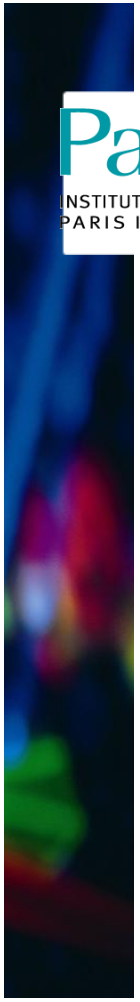
Compression Based Analysis of Image Artifacts: Application to Satellite Images

Avid Roman-Gonzalez



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0062

EDITE ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

Télécom ParisTech

Spécialité “ Signal et Images ”

présentée et soutenue publiquement par

Avid ROMAN GONZALEZ

le 02 Octobre 2013

Compression Based Analysis of Image Artifacts: Application to Satellite Images

Directeur de thèse : **Mihai DATCU**

Jury

Mme.	**Dana SHAPIRA**,	Professeur, Ashkelon Acad. College	Rapporteur
M.	**Peter REINARTZ**,	Professeur, DLR	Rapporteur
Mme.	**Isabelle BLOCH**,	Professeur, TELECOM ParisTech	Examinateur
M.	**Emmanuel TROUVÉ**,	Professeur, Université de Savoie	Examinateur
M.	**Alain GIROS**,	CNES	Examinateur
M.	**Gottfried SCHWARZ,**	DLR	Invité

T H È S E

Télécom ParisTech
école de l’Institut Mines Télécom – membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

To my parents Gregorio Roman Caballero and Alicia Gonzalez Rojas with profound love

Acknowledgments

There are many people without whom this work would not have been possible. First, I would like to thank my thesis director, Mihai Datcu, who always made himself available to me and knew how to motivate me with his enthusiasm and scientific rigor. I would also like to thank Alain Giros for his advice, reviews and tips. Thank you both very much for your support.

In addition, I would like to thank Dana Shapira and Gottfried Schwarz for their advice and support during the development and revision of this work. Also, thanks to the jury for their participation in my defense and for their interest in my thesis work.

This work has been developed in three different places, so I want to thank all the members of the research groups at TELECOM ParisTech - Department TSI, the German Aerospace Center (DLR) - Remote Sensing Technology Institute (IMF), and the Centre National d'Etudes Spatiales (CNES) - AP Service for their assistance during my temporary internships and for their inspiring discussions.

I would like to give special thanks to my parents, Gregorio Roman Caballero and Alicia Gonzalez Rojas, for all their dedicated support over the years, for always being there to give me encouragement and strength to continue to move forward. Thank you to my sister Dunia Terrazas Gonzalez and my brother Jaime Terrazas Gonzalez for their support and their energy.

Finally, I want to express my thanks to Natalia Indira Vargas Cuentas and Karen Fabiola Burga Perez, on whom I could always count for their advice, encouragement and emotional support while living away from my country.

Contents

Abstract

This thesis aims at an automatic detection of artifacts in optical satellite images such as aliasing, A/D conversion problems, striping, and compression noise; in fact, all blemishes that are unusual in an undistorted image.

Artifact detection in Earth observation images becomes increasingly difficult when the resolution of the image improves. For images of low, medium or high resolution, the artifact signatures are sufficiently different from the useful signal, thus allowing their characterization as distortions; however, when the resolution improves, the artifacts have, in terms of signal theory, a similar signature to the interesting objects in an image. Although it is more difficult to detect artifacts in very high resolution images, we need analysis tools that work properly, without impeding the extraction of objects in an image. Furthermore, the detection should be as automatic as possible, given the quantity and ever-increasing volumes of images that make any manual detection illusory. Finally, experience shows that artifacts are not all predictable nor can they be modeled as expected. Thus, any artifact detection shall be as generic as possible, without requiring the modeling of their origin or their impact on an image.

Outside the field of Earth observation, similar detection problems have arisen in multimedia image processing. This includes the evaluation of image quality, compression, watermarking, detecting attacks, image tampering, the montage of photographs, steganalysis, etc. In general, the techniques used to address these problems are based on direct or indirect measurement of intrinsic information and mutual information. Therefore, this thesis has the objective to translate these approaches to artifact detection in Earth observation images, based particularly on the theories of Shannon and Kolmogorov, including approaches for measuring rate-distortion and pattern-recognition based compression. The results from these theories are then used to detect too low or too high complexities, or redundant patterns. The test images being used are from the satellite instruments SPOT, MERIS, etc.

We propose several methods for artifact detection. The first method is using the Rate-Distortion (*RD*) function obtained by compressing an image with different compression factors and examines how an artifact can result in a high degree of regularity or irregularity affecting the attainable compression rate. The second method is using the Normalized Compression Distance (*NCD*) and examines whether artifacts have similar patterns. The third method is using different approaches for *RD* such as the Kolmogorov Structure Function and the Complexity-to-Error Migration (*CEM*) for examining how artifacts can be observed in compression-decompression error maps. Finally, we compare our proposed methods with an existing method based on image quality metrics. The results show that the artifact detection depends on the artifact intensity and the type of surface cover contained in the satellite image.

Chapter 1

Introduction

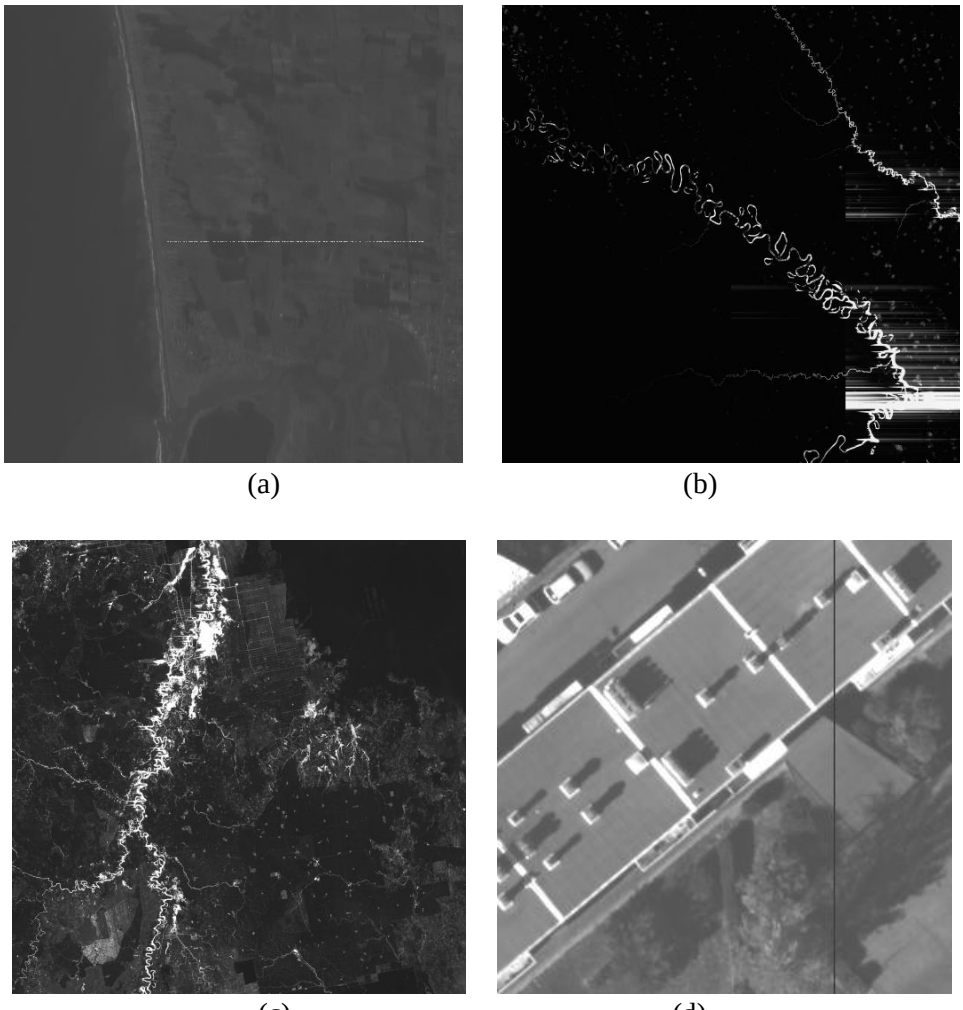
The growing volume of data provided by different imaging instruments requires the use of automated tools to perform application-oriented image analysis routinely; for example, similarity detection, classification, object recognition, etc. All these analysis and interpretation steps may be affected if an image is deteriorated by artifacts. Artifacts are artificial structures being contained in an image product and represent a perturbation of the signal. The artifacts can be produced by a variety of causes. Among them are instrumental effects such as sensor saturation or A/D conversion problems. Further, aliasing effects may occur when the scene contains highly detailed structures that the imaging instrument cannot resolve properly. Finally, data processing may be another source of artifacts; typical examples are compression-decompression effects or calibration residuals.
The presence of artifacts degrades the performance of image analysis, and it makes the analysis process more difficult. The presence of distortions can decrease the efficiency of interpretation and identification algorithms; it may interfere with the recognition of textures and/or the quantitative determination of features; it can also induce errors in the indexing of the images, etc. As intentional markings such as watermarks or image fakery have the same problem and affect the quality of an image, we present a comparison with other methods for image quality assessment published in the literature.

Some typical examples of artifacts are presented in Figure 1.1. In Figure 1.1 (a), we can see a line with partially inconsistent pixel values; these may be due to an intermittently stuck bit in the A/D converter. Figure 1.1 (b) shows trailing charges that sometimes occur during detector read-out. In Figure 1.1 (c), the image is affected by saturation; consequently, no radiometric detail will be available in these areas. In Figure 1.1 (d), we can see a vertical column generated by an un-calibrated dead pixel of a line scan instrument; no information is available in this column.

These artifacts perturb any subsequent interpretation process. The nature of these artifacts can be known or unknown, predictable or unpredictable; some artifacts can be described by models; however, the modeling process is sometimes very difficult. Therefore, it is necessary to implement methods being able to detect hopefully all artifacts regardless of the model which describes their formation. This thesis claims that an artifact is a more

complex or more regular region than the local environment under analysis; the artifact is uncommon when we compare many images, and it is regular itself.

A classical artifact correction approach is to create specific algorithms for each known type of artifact using a model of the artifact characteristics. There are correction methods as presented, for instance, by Jung (Jung et al. 2010) for the restoration of defective image lines; these existing methods aim at specific artifacts; however, other artifacts may remain after applying a specific correction.



(a) (b)

(c) (d)

Figure 1.1: Examples of artifacts found in SPOT images: (a) A/D conversion problems, (b) Trailing charges, (c) Detector saturation, (d) Uncorrected dead pixel creating a dark column.

Artifacts produce changes in images, these changes are visible or not; these changes may result in an alteration of the statistics of the image or other parameters. This situation can be analyzed with image analysis approaches like presence of hidden information, or the presence of a watermark, and/or presence of super-imposed artificial structures in images. What all these image analysis approaches have in common is the analysis of changes in the statistics and information content of the image. It is for this reason that these approaches mostly use mutual information as a basis for the detection of these variations. In that sense,

the general scheme for an imaging system with artifacts is given by Figure 1.2 where *S* is an artifact-free satellite image; *A* is the distortion or artifact introduced by process P_i. The process that introduces an artifact can occur, for instance, during image acquisition, or during image processing. In this scheme, the artifact detection process is located in the processing channel and it can use an artifact model if the model is known. If the model is known, one could correct the artifact. *S'* is the estimated image without artifacts if the correction process is implemented

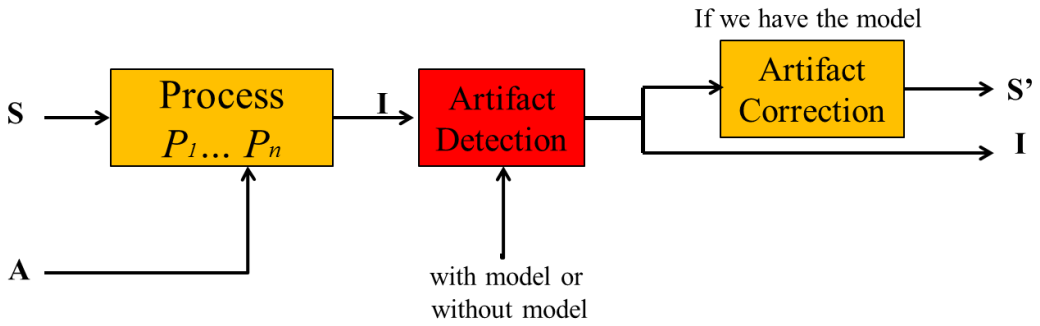


Figure 1.2: General artifact detection scheme. *S* is an artifact-free image product; *A* is an artifact; $P_1 \ldots P_n$ are the different processes that can introduce artifacts; *I* is the artifact affected image; *S'* is the estimated image after artifact correction.

This thesis is a continuation of the work presented by Mallet (Mallet & Datcu 2008a), (Mallet & Datcu 2008b) and Cerra (Cerra et al. 2010). We propose the use of compression techniques−both lossless and lossy compression techniques−as a parameter-free method for artifact detection aiming at aliasing, striping, saturation, etc. The goal for using these compression techniques is to evaluate the level of regularity or irregularity that an artifact may have.

We propose different methods based on compression; they are presented in Figure 1.3. The first method uses lossy compression to calculate the rate-distortion function. Rate-distortion analysis allows us to evaluate how much the image data is being distorted at a given compression rate. We further develop and assess the method contained in (Mallet & Datcu 2008) based on the analysis of the compression error for lossy compression with variable compression rates. The error behavior of the image sectors with artifacts is different from the sectors that do not contain artifacts.

A second method uses lossless compression to calculate the Normalized Compression Distance (*NCD*); the *NCD* is a method proposed in (Li et al. 2004) to determine the similarity between two files using a distance measure based on Kolmogorov complexity.

As a third method we also present some approximations of the Rate-Distortion (*RD*) function using an approximation of the image complexity based on data compression. Here, we do not only present an analysis of the original and the compressed-decompressed image, but also an analysis of the residuals, i.e. the error between the original image and the compressed-decompressed image (Complexity-to-Error Migration). Then we obtain a multi-dimensional analysis of the distortion performance with different compression factors. These

approximate the *RD* curve based on complexity and can be used as a metric for evaluating the image quality.

Finally, the proposed methods are compared with an already existing method which uses image quality metrics for artifact detection based on the work described in (Avcibas et al. 2003) where the authors use image quality metrics for steganalysis.

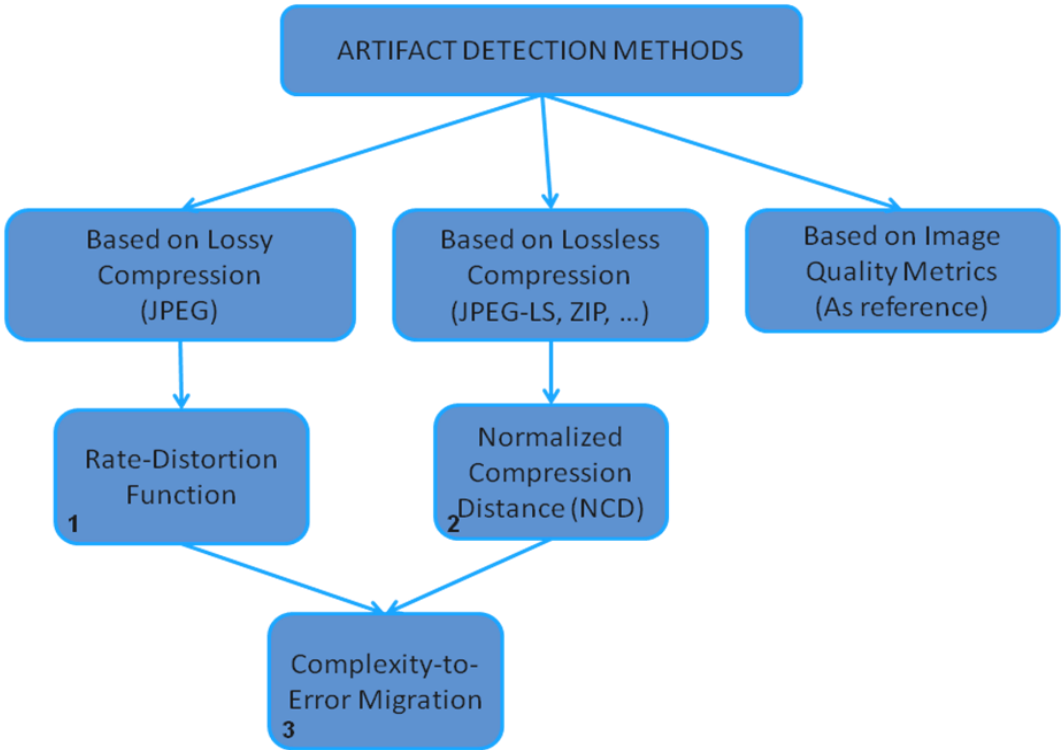


Figure 1.3: Artifact detection methods.

In order to evaluate the various proposed methods, we use a database with synthetic artifacts to analyze the success rate of detection depending on the intensity of the artifact. We use cloud-free images without haze effects in order to avoid problems related to atmospheric phenomena.

The thesis is structured as follows: Chapter 2 presents basic aspects of optical remote sensing and we demonstrate typical artifacts encountered in remote sensing images. Chapter 3 shows similar problems in other application areas. In Chapter 4, we present an overview about information theory, entropy and complexity. Chapter 5 presents new methods for artifact detection based on rate-distortion analysis, artifact detection based on Normalized Compression Distance, artifact detection based on Complexity-to-Error migration, and−as a comparison to an existing method− artifact detection based on image quality metrics. In Chapter 6, we show results and quality metrics applications. Conclusions are contained in Chapter 7.

Chapter 2

Basic Aspects of Optical Remote Sensing

Formally speaking, *remote sensing* deals with extracting information about a remote object. In our case, however, remote sensing is understood as a common measurement technique for airborne or space-borne instruments observing the Earth (Malacara & Thompson 2001). Prominent examples are satellites carrying optical imaging instruments (i.e., space qualified cameras). A typical scenario is shown in Figure 2.1; here, the Sun illuminates the surface of the Earth, while a satellite equipped with a camera is taking images. The digitized image data will then be transmitted from the satellite via a radio link to a ground station on Earth that receives the image data. There, the image data will be further processed, calibrated and interpreted.

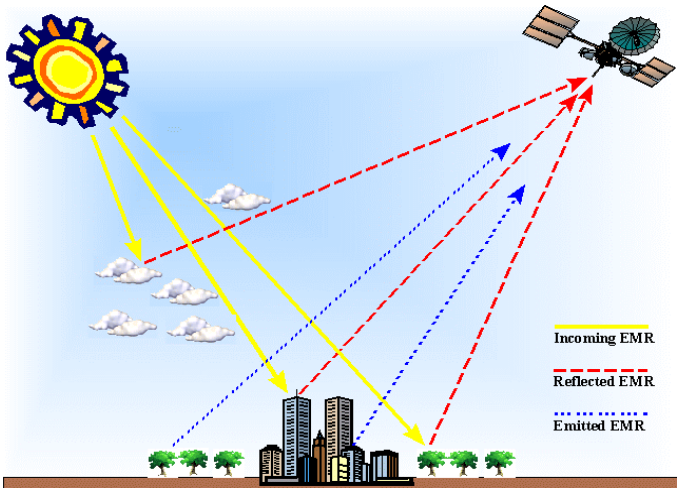


Figure 2.1: Image acquisition by a remote sensing platform [EGI – Energy & Geoscience Institute]

When we acquire images as shown in Figure 2.1, our source of energy is the Sun. The sunlight traverses the atmosphere and is reflected by the surface of the Earth. A fraction

of the overall sunlight will then be collected by the camera aboard the satellite. Apart from the sunlight finally being collected by the instrument, much sunlight will be scattered (i.e. re-directed) and absorbed by various physical processes in the atmosphere and on ground.

2.1 Principles of Electromagnetic Waves

The reflected sunlight seen by the instrument will comprise various wavelengths as shown in Figure 2.2. In principle, we can observe the electromagnetic spectrum from shorter wavelengths (< 10^{-5} μm) to longer wavelengths (> 10^{6} μm). Common optical remote sensing techniques use several regions of this electromagnetic spectrum for different applications.

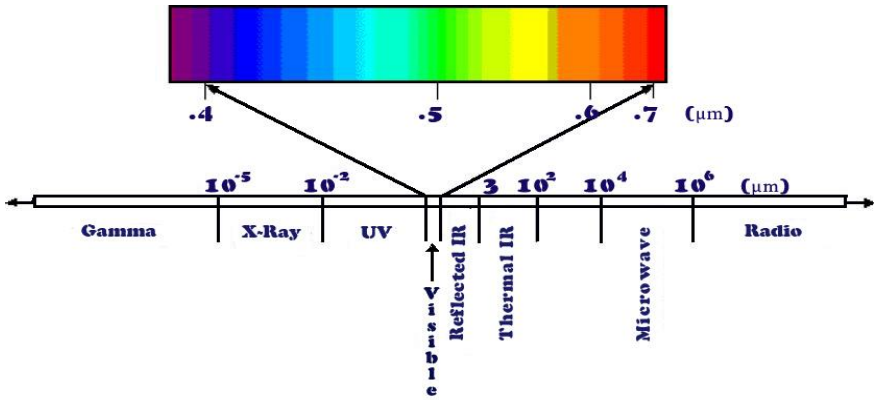


Figure 2.2: Electromagnetic radiation, from shorter wavelengths (< 10^{-5} μm) to longer wavelengths (> 10^{6} μm).

For our applications, we mainly consider the visible and infrared spectrum. The visible spectrum has a wavelength range from approximately 0.4 to 0.7 μm (from violet to red). This range is the portion of the spectrum that comprises the visual colors seen by a human observer. Apart from that, the ultraviolet portion of the spectrum is useful for optical remote sensing of the Earth's surface because some rocks and minerals fluoresce when they are illuminated by ultraviolet radiation (Note that most of the ultraviolet radiation is absorbed by the atmosphere making the fluorescence effects of the land cover invisible to a space-borne instrument). Another range of the spectrum of interest is the infrared region from approximately 0.7 μm to 100 μm. The infrared region can be sub-divided into three regions with different properties: the near-infrared, the mid-infrared, and the far-infrared region. The near-infrared region ranges from approximately 0.7 μm to 2.5 μm; this range is often used to characterize vegetation. The mid-infrared region extends from approximately 2.5 μm to 10 μm; this range is mainly used for the detection of high temperature events like fires or volcano eruptions (the emitted radiation is not a direct reflection of the incident sunlight; the emissions can also be measured on the night side of the Earth). The far-infrared range from approximately 10 μm to 100 μm is used for thermal radiation budget measurements of the Earth and is also not a direct reflection of the incident sunlight.

However, in remote sensing of the Earth's surface we cannot use the full optical spectrum as many spectral regions are masked by atmospheric absorption. The sunlight has to traverse the Earth's atmosphere where particles and gases will affect the radiation due to scattering or absorption. Scattering means a redirection of the photons due to their interaction with particles and gaseous molecules. On the other hand, absorption is a phenomenon where molecules in the atmosphere absorb energy at various wavelengths. In Figure 2.3, we can see a plot of the atmospheric absorptions produced by water vapor and carbon dioxide. The water vapor and the carbon dioxide contained in the atmosphere produce pronounced absorption features mainly around 1.4 and 1.9 µm; these absorptions reduce the reflected energy almost completely, so an optical remote sensing instrument operated in these regions could not get much information about the surface of the Earth.

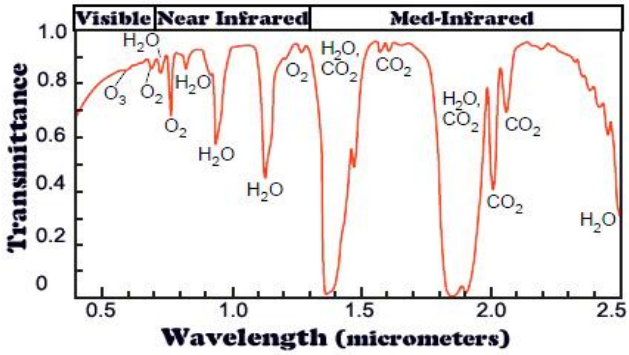


Figure 2.3: Atmospheric absorptions for electromagnetic wavelengths produced by water vapor and carbon dioxide.

2.2 Data Acquisition and Data Reception

Before we deal with artifact detection in optical satellite images, we have to understand the basic image acquisition and processing chain of a classical satellite camera. It collects the electromagnetic radiation entering the instrument and a semiconductor detector (carrying a line or a matrix of "picture elements" = pixels) converts the electromagnetic radiation into electrical signals for each pixel. The electrical charges of the pixels are then read out, amplified and digitized into "digital units" (i.e. brightness counts). The digitized data are then transmitted to a ground station. Within the chain we have to be aware of a number of mostly instrumental effects that may affect and degrade the quality of the acquired images. When the effects have known causes, they can be grouped and called artifacts.

Based on the concept illustrated in Figure 2.4 and Figure 2.5, one can then see where typical artifacts may be introduced into the image generation and processing chain.

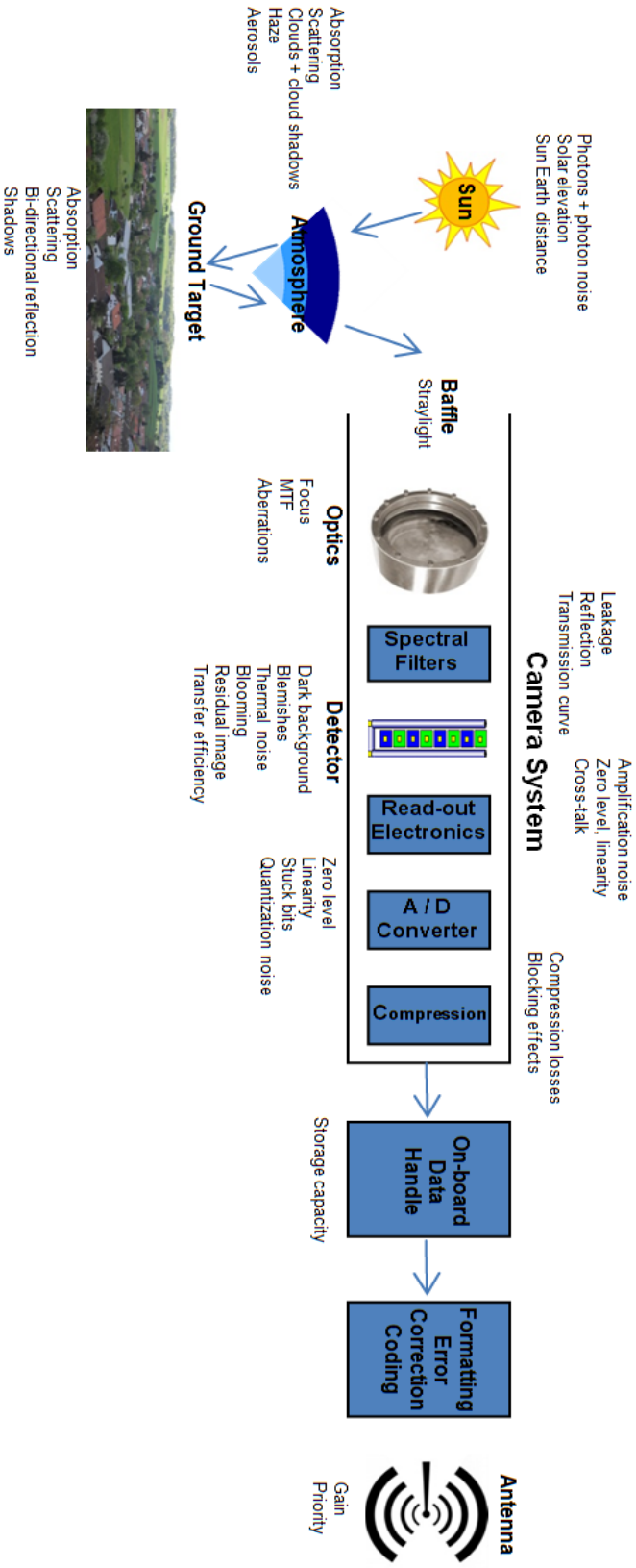


Figure 2.4: Basic image acquisition and processing chain of a classical satellite camera; we can see where typical artifacts may be introduced into the image generation and processing chain (space segment).

