clone (since the resampling factor is the same).

However, we will make the hypothesis that most of the time the pasted regions must be geometrically adapted to the scene. Therefore, to determine which parts of the image are copies and which parts are their sources, we have to analyze the neighborhood of each region. By taking square blocks that only include neighboring pixels of each region, i.e., $\mathcal{M}_1$ or $\mathcal{M}_2$, and removing the pixels that belong to the area under analysis, the resampling factor of the neighborhood can be estimated. Finally, for classifying each region, we know that an original region will have the same resampling factor in the neighborhood and inside the region,



(a) Original region (b) Test statistic (c) Detected cyclic frequencies

(d) Duplicated region (e) Test statistic (f) Detected cyclic frequencies

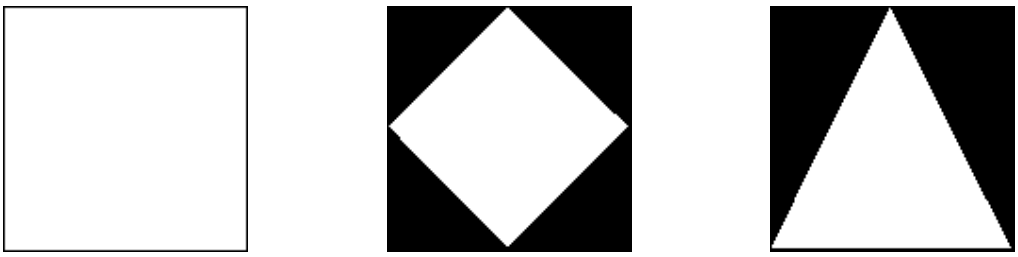
Figure 2.7: Application of the two-dimensional statistical test to one of the pair of matched regions in the BP image: $\mathcal{M}_3$ (top row) and $\mathcal{M}_4$ (bottom row).

while the tampered regions will reach different values in each part.

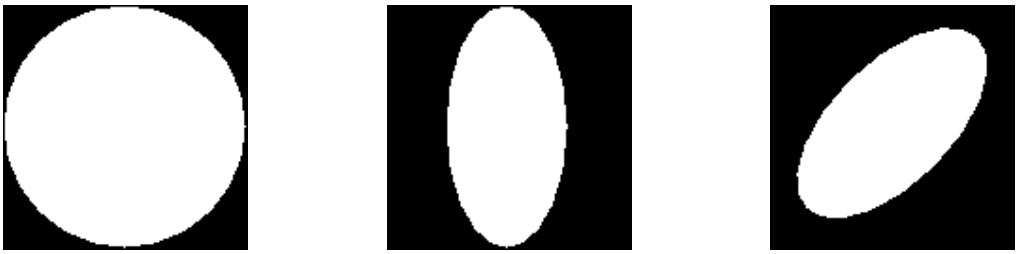
2.5.4. Experimental Results

For the evaluation of this enhanced copy-move detector with resampling estimation, we use 100 images from a personal image database composed by several realistic scenarios with different indoor and outdoor scenes. All the images in this collection have been captured in a raw format by a Nikon D60 digital camera and have been converted into uncompressed TIFF images in the RGB color space. The original resolution of each image was 3872×2592, but for the sake of reducing the computational complexity, all the images were cropped to the center block of 1024×1024 pixels. The resampling factor of each color channel is equal to 2, due to the Color Filter Array (CFA) interpolation performed inside the camera. This fact will be taken into consideration along the application of the resampling-based method and for simplicity we will only process the green component of the RGB color space.

To test the performance of the proposed scheme (Figure 2.5), as a first step we evaluate the SIFT-based method and the resampling-based method separately, combining them later to see how the results of the forensic analysis improve. In order to get more realistic forgeries in our experiments, six different patterns like the ones depicted in Figure 2.8 are used for shaping the duplicated areas. These



(a) Square (b) Rotated square (c) Triangle



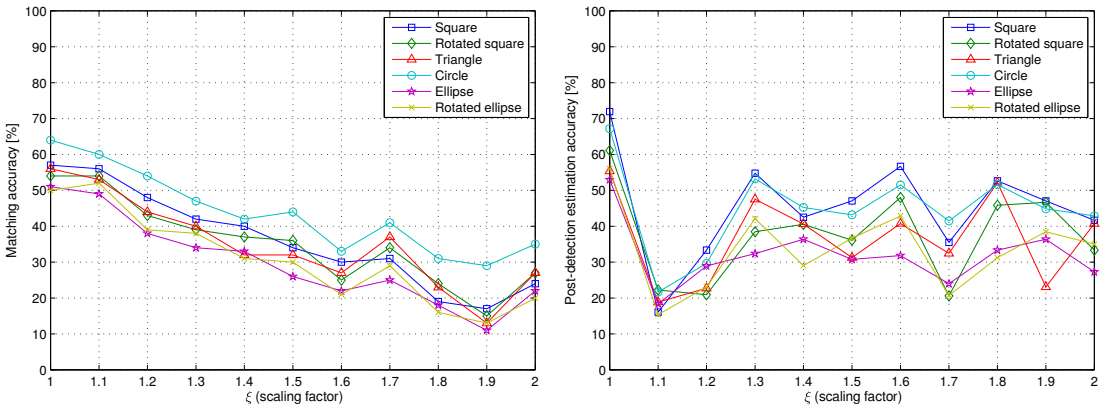
(d) Circle (e) Ellipse (f) Rotated ellipse

Figure 2.8: Different masks used to test the performance of the proposed forensic tool.

masks are initially exploited to copy a region located at a random position of each image and, subsequently, the copied region is uniformly scaled by one of the scaling factors ξ in the set $\{1, 1.1, 1.2, \ldots, 2\}$. Finally, the scaled duplicate of the original region is pasted into a distinct random location within the same image. In order to make a fair comparison, we use the same random location for testing all the considered masks. However, for each new image or scaling factor under test, a different random position is generated. Since the tampered regions tend to be relatively small, we have conducted the experiments in such a way that the resampled region fits always in a 128×128 block.

For the SIFT-based method we fix the threshold Υ at 0.6 and we remove false positive matching keypoints whenever their distance is smaller than 10 (i.e., $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 < 10$ for any $j \neq i$). Once the hierarchical clustering has been completed, the outliers of each region are removed. A keypoint is deemed as an outlier when the distance between the keypoint and the mean point of its cluster is higher than 3 times the variance of the points in the associated cluster. The implementation of the SIFT algorithm used in the following experiments has been taken from [65] and for the RANSAC homography estimation the functions available from [66] have been used.

The configuration of the resampling-based method is almost the same as the one used in Section 2.4 or in [67] (i.e., we use a spectral window to smooth the periodogram of size 11×11 and a set of $K = 9$ lags), but we do not use the threshold Γ to detect the cyclic frequencies. For the sake of simplicity, we only estimate the applied transformation (i.e., the scaling factor) from the cyclic



(a) SIFT-based method (b) SIFT-based method

Figure 2.9: Matching accuracy and post-detection estimation accuracy (in terms of percentage), obtained with the SIFT-based method for different masks and scaling factors.

frequency with largest magnitude, excluding the frequency at zero (DC).

Detection results using the SIFT-based method

Taking into account the described set of tampered images, we ascertain a correct matching of a forged area whenever the SIFT-based method is able to find at least four common points between the original and the possibly duplicated region. Figure 2.9(a) depicts the matching accuracy of this method in terms of percentage, showing the different results for each used mask and for the different values of the scaling factor ξ.

Next to this graph, Figure 2.9(b) shows the (post-detection) estimation accuracy of the affine transformation applied between the previously matched regions, using the RANSAC method. Note that we are plotting the post-detection estimation accuracy, i.e., the estimation accuracy of the scaling factor applied between the correctly matched regions in the previous step (thus, it is clear that the represented percentage of accurate estimation is not relative to the 100 images of the database). In this case, we cannot know which region is the original, so we get two possible estimations: $\hat{\mathbf{H}}_{12} \approx \mathbf{H}_{12}$ or $\hat{\mathbf{H}}_{12} \approx \mathbf{H}_{12}^{-1}$. On the other hand, denoting each (i,j)-th element of matrix $\hat{\mathbf{H}}_{12}$ by $\hat{\mathbf{H}}_{12}(i,j)$ (with $i,j \in \{0,1\}$), it is clear that the elements $\hat{\mathbf{H}}_{12}(0,0)$ and $\hat{\mathbf{H}}_{12}(1,1)$ represent the estimation of the scaling factor applied to each axis of the affine transformation. Thus, finally, a correct estimation is declared when either $|\hat{\mathbf{H}}_{12}(0,0) - \xi| < 0.05$ or $|\hat{\mathbf{H}}_{12}(1,1) - \xi| < 0.05$.

As it can be observed from the two graphs of Figure 2.9, with the SIFT-based method it is easier to match and estimate copy-move forgeries without content adaptation than duplicated regions that have been geometrically transformed.

However, from the estimation point of view, it is more difficult to estimate the homography for scaling factors near one like $\xi = 1.1$ or $\xi = 1.2$, than for higher values. The matching accuracy is not very high, due to the lack of reliable keypoints in several images of the dataset (the number of keypoints per image was in the range $[250, 17500]$), but, as it was mentioned earlier, this is an intrinsic limitation of any SIFT-based method. With respect to the used masks, the intuitive idea that small areas are more challenging for detection and estimation purposes, comes up in both plots.

Detection results using the resampling-based method

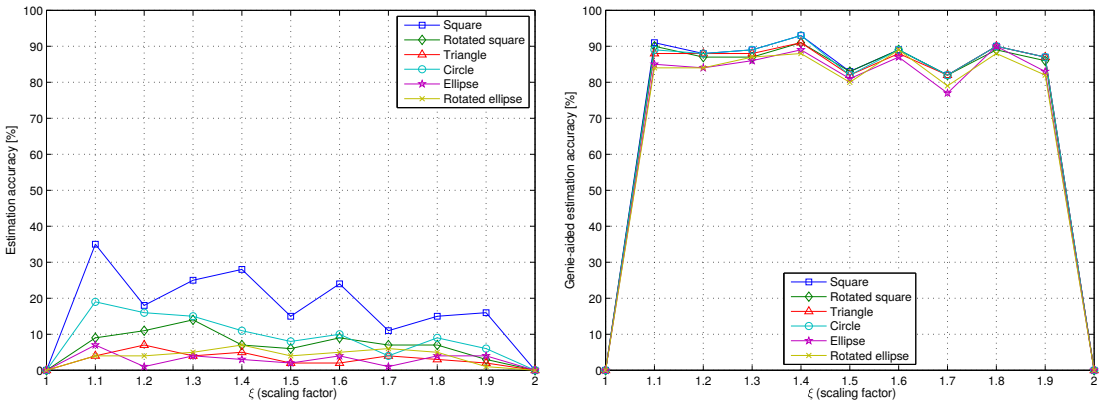
Before considering the combination of the two methods, we evaluate the resampling-based one when it is applied to the whole image, following a block-by-block procedure to find inconsistencies in the resampling factor ξ. As it was previously noticed, due to the CFA interpolation applied by the camera, we know that the real resampling factor, which will be denoted by ξ_{real}, of each non-scaled block is actually $\xi_{\text{real}} = 2 \times 1$ (instead of being equal to 1), and so the corresponding value of a scaled version by ξ will be $\xi_{\text{real}} = 2 \times \xi$. Therefore, every time we get an estimated resampling factor with a different value from 2 we tag the block under analysis as a digitally forged region. In this case, because the tampered regions have a similar size, we use a block $\mathbf{Z}$ of size 128×128 pixels for the analysis.

The classification of every single block is performed by analyzing the test statistic $\mathcal{T}_{zz}$. As we have said at the beginning of Section 2.5.4, the resampling factor is estimated from the cyclic frequency (α_1, α_2) with largest magnitude (excluding DC), and using the following relation:

$$\hat{\xi}_{\text{real}} = \max_{i \in \{1,2\}} \hat{\xi}_i = \max_{i \in \{1,2\}} \frac{2\pi}{|\alpha_i|},$$

where we have exploited the fact that in this case, $\xi_{\text{real}} \geq 2$, since $1 \leq \xi \leq 2$. We confirm that the detection of the tampered region is correct if any inconsistency in the resampling factor is discovered (i.e., whether $\hat{\xi}_{\text{real}} \neq 2$) and the resulting estimation $\hat{\xi}_{\text{real}}$ satisfies $|\hat{\xi}_{\text{real}} - 2\xi| < 0.05$, or since in some cases the interference created by the CFA pattern may not be so strong, we will also check if $|\hat{\xi}_{\text{real}}/(\hat{\xi}_{\text{real}} - 1) - \xi| < 0.05$ is satisfied.

Applying this approach to the tampered images of the database, we obtain the results shown in Figure 2.10(a). As we have stated before, this method cannot detect copy-move forgeries without content adjustment, since there are no inconsistencies in the resampling factor along the whole image. That is the reason why the estimation accuracy is equal to zero at $\xi = 1$. Given the ambiguity created by the estimation, caused by the CFA pattern, we are not able to distinguish between a scaling factor $\xi = 1$ or $\xi = 2$, and that is why the estimation accuracy



(a) Processing block by block (b) Using exact matching (genie-aided)

Figure 2.10: Estimation accuracy (in terms of percentage), obtained through the application of the resampling-based method in two different scenarios for several masks and scaling factors.

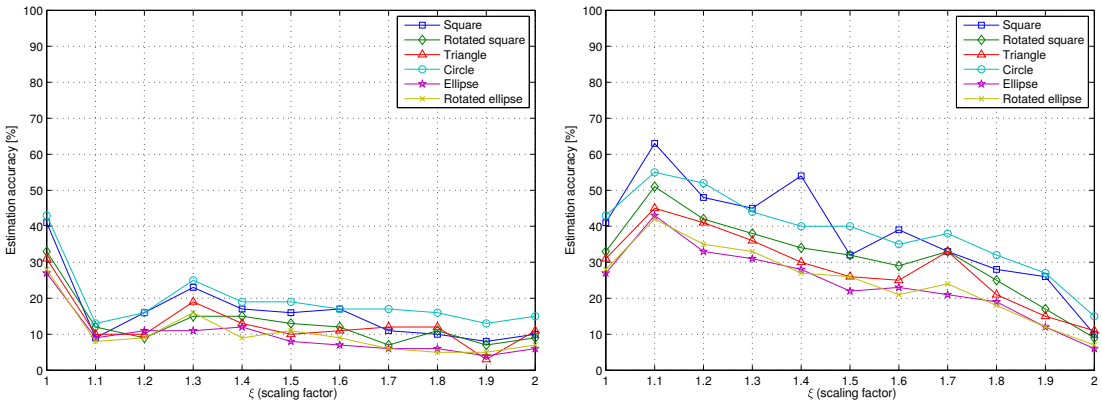
is also zero for $\xi = 2$. The rate of accurate estimation of the tampered region is not very high for any of the used masks (in the best case we barely reach a 35%), so this method presents very bad performance when identifying forgeries through this block-by-block procedure.

Nevertheless, to demonstrate the generally good performance of the resampling estimator, we analyze the estimation accuracy in an ideal case where we use the information supplied by a *genie* that tells us exactly the location of the original region and that of the tampered region (the application of a "genie-aided" detection is commonly used in communications to determine performance bounds). Thus, knowing exactly the location of both regions in the pixel domain and using the same criteria for the estimation of ξ_{real}, as in the previous scenario, we show in Figure 2.10(b) the results of the correct identification of which region is the original and which is the duplicate. As it was said before, the correct distinction of the two regions when a spatial transformation has not been applied is not possible with the resampling-based method. However, the detection performance is very high (around a 90% for all the masks) if we compare it with that obtained when the image is processed block by block.

So, according to the results obtained in this ideal case, the problem does not lie in the resampling estimator itself, but in the correct matching of the tampered area, and that is the reason why a SIFT-based method is needed.

Detection results combining both methods

Along Section 2.5 we have been discussing that the combination of both methods provides a deeper and enhanced forensic analysis of the tampered regions



(a) SIFT-based method (b) Proposed forensic tool

Figure 2.11: Comparative results of the estimation accuracy for the SIFT-based method and the proposed forensic tool.

(since we are able to identify which region is the source and which is the duplicated one) and it also brings a way to compensate the drawbacks of each method with the advantages of the other.

Certainly, given that the SIFT-based method is not capable of finding all the duplicated regions, mostly due to the unavoidable lack of reliable keypoints, combining both approaches we would get worse results than those depicted in Figure 2.10(b) (i.e., the ideal "genie-aided" case where we perfectly match all the regions). Nonetheless, with the use of the SIFT-based method, the detection of the tampered regions is more accurate than processing the image block by block, so we will get better results than those included in Figure 2.10(a). Finally, since the estimation of the resampling factor is not so dependent on outliers as it is the case for the estimate of the homography, we will also get better results than those comprised in Figure 2.11(a), where we represent the estimation accuracy of the SIFT-based method when it is able to jointly match the two regions and correctly estimate their geometric relation. Explicitly, the estimation accuracy plotted in Figure 2.11(a), corresponds to the product of the accuracy rates achieved in the matching step (Figure 2.9(a)) and in the post-detection estimation step (Figure 2.9(b)).

In Figure 2.11(b) we can see the reached estimation accuracy of the combined forensic tool for different masks and scaling factors. If we compare this plot with the corresponding estimation accuracy obtained with the SIFT-based method alone (depicted in Figure 2.11(a)), we can observe that with the scheme described in Figure 2.5, the performance is improved for almost all the scaling factors and masks considered. It is important to note that the resampling estimator takes as input the exact matching of the detected regions by the SIFT-based method, so the provided results can be viewed as an upper bound on the performance in terms of estimation accuracy that can be attained with this approach.

Note also that, even with the combination of both methods, we are still not able to distinguish the original region from the tampered one when a copy-move forgery without content adaptation is carried out. Besides, in this particular case, occasioned by the CFA interpolation of the camera, we are neither able to identify regions duplicated with a factor $\xi = 2$. Hence, the estimation accuracy should be strictly zero for the scaling factors $\xi = 1$ and $\xi = 2$ in Figure 2.11(b). However, since with the SIFT-based method we are able to match the involved regions in the tampering and also their relation, then we add the estimation accuracy of this method in both cases, and that is the reason why we have the same values of estimation accuracy for the scaling factors $\xi = 1$ and $\xi = 2$ in both graphs of Figure 2.11.

By comparing the estimation accuracy of the resampling-based method (processing block by block) with that obtained with the concatenation of both methods, we achieve an important improvement in the exact classification of each region for all the scaling factors and distinct masks. In addition, as it was expected, the best results are achieved with those masks that cover the largest area of the block under analysis.

According to the results shown in this section, we can conclude that the proposed practical solution provides a more accurate forensic analysis since we can identify in an image where and which are the original regions and the tampered ones when a region duplication forgery is performed. Moreover, the performance in terms of estimation accuracy is increased with respect to the sole use of either the SIFT-based or resampling-based methods.

2.6. Conclusions

In the first part of this chapter, we have proposed a method to estimate the scaling factor and the rotation angle of spatially transformed images that performs better than that in [11]. Within a cyclostationarity framework, the resampling estimation problem is addressed by extending existing concepts to the two-dimensional case for dealing with images. Although the resulting approach is more time consuming than that in [11] (mainly due to the processing in the two-dimensional space), the proposed estimate circumvent several ambiguities caused by indistinguishable periodic patterns in the one-dimensional case.

In the second part of the chapter, we have introduced a new scheme for image forensic analysis by combining two complementary methods. The former, based on SIFT, is capable of finding duplicated regions and the latter, based on a resampling estimator such as the one above proposed, enables the identification of which region is the source and which is the tampered one. The proposed scheme provides better estimation results than considering each method separately.

Chapter 3

Prefilter Design for Forensic Resampling Estimation

Starting from a theoretical analysis of the resampling estimation problem for image tampering detection, this chapter presents a study, based on the previously introduced cyclostationarity theory, about the use of prefilters to improve the estimation accuracy of the resampling factor. Focusing on the methods that perform the estimation by analyzing the spectrum of the covariance of a resampled region, we propose an analytical framework that allows the definition of a cost function which measures the degree of detectability of the spectral peaks. Based on this measure, the design of the optimum prefilters for a particular resampling factor can be solved numerically. Experimental results validate the developed analysis and illustrate the enhancement of the performance in a real scenario.

3.1. Introduction

Even though today anyone can simply manipulate the information represented by a picture without leaving perceptual traces, we have remarked in Chapter 1 that the subsequent change introduced in the intrinsic properties of the image may enable the detection of such alterations. For instance, the application of a geometric transformation to a portion of an image (which is often required to adapt a new content to the captured scene) modifies the original sampling grid of this region, producing resampling traces that can be detected, as explained in Section 1.2. Moreover, the characteristic evolution of these traces along the content also makes possible the estimation of the transformation locally applied as described in Chapter 2.

Although different approaches are contemplated to detect these resampling traces and estimate the applied transformation (cf. Section 1.2.3), the vast ma-

jority of the proposed methods work, at some point, in the frequency domain to finally detect or estimate the periodicities that are left behind by the resampling process. Specifically, in [10, 11, 67, 68], the spectrum of the covariance of the resampled blocks is computed to detect the frequency peaks that allow the estimation of the applied spatial transformation. Derivative filters are used in these resampling-based methods as a way to substantially enhance the spectral lines and thus to improve the estimation performance.

Since the use of certain derivative prefilters increases the estimation accuracy of the applied resampling factor (as featured in Section 2.4 with the Laplacian operator), the question of whether there exist other prefilters yielding better results becomes very relevant. Dalgaard et al. took the first step in that direction by analytically showing that for asymptotically large values of the resampling factor, the use of derivative filters enhances the detection of the resampling traces [12]. Nevertheless, in order to avoid visible distortions in the forged image due to the spatial transformation, the resampling factor is usually close to 1 and rarely larger than 2, so the hypothesis of an asymptotically large value of this factor does not hold in a realistic scenario. Although resampling factors smaller than 1 are also commonly used, we do not tackle their analysis in this chapter. Bearing this in mind, our main goal is to present an analytical framework that supports the definition of a cost function which gives a measure of the detectability of resampling traces for resampling factors within the interval $(1, 2)$. Using this criterion, we study different prefilters and compute numerically their performance in the mentioned range of resampling factors, so as to reach the optimum prefilter for each factor. Using a database of real images, we also provide empirical results to endorse our analysis.

The chapter is organized as follows. The next section introduces the bases of the exposed problem following a one-dimensional (1-D) characterization of the resampling process. The description of the model used for natural images and the Fourier analysis for the detection of resampling traces is presented in Section 3.3. The design of the prefilters is then addressed in Section 3.4 and the evaluation of the resulting prefilters with real images is carried out in Section 3.5. Finally, Section 3.6 concludes the chapter.

3.2. Preliminaries and Problem Statement

In the first part of Chapter 2, we have seen that the application of geometric transformations on images introduces periodically correlated fields in the two-dimensional (2-D) space that make possible the detection of the applied resampling process (cf. Section 2.2.2). However, in light of the associated complexity that implies the analysis of periodic correlations in the 2-D case, in this chapter, we tackle the frequency analysis using a simplified model in the 1-D space. On

the other hand, contrary to the asymptotic analysis carried out in [12], we are more interested in the design of prefilters in the range $1 < \xi < 2$ since, as we have stated above, in a real scenario the tampered regions are only slightly rotated or minimally scaled to mitigate the insertion of visual distortions.

Along the description of the 2-D resampling process in Section 1.2.2, it has been assumed that the specified kernels are of separable nature (which are the most commonly available), such that the interpolation filter is separately applied along each single dimension of the image. In other words, a 1-D resampling process is first performed over each row of the image and then the same procedure is followed along each column, or vice versa. Therefore, without loss of generality, we adhere to the 1-D resampling model for the sake of simplicity, and as just noted, its extension to the 2-D case is rather straightforward. In the following, we introduce the simplified model of the resampling process to afford later a more tractable design of prefilters.

The general case of sampling rate conversion of a 1-D input signal $x(n)$ by a factor $\xi \triangleq \frac{L}{M}$ (with L and M integer values and relatively primes), is carried out by first performing interpolation by the factor L and then decimating the output of the interpolator by the factor M. The resulting resampled signal $y(n)$, using any interpolation filter $h(t)$, can be expressed as:

$$y(n) = \sum_k x(k) h\left(n\tfrac{M}{L} - k\right), \tag{3.1}$$

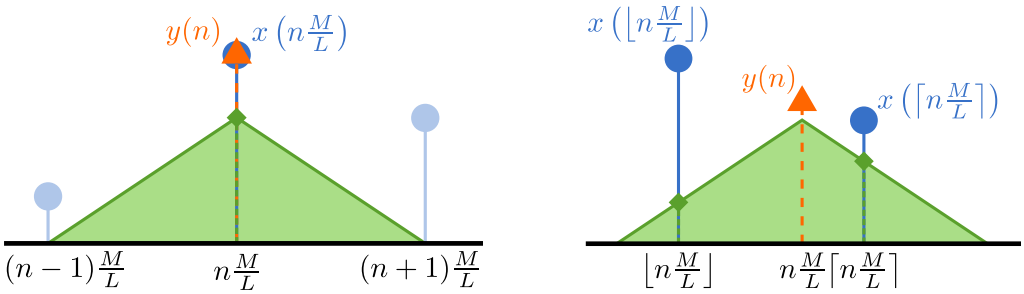
where $\frac{M}{L} = \xi^{-1}$ represents the interval between samples in the resampled signal and no shift has been applied between the two sampling grids. The interpolation filter used to preserve the desired spectral characteristics of the input signal $x(n)$ can be any of those gathered in Table 1.1, i.e., linear, cubic or a truncated sinc; but, in this case, with the aim of having a simplified model, we choose the linear filter

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

Taking as reference the illustration shown in Figure 3.1, the expression of the resampled signal in (3.1) considering the above linear filter can be formulated as follows:

$$\begin{aligned} y(n) &= \begin{cases} x\left(\lfloor n\tfrac{M}{L} \rfloor\right) h\left(n\tfrac{M}{L} - \lfloor n\tfrac{M}{L} \rfloor\right) + x\left(\lceil n\tfrac{M}{L} \rceil\right) h\left(n\tfrac{M}{L} - \lceil n\tfrac{M}{L} \rceil\right), & \text{if } n\tfrac{M}{L} \notin \mathbb{Z} \\ x\left(n\tfrac{M}{L}\right), & \text{if } n\tfrac{M}{L} \in \mathbb{Z} \end{cases} \\ &= x\left(\lfloor n\tfrac{M}{L} \rfloor\right)\left(1 - \operatorname{mod}\left(n\tfrac{M}{L}, 1\right)\right) + x\left(\lceil n\tfrac{M}{L} \rceil\right) \operatorname{mod}\left(n\tfrac{M}{L}, 1\right). \end{aligned} \tag{3.2}$$

Assuming that the input signal $x(n)$ is zero-mean, the covariance of the resampled signal corresponds to the correlation $c_{yy}(n;\tau) = E\{y(n)y(n+\tau)\}$. Taking into account the simplified version of $y(n)$ in (3.2) and using $v(n) \triangleq \operatorname{mod}\left(n\frac{M}{L}, 1\right)$, we



(a) $n\frac{M}{L} \in \mathbb{Z}$ (b) $n\frac{M}{L} \notin \mathbb{Z}$

Figure 3.1: Illustrative example of the two cases that show up when performing a piecewise linear interpolation.

get

$$\begin{aligned} c_{yy}(n;\tau) = {} & E\left\{x\left(\lfloor n\tfrac{M}{L}\rfloor\right) x\left(\lfloor (n+\tau)\tfrac{M}{L}\rfloor\right)\right\}(1-v(n))\,(1-v(n+\tau)) \\ & + E\left\{x\left(\lfloor n\tfrac{M}{L}\rfloor\right) x\left(\lceil (n+\tau)\tfrac{M}{L}\rceil\right)\right\}(1-v(n))\,v(n+\tau) \\ & + E\left\{x\left(\lceil n\tfrac{M}{L}\rceil\right) x\left(\lfloor (n+\tau)\tfrac{M}{L}\rfloor\right)\right\}v(n)\,(1-v(n+\tau)) \\ & + E\left\{x\left(\lceil n\tfrac{M}{L}\rceil\right) x\left(\lceil (n+\tau)\tfrac{M}{L}\rceil\right)\right\}v(n)v(n+\tau), \end{aligned} \tag{3.3}$$

that represents the general expression of the correlation of a zero-mean signal interpolated by a linear filter.

In order to determine whether the resampled signal $y(n)$ is (wide-sense) cyclostationary, we have to check if the above expression (3.3) varies periodically. As indicated in the previous chapter, Sathe and Vaidyanathan showed in [57] that the resampled signal is cyclostationary with period $L/\mathrm{GCD}(L,M)$ when the input signal $x(n)$ is wide-sense stationary and the interpolation filter is not ideal. Note that, in this case, L and M are coprime so $\mathrm{GCD}(L,M)=1$, and consequently this is equivalent to saying that the resampled signal $y(n)$ is a cyclostationary process of period L whenever $x(n)$ is wide-sense stationary and the interpolator is not ideal (which is actually the case with a linear kernel).