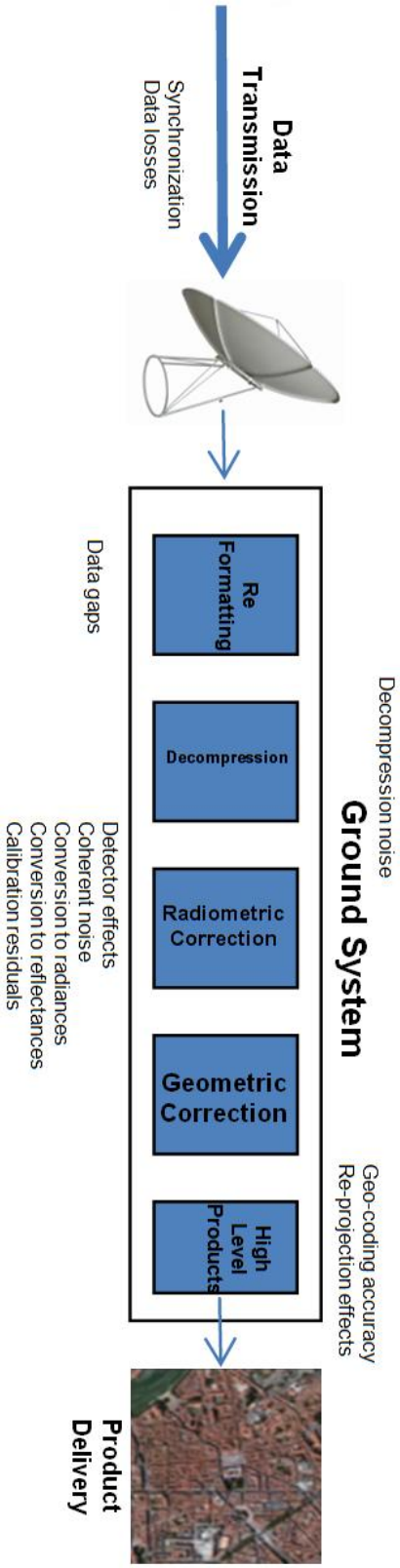


Figure 2.5: Basic image acquisition and processing chain of a classical satellite camera; we can see where typical artifacts may be introduced into the image processing chain (ground segment).

For nearly each of the major components shown in Figure 2.4 and Figure 2.5, there exist typical risks to include artifacts (i.e., effects that cannot be modeled by simple additive Gaussian noise sources). Interestingly, one of the few system components that preclude artifacts is the data transmission link between the satellite and the ground station. Here, the use of modern error recognition and correction codes guarantees a near perfect data transmission provided that the data link is not interrupted and has a sufficient signal-to-noise ratio. As we have sufficient expertise about the nature and the typical characteristics of most of the potential artifacts we can address the problem of identifying them routinely. This will be demonstrated in the next two chapters.

The keywords contained in Figure 2.4 and Figure 2.5 are explained in more detail in Table 2.1. For each optical/electronic component of a satellite camera system we list the typical artifact-prone effects, explain their causes and effects, show what type of technology is most affected by this kind of artifact, and assess the importance and correctability of the effect.

Component	Artifact and Effect	Explanation	Critical Technology	Importance	Correctability
Sun	Photon noise	The solar photon flux fluctuates obeying an arrival process: Photon noise = square root of the number of photons.	This effect is independent of the instrument design and its implementation technology.	The effect yields a basic assessment of the signal-to-noise behavior of an optical system.	Stacking of highly similar images may yield higher signal-to-noise results (for low light level images).
	Solar elevation	The solar elevation determines the amount of photons actually impinging on a unit area on ground. The solar elevation depends on the date and the target area location.	This effect is independent of the instrument design and its implementation technology.	The analysis of time series data requires elevation corrected (i.e. inter-comparable) data.	Simple trig formula; the required parameters may be taken from image metadata or computed based on date, time, and location.
	Sun-to-Earth distance	The actual Sun to Earth distance affects the image brightness.	This effect is independent of the instrument design and its implementation technology.	For image time series data this is a 3% effect.	Simple one-line correction formula approximating the actual distance versus date.
Atmosphere	Absorption	A sizeable percentage of the photons of a given wavelength will be absorbed. The actual absorption depends on the spectral region, the air pressure, temperature, and trace gases such as water vapor and aerosol content of the atmosphere.	One should avoid heavily affected wavelength regions.	The severity of the phenomenon depends on the spectral region.	A high quality modeling of the actual absorption parameters is a demanding task and does not yet correct the effects.
	Scattering	See “Absorption”	Avoid imaging in the blue.	Important for bad contrasts in the blue and for conversion to surface reflectances (if any).	A high quality modeling of the lost and gained signal components is a demanding 3D task and may be not yet a remedy.
	Clouds and their shadows	Clouds prevent imaging of the surface. Cloud shadow regions have a low signal level.	Some instruments provide cloud flags within their higher level products.	Clouds prevent surface classification.	Disregard affected regions (this requires precise cloud annotation in the image product).

Component	Artifact and Effect	Explanation	Critical Technology	Importance	Correctability
Atmosphere	Haze	Haze degrades the image contrast and destroys fine details.	Avoid imaging under hazy conditions.	Important in image time series with different haze levels (this makes inter-comparisons difficult).	One can apply contrast enhancement; however, this may introduce new artifacts (e.g., near edges in images).
	Aerosols	Aerosol effects may range from image contrast reduction to sandstorm imaging.	Avoid aerosol imaging or determine aerosol parameters.	See above	See above
Imaged Surface Area	Absorption	Reduces the surface brightness of some materials.	This effect is independent of the instrument design and its implementation technology.	Allows surface classification.	Shall not be corrected for (needed for image interpretation).
	Scattering	Reduces the surface contrast.	See above	Mostly of minor severity	Shall not be corrected for.
	Bi-directional reflection [distribution] function (BDRF)	The observed surface brightness depends on the local illumination/ observation geometry.	Avoid imaging with sub-optimal geometries.	Could become important for instruments with highly agile pointing.	A correction would require a reliable knowledge of the physical parameters.
	Shadows	May prevent correct surface classification.	Does not depend on the instrument technology (besides contrast fidelity).	Depending on scene details and terrain shape.	Could be required for conversion to surface reflectance.
	Occlusions	Some surface details may be not visible due to a slant viewing geometry.	Avoid slant viewing in highly structured terrain.	Important in special cases only.	Not correctable
Baffle	Stray light	The baffle shall reduce stray light from non-target directions.	There is always a conflict between the desired baffling capabilities and the baffle size.	Images with stray light from sunlit snow surfaces might be degraded.	A full stray light correction is difficult (needs modeling).
Optics	Modulation transfer function (MTF) Focus Aberrations	Sharp transitions are smoothed by the transfer function of the optics, the actual resolution depends on the focusing, and some additional aberrations may be introduced.	Some low pass filtering is often a design goal; after the launch of a spacecraft we often have unwanted mechanical mis-alignments of optical systems. Color aberrations are most often uncritical.	MTF effects in remote sensing are only critical if we are interested in extremely small details.	In principle, MTF effects can be inverted but a correction often introduces new artifacts (e.g., ringing).
Spectral filters	Transmission curve Leakage Reflections	Each spectral filter has a pass-band and shall block all other colors. Some filters show spectral leakages and cause unwanted multiple reflections.	Some desired filter shapes are difficult to produce in space qualified technology. The filter aging characteristics and their reflection behavior has to be tested carefully.	The analysis of multi-spectral images often relies on differences between spectral bands.	Any color correction is rather difficult and calls for extensive modeling.

Component	Artifact and Effect	Explanation	Critical Technology	Importance	Correctability
Detector(s)	Type and operations Responsivity Dark current Blemishes Thermal noise Blooming Transfer efficiency Residual image Ghost images Saturation Color band displacement	Visible sensor artifacts depend on the sensor type, the image recording and the read-out concept (e.g., residual images caused by CMOS detectors, smearing caused by TDI recording, and artifacts due to read-out with on-going illumination). The most visible artifacts are due to non-uniform pixel responsivity, non-uniform dark current signatures, thermal noise, blooming of bright targets, insufficient transfer efficiency during sensor read-out, existence of residual images, or ghost images due to unwanted reflections, saturation of extended areas, or geometrical displacements between color sensors.	Improvements in sensor technology have improved the situation considerably. Sensor butting has become less important; dead or hot pixels, blemishes and striping can be avoided by careful sensor selection. A camera design with reduced read-out speed will reduce the noise level. Detector overflow calls for appropriate camera operations.	The quantitative analysis of digital images calls for artifact-free sensor signals as they distort the recorded scene.	A lot of experience is available how to calibrate sensors with respect to pixel and read-out signatures, how to correct single pixel blemishes by interpolation, and how to characterize thermal noise and residual images.
Read-out electronics	Linearity Zero level stability Low pass effects Cross-talk Multiplexing effects Noise effects	The read-out electronics (and its amplifiers) can add additional artifacts to an image: amplifiers will not be perfectly linear in a thermally varying environment, their zero level may drift, and random noise will be introduced.	The critical point is to design high-speed electronics with low noise.	The most important detail is the fidelity of multiple (i.e., parallel) read-out channels. Each channel needs similar thermal responses.	Linearity, zero levels and low pass effects have to be verified in detail and can be calibrated as systematic effects. In contrast, cross-talk and noise effects often cannot be treated as systematic effects and remain difficult to correct for.
A/D converter	Linearity Zero level Stuck bits Quantization noise	Any technical imperfection during A/D conversion will generate image artifacts.	The A/D converter has to fast enough, shall be linear, and shall not be prone to stuck bits. The transformation of analogue signals into quantized steps will introduce quantization noise that depends on the quantization step size.	The A/C converter quality affects each pixel of every image. Therefore, a good design is very critical.	A known non-linearity can be calibrated. A good zero level stability can be reached via correlated double sampling. Stuck bits need special software tools and quantization noise has to be taken for granted.

Component	Artifact and Effect	Explanation	Critical Technology	Importance	Correctability
Data compression	Compression losses	On-board data compression prior to transmission to ground allows high volume imaging. However, high rate compression causes compression effects (e.g., so-called blocking effects).	Existing data compression techniques (e.g., JPEG type image compression methods) have to be applied in a way not to degrade the image data too much. The analysis of the image content must not be compromised. This can be verified by studying the histograms of images after de-compression.	It is important to select a set of compression parameters that do not falsify the image content to be interpreted. A clever selection of the compression parameters necessitates a lot of test runs with typical examples.	There exist some enhancement methods for the reduction of blocking effects. However, these methods create new problems as they degrade image details.
On-board data handling	Storage capacity	Comfortable imaging needs sufficient storage capacity to store images prior to downlinking them to ground. Otherwise, we are faced with interrupted imaging (the ultimate "artifact").	Nowadays, technology for on-board digital data storage is available.	The data rate produced by an imaging instrument has to be designed in accordance with the data transmission capabilities of the satellite.	One can try to optimize the parallel or sequential operations of all data generating instruments aboard a satellite.
Formatting and error correction coding	(none)	Nowadays, powerful error protection coding is available that provides either perfect data quality or a loss of data.	This technology has become an integral part of data transmission and is no longer critical.	Error protection is a pre-requisite for the transmission of compressed data (e.g., images).	Existing software packages with a proven record should be used.
Data transmission and reception	Antenna gain Antenna pointing Amplifier cooling Receiver synchronization	(The data analysis community is not involved in this field of activity.)	(not for data analysts)	(not for data analysts)	(not for data analysts)
Data re-formatting	Data gaps	(see above)	(see above)	(see above)	(see above)
Data de-compression	Decompres-sion noise	Software packages being used for decompression may include selectable optimization parameters.	(see above)	(see above)	(see above)

Component	Artifact and Effect	Explanation	Critical Technology	Importance	Correctability
Radiometric correction	Calibration concept Detector and read-out effects Conversion to radiances Conversion to reflectances Calibration residuals	Common radiometric calibration concepts contain a signal correction part for known instrumental effects, and routines for the conversion of (corrected) detector counts into physical units (radiances and/or reflectances). Uncorrected effects appear as artifacts. The long term monitoring of the calibration quality may use statistics of residual errors.	The signal correction part and the conversion steps are dynamic processes: after an initial on-ground calibration, in-flight calibration experiences lead to improved methods and results. Often, one faces new calibration problems during the lifetime of an instrument.	A proper radiometric correction is a basic pre-requisite for quantitative image analysis and interpretation. Improperly calibrated images may lead to wrong conclusions.	As a rule, known instrument characteristics that can be corrected with acceptable implementation effort will be contained in common radiometric correction packages. On the other hand, one can never expect a perfect radiometric calibration.
Geometric correction	Rectification approach Geo-coding accuracy Re-projection effects Terrain effects	Geometric correction packages re-project radiometrically corrected images onto a (sometimes selectable) common map projection. (This topic is of secondary importance in this dissertation and we will not provide too many details here.)	One can use rectification approaches ranging from tie-point and interpolation techniques to highly accurate rational function models and additional support by existing DEMs. The selected method determines the kind of artifacts generated during the rectification. Important points to consider are the accuracy of available geo-data, re-projection artifacts, and the handling of terrain effects.	Geometrically corrected images are required for applications that need absolute locations of objects, or the comparison of locations in data of different instruments, etc.	Some geometry routines allow the interactive selection of tie-points. The use of additional tie-points may reduce local blunders.
Higher level products	Algorithmic stability	Derived quantities in higher level products may be imprecise (for instance, vegetation parameters derived from multi-spectral images).	The transformation of basic image products into derived physical quantities needs appropriate algorithms (that are not always available or of good quality).	The acceptance of higher level products by the user community hinges on their correctness.	Comparisons with ground truth measurements can give clues to the quality of the used algorithms but ground truth measurements are not always available.

Table 2.1: List of potential artifacts

On the other hand, if the available image data are not affected by too serious artifacts, the image content can be analyzed and interpreted. Besides the recognition of spatial objects (such as a road or a bridge) in an image, one can − in the case of a camera with sufficiently many separate spectral channels − also use the shape of a spectral curve to identify and distinguish different materials. For example, vegetation has a high reflectance in the near-infrared range, in contrast to inorganic materials (e.g., rocks) that have specific absorption bands that we can use to detect the presence of minerals. This is illustrated in Figure 2.6.

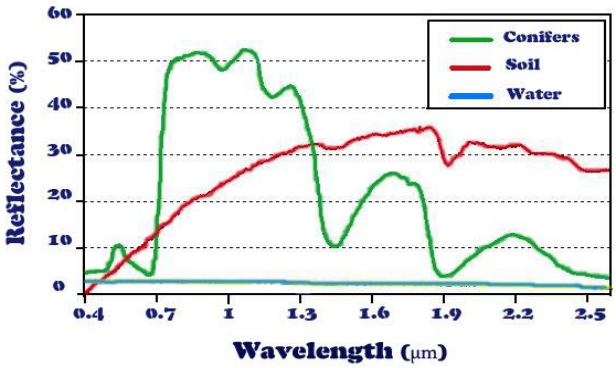


Figure 2.6: Typical spectral signatures; for example, vegetation has a high reflectance in the near-infrared range, in contrast to inorganic materials (e.g., rocks) that have specific absorption bands that we can use to detect the presence of minerals.

The imaging scenario given above has to be discussed in conjunction with the resolution capabilities of a camera. The resolution of a camera can be defined as its spatial resolution (i.e., the capability to discriminate two adjacent point targets), its radiometric resolution (i.e., the capability to discriminate two similar brightness levels), its spectral resolution (i.e., the capability to discriminate two similar colors) and its temporal resolution (i.e., its capability to resolve temporal changes in image time series. These four kinds of resolution will be explained below in more detail.

2.2.1 Spatial Resolution

The spatial resolution of an optical instrument is basically constrained by the size of a detector pixel, the focal length of the camera optics, the distance from the satellite to the target, and the assumption that all neighboring pixels are immediately adjacent. Then these parameters define the nominal footprint of a detector pixel projected onto the surface of the Earth. This footprint size can be used to determine the capability to resolve two adjacent point targets. However, the footprint is defined for an ideal instrument; in practice, the atmosphere may blur the image, the camera optics has a defined transfer function, the detector and its read-out electronics may act as low pass filters, etc. As an additional complication, the term “spatial resolution” is often confounded with the pixel spacing used

for the representation of an image product. The actual pixel spacing used in a digital image – for instance, after a geometric re-projection – may correspond to a pixel grid that differs considerably from a grid commensurate with the ideal spatial resolution.

2.2.2 Radiometric Resolution

The radiometric resolution of a camera is limited by the number of bits provided by the A/D converter that quantizes the electrical charges (i.e., the electrons) of a detector pixel (after read-out and amplification) into digital units. Of course, this limitation is not yet a fully valid criterion: We also must be sure that the lower bits of the A/D converter do not only contain noise but useful information. Otherwise, we would have a nominal radiometric resolution that does not fulfill its promises. Other points to consider are the full-well capacity of the detector pixels and the setting of the read-out amplification factor prior to A/D conversion. The amplification factor has to be selected according to the signal level, the signal dynamics, and the quantization capabilities of the A/D converter.

2.2.3 Spectral Resolution

The spectral resolution of a remote sensing instrument is determined by the number of individual spectral bands that the sensor(s) can detect and handle. Classical remote sensing cameras have four or more spectral bands: at least a red band, a green band, a blue band, and an infrared band; there are also instruments which produce multispectral images that have approximately 10 to 15 bands. In addition, there are also hyperspectral imagers that provide more than 100 spectral bands. From a technical standpoint this means that either individual detectors are available for each spectral band (a classical design approach with sometimes displaced viewing directions for each spectral band), or a detector provides on-chip color band discrimination (for instance, a so-called Bayer type image sensor), or – in the case of hyperspectral instruments – a prism generates a multi-element color spectrum for each single point target on ground to be recorded as a full sensor line (the "imaging spectrometer" concept). Within this framework, we have to consider that the technical solution adopted for a specific camera has to provide a reasonable signal-to-noise level: narrow spectral bands have a lower signal level than broader spectral bands. The lack of sufficient energy can only be compensated by longer exposure times; however, the ground-track motion of the satellite and the read-out strategy of the detectors constrains the admissible exposure times.

2.2.4 Temporal Resolution

When a satellite is orbiting the Earth on a (near-) polar orbit, also the Earth is rotating about its axis. While a typical satellite orbit has an orbit period of about 100 minutes, the Earth requires 24 hours for a full rotation. This means that after a full satellite orbit cycle, the Earth has rotated by more than 20 degrees and our new satellite ground track will be displaced form the previous one accordingly. Thus, it takes some time until the same location on the Earth's surface can be imaged again. This re-visit time period (usually several days) limits the temporal resolution of a satellite camera. The actual temporal resolution of a camera

system depends on more factors, including the lateral (off-nadir) pointing capability of the camera to re-acquire the same location again by slant viewing, the swath width of the sensor and its desired overlap, and the geographical latitude of the target area: Near the equator the longitudinal inter-track spacing is greatest; near the poles, it is minimal, but we also have to take into account that cloud cover could make a number of image takes unusable and this could lead to a degraded temporal resolution.

2.3 Earth Observation Image Information Content and Quality

In this section, we will describe the information extraction from satellite images and the importance of image quality for satellite images. We will also study in more detail the artifacts introduced in Section 2.2 and their influence on information extraction.

We concentrate on the information that one can extract from these images in order to interpret and apply this information in different fields. Here, the image quality plays an important role. If we want to get the highest amount of information from an image, we need to have a good image quality.

This is a pre-requisite for the detection, measurement, identification and interpretation of different targets. Targets in remote sensing images may be any feature, object, texture, shape, structure, spectrum, or land cover being contained in the image.

Data processing and analysis in remote sensing can be performed either manually or automatically. Currently, there are many research groups that develop automated tools to detect, identify, extract information and interpret targets without manual intervention by a human interpreter.

2.3.1 Information Content

To extract the information contained in satellite images, one has to extract different characteristics such as color, texture or shape.

Color features based on spectral information can be used to extract information from multi-band image pixels for analysis, classification, indexing, segmentation, etc. A typical example is the analysis and comparison of spectral signatures.

Another way to extract information from an image is using its texture. In general, texture is defined as a segment with homogeneous properties or characteristics; it represents a repetition of motifs that create a visually homogeneous image maintaining a spatial relation. There are many researches in this field to characterize the different spatial relationships and evaluate the amount of information that can be extracted from satellite images as shown in (Dowman & Peacegood 1989). For instance, Haralick et al. 1973 used statistical measures to discriminate the different structures in an image; the authors propose 14 features calculated from a co-occurrence matrix corresponding to the second-order statistics of pixel neighborhoods.

The extraction of shape information is based on geometric characteristics such as area, eccentricity, longitude axis, invariant algebraic momentum, etc.

It is also possible to combine spectral, texture and geometric characteristics for the analysis of satellite images that results in extracted features.

Figure 2.7: Example of remote sensing image analysis [International Charter on Space and Major Disasters].

After a feature extraction process, a classification step is performed; for classification, one can use supervised or unsupervised classification. Supervised classification needs data for training; one needs a good selection of training data with relevant and appropriate samples to obtain an optimal parameterization. The parameters that are the result of the training step are then used for the final classification of the remaining data.

In contrast, unsupervised classification does not need a training step. This type of classification uses clustering algorithms to determine the grouping of the data. The type of algorithm determines which features will be the most predominant ones for clustering−either color, texture or shape. In most cases, however, one has to specify the number of data clusters, the selected distance parameter, etc.

Whatever classification type will be used, the final result of the analysis process is a mosaic either composed of pixels or image patches in which each element represents a category of interest.

2.3.2 Earth Observation Image Quality

A satellite image must comply with the needs of end users, thus the required quality of the image depends on the type of application, which could be photo-interpretation, forest and deforestation monitoring, agricultural land monitoring, meteorology, water color analysis, monitoring of natural disasters (earthquakes, floods, etc.), defense applications, etc.

Generally, there are well-known criteria to evaluate the quality of a satellite image in three important domains: Geometric quality, radiometric quality, and quality aspects related to the actual resolution.

In the geometric domain of images, where the interesting topic is to determine the position of the pixels with high accuracy, the criteria to define the image quality have to evaluate the preservation of locations and distances, including the precise overlapping of color bands, and accurate heights.

In the radiometric domain, the important thing is the brightness level of each pixel; the criteria to be evaluated are the calibration accuracy, the radiometric noise, etc.

In the domain related to resolution, the important issues are the resolution capabilities, and the perception and reproduction of details; the topics which must be taken into account for the quality assessment are the sampling technique, the modulation transfer function, etc.

These criteria for assessing the quality of a satellite image serve to define the specifications of different components for an Earth observation project and to assess its capabilities to satisfy the user needs.

2.3.3 Earth Observation Image Artifacts

In the literature, we found no standard definition of an image artifact. Therefore, we had to find a definition based on our experiences. As will be shown in Section 2.3.4 in more detail,

- We define artifacts as artificial structures that represent a structured perturbation of the signal.

In most cases, these artifacts are produced by the instrument; for instance, technical design problems, saturation of the sensor, or on-board signal processing. Therefore, these artifacts induce errors during image analysis.

In the following, we will describe typical artifacts that may occur (or have been simulated) in images that were taken by the SPOT 5 instrument, by IKONOS, by MERIS, and by the hyperspectral instrument ROSIS. Each of these instruments is prone to typical artifacts. Therefore, we begin with a short survey of each instrument and its critical components.

SPOT 5 (launched in May 2002) represents a classical optical line scan instrument with 60 km swath width that has a long heritage of predecessor instruments dating back to SPOT 1, SPOT 2, etc. The polar and sun-synchronous orbit of SPOT 5 is characterized by its mean height of 822 km, its period of 101 minutes and its equator crossing time (on its descending branch) of 10:30 AM local time. The orbit repeat cycle of 26 days (combined with off-nadir imaging) allows a re-visit period of a target area within typically 2 to 3 days.

SPOT 5 carries two camera instruments to be considered here: HRG and HRS. (The SPOT 5 payload also comprises an optical vegetation monitoring instrument that is not considered here.) HRG represents a typical multi-color instrument with a high resolution panchromatic channel (2.5 or 5 m resolution) and four additional color channels with lower resolution (10 m or 20 m resolution). All channels are recorded by line sensors. HRS is a stereo pair generating instrument delivering simultaneously acquired and geometrically overlapping forward and backward looking strip images. The image strips acquired by CCD line sensors (via push-broom technology) have a panchromatic resolution of at least 10 m and a swath width of 120 km and can be used to generate digital elevation models.

The A/D conversion of SPOT 5 delivers 8 bits per sample. This should allow us to see persistent artifacts that are not due to an excessive signal amplification and quantization.

When we concentrate on potential artifacts of the HRG instrument, we can expect to see all typical features of a classical multi-spectral push-broom instrument such as residual detector blemishes and discrepancies between the individual color bands. As SPOT 5 can be considered as a typical "workhorse instrument", also the SPOT 5 artifacts should be very typical of this type of instruments. Thus, SPOT 5 images are a good choice for studying artifacts in satellite images.

Another instrument is IKONOS, a very high resolution push-broom imager: IKONOS was the first commercial sub-meter resolution satellite to acquire panchromatic image strips with 0.82 meter resolution and multispectral images with 3.2 meter resolution. The high resolution is reached by its telescope focal length of nearly 10 m and represents the state of civilian technology of 1999 that developed further, of course, during the last years (e.g., into the Pleïades or WorldView satellite series).

The polar and sun-synchronous orbit of IKONOS is characterized by its mean height of 681 km, its period of 98 minutes and its equator crossing time (on its descending branch) of 10:30 AM local time. This, together with the off-nadir pointing capabilities of the spacecraft, and an instrument swath width of 12 km allows a revisit period of a target area of about 3 days. The downlinked image data can be acquired by several ground stations around the world that can be operated independently from each other.

IKONOS provides simultaneously recorded panchromatic and multispectral image strips via separate line detectors: a panchromatic CCD detector with 13,500 cross-track pixels (operated in TDI mode), and 4 photodiode color detectors with 3375 cross-track pixels. The color separation is accomplished by multi-layer spectral filters on glass. Besides its high geometrical resolution, IKONOS also provides a high radiometric resolution: The A/D converter delivers quantized data with 11 bits per sample. These data will be compressed via adaptive differential Pulse Code Modulation with a compression rate of 4.25.