Moreover, we can generalize this property by proving that the resampled signal is (wide-sense) almost cyclostationary if the above expression satisfies $c_{yy}(n;\tau) = c_{yy}\left(n+k\frac{L}{M};\tau\right)$ for some $k \in \mathbb{Z}$. To demonstrate that, we have to show that $v(n)$ is periodic and also that the four terms within expectations $E\{\cdot\}$ in (3.3) are periodic. Accordingly, starting with the signal $v(n)$, it is easy to see that:

$$\begin{aligned} v\left(n+k\tfrac{L}{M}\right) &= \mathrm{mod}\left(\left(n+k\tfrac{L}{M}\right)\tfrac{M}{L},1\right) \\ &= \mathrm{mod}\left(n\tfrac{M}{L}+k,1\right) \\ &= \mathrm{mod}(n\tfrac{M}{L},1) \\ &= v(n). \end{aligned}$$

Regarding the expectation term $E\left\{x\left(\lfloor n\frac{M}{L}\rfloor\right) x\left(\lfloor (n+\tau)\frac{M}{L}\rfloor\right)\right\}$, by taking into account that $x(n)$ is wide-sense stationary, we know that this expression depends

only on the difference between $\lfloor n\frac{M}{L} \rfloor$ and $\lfloor (n+\tau)\frac{M}{L} \rfloor$ and such difference has to be cyclic with period $\frac{L}{M}$, i.e.,

$$\begin{aligned}\left\lfloor \left(n + k\tfrac{L}{M}\right)\tfrac{M}{L} \right\rfloor - \left\lfloor \left(n + k\tfrac{L}{M} + \tau\right)\tfrac{M}{L} \right\rfloor &= \left\lfloor n\tfrac{M}{L} + k \right\rfloor - \left\lfloor (n+\tau)\tfrac{M}{L} + k \right\rfloor \\ &= (n\tfrac{M}{L} + k) - \mathrm{mod}(n\tfrac{M}{L} + k, 1) \\ &\quad - ((n+\tau)\tfrac{M}{L} + k) + \mathrm{mod}((n+\tau)\tfrac{M}{L} + k, 1) \\ &= -\mathrm{mod}(n\tfrac{M}{L}, 1) - \tau\tfrac{M}{L} + \mathrm{mod}((n+\tau)\tfrac{M}{L}, 1) \\ &= \left\lfloor n\tfrac{M}{L} \right\rfloor - \left\lfloor (n+\tau)\tfrac{M}{L} \right\rfloor,\end{aligned}$$

where we have used the relation:

$$\left\lfloor n\tfrac{M}{L} \right\rfloor = n\tfrac{M}{L} - \mathrm{mod}(n\tfrac{M}{L}, 1). \tag{3.4}$$

The same applies for the other three expectation terms in (3.3), where additionally we have to use that

$$\left\lceil n\tfrac{M}{L} \right\rceil = n\tfrac{M}{L} + \mathrm{mod}(-n\tfrac{M}{L}, 1). \tag{3.5}$$

Therefore, since $c_{yy}(n;\tau)$ is cyclic with an almost-integer period $\frac{L}{M}$, we can conclude that if the input signal $x(n)$ is wide-sense stationary then the resampled signal $y(n)$ will be almost cyclostationary.

Several works [10, 11, 67] have noticed this periodicity considering a random i.i.d. signal as input, but in this case we are generalizing this fact for any wide-sense stationary input signal and a linear interpolator. Our main goal is to analytically characterize the correlation of the resampled signal in the frequency domain, since the estimation of the resampling factor is performed in such domain through the detection of the cyclic frequencies.

Inasmuch as the study of the cyclic correlation is tackled in the Fourier domain, it is apparent that a Gaussian white-noise signal will not lead to an accurate model for a natural image, so we need a model that better captures the local correlation of natural images. For this reason, we propose to use a 1-D autoregressive (AR) process of the first order which provides a good fit to the power spectral density of real images (cf. Section 2.10 in [69]). Next section describes the used model and the Fourier analysis carried out that will lead us to the design of the optimum prefilter for resampling estimation.

3.3. Model Description and Fourier Analysis

Since a white noise process is not very representative of a non-compressed natural image, we use a more convenient approximation that corresponds to a first-order AR process with a correlation coefficient ρ that satisfies $|\rho| < 1$. The value of ρ enables the adjustment of the model as necessary. For instance, values

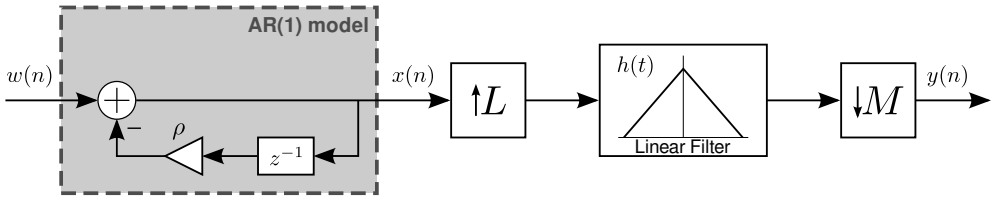


Figure 3.2: Block diagram representation of the complete image resampling model.

of ρ near 1 (e.g., $\rho = 0.95$) can be employed to model the power spectral density of natural images, while for values of ρ near zero behave like a white noise process, and near -1 (e.g., $\rho = -0.95$) could represent synthetic images with high frequency content [70].

As illustrated in Figure 3.2, the resampling process takes now as input signal $x(n)$ a sequence generated by a first-order AR model with parameter ρ, such that

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

where $w(n)$ is a white noise process with zero mean and unit variance. It is easy to check that the input signal is a zero-mean process, whose correlation is given by

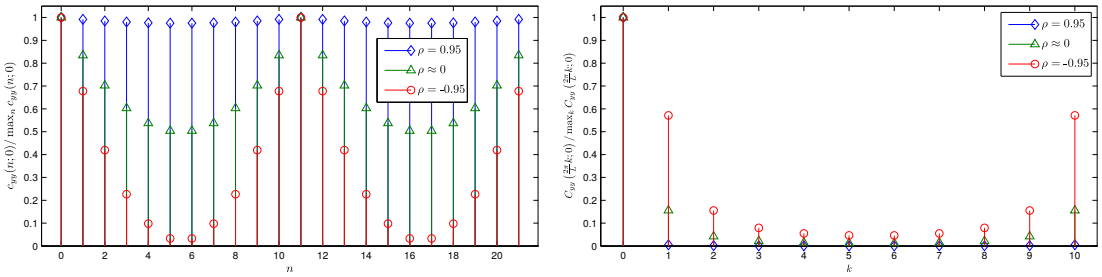
$$c_{xx}(n;\tau) = E\{x(n)x(n+\tau)\} = \frac{\rho^{|\tau|}}{1-\rho^2}. \tag{3.6}$$

The resulting correlation of the resampled signal $y(n)$, can be directly obtained by combining (3.3) and (3.6), yielding

$$\begin{aligned} c_{yy}(n;\tau) = \frac{1}{1-\rho^2} \Big[& \rho^{\left|\lfloor n\frac{M}{L}\rfloor - \lfloor (n+\tau)\frac{M}{L}\rfloor\right|} (1-v(n))\,(1-v(n+\tau)) \\ & + \rho^{\left|\lfloor n\frac{M}{L}\rfloor - \lceil (n+\tau)\frac{M}{L}\rceil\right|} (1-v(n))\,v(n+\tau) \\ & + \rho^{\left|\lceil n\frac{M}{L}\rceil - \lfloor (n+\tau)\frac{M}{L}\rfloor\right|} v(n)\,(1-v(n+\tau)) \\ & + \rho^{\left|\lceil n\frac{M}{L}\rceil - \lceil (n+\tau)\frac{M}{L}\rceil\right|} v(n)v(n+\tau) \Big]. \end{aligned} \tag{3.7}$$

Given that $x(n)$ is wide-sense stationary, we know from the previous analysis that the resampled signal $y(n)$ is almost cyclostationary with period $\frac{L}{M}$, while if we only assume the existence of pure cyclostationary processes, then $y(n)$ is cyclostationary with period L.

Figure 3.3(a) shows an example of the normalized version of $c_{yy}(n;\tau)|_{\tau=0}$ for $\xi = \frac{11}{10}$ and different values of ρ. Two periods of size $L = 11$ are represented and, as we can see, the periodicity becomes apparent for $\rho = -0.95$ and also for $\rho \approx 0$, whereas for $\rho = 0.95$ the correlation of the resampled signal seems to be constant. From this example, it can be inferred that the estimation in the frequency domain of the resampling factor for an AR process with $\rho = 0.95$ (i.e., natural images)



(a) Correlation $c_{yy}(n;\tau)|_{\tau=0}$ (b) Cyclic correlation $C_{yy}(\frac{2\pi}{L}k;\tau)|_{\tau=0}$

Figure 3.3: Normalized version of the correlation and cyclic correlation of the resampled signal $y(n)$ for $\xi = \frac{11}{10}$, $\tau = 0$ and different values of ρ.

will be more challenging than for $\rho = 0$ or $\rho = -0.95$ (i.e., synthetic images). In order to study the complexity of finding the resampling traces, we have to analyze the correlation in the frequency domain.

In view of the fact that the correlation $c_{yy}(n;\tau)$ is periodic over n with period L, such signal can be written in terms of a Fourier Series expansion whose spectral coefficients are $C_{yy}(k;\tau)$ with $k \in \{0,\dots,L-1\}$. The development of a closed-form expression is not straightforward, but we can derive the spectral coefficients of (3.7), by determining the Discrete-Time Fourier Series (DTFS) of the signal $v(n)$ and then writing each term $\rho^{|\cdot|}$ as a function of $v(n)$. Therefore, starting from the signal $v(n)$, we know that its DTFS corresponds to:

$$V(k) = \mathrm{DTFS}\left(v(n)\right) = \begin{cases} \frac{(L-1)}{2L}, & \text{if } k = 0 \\ -\frac{1}{2L} + j\frac{1}{2L\tan\left(\pi\frac{\tilde{M}^{-1}}{L}k\right)}, & \text{if } 1 \le k \le (L-1) \end{cases},$$

where $\tilde{M}^{-1}$ is the modular multiplicative inverse of M modulo L. From the previous relations (3.4) and (3.5), it is possible to formulate each of the terms $\rho^{|\cdot|}$ as a function of $v(n)$. As an example, using (3.4) and assuming that $\tau \ge 0$, we can rewrite the first term as:

$$\begin{aligned} \rho^{\left|\lfloor n\frac{M}{L}\rfloor - \lfloor (n+\tau)\frac{M}{L}\rfloor\right|} &= \rho^{\lfloor (n+\tau)\frac{M}{L}\rfloor - \lfloor n\frac{M}{L}\rfloor} \\ &= \rho^{\left((n+\tau)\frac{M}{L} - v(n+\tau) - \left(n\frac{M}{L} - v(n)\right)\right)} \\ &= \rho^{\left(\tau\frac{M}{L} - v(n+\tau) + v(n)\right)} \\ &= \rho^{\left(\lfloor \tau\frac{M}{L}\rfloor + v(\tau) - v(n+\tau) + v(n)\right)} \\ &= \rho^{\lfloor \tau\frac{M}{L}\rfloor}\rho^{(v(n)+v(\tau)-v(n+\tau))}. \end{aligned}$$

From the last equality, it can be checked that $(v(n) + v(\tau) - v(n+\tau))$ results in a binary signal which can only take discrete values 0 or 1, thus obtaining the final relation:

$$\rho^{\left|\lfloor n\frac{M}{L}\rfloor - \lfloor (n+\tau)\frac{M}{L}\rfloor\right|} = \rho^{\lfloor \tau\frac{M}{L}\rfloor}\left((1-(1-\rho)\left(v(n)+v(\tau)-v(n+\tau)\right)\right).$$

A similar analysis for the remaining terms $\rho^{|\cdot|}$ allows us to write $c_{yy}(n;\tau)$ as a function of ρ, $v(n)$ and some constants. Consequently, by using several properties of the DTFS, we can obtain the theoretical expression of the Fourier coefficients $C_{yy}(k;\tau)$. For the sake of brevity, we only give the expression of the spectral coefficients for the particular case $\tau = 0$:

$$\begin{aligned} C_{yy}(k;0) &= \mathrm{DTFS}\left(c_{yy}(n;0)\right) \\ &= \frac{1}{1-\rho^2}\left[B(k) - 2V(k) + 2\sum_{l=0}^{L-1} V(l)V(k-l)\right. \\ &\quad \left. +2\left(G(k) \circledast \left(V(k) - \sum_{l=0}^{L-1} V(l)V(k-l)\right)\right)\right], \end{aligned} \tag{3.8}$$

where $\circledast$ stands for the circular convolution operation of period L, $B(k)$ corresponds to the DTFS of a constant signal equal to 1, and $G(k)$ is

$$G(k) = \begin{cases} \frac{1+(L-1)\rho}{L}, & \text{if } k = 0 \\ \frac{1-\rho}{L}, & \text{if } 1 \leq k \leq (L-1) \end{cases}.$$

In Figure 3.3(b), we represent the normalized magnitude of the cyclic correlation $C_{yy}(\omega;\tau)\,|_{\tau=0}$ with $\omega \triangleq \frac{2\pi}{L}k$ and $k \in \{0,\ldots,L-1\}$, through the Fourier coefficients $C_{yy}(k;\tau)\,|_{\tau=0}$ in (3.8), for the different values of ρ pointed out before and keeping the resampling factor at $\xi = \frac{11}{10}$. From the drawn results, we can conclude that the magnitude of the spectral coefficients (excluding the DC component at $k = 0$) is very small for $\rho = 0.95$. This is due to the fact that the correlation $c_{yy}(n;0)$, as it is shown in Figure 3.3(a), is almost constant and therefore the periodicity is hidden. Given that the estimation of the resampling factor depends on the magnitude of those frequencies, it is evident that those peaks must be enhanced for a correct operation.

3.4. Prefilter Design

As it has been brought out in [10, 11, 12, 67, 68], the use of a prefilter improves the resampling detection or estimation performance. In this section, we define a measure that makes possible the design of prefilters which improves the estimate of the resampling rate.

The prefiltering of a resampled signal $y(n)$, with a FIR filter of order P, gives a new signal $z(n)$ with the form

$$z(n) = \sum_{l=0}^{P} p_l y(n-l),$$

where p_l denotes the real-valued coefficients of the prefilter $\mathbf{p}$. The output correlation of this filtered version of the resampled signal $y(n)$ becomes

$$\begin{aligned} c_{zz}(n;\tau) &= \sum_{l=0}^{P}\sum_{m=0}^{P} p_l p_m E\left\{y(n-l)y(n+\tau-m)\right\} \\ &= \sum_{l=0}^{P}\sum_{m=0}^{P} p_l p_m c_{yy}(n-l;\tau+l-m), \end{aligned}$$

that is, a linear combination of shifted versions of the correlation described in (3.7), evaluated at different values of τ. In the Fourier domain, the general expression of the spectral coefficients $C_{zz}(k;\tau)$ can be directly expressed as

$$C_{zz}(k;\tau) = \sum_{l=0}^{P}\sum_{m=0}^{P} p_l p_m C_{yy}(k;\tau+l-m)e^{-j\frac{2\pi k}{L}l} \ ,$$

where $C_{yy}(k;\tau)$ corresponds to the Fourier series coefficients of (3.7), that have been analytically characterized in the previous section.

As we have seen before, the resampled signal $y(n)$ is almost cyclostationary with period $\frac{L}{M}$ and since the prefilter used is a linear time-invariant system, this also holds for the prefiltered signal $z(n)$. From this periodicity and considering the fact that spectral coefficients are symmetric for real-valued signals (i.e., $|C_{zz}(i;\tau)| = |C_{zz}(L-i;\tau)|$), the corresponding cyclic frequencies $\alpha \triangleq 2\pi\frac{M}{L}$ and the replica $\alpha' \triangleq 2\pi\frac{L-M}{L} = -2\pi\frac{M}{L}$ will have a larger magnitude than the rest of frequencies (excluding the DC component). For example, given the cyclic correlation with period $\xi = \frac{11}{10}$ shown in Figure 3.3(b), we can check that the AC spectral coefficients with largest magnitude are $C_{yy}(1;0)$ and $C_{yy}(10;0)$ that match with the corresponding cyclic frequencies $\alpha' = 2\pi\frac{1}{11}$ and $\alpha = 2\pi\frac{10}{11}$, respectively.

Therefore, given that the estimation of the resampling rate can be carried out from the AC spectral coefficients with largest magnitude, because they identify the cyclic frequencies, we use the following criterion to define the target function Θ for a fixed resampling factor $\xi = \frac{L}{M}$ and a given value of ρ as:

$$\Theta(\mathbf{p}) \triangleq \frac{\frac{1}{2}(|C_{zz}(M;0)|^2 + |C_{zz}(L-M;0)|^2)}{\frac{1}{L-2}\sum\limits_{\substack{k=0 \\ k\neq M,L-M}}^{L-1} |C_{zz}(k;0)|^2},$$

where $\mathbf{p}$ stands for a vector containing all the coefficients of the FIR prefilter. Note that the above target function can be viewed as an SNR, where the magnitude of the cyclic frequencies $\alpha = 2\pi\frac{M}{L}$ and $\alpha' = 2\pi\frac{L-M}{L}$ represent the signal part and the remaining spectral coefficients are considered as noise. In fact, Θ can be interpreted as a measure of the detectability of the resampling traces.

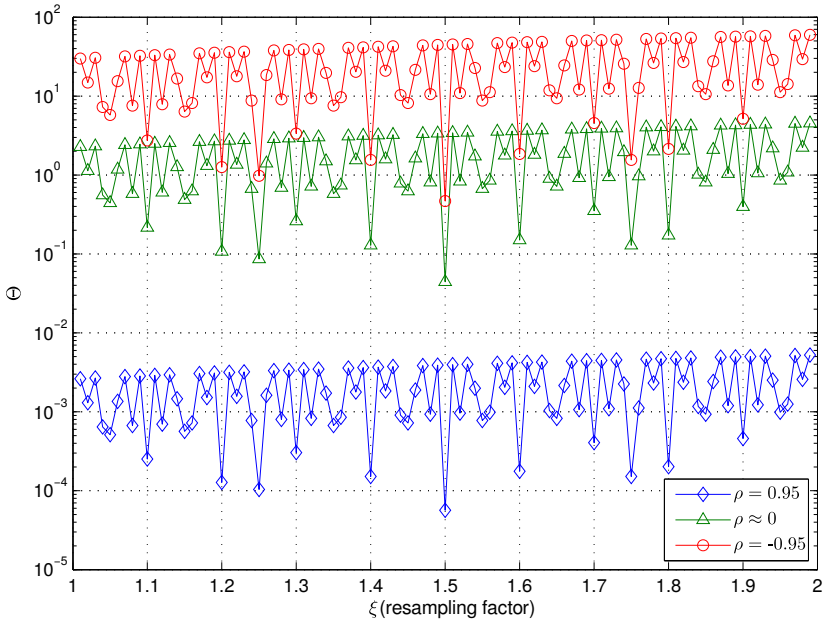


Figure 3.4: Objective function Θ for resampling factors in the interval $1 < \xi < 2$ and for different values of ρ. No prefilter is applied, equivalently, the prefilter is a Kronecker delta.

Our final aim is to maximize this objective function Θ for given values of ρ and resampling factor $\xi = \frac{L}{M}$, so as to obtain an estimate of the optimum prefilter $\hat{\mathbf{p}}$, i.e.,

$$\hat{\mathbf{p}} = \arg \max_{\mathbf{p} \in \mathbb{R}^{P+1}} \Theta(\mathbf{p}).$$

The lack of a closed-form solution to the maximization of Θ makes it difficult to find the fixed optimum prefilter for a range of values of ξ and ρ. Nevertheless, since all the cyclic correlations $C_{zz}(k;0),\ \forall k \in \{0,\dots,L-1\}$ can be straightforwardly evaluated from their analytical expressions, we can numerically find the optimal prefilter maximizing Θ.

In Figure 3.4 we evaluate the target function for three different values of ρ and resampling factors in the range $1 < \xi < 2$, when no prefilter is applied. As it was expected, we can observe that the worst performance is reached when the AR process approximates that of natural images, i.e., when the correlation coefficient is $\rho = 0.95$.

Focusing on the case $\rho = 0.95$, we start considering a prefilter of order 1 and we analyze the target function Θ for a particular resampling factor, e.g., $\xi = \frac{11}{10}$. Figure 3.5 shows the values of Θ for the coefficients p_0 and p_1 in the range $[-5, 5]$. From the representation, it is easy to perceive that the filters that satisfy the condition $p_0 = -p_1$ reach the maximum value of Θ. So, in this particular case, the first-order derivative with $p_0 = 1$ and $p_1 = -1$ is optimal.

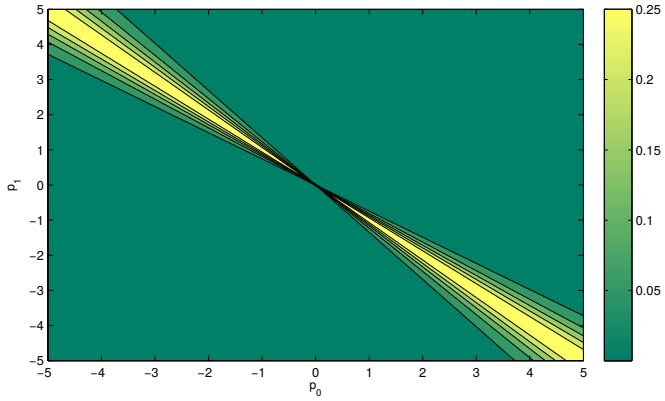


Figure 3.5: Evolution of the objective function Θ using a first-order prefilter ($P = 1$), and varying the coefficients p_0 and p_1 in the range $[-5, 5]$. This example has been particularized for $\xi = \frac{11}{10}$ and $\rho = 0.95$.

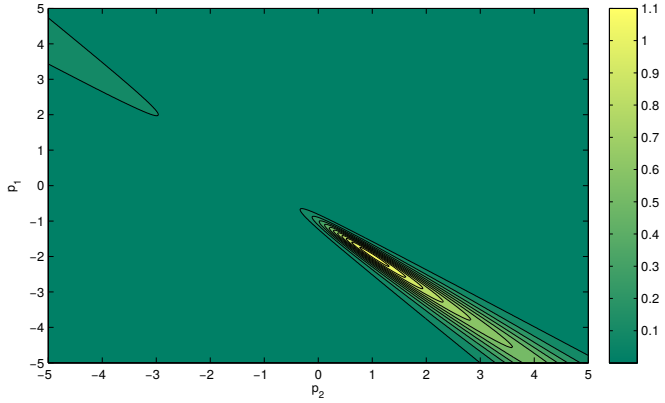


Figure 3.6: Evolution of the objective function Θ using a second-order prefilter ($P = 2$), fixing $p_0 = 1$ and varying the coefficients p_1 and p_2 in the range $[-5, 5]$. This example has been particularized for $\xi = \frac{11}{10}$ and $\rho = 0.95$.

The same analysis is carried out for a FIR filter of order 2, but in order to get representable results, we fix the first coefficient $p_0 = 1$, without loss of generality. Figure. 3.6 represents the variation of the objective function Θ with respect to the prefilter coefficients p_1 and p_2 in the range $[-5, 5]$. The largest value of Θ is achieved at $p_1 = -2$ and $p_2 = 1$. Then, in this case, the optimum prefilter corresponds to the second-order derivative filter.

Thus, these results support the idea of using derivative filters to enhance the spectral peaks. In Figure 3.7, we show the values of Θ considering different orders for the derivatives. As we can see, there is a huge gap between the results obtained without any prefilter and the cases where the derivative filters are used. From these plots we can infer that derivative filters improve the detectability of the cyclic frequencies for all the resampling rates in the range $1 < \xi < 2$.

Interestingly, the third-order derivative presents lower values of Θ than the

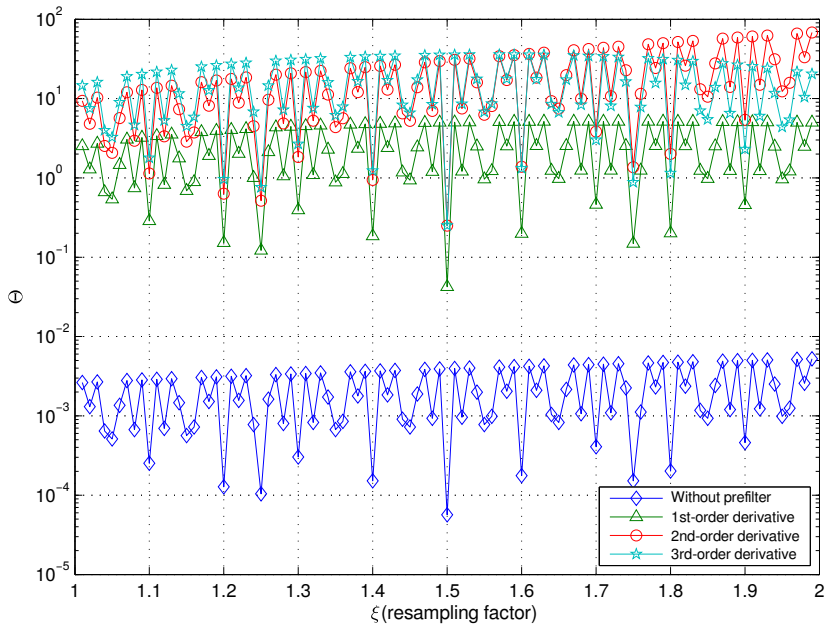


Figure 3.7: Objective function Θ, considering different prefilters, for resampling factors in the interval $1 < \xi < 2$ and for $\rho = 0.95$.

Table 3.1: Optimum prefilters of order 3 for some values of ξ.

Range of	**Coefficients of the prefilter**			
ξ	$\hat{p}_0$	$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$
1.05 - 1.10	1	-2.4	2.4	-1
1.30 - 1.35	1	-2.75	2.75	-1
1.40 - 1.45	1	-2.8	2.8	-1
1.60 - 1.65	1	-5	7.5	-3.5
1.85 - 1.95	1	-2	1.1429	-0.1429

second-order filter when $\xi > 1.6$. An important question is whether better results can be obtained with other kinds of filters. The answer is positive, in fact, as we increase the order of the filter, the optimum prefilter becomes more dependent on the examined resampling rate and other types of prefilters show up. Performing an exhaustive search for the first and second order prefilters, the optimizers of Θ turned out to be respectively the first and second order derivative filters. On the other hand, for third-order prefilters, the optimal filters turned out to be dependent on the resampling factor. Table 3.1, shows some of the prefilters achieved for the different values of ξ.

3.5. Experimental Results

While the obtained prefilters in the previous section can be optimal for a 1-D AR process with $\rho = 0.95$, we wish to evaluate how the prefilters so designed perform with real images. To this end, we carried out an experiment with natural images where we study the estimation accuracy for different scaling factors separated by a distance of 0.05, i.e., $\xi \in \{1.05, \ldots, 1.95\}$. For the evaluation of the prefilters, we use 150 images from a personal image database composed of several realistic scenarios with different indoor and outdoor scenes. All the images in this collection have been captured in a raw format by a Nikon D60 digital camera and have been converted into uncompressed grayscale TIFF images. Each image has been downsampled by a factor of two in order to avoid the interpolation carried out by the camera, due to the color filter array, obtaining images of size 1936×1296.

To reproduce the conditions of the considered model, we first resize each image by the corresponding factor ξ with a (separable) bilinear interpolation kernel (cf. first cell in Table 1.1) and then we take a large image block of size 1024×1024 pixels. Next, we subtract the mean value of this portion of the image in order to get a zero-mean block, we subsequently apply the corresponding prefilter and, finally, we compute the 2-D Fourier transform of the correlation of the block for $\tau = 0$ (i.e., the cyclic correlation). Notice that to exclude the DC component, we just subtract the mean value of the correlation before the computation of the Fourier transform. Considering this 2-D spectrum, the resampling rate is obtained from that frequency pair (ω_1, ω_2) with the largest magnitude. Note that ω_1 represents the horizontal frequency axis and ω_2 the vertical one. Since the range of resampling factors that we employ is $1 < \xi < 2$, the estimated value is computed as follows:

$$\hat{\xi} = \frac{2\pi}{2\pi - \max_{i\in\{1,2\}} |\omega_i|},$$

where we use $\max_{i\in\{1,2\}} |\omega_i|$ to avoid the case when one of both components is equal to zero (i.e., the cyclic frequency is located over one of the axes). In this occasion, we deem the estimation as correct when the detected cyclic frecuency (ω_1, ω_2) is in the range defined by the resolution in the frequency domain, i.e.,

$$\left| \max_{i\in\{1,2\}} |\omega_i| - \alpha \right| \leq \frac{2\pi}{1024},$$

where $\alpha \triangleq 2\pi - \frac{2\pi}{\xi} = 2\pi\frac{L-M}{L}$ is the theoretical value of the cyclic frequency.

Figure 3.8 shows the obtained estimation accuracy for the different values of ξ. From this plot, we can observe that the proposed analysis and target function yield satisfactory results, as better performance is achieved with those prefilters that reach a larger value of Θ. For instance, comparing the values obtained for the second and third order prefilters we see that the performance of the former

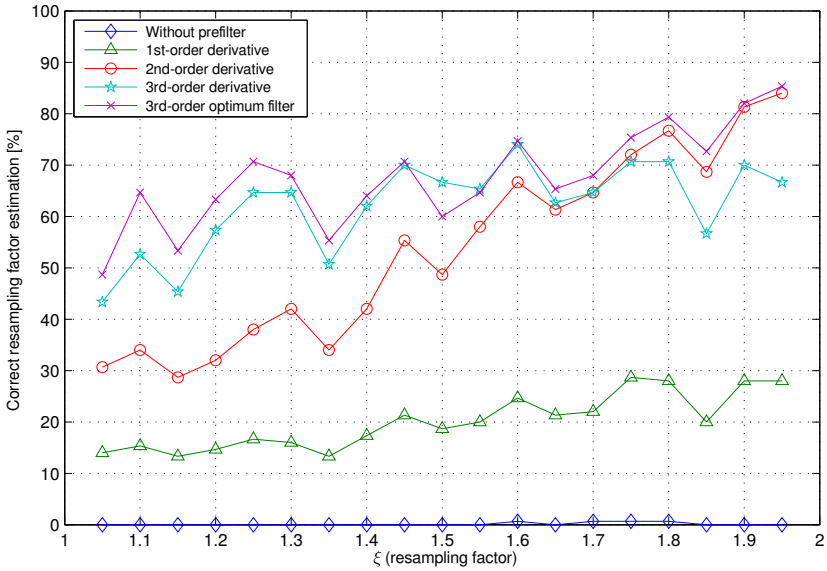


Figure 3.8: Estimation accuracy of the resampling factor for image blocks of size 1024×1024 pixels, using different prefilters.

improves when $\xi > 1.6$ as it was shown in Figure 3.7. We can also confirm the worse performance of the third-order derivative filter with respect to the numerically computed third-order optimum prefilter, so we can conclude that derivative filters are no longer the best solution once we increase the order of the prefilter above 2.

Focusing on the estimation performance, the obtained results cannot be considered very optimistic, since the prefilter that reach the best results is far from the perfect estimation. This is due to our model only capturing the deterministic value of the cyclic correlation without considering any other effects. In this case, windowing (by taking a block of size 1024×1024) introduces further components at all frequencies, but especially those near DC (i.e., the frequencies included in the main lobe of the window). The magnitude of the latter is heavily influenced by the DC component, so in many cases the cyclic frequency is incorrectly detected, due to the fact that the largest components are located within the DC main lobe. By leaving those components (i.e., $\omega_i \leq 2\pi/1024$) out during the detection process, we obtain the results shown in Figure 3.9. This way, the estimation accuracy is highly improved for all the analyzed prefilters, achieving with our proposed design an estimation accuracy close to 90%[1].