As a result of these demanding design parameters, one can expect typical geometrical and brightness artifacts due to the very stringent spatial and temporal imaging requirements such as contrast degradations, residual TDI mismatch, etc. A critical issue is the attainable image classification accuracy.

Finally, we will also consider the MERIS instrument, an imaging spectrometer operating in the visible and near-infrared spectral range from 390 nm to 1040 nm. From this overall spectral range, one can select 15 spectral bands by ground command.

MERIS was flown aboard the now defunct ENVISAT satellite with the following orbit parameters: polar sun-synchronous orbit with a mean height of 800 km, a period 101 minutes and an equator crossing time (on its descending branch) of 10:00 AM local time. These parameters yield a repeat cycle of 35 days.

One of the most important points for artifact detection is the wide cross-track field of view of the MERIS instrument covering 68.5° requiring a polarization scrambler and resulting in a swath width of 1150 km; technically, this wide field of view is accomplished by installing five identical optical modules arranged in a side-by-side configuration. As each module has to be calibrated individually, small residual offsets among the calibration parameters of a module will immediately cause visible offsets in image mosaics.

Within each optical module, MERIS uses a separate frame detector with an imaging zone of 740 spatial × 520 spectral pixels. Each frame detector consists of a CCD where the spatial information is recorded in an image row (following a classical push-broom imaging concept), while the spectral content of a push-broom row is mapped along the columns of the CCD. This concept necessitates elaborate read-out strategies that result in spectra with a high signal-to-noise ratio. One can imagine that a number of potential read-out artifacts may jeopardize the quality of the spectra. The radiometric quality of the data is preserved by an A/D conversion with 12 bits per sample. This high resolution necessitates elaborate calibration concepts that are − of course − again prone to artifacts. Therefore, MERIS data are a prime target for artifacts appearing as small but sometimes annoying spatial and/or spectral details.

ROSIS (Reflective Optics System Imaging Spectrometer) is a hyperspectral airborne instrument; it has 115 spectral bands in the range from 430 to 860 nm with 14 bits of radiometric resolution. Typical artifacts for this class of instruments are radiometric calibration errors due to insufficiently corrected atmospheric effects (e.g., water vapor profile residues).

Figure 2.8 shows some examples of artifacts, in (a) and (b) we can see texture changes due to aliasing; (c) and (d) show the existence of horizontal lines which could be erroneously detected as bridges; (e) shows saturation, and (f) shows blocking effects; (g) and (h) contain strips of the 2^1 bit in one band of the hyperspectral image. These artifacts may interfere with the recognition of a texture, target identification, land cover segmentation or the quantification of features. The nature of some of these artifacts is mostly known; however, these artifacts cannot be described by a single model; that is why we aim to detect these artifacts regardless of the specific formation model, i.e. we look for parameter-free methods.

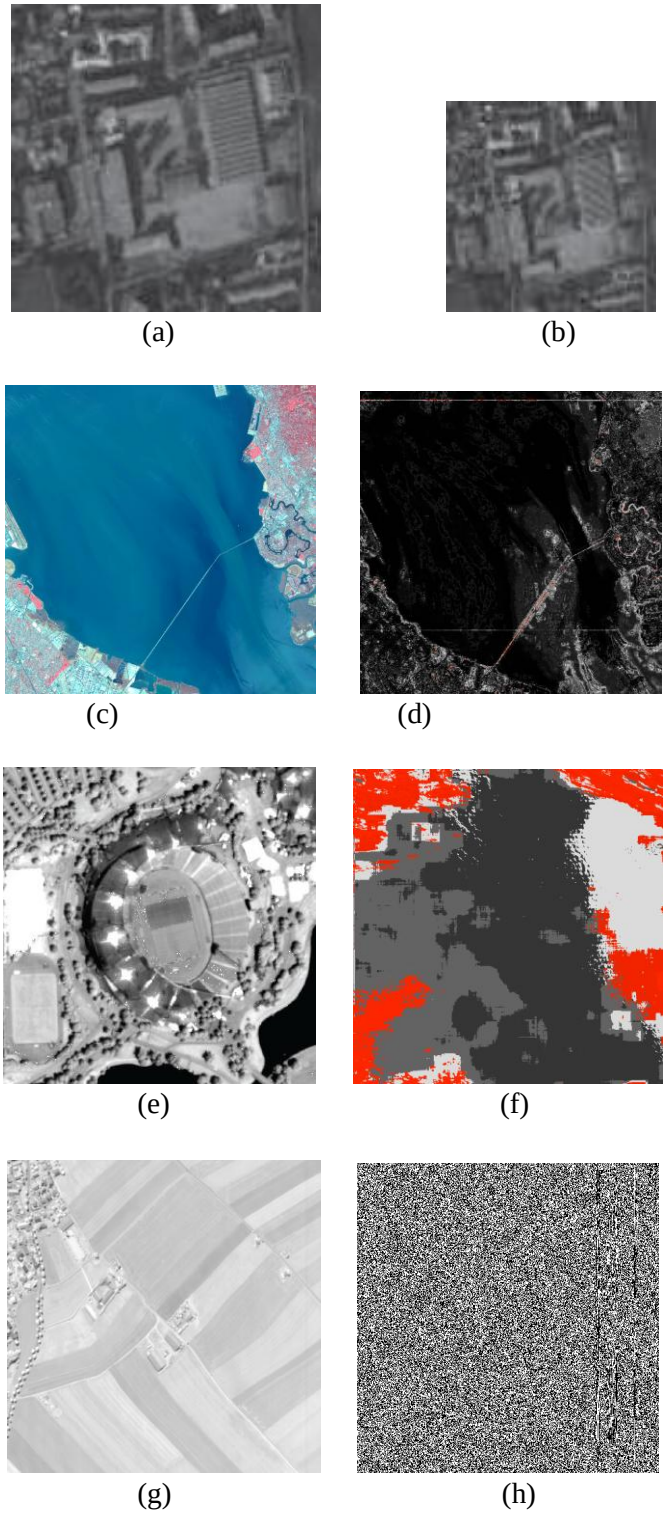
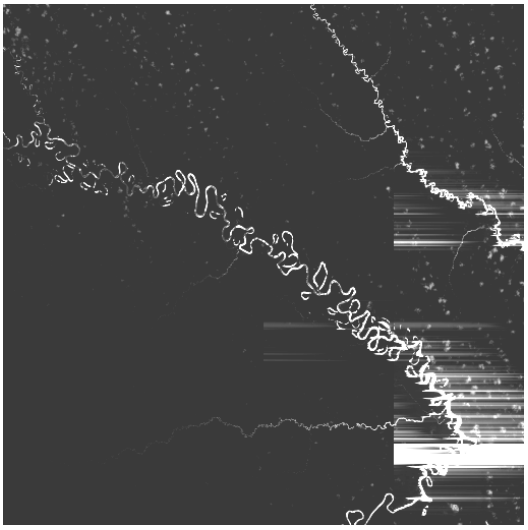
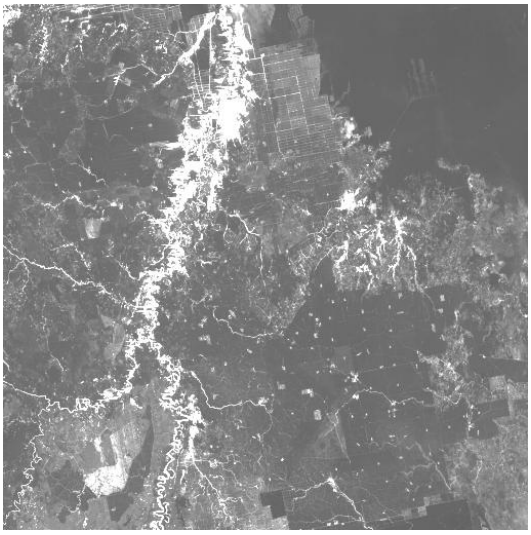


Figure 2.8 Typical examples of artifacts: (a) and (b) Change of texture by aliasing after processing. (c) and (d) Horizontal lines appearing after contrast enhancement (e) Sensor saturation. (f) Blocking effects. (g) and (h) contain strips of the 2^1 bit in one band of the hyperspectral image.

(a) A/D conversion problem (SPOT)



(b) Trailing charges



(c) Saturation

(d) Dead column

Figure 2.9: Some examples of our artifact database: (a) A/D conversion problem (SPOT) (b) Trailing charges. (c) Sensor saturation. (d) Dead column.

In the following, we present and describe some specific actual artifacts found in satellite images in more detail. In Figure 2.9, we see an excerpt of our database which has been developed to evaluate artifact detection methods. In Figure 2.9 (a), we show an A/D conversion problem in a SPOT image; this defect represents an electronic signal disturbance and appears as a "salt and pepper" row in the image. Figure 2.9 (b) illustrates a trailing

charge problem: during detector read-out, a high brightness pixel creates a decaying brightness trail.

In Figure 2.9 (c), we see saturation problems; saturation occurs when the sensor reaches its maximum full-well capacity; this leads to a loss of information because the sensor does not measure the true value; saturation often produces side effects in adjacent pixels when they also become saturated. In Figure 2.9 (d) we can see a dead column; this defect occurs due to an uncorrected dead pixel of a push-broom line sensor. If we use a frame type sensor, the presence of an uncorrected dead pixel would yield a black point in the image.

In general, the generation of a standard product of a satellite image includes a correction of dead pixels, etc. (Jung et al. 2010); however, some artifacts may be remaining after this process.

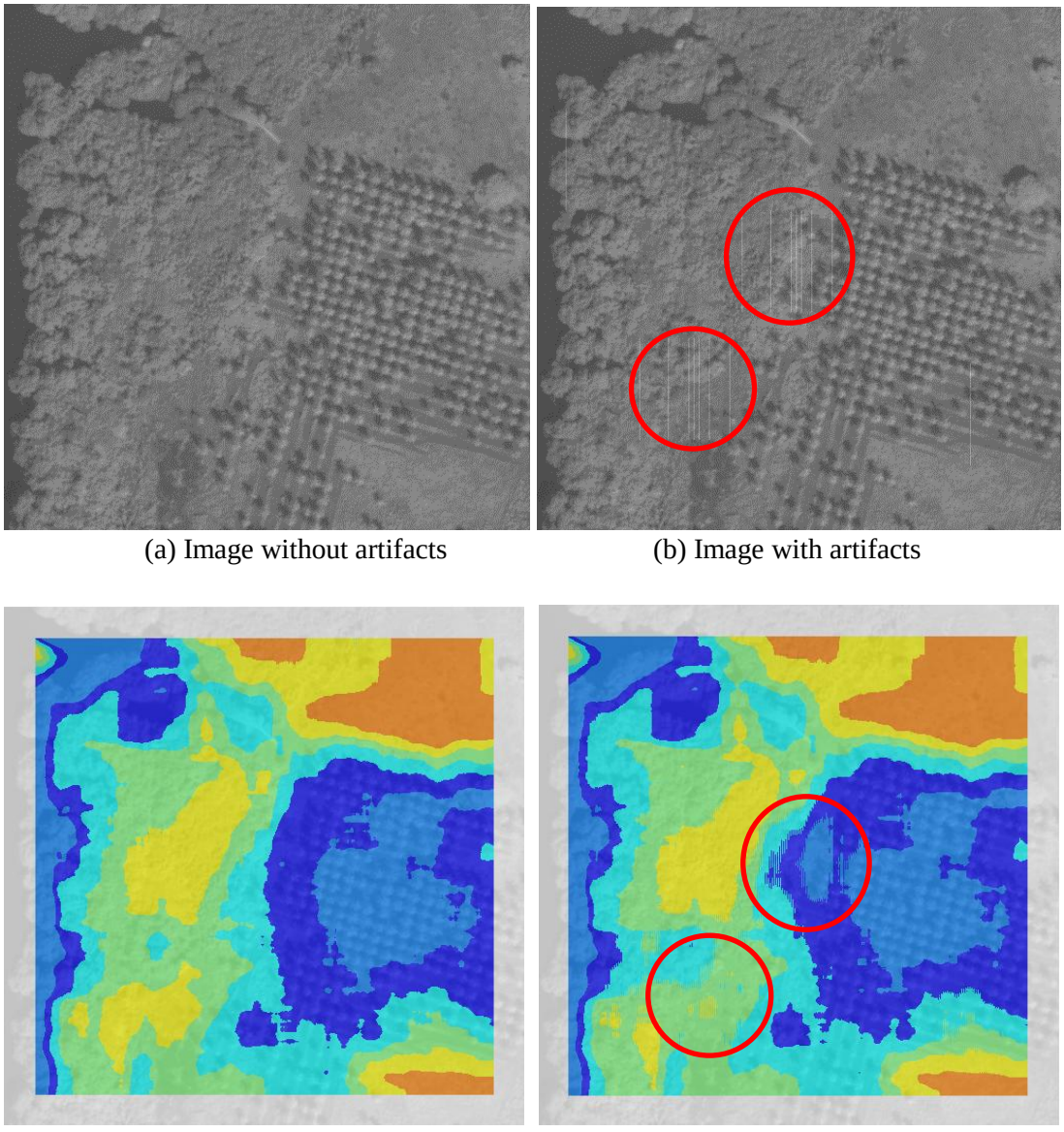
2.3.4 Impact of Artifacts on Image Analysis

Automatic image analysis tools such as similarity detection, classification, pattern recognition, etc. often rely on data of sufficient quality; if they cannot take into account some defects like noise, blurring, etc. they are not universal. Artifacts can complicate the analysis of images and may decrease the efficiency of the analysis process; but we do not know exactly how the artifacts affect the image analysis. In this section, we present an assessment of the classification variation due to artifacts being present in the satellite images. For making the evaluation, we select different feature extraction processes such as Gabor Wavelet features presented in (Manjunath & Ma 1996), quadrature mirror filters (*QMF*) used in (Campedel et al.; Simoncelli et al. 1989), and features based on co-occurrence matrix analysis (Presutti 2004).

The Gabor Wavelet features contain the average energy output for each filter; this analysis uses the spatial and frequency components to analyze differences between textures. The result is a direct response from the decomposition of the original image into several filtered images with limited spectral information; the method is used as a simple statistical characteristic of gray-scale values of the filtered images using the k-means algorithm.

In a *QMF* bank, a pair of parallel filters is used followed by sub-samplers; the resulting features are quantized and coded using an entropy encoder. Again, classes are assigned following the k-means algorithm.

The co-occurrence matrix describes the frequency of a gray level that appears in a specific spatial relationship with another gray value within the area of a particular window. The co-occurrence matrix is a summary of how often pixel values occur adjacent to another value in a small window. The k-means method gives us classes.



(a) Image without artifacts (b) Image with artifacts

(c) Classification of image (a) (d) Classification of image (b)

Figure 2.10: Example of two artifact assessments with two different image classification results; the variations represent between 4 and 5 percent of the pixels.

Figure 2.10 shows two image classification results. Figure 2.10 (a) contains a forest scene for image classification. Figure 2.10 (b) shows the same image as Figure 2.10 (a), but with some manually introduced artifacts; these artifacts are defective columns introduced into the red circled zone. We classified the images of Figure 2.10(a) and Figure 2.10(b); a five group classification was made for different forest types; we can see the differences of

the classification results in Figure 2.10 (c) and Figure 2.10 (d); the differences affect between 4 and 5 percent of the pixels in the indicated zones.

In general, the presence of artifacts in satellite images can produce discrepancies of classification or indexing results; the differences affect the classes (e.g., city, forest, sea, or agriculture); the variation can range from 3 to 10 percent and − depending on the application − the error can become very important.

In Figure 2.11, we see another example of how artifacts affect the classification results of 8 bit satellite images. This example shows a forest image, which was classified into three groups using QMF features. In a second run, some synthetic artifacts were introduced into the image; in this case, 16 vertical strips with a width of 1 pixel and a length of 100 pixels and selected brightness levels were added to the original image (having a mean brightness level of 39). We can see that for an additional brightness level of 10, there is a variation with respect to the classification of the original image, and if we use strips with an additional brightness level of 20, the variation is much greater. The percentages of classification differences are shown in Table 2.2 where we show the values for each class and for different strip intensities.

Added Brightness	**Class 1 Blue**	**Class 2 Cyan**	**Class 3 Yellow**
0	46.20 %	42.98 %	10.82 %
10	46.30 %	43.21 %	10.49 %
20	42.60 %	46.30 %	11.10 %

Table 2.2: Classification percentage for different strip intensities

The classification was made for 3 classes, (class 1 coded in blue, class 2 coded in cyan, and class 3 coded in yellow). The introduction of striping with an intensity of 10 produces a minor classification variation of 0.66% in total, which can be clearly seen in Figure 2.11. This minor variation in the classification results has to be understood in conjunction with the mean image brightness of 42 counts; however, when we add striping with an intensity of 20, this leads to a classification variation of 4% in total, which obviously is a big difference and can − depending on the application − dramatically affect the classification and indexing results.

FOREST

Original image

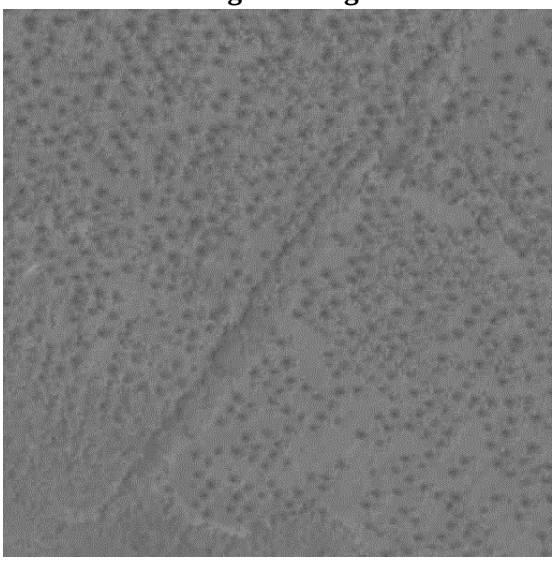
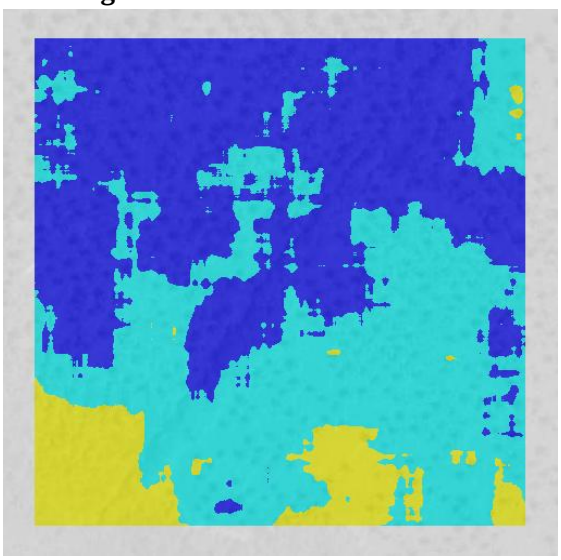
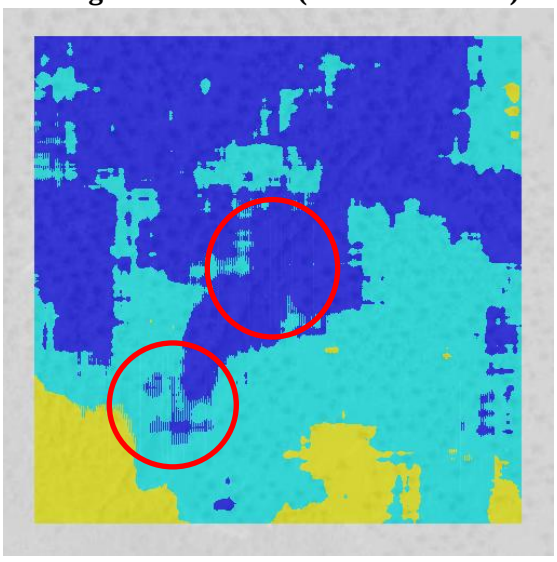


Image classification without artifacts



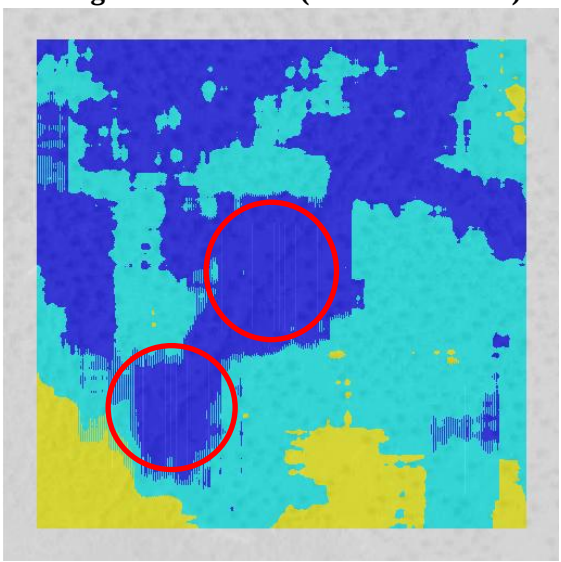
QMF features - 3 classes (449×449 pixels)
Class 1 -> 46.20 % (blue)
Class 2 -> 42.98 % (cyan)
Class 3 -> 10.82 % (yellow)

Image with artifacts (10 counts added)



QMF features - 3 classes (449×449 pixels)
Class 1 -> 46.30 % (blue)
Class 2 -> 43.21 % (cyan)
Class 3 -> 10.49 % (yellow)

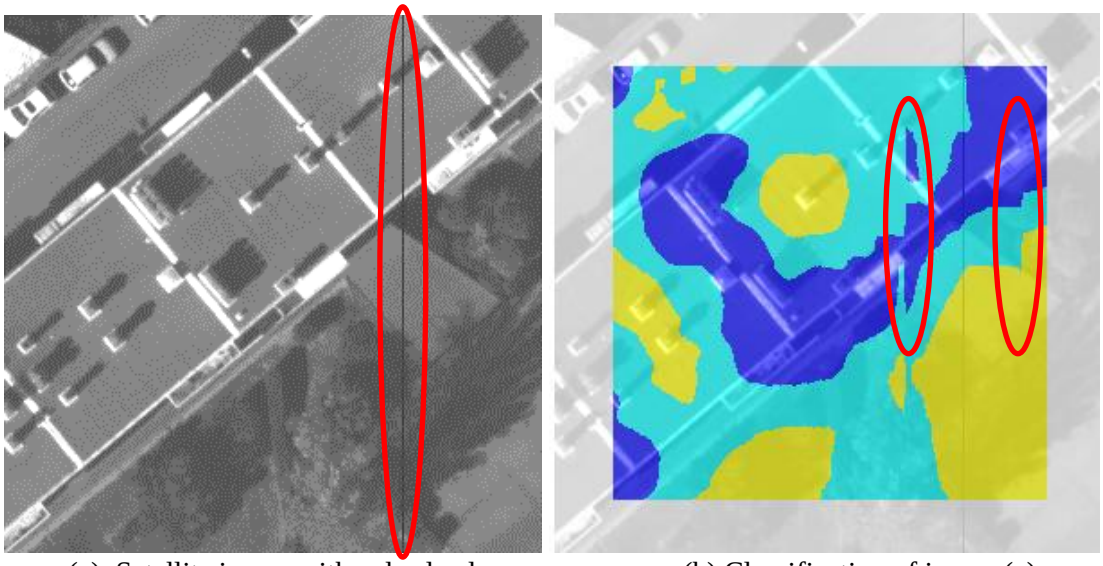
Image with artifacts (20 counts added)



QMF features - 3 classes (449×449 pixels)
Class 1 -> 42.60 % (blue)
Class 2 -> 46.30 % (cyan)
Class 3 -> 11.10 % (yellow)

Figure 2.11: Assessment of synthetic artifacts: a forest image, which was classified into 3 classes using QMF features and the k-means algorithm.

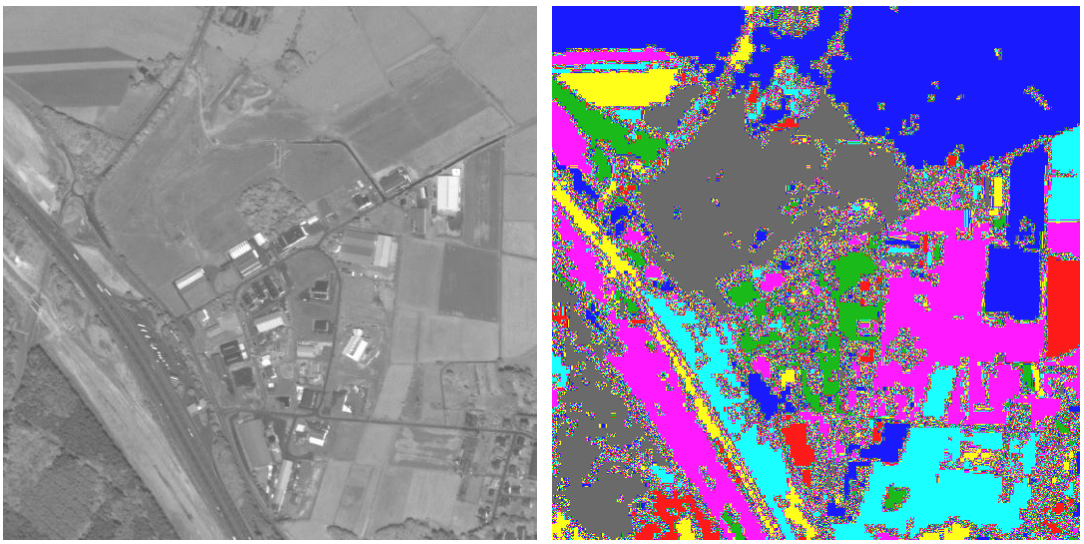
Another artifact example is shown in Figure 2.12. In this case, we see an air-borne image product with a dead column. Here we cannot make a quantitative assessment as to the classification changes because we do not have the corresponding undisturbed image. However, we can visually observe that the presence of this column produces two strange laterally displaced patterns (see the red ellipse in Figure 2.12 (b)). For the classification, we used Gabor filter features and a patch size of 64×64 pixels.



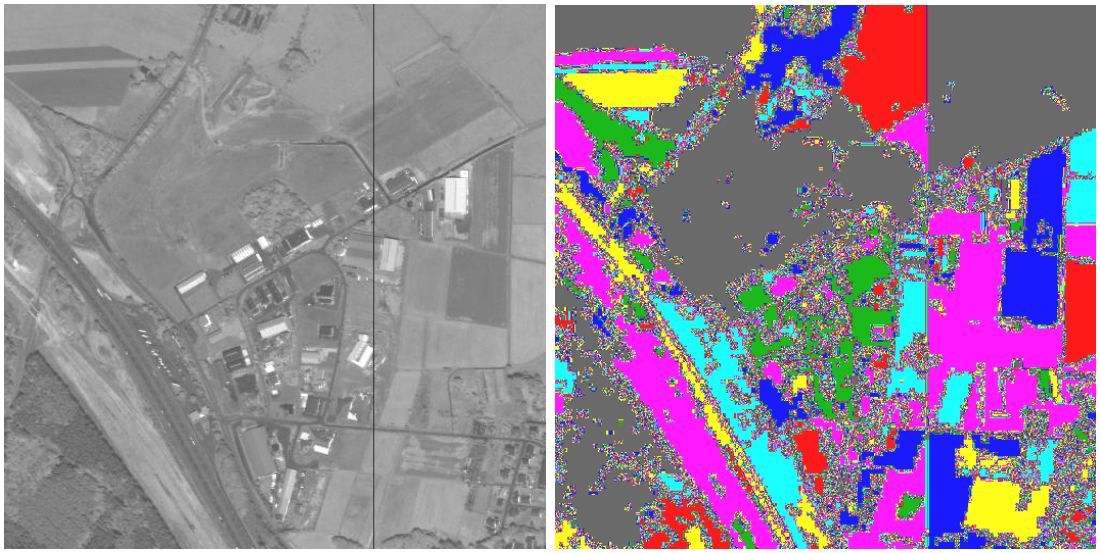
(a) Satellite image with a dead column (b) Classification of image (a)

Figure 2.12: Artifacts of an actual image (presence of a dead column).