[1]This comes at the price of missing resampling factors $1 < \xi \leq 1.001$.

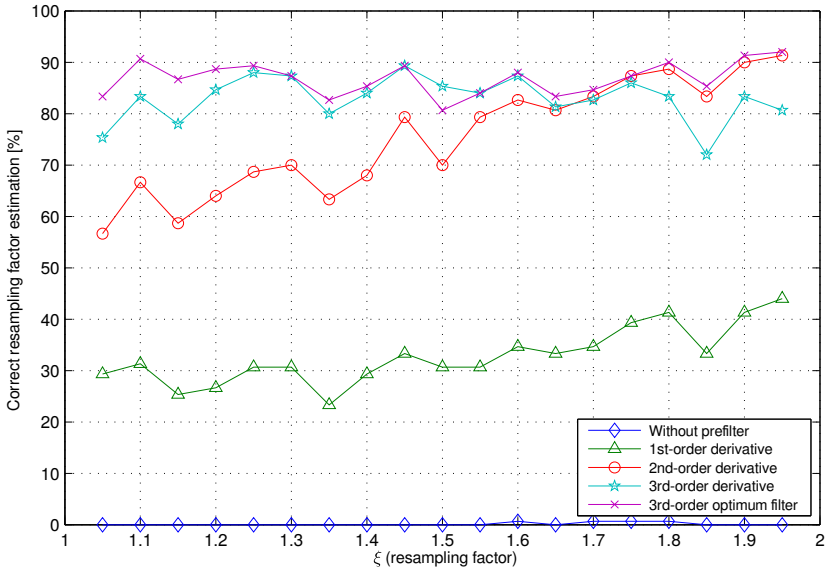


Figure 3.9: Estimation accuracy of the resampling rate, excluding near-zero frequencies, for image blocks of size 1024×1024, using different prefilters.

3.6. Conclusions

In this chapter, the design of prefilters to improve the estimation accuracy of the resampling factor of spatially transformed images has been analytically investigated. Although the proposed analytical framework only models the deterministic value of the cyclic correlation, experimental results validate the use of the defined objective function for the design of prefilters.

Chapter 4

ML Estimation of the Resampling Factor

This chapter addresses the problem of resampling factor estimation for tampering detection following the maximum likelihood criterion. Pursuing the line of research established in the previous chapter, the design of prefilters has been further investigated modeling the influence of the rounding operation applied after resampling. From the study of its effect on resampling detection/estimation we have realized that instead of modeling this operation as a noisy component, we could benefit from the structure it imposes upon the resampled data. As a result, by relying on the rounding operation applied after resampling, an approximation of the likelihood function of the quantized resampled signal is obtained. Then, 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.

4.1. Introduction

From the review of the literature carried out in Section 1.2.3, we have seen that seminal works addressing forensic resampling detection were focused on the analysis of the particular linear dependencies introduced among neighboring pixels by the resampling process when applying a spatial transformation (e.g., scaling or rotation). On the other hand, from the last two chapters, we are aware that the resampling operation can be modeled as a time-varying filtering that induces periodic correlations, so links have been established between this problem and the cyclostationarity theory in [67, 68, 71], providing a theoretical framework for the estimation of the parameters of the transformation.

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 close to 1).

With these shortcomings in mind, in this chapter we approach the estimation of the resampling factor following the Maximum Likelihood (ML) criterion. The approximation of the likelihood function of the resampled signal relies on the rounding operation applied after 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 is provided. The proposed approach will only consider one-dimensional 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.

The rest of the chapter is organized as follows: in Section 4.2, the problem we want to solve is formally introduced, while the description of the method for estimating the resampling factor, based on the ML criterion, is addressed in Section 4.3. Experimental results with synthetic and real signals are reported in Section 4.4 for evaluating the performance of the estimator. Finally, the chapter is closed with some conclusions in Section 4.5.

4.2. Preliminaries and Problem Formulation

The alteration of an image should not introduce visible distortions, hence a forger will be restricted to apply only slight transformations. This implies that the resampling estimator should achieve good performance for resampling factors near 1. This chapter only deals with the case where the resampling factor is larger than 1. Of course, 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 2-D case. However, we will see in Section 4.3 that the obtained method following the ML criterion can be easily extended to the 2-D case. The same holds for the interpolation

filter taken into account. 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.

A formal description of all the steps involved in the change of the sampling rate of a 1-D signal $x(n)$ by a resampling factor ξ, has already been covered in the previous chapter, specifically in Section 3.2. Therefore, we avoid rewriting the whole procedure again. Instead, we focus on the modeling of the rounding operation applied after the resampling process.

Regarding the set of values that the original signal can take, we will assume that all the samples from $x(n)$ 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 of $y(n)$ (cf. Eq. (3.2)) 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

$$\begin{aligned} z(n) &= Q_\Delta\left(y(n)\right) \\ &= \begin{cases} Q_\Delta\left(x\left(\lfloor n\frac{M}{L}\rfloor\right)\left(1-\operatorname{mod}\left(n\frac{M}{L},1\right)\right)+x\left(\lceil n\frac{M}{L}\rceil\right)\operatorname{mod}\left(n\frac{M}{L},1\right)\right), & \text{if } n\frac{M}{L}\notin\mathbb{Z} \\ x\left(n\frac{M}{L}\right), & \text{if } n\frac{M}{L}\in\mathbb{Z} \end{cases}, \end{aligned} \tag{4.1}$$

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 (4.1), it is evident that some of the original samples are "visible" in its 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.

4.3. 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 N_x samples and the corresponding quantized resampled signal by the vector $\boldsymbol{z}$ with N_z samples. For convenience, we will assume that the length of the original signal is $N_x = N + 1$ with N a multiple of M, and so, the corresponding length of the resampled signal will be $N_z = \xi N + 1$. We will find it convenient to model $\boldsymbol{x}$ 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 following the ML criterion relies on finding the conversion rate 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 (4.1). 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. (4.1)). 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, as shown in (4.1), and also the linear relation established between the remaining samples.

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

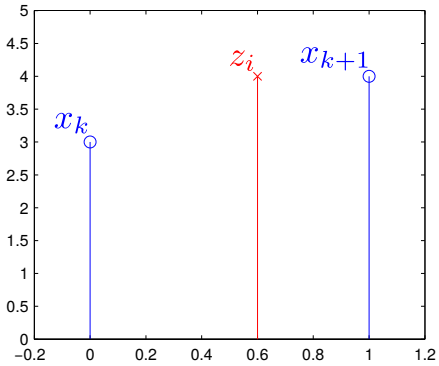
Along the derivation of the joint pdf $f_{\boldsymbol{Z}|\Xi}(\boldsymbol{z}|\xi)$, for the sake of notational simplicity, we will refer to this one as $f_{\boldsymbol{Z}}(\boldsymbol{z})$, considering implicitly that we are assuming a particular resampling factor ξ. From the dependence between the quantized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

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

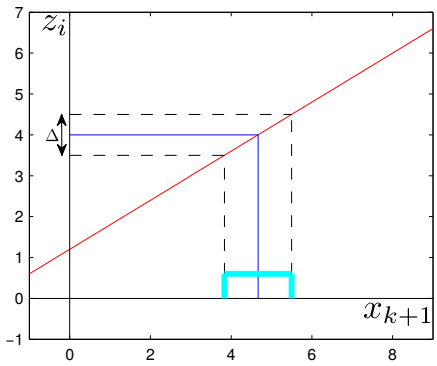
We assume that no a priori knowledge on the distribution of the input signal is available. This is equivalent to considering that $f_{\boldsymbol{X}}(\boldsymbol{x})$ is uniform and, consequently, the joint pdf can be approximated by the following relation

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

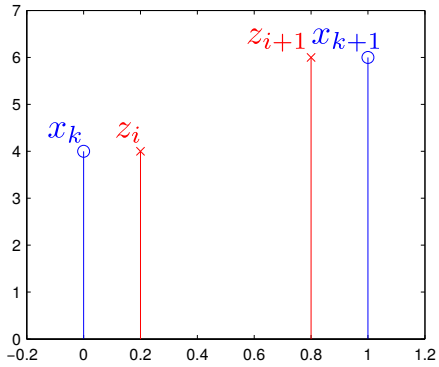
Equation (4.1), indicates that every L samples of the observed vector $\boldsymbol{z}$, we have a visible sample from the original signal. This implies that the random variable Z_i, given X_k, is deterministic whenever $i \in L\mathbb{Z}$ and $k \in M\mathbb{Z}$. For this reason,



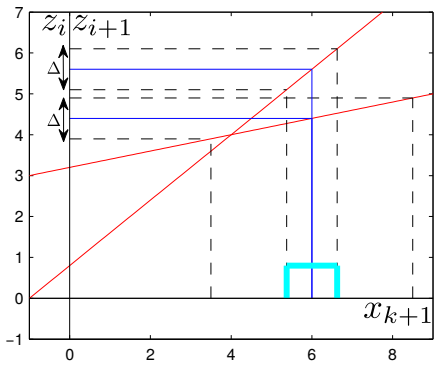
(a) z_i between x_k and x_{k+1}



(b) Feasible interval and pdf



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



(d) Feasible interval and pdf

Figure 4.1: Illustrative example, showing the last two possible cases for z_i. Pdfs obtained are shown graphically. Note that $\Delta = 1$.

the previous joint pdf can be obtained by processing $(N_z - 1)/L = N/M$ distinct and disjoint blocks, i.e.,

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) \approx \prod_{j=0}^{N/M-1} \int_{\mathbb{R}^M} f_{\boldsymbol{Z}_{Lj}|\boldsymbol{X}_{Mj}}(\boldsymbol{z}_{Lj}|\boldsymbol{x}_{Mj}) d\boldsymbol{x}_{Mj}, \tag{4.2}$$

where $\boldsymbol{Z}_{Lj}$ and $\boldsymbol{X}_{Mj}$ (and also their corresponding outcomes) are vectors of size L and M starting at indices Lj and Mj, respectively.

The calculation of the contribution of each block of L samples from the vector of observations $\boldsymbol{z}_{Lj}$ in (4.2), 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. (4.1)), 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 Figure 4.1(a). In this case, if the variance of the original signal is large enough compared 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 (4.1). A graphical representation, depicted in Figure 4.1(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 Figure 4.1(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. Figure 4.1(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 (4.2) must be evaluated with respect to the corresponding original sample x_k. Intuitively, we can observe that the calculation of (4.2) 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 4.4.

4.3.2. 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 is tested, which we will denote by ξ_t. 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 Figure 4.2(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 $f_{Z_1|X_0,X_1,\Xi}(z_0|x_0,x_1,\xi_t) = \Pi\left(\frac{a_1x_0+b_1x_1-z_1}{\Delta}\right)$.

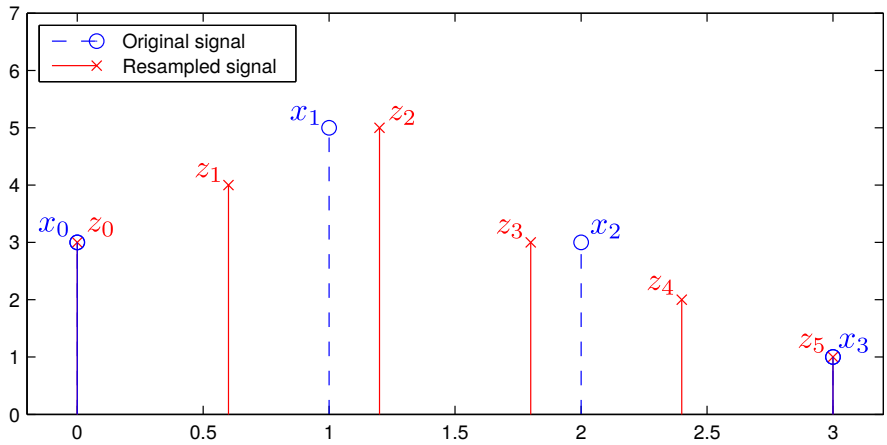
   Figure 4.2(b) shows with a red line the linear relation between the interpolated value and the original ones $y_1 = a_1x_0 + b_1x_1$, with the value of x_0 fixed, i.e., according to the previous step $x_0 = z_0$. From the value of z_1 we obtain the feasible interval of x_1 (represented with dashed black 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 $f_{\boldsymbol{Z}_2|X_1,X_2,\Xi}(\boldsymbol{z}_2|x_1,x_2,\xi_t) = \Pi\left(\frac{a_2x_1+b_2x_2-z_2}{\Delta}\right)\Pi\left(\frac{a_2x_1+b_2x_2-z_3}{\Delta}\right)$.

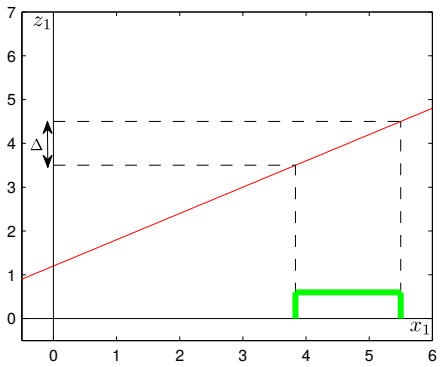
   Figure 4.2(c) shows in this case the corresponding two linear relations for $y_2 = a_2x_1+b_2x_2$ and $y_3 = a_3x_1+b_3x_2$. Be aware that in this case x_1 can take any value in the range obtained in Step 2, and that is the reason why the dashed red lines are plotted. From the product of the two rectangular pdfs, we obtain the feasible interval for x_2 (whose pdf is represented in cyan).

   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.

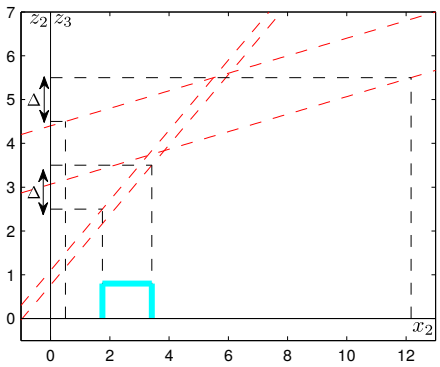
   If the factor cannot be discarded, then we must compute the convolution of the uniform pdf here obtained with the one resulting from Step 2. The result is plotted in green in Figure 4.2(d).



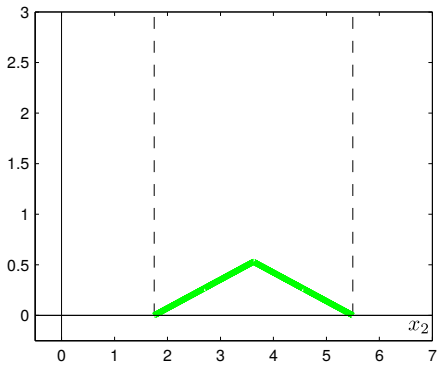
(a) Original and resampled signals



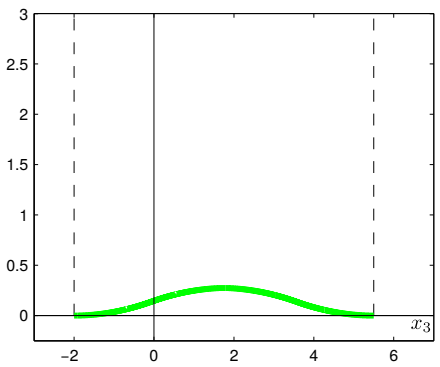
(b) Resulting pdf in Step 2



(c) Feasible interval for x_2



(d) Resulting pdf in Step 3



(e) Resulting pdf in Step 4

Figure 4.2: Graphical representation of the method description. Note that $\Delta = 1$.

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 Figure 4.2(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.

4.4. 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 [72] (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 Section 4.1, comparative results with a 1-D version of the resampling detector proposed by Popescu and Farid in [8] are also provided.

4.4.1. Performance Analysis with Synthetic Signals

In this case, we consider as synthetic signal a first-order autoregressive process, parameterized by a single correlation coefficient ρ. As indicated in the previous chapter, an 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 (cf. Section 3.3); 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 white noise 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

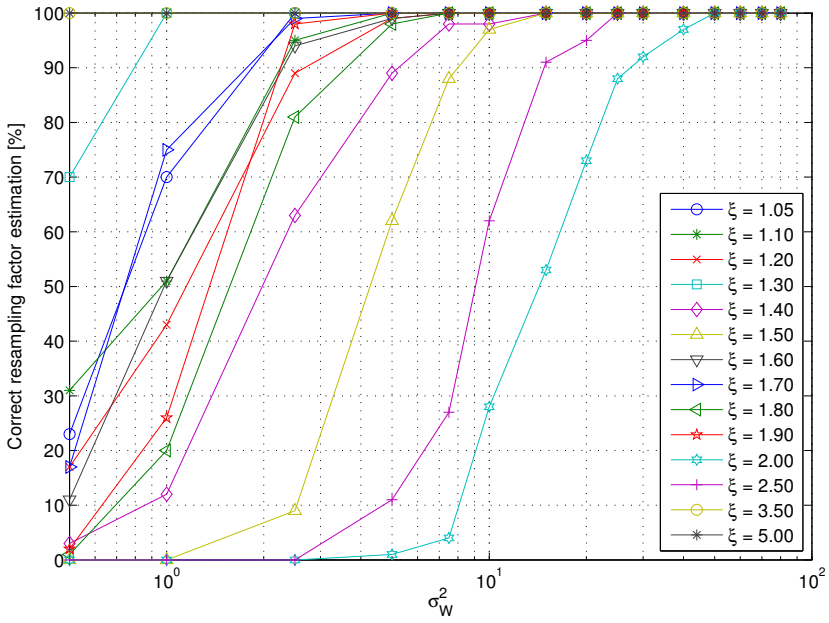


Figure 4.3: Correct resampling factor estimation percentage for different resampling factors as a function of σ^2_W. $\rho = 0.95$, and 500 Monte Carlo realizations are considered.

of the AR(1) process, so results will be drawn as a function of σ^2_W 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 $(1, 5]$. Note that we have sampled with a smaller step the range between $(1, 2]$ because we want to analyze the perforemance of the estimator for several values close to 1. Notice 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 $N_z = 400$.

Figure. 4.3 shows the percentage of correct estimation for some of the resampling factors in the set as a function of σ^2_W. From this plot, we can observe that the performance of the estimator strongly depends on the mentioned variance of the 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 ($\sigma^2_W = 0.5$), is required to correctly estimate the resampling factor for all the experiments; nevertheless, for $\xi = 2$, almost a value of $\sigma^2_W = 50$ will be necessary for getting the same estimation performance. In general, and in accordance with the assumptions backing the analysis introduced in the previous

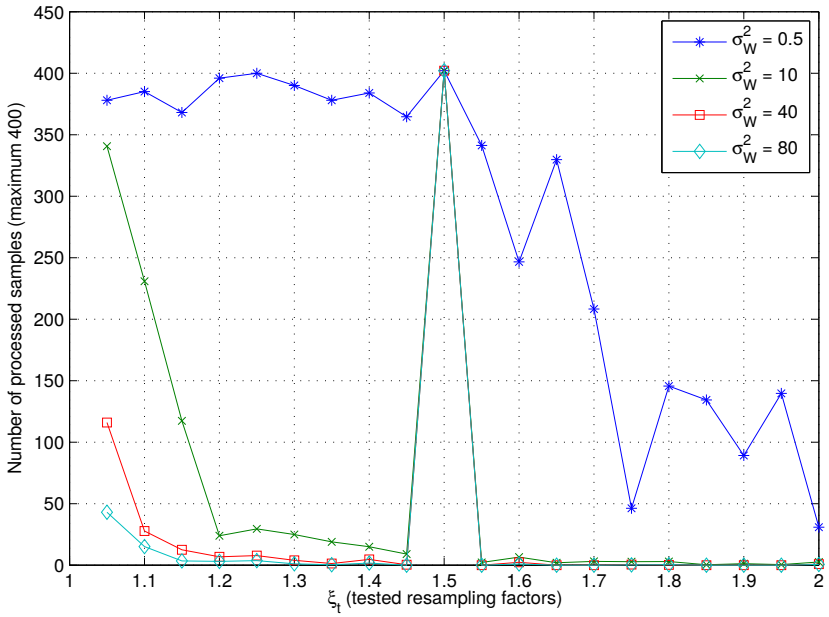


Figure 4.4: Number of discarded samples for different values of σ_W^2, as a function of ξ_t. The true resampling factor is $\xi = \frac{3}{2}$. 500 Monte Carlo realizations were performed.

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 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 4.3.2).

This is illustrated in Figure 4.4, where the number of samples required for discarding the candidate resampling factor is shown for different values of σ_W^2, when $\xi = \frac{3}{2}$. As it can be checked in that figure, whenever $\xi_t = \xi$, the tested resampling factor will not be discarded, even when the full vector of observations has been processed, as it should be expected. It is also important to point out that the larger the value of σ_W^2, i.e., the more accurate the fine-quantization assumption is, the smaller number of samples is required for discarding a wrong resampling factor under test ξ_t.

4.4.2. Performance Analysis with Real Audio Signals

For the evaluation of the estimator in a real scenario, we consider the "Music Genres" audio database [72], composed of 1000 uncompressed audio files with 10

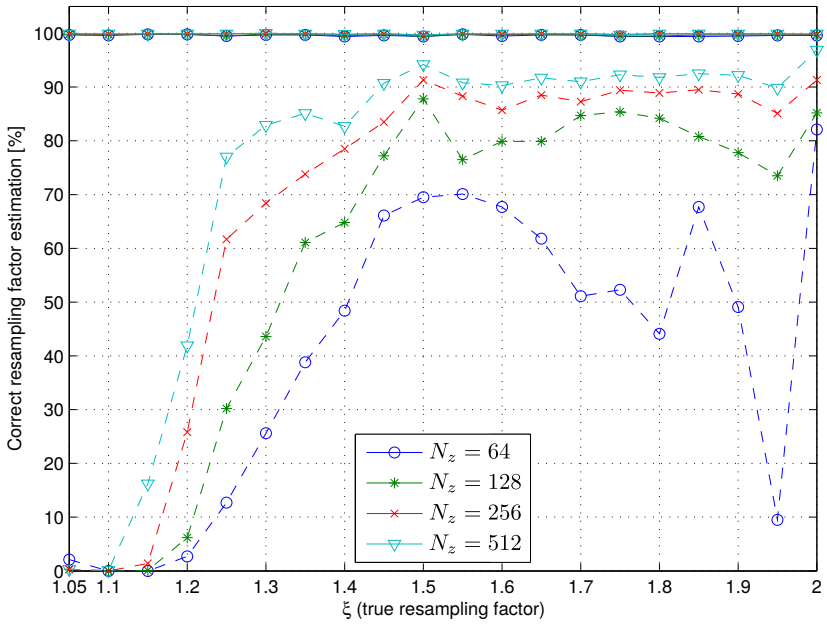


Figure 4.5: Comparison of the correct estimation percentage of the proposed MLE versus the method in [8]. Solid lines represent the obtained results with the MLE, while dashed lines are used for Popescu and Farid's method.

different music styles (some 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 that can be easily adapted to the 1-D case, i.e., the one proposed by Popescu and Farid in [8].[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 $N_z \in \{64, 128, 256, 512\}$.

The results obtained with both methods are shown in Figure 4.5. 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 p-map has been computed) for the detection of the resampling factor, as we pointed out in Section 4.1. 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

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

available, thus leading to a very practical forensic tool.

Although the performance of the MLE is very good, if we apply the same analysis to 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.

4.5. Conclusions

The problem of resampling factor estimation following the ML criterion has been investigated in this chapter 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.

Chapter 5

Set-Membership Identification of Resampled Signals

A new direction regarding the problem of resampling factor estimation is explored in this chapter. Over the last years, most of the proposed techniques for tampering detection through the analysis of resampling traces put emphasis on characterizing the periodic linear dependencies induced in the resampled signal by the application of a spatial transformation. However, less effort has been devoted to the analysis of the constraints imposed by the resampling process on the resulting signal. Taking as reference the work described in the previous chapter and pursuing this new line of research, we tackle the problem of resampling factor estimation in terms of set-membership estimation theory. The proposed technique constructs a model of the problem according to available a priori knowledge and in consonance with a finite number of observations that comes from the resampled signal under study. With this information, the derived technique is able to provide an estimate of the resampling factor applied to the original signal and, if required, an estimate of such signal together with an estimate of the interpolation filter. The performance in terms of accuracy and MSE of this approach is evaluated and comparative results with state-of-the-art methods are reported.

5.1. Introduction

The detailed techniques in Section 1.2.3 dealing with resampling detection work remarkably well when uncompressed signals are used, but the corresponding detectors can be easily deluded when a post-processing or simply a lossy-compression is applied to their content, as it is described in [73]. Furthermore, all these approaches are based on the study of the periodic correlation that is inherently induced in the resulting signals after applying a resampling operation. The main drawbacks of the frequency analysis for resampling factor estimation

have just been covered in Section 4.1 from the previous chapter. Despite the unavoidable ambiguity in the identification of the resampling factor due to frequency aliasing, the main issues are: 1) a considerably large number of samples are necessary to circumvent the windowing effect in the frequency domain; 2) the presence of periodic patterns in the content usually leads to a wrong detection or estimation. By relying on the rounding operation applied after resampling, the estimator derived in the previous chapter is able to sort out these problems, as it can be checked in Section 4.4. However, its applicability is quite limited since only a fixed linear interpolation filter is considered through the definition of the estimator.

To overcome these deficiencies and pursuing the idea behind the work in Chapter 4, which gave important insights about how to perform resampling factor estimation, a new approach for the identification of resampled signals is described next. The procedure derived in the earlier Section 4.3, where a vector of observations coming from a linearly resampled signal is tested against a set of plausible resampling factors to find the correct one, is able to quickly discard the tested resampling factors that lead to an empty feasible set for the original signal. This formulation of the problem can be linked to the set-membership estimation theory (a.k.a., set-theoretic estimation), which is well known in the field of automatic control and also in certain signal processing areas [74, 75].

Set-membership estimation is commanded by the concept of feasibility and provides solutions whose singular characteristic is to be consistent with all information arising from the observed data and the a priori knowledge about the problem to solve. As it was stated above, frequency-based methods cannot always provide reliable solutions. Indeed, such solutions could infringe known constraints about the problem. However, when the problem is approached in set-membership terms, the provided solution will be consistent with all the known constraints, according to the observed data. This is very important from the point of view of a forensic analyst that must always provide objective judgments on the identification of forgeries, basing his decisions on evidences, i.e., on the observed data, and on the prior knowledge about the problem under analysis. To this extent, by relying on the set-membership theory and generalizing the work carried out in Chapter 4 to any interpolation filter, we propose a new methodology for resampling factor estimation.

The structure of this chapter is as follows: Section 5.2 describes the formulation of the problem in mathematical terms and following the set-membership framework; Section 5.3 introduces a practical implementation to solve the derived problem; Section 5.4 shows the experimental results obtained under different settings; and finally, Section 5.5 points out drawn conclusions.

5.2. Problem Formulation

Before introducing the set-membership formulation, the description of all the steps involved in the sampling rate conversion of a 1-D signal will be presented. Note that, though a bit different, the following description is still equivalent to the one carried out in Chapters 3 and 4, and it is of course compatible with the 2-D model detailed in Section 1.2.2. Once more, we will only focus the analysis on 1-D signals to keep the definition of the problem more tractable, but it is easy to check that the 2-D extension can be straightforwardly obtained.

To avoid confusion between the parameters that are actually used to generate the observed data and those that are evaluated as candidates under test, the following notational convention will be used: only the variables involved in the generation of the observed signal will be denoted with the grapheme $\tilde{}$, for instance, $\tilde{\xi}$ stands for the resampling factor used for generating the observed resampled signal.

Let $\tilde{\mathbf{x}}$ be a column vector that contains $\tilde{N}_x$ samples from the original signal before being resampled. The applied resampling factor is defined as $\tilde{\xi} \triangleq \frac{\tilde{L}}{\tilde{M}}$, i.e., the ratio between the upsampling factor $\tilde{L} \in \mathbb{N}^+$ and downsampling factor $\tilde{M} \in \mathbb{N}^+$. Regarding the interpolation filter, denoted by the column vector $\tilde{\mathbf{h}}$, we consider a freely designed low-pass FIR filter of order $\tilde{N}_h - 1$ with cutoff frequency $\tilde{\omega}_c = \min\left(\frac{\pi}{\tilde{M}}, \frac{\pi}{\tilde{L}}\right)$ in order to avoid aliasing. Under these premises, the resampled version of $\tilde{\mathbf{x}}$, can be written as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{X}}\tilde{\mathbf{h}},$$

where $\tilde{\mathbf{X}}$ is a matrix of size $\tilde{N}_z \times \tilde{N}_h$ with $\tilde{N}_z = \frac{\tilde{L}}{\tilde{M}}\tilde{N}_x$,[1] which is constructed from the samples of $\tilde{\mathbf{x}}$, i.e., $\tilde{x}_i$ with $i = 0, \ldots, \tilde{N}_x - 1$, and as a function of the employed resampling factor $\tilde{\xi}$. Each element (i, j) of the matrix $\tilde{\mathbf{X}}$ is denoted by $\tilde{X}_{ij}$ and is defined as:

$$\tilde{X}_{ij} \triangleq \begin{cases} \tilde{x}_{\frac{i\tilde{M}+k-j}{\tilde{L}}}, & \text{if } \frac{i\tilde{M}+k-j}{\tilde{L}} \in \left(\left\lceil \frac{i\tilde{M}-k}{\tilde{L}} \right\rceil, \left\lfloor \frac{i\tilde{M}+k}{\tilde{L}} \right\rfloor\right) \cap \mathbb{Z} \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

with $k \triangleq \frac{\tilde{N}_h - 1}{2}$, $i = 0, \ldots, \tilde{N}_z - 1$ and $j = 0, \ldots, \tilde{N}_h - 1$. In the above expression, $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and floor functions, respectively.

The interpolated values of $\tilde{\mathbf{y}}$ will be generally represented with more bits than for the original signal $\tilde{\mathbf{x}}$, hence a requantization to the original precision is commonly done prior to saving the resulting signal. This quantized version of the resampled signal, denoted by $\tilde{\mathbf{z}}$, is expressed as

$$\tilde{\mathbf{z}} = Q_{\tilde{\Delta}}\left(\tilde{\mathbf{y}}\right) = Q_{\tilde{\Delta}}\left(\tilde{\mathbf{X}}\tilde{\mathbf{h}}\right), \tag{5.2}$$

[1] Without loss of generality and for the sake of simplicity, we will assume that $\tilde{N}_x$ is a multiple of $\tilde{M}$ and also that $\tilde{N}_h$ is an odd number.

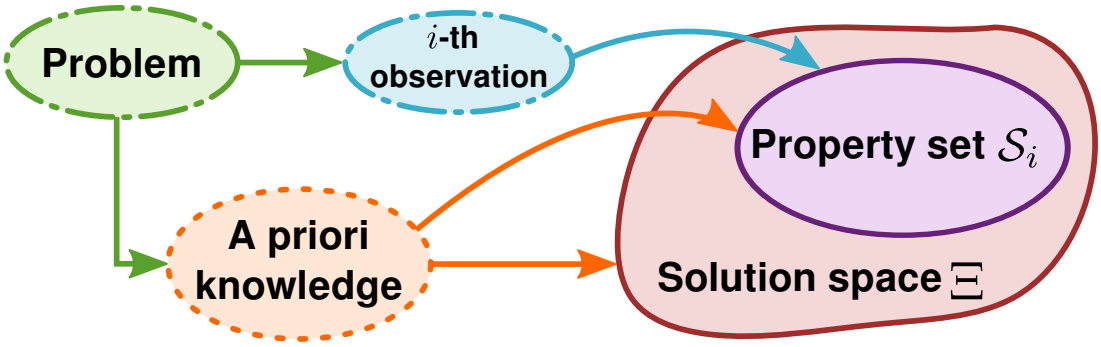


Figure 5.1: Illustrative scheme of the set-membership formulation of a general problem whose solution belongs to the space Ξ.

where $Q_{\tilde{\Delta}}(\cdot)$ represents a uniform scalar quantization with step size $\tilde{\Delta}$ (i.e., the same one used for the original signal).[2]

5.2.1. Set-Membership Formulation

As pointed out above, the set-membership theory is governed by the concept of feasibility; hence, once applied to a particular problem, its main goal is to find a solution that satisfies simultaneously all the constraints defined through the observed data and the a priori knowledge about the problem. In those cases where there exists no solution fulfilling all the requirements at the same time, the problem does not have a feasible solution.

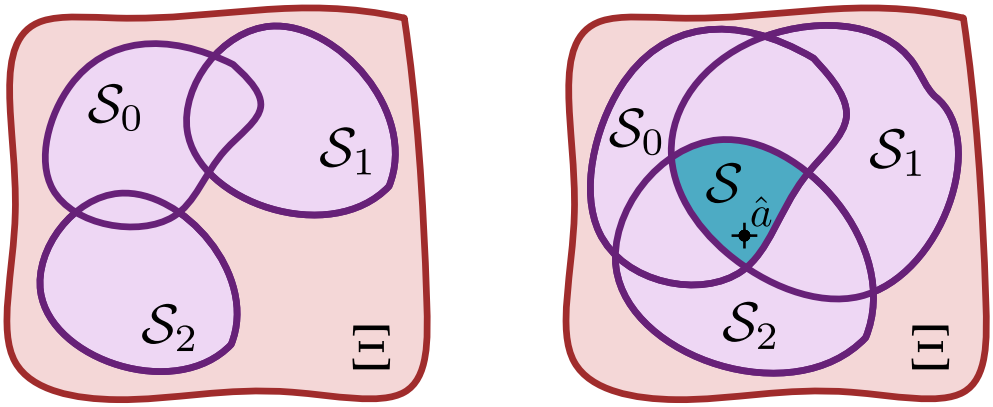
Let us first introduce the set-membership formulation of a general problem whose solution belongs to a space Ξ. Each piece of information from the observed data, i.e., each i-th observation, is associated with a property set $\mathcal{S}_i$ in the solution space Ξ and can be defined as follows

$$\mathcal{S}_i = \{a \in \Xi : a \text{ satisfies } \Psi_i\},$$

where Ψ_i represents a constraint of the problem and a is an arbitrary point of the solution space Ξ. Figure 5.1 illustrates in a graphical manner such problem formulation within a set-membership framework.

Each subset $\mathcal{S}_i$ represents all the estimates that are consistent with the i-th observation. Therefore, the feasible set of solutions for the problem will be composed by the intersection of all the property sets that are obtained with N available observations, thus having $\mathcal{S} = \cap_{i=0}^{N-1}\mathcal{S}_i$, where $\mathcal{S}$ is also commonly known as the solution set. If the solution set is empty, i.e., $\cap_{i=0}^{N-1}\mathcal{S}_i = \emptyset$, then the problem is designated as *infeasible*, as exemplified in Figure 5.2(a). Otherwise,

[2]Note that having the same quantization step size in both cases is not a limiting condition, since the problem can be reformulated if it is not so.



(a) Infeasible problem (b) Feasible problem

Figure 5.2: Examples of possible outputs in a set-theoretic formulation.

the problem is *feasible* and a set-membership estimate consists in choosing any point $\hat{a} \in \mathcal{S}$, as shown in Figure 5.2(b).

Set-membership theory allows us to define a feasibility problem for checking whether a vector of observations $\mathbf{z}$ of length N_z has been resampled or not with a candidate resampling factor $\xi \triangleq \frac{L}{M}$, with $L, M \in \mathbb{N}^+$. Note that in this case we will assume that N_z is a multiple of L for the sake of simplicity and without loss of generality. To characterize this problem in set-membership terms, we need to define the solution space Ξ, which in this case turns out to be the Cartesian product of two sets, i.e., $\Xi = \mathcal{X} \times \mathcal{H}$, where the set $\mathcal{X}$ represents the domain of the original signal, and the set $\mathcal{H}$ specifies the domain of the interpolation filter.

Prior knowledge about the problem helps us define these two sets. For the original signal, we infer that each sample x_i has been quantized with step size Δ, so we could assume that $x_i \in \Delta\mathbb{Z}$, but this assumption would make the resolution of the subsequent optimization problem notably more complicated. In order to lighten the consequent computational burden, we assume without loss of generality that each sample lies in a real interval $[x_{\min}, x_{\max}]$, thus having

$$\mathcal{X} = \left\{ \mathbf{x} \in \mathbb{R}^{N_x} \,:\, x_{\min} \leq x_i \leq x_{\max},\ i = 0, \ldots, N_x - 1 \right\},$$

where N_x represents the dimension of the set and is defined as a function of the number of observations and the candidate resampling factor, i.e., $N_x = N_z \frac{M}{L}$. Regarding the interpolation filter, we assume that each coefficient falls in a real interval $[h_{\min}, h_{\max}]$, hence

$$\mathcal{H} = \left\{ \mathbf{h} \in \mathbb{R}^{N_h} \,:\, h_{\min} \leq h_i \leq h_{\max},\ i = 0, \ldots, N_h - 1 \right\},$$

where the dimension of the set comes from the order of the FIR filter, which is assumed to be $N_h - 1$. The interval $[h_{\min}, h_{\max}]$ can be specified according to any particular filter, for instance, for a linear interpolator we could presume $h_i \in [0, 1]$, $\forall i$.

In order to check if each component z_i of the vector of observations has been generated through the sampling rate conversion of a vector $\mathbf{x} \in \mathcal{X}$ by a candidate resampling factor ξ and using an interpolation filter $\mathbf{h} \in \mathcal{H}$, we must rely on the quantization applied to the resampled signal in (5.2). Since we assume as known the size of the quantization step, i.e., Δ, we have information about the interval where the values of the resampled signal $\mathbf{y} = \mathbf{X}\mathbf{h}$ will lie on.[3] Therefore, any pair $(\mathbf{x}, \mathbf{h})$ from the solution space must generate values of the resampled signal $\mathbf{y}$ with the candidate resampling factor ξ inside the interval defined by the quantization error of the scalar quantizer with step size Δ, that can be written as

$$z_i - \frac{\Delta}{2} < y_i \leq z_i + \frac{\Delta}{2}, \quad \text{for } \; i = 0, \ldots, N_z - 1.$$

Consequently, we assume that the feasible region imposed by each observation z_i of the signal under analysis is limited by two hyperplanes that yield the following property sets

$$\mathcal{S}_i = \mathcal{X}_i \times \mathcal{H}_i = \left\{ (\mathbf{x}, \mathbf{h}) \in \Xi \, : \, -\frac{\Delta}{2} < \mathbf{x}_i^T \mathbf{h} - z_i \leq \frac{\Delta}{2} \right\}, \tag{5.3}$$

for $i = 0, \ldots, N_z - 1$, and where $\mathbf{x}_i$ is a column vector built up with the N_h elements of the i-th row of matrix $\mathbf{X}$. Finally, the feasible solution set for our problem will be the intersection of these N_z property sets: $\mathcal{S} = \cap_{i=0}^{N_z-1}(\mathcal{X}_i \times \mathcal{H}_i)$. If such intersection leads to $\mathcal{S} = \emptyset$, then there exists no $\mathbf{x} \in \mathcal{X}$ and $\mathbf{h} \in \mathcal{H}$ that would generate the vector of observations $\mathbf{z}$ with such candidate resampling factor ξ. Otherwise, an estimate of the original signal $\hat{\mathbf{x}}$ together with an estimate of the interpolator $\hat{\mathbf{h}}$ can be obtained by taking any $(\hat{\mathbf{x}}, \hat{\mathbf{h}}) \in \mathcal{S}$.

5.3. Practical Algorithms

One of the widely-known methods for solving feasibility problems in terms of set-membership theory is the Optimal Value Ellipsoid (OVE) algorithm [76]. However, this method can only be applied when constraints are convex and, in our particular case, the modeling of the resampling identification problem requires nonconvex terms. As it can be observed from the definition of the property sets in (5.3), the constraints of our problem are actually bilinear, due to the product between the variables $\mathbf{x}$ and $\mathbf{h}$. Under these conditions, the feasible solution set is not necessarily convex, leading to consider nonlinear programming algorithms as a way to solve the problem.

Before explaining the particular strategy we have designed, we formally introduce the feasibility problem (derived from Section 5.2.1) that is addressed for the

[3]Note that the matrix $\mathbf{X}$ with size $N_z \times N_h$ is generated according to (5.1) but with the elements of the vector $\mathbf{x}$.

identification of resampled signals: given a vector of observations $\tilde{\mathbf{z}}$, a candidate resampling factor ξ, and a particular length for the interpolation filter N_h, we want to

$$\begin{aligned} &\text{find } \mathbf{x}, \mathbf{h}, \\ &\text{subject to } \mathbf{x} \in \mathbb{R}^{N_x}, \mathbf{h} \in \mathbb{R}^{N_h}, \\ &\qquad x_{\min} \leq x_i \leq x_{\max}, \qquad i = 0, \ldots, N_x - 1, \\ &\qquad h_{\min} \leq h_j \leq h_{\max}, \qquad j = 0, \ldots, N_h - 1, \\ &\qquad -\tfrac{\Delta}{2} < \mathbf{x}_k^T \mathbf{h} - \tilde{z}_k \leq \tfrac{\Delta}{2}, \; k = 0, \ldots, \tilde{N}_z - 1. \end{aligned} \tag{5.4}$$

If the problem proves to be feasible, then the forensic analyst could also be interested in finding an estimation of both the original signal and interpolation filter that have generated the vector of observations $\tilde{\mathbf{z}}$. This can be done by considering an objective function that measures the squared error between the resampled signal $\mathbf{y} = \mathbf{X}\mathbf{h}$ and the vector of observations $\tilde{\mathbf{z}}$, leading to the following optimization problem

$$\begin{aligned} &\text{minimize } \|\mathbf{X}\mathbf{h} - \tilde{\mathbf{z}}\|_2^2, \\ &\text{subject to } \mathbf{x} \in \mathbb{R}^{N_x}, \mathbf{h} \in \mathbb{R}^{N_h}, \\ &\qquad x_{\min} \leq x_i \leq x_{\max}, \qquad i = 0, \ldots, N_x - 1, \\ &\qquad h_{\min} \leq h_j \leq h_{\max}, \qquad j = 0, \ldots, N_h - 1, \\ &\qquad -\tfrac{\Delta}{2} < \mathbf{x}_k^T \mathbf{h} - \tilde{z}_k \leq \tfrac{\Delta}{2}, \; k = 0, \ldots, \tilde{N}_z - 1, \end{aligned} \tag{5.5}$$

where $\|\cdot\|_2^2$ denotes the squared Euclidean norm. We remark that since this is a nonconvex problem, the resulting estimates $\hat{\mathbf{x}}$ and $\hat{\mathbf{h}}$ will probably correspond to local minima. Given this situation, we have first considered global optimization techniques (e.g., branch-and-bound strategies). However, we have found difficulties in handling large-scale problems (with a few hundreds of variables), thus deciding to use a local optimization method as a practical way to solve our problem.

5.3.1. Solver Based on Local Optimization

The main goal of local optimization is not the search for a globally optimal solution of the problem, but only the pursuit of a locally optimal point that minimizes the objective function within a feasible region close to it. Local optimization has been deeply studied with the aim of solving nonlinear problems, and many different algorithmic approaches can be found in the literature. In our case, we have selected an interior-point method, that is available through the function `fmincon` of MATLAB.

In general, local solvers are less computationally demanding than global ones and, consequently, they can handle in a more suitable way large-scale problems. Nevertheless, local solvers require a good starting point for the optimization variable in order to work properly. The selection of the starting point is crucial since it affects the final result provided by the solver. For instance, by choosing a

starting point that is far from a feasible region, the solver could wrongly classify a feasible problem as infeasible. This could lead the forensic analyst to wrongly declare that the observed signal was not resampled by a factor ξ when it actually was so.

In the following, we focus on the process we have designed to obtain a starting point near the feasible region of our problem (whenever the problem is actually feasible). Thus, given a vector of observations $\tilde{\mathbf{z}}$, a candidate resampling factor ξ, and the length of the filter N_h, the following steps are taken:

1. An approximation $\mathbf{x}^{(0)}$ of the original signal is first obtained (note that we use the superindex $(\cdot)^{(0)}$ to indicate starting point variables). To that end, the vector of observations $\tilde{\mathbf{z}}$ is resampled by a factor equal to the inverse of the candidate resampling factor,[4] i.e., by $\xi^{-1} = \frac{M}{L}$.

2. Since $\mathbf{z} = Q_{\Delta}(\mathbf{Xh}) \approx \mathbf{Xh}$, an approximation of the interpolation filter can also be obtained if $\mathbf{X}$ is known. For this purpose, an approximation of matrix $\mathbf{X}$, denoted by $\mathbf{X}^{(0)}$, is obtained according to (5.1) using the components of vector $\mathbf{x}^{(0)}$ (calculated in the previous step) and using the considered values for ξ and N_h.

3. After obtaining $\mathbf{X}^{(0)}$, an approximation $\mathbf{h}^{(0)}$ of the interpolation filter is constructed as $\mathbf{h}^{(0)} = \left(\mathbf{X}^{(0)}\right)^{+} \tilde{\mathbf{z}}$, where $\left(\mathbf{X}^{(0)}\right)^{+}$ denotes the Moore-Penrose pseudoinverse of matrix $\mathbf{X}^{(0)}$.

Even though the obtained starting point, composed of $\mathbf{x}^{(0)}$ and $\mathbf{h}^{(0)}$, might not strictly belong to the solution space nor satisfy all the constraints of the problem, it will be sufficiently close to a feasible region (again, whenever the problem is actually feasible) and the local solver will be able to find a feasible solution after several iterations. Notice that when the candidate resampling factor ξ does not match the actual one $\tilde{\xi}$, the obtained starting point will probably be far from the true feasible region, thus yielding an infeasible solution.

In practice, for solving the feasibility problem in (5.4), a constant objective function can be considered. As we will show in next section, this practical implementation, i.e., the local solver together with a good starting point, is able to successfully solve the feasibility problem in (5.4). Moreover, in those cases where the resulting solution set is not empty after solving (5.4), this practical approach is also able to provide locally optimal solutions by further addressing the optimization problem in (5.5).

[4]The low-pass filter used in this particular case is designed to avoid aliasing and it is constructed from a spectral Kaiser window, independently of $\tilde{\mathbf{h}}$.

Table 5.1: Details of the interpolation filters for different scenarios.

	Scenario 1 ($k_w = 2$)	**Scenario 2** ($k_w = 4$)
$\tilde{\xi} < 1$	Kaiser, $\tilde{N}_h = \tilde{M} k_w + 1$	Kaiser, $\tilde{N}_h = \tilde{M} k_w + 1$
$\tilde{\xi} > 1$	Linear, $\tilde{N}_h = \tilde{L} k_w + 1$	Cubic, $\tilde{N}_h = \tilde{L} k_w + 1$

5.4. Experimental Results

The performance analysis of the proposed technique is twofold. In the first part, synthetic signals are used to quantify the accuracy in solving the feasibility problem in (5.4), and also to measure the Mean Squared Error (MSE) of the estimates obtained through the optimization problem in (5.5). In the second part, a realistic scenario with audio signals is considered.

5.4.1. Performance Analysis with Synthetic Signals

For the evaluation of the feasibility problem in (5.4), we construct the original signal $\tilde{\mathbf{x}}$ using 8-bit precision samples gathered from a discrete uniform distribution in the interval $[0, 255]$, thus having $x_{\min} = 0$, $x_{\max} = 255$ and $\tilde{\Delta} = \Delta = 1$. We take into consideration a finite discrete set of resampling factors, obtained by sampling the interval $[0.6, 3]$ with step sizes 0.1 (from 0.6 to 2) and 0.5 (from 2 to 3). The same set is used for the true resampling factor $\tilde{\xi}$ and for checking the feasibility problem with ξ. Regarding the interpolation procedure, we employ the filters specified under Scenario 1 in Table 5.1: a linear interpolator for $\tilde{\xi} > 1$, and a low-pass FIR filter designed through a spectral Kaiser window when $\tilde{\xi} < 1$. Note that both filters have their coefficients inside the interval $[-1, 1]$, thus we assume $h_{\min} = -1$ and $h_{\max} = 1$. For simplicity, N_h is selected according to Table 5.1, but using ξ.

Taking into account all these settings and fixing the number of observations to $\tilde{N}_z = N_z = 512$, the study of the feasibility problem is carried out with the proposed local solver providing a starting point (computed as in Section 5.3.1). In Figure 5.3, the obtained results are shown in a graphical manner, where the horizontal axis represents the true resampling factor $\tilde{\xi}$, and the vertical axis contains the tested candidate resampling factor ξ. Green boxes mean that the problem has a feasible solution for the pair $(\tilde{\xi}, \xi)$, while blue ones symbolize that there exists no solution that satisfies all the constraints of the problem.

There are three important aspects that become apparent from the results shown in Figure 5.3:

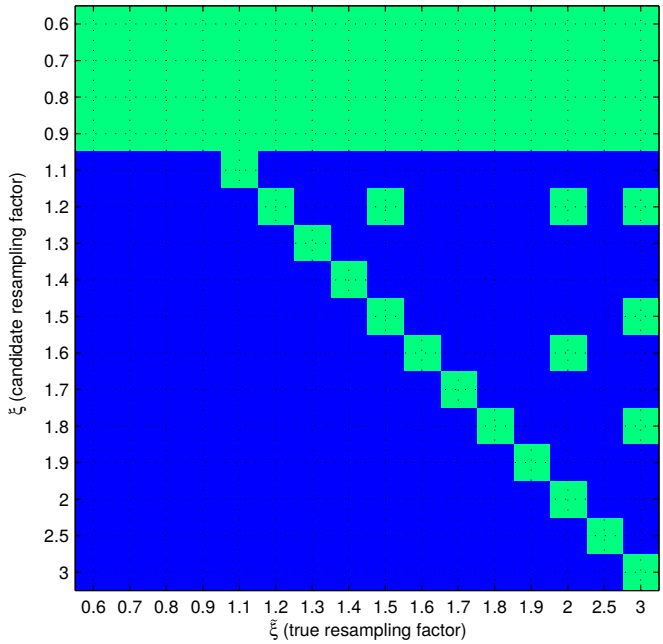


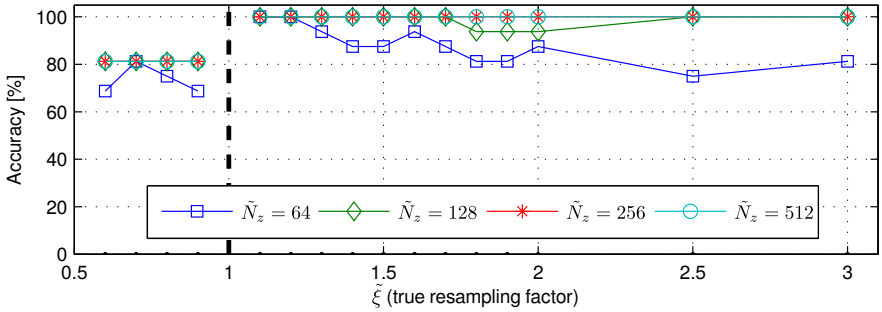
Figure 5.3: Illustrative representation of the solutions given by the local solver to the feasibility problem in (5.4), for the Scenario 1 in Table 5.1. Green boxes imply feasibility, whereas blue boxes represent infeasibility.

1. When the feasibility problem is evaluated for a candidate resampling factor $\xi < 1$, there is always a feasible solution regardless of the true resampling factor. We must remark that this is not an error due to the set-membership approach; instead, in this case there is not sufficient information (prior or observed) to rule out such ξ. In mathematical terms: the number of degrees of freedom of the problem, which is the dimension of the solution space, i.e., $N_x + N_h$, is larger than the number of observations N_z, given that $N_x = N_z \frac{M}{L}$. This problem could be overcome by adding enough a priori knowledge about the distribution of the original signal.

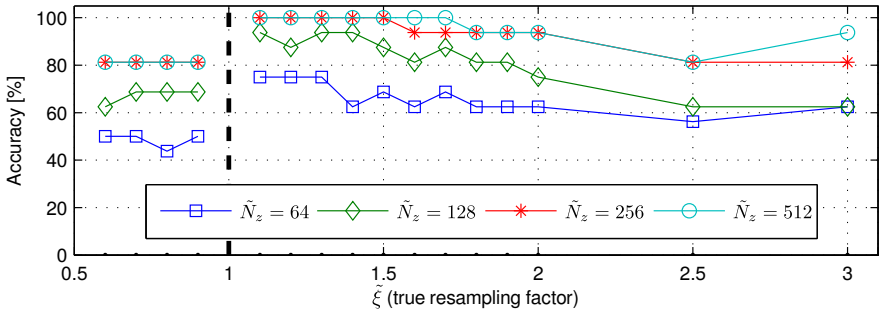
2. All the cases where the candidate resampling factor ξ coincides with the true one $\tilde{\xi}$ have always been categorized as feasible problems. This is an intrinsic property of the set-membership formulation of the problem and perhaps the most valuable feature of this method.

3. For several resampling factors $\tilde{\xi} > 1$ (e.g., $\tilde{\xi} \in \{1.5, 2, 3\}$), when $\xi > 1$ the solver is capable of finding a feasible solution, even if the true resampling factor is not equal to the candidate factor (e.g., $\tilde{\xi} = 1.5$ and $\xi = 1.2$). This is due to the existence of solutions that are theoretically feasible. However, given that the opposite case (e.g., $\tilde{\xi} = 1.2$ and $\xi = 1.5$) will not yield a feasible solution, no ambiguities are possible.

From the last point, we have found that, when an original signal is resampled by



(a) Scenario 1



(b) Scenario 2

Figure 5.4: Accuracy of the proposed approach achieved by the local solver under the two scenarios of Table 5.1, for different number of observations.

a factor $\tilde{\xi} > 1$, then the set of all the possible candidate resampling factors $\xi > 1$ that lead to a feasible solution (besides the case $\xi = \tilde{\xi}$), are:

$$\xi \in \left\{ \frac{L}{M} \ : \ \frac{L}{M} < \tilde{\xi}, (L = k\tilde{L}) \wedge (M > k\tilde{M}), k \in \mathbb{N}^+ \right\}, \tag{5.6}$$

where $L \in \mathbb{N}^+$ and $M \in \mathbb{N}^+$ must be coprime, and $\wedge$ represents the logical conjunction operation. This property also holds for the second scenario in Table 5.1.

As a conclusion, excepting the cases where the resampling factor $\tilde{\xi} < 1$, if we have a sufficiently large number of observations, then we are able to exactly match the resampling factor applied to the original signal.

5.4.1.1. Accuracy analysis for different number of observations

To quantify the performance of the method solving the problem in (5.4) we use the accuracy, defined as the following ratio:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

where TP, TN, FP, FN represent the number of true positives, true negatives, false positives and false negatives, respectively. In our problem, a true positive occurs when a feasible solution is found in (5.4) and the candidate resampling factor matches the actual one, i.e., $\xi = \tilde{\xi}$. On the other hand, a true negative takes place when no feasible solution is achieved in (5.4) and the candidate resampling factor is indeed different from the true one, i.e., $\xi \neq \tilde{\xi}$. Note that for those cases where a feasible solution is theoretically possible even if $\xi \neq \tilde{\xi}$ (i.e., for the candidate resampling factors in (5.6) and for $\xi < 1$ when $\tilde{\xi} > 1$), we will consider that a true positive case occurs if such feasible solution is found.

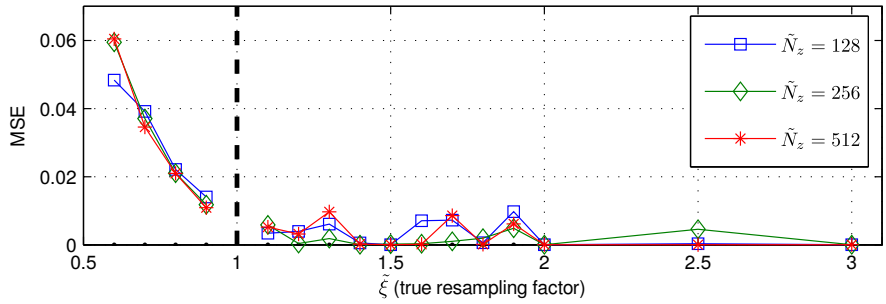
Figure 5.4 shows the accuracy obtained in the two scenarios described in Table 5.1 as a function of the true resampling factor $\tilde{\xi}$ and for different number of observations $\tilde{N}_z \in \{64, 128, 256, 512\}$. From this plot, we can observe that the accuracy improves as the number of observations increases, which is the expected behavior, since with each new piece of information the feasible set in the solution space generally gets smaller. Furthermore, by comparing the results gathered from the two scenarios, the dependence between the number of observations and the degrees of freedom of the problem becomes evident, obtaining generally worse performance in the second scenario where the order of the interpolation filters is larger. Such dependence also justifies the smaller accuracy when $\tilde{\xi} < 1$ in both scenarios.

5.4.1.2. MSE analysis for different number of observations

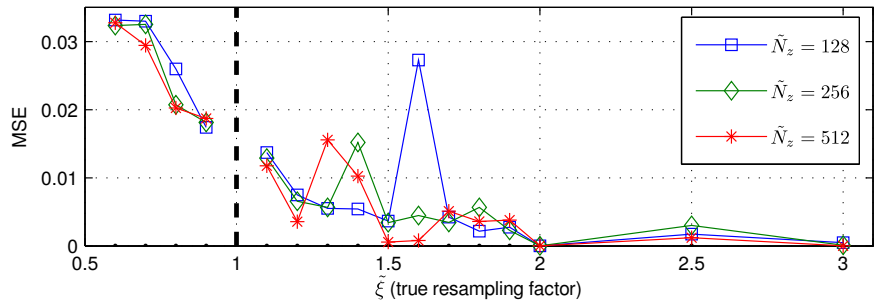
Concerning the results obtained when the optimization problem in (5.5) is solved (after having reached a solution in (5.4)), we will only show, for the sake of brevity, the empirical MSE of $\hat{\mathbf{h}}$ (i.e., $(1/N_h)\|\tilde{\mathbf{h}} - \hat{\mathbf{h}}\|^2$). Taking into account the two scenarios defined in Table 5.1, the evolution of such empirical MSE as a function of the resampling factor and for different number of observations $\tilde{N}_z \in \{128, 256, 512\}$, is depicted in Figure 5.5. As we can observe, the MSE of $\hat{\mathbf{h}}$ decreases as the resampling factor increases and, although the differences are not very significative, smaller values are generally attained when the number of observations increases. The important reduction of the estimation error for $\tilde{\xi} > 1$ is mainly due to the higher redundancy that is present on those resampled signals. The noisy shape of the MSE (e.g., $\tilde{\xi} = 1.6$ in Figure 5.5(b)) is a consequence of the local optimization performed, which in some cases converges to a local minimum that can be far from the global optimum point, but still yielding a feasible solution.

5.4.2. Performance Analysis with Real Audio Signals

For the evaluation of the set-membership approach solving the feasibility problem in (5.4) within a real scenario, we use the "Music Genres" audio database



(a) Scenario 1



(b) Scenario 2

Figure 5.5: MSE of $\hat{\mathbf{h}}$ when solving the optimization problem in (5.5) with the local solver, under the scenarios of Table 5.1, and for different number of observations.

[72], from which we take a subset of 100 uncompressed audio files with 10 different music styles. Each original audio signal is quantized to a 16-bit precision per sample, thus having $x_{\min} = 0$ and $x_{\max} = 2^{16} - 1$. For comparison, the same tests are carried out with two state-of-the-art methods: the "EM method" proposed in [8], and the "ML method" in [77] (i.e., the one explained in Chapter 4). Given that the ML method has only been defined for linear interpolators and $\xi > 1$, we consider a discrete set of resampling factors in the interval $[1.1, 2]$ (sampled with a step size of 0.1) and a linear interpolation filter as the one specified in Scenario 1 from Table 5.1.