Another artifact example is shown in Figure 2.13. In this case, we applied a segmentation algorithm to a satellite image without apparent artifacts; after that, we introduced a column in the image simulating a dead pixel, and applied the segmentation algorithm again.



(a) Satellite image without artifacts (b) Segmentation of image (a)



(c) Satellite image with dead column artifact (d) Segmentation of image (c)

Figure 2.13: Artifact assessment for images with a column artifact; in this case, we applied a segmentation algorithm (EDISON software package).

We can see that the presence of the artifact changes the result of the segmentation algorithm. The segmentation algorithm was the Mean-shift algorithm implemented in the EDISON software package (Comaniciu & Meer 2002). The Mean-shift algorithm is a non-parametric feature-space analysis technique used for clustering data in computer vision and image processing. EDISON stands for Edge Detection and Image SegmentatiON, a system with algorithms described in (Comaniciu & Meer 2002).

2.4 Conclusions

In this chapter, we have seen a basic description of the principles of remote sensing cameras; in addition, we described the main components of the acquisition chain of satellite images.

We also identified possible stages which could produce artifacts; for example, a sensor that does not work properly can produce strip artifacts; the instrument transfer function may introduce aliasing; further, blocking artifacts due to lossy image compression may be generated. In addition, we outlined the importance of information extraction from satellite images and the importance of image quality during the information extraction process. The image quality is affected by the presence of artifacts. We also described some artifacts and their influence on the information extraction from satellite images. In summary,

- An artifact always alters the image information content
- An artifact alters the image quality either apparently or not
- An artifact may have a local or global impact
- An artifact affects each image analysis process differently

Therefore, our definition given in Section 2.3.3 was: "We define artifacts as artificial structures that represent a structured perturbation of the signal".

Due to the importance of a correct analysis process, it is necessary to design methods that reliably detect the artifacts. However, the detection of artifacts has to be seen from an image quality standpoint and we have to understand the causes of the artifacts.

Therefore, the next chapter shows how to better understand image quality and its degradation due to different causes. To this end, we also have to include other related fields such as multimedia image quality, watermarking, steganalysis and image fakery.

Chapter 3

Hidden Information Analysis: A Base for Artifact Detection

When we look at artifact detection from a scientific viewpoint, we have to consider the problem at large and its relationships to various fields of science. First of all, we have to understand the nature of image artifacts based on the principles of signal processing. We also have to know under which theoretical and practical conditions we have to face potential image degradations. This does not only refer to instrumental effects in images that have been described from a technical perspective of optical instruments in Chapter 2; now we have to include the principles of prescribed pseudo-artifacts generated intentionally for watermarking, steganography and image fakery. This obviously refers to the field of information theory with all its approaches how to generate hidden information, how to include it to existing data sets and, on the other hand, how to detect and retrieve this hidden information. This includes, of course, a lot of metrics that already exist in the community.

In order to detect artifacts and to analyze them efficiently we have to find ways how to describe artifacts from a mathematical perspective. To this end, we have to select approaches based on statistics and related models. In our case, this leads to a selection of techniques based on information content and mutual information. In the following, we start with basic metrics and technical descriptions in this chapter, while the principles of information content, entropy and complexity will be dealt with in Chapter 4. There we concentrate on entropy, complexity and rate distortion. Table 3.1 explains the relationships addressed in this thesis. Later, we will describe artifact detection methods (see Chapter 5) and analyze our results (see Chapter 6).

First of all, however, we will present a state of the art survey how to analyze hidden information. For this purpose we begin with multimedia data quality as many image data quality publications are based on multimedia applications.

Basic types of image artifacts	- Destructive artifacts (e.g., sensor effects) - Intentional pseudo-artifacts (e.g., hidden signal-dependent watermarking)
Fields of occurrence	- Instrumental effects - Data compression and image processing - Watermarking - Steganography - Image fakery
Mathematical descriptions	- Image statistics and related models - Information content approaches: Entropy, complexity and rate distortion

Table 3.1: Relationships between various scientific fields addressed in this thesis.

3.1 Multimedia Image Quality

Multimedia images are always subject to a variety of distortions and modifications during the process of compression, transmission, reproduction, etc.

It is important to measure and identify the quality and quality degradation in the data in order to keep the degradation under control and to have a chance to improve the quality of the images.

To evaluate the quality of images, some methods use measures of comparison against a reference. In that sense, we have three approaches (Wang et al. 2005b):

- The "full-reference" (FR) approach
  The full-reference method requires full access to the original image as a reference. It is based on the following philosophy:

  Distorted Signal = Reference Signal + Error Signal

  We assume that the reference signal has a perfect quality, and we quantify the error according to some quantitative metric.

- The "non-reference" (NR) approach
  The non-reference approach does not require any access to the original image, but the quality assessment without reference is a very difficult task. Several researchers have done some work for the evaluation of specific distortions.

- The "reduced-reference" (RR) approach
  The reduced-reference approach does not require full access to the original image but needs some partial information as references such as a set of extracted features.

The related on-going research develops methods and algorithms that can automatically assess the quality of an image. Some years ago, (Wang et al. 2006) already presented a concept for quality-aware images. They use features extracted from the original image; the feature extraction is based on wavelet coefficients. (Sheikh & Bovik 2006b)

proposed how to quantify lost image information and explore a relationship between image information and image quality. The authors of (Rajashekar et al. 2006) investigated whether observers used structural cues to direct their eye fixation as they searched for simple embedded geometric targets at very low signal-to-noise ratios; the authors demonstrated that even in case of very noisy displays, observers do not search randomly, but in many cases they deploy their fixation to stimulus regions that resemble some aspect of the target in their local image features. (Sheikh et al. 2006a) show an evaluation of different recent full reference image quality assessment methods, where they performed a subjective evaluation.

3.1.1 Metrics for Image Quality

In the literature, we can see that many metrics have been developed within the full-reference approach to allow comparisons between undistorted and distorted images. Thus, we obtain a quantitative image quality assessment. Some quality metrics to assess images using the full-reference approach have been evaluated in (Sheikh et al. 2006a), (Avcibas et al. 2002) and (Avcibas 2001).

In the following, we will present the results of these authors and adopt their individual notation (see the 22 methods listed below).

In most cases, C presents the original (i.e. undistorted) or reference image and $\hat{C}$ represents the distorted image (comprising N pixels) of which we measure its quality. Likewise i, j represent the position of each pixel and k corresponds to the spectral band where K is the total number of spectral bands. *MSE* is the Mean Squared Error, L is the maximum dynamic range, i.e., for 8 bits/pixel gray-scale images, $L = 255$. In contrast, in (Sheikh et al. 2006a) μ_x, σ_x and σ_{xy} represent the mean, the standard deviation and the cross-correlation of an undistorted image x and a distorted image y; C_1, C_2 and C_3 are constants.

There are quality measures based on the difference of pixels such as:

- The Minkowski metric is the base for different quality metrics; it is only necessary to select a γ value. It is widely used to determine overall error-rates across different sub-bands or channels (Avcibas et al. 2002) and (Avcibas 2001).

$$M_0 = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{N^2}\sum_{i,j=0}^{N-1}\left|C_k(i,j)-\hat{C}_k(i,j)\right|^{\gamma}\right\}^{1/\gamma}. \tag{3.0}$$

- Mean squared error: (*MSE*) minimizes the mean of the squared residual errors. It assumes that the noise follows a normal distribution. Thus, it is a specific form of the Minkowski metric. This metric is sensitive to the impact of noise; it is also most sensitive to distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001).

$$M_1 = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{N^2}\sum_{i,j=0}^{N-1}\left|C_k(i,j)-\hat{C}_k(i,j)\right|^{2}\right\}^{1/2}. \tag{3.1}$$

- Mean absolute error: (*MAE*) minimizes the mean of the absolute value of residual errors assuming that the noise follows a double exponential distribution.

It is one of the more common metrics that is also most sensitive to distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001).

$$M_2 = \frac{1}{K}\sum_{k=1}^{K}\left\{\frac{1}{N^2}\sum_{i,j=0}^{N-1}\left|C_k(i,j)-\hat{C}_k(i,j)\right|^2\right\} \quad . \tag{3.2}$$

- Modified infinity norm: This metric is the result of a modified Minkowsky metric; it is rather robust since it is based on a ranked list of pixel differences. This measure is most sensitive to distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001). One produces the squared differences between the undistorted and the distorted image, and performs a pixel-wise sorting of the squared differences by magnitude. Then one selects an upper limit *r* (e.g., 10% of the total number of pixels) up to which the following formula is applied.

$$M_3 = \sqrt{\frac{1}{r}\sum_{m=1}^{r}\Delta_m^2(C-\hat{C})} \quad . \tag{3.3}$$

- L*a*b* perceptual error: The objective of this metric is to yield a perceptually uniform spacing of colors that exhibit color differences greater than the Just Noticeable Difference (*JND)* threshold. Obviously, this measure applies to color images only and cannot be generalized to arbitrary multispectral images. Therefore, it has been used only for face images and texture images, and not for satellite images (Avcibas et al. 2002) and (Avcibas 2001).

$$M_4 = \frac{1}{N^2}\sum_{i,j=0}^{N-1}\left[\Delta L^*(i,j)^2 + \Delta a^*(i,j)^2 + \Delta b^*(i,j)^2\right] \tag{3.4}$$

where L, a, b represent the 1976 CIE L*a*b* (CIELab) color space.

- Neighborhood error: A metric to measure image distortion on gray level differences per pixel and/or from local displacements of pixels. This metric penalizes in a graduated way spatial displacements in addition to gray level differences (Avcibas et al. 2002) and (Avcibas 2001). One looks at the minimum differences between a single pixel and a small window *w* around this pixel in the counterpart image, *l* and *m* denote the local pixel positions within *w*.

$$M_5 = \sqrt{\frac{1}{2(N-w)^2}\sum_{i,j=w/2}^{N-w/2}\left(\min_{l,m\in w_{i,j}}\left\{d[C(i,j),\hat{C}(l,m)]\right\}\right)^2+\left(\min_{l,m\in w_{i,j}}\left\{d[\hat{C}(i,j),C(l,m)]\right\}\right)^2} \quad . \tag{3.5}$$

- Multiresolution error: This measure resembles image perception in the human visual system, by assigning larger weights to low resolution phenomena and smaller weights to image details. Such measures are also more realistic in machine vision tasks that often use local information only. This metric is capable of discriminating the coder type of previous image compression (JPEG and SPIHT) (Avcibas et al. 2002) and (Avcibas 2001).

$$M_6 = \frac{1}{K}\sum_{k=1}^{K}\sum_{r=1}^{R} d_r^k$$
$$d_r^k = \frac{1}{2^r}\frac{1}{2^{2r-2}}\sum_{i,j=1}^{2^{r-1}}\left|g_{ij}^k - \hat{g}_{ij}^k\right| \tag{3.6}$$

where r represents the various levels of resolution $r \geq 1$, k stands for the selected band; g is the block average gray level.

- Peak signal to noise ratio: This metric is used specially for quality measurement after lossy image compression. The PSNR commonly is used as an approximation to human perception; different researchers have shown that the PSNR measure is a very good indicator of subjective preference (Avcibas et al. 2002) and (Avcibas 2001).

$$M_7 = 10\log_{10}\left(\frac{L^2}{MSE}\right). \tag{3.7}$$

There are also metrics based on correlation such as:

- Normalized cross correlation: This metric can measure the similarity between two images in terms of the correlation function; hence, it is complementary to the difference-based measures (Avcibas et al. 2002) and (Avcibas 2001).

$$M_8 = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1} C_k(i,j)\hat{C}_k(i,j)}{\sum_{i,j=0}^{N-1} C_k(i,j)^2}. \tag{3.8}$$

- Image fidelity: This metric quantifies the distortion of a processed color image relative to its original version (Avcibas et al. 2002) and (Avcibas 2001).

$$M_9 = 1 - \left(\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}\left[C_k(i,j) - \hat{C}_k(i,j)\right]^2}{\sum_{i,j=0}^{N-1} C_k(i,j)^2}\right). \tag{3.9}$$

- Czekanowski correlation: The Czekanowski coefficient (also called the percentage similarity) measures the similarity between different pixels and patches. The Czekanowski distance is a useful metric to compare vectors with strictly non-negative components, as in the case of images (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{10} = \frac{1}{N^2}\sum_{i,j=0}^{N-1}\left(1-\frac{2\sum_{k=1}^{K}\min(C_k(i,j),\widehat{C}_k(i,j))}{\sum_{k=1}^{K}(C_k(i,j)+\widehat{C}_k(i,j))}\right). \tag{3.10}$$

- Mean angle similarity: This metric uses the moments of the spectral (chromatic) vector differences as distortion measures; this metric uses the mean of the angle differences and the mean of the combined angle-magnitude differences. This measure is most sensitive to distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001). The computation shown below is performed over all spectral bands.

$$M_{11} = 1-\frac{1}{N^2}\sum_{i,j=1}^{N}\frac{2}{\pi}\cos^{-1}\frac{\left\langle C(i,j),\widehat{C}(i,j)\right\rangle}{\left\|C(i,j)\right\|\left\|\widehat{C}(i,j)\right\|}. \tag{3.11}$$

- Mean angle-magnitude similarity: This metric is an extension of the Mean angle similarity**;** this measure is most sensitive to distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001). Again, the computation is performed over all spectral bands.

$$M_{12} = \frac{1}{N^2}\sum_{i,j=1}^{N}\left(1-\left[1-\frac{2}{\pi}\cos^{-1}\frac{\left\langle C(i,j),\widehat{C}(i,j)\right\rangle}{\left\|C(i,j)\right\|\left\|\widehat{C}(i,j)\right\|}\right]\left[1-\frac{\left\|C(i,j)-\widehat{C}(i,j)\right\|}{\sqrt{K\times L^2}}\right]\right). \tag{3.12}$$

In addition, there are quality measurements based on edges as:

- Pratt edge measure: It is a metric introduced by Pratt; it considers both edge location accuracy and missing / false alarm edge elements. This measure is based on the knowledge of an ideal reference edge map, where the reference edges should have preferably a width of one pixel (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{13} = \frac{1}{\max\{n_d, n_t\}}\sum_{i=1}^{n_d}\frac{1}{1+ad_i^2} \tag{3.13}$$

  where n_d and n_t are the number of detected and ground-truth edge points, d represents the distance to the closest detected edge, and a stands for a selectable scaling factor.

- Edge stability measure: This measure is defined as the consistency of edge evidences across different scales in both the undistorted and distorted images. This metric is sensitive to JPEG compression, to Set Partitioning In Hierarchical

Trees (SPIHT) compression, and to blurring effects; it is also sensitive to further distortion artifacts (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{14} = \frac{1}{n_d} \sum_{i,j=0}^{n_d} \left[Q(i,j) - \widehat{Q}(i,j) \right]^2 \tag{3.14}$$

where Q is an edge stability map and the number of edges is given by n_d. A property complementary to edge information could be the surface curvature which is a useful feature for scene analysis, feature extraction and object recognition.

Some metrics based on the spectral information are:

- Spectral phase error: This metric is obtained from the complex Fourier spectrum of images; this spectral distortion measure can be extended to multispectral images by considering the spectral phases and magnitudes. This metric is sensitive to measuring a blur effect (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{15_k} = \frac{1}{N^2} \sum_{u,v=0}^{N-1} \left| \varphi_k(u,v) - \widehat{\varphi}_k(u,v) \right|^2$$

$$\varphi_k(u,v) = \arctan\left[\Gamma_k(u,v) \right] \tag{3.15}$$

$$\Gamma_k(u,v) = \sum_{m,n=0}^{N-1} C_k(m,n) \exp\left[-2\pi i m \frac{u}{N} \right] \exp\left[-2\pi i n \frac{v}{N} \right].$$

- Spectral phase-magnitude error: This measure, derived from the Spectral phase error, can be extended in a straightforward manner to multispectral images. This metric is sensitive to JPEG compression, to SPIHT compression, to blurring effects, and is also sensitive to further distortion artifacts. This measure is the best in discriminating compression distortions (Avcibas et al. 2002) and (Avcibas 2001). It includes a weighting factor λ that balances phase and magnitude effects.

$$M_{16} = \frac{1}{N^2} \left(\lambda \sum_{u,v=0}^{N-1} \left| \varphi(u,v) - \widehat{\varphi}(u,v) \right|^2 + (1-\lambda) \times \sum_{u,v=0}^{N-1} \left| M(u,v) - \widehat{M}(u,v) \right|^2 \right) \tag{3.16}$$

$$M(u,v) = \left| \Gamma(u,v) \right|.$$

- Spectral magnitude error: This measure, also derived from the Spectral phase error, can be extended in a straightforward manner to multiple band images (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{17} = \frac{1}{KN^2} \sum_{k=1}^{K} \sum_{u,y=0}^{N-1} \left\| \left| \Gamma_k(u,v) \right| - \left| \widehat{\Gamma}_k(u,v) \right| \right\|^2 . \tag{3.17}$$

- Block spectral phase error: This metric uses the localized nature of distortion and/or the non-stationary image field (Note that additional Minkowsky averaging of block spectral distortions may be more advantageous) (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{18} = \frac{1}{KN^2} \sum_{k=1}^{K} \sum_{u,y=0}^{N-1} \left\| \left| \varphi_k(u,v) \right| - \left| \widehat{\varphi}_k(u,v) \right| \right\|^2 \tag{3.18}$$

A context-based metric is:

- Spearman rank correlation (*SRC*): It is used to discover the strength of a link between two sets of data. The block *SRC* measure is calculated by computing the rank scores of the gray levels in the given bands and their largest correlation for each pixel neighborhood within a selectable block *u* (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{19} = \sum_{u=1}^{b} \max_{k=1,\ldots,K} \left\{ SRC_u^k \right\}. \tag{3.19}$$

Metrics based on the human visual system (HVS) are:

- HVS absolute norm: The HVS model is an objective measure leading to a better correlation with subjective ratings. This metric is sensitive to JPEG compression, blurring effects, and is also sensitive to further distortion artifacts; this metric is capable of discriminating the compression coder type (JPEG versus SPIHT). This measure is very sensitive in discriminating distortions due to compression. (Avcibas et al. 2002) and (Avcibas 2001). The method includes a band pass filter *H* that models the human visual system.

$$M_{20} = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i,j=0}^{N-1} \left| U\{C_k(i,j)\} - U\{\widehat{C}_k(i,j)\} \right|}{\sum_{i,j=0}^{N-1} \left| U\{C_k(i,j)\} \right|} \tag{3.20}$$

$$U\{C(i,j)\} = DCT^{-1}\left\{ H(\sqrt{u^2+v^2})\Omega(u,v) \right\}$$

where Ω is a 2-D Discrete Cosine Transform (*DCT*), DCT^{-1} is the inverse of the *DCT*.

- HVS L2 norm: This metric is derived from the HVS absolute norm and, in order to obtain a closer relation with the assessment by the human visual system, both the undistorted and the distorted images are preprocessed via filters that simulate the HVS. This metric is sensitive to JPEG compression, to SPIHT compression, and is also sensitive to further distortion artifacts; this metric is also capable of discriminating the compression coder type (JPEG versus SPIHT). This measure is also very sensitive in discriminating distortions due to compression (Avcibas et al. 2002) and (Avcibas 2001).

$$M_{21} = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{N^2}\sum_{i,j=0}^{N-1}\left|U\{C_k(i,j)\} - U\{\widehat{C}_k(i,j)\}\right|^2\right]^{1/2}. \tag{3.21}$$

Another metric is:

- SSIM (Structural Similarity): This metric uses the luminance, the contrast and the structure to calculate the similarity between two images. SSIM considers image degradation as a perceived change in structural information. The idea is to consider the strong inter-dependencies between pixels (Sheikh et al. 2006a). The metric uses a selectable combination function *f*.

$$\begin{aligned} M_{22} &= f(l(x,y), c(x,y), s(s,y)) \\ l(x,y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x,y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x,y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{3.22}$$

Each of the metrics listed above works better or worse in case of specific distortions. One of the best known metrics is the PSNR (peak signal to noise ratio) even if some results may appear to be inconsistent. For instance, if an equal amount of additive noise is added to different sections of an image, we obtain different image quality results based on a visual assessment as shown in Figure 3.1. Here, however, both images have the same PSNR = 35.29.

Figure 3.1: Two images with same PSNR. The same amount of noise has been added to rectangular areas at the top (left) and at the bottom (right) of this image

The visual effect of the distortions depends on the section where the artifacts have been placed. For this reason, we need to evaluate the artifact detection also for different land cover cases in satellite images.

3.1.2 Quality-Aware Images

The analysis of Quality-aware images is a reduced-reference (RR) approach proposed in (Wang et al. 2005b) for assessing image quality. It is based on the extraction of features from an original image and embedding them as hidden information within the same image for a posteriori evaluation. In parallel, we transmit a key from the data source to the final data user. This transmission must be free of any failures. The final user then decodes the embedded nominal features (by means of the perfect key), extracts the actual features from the received image, and checks the similarity of both feature sets. The features are expected to remain after the various processes that an image has to undergo during data distribution, such as compression, decompression, filtering, etc. The system must provide a good trade-off between data hiding, embedding distortion, robustness, and the accuracy of image quality prediction (Wang et al. 2005b). The entire process can be seen in Figure 3.2.

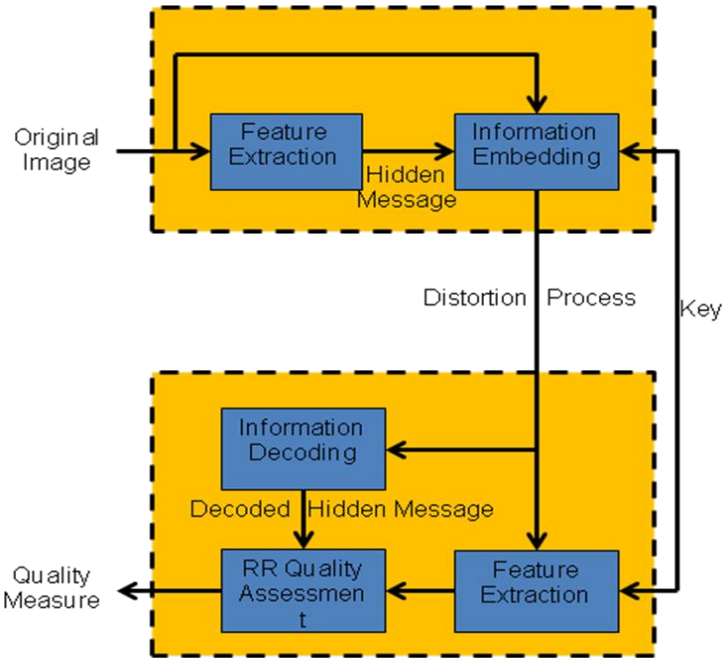


Figure 3.2: Process for Quality-aware images (Wang et al. 2005b)

During the feature extraction process, it is necessary to obtain as much information as possible about the image with the limits of an embedded system, so we have to make a careful selection of the features to consider. These features should give a correct summary of the original image. The feature extraction process begins with the application of a wavelet transform, followed by the selection of important wavelet coefficients; then, we estimate the parameters of the coefficients and, finally, perform a quantization.

The wavelet transform domain provides a framework for the identification of spatially and spectrally representative characteristics. To calculate the transform, the authors used 3 scales and 4 orientations to decompose the image into 12 sub-bands. Parameter extraction is performed for each sub-band; 6 of the 12 sub-bands were selected as characteristics (these 6 sub-bands are not adjacent to prevent overlap). Finally, these selected characteristics were quantized using 8 bits for a mantissa and 3 bits for the exponent. The final result is $(8 + 8 + 8 + 3) \times 6 = 162$ bits for embedding them into the image.

It has to be noted, however, that the embedded system is limited since one cannot embed a big amount of information, and the embedding process should not alter too much the statistical characteristics of the image. To embed the features within the image, we first apply a wavelet transform; then we select some parameters to generate a new measurement based on the extracted features; finally, we apply the inverse transform. The method chosen for the embedding is an existing watermarking technique; it involves quantization, modulation and indexing of the information and allows decoding without having access to the reference image. Five separable scales of the wavelet transform are used to decompose the reference image into 16 sub-bands including horizontal, vertical and diagonal bands for each scale. For embedding additional bits of information into a wavelet coefficient, the coefficient must be modified and quantized. Wang et al. also used error protection techniques to improve the robustness.

At the receiver side, one applies the same wavelet transform and the embedded bits are extracted from the wavelet coefficients by a decoding process. After that, one performs the feature extraction process as being done on the encoder side for comparison with the decoded characteristics. For every difference, a fail signal is sent and finally one counts the fail signals, thus reporting a score of the received image quality; for this final step, the authors use the Kullback-Leibler distance (*KLD*).

3.2 Watermarking

Watermarking is a technique whose main purpose is to highlight the illegal use of a digital service by an unauthorized user. Specifically, this technique involves inserting a message (hidden or not) within a digital object, as could be images, audio, video, text, software, etc. This message is formed by a group of bits containing information about the author or copyright owner of the digital object.

Watermarking for images has become an area of increased research activity over the last decades. Digital image watermarking includes three categories from the application point of view: robust watermarking, fragile watermarking, and semi-fragile watermarking (Song et al. 2010). Robust watermarking can ensure the copyright protection of images. Fragile watermarking is applied for the authentication of image content. Semi-fragile watermarking has the characteristics of both robust and fragile watermarking.

There are several watermarking techniques that may create visible or invisible marks. A visible watermark is intended to be perceptible to the user and typically contains a visual message or company logo indicating the ownership of the image; visible watermarks are especially useful for covering an immediate claim of ownership. An invisible watermark is intended to be imperceptible but is detected and extracted by an appropriate piece of software when the need arises; an image containing an invisible watermark should look similar to the original unmarked image; it is required that the watermarked image should suffer no perceptible quality degradation from the original (Samuel & Penshom 2004).

To insert a visible mark we can proceed as follows: denote the original image *f*, the mark as *w*, and the marked image as f_w, and finally apply the following process:

$$f_w = (1-\alpha) f + \alpha w ,$$

where α is a constant of mark visibility.

If we want to introduce an invisible mark, this will not be visually distinguishable, but it will be possible to detect or recover it using codes and algorithms designed for this purpose. The invisibility is ensured by the inclusion of redundant information.

For example, we can insert the mark in the last 2 least significant bits of the image according to:

$$f_w = 4(\frac{f}{4}) + \frac{w}{64} .$$

In Figure 3.3, we can see some examples of visible watermarking, either in the entire image or only in one corner. Figure 3.3 (a) shows the original image, in Figure 3.3 (b) we can see the watermark, Figure 33 (c) shows an example of watermarking in the complete

image and in Figure 3.3 (d) we can see a watermarking in some part of the image. The intensity of visibility may also vary depending on the objectives and applications.