In this case, we are interested in comparing the percentage of correct resampling factor estimation for different number of observations: $\tilde{N}_z \in \{64, 128, 256, 512\}$. In Figure 5.6, we report the obtained results with each method. The best performance is achieved by the ML method, which actually never fails with any of the considered parameters. These optimal results are possible due to the complete knowledge of the original filter used in the resampling process. Interestingly, a similar performance is obtained with the proposed set-membership approach (except for $\tilde{N}_z = 64$), where limited assumptions are made about the filter, thus increasing the applicability of the method. On the other hand, the EM method clearly exhibits some of the shortcomings mentioned in the Introduction, i.e., a high dependency on the number of observations and a worse

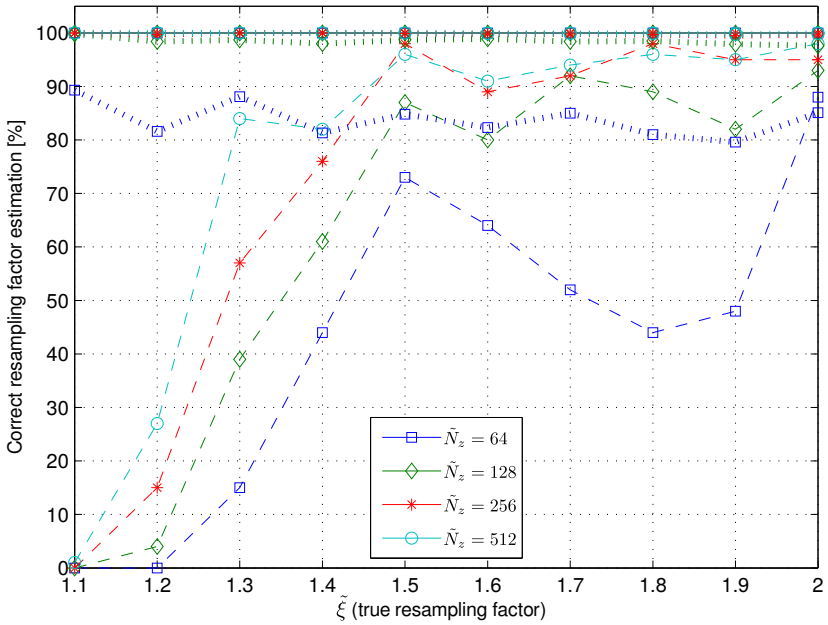


Figure 5.6: Comparison of the correct estimation percentage of the proposed set-membership technique (dotted lines) versus the ML method (solid lines) and the EM method (dashed lines).

performance for those resampling factors close to 1 due to the windowing effect. These limitations are not an issue for the proposed set-membership technique.

5.5. Conclusions

Set-membership estimation theory has proven to be a useful resource for addressing the problem of resampling factor estimation. The presented technique provides reliable solutions that do not violate any constraint of the problem, and thus are a valuable asset for a forensic analyst, who needs to provide unquestionable proofs of tampering. Moreover, the evaluation of the proposed approach in a real scenario with audio signals has demonstrated its good performance.

Chapter 6

An SVD Approach to Forensic Image Resampling Detection

On researching open questions that have arisen from the previous chapter, such as the number of observations that are needed to discard a candidate resampling factor, we have found out that image resampling detection (whenever the applied resampling factor is larger than one) can be performed via subspace decomposition. In particular, delving into the linear dependencies induced in an image after the application of an upsampling operation, we have discovered that interpolated images belong to a subspace defined by the interpolation kernel. Within this framework, by computing the SVD of a given image block and a measure of its degree of saturated pixels per row/column, we derive a simple detector, described along this chapter, which is capable of discriminating between upsampled and genuine images. Furthermore, the proposed detector shows remarkable results with blocks of small size and outperforms state-of-the-art methods.

6.1. Introduction

A first attempt to characterize the linear dependencies induced by resampling through the SVD of a resampled image has been carried out in [22] by Wang and Ping, resorting to an SVM classifier to perform the detection of resampling. A deeper understanding of the linear correlations originated locally has been described by Kirchner in [13], where a local predictor per each row/column of the image is computed. By analyzing in the frequency domain the differences between the obtained predictor coefficients, Kirchner provided an effective resampling detector, especially for downsampling. However, as indicated in [78], the frequency analysis presents some drawbacks impairing the performance of the detector when a reduced number of samples is available and also when a regular structure or a periodic pattern is present in the original image under analysis.

With regard to the last two works, and motivated by their shortcomings, here we investigate the local linear dependencies introduced once an upsampling process is applied to an image. As a result, we propose a very simple method that relies on the calculation of the SVD of a given block from an image, without resorting to an SVM classifier and being able to produce suitable results by processing blocks of small size. Note that we do not address the downsampling process in this chapter, but we further provide in Chapter 9 (cf. Section 9.1) some insights about how the proposed detector could be adapted to deal with downscaled images.

The remaining of the chapter is organized as follows: the theoretical analysis of the linear dependencies in upsampled images and the basic idea behind the proposed detector are explained in Section 6.2. The formal definition of the developed detector is tackled in Section 6.3, while the experimental results are treated in Section 6.4. Finally, conclusions are reported in Section 6.5.

6.2. Problem Modeling

Recalling the thorough description of the 2-D resampling process introduced in Section 1.2.2, let us define a digital image with a single color channel as a $P \times Q$ matrix $\mathbf{F}$ with elements $F_{p,q}$ and indices $p \in \{0, \ldots, P-1\}$ and $q \in \{0, \ldots, Q-1\}$. The values of each element $F_{p,q}$ are discrete quantities whose range is determined according to the image bit depth.

The resampling operation is assumed to be linear, so each pixel value in the resampled image $\mathbf{G}$ is computed by linearly combining a finite set of neighboring samples coming from the original image. We consider that the applied resampling factor ξ uniformly scales each dimension of the original image and we define it as $\xi \triangleq \frac{L}{M}$, i.e., the ratio between the upsampling factor $L \in \mathbb{N}^+$ and the downsampling factor $M \in \mathbb{N}^+$. The application of this resampling operation involves two main steps: the definition of the resampling grid with the new pixel locations, and the computation of the values in those locations. In a single expression, each pixel value $G_{i,j}$ of the resampled image can be obtained as follows:

$$G_{i,j} = \sum_{k=0}^{P-1} \sum_{l=0}^{Q-1} h\left(i\frac{M}{L} + \delta - k\right) h\left(j\frac{M}{L} + \delta - l\right) F_{k,l}, \tag{6.1}$$

where δ denotes a shift between the two sampling grids[1] and $h(\cdot)$ represents the impulse response of an interpolation kernel whose length or width is denoted by k_w.

[1] As noted in Section 1.2.2, in practice, the shift corresponds to $\delta \triangleq \frac{1}{2}\left(1 + \frac{M}{L}\right)$.

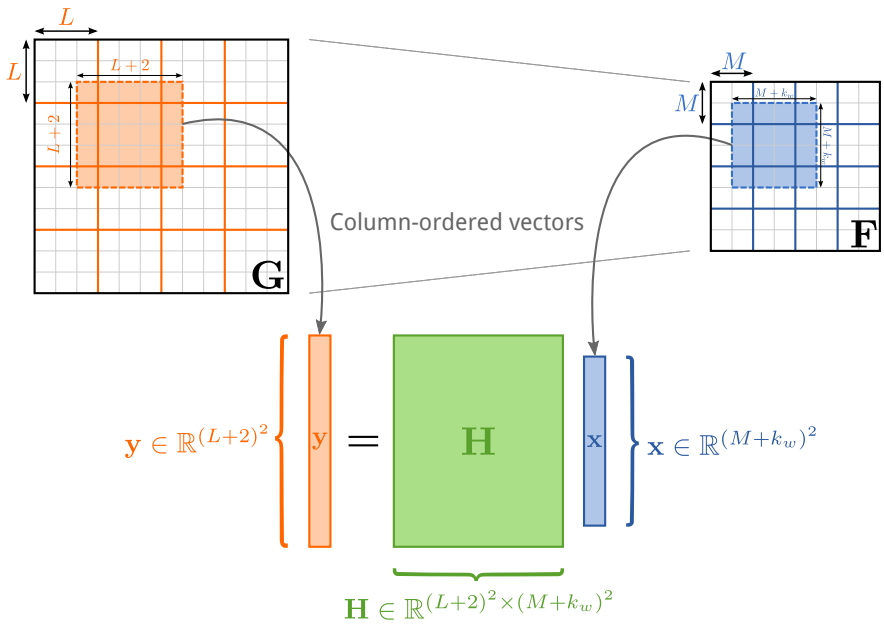


Figure 6.1: Illustrative example showing how the linear transformation is established.

The analysis in the remainder of this section is applicable to any linear kernel regardless of its width or impulse response, such as those gathered in Table 1.1; however, for non-linear kernels, the forthcoming modeling could only be understood as an approximation.

Notice that after computing all the pixels of the resampled image, its values must fit the original resolution or bit depth of the input image. Therefore, as a last step, the resampled values must be quantized to the original precision, having

$$R_{i,j} = Q_\Delta \left(G_{i,j} \right), \tag{6.2}$$

where $R_{i,j}$ denotes each element of the quantized resampled image $\mathbf{R}$ and $Q_\Delta(\cdot)$ represents a uniform scalar quantizer with step size Δ.

From the resampling operation shown in (6.1) and following the graphical example from Figure 6.1, it can be easily checked that any resampled value $G_{i,j}$ is calculated using exactly the same interpolation weights as for $G_{i+k_1 L, j+k_2 L}$ with $k_1, k_2 \in \mathbb{N}$. More precisely, for any $k_1, k_2 \in \mathbb{N}$, the column-ordered vector $\mathbf{y} \in \mathbb{R}^{(L+2)^2}$ that is built up from an $L \times L$ block of the resampled image starting at sample $G_{k_1 L, k_2 L}$ and adding a surrounding border of one pixel, is computed through the linear combination of samples from the column-ordered vector $\mathbf{x} \in \mathbb{R}^{(M+k_w)^2}$, which, in turn, is set up from an $M \times M$ block of the original image starting at sample $F_{k_1 M, k_2 M}$ and adding a border of $k_w/2$ pixels.[2] Notice that the

[2]For the sake of simplicity, we assume that the width of the kernel is an even number, as it is the case for those collected in Table 1.1.

amount of border pixels to add to each block depends on the shift δ introduced between the original and the resampled grid.

According to this observation, each column-ordered vector from the resampled signal $\mathbf{y}$ is obtained through the following linear transformation

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \tag{6.3}$$

where $\mathbf{H}$ represents the interpolation matrix containing the weights of the interpolation kernel, as depicted in Figure 6.1. From the linear transformation in (6.3), it is clear that each column-ordered vector $\mathbf{y}$ belongs to an $(M + k_w)^2$-dimensional subspace of $\mathbb{R}^{(L+2)^2}$ generated by matrix $\mathbf{H}$. We will use $\mathcal{Y}$ for denoting this subspace, which is defined as

$$\mathcal{Y} \triangleq \left\{ \mathbf{w} \in \mathbb{R}^{(L+2)^2} : \mathbf{w} = \mathbf{H}\mathbf{s}, \mathbf{s} \in \mathbb{R}^{(M+k_w)^2} \right\}.$$

However, note that due to the quantization applied after performing the resampling operation in (6.2), the observed vectors $\mathbf{z}$ of length $(L+2)^2$ (i.e., starting at sample $R_{k_1 L, k_2 L}$ with $k_1, k_2 \in \mathbb{N}$ and adding a surrounding border of one pixel), are a perturbed version of the interpolated ones. As long as the statistical distribution of the input signal is smooth and its variance is much larger than the square quantization step Δ^2, each vector $\mathbf{z}$ can be modeled as

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \boldsymbol{n},$$

where the new variable $\boldsymbol{n}$ is a random vector, whose components are i.i.d. with zero mean and variance $\sigma_N^2 = \frac{\Delta^2}{12}$ (i.e., the mean and variance of the quantization noise).

Based on this model, by stacking K vectors $\mathbf{z}$ into a $K \times (L+2)^2$ matrix, we obtain an observation matrix $\mathbf{Z}_K$ that can be represented in terms of its singular value decomposition, having

$$\mathbf{Z}_K = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{R}^{K \times K}$ is a unitary matrix whose columns are the left-singular vectors of $\mathbf{Z}_K$; $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times (L+2)^2}$ is a rectangular diagonal matrix whose diagonal elements σ_i (with $i \in \{0, \ldots, (L+2)^2 - 1\}$), are known as the singular values of $\mathbf{Z}_K$ which are sorted in descending order; and, finally, $\mathbf{V} \in \mathbb{R}^{(L+2)^2 \times (L+2)^2}$ is a unitary matrix with the right-singular vectors of $\mathbf{Z}_K$.

From this decomposition, it is expected that the $(M + k_w)^2$ dominant right-singular vectors of $\mathbf{Z}_K$ (i.e., those corresponding to the largest singular values) span the signal subspace $\mathcal{Y}$ (induced by the resampling operation), while the remaining ones span the noise subspace (caused by the rounding operation), i.e., for $i \geq (M + k_w)^2$ we have $\sigma_i \approx \sqrt{K\sigma_N^2}$, for K large enough.

In Figure 6.2, we show an example of the evolution of the singular values when matrix $\mathbf{Z}_K$ is built from a block of size 512×512 of an image resampled

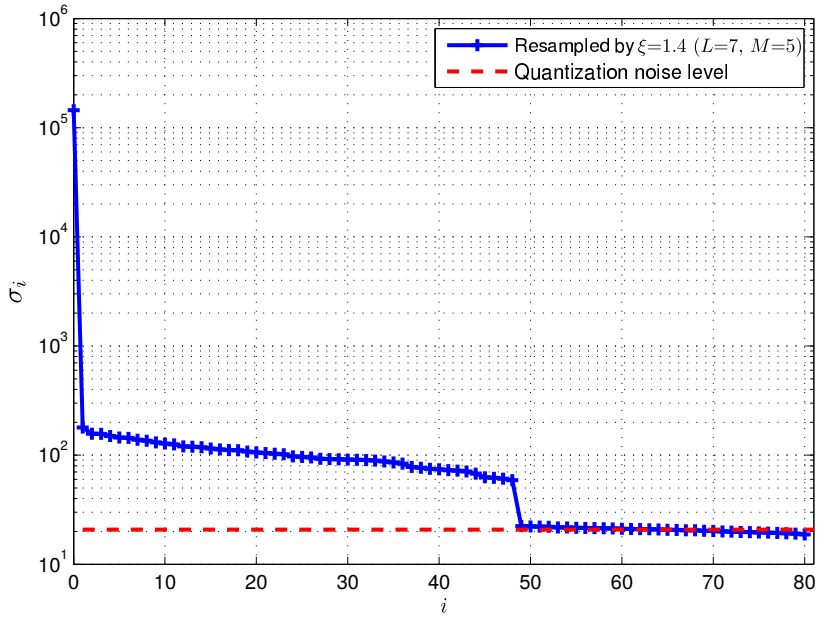


Figure 6.2: Evolution in logarithmic scale of the singular values of matrix $\mathbf{Z}_K$ built from a block of size 512×512 from the green channel of a quantized resampled image without demosaicing traces. $\xi = \frac{7}{5} = 1.4$ and linear kernel.

by $\xi = \frac{7}{5} = 1.4$ with a linear kernel ($k_w = 2$). It is easy to see how the first $(M + k_w)^2 = 49$ singular values (out of $(L + 2)^2 = 81$) have a magnitude above the quantization noise level $\sqrt{K\sigma_N^2}$ (with $K = 5184$ and $\Delta = 1$), as it was anticipated.

Note that the proposed scheme assumes L to be known at the detector; of course, this does not hold in real forensic scenarios. As a plausible solution, an iterative procedure could be considered covering all the possible values of L, but this would significantly increase the computational burden. Therefore, we simplify the process of resampling detection by directly resorting to the singular value decomposition of an image block. Then, we explore when the singular values vanish in such a manner that the signal subspace is discernible from the noise subspace.

6.2.1. Practical Solution

Let us define $\mathbf{Z}$ as a matrix gathering pixel intensity samples from a block of size $N \times N$ of a quantized resampled image $\mathbf{R}$ under test. Due to the presence of noise (e.g., rounding errors after resampling), we will initially assume that matrix $\mathbf{Z}$ has N non-zero singular values, i.e., $\mathbf{Z}$ has full rank, but at the end of this section, a discussion on how to manage rank-deficient matrices will also be introduced.

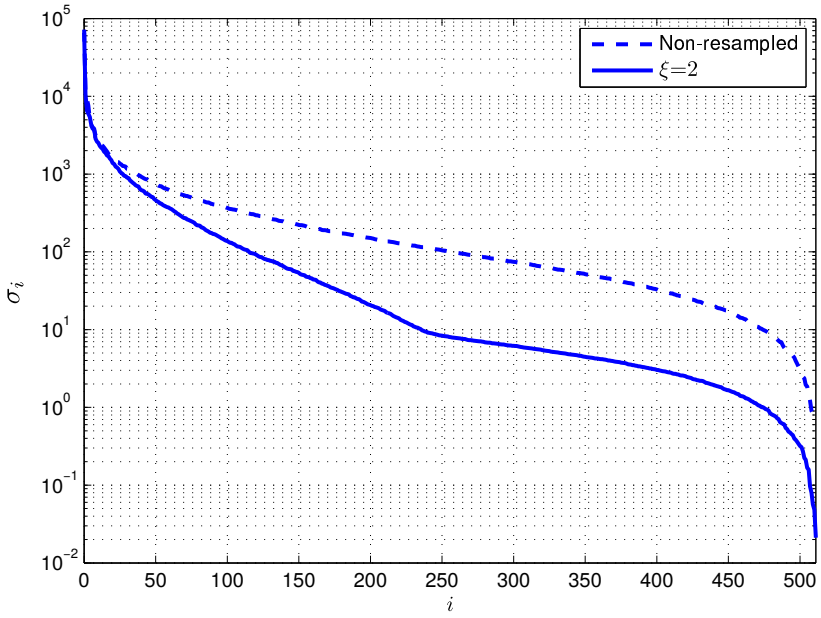


Figure 6.3: Evolution in logarithmic scale of the singular values of an image block $\mathbf{Z}$ of size 512×512. Dashed lines represent the results from a non-resampled image and solid lines correspond to an image resampled with $\xi = 2$ and linear kernel.

The rows (or columns) of $\mathbf{Z}$ can be treated as N points belonging to an N-dimensional space. In the previous analysis, we have seen throughout the application of the SVD to $\mathbf{Z}_K$ that in a resampled image, the vectors of length $(L+2)^2$ can be represented by $(M+k_w)^2$ dimensions, yielding a dimensionality reduction of a factor around ξ^2 (i.e., the applied upsampling factor in each direction). By considering now each row/column of $\mathbf{Z}$ as a vector of length N, it will be possible to represent the set of N points with a smaller number of dimensions $k \approx \frac{N}{\xi}$, since each single row/column has been resampled solely by ξ.

This will be reflected in the calculation of the SVD of $\mathbf{Z}$, where only the first $k \approx \frac{N}{\xi}$ singular values will have a considerably larger magnitude than the rest. Conversely, if we take a block $\mathbf{Z}$ from a never-resampled image, there will not exist such characteristic linear dependency between neighboring samples and all singular values will have a magnitude significantly larger than the noise level in a resampled image.

Figure 6.3 draws a comparison in terms of singular values in two different cases, i.e., when matrix $\mathbf{Z}$ is built from a block of size 512×512 from the green channel of a non-resampled image without demosaicing traces (dashed line) and from its resampled version, obtained by using $\xi = 2$ and a linear interpolation kernel (solid line). In both cases, matrix $\mathbf{Z}$ has $N = 512$ non-zero singular values, but as it can be checked in Figure 6.3, the magnitude of the singular values from the resampled image drops more sharply than the corresponding one coming from the non-resampled image. For the resampled image, the number k

of singular values significantly larger than the noise level approaches $\frac{N}{\xi}$, so in this particular case $k \approx \frac{512}{2} = 256$. This is due to the fact that approximately half of the samples of each row/column of $\mathbf{Z}$ can be computed in this resampled case as a linear combination of the remaining ones. For non-resampled images, there should only be a significant drop-off in indices close to the rank of the matrix under analysis.

Since we are assuming the applied resampling factor to be larger than one, we can state that typically the total amount of variance of the input signal explained by those singular values with indices smaller than $i \approx \frac{N}{\xi} - 1$ for any image resampled by ξ, will be larger than for a non-resampled image. As a consequence, the magnitude of the singular values at such index is also expected to be smaller. This fact indicates that a statistic accounting for the magnitude of a singular value at the correct position can be discriminative for detecting the application of a resampling operation in a given image block.

As indicated above, any matrix $\mathbf{Z}$ built from an image block is assumed to have N non-zero singular values; however, in practice, undesirable artifacts as pixel saturation may arise, thereby removing part of the noise due to rounding and yielding singular values with negligible magnitude. The presence of saturation and the possibility of having some linear dependency between samples will affect the expected evolution of the singular values of $\mathbf{Z}$, producing two possible outcomes:

1. The number of non-zero singular values is substantially smaller than N. This rarely happens unless several rows/columns of matrix $\mathbf{Z}$ are completely saturated.

2. The number of non-zero singular values is close to N, but their magnitude vanishes more sharply than usual. This takes place when linear dependencies are present among the rows/columns of $\mathbf{Z}$, and can be boosted by saturations.

Therefore, the detector will have to deal with the degree of saturation borne by any image block under analysis and it will also probably consider a means for deciding when a computed singular value is negligible. In the next section, the adopted detection strategy is described.

6.3. Proposed Detector for $\xi > 1$

From the analysis carried out on the previous section, it is clear that by exploiting the magnitude of the singular values at a certain index we can derive a hypothesis test for image resampling detection. In the definition of our hypothesis

test, the observed data consists of an $N \times N$ matrix $\mathbf{Z}$ containing a block of samples coming from one of the color channels of a digital image which may have been resampled or not. Under the null hypothesis, i.e., $\mathcal{H}_0$, we assume that the observed data $\mathbf{Z}$ has never been resampled; while, under the alternative hypothesis, i.e., $\mathcal{H}_1$, we assume that the observed data has been resampled by any factor greater than one.

The definition of the test statistic ρ will depend on the degree of saturation that the image block under analysis may have experienced, as pointed out in the last part of Section 6.2.1. By denoting γ_{row} (respectively, γ_{column}) as the quotient between the total number of saturated pixels in a block (i.e., pixels equal to 255 for an 8-bit depth image) and the number of rows (respectively, columns) that contain at least one saturated pixel, the degree of saturation s, is defined as

$$s \triangleq \frac{1}{N} \max\left\{\gamma_{\text{row}}, \gamma_{\text{column}}\right\}.$$

On the other hand, to determine which of the computed singular values are negligible, we use a tolerance level that is defined as a function of σ_0 (the largest singular value of $\mathbf{Z}$) and N, i.e., $N\epsilon(\sigma_0)$.[3] Moreover, we define a variable $r \leq N$ that represents the total number of singular values above this tolerance level. Accordingly, the proposed test statistic is:

$$\rho \triangleq \begin{cases} 0, & \text{if } r < 0.1N, \\ \log\left(\sigma_{\nu - \lfloor 0.05N \rfloor - 1}\right), & \text{if } s \geq 0.45 \text{ and } r > 0.95N, \\ \log\left(\sigma_{\nu-2}\right), & \text{otherwise,} \end{cases} \tag{6.4}$$

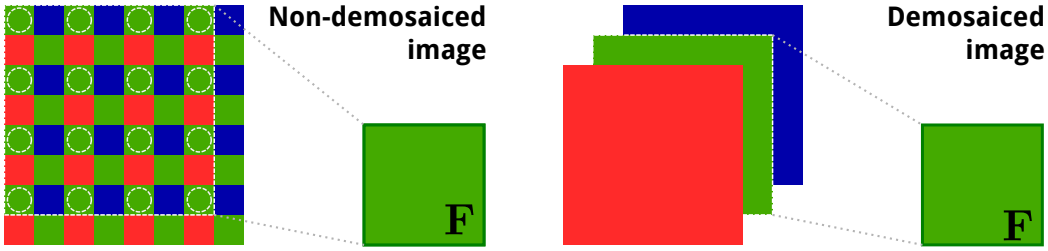
where $\nu = \left\lfloor \frac{r}{\xi_{\min}} + 0.5 \right\rfloor$ represents the rounded version of the maximum number of significant dimensions that could be achieved by a resampled image with any $\xi \geq \xi_{\min}$. Hence, $\xi_{\min}$ is the minimum resampling factor that can be detected by our detector. Notice that the first two cases contemplated in (6.4) have been heuristically derived and are set to avoid the two effects caused by pixel saturation discussed at the end of Section 6.2.1.

Assuming all the particularities for obtaining the test statistic, we expect to find larger values of ρ for non-resampled images, thus accepting the hypotheses according to the following conditions:

$$\begin{aligned} \mathcal{H}_0 : &\quad \rho > T, \\ \mathcal{H}_1 : &\quad \rho \leq T, \end{aligned}$$

where T is a predefined threshold. Several experiments are performed next to study the validity of the proposed approach.

[3]Function $\epsilon(x)$ represents the function `eps` in MATLAB, measuring the positive distance from $|x|$ to the next larger in magnitude floating point number of the same precision as x.



(a) Without demosaicing traces (b) With demosaicing traces

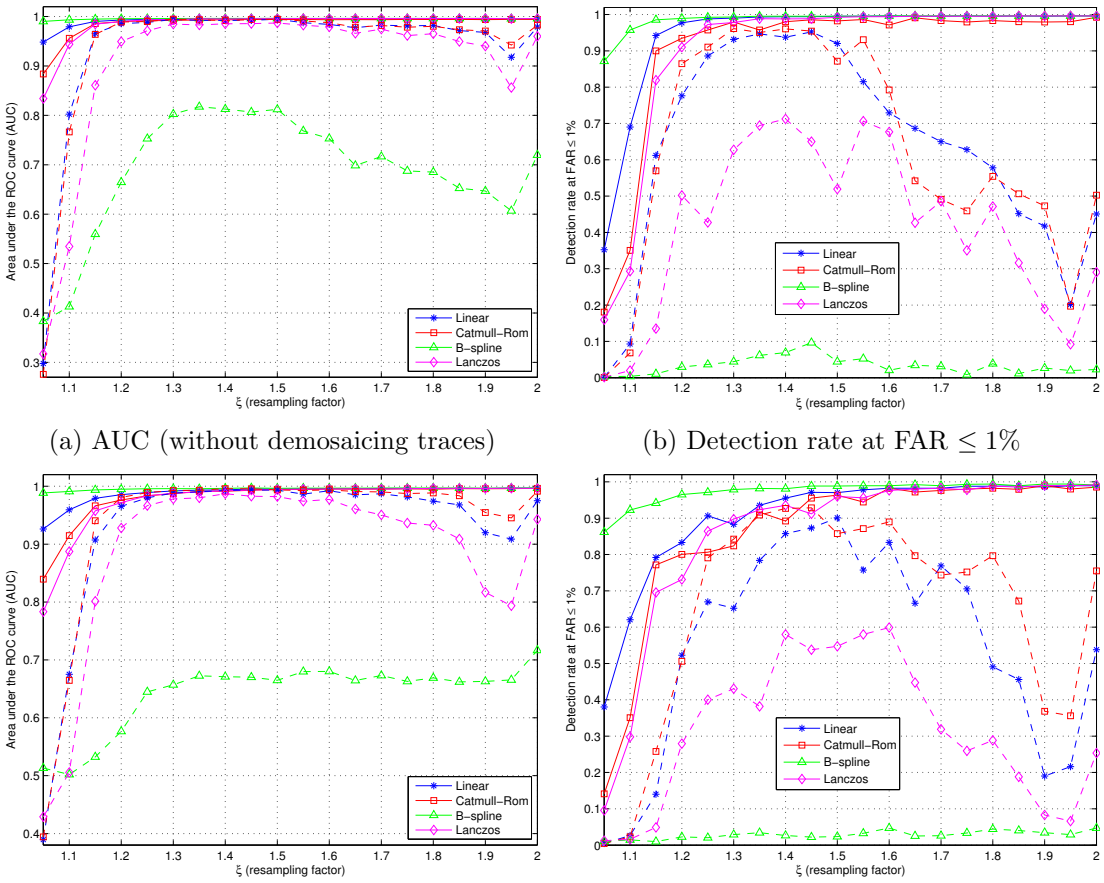
Figure 6.4: Schematic representation of how the non-resampled image **F** is built.

6.4. Experimental Results

The designed detector is tested over all the uncompressed images belonging to the Dresden Image Database [79] and stemming from Nikon cameras (a total of 1317 images). To perform each full-frame resampling operation, the image processing tool `convert` from ImageMagick's software is used. As interpolation kernels, we select those that are commonly available in any image processing tool, namely: *Linear*; from the family of cubic filters we choose *Catmull-Rom* and *B-spline*; and, finally, a three-lobed *Lanczos*-windowed kernel. The employed discrete set of resampling factors is defined in the interval $[1.05, 2]$ (sampled with a step size of 0.05), given that these are the most appropriate upsampling factors to avoid the introduction of visible distortions.

Since our main objective is to unveil tampered regions (which might be small) through the detection of resampling inconsistencies, we are interested in studying the achieved performance of our detector with blocks of small size, thus leading us to process $N \times N$ image blocks $\mathbf{Z}$ with $N = 32$. The analysis of resampling traces is then carried out by taking the center 32×32 block of the green channel from each image under study. The evaluation of the performance of the proposed detector is conducted in terms of AUC (Area Under the ROC Curve) and detection rate at a fixed False Alarm Rate (FAR), i.e., concretely at FAR $\leq 1\%$. All results are compared with the detector proposed by Kirchner in [13], because this method outperforms Popescu and Farid's detector [8], which is usually considered to be the most reliable detector.

The performance analysis is twofold: first, images without demosaicing traces are processed, and then, demosaiced images are tested. The process for obtaining non-resampled images without demosaicing traces consists in getting access to the output of the camera sensor (through the image processing tool `dcraw`) and then picking always the same-positioned green pixel from the two available samples in each 2×2 Bayer pattern, as illustrated in Figure 6.4(a). On the other hand, non-resampled images with demosaicing traces are obtained by extracting directly the green channel of the demosaiced image under analysis, as depicted in Figure 6.4(b). In each of these cases, both detectors must also be applied on



(a) AUC (without demosaicing traces)

(b) Detection rate at FAR $\leq 1\%$

(c) AUC (with demosaicing traces)

(d) Detection rate at FAR $\leq 1\%$

Figure 6.5: Evaluation of the proposed detector (solid lines) against Kirchner's detector (dashed lines) in terms of AUC and detection rate for blocks of size 32×32. The first row contains the results from images without traces of demosaicing, while the second row is for images with demosaicing traces.

all the non-resampled images in order to fix the detection thresholds (i.e., T). For our test statistic we take $\xi_{\min} = 1.05$, and a neighborhood size $K = 3$ for Kirchner's detector, as specified in [13].

The first row of Figure 6.5 shows the performance of the proposed approach when testing images without demosaicing traces. From these results, we can state that our method shows better performance with B-spline and Linear interpolation kernels than with Catmull-Rom and Lanczos, which commonly get the worst results. Our detector presents some difficulties with resampling factors close to one, i.e., for $1.05 \leq \xi \leq 1.2$, whereas less issues arise when the resampling factor approaches 2. Although not being reported, additional experiments have been performed increasing the size of the block (e.g., with $N = 128$), obtaining values of AUC ≥ 0.998 for all tested filters and $\xi \geq 1.1$.

An interesting aspect is that our detector shows a strong gain with respect to

Kirchner's when images are resampled with the B-spline kernel, regardless of the value of ξ. For instance, in Figure 6.5(b), Kirchner's detection rate is below 0.1 for all tested ξ, while our detector shows a detection rate almost always above 0.9 (excepting $\xi = 1.05$). Therefore, the proposed SVD-based analysis has proven to be very convenient for B-spline resampling detection. A second stimulating feature is that our detector needs a very small set of samples (i.e., 32×32 pixels) to work remarkably well, while this particular size starts to be a problem for Kirchner's detector.

The second row of Figure 6.5 collects the results arising from images with demosaicing traces. By comparing the achieved outcomes in this case with respect to the previous ones (i.e., without traces of demosaicing), it becomes apparent that our detector works better when it has to distinguish purely non-resampled (i.e., non-demosaiced) images against their upsampled version. The reason is that when non-resampled images exhibit demosaicing traces, there exist unavoidable linear dependencies which affect the expected value of the statistic ρ for genuine images. Usually, these linear correlations caused by the demosaicing process are not so strong as the ones introduced by the resampling operation (mainly because current demosaicing algorithms are adaptive and, commonly, non-linear), but this will harm to some extent the idea behind the use of the SVD as a means to distinguish linearly correlated data against uncorrelated data.

Apart from this global lost in performance, the behavior of our detector is almost identical to the one discussed for images without demosaicing traces. In general, all the experimental results show that our detector is a reliable solution for image resampling detection.

6.5. Conclusions

In this chapter, a simple strategy for resampling detection has been derived. The proposed detector only needs to compute the SVD of a given image block and a measure of its degree of saturated pixels per row/column, for discerning upsampled images from genuine ones. The achieved performance is promising and when compared with Kirchner's state-of-the-art method, our detector outperforms it.

Part II

Forensic Analysis of Video Sequences

Chapter 7

Detection of Video Double Encoding with GOP Size Estimation

Video forensics is an emerging discipline that aims at inferring information in a blind fashion about the processing history undergone by a digital video. Currently, most of the available techniques follow concepts inherited from image forensics or work under restrictive assumptions such as the use of a fixed quantization structure which is rarely adopted in real scenarios due to bandwidth or storage constraints. However, in this chapter, we introduce a new forensic footprint, whose origin is directly related to the encoding strategy followed by a video encoder when taking decisions on which coding mode is the most convenient (in terms of quality and bitrate) for compressing a particular macroblock. Based on the variation of such footprint when double compression is performed, we propose a method for detecting whether a video has been encoded twice and, if that is the case, we estimate the size of the Group Of Pictures (GOP) employed during the first encoding. As shown in the experiments, the derived approach proves to be very robust even under realistic settings (i.e., when encoding is carried out using typical compression rates), that are barely treated by existing techniques.

7.1. Introduction

Edition and composition of video sequences is nowadays easier due to the availability of a large number of video editing software tools. As pointed out in Section 1.3.1, these tools do not work directly on the compressed domain, but on the recovered spatio-temporal domain. Therefore, when editing a single compressed video, an initial decoding step and a posterior re-encoding process are

required. In most cases, the second encoding will leave a characteristic footprint in the resulting video sequence that can be detected and further analyzed to extract information about the processing history of the original video sequence.

Most of the related works investigating footprints left by this double encoding process have been covered in Section 1.3.3. For instance, by relying on the resulting double quantization, authors in [43] propose a method to identify tampered regions on MPEG-2 video sequences with only I-frames, i.e., in a similar way as with digital images. The same authors propose in [41] to take into account the information about the motion error when P-frames are used, as a means to detect deletion or addition of frames.

Although these two techniques make the localization of tampered regions possible (either in the spatial or temporal domain), they do not allow to acquire knowledge about the origin of a given video stream. In this sense, some works have been developed trying to retrieve information about the processing history of a compressed video. As an example, estimation of video coding parameters has been addressed in [80], providing a method to estimate MPEG-2 settings from the decoded video stream. Valenzise et al., in [81], later extend this work to H.264 video, estimating the quantization parameter and motion vectors from decoded frames.

Concerning the first steps in the processing history of a digital video, Bestagini et al. have proposed an approach in [39] for the identification of the first codec applied (out of three possible ones) to a video sequence that has been doubly encoded. This method works by recompressing the video under analysis with the three possible codecs and computing a similarity measure between the two sequences. Based on the same approach, Luo et al. [38] propose a method for detecting double encoding in MPEG-2 compressed videos, by recompressing a given sequence with different GOP lengths and then performing an analysis of blocking artifacts. The main drawback of all the proposed techniques for detection of double encoding is the way they are affected by the second encoding, since their performance drops very rapidly as the strength of the last compression increases.

Motivated by these shortcomings and with the aim of generalizing the double encoding detection to a scenario with several codecs, different GOP sizes, and distinct target bitrates, we propose to use a robust and very distinctive footprint that arises from the second encoding of a video sequence. Assuming that a different GOP size is applied during the second compression, an anomalous variation of the macroblock prediction types takes place on the P-frames that were originally encoded as I-frames in the first encoding. An advantage of this Variation of Prediction Footprint (VPF) is that its presence can be unveiled by partially decoding the video, without requiring subsequent recompressions as in [38, 39]. Furthermore, given that the VPF becomes apparent only in P-frames that were encoded as intra in the first encoding, we also describe a method to estimate the

length of the GOP used in the first compression.

GOP size estimation is not only an important step toward assessing the processing history of a digital video, but can also act as a catalyst for further forensic analysis, e.g., tampering detection. Although in the succeeding sections we are not targeting video doctoring detection, in the next chapter we will see how the estimation of the GOP size can help to localize intra-frame forgeries in MPEG-2 videos.

In the next section, we introduce the considered scenario for double encoding detection, analyzing why the VPF appears. In Section 7.3, we explain how this particular footprint can be measured and discuss our method for the estimation of the first compression GOP size. The experimental results for validating the detection accuracy and the performance of the estimator are presented in Section 7.4. Finally, Section 7.5 concludes this chapter.

7.2. Preliminaries and Problem Statement

In this chapter, the forensic analysis of compressed video sequences is investigated on three major video coding standards, namely MPEG-2 [24], MPEG-4 [25] and H.264 [26]. From the description in Section 1.3.2, we have seen that each of these standards defines its own coding characteristics, but their design is built over a common block-based hybrid video coding approach, thus sharing several syntactic features. In this sense, the following analysis will not focus on a particular compression standard, since the footprint we introduce here relies on principles that are valid for the three mentioned standards.

For the sake of simplicity, we do not contemplate the use of B-frames in this work. Therefore, we constrain the compression to be performed according to the baseline profile for H.264 and to the equivalent simple profile for MPEG-2 and MPEG-4. These profiles support only I-frames and P-frames, along with three main types of macroblocks: intra-coded macroblocks (I-MB), predictive-coded macroblocks (P-MB) and skipped macroblocks (S-MB). In particular, the macroblocks of an I-frame can only be encoded by means of intra coding modes (i.e., I-MB), while in P-frames any of the available coding modes can be used, thus containing macroblocks of any type (i.e., I-MB, P-MB, or S-MB). Every standard proposes its own coding modes for each type of frame with the final goal of increasing the coding efficiency and some of them are collected in Table 7.1. Note that even if the same name is used, the particular implementation of each mode could be different from one standard to another, but maintaining a similar functionality. Other existing modes and sub-modes are not presented for brevity. With the aim of easily identifying a coding mode applied to a macroblock (independently of the standard), the last row of Table 7.1 provides the three types of

Table 7.1: Available coding modes for I- and P-frames for each standard.

Modes / Standards	Intra Coding	Inter Coding	
MPEG-2	INTRA-16×16	SKIP	INTER-16×16
MPEG-4	INTRA-16×16	SKIP	INTER-16×16
			INTER-8×8
H.264	INTRA-4×4 INTRA-16×16	SKIP	INTER-16×16
			INTER-16×8
			INTER-8×16
			INTER-8×8
Macroblock type	I-MB	S-MB	P-MB

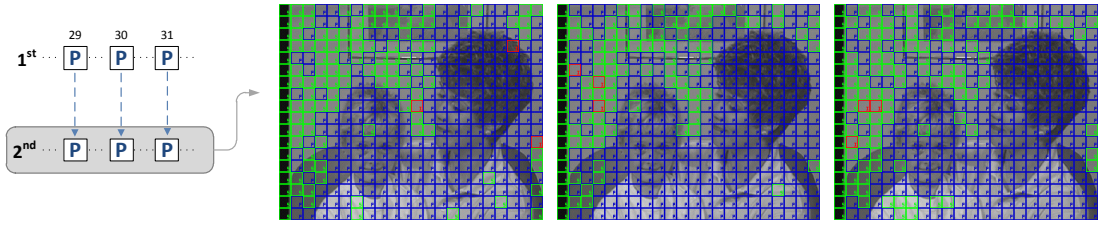
macroblocks we study along this chapter. The background color of each cell will allow the visual classification of these types of macroblock in further illustrative examples. Finally, we will assume that the GOP structure is fixed for each video sequence and for the extraction of the VPF we will only process the luminance component.

Let us now consider the following scenario. In the first place, during the capture of a scene, a first compression is performed with an arbitrary GOP size, denoted by G_1, and a fixed constant bitrate, represented by B_1. Then, after the reconstruction of the video sequence in a raw uncompressed video format, a second compression (temporally aligned with the first one) is carried out on the uncompressed sequence, but with a different GOP size, i.e., G_2 such that $G_2 \neq G_1$, and a fixed constant bitrate, i.e., B_2, that can be equal or different from the one used in the first compression. Assuming this double encoding framework, a specific variation of the number of I-MB and S-MB shows up in the P-frames previously encoded as I-frames in the course of the first compression.

To get a better understanding on this change of macroblock types, we first describe an example where this variation does not take place and then, we analyze the opposite situation. Figure 7.1 refers to the first case where a double encoding with $G_1 = 45$ and $G_2 = 50$ is taken into account. The conversion between the types of frames for the indices 29, 30 and 31[1] is illustrated in Figure 7.1(a), and as we can see, each P-frame in the first compression is encoded again as a P-frame. From Figures 7.1(b)-(d), where the macroblock types for the doubly compressed P-frames are overlaid, we cannot notice a clear variation of the number of each type of macroblock between the 3 depicted frames.

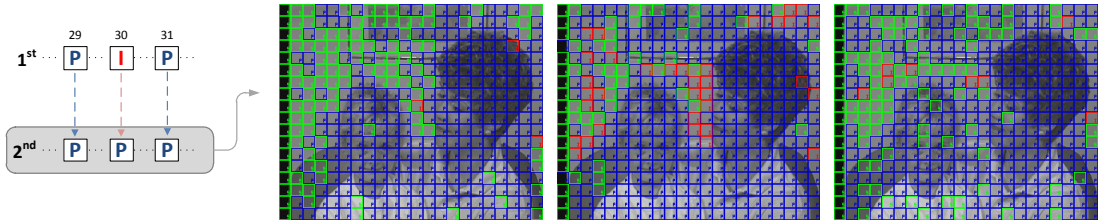
Nevertheless, if we just change the GOP size in the first compression to $G_1 = 30$ and we repeat the same double encoding, we get the results shown in Figure 7.2. In this case, as it is depicted in Figure 7.2(a), the frame with index

[1]Note that we assume that the frame indices start counting from 0.



(a) $G_1 = 45$, $G_2 = 50$ (b) Frame 29 (c) Frame 30 (d) Frame 31

Figure 7.1: Example where the VPF is not present. Leftmost picture shows the types of frames with indices 29, 30 and 31 for both compressions. The remaining three pictures represent the macroblock types for each frame. The color of each macroblock is established according to the last row of Table 7.1. Both first and second encodings are carried out using the `x264` library, with a QP fixed to 20.



(a) $G_1 = 30$, $G_2 = 50$ (b) Frame 29 (c) Frame 30 (d) Frame 31

Figure 7.2: Example where the VPF becomes apparent in frame 30. The above details in Figure 7.1 also apply to this figure.

30 is converted from an I-frame to a P-frame in the second compression. Checking the corresponding macroblock types for the frame 30 in Figure 7.2(c), we can easily appreciate a noticeable increase of I-MB and a considerable reduction of S-MB. Hence, the VPF takes place in the frame number 30. Since until this frame nothing changes from the previous case, we get exactly the same macroblock types for frame 29, and, as it can be observed, the number of each macroblock type in the frame 31 returns to its normal value, even if the underlying grid has changed.

The explanation of this effect is based on the different way an I-frame is encoded with respect to a P-frame. Generally, the quantization matrix or the quality factor for encoding an I-frame differs from the one considered for a P-frame because I-frames are used directly or indirectly as a reference for encoding several future frames. Besides, the following effects are observed:

- Change of P-MB or S-MB in homogeneous regions into I-MB. In general, the use of I-MB in a P-frame is intended for encoding more efficiently a region where there is not a good match in previous reference frames, like a new uncovered region.

In this case, the compression of a reconstructed I-frame with a P-frame (whose reference frame will probably not be so correlated with this uncompressed frame) will lead to a less efficient encoding in general. However, if the changes introduced by the I-frame are small in homogeneous regions (for instance, like a change in the DC component of a whole block), then those blocks will be more efficiently coded as I-MB than P-MB where at least a motion vector should be considered and more bits would be needed. That is the main reason why I-MB appear in smooth regions.

- Change of S-MB in static regions into P-MB. The use of skipped macroblocks is very likely for any encoder given that neither residual information nor motion vector are needed and a lot of bits are saved.

  Nevertheless, in the case we are studying, when a reconstructed I-frame comes into play during the encoding of a P-frame, small variations are introduced in static regions with respect to its reference frame and, thus, the use of S-MB is no longer possible. Consequently, P-MB must be used instead for satisfying the perceptual requirements.

As we stated earlier, even if each standard performs prediction and quantization in a different way, the common characteristics shared by the codecs make them agree with the behavior described above. Of course, the presence of VPF will also depend on the particular implementation of each codec, but since the main objective of any implementation is to reduce the bitrate according to a predefined quality, the observed behavior should also be consistent with any specific implementation.

As a conclusion, if we can detect those variations in the number of I-MB and S-MB, then we will be able to detect if a double encoding of the same sequence has been carried out and, if this is the case, we have a way to estimate the size of the first GOP from those variations.

7.3. Measuring the VPF

In this section we show how the VPF can be used to detect double encoding and to estimate the GOP size of the first compression from a given video sequence. The method we introduce is essentially based on two steps: firstly, the frames showing the VPF are located and the strength of the footprint is measured; secondly, provided that the obtained signal should show relevant peaks where I-frames of the first compression were located, a periodicity analysis is carried out.

In the rest of this section, the following notation is used: for a given video sequence $\underline{\mathbf{x}}(n)$, with $n = 0, \ldots, N-1$, being N the total number of frames, we

denote with $i(n)$ and $s(n)$, respectively, the number of I-MB and S-MB that are present in the n-th frame. We also recall that G_1 and G_2 are the GOP sizes used for the first and the second compression, respectively.

7.3.1. Peak Extraction

In this first phase, we jointly analyze the two signals $i(n)$ and $s(n)$. From Section 7.2, we know that the number of I-MB increases at the same time the number of S-MB decreases in those P-frames of the video under analysis that were originally coded as I-frames in the first compression. Note that signal $i(n)$ cannot be directly processed as it is, given that it also keeps track of all the I-MB used in the encoding of I-frames from the second encoding. To avoid peaks in $i(n)$ that are not related to the effect of the first encoding, we simply remove those values that are periodically located at the beginning of a GOP. Since G_2 is known, we substitute the elements of $i(n)$ at multiples of G_2 by the average value of its adjacent neighboring samples, i.e.,

$$i(k\mathrm{G}_2) = \frac{i(k\mathrm{G}_2 - 1) + i(k\mathrm{G}_2 + 1)}{2}, \quad \forall k \in \left\{0, \ldots, \left\lfloor \frac{N-1}{\mathrm{G}_2} \right\rfloor\right\}.$$

Notice that when values at the edges are not available, such as $i(-1)$ for $k = 0$ or $i(N)$ for $k = (N-1)/\mathrm{G}_2$, the closest sample value is directly assigned to $i(n)$. For the sake of clarity, we will denote by $\mathcal{P}$ the set of frames where the effect described in Section 7.2 is present, having

$$\begin{aligned}\mathcal{P} \triangleq \{n \in \mathbb{N} : (i(n-1) < i(n) \ \wedge \ i(n) > i(n+1)) \wedge \\ (s(n-1) > s(n) \ \wedge \ s(n) < s(n+1))\},\end{aligned}$$

where $\wedge$ represents the logical conjunction operation. Based on the above set, we define a new vector that quantifies the strength of the effect for every frame $n \in \{0, \ldots, N-1\}$ as follows

$$v(n) \triangleq \begin{cases} E(n), & \text{if } n \in \mathcal{P} \\ 0, & \text{otherwise} \end{cases}, \tag{7.1}$$

where $E(n)$ measures the energy of the effect in the n-th frame, being defined as

$$\begin{aligned}E(n) \triangleq &\left|(i(n) - i(n-1))(s(n) - s(n-1))\right| \\ &+ \left|(i(n+1) - i(n))(s(n+1) - s(n))\right|.\end{aligned}$$

This measure follows a simple intuition: first, we envision an increase in the number of I-MB together with a decrease in the number of S-MB and, then, we expect a decrease in the number of I-MB along with an increase in the number of S-MB, reaching the common proportion of these macroblock types in P-frames. Therefore, by taking the product of the variations of $i(\cdot)$ and $s(\cdot)$ we measure the strength of the sudden change in the prediction types, i.e., we quantify the VPF.

7.3.2. Analysis of Periodicity

The second phase of the proposed scheme consists in investigating the periodicity of the extracted feature. If no periodic behavior is detected, we can classify the video as singly encoded; conversely, if a periodicity is present, then it will allow us to estimate G_1.

Usually, the periodicity of a signal is well-exposed using its frequency representation, e.g., taking its Fourier transform. However, this approach is well-suited for cases where many periods of the signal are available, otherwise the resulting representation is noisy and periodicity estimation is inaccurate. On the other hand, we want our method to work also with a limited number of frames, so the frequency representation is not the best tool for our task. For these reasons, we propose a simple yet effective strategy for estimating the periodicity of peaks in $v(n)$, that is based on two steps: candidate GOP selection, and candidate evaluation.

The candidate GOP selection aims at determining a set of possible values for G_1. Since we are searching in a set of integer values $\mathcal{P}$, an element generating subsequent multiples of itself, it makes sense to restrict the search to the set of the Greatest Common Divisors (GCD) between all possible couples of elements of the sequence. Therefore, we define the set $\mathcal{C}$ of candidate GOPs as

$$\mathcal{C} = \{c \in \mathbb{N} : c = \mathrm{GCD}(n_1, n_2), \forall\ n_1, n_2 \in \mathcal{P}\}.$$

Notice that evaluating $\mathcal{C}$ requires at most N^2 runs of the GCD algorithm, whose complexity is quadratic in the number of base-10 digits of its argument ($\lceil \log_{10} N \rceil$ at most, in our case). However, provided that the signal $v(n)$ is typically sparse (in the experiments presented in Section 7.4, $\approx 90\%$ components are null on average), the practical computational effort is surely affordable.

In the GOP estimation stage, each candidate value $c \in \mathcal{C}$ is associated with a fitness value $\phi : \mathcal{C} \to \mathbb{R}$, that measures how well the choice of c models the periodicity of the signal $v(n)$. Before giving the formal definition of $\phi(c)$, we briefly develop the intuition behind this measure. Due to content related issues, like sudden changes of scene or strongly textured regions, the signal $v(n)$ could contain some noisy components, or could be missing some expected peaks in multiples of G_1. With this in mind, it is essential to define a fitness measure that takes into account, for each candidate value $c \in \mathcal{C}$, the following aspects:

1. The energy of peaks that are located at multiples of c, given by

$$\phi_1(c) = \sum_{k=0}^{\left\lfloor \frac{N-1}{c} \right\rfloor} v(kc).$$

2. The absence of peaks that would be expected in multiples of c, quantified as

$$\phi_2(c) = \beta|\mathcal{A}_c|, \quad \text{with} \;\; \mathcal{A}_c \triangleq \left\{kc : k = 0, \ldots, \left\lfloor \tfrac{N-1}{c} \right\rfloor\right\} \setminus \mathcal{P},$$

   where $|\cdot|$ stands for the cardinality of a set, $\setminus$ represents the set-theoretic difference, and β is a penalization factor for missing peaks, that can be taken as $\beta \triangleq 0.1 \max_n v(n)$.

3. The energy of the most relevant periodic component with a period smaller than c, defined as

$$\phi_3(c) = \max_{i=1,\ldots,c-1} \sum_{k=0}^{\left\lfloor \frac{N-1}{c} \right\rfloor - 1} v(kc + i).$$

Then, we combine these three measures to define the function $\phi(c)$ as

$$\phi(c) = \phi_1(c) - \phi_2(c) - \phi_3(c), \tag{7.2}$$

where it is evident that ϕ_2 and ϕ_3 act as a penalization for the candidate c. Once the fitness of every candidate in $\mathcal{C}$ has been evaluated, we can classify the video as singly or doubly encoded and, in the latter case, provide the estimate for G_1. The video $\underline{\mathbf{x}}(n)$ is assigned to a class with the following rule:

$$C(\underline{\mathbf{x}}) = \begin{cases} 1, & \text{if } \max_{c \in \mathcal{C}} \phi(c) > T_\phi \\ 0, & \text{otherwise} \end{cases}, \tag{7.3}$$

where T_ϕ is a threshold, $C(\underline{\mathbf{x}}) = 1$ accounts for videos classified as doubly encoded, and $C(\underline{\mathbf{x}}) = 0$ stands for videos classified as singly encoded. Whenever a video is classified as doubly encoded, the estimate of G_1 is

$$\hat{\mathrm{G}}_1 = \arg\max_{c \in \mathcal{C}} \phi(c). \tag{7.4}$$

7.4. Experimental Results and Discussion

The performance of the proposed approach for double encoding detection and GOP size estimation is evaluated in this section. A realistic experimental setup, which is often challenging for video forensics, is designed for conducting all the tests. We build the datasets for our experiments using 14 video sequences with CIF resolution, i.e., 352×288 pixels, that are available in YUV-uncompressed format.[2] Given that these sequences have different lengths, we always limit ourselves

[2]Freely available at this website: `http://trace.eas.asu.edu/yuv`
Chosen sequences are: *akiyo*, *bridge-close*, *bridge-far*, *coastguard*, *container*, *foreman*, *hall*, *highway*, *mobile*, *news*, *paris*, *silent*, *tempete*, and *waterfall*.

Table 7.2: Parameters for creating doubly encoded sequences.

Parameters	**1st encoding**	**2nd encoding**
Encoder	{MPEG-2, MPEG-4, H.264}	{MPEG-2, MPEG-4, H.264}
Bitrate (kbps)	$\{100, 300, 500, 700\}$	$\{100, 300, 500, 700\}$
GOP size	$\{10, 15, 30, 40\}$	$\{9, 16, 33, 50\}$

to process only their first 250 frames (that is, 10 seconds of video at 25 fps), in order to investigate the reliability of the proposed approach in presence of short clips. Furthermore, in all the experiments, video encoding is performed specifying a target constant bitrate, i.e., without using a fixed quantization structure, because this is the typical encoding setting in a realistic scenario. As it was mentioned in Section 7.2, adaptive GOP structures are not tackled in this work. For all the tests, we have used the `libavcodec` and `x264` libraries (through `FFmpeg`) to encode/decode all the video sequences.

Because we propose to use the VPF both for double encoding detection and GOP size estimation, we split the experiments in two parts; this choice also accounts for the different nature of these tasks, since detection and estimation methods need different evaluation criteria.

7.4.1. Double Encoding Detection

To test the discrimination capability of the proposed approach, we use the mentioned 14 raw sequences to create a dataset consisting of:

- 672 singly encoded videos, by using all combinations of encoders and parameters in the rightmost column of Table 7.2;
- 672 doubly encoded videos, by randomly selecting 48 joint configurations of 1st and 2nd encoding (from those collected in Table 7.2), per video sequence.

Since the proposed detection method relies on a threshold-based rule (see Eq. (7.3)), we use Receiver Operating Characteristic (ROC) curves to evaluate its performance: we report in Figure 7.3 the ROC of the proposed method on the whole dataset (dashed lines), and the ROCs obtained separately, differentiating the encoder employed for the second compression (which, of course, is known to the analyst). It is worth noting that when the second encoding is carried out using H.264 (as we have seen, the most commonly used nowadays), the detector yields its best performance (94% detection rate for a false positive rate of 5%). In fact, while the VPF will rarely appear in singly encoded sequences, independently from the codec being used, it cannot be taken for granted that it will show up clearly in a doubly encoded video: when the quality of the second compression

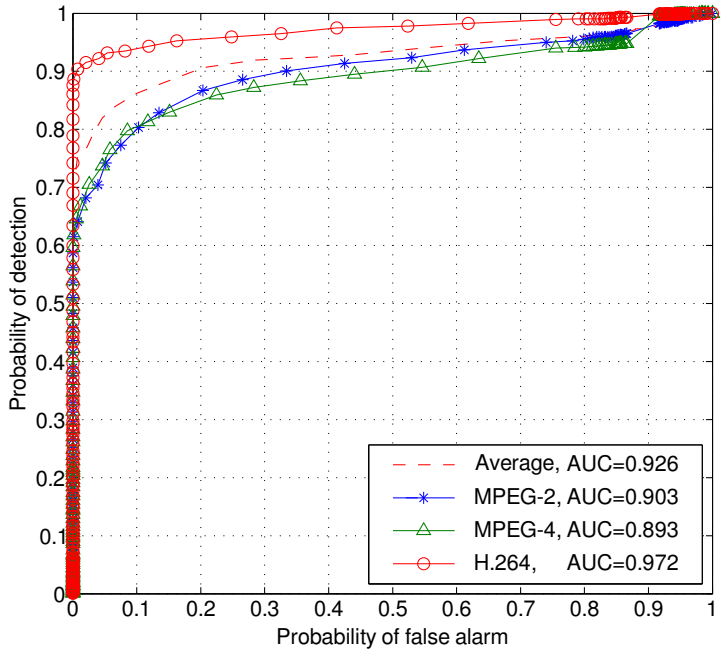


Figure 7.3: ROC curves for the proposed double encoding detector.

is very low (e.g., 100 kbps) the footprint could be hidden by spurious effects. This explains the behavior shown in Figure 7.3: since H.264 is known to provide better quality with respect to MPEG-x codecs for a fixed bitrate, it facilitates the detection of the VPF and, consequently, the correct classification of the video. Therefore, the proposed method retains considerable accuracy also when MPEG-x codecs are used, and yields on average a detection rate of 80% when the false positive rate is fixed at 5%.

7.4.2. First GOP Size Estimation

For studying the performance of the proposed GOP size estimation technique, we create a dataset with a total of 32256 doubly encoded videos, by compressing each of the 14 available sequences with all the possible combinations of settings given in Table 7.2. Each sequence is analyzed in about 1.4 seconds on a desktop computer[3], but the actual analysis, that starts when the macroblock types have been extracted, takes only 0.025 seconds on average.

We investigate the results of the estimation method from different points of view: as a function of 1st and 2nd bitrate, as a function of the 1st and 2nd encoder, and as a function of the 1st and 2nd GOP size. Each time we investigate a parameter, all the other settings are marginalized out, i.e., results are averaged over them. We assume a correct estimation (or exact match) whether the value $\hat{G}_1$

[3]Intel Core2Duo @3.4GHz, 8GB RAM, running Ubuntu 10.04.

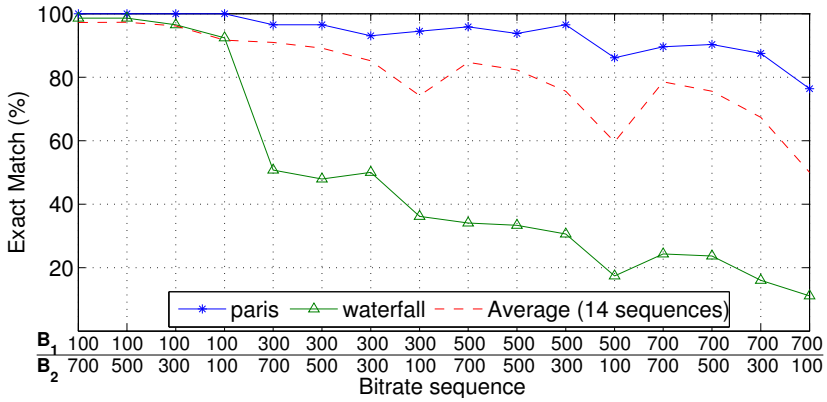


Figure 7.4: Performance of the method as a function of the $B_1 - B_2$ bitrate.

obtained from (7.4) actually matches G_1. Otherwise, we believe that having just an approximation of G_1 is not meaningful from a forensic point of view. Finally, given that we are using 14 different source sequences, for each experiment we report: i) average performance; ii) results from the video sequence yielding the best estimation percentage (*paris* in all the experiments); iii) results from the video sequence leading to the worst estimation percentage (*waterfall* in all the experiments).