(a) Original image

(b) Watermark

(c) Watermarking example 1

(d) Watermarking example 2

Figure 3.3: Some examples of visible watermarking. Figure (a) shows the original image, in Figure (b) we can see the watermark, Figure (c) shows an example of watermarking in the complete image and in Figure (d) we can see a watermarking in some part of the image.

In the literature, we can find many methods to put a watermark into an image; for example, (Liu & Ying 2012) present a technique for embedding a watermark using a wavelet decomposition. Another watermarking method based on wavelets is presented in (Song et al. 2010) where the authors show a novel semi-fragile watermarking technique. In (Samuel & Penzhom 2004) we can find another method based on DCT coefficients.

A general watermarking scheme is shown in Figure 3.4.

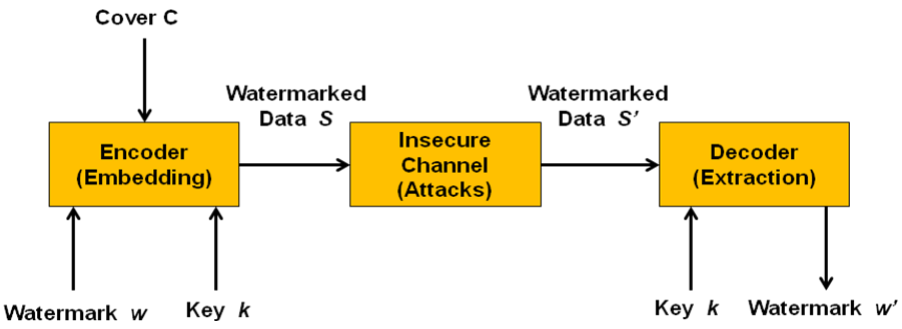


Figure 3.4: A general watermarking scheme that includes three important blocks: an encoder block, a channel block, and a decoder block.

In this scheme, we can observe the presence of a cover signal *C* which is the original signal where to introduce the watermark *w* using a key *k* for coding. As a result of this process, the encoder block produces a signal *S* that contains the watermark; this signal undergoes the different distortion processes. The signal reaching the receiver called *S'* is not identical to the signal *S* due to the distortion processes or attacks carried out in the channel block. At the receiver, we have the known key *k* to decode and interpret the watermark and we can identify the copyright of the signal. We obtain the watermark *w'* but it may not be identical to the original one due to the distortion processes.

3.2.1 Watermarking Detection

Classical watermarking detection methods use the original image or the key to detect the ownership of an image as shown in Figure 3.4; in order to make counterfeit attacks infeasible, one needs a careful selection of some additional requirements or standardization steps for watermarking schemes.

(Zeng & Liu 1999) propose a method where the watermark is detected without using the original image. The authors argue that if the watermark is perceptually invisible, it should be statistically visible; it should be detectable using some statistical techniques. In Figure 3.5, we can see the general building blocks of the watermarking system used in (Zeng & Liu 1999); the authors focus on the so-called feature-based watermarking schemes in which a sequence is embedded into a set of features derived from the original image.

The watermarked image can be constructed based on the modified feature set and other unmodified data. On the side of the watermark detector, the test features are first derived from the test image, and then correlated with a signature (sequence); the correlation should be high but may not be perfect. The correlator output will be compared to a threshold to determine if the test image contains the claimed watermarks. Within Figure 3.5 we can see an expectation operator denoted by *E*, a hypothesis test *H*, and an optimization step *Opt*.

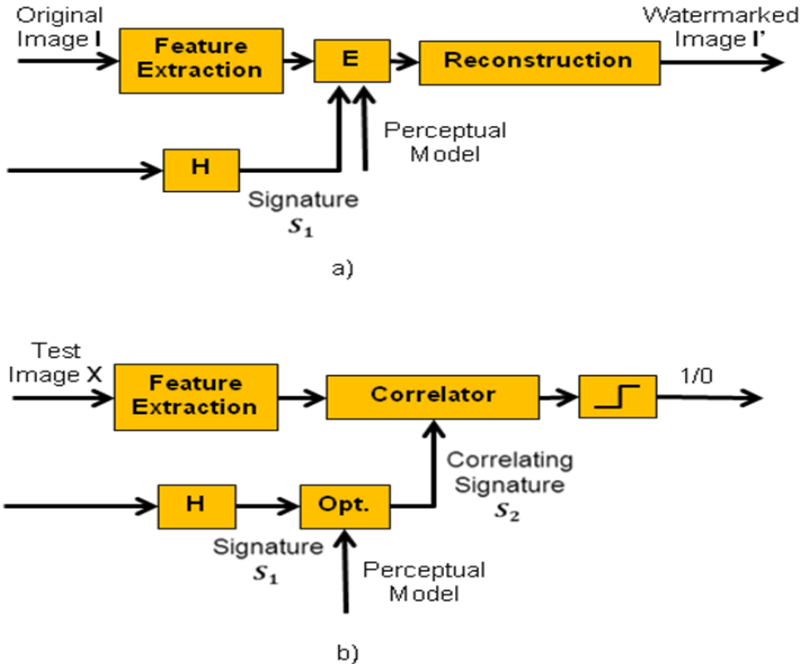


Figure 3.5: Block diagram of the watermarking system proposed by (Zeng & Liu 1999), Figure (a) shows the watermark embedding system, and Figure (b) shows the watermark detection system.

(Jha et al. 2010) present another method to improve the watermark detection; this method uses supra-threshold stochastic resonance. This analysis is based on wavelet transform coefficients. The idea is to improve the correlation between input and output.

3.3 Hidden Information (Steganography)

This subject has a relation with image quality and watermarking because in all these concepts, there is a change of the original image. Steganography involves information hiding techniques; steganography aims to embed a message within an innocuous-looking cover data (text, audio, image, video, etc.) so that a casual inspection of the resulting data will not reveal the presence of the message. In the literature, there are many publications related to information hiding techniques and the detection of hidden information for multimedia data. The detection of hidden information is known as steganalysis; the difficult task of steganalysis can be greatly aided by exploiting the correlation inherent in typical host or cover images. (Avcibas et al. 2003) present techniques for steganalysis based on the hypothesis that the steganographic schemes leave statistical evidence. (Moulin & O'Sullivan 2003) show an information-theoretic analysis about hidden information. (Cachin 2004) proposes an information-theoretic model for steganography with a passive adversary. (Lyu &

Farid 2006) describe a universal approach to steganalysis for detecting the presence of a hidden message based on multiscale and multi-orientation image decomposition. (Sullivan et al. 2006) investigate the detection-theoretic performance benchmarks for steganalysis when the cover data are modeled as a Markov chain.

The problem of hidden information means the following: we have a message *M* that can be embedded into data *S* and gives as a result *X*; this *X* can be subjected to various processes and attacks. This process is shown in Figure 3.6. There exists a key *K* that will help us in the coding and embedding process within the transmitter and during decoding at the receiver side. After the decoding process, we will not always get the original message *M* because it could be affected by different distortions in the transmission channel.

An information hiding system must meet two requirements: *X* has to be very similar to *S*, and the hidden message *M* must survive different processes (compression, resizing, etc.).

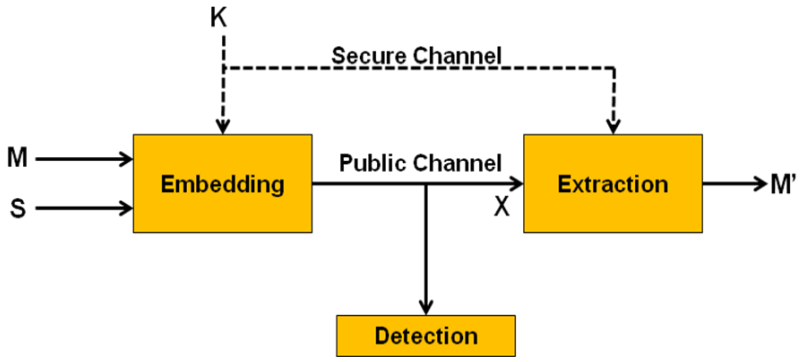


Figure 3.6: General scheme for hidden information. The key *K* will help us in the coding and embedding process within the transmitter and the decoding on the receiver side (Cachin 2004).

If we want to insert hidden information into an image, it must be remembered that an image is an array of numbers where each number is a brightness code, for instance:

- RGB 24 bits:
  0..255 0..255 0..255

- Gray scale 8 bits:
  0..255

A well-known technique to insert messages into an image is to use the least significant bit (LSB) as a carrier of information as shown in the following example:
Original 3×3 pixels in binary notation:

(11101101 00100100 10100001)
(00001111 00101101 11101111)
(00001111 00100111 10000111)

Message to insert: 'A' (10010111)

New pixels

(11101101 00100100 10100000)
(00001111 00101100 11101111)
(00001111 00100111 10000111)

3.3.1 Steganalysis Using Image Quality Metrics

Steganalysis is the process which analyzes an image and determines whether or not it contains hidden information. The method proposed in (Avcibas et al. 2003) is a steganalysis technique based on a comparison between an image and the result of applying a Gaussian filter to the same image; this comparison is made using different image quality metrics. The idea is to discriminate the distance metric for images that contain hidden information from those who do not. The Gaussian filtering was selected because it gave uniformly good results across all steganographic techniques (Avcibas et al. 2003).

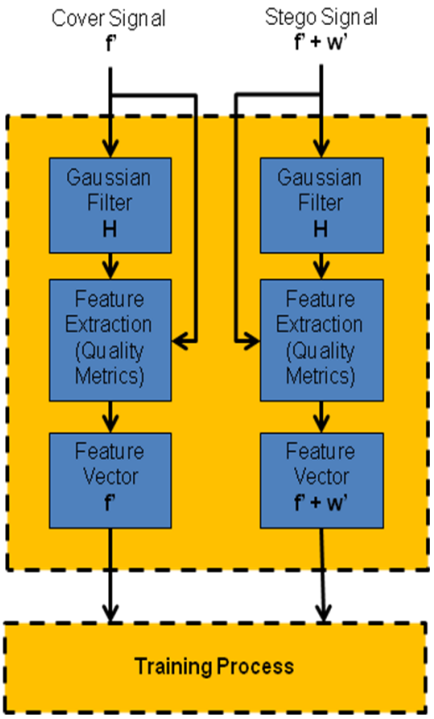


Figure 3.7: Schematic description for the steganalysis technique based on comparison between an image and the result of applying a Gaussian filter to the same image; this comparison is made using different image quality metrics.

In order to explain the analysis process, we first define the nomenclature. The *Cover Signal* is the image that does not contain hidden information, while the *Stego Signal* is the image that contains hidden information. Figure 3.7 outlines this method.

The feature extraction process is performed by comparing the original image and the image after Gaussian filtering; this comparison is made using image quality metrics; all of

the results define our feature vectors. Finally, with these feature vectors for both images (with and without hidden information), we do training for identification. The filter was chosen as a Gaussian smoothing filter; the width of the Gaussian filter was set experimentally to $\sigma = 0.5$ with a mask size of 3×3 pixels.

The metrics that were used by the Avcibas team are: Mean Squared Error (M_1 Equation 3.1), Mean Absolute Error (M_2 Equation 3.2), Normalized Cross-Correlation (M_8 Equation 3.8), Image Fidelity (M_9 Equation 3.9), Czekanowski Distance (M_{10} Equation 3.10), Angular Correlation (M_{11} Equation 3.11), and Spectral Magnitude Distortion (M_{17} Equation 3.17).

3.4 Image Fakery

The development of information technologies has enabled the integration of many electronic devices including cameras, which initially worked with rolls of film; it is now usual to share information taken by a digital camera, i.e., photos, copy them to a computer disk, email them or upload them in the preferred social network.

There are software applications that allow us to alter photographs, often without the user needing to be skilled; there is software for all users. It is possible to use simple applications such as Microsoft Windows Paint, or more complex ones such as Adobe Photoshop, Corel Draw or GIMP, if you prefer a GNU solution to change the essence or form of photography; the changes can be for good or bad reasons.

Figure 3.8: Example of image fakery where we can see a person having been removed from the picture; this fakery was made using a Photoshop tool (Image example courtesy of Camilo Reynaga).

Sometimes the image changes cannot be detected with the naked eye, resulting in most cases in a complicated process since it is not easy to determine the type of alteration that has been realized. An image can also be altered by removing a part of it, or by superimposing other parts making it appear that there are less people or that the landscape is different. Whatever method of altering was used, the resulting image becomes different from the original.

The art of faking images has a long history and today, in the digital era, making some changes to the data can be very easy without leaving a trace of manipulation. Figure 3.8 shows an example of image fakery.

3.4.1 Image Fakery Detection

Currently, there exist several research papers which have developed methods for automatic detection of manipulated images; these methods can be classified as pixel-based, format-based, camera-based, and physics/geometric-based (Cozzolino et al. 2012). The pixel-based methods analyze the correlation between pixels in the spatial domain or in a transformed domain. Format-based techniques exploit usual lossy compressors which introduce specific and recognizable marks. Camera-based approaches take advantage of specific features of a camera model as image signature. Physics/geometric-based techniques study higher-level inconsistencies between the imaged scene/objects and the assumed forgery source; however, some of these methods are specific, i.e. these methods are oriented towards the detection of a specific manipulation case. For example:

- Copy-duplicated image regions
- Interpolation and resampling effects
- Color inconsistencies
- Noise inconsistencies
- Lighting inconsistencies

The detection of lighting inconsistencies is very important and can prove the manipulation of images.

There are many research papers dedicated to the detection of manipulations in an image, as well as research related to the detection of watermarks and steganography; but as mentioned above, these works have the peculiarity of being designed to identify a specific type of manipulation or alteration; we can mention some works like (Mahdian & Saic 2008b) where the authors present a method for detecting inconsistently resampled parts as noise in altered images; the same authors in (Mahdian & Saic 2008a) and (Mahdian & Siac 2010) also have other methods for detecting duplicated regions; the article of (Farid 2009) presents a method for detecting inconsistencies in the illumination based on the reflection of light in the eyes of the protagonists; in (Li et al. 2009) the authors' idea is to extract horizontal and vertical edges due to JPEG artifacts; the work presented in (Popescu & Farid 2005) shows a method based on the observation that tampering may alter the underlying statistics; (Chen et al. 2008) and (Fridrich 2009) use the photo-response non uniformity (PRNU) as intrinsic fingerprints of an individual digital camera for tampering detection. However, each of these methods is based on some hypothesis which limits its applicability and, therefore, it is always possible to find cases where it fails (Cozzolino et al. 2012).

Since existing methods are specific only to a typical fakery, the use of these methods does not ensure a correct detection of a manipulated image when we do not know the type of alteration. For that reason, it is necessary to implement a more general method for detecting altered images regardless of the type of manipulation.

3.5 Conclusion

All these approaches to embedding information may converge towards artifact detection since the presence of watermarks, the presence of hidden information, or abnormalities in regions of the image produce changes in the original image; these changes are not visible but result in changes in image statistics or other parameters.

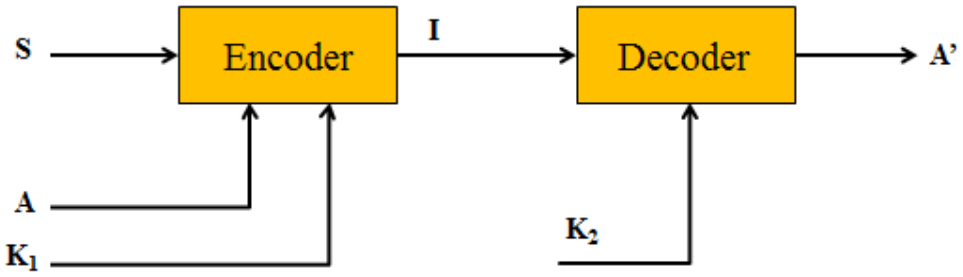


Figure 3.9: General scheme for image analysis and artifact detection. *S* is an artifact free image product; *A* is an artifact; *I* is the image with artifacts introduced; *A'* is the estimated artifact; K_1 and K_2 are the keys for introducing and detecting the artifacts.

Most of the methods within these areas use mutual information for the analysis of the quality and authenticity of an image. The principal reasons why we prefer mutual information as a comparison tool will be described in Chapter 4. In that sense, the general scheme is given by Figure 3.9 where for a system of image quality assessment, *S* is the reference (undistorted) image, *A* is the distortion and the keys K_1 and K_2 are known. In a process of watermark insertion, *A* would be the mark to be added and the keys K_1 and K_2 are also known for decoding and identifying the copyright. For a steganography process, *A* would be the hidden message to be sent and the keys K_1 and K_2 are known only to the transmitter and the direct receiver for decoding the message. In a steganalysis process, the keys K_1 and K_2 are unknown but, nevertheless, one could try to reach a good approximation for estimating the presence of hidden information or of a hidden message. Similarly, in the process for artifact detection, *A* is the artifact inserted in the image product *S* by a pre-processing process using the key K_1 producing image *I*; the keys K_1 and K_2 are unknown but in the artifact detection process there is the intention to estimate them to detect the presence of artifacts *A'*.

For this reason, we can combine these methods, evaluate them, and implement a parameter-free method for artifact detection.

Chapter 4

Image Information, Entropy and Complexity

As we saw in the previous chapter, the different fields of image quality, watermarking, hidden information, image fakery, etc. are all interrelated and each one refers to image information and image statistics; that is why we will provide some background description of information theory. In this chapter, we present the basic concepts of information theory, both from a probabilistic Shannon theory approach, and an algorithmic approach with Kolmogorov complexity. We also will try to explain the relationship between the two approaches and how this will help us to implement different methods for artifact detection in satellite images.

First, however, we explain why we consider the principles of information theory formulated by Shannon as a fundamental tool that we need for artifact detection. The reason is that we need quantitative information when we compare data sets: a data set affected by artifacts needs to be distinguishable from a second version of this data set that is artifact-free, or even from a third version that is affected by another type of artifact. A comparison among data sets has to include a quantitative dimension that allows us to search for artifacts and their strength.

This requirement leads us directly to information content as defined by Shannon and the use of entropy based on the probabilities of pixel brightness levels. Therefore, we suggest the use of mutual information as a primary tool when we compare data sets with and without artifacts. We assume that differences in the probability distributions are most likely a convincing clue to the existence of artifacts: even slight discrepancies that remain hidden during a visual inspection of images will become apparent in a numerical analysis of their probability distributions. This holds for rather regular artifacts as well as for irregular artifacts and becomes apparent when we use the information content of images for coding. Regular artifacts lead to advantages in coding, while irregular artifacts result in longer codes. This can also be seen in the entropy formulas devised by Shannon (see Section 4.1.3).

The principles of Shannon are based on probability distributions and do not explicitly address the complexity of an image that tells us whether we have an ordered or an unordered data set. Therefore, we resort to Kolmogorov complexity that tells us about both options. Unordered data sets have a higher complexity, while ordered data sets result in a lower complexity. Both cases are closely linked to artifacts. In most cases, artifacts increase the complexity of images. In order to quantify image complexity we could think of

Kolmogorov complexity that has been discussed in depth in literature; however, in practice, one cannot compute the Kolmogorov complexity directly and we have to resort to approximating it by the compression behavior of images and to compare the compression distances of images (see Section 4.4).

By using the compression behavior of images we obtain a quantitative tool to search for artifacts. We can either stay in Shannon's realm by applying a *rate distortion* approach, or we move to the domain of Kolmogorov and proceed with *complexity distortion*. It has to be noted that the compression/distortion approach as described in this dissertation represents a new concept for artifact detection. To our knowledge, there are no previous publications describing such a concept.

This chapter contains the necessary explanations how to understand distortion as a fundamental criterion, while practical compression-based methods for artifact detection will be presented in Chapter 5, being followed by comparative results contained in Chapter 6.

4.1 Shannon Information Theory

Information theory is a branch of mathematical theory, probability and statistics theory that allows us to study information and its storage in digital form. It was developed by Claude E. Shannon in (Shannon 1948) to find the fundamental limits of compression and the capacity bounds of data communication.

4.1.1 Principles for Information Measurement

Following probabilistic considerations, one can establish a first principle of information measurement. This principle establishes that a message which is more probable has less information than a message with lower probability. This can be expressed as follows:

$$I(x_i) > I(x_k) \Leftrightarrow p(x_i) < p(x_k), \tag{4.1}$$

where x_i is a data element, x_k is another data element, $I(x_i)$ is the amount of information provided by x_i, $p(x_i)$ is the probability of x_i, and i and k are indices.

According to this principle, it is the probability of a message to be sent and not its content, which determines its informational value. The content is important only as it has a probability. Please note that the amount of information that a message provides varies from one context to another because the probability of sending a message varies from one context to another.

A second principle states that if two messages X and Y are selected, the amount of information provided by both messages will be equal to the amount of information provided by X plus the amount of information provided by Y, as X has already been selected. This can be expressed as:

$$I(X;Y) = \sum_{x_i \in X} \sum_{y_i \in Y} p(x_i, y_i) \log \frac{p(x_i, y_i)}{p(x_i)\, p(y_i)}, \tag{4.2}$$

where X and Y are two messages, $I(X;Y)$ is the mutual information between X and Y, x_i is an element of the message X, y_i is an element of the message Y, $p(x_i)$ is the probability of x_i, $p(y_i)$ is the probability of y_i, and $p(x_i, y_i)$ is the joint probability of x_i and y_i.

4.1.2 Information Content Measure

If there is a message element x_i with a probability of occurrence $p(x_i)$, the information content $I(x_i)$ can be expressed as:

$$I(x_i) = \log_2 {}^{1}\!\!\diagup\!\!{}_{p(x_i)}. \tag{4.3}$$

Numerically, the information I is measured in bits. For a binary coding of a message element, the number of bits is equal to the logarithm of the number of choices made in base 2. For example, if a message specifies one of sixteen equally possible choices, one needs four bits of information ($2^4 = 16$).

4.1.3 Shannon Entropy

The Shannon entropy is a measure of uncertainty of a random variable X with a given probability distribution $p(x) = P(X = x)$.

$$H(X) = \sum_{x_i \in X} p(x_i) I(x_i) \tag{4.4}$$

$$H(X) = -\sum_{x_i \in X} p(x_i) \log_2 p(x_i) \tag{4.5}$$

where $H(X)$ is the entropy of X; X is a random variable, $I(x_i)$ is the information content of x_i, x_i is an element of the data X, $p(x_i)$ is the probability of x and i is an index.

This definition also can express the average length in bits needed to encode X; for example, independent fair coin flips have an entropy of 1 bit per flip; in general, a uniform distribution A has an entropy $H(A) = \log_2 v$, where v is the number of possible outcomes of A. We consider a as a string output by A in order to facilitate comparisons with algorithmic complexity. For a random (uniform) distribution of a, the entropy of A increases with the size of its alphabet. This implies that the uncertainty of each symbol in a grows, and so does its information content. On the contrary, a maximally redundant source B, for example, one that always generates a string b composed of a long sequence of 1's, independently from the number of its possible outcomes, has an entropy $H(B) = 0$, and every isolated symbol in b carries no information.

4.2 Kolmogorov Complexity

The Kolmogorov Complexity $K(x)$ of an object x is the minimum amount of computational resources needed to represent x. The Kolmogorov Complexity is defined by:

$$K(x) = \min_{q \in Q_x} |q| \quad , \tag{4.6}$$

where q is an element of the set Q_x; Q_x is the set of instantaneous codes that give x as output.

There is a dependency on the size of q due to the applied descriptive language, but this is of no concern as one can reduce the description difference to some constant. For instance, given two languages L_1 and L_2, and a string of symbols x, the complexity differences $|K_1(x)-K_2(x)|$ are less than k. In order to move from a description in L_1 to another one in L_2, we need a translator program for L_1, which converts L_1 to L_2. The translator program may be of any size, but its size is fixed; hence, it is a constant.

Within Information Theory, we can say that the Kolmogorov complexity (or algorithmic complexity) is the amount of information needed to recover x. It is important to note that $K(x)$ is a non-calculable function. The conditional complexity $K(x|y)$ of x given y is defined as the length of the shortest program with which we can obtain as output x from y.
It is easy to see that universal compression algorithms (when being used as an approximation to compute Kolmogorov complexity) give an upper bound on Kolmogorov complexity. The compressibility is a term derived from having a small program q that describes a string x. Therefore, if $K(x) < |x|$ we say that x is compressible. In fact, $K(x)$ is the best compression that one can achieve for an arbitrary string x. Given a data compression algorithm, we define $C(x)$ as the size of the compressed version of x and $C(x|y)$ as the compression achieved for x with a given y. For example, if the compressor is based on a textual substitution method, we could construct a dictionary of y and use the dictionary for compressing x.

An important thing to talk about is the complexity of complexity. If we have a program q which can represent x, and this program q is the shortest one of all programs that can represent x, then the length l of the program q is the complexity of x, thus $K(x) = l(q)$. If we wish to obtain the complexity of this program q, we cannot obtain a smaller representation, as its length would be greater than or practically equal to the complexity of x, $K(q) \geq K(x) = l(q)$ because if we got a shorter length, this would be the true complexity of x.

4.3 Relationship Between Shannon Entropy and Kolmogorov Complexity

Another aspect is to determine the relationship between Kolmogorov complexity and Shannon entropy. Shannon's entropy value indicates the average number of bits to describe a variable x, while the Kolmogorov complexity says that an object x has a complexity or information content equal to the minimum length of a binary program for x.

A relationship between Shannon entropy and Kolmogorov complexity is based on a theorem presented in (Gruenwald & Vitanyi 2008) which says that the sum of the expected

Kolmogorov complexities of all code words *x* which are the output of a random source *X* weighted by their probabilities *p(x)*, is equal to the statistical Shannon entropy *H(X)* of *X*:

$$H(X) \le \sum_x p(x)K(x) \le H(X) + K(p) + O(1)H(X) = \sum_x p(x)K(x|p) + O(1), \quad (4.7)$$

where *K(p)* is the complexity of the probability function *p(X)*, *K(x|p)* is the complexity of *x* knowing *p(X)* and *O(1)* defines the computational order complexity.

The algorithmic mutual information between two strings *x* and *y* is presented in (Gruenwald & Vitanyi 2008) and is given by

$$I_w(x:y) = K(x) + K(y) - K(x,y). \quad (4.8)$$

I_w is independent of any probability distribution.

The use of compressors as an approximation to the Kolmogorov complexity can help us to better understand the relationship between Shannon entropy and Kolmogorov complexity described in the text above. Actually, usual compressors use the principle of Shannon entropy coding to perform compression, and when we use the compression approximation to Kolmogorov complexity, we are using Shannon's theory implicitly linking it with complexity. We use compression as an approximation; when looking for a short program capable of representing an object *x*, what we do is also trying to compress the object *x* in a more compact representation. We could even use alternative compressors that are not based on entropy coding.

4.4 Normalized Compression Distance

An application of Kolmogorov Complexity is to estimate the shared information between two objects given by their Normalized Information Distance (*NID*) (Li et al. 2004). The *NID* is proportional to the length of the shortest program that can calculate *x* given *y*. The normalized information distance is calculated as follows:

$$NID(x,y) = \frac{K(x,y) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}}, \quad (4.9)$$

where *K(x)* is the Kolmogorov complexity of *x*, *K(y)* is the Kolmogorov complexity of *y*, *x* and *y* are two data sets to be compared, and *K(x,y)* is the joint Kolmogorov complexity of *x* and *y*. The *NID* result is a positive value *r* in the range of $0 \le r \le 1$, with $r = 0$ if the objects are identical, and $r = 1$ stands for the maximum distance between them. However, the *NID* is not computable and therefore we need a computable approximation. A well-known approach is the Normalized Compression Distance *NCD* defined by (Li et al. 2004) and by (Cilibrasi & Vitanyi 2005) considering *K(x)* as the compressed version of *x*, and taking it as a lower limit of what can be achieved with the compressor *C*.