In Figure 7.4 we report the percentage of exact match as a function of $B_1 - B_2$ combination of bitrates. We see that lower bitrates during the first encoding result in better performance, in agreement to what is said in Section 7.2: low bitrates require strong quantization which acts like a low-pass filter, thus increasing the number of blocks that will be more conveniently encoded as I-MB. This will be especially true for videos where uniform regions are available, like the *paris* sequence (which yields the best results), while textured content is against this phenomenon, as confirmed by the *waterfall* sequence (which is rich of textures) being the worst. From the second compression point of view, it is confirmed that low bitrates negatively affect the performance, since they reduce the possible choices for the encoder when assigning macroblock types; nevertheless, even in the worst conditions, the proposed footprint is able to correctly estimate G_1 half of the times.

Figure 7.5 shows the percentage of exact match for different combinations of codecs. We see that reliability increases when the second encoding is carried out with H.264, which is consistent with the observations made in Section 7.4.1 about the presence of VPF in doubly encoded videos.

Finally, we evaluate the performance for different combinations of G_1 and G_2 in Figure 7.6. Results show an intuitive fact: as G_1 increases, the accuracy of the method drops. The simplest justification stems from the fact that we are using a fixed number of frames for the estimation. Hence, the higher G_1, the less number of periods we are able to observe, and, as expected, this results in

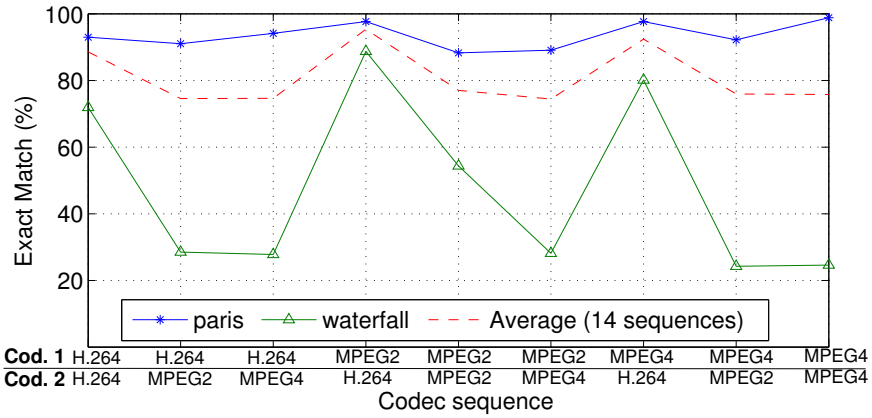


Figure 7.5: Performance of the method as a function of the codec combination.

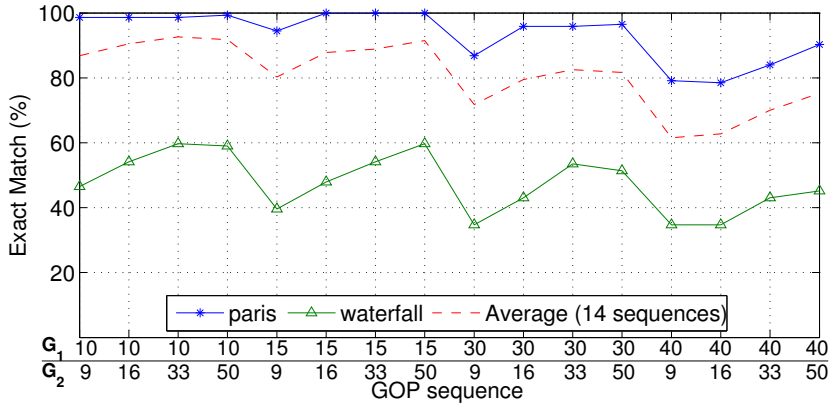


Figure 7.6: Performance of the method as a function of the $G_1 - G_2$ combination.

noisier estimates. Another appealing fact is that results improve as G_2 increases, mainly because this reduces the number of spurious effects induced by the GOP structure of the second compression. Interestingly, GOP size estimation is more reliable whenever the GOP used for the second compression is larger than the one employed during the first encoding.

As a last consideration, it is worth noting that the video *waterfall* proves to be the most challenging in all the experiments. This video sequence is very rich in textured frames, where the VPF will show up with more difficulty, according to the arguments given in Section 7.2.

7.5. Conclusions

Video forensics is an emerging field, targeting the investigation of the processing history of a digital video. To this extent, detecting whether a video has been compressed once or twice is an interesting task, especially if an estimation of

some of the first encoding parameters can be provided. In this chapter, we have introduced both a new kind of footprint based on the variation of the macroblock prediction types in the re-encoded P-frames (VPF), and a method to exploit this footprint to detect video double encoding and estimate the size of the GOP used in the first compression. Besides the inherent importance of discovering information about the processing undergone by a digital content, we believe that GOP size estimation can also be seen as a basic tool for more advanced analyses, that may target tampering detection as it will be noted in the next chapter.

Experiments show that, being based on a simple principle, the VPF is a very robust footprint. Both the detection of double encoding and GOP size estimation remain possible (although with some impact on performance) even when the second compression is stronger than the first one, while this configuration is prohibitive for most of the existing forensic methods.

Chapter 8

Localization of Forgeries in MPEG-2 Video through GOP Size and DQ Analysis

This chapter deals with intra-frame forgery localization in compressed videos, one of the less studied problems in video forensics up to now. Due to the high complexity associated to the formal treatment of video compression, most of the available techniques work under strong assumptions which limit their application in realistic scenarios. However, in a similar way as noted in the last section of Chapter 2, a practical solution can be obtained through the combination of existing tools, such that more complex problems can be addressed. The proposed method in this chapter is based on the analysis of Double Quantization (DQ) traces in frames that are encoded twice as intra in MPEG-2 video sequences. Using the derived approach in the previous chapter, doubly encoded I-frames are located in the video under analysis by estimating the GOP size used in the first compression. Then, the DQ analysis is devised for the MPEG-2 encoding scheme and applied to these located I-frames. By doing this, regions that were manipulated between the two encodings are detected. Compared to existing methods based on double quantization analysis, the proposed scheme makes forgery localization possible on a wider range of settings.

8.1. Introduction

In the previous review of the literature carried out in Section 1.3.3, we have argued that a relevant part of the research activity in video forensics is focused on detection of double encoding. Although double compression is almost always necessary for creating a tampered video, by simply assuring that a video sequence

has been encoded twice is not a sufficient proof for claiming its non-authenticity. For instance, it may be the case that the video is automatically re-encoded when it is downloaded from the acquisition device. For this reason, investigating the authenticity of a digital video entails taking a further step in the analysis. Furthermore, a distinction must be made between intra-frame and inter-frame video forgeries [3]: in the former, the attacker changes the content of some frames (e.g., by adding or removing an object), while in the latter at least one or more frames are entirely added/removed from the video sequence. Given the different nature of each problem, distinct techniques are needed to investigate each of these two tampering scenarios.

An effective method for detecting inter-frame forgeries, such as deletion of frames, was proposed in [41]. In this work, the de-synchronization (induced by the forgery) between the GOP used in the first and the second encoding is exposed through the detection of a periodic behavior in the magnitude of the prediction error of P-frames. Another strategy is presented in [42], which exploits the fact that the MPEG-2 video coding standard defines different quantization matrices for intra- and inter-coded frames. Looking for anomalies in the energy conveyed by high-frequency DCT coefficients, authors are able to find out GOP structure inconsistencies, thus revealing the forgery.

Regarding intra-frame forgery, Wang and Farid's seminal work in [43] was the first to separately apply a DQ analysis to each macroblock of a video under analysis, as a means to localize forged regions. The main idea lies in the fact that when some of the macroblocks in a frame show the double quantization effect and some others do not, the last ones have been probably pasted from another sequence. This idea is borrowed from JPEG image forensics and, as such, the analysis makes sense only on frames that have been encoded twice as intra. The authors work around this problem by assuming that only intra-coded pictures are used, thus heavily restricting the applicability of the method. Furthermore, the MB-by-MB analysis in a whole video sequence severely increases the computational burden.

In line with the last work, this chapter presents a method for intra-frame forgery localization in MPEG-2 compressed videos, allowing one to determine which parts of a frame under analysis have been altered. The method basically works by searching for traces of double quantization at a spatial level, enabling the construction of a fine-grained probability map of tampering for each analyzed frame. This is done by adapting and extending the method proposed in [82], which originally works on JPEG images, to the MPEG-2 encoding scheme. Due to how the video encoding is performed, this kind of analysis is only possible on frames that have been intra-coded twice. Therefore, we first adopt the VPF-based approach [83], which has been described in the previous chapter, for localizing the position of the I-frames in the first encoding, and then perform the proposed analysis on the suitable (double-encoded) I-frames.

Compared to state-of-the-art techniques targeting the same task, the proposed method achieves forgery localization under more realistic working scenarios (e.g., video encoded using motion prediction which are the vast majority), and exploits some advantages of the MPEG-2 encoding standard to improve the robustness of the analysis.

The chapter is structured as follows: Section 8.2 covers the basics on MPEG-2 video compression; then the proposed method is explained in Section 8.3 and experimentally validated in Section 8.4; finally, Section 8.5 reports conclusions.

8.2. MPEG-2 Video Compression

MPEG-2 video standard (ISO/IEC 13818-2/ITU-T recommendation H.262) [24] is a widely employed method for video compression, that basically works by reducing both spatial and temporal redundancy in a captured video sequence. The standard follows a block-based hybrid video coding approach (similar to the one depicted in Figure 1.6) and defines different types of pictures: intra-coded pictures, referred to as I-frames (only progressive videos are considered here), and predictive-coded pictures, commonly named P-frames and B-frames. Given the block-based structure, each frame of a video sequence is divided into macroblocks (MBs), i.e., blocks of 16×16 samples, which are encoded following several coding modes that are available according to the selected type of frame.

In a similar way as it happens with JPEG images, the MBs in I-frames are encoded without making reference to other frames: each MB of the luminance component (we obviate the chrominance for brevity) is divided into blocks of 8×8 pixels that are transformed according to the DCT and whose coefficients are later quantized (details about this step will be given in Section 8.3). Quantization in the DCT domain allows to remove spatial redundancy in a perceptually convenient way.

By compressing the whole video using only intra-coded pictures, this would lead to the so-called M-JPEG encoding, where temporal redundancy is not exploited. Notice that, although being very similar to the JPEG compression scheme, the mentioned procedure uses a slightly different quantization function. In MPEG-2, the coarseness of the quantization is selected by the encoder through the quantizer scale factor, denoted as Q, that ranges from 0 to 31 and maps the values of the multiplier k that is applied to the quantization matrix. Two different mappings are available in the standard, but in this chapter we will assume the one that corresponds to $k = 2\mathrm{Q}$ (except for $\mathrm{Q} = 0$, where no value is assigned to k). Therefore, by fixing the multiplier to a certain value, the factor Q enables to control the trade-off between the quality and bitrate of a compressed video. If the value of Q is constant, then a fixed quantizer will be used and a Variable

BitRate (VBR) will be provided, while if it is adapted on a frame to frame (or even on a MB to MB) basis, then a Constant BitRate (CBR) can be achieved.

In a general scenario, a strong correlation between adjacent frames will be present since the scene is captured at several frames per second, and this temporal redundancy should be exploited to increase the level of compression. This is obtained through motion compensation. For instance, when encoding a picture as a P-frame, each MB is compared with the respective area in neighborhood positions within the previous encoded and reconstructed frame (i.e., a reference frame), in order to find the region that better resembles the MB to encode. If a good match is found, then the MB is predictive-coded: the displacement vector (i.e., a motion vector) is stored and the residual difference with the reference MB is 8×8-DCT transformed and further quantized. However, if a good match is not available, then the MB is intra-coded like in an I-frame and we will refer to this type of macroblocks as I-MB (cf. Section 7.2). Finally, if after performing the predictive-coding there is no need to transmit the motion vector (because it is null) and the residual difference after quantization is also negligible, then the standard defines a specific type of macroblock, called skipped MB and we will refer to this type of macroblocks as S-MB (cf. Section 7.2).

The only difference between P- and B-frames is that the MBs on B-frames can be bidirectionally predictive-coded, in such a way that the motion compensation can be carried out from a past and/or a future reference frame. However the latter type of frames will not be addressed in the following.

8.3. Proposed Method

In this section, we present a new method for localizing forgeries in MPEG-2 videos. We focus on the intra-frame forgery scenario, and we assume that, starting from an MPEG-2 video sequence, the attacker decodes the video, alters the content of a group of frames, and finally encodes the resulting sequence again with an MPEG-2 encoder, using a different GOP size. In the following, we assume a fixed quantizer and that the default quantization matrix is employed, leading to a VBR coding.

The proposed approach makes use of the method presented in Chapter 7 (i.e., [83]) to retrieve the GOP size of the first compression. By knowing this, the location of the I-frames in the first encoding is inferred and the DQ effect is studied in those frames that have been encoded as intra both in the first and second encoding. In this work, we do not consider the removal/addition of whole frames: this would cause a misalignment in the GOP structures, complicating the localization of frames that have been encoded twice as intra.

8.3.1. Detection of Frames Encoded Twice as Intra

As just seen in the previous chapter, the Variation of Prediction Footprint (VPF) has been proposed as a method to detect double video encoding. This footprint captures a characteristic phenomenon that occurs when an I-frame is re-encoded as a P-frame: in such a frame, the number of S-MB noticeably decreases, while the number of I-MB strongly increases. By measuring the presence and the periodicity of this anomalous variation, an algorithm has been proposed to detect double compression and to estimate the size of the GOP used for the first encoding.

Let us assume that a video, composed by N frames, has been encoded twice using G_1 and G_2 as the GOP size for the first and second encoding respectively, where $\mathrm{G}_1 \neq m \cdot \mathrm{G}_2, \forall m \in \mathbb{N}$. Assuming a fixed GOP structure, the set of indices of the frames that have been intra-coded twice is

$$\mathcal{C}_{\mathrm{G}_1,\mathrm{G}_2} = \{n \in \mathbb{N} : n = m \cdot \mathrm{lcm}(\mathrm{G}_1, \mathrm{G}_2) \wedge n \leq N, \forall m \in \mathbb{N}\},$$

where $\mathrm{lcm}(\mathrm{G}_1, \mathrm{G}_2)$ represents the least common multiple between G_1 and G_2. The cardinality of the set $\mathcal{C}_{\mathrm{G}_1,\mathrm{G}_2}$ is simply given by

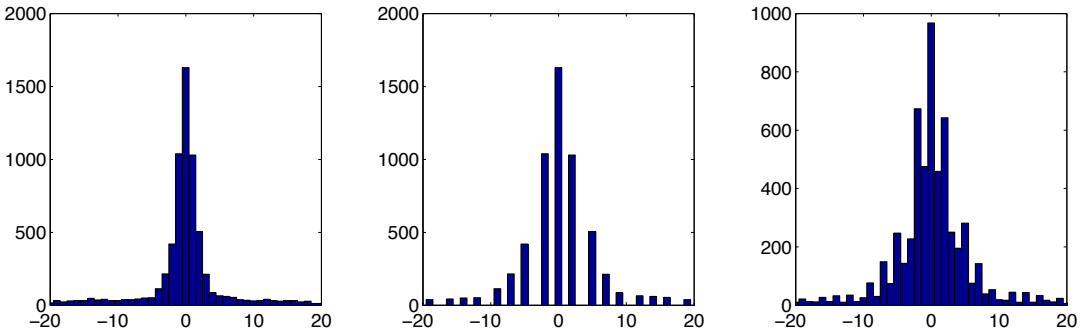
$$|\mathcal{C}_{\mathrm{G}_1,\mathrm{G}_2}| = 1 + \left\lfloor \frac{N}{\mathrm{lcm}(\mathrm{G}_1, \mathrm{G}_2)} \right\rfloor,$$

where $\lfloor \cdot \rfloor$ stands for the floor function. In other words, forgery localization can be performed every $\mathrm{lcm}(\mathrm{G}_1, \mathrm{G}_2)$ frames. Therefore, for relatively prime values of G_1 and G_2 the analysis can be carried out only once every $\mathrm{G}_1 \times \mathrm{G}_2$ frames, and this value might be not appealing in practice. On the other hand, the GOP size is usually chosen from a set of possibilities, like 12 for PAL videos, 15 for NTSC videos, while recording devices often choose a GOP size around 30. At a frame rate of 25 fps, combinations of the mentioned values for G_1 and G_2 result in a satisfactory time resolution for the analysis.

Note that the adoption of the VPF-based approach will have to be evaluated again in this chapter, given that in Chapter 7 experiments were conducted on double encoded videos without modifications between the first and the second encoding. In contrast, now we are assuming that the video is manipulated (by altering the content of a group of frames) before the second compression takes place. Therefore, the robustness of the VPF in this scenario must be investigated, and this task will be addressed in Section 8.4.

8.3.2. Forgery Localization Based on DQ Analysis

According to the assumed forgery scenario, tampered frames that have been encoded twice as intra will consist of two groups of pixels: one that has not been



(a) Single compression (b) Double compression (c) Single + Double

Figure 8.1: Histogram of a DCT coefficient from a single compressed frame (a), double compressed frame (b) and tampered frame (c). Notice that the histogram from the tampered frame can be seen as a mixture of the previous two histograms.

modified, thus undergoing a double quantization, and another that has been introduced between the encodings. Even when these latter pixels come from a compressed sequence, they will unlikely be pasted respecting the 8×8 quantization grid of the host frame and, therefore, will not show traces of double quantization after the second encoding, thus making localization possible. A thorough explanation of this model is given in [84]. Looking at the histogram of a specific DCT coefficient (e.g., the one at position $(0, 1)$ in all 8×8 blocks) from a tampered frame, we should see a mixture of two components: a "standard" component due to the new added regions (see Figure 8.1(a)), and a comb-shaped component due to double compressed regions that remain unaltered (see Figure 8.1(b)). The final result is plotted in Figure 8.1(c) where a mixture of both components can be appreciated.

In [82], a Bayesian inference method is proposed to compute the probability that each 8×8 block in a frame has been tampered with. This approach first assigns to each DCT coefficient from an 8×8 block its probability of belonging to any of the two above illustrated components (i.e., "standard" or "comb-shaped"). Then it accumulates these probabilities for all the coefficients of the block, yielding an aggregated probability for the whole block of being/not being doubly compressed. The resulting output is a map associating to each 8×8 block of pixels its probability of being tampered (i.e., not showing the DQ effect) or untouched (i.e., showing the effect). In order to compute such a map, the mentioned algorithm basically performs the following steps (see [82] for a formal presentation) for each group of DCT coefficients sharing the same position:

1. From the observed DCT coefficients, estimate the histogram $h(x)$ that would result after a single encoding with the quantization step used in the second compression.

2. Estimate the quantization step that was used during the first compression.

3. Knowing both quantization steps, compute a function $n(x)$ that gives the number of bins of the original histogram that are mapped in the bin corresponding to the value x in the double quantized histogram.

Then, denoting with $\mathcal{H}_0$ and $\mathcal{H}_1$ the hypothesis of being tampered and original, respectively, for each coefficient x it is obtained that

$$p(x|\mathcal{H}_0) = h(x) \tag{8.1}$$

and

$$p(x|\mathcal{H}_1) = n(x)h(x), \quad \text{with } x \neq 0. \tag{8.2}$$

These steps are carried out separately for each DCT coefficient (usually only the first dozen of AC coefficients are used for the analysis). Then, for each 8×8 block, the probability of being tampered is "accumulated" as:

$$p = \frac{1}{\prod_{i|x_i \neq 0} n_i(x_i) + 1}, \tag{8.3}$$

where $n_i(x)$ is the $n(x)$ function for the i-th coefficient.

Since DCT coefficients quantization is a key step both in JPEG and MPEG-2 coding, the above method can be borrowed from image to video forensics, as suggested in [43]. However, some significant differences must be considered to devise a correct model for MPEG-2:

1. The dequantization formula in JPEG differs from that of MPEG-2 [24].

2. In JPEG, the 8×8 quantization matrix is declared in the header and it is usually not governed by the quality factor; in MPEG-2, instead, the adopted matrix (the default or a custom one) is parameterized by the multiplier k (cf. Section 8.2), to adjust the quantization strength.

3. In JPEG, the quantization matrix is the same along the whole image. This also holds for MPEG-2 when a fixed quantizer is used, while the quantization matrix may change from frame to frame or MB to MB, for instance, for CBR coding.

Each of these facts has a direct implication on the model described in [82]. In light of the fact that a different quantization formula is used in MPEG-2, the function $n(x)$ will likely change. On the other hand, since all the quantization coefficients are determined by the multiplier k, it is not necessary to estimate a different quantization step for each coefficient (we can directly estimate k). Finally, in the case of CBR coding, that is left for future work, MBs that are not quantized using the same k must be analyzed separately.

Inspired by [82], the approach we follow is to model the histogram of DCT coefficients in tampered frames as a mixture between a double quantized component and a single quantized component. To get a reliable estimate of single quantized coefficients (by the quantization factor employed in the last encoding), we make use of the calibration technique [85]: the frame is cropped by one row and one column, and the result is quantized with the second quantization matrix. Consistently with [82], this component will be indicated by $h(x)$. Therefore, given a coefficient x, its probability of belonging to a tampered region is estimated as in (8.1).

To get an estimate of the double quantized component, we need to derive the appropriate function $n(x)$ for the MPEG-2 quantization scheme. In the following, we denote the never-compressed DCT coefficient on the i-th row and j-th column of an 8×8 block with $x(i,j)$, where $i,j \in \{0,\dots,7\}$. Similarly, the quantized version of the coefficient is denoted by $u_1(i,j)$, its dequantized version by $x_1(i,j)$, and the re-quantized version by $u_2(i,j)$. On the other hand, each element in the 8×8 quantization matrix is represented by $W(i,j)$, and the multipliers that parameterize the quantization matrix in the first and second compression, are labeled as k_1 and k_2, respectively. According to the MPEG-2 standard [24], and following the proposed notation, the dequantized version of the DCT coefficients coming from a single compressed intra-coded frame corresponds to

$$x_1(i,j) = \operatorname{sign}\left(u_1(i,j)\right) \left\lfloor \frac{W(i,j)\,|u_1(i,j)|\,k_1}{16} \right\rfloor, \tag{8.4}$$

for all coefficients apart from the DC, and where $|\cdot|$ is the absolute value operator. Taking (8.4) as reference, the most intuitive way to define the quantization is

$$u_1(i,j) = \left\lfloor \frac{16\ x(i,j)}{k_1 W(i,j)} \right\rceil, \tag{8.5}$$

where $\lfloor \cdot \rceil$ represents the rounding to nearest integer operation. Using (8.4) and (8.5), the re-quantized version of the DCT coefficients (i.e., the double quantized coefficients), can be written as

$$u_2(i,j) = \left\lfloor \frac{16}{k_2 W(i,j)} \left(\operatorname{sign}\left(\left\lfloor \frac{16\ x(i,j)}{k_1 W(i,j)} \right\rceil \right) \left\lfloor \frac{W(i,j) \left| \left\lfloor \frac{16\ x(i,j)}{k_1 W(i,j)} \right\rceil \right| k_1}{16} \right\rfloor \right) \right\rceil .$$

From this formula, the function $n(x(i,j))$ for each DCT coefficient $x(i,j)$ can be obtained. For the sake of notational simplicity, we omit the position indices, yielding

$$n(x) = \frac{k_1 W}{16} \left(\left\lceil \frac{16}{k_1 W} \left\lceil \frac{k_2 W}{16} \left(u_2 + \frac{1}{2} \right) \right\rceil \right\rceil - \left\lceil \frac{16}{k_1 W} \left\lceil \frac{k_2 W}{16} \left(u_2 - \frac{1}{2} \right) \right\rceil \right\rceil \right), \tag{8.6}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. In the above equation, k_1 is the only parameter that must be estimated, given that k_2 and the values of W are available

from the bitstream. The multiplier k_1 is defined by its relation with the quantizer scale factor used in the first encoding Q_1 as explained in Section 8.2, thus being a possible value within the set $\mathcal{K}_1 = \{2\mathrm{Q}_1 : 1 \leq \mathrm{Q}_1 \leq 31\}$. If we assume to have the actually used k_1, i.e., $\tilde{k}_1$, the histogram of doubly quantized coefficients can be obtained from $h(x)$ as $n(x;\tilde{k}_1)h(x)$.[1] Therefore, in general, we can write the probability distribution of the observed coefficients as the following mixture

$$p(x;k_1,\alpha) = (1-\alpha)h(x) + \alpha\ n(x;k_1)h(x),$$

where $\alpha \in [0,1]$. As suggested in [82], an effective way to get an estimate of k_1 is to iteratively search the value $\hat{k}_1$ that minimizes the difference between the observed histogram $\tilde{h}(x)$ and $p(x;\tilde{k}_1,\alpha)$, choosing the optimal α in the least square sense (formula is given in [82]).

Contrary to the JPEG case, the minimization process here is simplified, given that the quantization matrix is specified by the MPEG-2 standard [24], and all the coefficients share the same k_1. Thus, we define the following vector from the observed histograms

$$\tilde{\mathbf{h}} \triangleq \left(\tilde{h}_1(-\tfrac{B}{2}),\ldots,\tilde{h}_1(-1),\tilde{h}_1(1),\ldots,\tilde{h}_1(\tfrac{B}{2}),\tilde{h}_2(-\tfrac{B}{2}),\ldots,\tilde{h}_C(\tfrac{B}{2})\right)^T,$$

where $B+1$ is the number of bins of $\tilde{h}(x)$ and C is the number of considered coefficients. We similarly build up $\mathbf{h}$ from the histogram obtained using the calibration technique and $\mathbf{n}$ by gathering values according to (8.6). Then, we can write:

$$\mathbf{p}(k_1,\alpha) = (1-\alpha)\mathbf{h} + \alpha\ \mathbf{n}(k_1) \odot \mathbf{h},$$

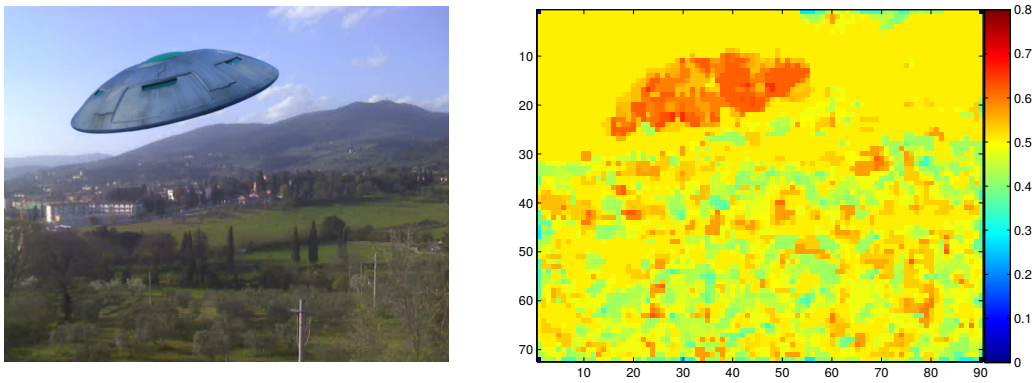
where $\odot$ denotes the element-wise product of vectors. Finally, $\hat{k}_1$ is obtained as

$$\hat{k}_1 = \arg\min_{k_1 \in \mathcal{K}_1} ||\tilde{\mathbf{h}} - \mathbf{p}(k_1,\alpha)||^2.$$

By using all the coefficients to estimate k_1, a more robust estimation is obtained. This is a crucial benefit, especially if we consider that: i) the quantization steps in $W(i,j)$ are quite large even for small i and j, ii) the spatial resolution of videos is usually smaller than that of images. Both of these facts reduce the number of DCT coefficients that can be fruitfully exploited for the estimation.

At this point, the probability in (8.2) can be computed by means of the resulting $\hat{k}_1$ and the $n(x)$ function defined in (8.6) for the MPEG-2 case. Finally, the probability for each 8×8 block of being tampered is obtained through equation (8.3). Figure 8.2 shows a forged frame along with the probability map generated by the proposed method.

[1] For the sake of a better readability, we define $n(x;\tilde{k}_1) \triangleq n(x)|_{k_1=\tilde{k}_1}$.



(a) Tampered frame (b) Probability map

Figure 8.2: An intra-frame tampering (left figure) and the computed probability map (right figure). For showing purposes, a 3×3 median filter has been applied to the p-map.

8.4. Experimental Results

Experiments have been conducted on a set of well-known videos, containing several heterogeneous scenes,[2] which have been cropped to a 576p resolution, i.e., 720×576 pixels. All the tests are performed using the built-in MPEG-2 encoder and decoder from the `FFmpeg` coding software. The encoder is configured on VBR mode, i.e., using a fixed quantizer for each frame.

The experimental validation follows this workflow: each video is compressed with a quantizer scale factor Q_1; then it is decoded and a square block of 200×200 pixels is replaced with the same content coming from its genuine uncompressed version; finally, the resulting video is re-encoded with a factor Q_2. Using the uncompressed version of the same video as a source for tampered pixels, it is possible to create a forgery that is practically imperceptible to the eye, thus mimicking the work of an editing expert.

Given that the VPF does not strongly depend on the size of GOPs (cf. Section 7.4), we set two GOP sizes: $G_1 = 12$ and $G_2 = 15$ for the first and second compression, respectively. Note that the selection of these particular values for G_1 and G_2 has been motivated in Section 8.3.1. Furthermore, we limit ourselves to use P-frames, provided that GOP size estimation in presence of B-frames is not possible with the VPF-based approach. As shown in [82], forgery localization generally works better when the second compression is not as strong as the first one. For this reason, we choose $Q_1 \in \{6, 8, 10, 12\}$ and $Q_2 \in \{2, 3, 4, 5\}$. Then, all the possible combinations between these two sets are used for generating tam-

[2] Freely available at: `http://media.xiph.org/video/derf`
Selected videos are: *ducks_take_off*, *in_to_tree*, *old_town_cross*, *park_joy*, *shields*, *sunflower*, and *touchdown_pass*.

Table 8.1: AUC obtained with the proposed method.

Q_1 \ Q_2	**2**	**3**	**4**	**5**
6	0.98	0.97	0.68	0.63
8	0.98	0.98	0.96	0.93
10	0.98	0.98	0.91	0.94
12	0.98	0.98	0.97	0.94

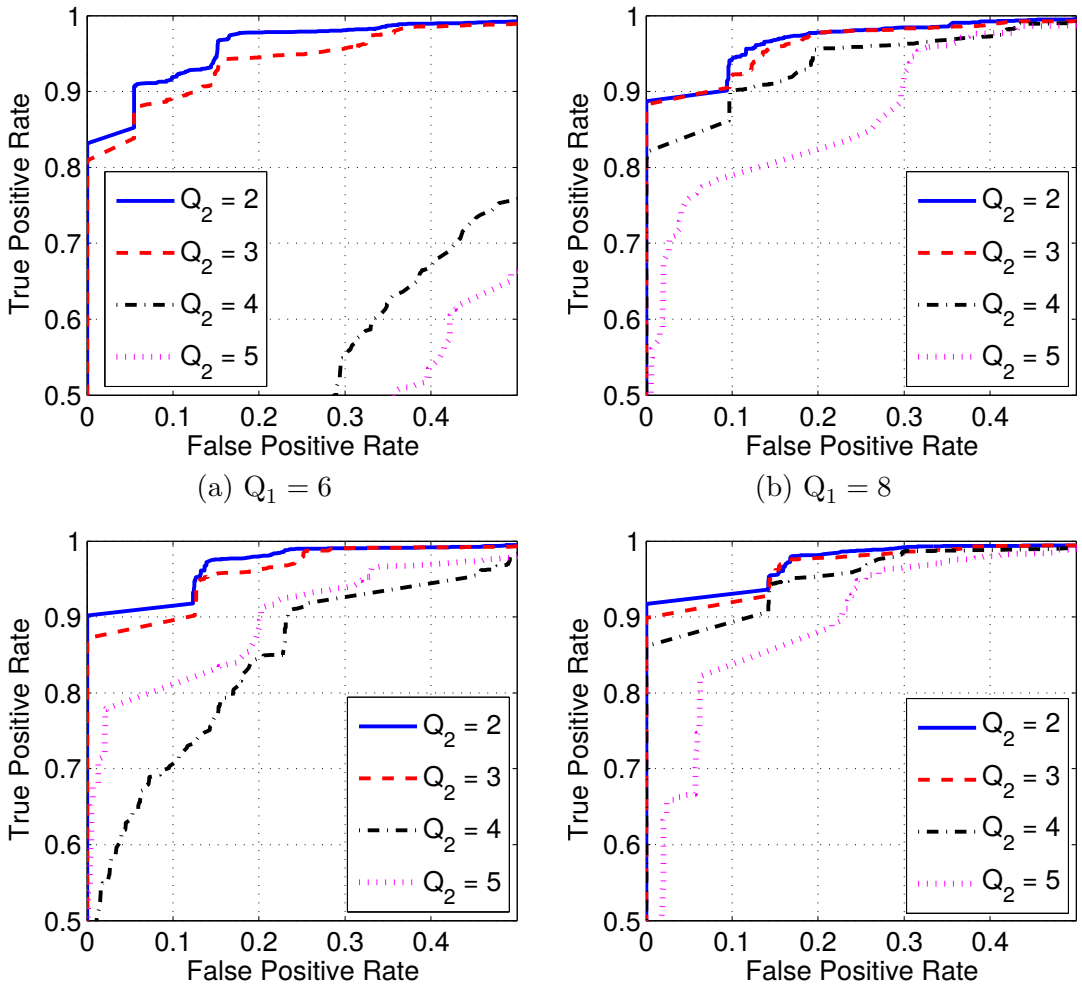
pered videos. Finally, since the model proposed in Section 8.3.2 has been derived assuming that the fixed quantizer is uniform, the dead-zone of the quantizer implemented in `FFmpeg` is fixed to the interval $[-\Delta/2, \Delta/2]$ (where Δ denotes the quantization step), by setting the parameter `ibias` equal to 128. Notice that the model can be easily adjusted to work with different dead-zones.

As a first step in the evaluation, we have investigated the reliability of the VPF-based GOP size estimator in the described scenario, because in the case of a wrong estimation, the proposed practical solution would fail. The GOP size was estimated from the available set of tampered videos and the number of exact estimations of G_1 was calculated as in Section 7.4. The estimation never failed under the considered settings, thus confirming that VPF can be safely used in the proposed chain of analysis.

After retrieving the GOP size of the first compression, i.e., $\hat{G}_1$, for each tampered video, the DQ analysis is applied to the specific frames indexed by the elements in the set $\mathcal{C}_{\hat{G}_1,15}$, defined in Section 8.3.1. Only the first 5 AC coefficients in the zig-zag ordering have been used for the analysis. The probability map produced from each frame is then thresholded and compared to the ground truth mask, allowing us to calculate the true positive and false positive rate. These values are averaged over all videos sharing the same combination of Q_1 and Q_2.

By covering different values of the threshold for all the explored combinations of Q_1 and Q_2, Receiver Operating Characteristic (ROC) curves are obtained and depicted in Figure 8.3. The resulting AUC is calculated in each case, collecting the achieved results in Table 8.1. From these outcomes, we clearly see that, for a given factor Q_1, smaller values of Q_2 facilitate the forgery localization. At the same time, larger values of Q_1 yields better performance.

Interestingly, it can be observed that better results are achieved in terms of AUC when Q_1 is an integer multiple of Q_2. For example, when fixing $Q_1 = 10$, a larger value of AUC is achieved with $Q_2 = 5$ (i.e., AUC = 0.94) with respect to $Q_2 = 4$ (i.e., AUC = 0.91), even if in the former case a slightly stronger compression has been applied. The reason of this fact still remains unclear, but will be further investigated.



(a) $Q_1 = 6$ (b) $Q_1 = 8$

(c) $Q_1 = 10$ (d) $Q_1 = 12$

Figure 8.3: ROC curves obtained for the examined combination of Q_1 and Q_2. From left to right and top to bottom, increasing values of Q_1 are considered, and performance for varying values of Q_2 are plotted (notice that curves have been magnified to improve readability). As expected, lower values for the second compression facilitate the localization.

8.5. Conclusions

In this chapter, a method for localizing intra-frame forgeries in MPEG-2 compressed video sequences has been proposed. The method works by first locating frames that have been intra-coded twice, and then applying a double quantization analysis to them. The double quantization analysis has been specifically designed to fit the standardized MPEG-2 dequantization process. As a key contribution, our method exploits the characteristics of MPEG-2 coding, and it is the first allowing to apply DQ analysis to videos that have been encoded using P-frames.

Chapter 9

Conclusions and Future Work

This thesis has presented an array of new techniques in the field of multimedia forensics, focusing on the estimation of the processing a multimedia content has gone through. Two lines of research have been covered: firstly, the forensic analysis of resampled signals has been cast in a theoretical framework and secondly, more complex signals such as video compressed sequences have been characterized from a forensic point of view. In both cases, practical forensic tools have been proposed for detecting particular forgeries applied to multimedia contents.

The modeling of the resampling problem has been addressed from different perspectives along this thesis. We started working with cyclostationarity theory, adapting first the underlying concepts to build a comprehensible framework, and then tackling the estimation of the resampling factor as a means to identify the spatial transformation undergone by a certain region of a digital image. From this analytical description, we have realized that the use of a prefilter, prior to perform the search for cyclostationarity, improves the estimation accuracy. The design of prefilters has therefore been further investigated within this framework, providing a measure that enables the derivation of better prefilters for resampling factor estimation.

After remarking the influence on the previous model of the rounding operation, used as a last step in the resampling process, we have moved to a statistical characterization of the resampling problem relying on the maximum likelihood criterion. In this context, a new approach has been derived by establishing connections between the linear dependencies introduced by the resampling process and the structure imposed by the quantization of the resampled values due to rounding. Interestingly, this method only needs a few number of samples to correctly estimate the applied resampling factor to a given signal. This statistical analysis has served as a bridge for linking the resampling problem to set-membership estimation theory. This theory has proven to be a useful resource for addressing the problem of resampling factor estimation, provided that the

obtained solutions have the singular characteristic of being consistent with all information arising from the observed data. Hence, the presented technique becomes a valuable asset for a forensic analyst who needs to provide unquestionable proofs of tampering.

Deepening the understanding of the linear dependencies induced by the resampling process among neighboring samples, a simple strategy for resampling detection has been proposed. In particular, we have shown that interpolated images belong to a subspace defined by the interpolation kernel. As a result, the proposed detector only needs to compute the SVD of a given image block for discerning upsampled images from genuine ones (nonetheless, a measure of the degree of saturated pixels is also needed to avoid misdetections). In addition, the detector can cope with very small regions, which is appealing for exposing tampered regions (which might be small) through the detection of resampling inconsistencies.

A qualitative assessment of all the proposed resampling-based methods is summarized in Table 9.1, where the advantages and disadvantages of each technique are highlighted. Note that the first three rows in the table collect information about the resampling estimators covered in Chapters 2, 4, and 5, while the last one corresponds to the resampling detector described in Chapter 6. As it can be observed, depending on the feature exploited, each technique offers distinctive characteristics, but also associated drawbacks. The comparison is established in terms of computational complexity, domain of application (i.e., 1-D or 2-D), interpolation kernel support, number of needed samples, and other peculiarities of each technique, such as: alignment with the resampling grid, presence of periodic patterns, saturation, or flat regions, etc.

Apart from these approaches, a practical forensic tool has been designed to distinguish original from duplicated regions in a copy-move forgery with content adaptation. This practical solution combines a SIFT-based method and a resampling-based approach with the final aim of detecting duplicated regions and then revealing which of the detected regions have been adapted (i.e., the duplicates) and which of them have not (i.e., the originals).

Concerning the forensic analysis of video sequences, we have mainly focused our research on three aspects: the study of double compression detection regardless of the codec used, the estimation of the GOP size adopted in the first compression from a double encoded video, and finally, the localization of intra-frame forgeries for the particular MPEG-2 video coding standard.

As a key contribution, we have revealed the presence of a new footprint emerging from the double encoding of video sequences. This new footprint has proven to be robust after transcoding and also under a CBR coding. Significantly, both the detection of double encoding and the GOP size estimation are feasible even when the second compression is considerably stronger than the first one, while

Table 9.1: Qualitative comparison of the proposed resampling estimators (first three rows) and the resampling detector (last row).

Methods	**Advantages**	**Disadvantages**
Cyclo-based (Ch. 2)	1. 2-D analysis 2. Scaling factor and rotation angle estimation 3. Supports any linear kernel 4. No alignment is needed	1. Large number of samples 2. High computational complexity 3. Affected by periodic patterns
ML-based (Ch. 4)	1. Small number of samples 2. Low computational complexity 3. Not affected by periodic patterns	1. 1-D analysis 2. Supports only a fixed linear kernel 3. Needs alignment with resampling grid
SME-based (Ch. 5)	1. Small number of samples 2. Consistent solutions with all the observed data 3. Not affected by periodic patterns 4. Supports any linear kernel	1. 1-D analysis 2. High computational complexity 3. Needs alignment with resampling grid
SVD-based (Ch. 6)	1. 2-D analysis 2. Low complexity 3. Small number of samples 4. No alignment is needed 5. Supports any linear kernel	1. Affected by flat or saturated regions 2. Impaired by demosaicing 3. Does not support downsampling detection

this configuration is prohibitive for most of the existing forensic methods.

Deriving a method for the localization of intra-frame forgeries, a first step has been taken in one of the most challenging problems on video forensics. The proposed approach works by first locating frames that have been intra coded twice (through the use of the earlier mentioned footprint) and then applying a double quantization analysis to expose manipulated regions. Although the proposed approach can only handle MPEG-2 videos compressed under the simple profile, this scenario for the localization of intra-frame forgeries has rarely been considered before.

In conclusion, distinct theoretical aspects have been investigated in this thesis encompassing the forensic analysis of multimedia contents. As a consequence, practical solutions have arisen facing realistic scenarios and providing promising results that could be used in the future to unveil forgeries in multimedia contents.

9.1. Future Research Lines

The work carried out in this thesis is sometimes constrained to specific settings which could be hindering its applicability in real-world scenarios. Therefore, this leaves room for improvement and several extensions are further proposed.

Forensic Analysis of Resampled Signals

The main topics that should be investigated in the future regarding the resampling problem are summarized below:

1. The analysis of the resampling operation has always been addressed assuming as input an uncompressed signal. However, most times, audio signals and images are only available in a compressed format. Although increasing support is given, for instance, to raw image formats, the vast majority of shared images through the Internet are JPEG compressed, thus being more likely to be manipulated. Consequently, the modeling of the processing chain build upon the combination of the resampling operation and the JPEG compression should be tackled to provide more reliable resampling detectors/estimators.

2. In relation to the previous point, the analysis of the resampling operation has almost always been discussed considering that the resampling factor is larger than one, i.e., resulting in an upsampling operation. Nevertheless, it is very likely that a forger use downsampling as a last step to minimize possible visible distortions after manipulating an image. Hence, modeling accurately the downsampling problem becomes a priority in future research lines. As a first attempt in this direction, we believe that the same idea derived in Chapter 6, could be adapted for detecting downsampling operations, by jointly exploiting the traces left by the demosaicing process and the downsampling in the three color components of a digital image.

3. A constant assumption in our model is the linearity of the resampling process. However, current demosaicing algorithms, for instance, are adaptive and commonly non-linear. Therefore, a more complex analysis should be developed or at least the derived approaches should be tested in a non-linear scenario to evaluate the achieved performance.

4. The analysis of resampling inconsistencies to unveil tampered regions has always been addressed in a block-based fashion. An alternative shape could be more appropriate in certain cases. As discussed at the end of Chapter 2, the resampling factor estimation should also be investigated on non-square areas, provided that the zero-padding technique is not the optimal one in any case.

Forensic Analysis of Video Sequences

Future work on video forensics should be oriented towards modeling more complex scenarios with less restrictive assumptions. Some possible lines for extending our work are detailed below:

1. Current camera devices have enough computation power to capture video sequences using B-frames in real-time. The assumption of a baseline or simple profile as indicated in Chapters 7 and 8, does no longer hold and it leads us to a less realistic scenario. Therefore, the analysis of the VPF in presence of B-frames should be further investigated.

2. GOP size estimation assuming adaptive GOPs entails a difficult task because the GOP size changes according to the captured scene, without following a periodic behavior. However, it would be interesting to analyze whether the proposed detector in Chapter 7 could be able to detect double video encoding under this particular scenario with adaptive GOPs.

3. Intra-frame forgery has been addressed assuming that both encodings are performed using always the same quantizer scale factor. This scenario is too restrictive, since nowadays most camera devices automatically change the quantizer scale factor to fit the bandwidth or storage constraints. Therefore, as a next step in the modeling of the forgery localization, the case where one or both compressions are at a constant bitrate should be investigated.

4. The experimental validation of the intra-frame forgery localization has been performed processing automatically generated video forgeries. Nevertheless, a more challenging procedure should be considered. In particular, the experimental validation should be extended to realistic, hand-made, video forgeries.

Bibliography

[1] Alessandro Piva. An overview on image forensics. *ISRN Signal Processing*, 2013. Article ID 496701, 22 pages, 2013.

[2] Robert C. Maher. Audio forensic examination. *IEEE Signal Processing Magazine*, 26(2):84–94, March 2009.

[3] Simone Milani, Marco Fontani, Paolo Bestagini, Mauro Barni, Alessandro Piva, Marco Tagliasacchi, and Stefano Tubaro. An overview on video forensics. *APSIPA Transactions on Signal and Information Processing*, 1, August 2012.

[4] Photo Tampering troughout History. `http://www.fourandsix.com/photo-tampering-history`.

[5] David Vázquez-Padín, Pedro Comesaña, and Fernando Pérez-González. An SVD approach to forensic image resampling detection. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pages 2112–2116, September 2015.

[6] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computergraphics. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 221–228, August 1988.

[7] Andrew S. Glassner. *Graphics Gems*. Academic Press, Inc., Orlando, FL, USA, 1990.

[8] Alin C. Popescu and Hani Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, February 2005.

[9] Matthias Kirchner. Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue. In *Proceedings of the 10th ACM Workshop on Multimedia and Security (MM&Sec)*, pages 11–20, September 2008.

[10] Andrew C. Gallagher. Detection of linear and cubic interpolation in JPEG compressed images. In *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV)*, pages 65–72, May 2005.

[11] Babak Mahdian and Stanislav Saic. Blind authentication using periodic properties of interpolation. *IEEE Transactions on Information Forensics and Security*, 3(3):529–538, September 2008.

[12] Nahuel Dalgaard, Carlos Mosquera, and Fernando Pérez-González. On the role of differentiation for resampling detection. In *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, pages 1753–1756, September 2010.

[13] Matthias Kirchner. Linear row and column predictors for the analysis of resized images. In *Proceedings of the 12th ACM Workshop on Multimedia and Security (MM&Sec)*, pages 13–18, September 2010.

[14] Stefan Pfennig and Matthias Kirchner. Spectral methods to determine the exact scaling factor of resampled digital images. In *Proceedings of the 5th International Symposium on Communications Control and Signal Processing (ISCCSP)*, pages 1–6, May 2012.

[15] Jesús Valera and Narciso García. Unambiguous interpolation rate estimation in uncompressed resized color digital images. In *Proceedings of the 2nd International Workshop on Biometrics and Forensics (IWBF)*, pages 1–4, April 2013.

[16] Weimin Wei, Shuozhong Wang, Xinpeng Zhang, and Zhenjun Tang. Estimation of image rotation angle using interpolation-related spectral signatures with application to blind detection of image forgery. *IEEE Transactions on Information Forensics and Security*, 5(3):507–517, September 2010.

[17] Chenglong Chen, Jiangqun Ni, and Zhaoyi Shen. Effective estimation of image rotation angle using spectral method. *IEEE Signal Processing Letters*, 21(7):890–894, July 2014.

[18] Seung-Jin Ryu and Heung-Kyu Lee. Estimation of linear transformation by analyzing the periodicity of interpolation. *Pattern Recognition Letters*, 36:89–99, January 2014.

[19] S. Prasad and K.R. Ramakrishnan. On resampling detection and its application to detect image tampering. In *Proceedings of the 7th IEEE International Conference on Multimedia and Expo (ICME)*, pages 1325–1328, July 2006.

[20] Matthias Kirchner and Thomas Gloe. On resampling detection in recompressed images. In *Proceedings of the 1st IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 21–25, December 2009.

[21] Tiziano Bianchi and Alessandro Piva. Reverse engineering of double JPEG compression in the presence of image resizing. In *Proceedings of the 4th IEEE*

International Workshop on Information Forensics and Security (WIFS), pages 127–132, December 2012.

[22] Ran Wang and Xijian Ping. Detection of resampling based on singular value decomposition. In *Proceedings of the 5th International Conference on Image and Graphics (ICIG)*, pages 879–884, September 2009.

[23] Xiaoying Feng, Ingemar J. Cox, and Gwenaël Doërr. Normalized energy density-based forensic detection of resampled images. *IEEE Transactions on Multimedia*, 14(3):536–545, June 2012.

[24] ITU-T and ISO/IEC JTC1. *Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video*. ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2 Video), 1994.

[25] ITU-T and ISO/IEC JTC1. *Coding of Audio-Visual Objects-Part 2: Visual*. ISO/IEC 14496-2 (MPEG-4 Visual), 1999.

[26] ITU-T and ISO/IEC JTC1. *Advanced Video Coding for Generic Audiovisual Services*. ITU-T Recommendation H.264 and ISO/IEC 14496-10 (AVC), 2003.

[27] Ian E. Richardson. *The H.264 Advanced Video Compression Standard*. John Wiley & Sons, 2nd edition edition, 2010.

[28] ITU-T and ISO/IEC JTC1. *High Efficiency Video Coding*. ITU-T Recommendation H.265 and ISO/IEC 23008-2 (HEVC), 2013.

[29] ITU-T and ISO/IEC JTC1. *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video*. ISO/IEC 11172-2 (MPEG-1 Video), 1993.

[30] Yuting Su and Junyu Xu. Detection of double-compression in MPEG-2 videos. In *Proceedings of the 2nd International Workshop on Intelligent Systems and Applications (ISA)*, pages 1–4, May 2010.

[31] Junyu Xu, Yuting Su, and Qingzhong Liu. Detection of double MPEG-2 compression based on distributions of DCT coefficients. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(01):(1354001)1–(1354001)21, February 2013.

[32] Dandan Liao, Rui Yang, Hongmei Liu, Jian Li, and Jiwu Huang. Double H.264/AVC compression detection using quantized nonzero AC coefficients. In *Proceedings of SPIE, Media Watermarking, Security, and Forensics III*, volume 7880, pages 78800Q–78800Q–10, February 2011.

[33] Xinghao Jiang, Wan Wang, Tanfeng Sun, Yun Q. Shi, and Shilin Wang. Detection of double compression in MPEG-4 videos based on Markov statistics. *IEEE Signal Processing Letters*, 20(5):447–450, May 2013.

[34] Dongdong Fu, Yun Q. Shi, and Wei Su. A generalized benford's law for JPEG coefficients and its applications in image forensics. In Edward J. Delp and Ping Wah Wong, editors, *Proceedings of SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 65051L–65051L–11, February 2007.

[35] Wen Chen and Yun Q. Shi. Detection of double MPEG compression based on first digit statistics. In *Digital Watermarking*, volume 5450 of *Lecture Notes in Computer Science*, pages 16–30. Springer Berlin Heidelberg, November 2008.

[36] Tanfeng Sun, Wan Wang, and Xinghao Jiang. Exposing video forgeries by detecting MPEG double compression. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1389–1392, March 2012.

[37] Simone Milani, Paolo Bestagini, Marco Tagliasacchi, and Stefano Tubaro. Multiple compression detection for video sequences. In *Proceedings of the 14th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 112–117, September 2012.

[38] Weiqi Luo, Min Wu, and Jiwu Huang. MPEG recompression detection based on block artifacts. In *Proceedings of SPIE, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 68190X–68190X–12, March 2008.

[39] Paolo Bestagini, Ahmed Allam, Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Video codec identification. In *Proceedings of the 37th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2257–2260, March 2012.

[40] Paolo Bestagini, Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Video codec identification extending the idempotency property. In *Proceedings of the 4th European Workshop on Visual Information Processing (EUVIP)*, pages 220–225, June 2013.

[41] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *Proceedings of the 8th Workshop on Multimedia and Security (MM&Sec)*, pages 37–47, September 2006.

[42] Yuting Su, Weizhi Nie, and Chengqian Zhang. A frame tampering detection algorithm for MPEG videos. In *Proceedings of the 6th IEEE Joint International on Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 2, pages 461–464, August 2011.

[43] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting double quantization. In *Proceedings of the 11th ACM workshop on Multimedia and Security (MM&Sec)*, pages 39–48, September 2009.

[44] Paolo Bestagini, Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Local tampering detection in video sequences. In *Proceedings of the 15th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 488–493, September 2013.

[45] Weihong Wang and Hany Farid. Exposing digital forgeries in video by detecting duplication. In *Proceedings of the 9th Workshop on Multimedia and Security (MM&Sec)*, pages 35–42, September 2007.

[46] Ghulam Qadir, Syamsul Yahaya, and Anthony T.S. Ho. Surrey university library for forensic analysis (SULFA) of video content. In *Proceedings of the IET Conference on Image Processing (IPR)*, July 2012.

[47] Alessandra Gironi, Marco Fontani, Tiziano Bianchi, Alessandro Piva, and Mauro Barni. A video forensic technique for detecting frame deletion and insertion. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6226–6230, May 2014.

[48] Matthew C. Stamm and K.J. Ray Liu. Anti-forensics for frame deletion/addition in MPEG video. In *Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1876–1879, May 2011.

[49] Matthew C. Stamm, W. Sabrina Lin, and K.J. Ray Liu. Temporal forensics and anti-forensics for motion compensated video. *IEEE Transactions on Information Forensics and Security*, 7(4):1315–1329, August 2012.

[50] Paolo Bestagini, S. Battaglia, Simone Milani, Marco Tagliasacchi, and Stefano Tubaro. Detection of temporal interpolation in video sequences. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3033–3037, May 2013.

[51] Marco Visentini-Scarzanella and Pier Luigi Dragotti. Video jitter analysis for automatic bootleg detection. In *Proceedings of the 14th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 101–106, September 2012.

[52] Marco Visentini-Scarzanella and Pier Luigi Dragotti. Modelling radial distortion chains for video recapture detection. In *Proceedings of the 15th IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 412–417, September 2013.

[53] Paolo Bestagini, Marco Visentini-Scarzanella, Marco Tagliasacchi, Pier Luigi Dragotti, and Stefano Tubaro. Video recapture detection based on ghosting artifact analysis. In *Proceedings of the 20th IEEE International Conference on Image Processing (ICIP)*, pages 4457–4461, September 2013.

[54] Thomas Gloe, André Fischer, and Matthias Kirchner. Forensic analysis of video file formats. *Digital Investigation*, 11, Supplement 1:S68 – S76, May 2014. Proceedings of the First Annual DFRWS Europe.

[55] Amod V. Dandawaté and Georgios B. Giannakis. Statistical tests for presence of cyclostationarity. *IEEE Transactions on Signal Processing*, 42(9):2355–2369, September 1994.

[56] H. Hurd, G. Kallianpur, and J. Farshidi. Correlation and spectral theory for periodically correlated fields indexed on $\mathbb{Z}^2$. *Center for Stochastic Processes Tech Report*, 448, 1997.

[57] Vinay P. Sathe and P. P. Vaidyanathan. Effects of multirate systems on the statistical properties of random signals. *IEEE Transactions on Signal Processing*, 41(1):131–146, January 1993.

[58] Georgios B. Giannakis. *Cyclostationary Signal Analysis*. Digital Signal Processing Handbook, 1998.

[59] Husrev T. Sencar and Nasir Memon. *Overview of State-of-the-art in Digital Image Forensics, Part of Indian Statistical Institute Platinum Jubilee Monograph series titled 'Statistical Science and Interdisciplinary Research'*. World Scientific Press, 2008.

[60] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. Geometric tampering estimation by means of a SIFT-based forensic analysis. In *Proceedings of the 35th IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1702–1705, March 2010.

[61] Xunyu Pan and Siwei Lyu. Region duplication detection using image feature matching. *IEEE Transactions on Information Forensics and Security,*, 5(4):857–867, December 2010.

[62] Francesca Uccheddu, Alessia De Rosa, Alessandro Piva, and Mauro Barni. Detection of resampled images: Performance analysis and practical challenges. In *Proceedings of the 18th European Signal Processing Conference (EUSIPCO)*, pages 1675–1679, August 2010.

[63] Hieu Cuong Nguyen and Stefan Katzenbeisser. Performance and robustness analysis for some re-sampling detection techniques in digital images. In *Digital Forensics and Watermarking*, volume 7128 of *Lecture Notes in Computer Science*, pages 387–397. Springer Berlin Heidelberg, October 2011.

[64] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, January 2004.

[65] SIFT keypoint detector. `http://www.cs.ubc.ca/~lowe/keypoints`.

[66] RANSAC algorithm.
`http://www.csse.uwa.edu.au/~pk/research/matlabfns`.

[67] David Vázquez-Padín, Carlos Mosquera, and Fernando Pérez-González. Two-dimensional statistical test for the presence of almost cyclostationarity on images. In *Proceedings of the 17th IEEE International Conference on Image Processing (ICIP)*, pages 1745–1748, September 2010.

[68] Babak Mahdian and Stanislav Saic. A cyclostationarity analysis applied to image forensics. In *Proceedings of the 9th Workshop on Applications of Computer Vision (WACV)*, pages 1–6, December 2009.

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

[70] Oleksiy Koval, Sviatoslav Voloshynovskiy, Fernando Pérez-González, Frédéric Deguillaume, and Thierry Pun. Spread spectrum watermarking for real images: is it everything so hopeless? In *Proceedings of the 12th European Signal Processing Conference (EUSIPCO)*, pages 1477–1480, September 2004.

[71] David Vázquez-Padín and Fernando Pérez-González. Prefilter design for forensic resampling estimation. In *Proceeding of the 3rd IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, December 2011.

[72] Audio Database. `http://opihi.cs.uvic.ca/sound/genres.tar.gz`.

[73] Matthias Kirchner and Rainer Böhme. Hiding traces of resampling in digital images. *IEEE Transactions on Information Forensics and Security*, 3(4):582–592, December 2008.

[74] John R. Deller. Set membership identification in digital signal processing. *IEEE ASSP (Acoustics, Speech, and Signal Processing) Magazine*, 6(4):4–20, October 1989.

[75] Patrick L. Combettes. The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208, February 1993.

[76] Man-Fung Cheung, Stephen Yurkovich, and Kevin M. Passino. An optimal volume ellipsoid algorithm for parameter set estimation. *IEEE Transactions on Automatic Control*, 38(8):1292–1296, August 1993.

[77] David Vázquez-Padín and Pedro Comesaña. ML estimation of the resampling factor. In *Proceedings of the 4th IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 205–210, December 2012.

[78] David Vázquez-Padín, Pedro Comesaña, and Fernando Pérez-González. Set-membership identification of resampled signals. In *Proceedings of the 5th IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 150–155, November 2013.

[79] Thomas Gloe and Rainer Böhme. The Dresden Image Database for benchmarking digital image forensics. In *Proceedings of the 25th ACM Symposium on Applied Computing (SAC)*, pages 1584–1590, March 2010.

[80] Huiying Li and Søren Forchhammer. MPEG-2 video parameter and no reference PSNR estimation. In *Proceedings of the 27th Picture Coding Symposium (PCS)*, pages 1–4, May 2009.

[81] Giuseppe Valenzise, Marco Tagliasacchi, and Stefano Tubaro. Estimating QP and motion vectors in H.264/AVC video from decoded pixels. In *Proceedings of the 2nd ACM workshop on Multimedia in forensics, security and intelligence (MiFor)*, pages 89–92, October 2010.

[82] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved DCT coefficient analysis for forgery localization in JPEG images. In *Proceedings of the 36th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447, May 2011.

[83] David Vázquez-Padín, Marco Fontani, Tiziano Bianchi, Pedro Comesaña, Alessandro Piva, and Mauro Barni. Detection of video double encoding with GOP size estimation. In *Proceedings of the 4th IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 151–156, December 2012.

[84] Zou Chen Lin, Jun Feng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, November 2009.

[85] Jan Lukáš and Jessica Fridrich. Estimation of primary quantization matrix in double compressed JPEG images. In *Digital Forensic Research Workshop*, August 2003.