One can approximate *K(x)* with $C(x) = K(x) + k$, where *C(x)* is the length of the compressed version of *x* obtained by a lossless compressor *C* that is equal to the Kolmogorov

Complexity *K(x)* plus an unknown constant *k*. The presence of *k* is necessary because one cannot estimate how closely we approximate *K(x)*. As an example, we take two strings *b* and *p* that have the same length *n*, where the former is the output of a random process and the latter represents the first *n* digits of π. The quantity *K(p)* would be smaller than *K(b)* because there is a program of length *K(p)* << *n* whose output is the number π, while a program outputting a random sequence of bits would have a length close to *n*, so *K(p)* << *K(b)*. Thus, the Normalized Compression Distance (*NCD*) can be defined as shown in the following equation:

$$NCD(x,y) = \frac{C(x,y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}, \tag{4.10}$$

where *C(x, y)* represents the size of compressed file obtained by the concatenation of *x* and *y*. We use this equation to estimate the *NID*. See Chapter 7 for a description of concatenation options.

The *NCD* can be calculated easily between two strings or two files *x* and *y*, and it shows how different these files are. We can use the *NCD* for various applications with different classes of data as a parameter-free approach (Cerra et al. 2010; Keogh et al. 2004). The NCD can also be used to classify the data by unsupervised methods (Cilibrasi & Vitanyi 2005). The *NCD* returns a positive result $0 \leq NCD \leq 1+e$, with *e* as a representation of potential inefficiencies of the compression algorithms. Please note that the *K(x)* approximation via *C(x)* depends on the data with which to work; we know that common compressors are based on different hypotheses; some are more efficient than others with specific data.

4.5 Rate-Distortion Theory

When we want to know the number of bits needed to encode the different representations of a variable *X* allowing that there may be some loss of information, we can use Rate-Distortion theory. Within this context, a Rate-Distortion (*RD*) function gives the minimum value of mutual information between source and receiver under pre-defined distortion limits.

$$R(D^*) = \min_{D \leq D^*} I(U;V) \tag{4.11}$$

where *U* represents the undistorted data on the source side, *V* stands the distorted data on the receiver side, *I(U;V)* is the Mutual Information between *U* and *V*, *D* is the distortion between *U* and *V* measured by a quality metric, and D^* is the distortion limit.

In principle, the *RD* function shows the compression error resulting from different compression ratios. The *RD* function is a convex and decreasing function (Cover & Thomas 1991): the more we compress, the more information we lose.

In the following, we will combine the RD concept given here with complexity ideas of Kolmogorov given in Section 4.2.

4.6 Complexity-Distortion Function

In Shannon's Rate-Distortion theory, the rate is the minimum number of bits that are needed to encode a distorted message, so the *RD* function determines the minimum mutual information under a distortion constraint allowed. Figure 4.1 explains the process of encoding and decoding of a message. Here we can see a message passing through a coded channel to reach a receiver. The receiver decodes the message and the retrieved information may differ from the original data due to distortions created anywhere within the system.

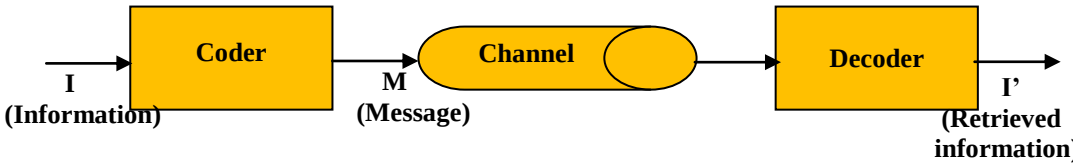


Figure 4.1: General communication diagram: encoding, transfer and decoding of a message.

If we want to introduce complexity into this context, we could exchange the encoder and decoder blocks for a Universal Turing Machine (*UTM*); for each selectable length (that causes loss of information), the coding *UTM* will produce a shortest program (according to Kolmogorov's theory).

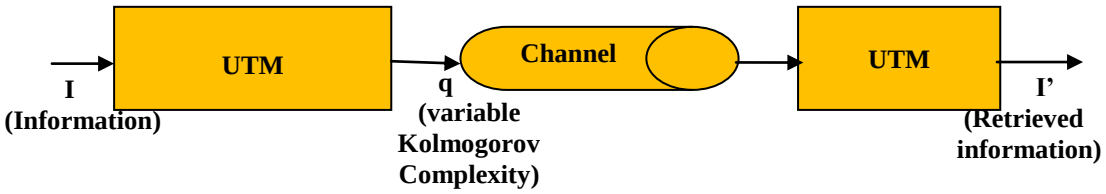


Figure 4.2: Block diagram for complexity distortion.

Figure 4.2 explains the new system components. The system would allow us to calculate a complexity-distortion function. We can see the *UTM* components. A practical implementation would need a lossless and a lossy compressor. The former is needed to approximate the complexity of the data, while the latter is required to generate distorted data. Instead, we can also use dictionaries (Reynar et al. 1999).

4.7 Kolmogorov Structure Function

As an additional alternative to the distortion functions described above, one can also use structure functions. In our case, we selected the Kolmogorov Structure Function (*KSF*) that defines the relationships between the data and a model based on the relationships between original data (following an unknown model distribution) and potential alternative

representations. In (Vereshchagin & Vitanyi 2002) the authors present a definition and analysis of a Kolmogorov Structure Function. The relation between given data and its model is expressed by the *KSF*.

The original Kolmogorov structure function for given data *x* is defined by:

$$h_x(\alpha) = \min_S \{\log_2 |S| : S \ni x, K(S) \leq \alpha\}, \tag{4.12}$$

where *S* is a set of representations for *x* and α is a non-negative integer value bounding the complexity of *S*. The Kolmogorov structure function $h_x(\alpha)$ tells us about all stochastic properties of the data *x* (Vereshchagin & Vitanyi 2002); however, it does not tell us how to generate a set of representations.

The Kolmogorov structure function is non-computable since it is based on the Kolmogorov complexity that is a non-computable function (that is the reason why we use the size of the compressed data as an approximation to complexity).

The idea is to use the Kolmogorov structure function as an approximation to a rate-distortion (*RD*) analysis, where we consider $h_x(\alpha)$ being equivalent to *D*, while *K(S)* corresponds to *R*. We determine $h_x(\alpha)$ via Eq. 4.12 by using the output of lossy compression with different compression parameters as different representations and we use the size of the compressed data as *K(S)* for describing the behavior of artifacts and for developing a parameter-free artifact detection method.

4.8 Data Compression

The objective of data compression is to reduce the volume of manageable information (e.g., data to be processed, transmitted, or recorded). In principle, compression means to transport the same information, but using a lower amount of data volume.

The volume occupied by coded information without compression is the product of the number of (tightly packed) samples times the number of bits per sample. Therefore, the more bits are used the file size becomes bigger. However, the number of bits per sample is given by the data acquisition system and cannot be altered; hence, compression is used to transmit the same amount of information with less bits.

Historically, simple data compression is based on a search for data redundancies and storage of data with the number of times to repeat. For example, if a sequence in a file reads "AAAAAA", instead of occupying 6 bytes, one could simply store "6A" which occupies only 2 bytes. In practice, however, the process is somewhat more complex, and does not find all repeating patterns.

Currently, two basic types of data compression are available. One of them is lossless compression that is normally used to communicate or to store text or binary files without information losses (Raja & Saraswathi 2011); however, the attainable compression of the latter type is rather low. The reasons for lossless compression capabilities are redundancies contained in the data and coding techniques based on statistical occurrences.

The other compression type is lossy compression, which is commonly used for higher-rate compression to compress image, video or audio data files for transmission or storage purposes and leads to information losses.

In the following, three compression techniques are presented in more detail.

4.8.1 JPEG Compression

JPEG is a standardized image compression method that stands for Joint Photographic Experts Group, the original name of the committee that wrote the standard. Currently, there are three JPEG versions being available:

- The baseline JPEG standard of 1992 that has been developed for the compression of single images based on discrete cosine transforms; in 1993 a lossless option has been added.
- A JPEG-LS published in 1999 for lossless image compression using a LOCO-I algorithm.
- The JPEG 2000 version based on wavelet transforms that also supports the compression of image sequences. JPEG 2000 supports lossless as well as lossy data compression.

JPEG has been designed for compressing full-color or gray-scale digital images. In the following, we take a more detailed look into the three JPEG versions listed above

- The (lossless) JPEG-LS uses a predictive coding model called Differential Pulse Code Modulation (*DPCM)*. *DPCM* encodes the differences between the predicted and actual samples instead of encoding each sample independently. The differences from one sample to the next are usually close to zero which, in general, leads to better compression.
  The prediction process of JPEG-LS combines three neighboring samples at *W* (west), *N* (north) and *NW* (north-west), in order to produce a prediction of the actual sample value. The three neighboring samples must be already predicted samples. JPEG-LS contains 8 options of linear combinations of *W*, *N* and *NW* that can be used as predictors to estimate the actual sample. After prediction, an entropy coding based on a Huffman code (Huffman 1952) is performed.

- On the contrary, the JPEG 1992 (“baseline”) standard is a lossy image compression method. In a first step, a digital image is split into a sequence of blocks of 8×8 pixels. Each block is then compressed by the following sequence of transformations:

  1. Integer Discrete Cosine Transformation (*DCT*) of the 64 pixels in each block;
  2. Quantization of the *DCT* coefficients thereby producing a set of 64 smaller integers. This step causes a loss of information but makes the data more compressible.
  3. Entropy coding of the quantized *DCT* coefficients. Baseline JPEG uses Huffman coding in this step, but the JPEG standard also allows arithmetic coding as a possible alternative.

  The decompression process just reverses the actions. The first step is to apply Huffman decoding, after that, de-quantization of the coefficients follows, and finally, an inverse *DCT* is applied to obtain a set of values. Because of the quantization step, the reconstructed set includes only approximated values. The coefficient in position

(0,0) (left upper corner of an 8×8 block) is called *DC* (mean value) coefficient and the 63 remaining values are called *AC* coefficients.
Baseline JPEG uses two different static Huffman trees to encode the data. The first one encodes the lengths in bits (1 to 11) of the value in each *DC* field. The second tree encodes information about the sequence of *AC* coefficients.

- Finally, JPEG2000 compression is based on wavelet transformation. The process flow of JPEG2000 compression is as follows: The first step, in case of color images, is to transform the color components from RGB to YCbCr components. After that, and after tiling the image into patches, the compressor applies a wavelet transformation to each patch. After the transformation, the next step is quantization; the coefficients are scalar-quantized to reduce the number of bits. In this step a loss of information is produced. Finally, a coding step is needed; this coding step is performed by a process called *EBCOT* (Embedded Block Coding with Optimal Truncation).
  The JPEG2000 compression improves compression performance over Baseline JPEG. The JPEG2000 compression supports very low to very high compression rates and its strength is the ability to handle large bit rates very effectively.

4.8.2 GZIP Compression

Gzip compression is a lossless adaptive single pass encoding based on a combination of *LZ77* (Ziv & Lempel 1978) and Huffman coding. An input file is divided into a sequence of 32K byte blocks, where each block is compressed using a combination of *LZ77* and Huffman coding. The Huffman trees for each block are independent of those for previous or subsequent blocks; however, the *LZ77* algorithm uses a reference for duplicated strings from one block to another block. Each block has two parts: the Huffman code trees that describe the representation of the compressed data part, and a compressed data part. (The Huffman trees themselves are compressed using Huffman encoding.) The compressed data consist of a series of elements of two types: literal bytes (strings that have not been detected as duplicated within the previous 32K input bytes), and pointers to duplicated strings, where a pointer is represented as an ordered pair (offset, length), where the offset is the backward distance to the previous occurrence and the length is the number of characters in the re-occurring string.

4.8.3 Delta Compression

Differencing compression is the process of computing a compact and invertible encoding of a target file *T* with respect to a source file *S*. The output is called a delta file, and will be denoted here by $\Delta(T,S)$. Compression can be thought of as a special case of differencing in which the source data is empty. The general approach for differencing algorithms is to compress *T* by finding common substrings between *S* and *T* and replacing these substrings by a copy reference. The way the representation of such copy items is implemented determines the minimum length of a copy item. The delta file is then encoded as a sequence of elements which are either pointers to an occurrence of the same substring in *S*, or

individual characters that are not part of any common substring. To improve compression performance, pointers to previously occurring substrings in *T* are also used. Parsing of *T* is based on string matching algorithms, such as suffix trees or hashing with different time and space performance characteristics.

There are several practical applications that benefit from the use of delta compression since the new information that is received or generated is highly similar to information already being present. Such applications include distribution of software revisions, incremental file system backups, and archive systems, where using delta techniques is much more efficient than using regular compression tools. For example, incremental backups can not only avoid storing files that have not changed since the previous back-up and save space by standard file compression, but can also save space by differential compression of a file with respect to a similar but not identical version saved in the previous backup.

4.9 Redundancy

As compression is also based on redundancy, we need another look at this topic. The code of an image represents the body of information through a set of symbols. The elimination of redundant code is to use a lower number of symbols to represent the information.

Compression techniques using Huffman or arithmetic coding perform statistical calculations in order to eliminate redundancy and to reduce the original volume of the data. Most images have similarities or correlation between their pixels. These correlations are due to the existence of similar structures in the images since they are not completely random. Thus, the value of a pixel can be used to predict its neighbors.

There are three types of redundancy in monochrome single images: inter-pixel, psychovisual and coding redundancy (Karadimitriou 1996). When we have a set of similar images, we could have an inter-image redundancy; in this case, the images have similar pixel intensities in the same areas, comparable histograms, similar edge distributions, and analogous distributions of features. Set-redundancy is the inter-image redundancy that exists in a set of similar images and refers to the common information found in more than one image in the given set (Karadimitriou 1996). The set-redundancy can be used to improve compression. A limit to compression is imposed by the image entropy; lower entropy means higher compressibility.

4.10 Conclusions

We described probabilistic and algorithmic information theory approaches and the relationship between them. We also outlined the relationship of the two approaches with data compression. Shannon entropy evaluates the probability of each element of a message to assign short codes to elements with higher probability and long codes to elements with lower probability; if we reduce the bits required to encode a message, we obtain compression. In contrast, Kolmogorov complexity looks for the shortest program that can reproduce data; somehow we are compressing the way to represent the data, which is why an approximation of Kolmogorov complexity by a compression factor is a logical approach.

In the following chapters we will develop different methods for artifact detection based on compression techniques, since the presence of artifacts has effects on the mutual information between an original and its compressed-decompressed counterpart; mutual information is studied by information theory, and we have already seen that from information theory we can reach into the domain of compression techniques.

Chapter 5

Proposed Artifact Detection Methods

In this chapter we will come back to the theoretical aspects dealt with in Chapter 4 and revisit the basic rationale of *rate-distortion*: What we know even without deeper involvement into the theory and practice of image data compression is that, as a rule, a higher compression of an image will result in greater errors after compression-decompression. This is the basic background why one applies rate-compression as a simple metric for image quality assessment in systems which use data compression. In this dissertation we go one step further and will investigate whether rate-distortion can also be used as a tool for artifact detection.

Approaches such as rate-distortion are promising tools as they can be considered as being *parameter-free* and *model-independent*: a parameter-free approach does not depend on a clever selection of additional intricate input parameters, and a model-independent approach does not require a specific distribution of the input data. Thus, once one uses a parameter-free and model-independent approach, one disposes of a universal tool that can be applied widely.

In our case, we make an attempt to use the behavior of image data compression as a clue to image properties, for instance, the identification of patterns, the regularity of image data, or the presence of artifacts. As shown by (Li and Vitanyi, 2008), image data compression can help to analyze many data characteristics such as compression distances (i.e. locally different compressibilities), or image complexities (i.e., different irregularities). We follow two alternatives to fully exploit the behavior of image compression:

- We use lossless compression in order to explore redundancies and similarities in images. Our main tool will be the analysis of compression distances in the sense of "algorithmic mutual information" (i.e., the differences between independent and joint minimal encoding lengths of objects). This leads us to normalized compression distances, where the length of a code depends on the entropy of the data and remains constrained by the regularity of the data. When understood as a pattern recognition task, we can obtain hierarchical classifiers to be used for the grouping of artifacts in images. Further details can be found in Section 5.2.
- First, however, we use lossy compression in order to analyze data of variable quality (and thus with varying errors) as detailed in Section 5.1. When we compare compression results obtained by varying the compression quality parameter, we obtain "stacks" of errors that can be understood as variations of mutual information.

This opens the way towards classifications that separate regular from irregular data and help us to identify artifacts. This will be explained in Section 5.1.

In addition, this chapter is considered as one of the main contributions to this dissertation as it describes the systematic selection of three new promising candidate methods for artifact detection. In the following, we describe the rationale why we selected these four methods based on the information theoretical background outlined in Chapter 4, and how these methods can be applied in practice. Therefore, we also included an explanatory block diagram for each method that shall provide a quick overview.

- The first method shall prove how well Shannon's basic principles expressed as rate-distortion results can be used for artifact classification. To this end, we compress test images with different compression parameters and use the resulting compression errors as a feature vector that will undergo classification. The compression errors will be stored and analyzed as error maps that correspond to the selectable range of the compression parameter. For more details, see Section 5.1 and the block diagram contained in Figure 5.7.
- The second method shall demonstrate whether Kolmogorov complexity and normalized compression distances (being related to it) can be used as a basic alternative to the first method. We selected this second method as it is "Shannon-free" and can be used as a universal parameter-free tool. In our case, our goal is to discriminate images without artifacts from images that are affected by artifacts. Ideally, we would like to obtain a binary grouping. For more details, see Section 5.2 and Figure 5.15.
- Our third method shall tell us whether a new method consisting of a combination of *rate* (from the first method) and *compression distance* (from the second method) can outperform the other candidate methods. This is an exploratory approach that has to be carefully tested and verified based on sufficient cases and our established two-dimensional error maps; however, it represents a systematic approach to select the most promising sub-components from the first and the second method. For more details, see Section 5.3.1.1 ("Complexity-to-Error Migration") and Figure 5.22.
- Finally, we use common quality metrics as a reference for artifact detection with known tools. This approach is well-known in the steganalysis community and uses comparisons between an original image and low pass filtered versions of the image. The image differences can be described by error maps as defined for the first method together with well-known standard metrics of image processing. The results allow a straightforward comparison with the capabilities of the first and the second method. For more details, see Section 5.4 and Figure 5.41.

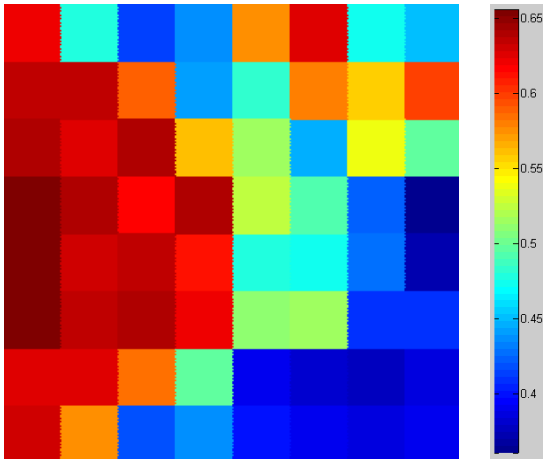
In a number of cases we had to add small experiments to help us in the understanding of some details of the proposed methods. This will be detailed in the next sub-sections, while the results of all the test runs that had to be conducted and our final conclusions will be described in Chapter 6 and 7.

In the following, we will use JPEG data compression packages to verify the size reduction of images. The JPEG packages are the three JPEG versions described in Chapter 4. Data compression can be achieved either without degradation of the data content (called lossless or noiseless compression), or we allow some content degradation (with some

distortion after compression-decompression, called lossy or noisy compression). In the latter case, one selects a JPEG quality parameter that controls internal quantization steps, and one obtains a compressed data set (Pennebaker and Mitchell, 1993). One has to note, however, that with JPEG the resulting compression rate is only available after compression (depending on the image content) and cannot be set a priori. Therefore, our tests with compressed data start with the selection of a quality factor, and end up with the measurement of the attained compression rate.

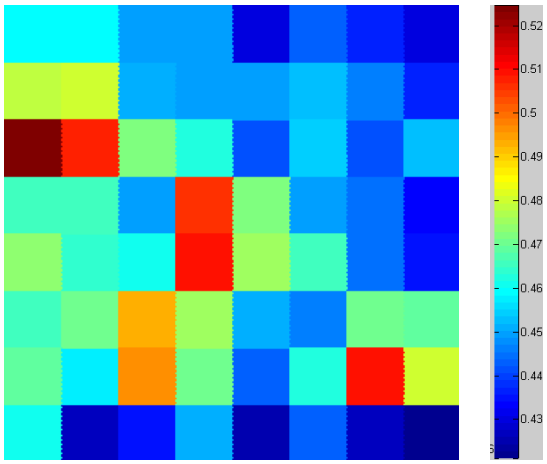
The main hypothesis that we propose to develop and apply the three methods listed above is that an artifact can have a high degree of irregularity, or a high degree of regularity compared with the local environment. From the point of view of the complexity, we can say that an artifact can be very complex or very simple; this is the reason why data compression is a useful tool to approximate and evaluate the complexity of the data.

(a) Satellite image of mixed land cover



(b) Complexity approximation

(a) Satellite image of a city



(b) Complexity approximation

Figure 5.1: Complexity approximation of satellite images using compression: (a) is a mixed land cover satellite image; (b) is the complexity approximation of the image (a); (c) is a satellite image of a city; (d) shows the complexity approximation of the image (c). The scale bars on the right show the size of the compressed file divided by the size of the original file.

In Figure 5.1, we can see an example of complexity approximation for optical satellite images. What is done is to take a satellite image and to cut it into adjacent patches of 64×64 pixels; each patch was compressed using the JPEG-LS compressor to obtain an inverse compression ratio (*ICR* defined here as size of the compressed image patch divided by the size of the original image patch; we use this value, because it is more easy to have a normalized value within the range from 0 to 1 when we have to compare different cases) that represents an estimation of the complexity of each patch; as results we obtained the illustrations of Figure 5.1 (b) and Figure 5.1 (d).

In this example, in Figure 5.1 (a) and its corresponding evaluation in Figure 5.1 (b), we can clearly see that the area of the city is more complex than the fields or than the water; red color is used for patches which have a higher *ICR* compared with the blue patches which have a lower *ICR*.

On the other hand, in Figure 5.1 (c) we show a satellite image of a city whose complexity should be nearly uniform across the whole image, but the presence of red zones in its corresponding complexity approximation shown in Figure 5.1 (d), indicate the possible presence of some kind of artifact.

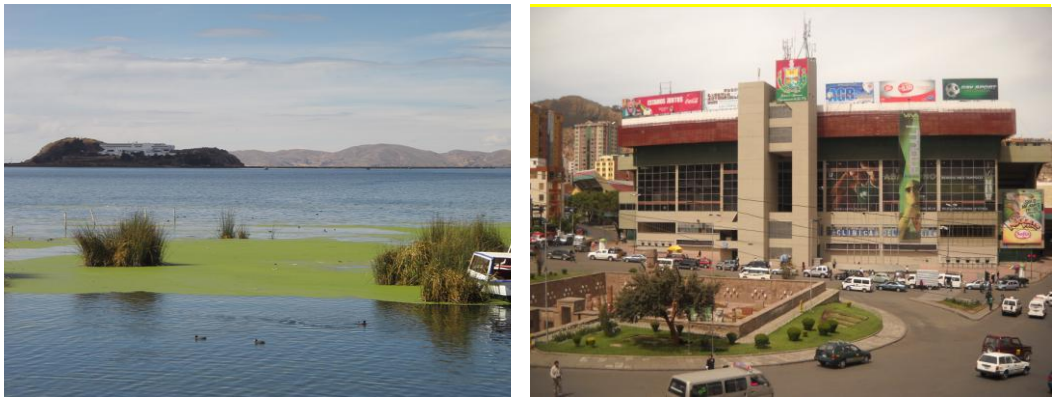
The idea is to use the *ICR* as an approximation to complexity and use it as a tool for artifact detection. The artifact detection only works when we have a homogeneous image where the complexity approximation is nearly uniform. In this case, if we have some sub-area being more complex than the remainder, we can say that there is a potential presence of artifacts.

5.1 Rate-Distortion Based Artifact Detection

In this section we will describe the first proposed method for artifact detection in optical satellite images; this method is based on the analysis of the Rate-Distortion (*RD*) function described in Section 4.5, basically the experimental curve obtained by using a series of variable compression parameters for images. The objective is to evaluate the behavior of the mean squared error between the original image and the compressed-decompressed image produced by each compression parameter. We first present a description of the principle on which the method is based and then some examples of obtaining experimental *RD* curves and how they vary from one image to another (without being degraded by artifacts), and we show the contrasting *RD* curves if the images were affected by an artifact. Then we describe the process for artifact detection using the proposed method; the analysis will take place in error or distortion space of the image. Then we apply this method to some examples to show their effectiveness in detecting artifacts.

5.1.1 Empirical *RD* Properties for Images with Artifacts

The Rate-Distortion function of a source with known probability distribution determines the minimum channel capacity required to transmit the source output as a function of the desired minimum average distortion, where the distortion function is a measure of agreement between source and system output specified by the user (Davisson 1972); for instance, one can use the mean squared error. In practice, the *RD* function is mainly used for sizing and designing data communication channels.



(a) Titicaca Lake (b) La Paz City

Figure 5.2: (a) Titicaca Lake is an image with an almost constant background; this image is less complex than image (b). (b) La Paz City is an image with a lot of details; it contains buildings, traffic, a public park, people, etc.

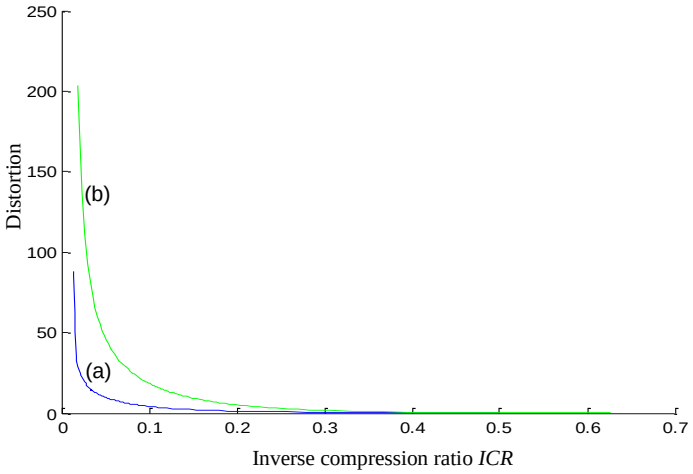


Figure 5.3: Measured Rate-Distortion Curve; the horizontal axis represents the inverse compression ratio *ICR* (size of compressed image file / size of original image file); the vertical axis represents the distortion calculated by *MSE* (mean squared error). Blue refers to Figure 5.2 (a) and green to Figure 5.2 (b); the *RD* values for the more complex image (b) are higher than the *RD* values for image (a).

For the purpose of this thesis, the Rate-Distortion (*RD*) is the bound of acceptable distortion for a resulting *ICR* (from a given compression quality parameter). The *RD* function measures indirectly the visual complexity of the images, for example, plotting the measured *RD* curve where the horizontal axis represents the inverse compression ratio *ICR* (size of compressed image file / size of original image file) and the vertical axis represents the distortion calculated here by the mean squared error. We can make an analysis of images, for example, Figure 5.2 shows two images: in (a) we have a picture without too many details where the background appears to be constant, and the lake and the sky are constant, too; while in (b) we show an image that has a lot of details because it contains buildings, traffic, a public park, people, etc. These conditions are reflected in the measured *RD* curve shown in Figure 5.3 in which we can see that the curve of the image (b), i.e., the green line, has higher values than the curve of image (a) which is shown in blue.

If artifacts increase the complexity of an image, we can perform the same experiment to see the change in the measured *RD* curve of an undisturbed image and of the image with added artifacts. Figure 5.4 (a) shows an undisturbed image of the Titicaca Lake in Peru and Figure 5.4 (b) shows the Titicaca Lake image with some defective columns. Figure 5.5 illustrates the measured *RD* curves for each image: blue for (a) and green for (b). We can see that the *RD* values of the image with defective columns exceed the values of the undisturbed image; this is because the defects create an image that is more irregular. To calculate the *RD* function, we used the baseline JPEG as a lossy compressor.

(a) Titicaca Lake (b) Titicaca Lake with defective columns

Figure 5.4: (a) Titicaca Lake is an undisturbed image. (b) Is image (a) with defective columns artificially introduced to analyze their influence on the measured *RD* curve.

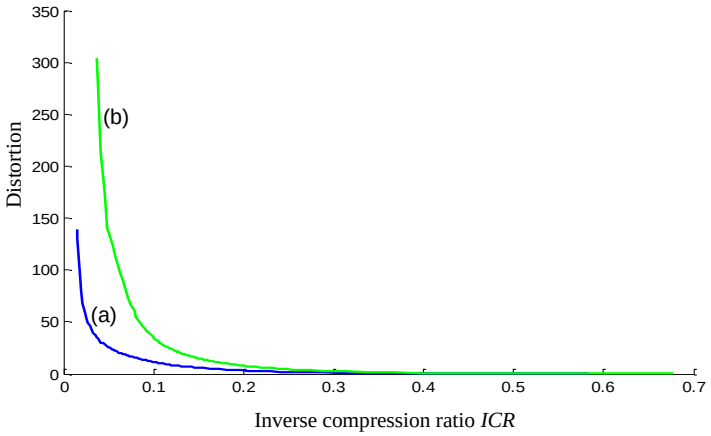


Figure 5.5: Measured Rate-Distortion curve; the horizontal axis represents the inverse compression ratio *ICR* as (size of compressed image file / size of original image file); the vertical axis represents the distortion calculated by *MSE* (mean squared error). The blue values represent Figure 5.4 (a), the green values refer to Figure 5.4 (b); the measured *RD* curve of the undisturbed image is lower than the measured *RD* curve of the image with defectives columns.

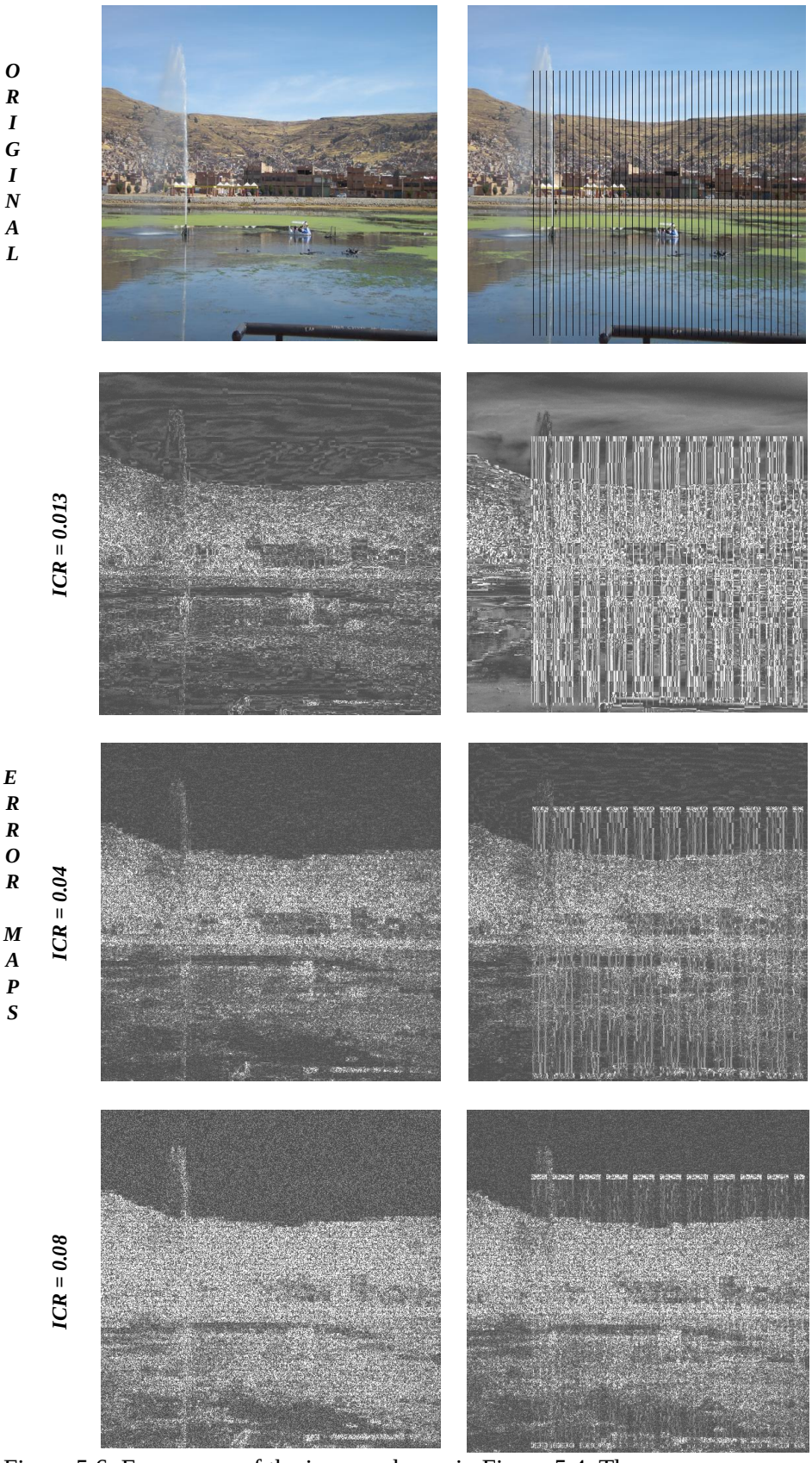


Figure 5.6: Error maps of the images shown in Figure 5.4. The error maps were computed for three inverse compression ratios *ICR* (0.013, 0.04, and 0.08).

In Figure 5.6, we present another small experiment related to the distortion space. Here we took the image of Figure 5.4 (a) and compressed it with three different quality factors resulting in three different *ICRs* (0.013, 0.04, and 0.08) using the baseline JPEG lossy compressor. The objective was to make a visual analysis of the error produced by the lossy compressor. The error maps E between the original image X and the compressed-decompressed image Y_q, are calculated as the absolute value of the differences between the original image X and compressed-decompressed image Y, $E_q = abs\ (X - Y_q)$, where q stands for the selected JPEG quality factor (0, 50, 75). The same process was made for the image of Figure 5.4 (b). Please note that all error maps have been individually stretched for better visualization.

In Figure 5.6, we can see that the defective columns of Figure 5.4 (b) persist in the error maps for each selected quality factor. The presence of the defective columns make the image more complex; the mean error for each selected compression quality factor is higher than the mean error produced by the image shown in Figure 5.4 (a).

In the following, we are going to use the characteristics explained above for the development of a parameter-free method for artifact detection in metric resolution satellite images based on the distortion space approach.

5.1.2 Artifact Classification in Error Maps

For artifact detection, we propose to use the measured *RD* curve obtained by compression of an image with different compression parameters and the evaluation of the distortion using the mean squared error (*MSE*) metrics. The idea is to examine the complexity of an artifact with a high degree of regularity or irregularity during image compression. The *RD* analysis for this purpose is shown as the block diagram contained in Figure 5.7.

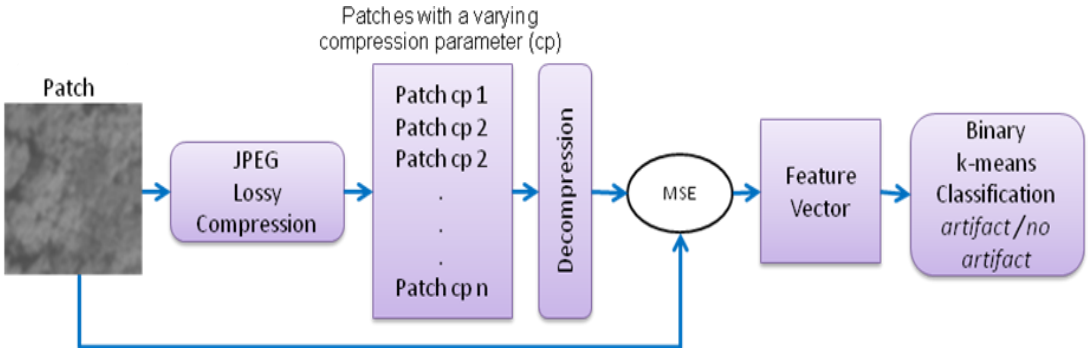


Figure 5.7: Block diagram for Rate-Distortion analysis: we take the image under test; we compress image patches with a varying compression parameter, then decompress the patches and calculate the error for each compression setting; based on the errors, we compose a feature vector and then apply classification methods.

The first step is to take the image under test *I*; then we cut the image *I* into *n* different patches X_i of 64×64 pixels as illustrated in Figure 5.8, with $i = 1, 2, \ldots, n$. A patch size of 64×64 pixels turned out to be the best compromise between a sufficient number of image pixels to compress and a good location accuracy defined by the patch size.

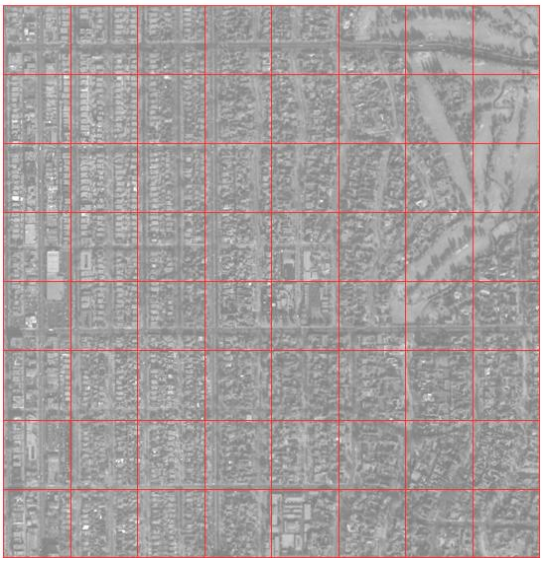


Figure 5.8: Image cut into patches of 64×64 pixels.

For each patch X_i, we compress the patch with a varying quality factor *q* using lossy compression; for this thesis, we used the baseline JPEG lossy compressor; using different quality factors, we obtain different compression factors. After that, we decompress the image and we obtain a decompressed image Y_{iq}. The next step is to calculate the error between the original patch X_i and the compressed-decompressed patch Y_{iq}; for the error calculation, we use the Mean Squared Error (*MSE*). Based on the errors for each quality factor *q* and for each patch X_i, we compose a feature vector $V_i = [F_{i1}, F_{i2}, \ldots F_{iq} \ldots F_{iQ}]$ where $F_{iq} = MSE(X_i, Y_{iq})$. Thus, we can arrange an easily calculable rectangular matrix of feature vectors:

$$V = \begin{bmatrix} F_{11} & F_{12} & \ldots & F_{1q} & \ldots & F_{1Q} \\ F_{21} & F_{22} & \ldots & F_{2q} & \ldots & F_{2Q} \\ . & . & \ldots & . & \ldots & . \\ F_{i1} & F_{i2} & \ldots & F_{iq} & \ldots & F_{iQ} \\ . & . & \ldots & . & \ldots & . \\ F_{n1} & F_{n2} & \ldots & F_{nq} & \ldots & F_{nQ} \end{bmatrix}.$$

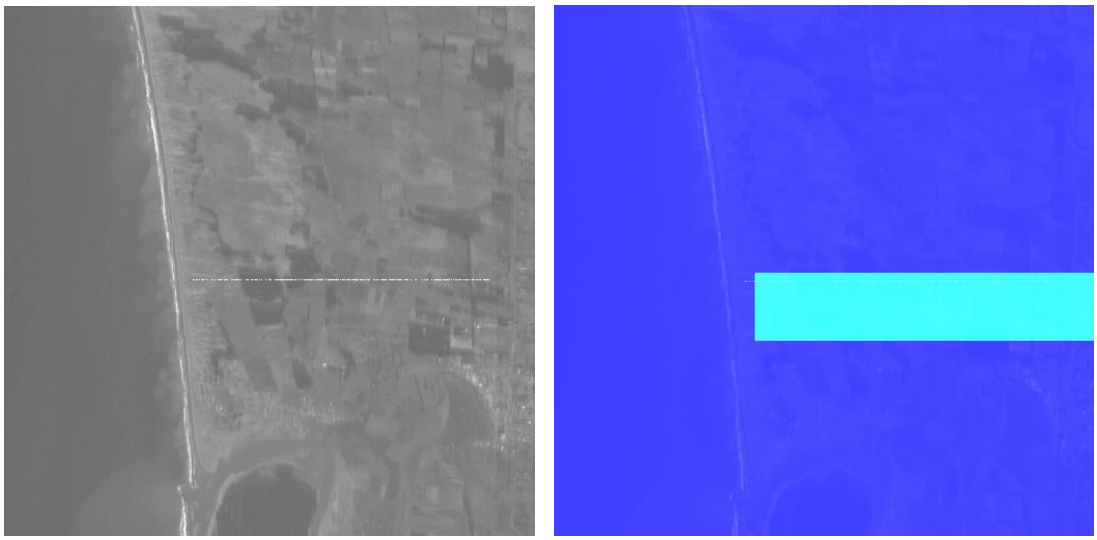
Finally, we apply a non-supervised classification method to this matrix *V*; we use the *K-MEANS* classification method, which is an unsupervised algorithm; *K-MEANS* classifies data into *K* (a positive integer value) classes. The algorithm calculates distances of the class centroids to all other data and then groups the classes according to the minimum distance principle. For the purpose of this thesis, we take a value of $K = 2$ that represents one group for images with artifacts and a second group for images without artifacts.

For us, the quality factor varies between *0* and *100*, so $Q = 101$, and for an image of 512×512 pixels, we obtain 64 patches of 64×64 pixels, thus $n = 64$.

5.1.3 Examples

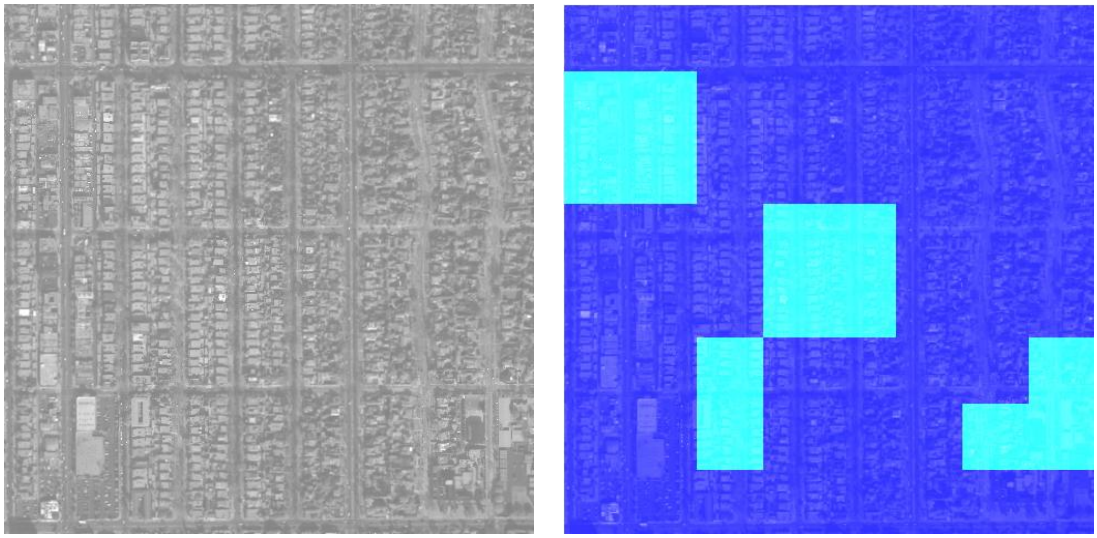
An example of the application of the method is presented in Figure 5.9 where stuck A/D converter bits appear; in this case, it is a SPOT image containing actual artifacts. The image was provided by the French Space Agency (CNES).

Another example for applying the proposed method is shown in Figure 5.10 where we have a city image with aliasing. The aliasing was generated synthetically; the result of the aliasing detection presents the areas that contain the artifacts.



(a) (b)

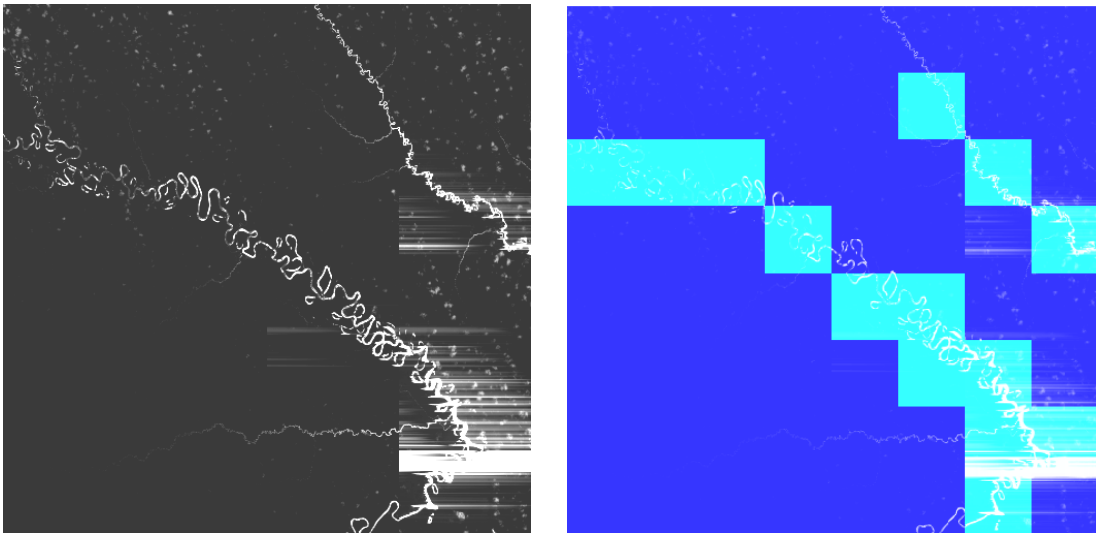
Figure 5.9: Stuck bits (SPOT image © CNES). (a) Stuck bits during A/D conversion create these saturated pixels. The artifact generates a line pattern (corresponding to the SPOT line sensor). (b) Detected patches with artifacts.



(a) Image with aliasing (b) Aliasing detection

Figure 5.10: (a) Satellite image with artificial aliasing. (b) Patch-wise aliasing detection using *RD* analysis.

A third example for applying the proposed method is shown in Figure 5.11 where we have an image with trailing charge problems.



(a) (b)

Figure 5.11: Trailing charge detection.

We can see that in both examples, the performance of the proposed method is rather good. The *RD* curve of an image with artifacts is different from the *RD* curve of an image without artifacts as illustrated in Figure 5.3 and Figure 5.5. The purpose of making an analysis by the *RD* function is to investigate about the error that results from different compression parameters; the error behaves differently for images with artifacts and images without artifacts; this analysis allows us to observe how an artifact is reflected in the error map produced by a compression-decompression comparison.

5.2 Normalized Compression Distance Based Artifact Detection

In this section we will propose another method for artifact detection in optical satellite images (called “second method” in the introduction to this chapter); this method is based on the analysis of similarity patterns between images with artifacts. This similarity is analyzed using the Normalized Compression Distance (*NCD*) function described in Section 4.4; here, we use a lossless compressor. The objective is to evaluate the existence of some similarity pattern between degraded images due to the presence of artifacts. We first present a description of the principle on which the method is based. Then we describe the process for artifact detection using the proposed method; the analysis will take place in the similarity space of the images. Then we apply this method to some examples to show their effectiveness in detecting artifacts.

5.2.1 Empirical Properties of *NCD* for Images with Artifacts

The Normalized Compression Distance (*NCD*) is a similarity metric based on compression techniques as an approximation to Kolmogorov complexity. The main contribution of this method is the fact that we do not need a feature extraction process for classification applications. This makes the method a parameter-free method; thus, it can be applied to any type of data, i.e., text, images, audio, etc.

To see some approximation of the relationship between complexity and compression, we make a small experiment. In Figure 5.12, we show two images with different visual complexity which is reflected in the inverse compression ratio *ICR*. In Figure 5.12 (a), we have a picture of a lake without many details where the background appears to be constant, and the lake and the sky are nearly constant, too; while in Figure 5.12 (b) we show an image that has a lot of details because it contains hills, houses, trees, people, cars, etc. The lake image has less visual complexity, and has a higher compressibility (and thus a lower *ICR*); while the other image contains many details; it is more complex and therefore, cannot be compressed as much (see Figure 5.13).

(a) Lake (b) Ski resort

Figure 5.12: (a) Image with an almost constant background; this image is less complex than image (b). (b) Image with a lot of details; it contains hills, houses, trees, people, cars, etc.

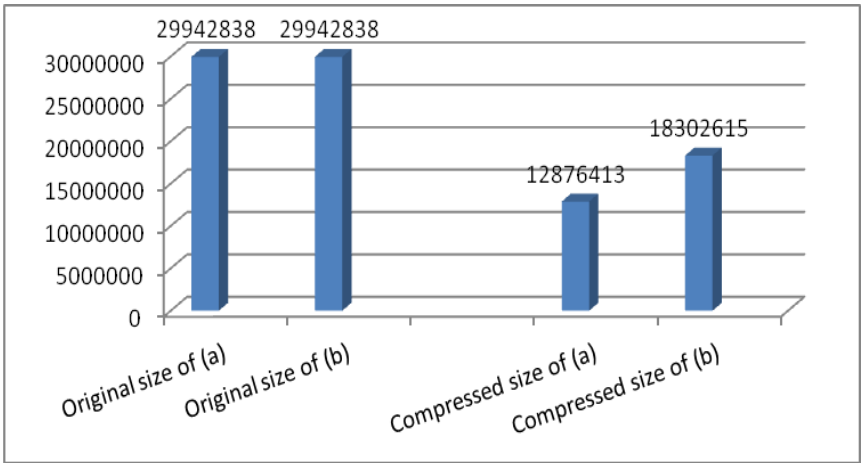
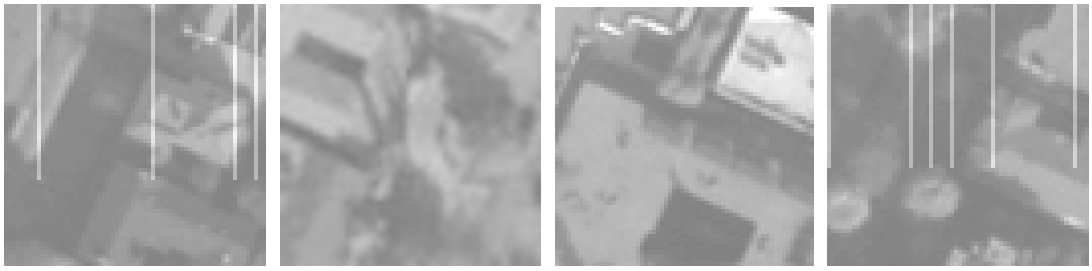


Figure 5.13: Original and compressed size of the images of Figure 5.12. The size is given in bytes. The compressor used was the JPEG-LS compressor.

In Figure 5.13, we can see that the size of the original images is 29,942,838 bytes for the images in Figure 5.12 (a) and Figure 5.12 (b). When we compress the images, we obtain a compressed size of 12,876,413 bytes for the image shown in Figure 5.12 (a) and 18,302,615 bytes for the image shown in Figure 5.12 (b). Thus, the *ICR* (size of compressed image file / size of original image file) is 0.43 and 0.61 for the image of Figure 5.12 (a) and the image of Figure 5.12 (b). These results show that an image with lower visual complexity has a higher compressibility than an image with higher visual complexity.

Since *NCD* is a similarity metric, we perform another small experiment. In Figure 5.14, we present four images that correspond to four city scenes. Figure 5.14 (a) and Figure 5.14 (d) show images with artifacts (strips). Figure 5.14 (b) and Figure 5.14 (c) present artifact-free images. For these four images, we calculate the *NCD* between all images pairs, i.e., *NCD(a,b), NCD(a,c), NCD(a,d), NCD(b,c), NCD(b,d)* and *NCD(c,d)*. The objective of this experiment is to analyze the potential similarities between the images that have artifacts and the images that do not have artifacts.



(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4

Figure 5.14: Four different city scenes. Scene 1 and Scene 4 have artifacts (strips).

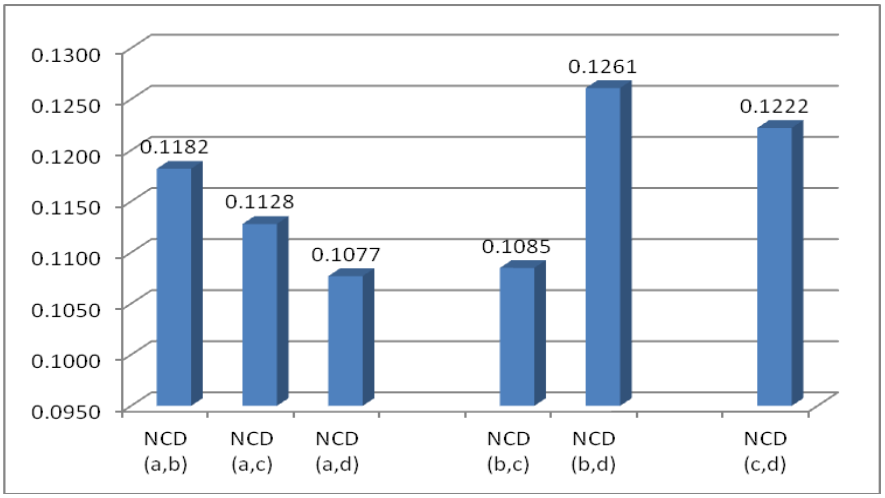


Figure 5.15: Normalized Compression Distances (*NCDs*) between the four different scenes of Figure 5.14, (a) for Scene 1, (b) for Scene 2, (c) for Scene 3 and (d) for Scene 4. The two smallest *NCD* values are for the *NCD* between (a) and (d) and the *NCD* between (b) and (c).

In Figure 5.15, we present the *NCD* values between the images of Figure 5.14. We can see that *NCD(a,b) = 0.1182, NCD(a,c) = 0.1128, NCD(a,d) = 0.1077, NCD(b,c) = 0.1085, NCD(b,d) = 0.1261* and *NCD(c,d) = 0.1222*. Based on these values, we can say that the image of Figure 5.14 (a) is more similar to the image of Figure 5.14 (d) than to the other ones; the image of Figure 5.14 (b) is more similar to the image of Figure 5.14 (d). This analysis indicates that the images with artifacts can be rather similar among themselves; likewise, the images without artifacts are more similar among themselves.

We are going to use these characteristics explained above to develop a parameter-free method for artifact detection in satellite images based on similarity using the Normalized Compression Distance.

5.2.2 Artifact Classification by Similarity

For artifact detection using the Normalized Compression Distance (*NCD*), we propose to analyze the potential similarity between images with artifacts. The reason for this is a potential similarity between artifact patterns. The first step is to take a satellite image *I* and to cut it again into *n* patches X_i of the same size. From these patches X_i, we calculate the distance matrix between them using the *NCD* described by Equation 4.10, thus $d_{ij} = NCD(X_i,X_j)$.

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1j} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2j} & \dots & d_{2n} \\ . & . & \dots & . & \dots & . \\ d_{i1} & d_{i2} & \dots & d_{ij} & \dots & d_{in} \\ . & . & \dots & . & \dots & . \\ d_{n1} & d_{n2} & \dots & d_{nj} & \dots & d_{nn} \end{bmatrix}.$$

The matrix *D* is a square matrix of size $n \times n$, where *n* is the number of patches. For larger images this can lead to considerable computing effort.

Finally, we apply a non-supervised classification to this distance matrix *D*. Our classification method is a hierarchical classification dendrogram as shown below in Figure 5.17. The dendrogram is a graphical representation of data that sorts them into sub-categories as a tree and the desired level of detail controls the classification depth. This type of representation allows a clear view of the relationships between data classes. To plot the dendrogram we use the Euclidean distance method for the evaluation of the distances between the data using the following MATLAB instructions:

- Calculate the Euclidean distance — `Distance = pdist(D)`
- Link the distances — `Tree = linkage(Distance)`
- Read the labels — `Labels = importdata('Labels.txt')`
- Plot the dendrogram:

```
dendrogram(Tree,'colorthreshold','default',
                'labels',Labels)
set(Dendrogram,'LineWidth',2)
```

We also used the *K-MEANS* classification method that has been described above with a parameter value of $K = 2$ that yields one group for images with artifacts and a second group for images without artifacts.

A block diagram of this method is presented in Figure 5.16.

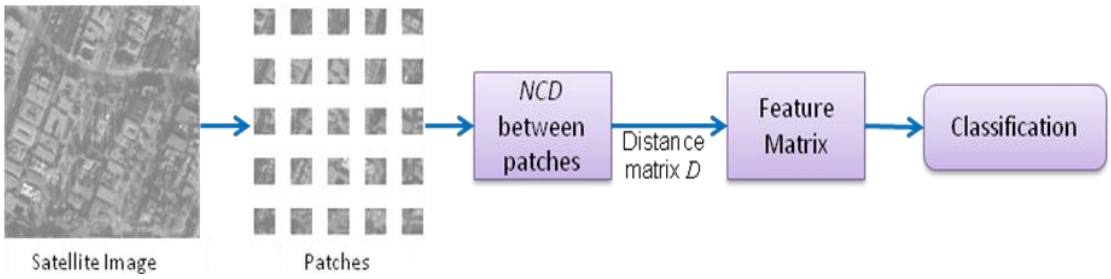


Figure 5.16: We take a satellite image and cut it into patches of 64×64 pixels. With these patches we calculate the distance matrix between them using *NCD*; finally, we apply a hierarchical classification method to cluster the patches into classes with and without artifacts.

We use the *NCD* to investigate whether patches with artifacts will have similar patterns and to see if we can create a single cluster for all artifact affected patches.

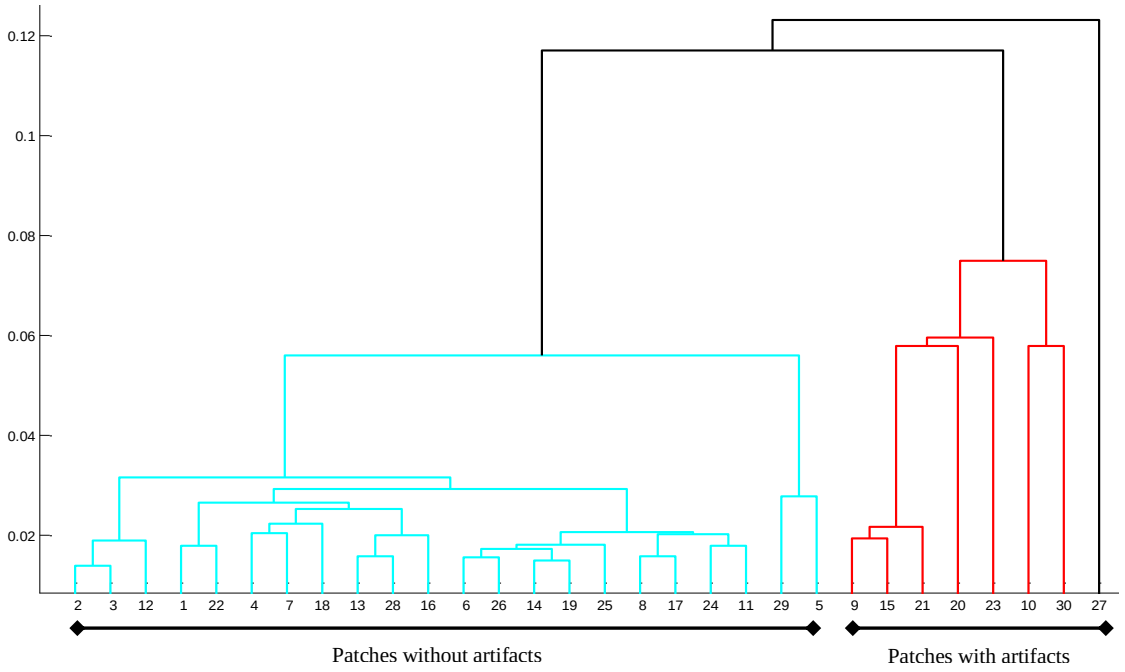


Figure 5.17: Hierarchical classification of patches: we can see two clusters, a red cluster for patches with artifacts, and a cyan cluster for patches without artifacts. On the ordinate axis we see the hierarchical level assigned by the MATLAB dendrogram routine.

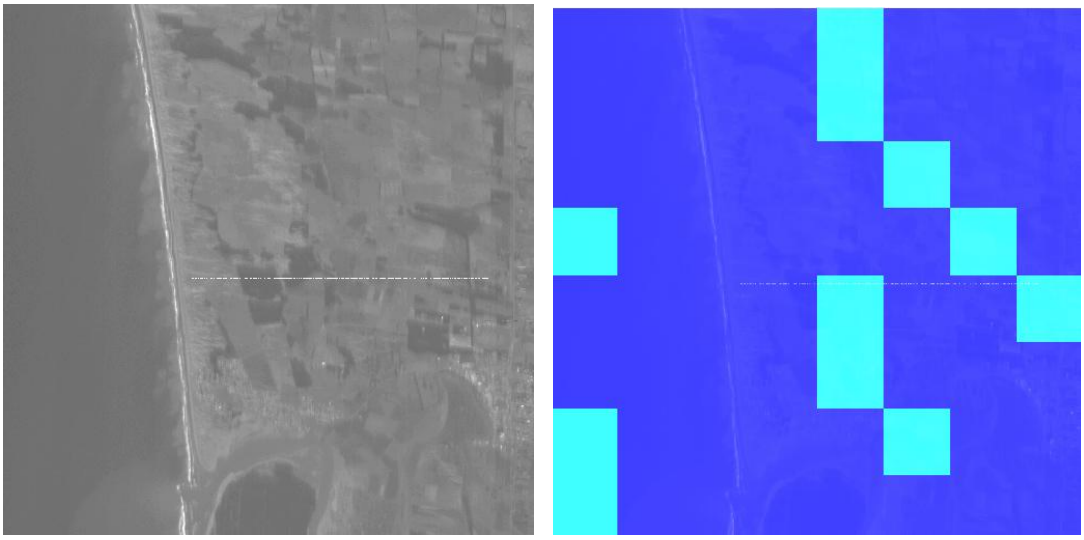
In this dendrogram representation, we can see how the patches with artifacts and the patches without artifacts can form two different clusters on a given hierarchical level. The analysis begins by determining two principal groups in the dendrogram; after that, we verify the patches that belong to one group and the patches that belong to the second group; with these results we prepare a confusion matrix to calculate the error. It is important to say that we do not have an automatic process to identify groups containing patches with artifacts and group containing patches without artifacts. Instead, it is necessary to visualize the patches of each group for a final decision.

5.2.3 Examples

An example of the application of the method is presented in Figure 5.18; we have again the image where a stuck A/D converter bit appears.

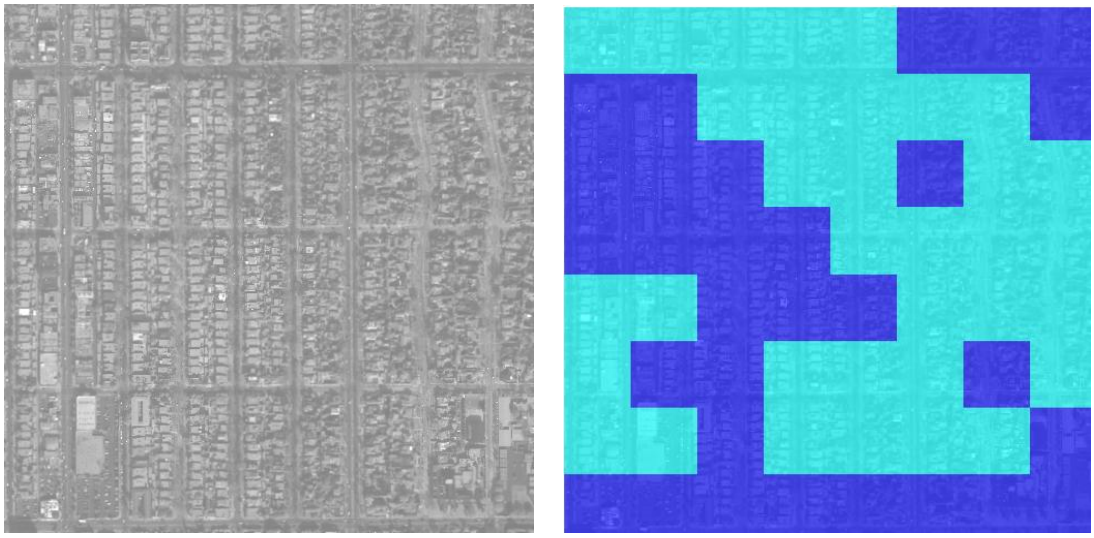
Another example for applying the proposed method is shown in Figure 5.19 where we have a city image with simulated aliasing.

A third example for applying the proposed method is shown in Figure 5.20 where we have an image with trailing charge problems.



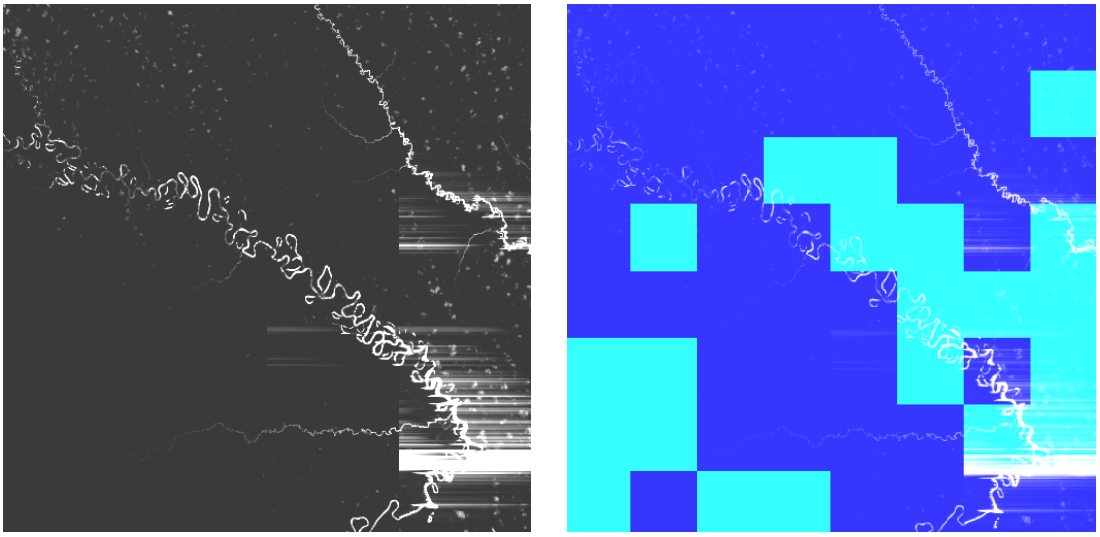
(a) (b)

Figure 5.18: Results of a stuck bit detection using the *NCD* method.



(a) Image with aliasing (b) Aliasing detection

Figure 5.19: (a) Satellite image with artificial aliasing. (b) Results of patch-wise aliasing detection using *NCD* analysis.



(a) (b)

Figure 5.20: Results of trailing charge detection.

A fourth example of the application of the *NCD* analysis method is to apply it to images with actual instrumental artifacts. We took images acquired by the ROSIS sensor; these data consist of hyperspectral images comprising 7946×512 pixels, 14 bits per pixel, and 115 spectral bands. For this experiment we work with sub-scenes of 512×512 pixels per spectral band. Our selected sub-scenes correspond to different land cover classes such as forest, or agricultural fields (see Figure 5.21) for a selected band.

The first thing to do is a manual analysis to determine the existence and the location of artifacts; for this task we take the hyperspectral image H that has a size of 7946×512×115 pixels; we extract different single band sub-scenes SI of 512×512 pixels; each sub-scene $SI_{512x512}$ is an image with 14 bits per pixel, $SI = [b_{13}\ b_{12}\ \ldots\ b_0]$. For each sub-scene SI we

make a visual analysis of each bit plane to detect potential perturbations; in effect, we detected strips in the two least significant bits b_0 and b_1 as shown in Figure 5.22.

Figure 5.21: Examples of 512×512 sub-scenes with actual instrumental artifacts of the ROSIS sensor (images provided by the German Aerospace Center - DLR).

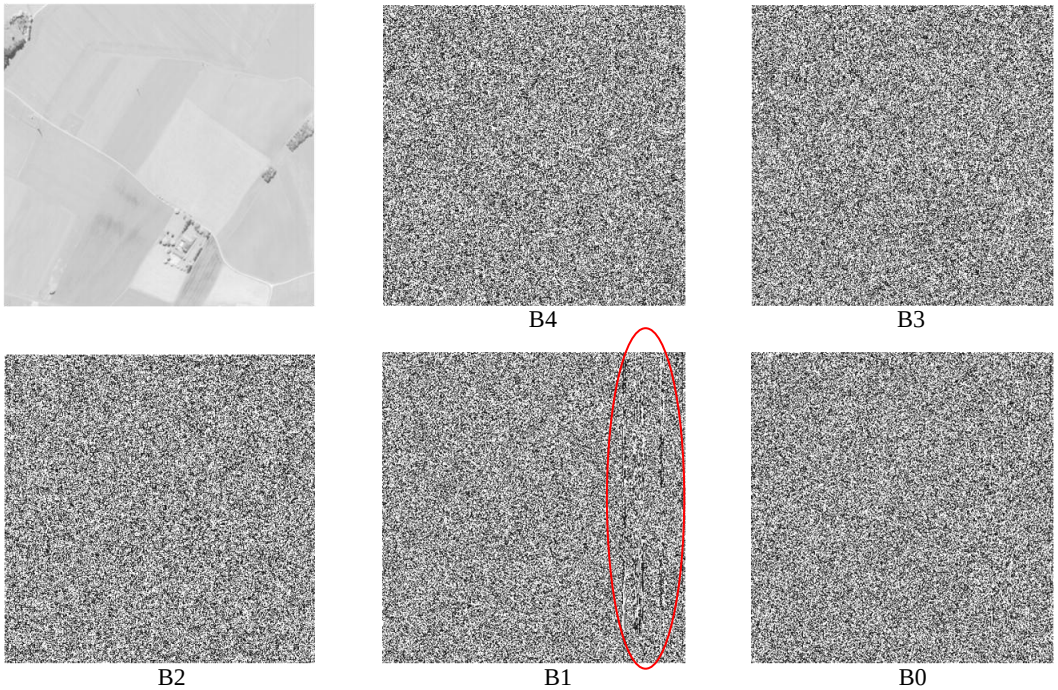


Figure 5.22: Bit plane analysis for strip detection in a ROSIS image: we can see the strips in the B1 bit plane.

We can see strips only in the B1 bit plane of the image; this is the reason why we cannot see the strips via a visual inspection of the full range image; however, the strip artifacts may produce errors during further image interpretation.

After the manual bit plane analysis, we know the existence and location of artifacts; now we apply the proposed automated method to the same sub-scene *SI* for artifact detection and check whether the method can detect the strips. The results are not encouraging and are shown in Table 5.1.

FOREST

Kind of Artifact			**NCD – JPEG-LS**	**NCD – CompLearn/zip**
Strips			61.11%	41.67%

AGRICULTURAL FIELDS 1 and 2

Kind of Artifact			**NCD – JPEG-LS**	**NCD – CompLearn/zip**
Strips	***1***		48.98%	51.02%
	2		60.94%	56.25%

Table 5.1: Artifact detection success rates with actual instrument data.

We have applied the method to different land cover sub-scenes such as forest and agricultural fields. We used both the JPEG-LS compressor as well as the CompLearn software package that contains a zip compressor. The results show a bad detection performance, we obtain only 61.11% as maximum success of detection; this result was obtained detecting strips in a forest scene, where we also have a considerable discrepancy in the performance of the two compressors. These bad results may be due to the fact that the strips only appear in the B1 bit plane, and due to their low intensity they are not detected in the full range image.

Given these results, we made another experiment; we took the B1 bit plane containing strips. The next step is to cut the bit plane image into patches X_i of 64×64 pixels as shown in Figure 5.23. For each patch X_i, we convert the patch into an ASCII text string S_i with values of zeros and ones; thus $S_i = [p_1, p_2, \ldots, p_j, \ldots, p_m]$ where p_j is the value of each pixel of the binary patch X_i; as the size of each patch is 64×64 pixels, the total length of the resulting string $m = 4096$. After that, we calculate the Normalized Compression Distance between all strings and we build a distance matrix $DS = \{d_{ij}\}$, where d_{ij} is the *NCD* value of strings S_i and S_j, $d_{ij} = NCD(S_i, S_j)$ using Equation 4.10. We obtain the distance matrix:

$$DS = \begin{bmatrix} d_{11} & d_{12} & \ldots & d_{1j} & \ldots & d_{1n} \\ d_{21} & d_{22} & \ldots & d_{2j} & \ldots & d_{2n} \\ . & . & \ldots & . & \ldots & . \\ d_{i1} & d_{i2} & \ldots & d_{ij} & \ldots & d_{in} \\ . & . & \ldots & . & \ldots & . \\ d_{n1} & d_{n2} & \ldots & d_{nj} & \ldots & d_{nn} \end{bmatrix},$$

where n is the number of patches or the number of strings; if we work with a sub-scene of 512×512 pixels, the number of 64×64 patches is $n = 64$.

Finally, with this distance matrix *DS*, we perform a non-supervised hierarchical classification for artifact detection as described above.

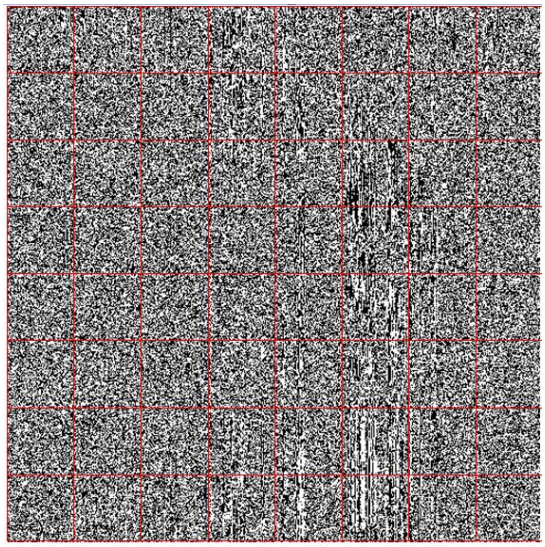


Figure 5.23: Bit plane patches of 64×64 pixels for the calculation of the distance matrix between patches.

For making the patch-to-string conversion, we analyzed two options: horizontal scanning and vertical scanning.

For horizontal scanning, we take the *i-th* bit plane patch X_i formed by $X_i = \{p_{ij}\}$ where p_{ij} is a pixel with the two possible values of zero or one; we create the *i-th* string S_i as follows:

$$S_i = [p_{11}, p_{12}, \ldots, p_{1j}, \ldots p_{1n}, p_{21}, p_{22}, \ldots, p_{2j}, \ldots, p_{2n}, \ldots, p_{i1}, p_{i2}, \ldots, p_{ij}, \ldots, p_{in}, \ldots, p_{n1}, p_{n2}, \ldots, p_{nj}, \ldots, p_{nn}].$$

For vertical scanning, we create the text string S_i as follows:

$$S_i = [p_{11}, p_{21}, \ldots, p_{i1}, \ldots p_{n1}, p_{12}, p_{22}, \ldots, p_{i2}, \ldots, p_{n2}, \ldots, p_{1j}, p_{2j}, \ldots, p_{ij}, \ldots, p_{nj}, \ldots, p_{1n}, p_{2n}, \ldots, p_{in}, \ldots, p_{nn}].$$

The horizontal scanning and the vertical scanning are shown in Figure 5.24. As each patch has 64×64 pixels and all rows or columns have to be put into a sequence, then each text string will comprise 4096 elements.

Then we have two data sets for further processing: a data set $S_{horizontal}$ as the result of horizontal scanning, and another data set $S_{vertical}$ as the result of vertical scanning. For each data set $S_{horizontal}$ and $S_{vertical}$, we calculate by using a zip compressor the distance matrices $DS_{horizonal}$ and $DS_{vertical}$. The success rate for artifact detection after a *K-MEANS* classification is shown in Table 5.2:

		Horizontal Scanning	**Vertical Scanning**
NCD for text strings		50%	81.25%

Table 5.2: Results for the two scanning processes.

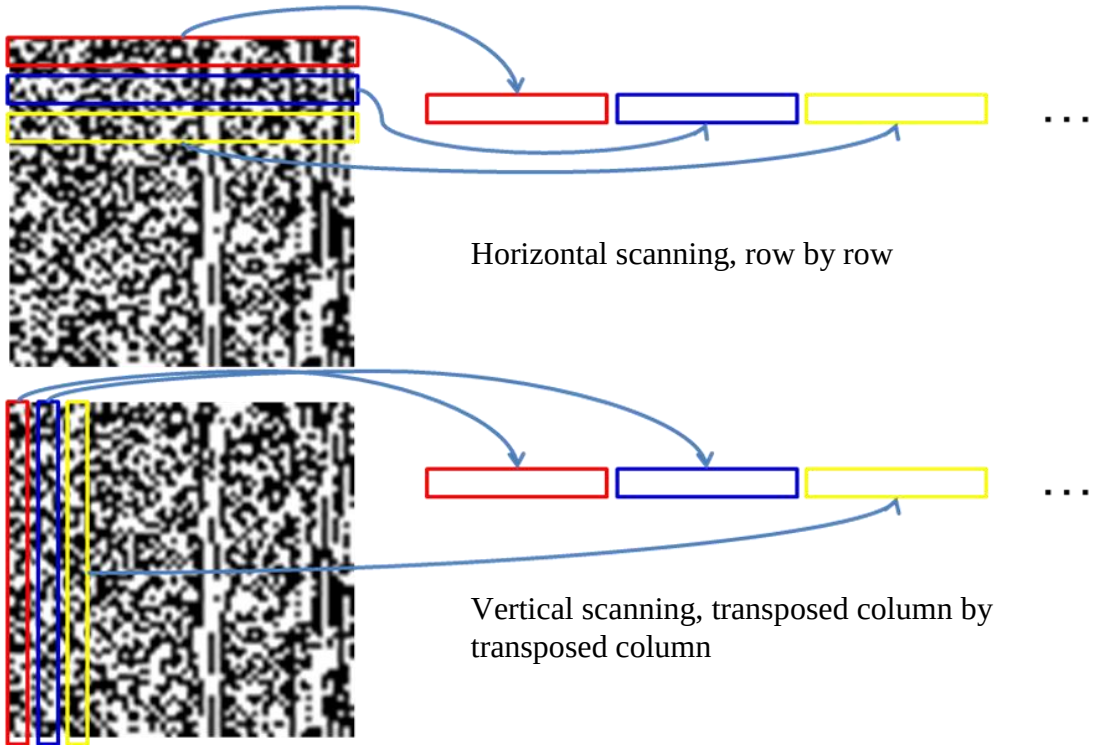


Figure 5.24: Formation of the text string based on bit plane images: we analyzed two options for the scanning process, horizontal scanning and vertical scanning. Finally, each text string will comprise 4096 elements.

We can see that we have no clear result for the dataset with horizontal scanning; in this case, the proposed method could not detect the artifacts properly; but if we use vertical scanning, the results are somewhat better; the method is capable to detect the artifacts with 81.25% of success. The reason why the results for vertical scanning are better could be that the strips have the same vertical orientation; then the method is capable of recognizing the artifacts due to their orientation; in contrast, when using horizontal scanning we miss the artifact patterns and it is not possible to detect them. In case of horizontal striping, we would expect the opposite result.

5.3 New Rate-Distortion Aspects for Artifact Detection

In this section, we will describe some new aspects of the Rate-Distortion curve; these new aspects are based on a different way to measure the distortion and the similarity between the original image and the compressed-decompressed image. We will use the complexity approach for similarity measurements and we will also calculate the similarity between the three elements in the lossy compression process: the original image, the compressed-decompressed image and the error between them. The objective is to use these three elements and obtain new curves for artifact detection analyzing the complexity-to-error migration. First, we present a short introduction to the new ideas. After that, we give a description of the Complexity-to-Error Migration (*CEM*) effects. Thirdly, we will present the Kolmogorov Structure Function. Then we will describe the artifact detection process using

the new approach. Finally, we apply this methodology to some examples to show their effectiveness in detecting artifacts.

In the previous sections, we presented two methods for artifact detection, one based on the *RD* curve, and a second one based on *NCD*. The *RD* curve can distinguish a complex image from another less complex image by its ordinate values. Here we assume that an artifact is something more complex or rather regular with respect to the local undisturbed complexity of an image. This led us to the second method proposed in Section 5.2 that is using Normalized Compression Distance (*NCD*) based on Kolmogorov Complexity.

Now we propose a third approach where we combine the *RD* and the *NCD* methods in order to profit from multi-dimensional data analysis. In Section 5.3.1 we present four approximations of the Rate-Distortion curve using image complexity. The first three approximations (Section 5.3.1.1) compare

- the original and the distorted image,
- the original and the residual error image
- the residual error and the distorted image

When we combine them we can perform a 3D distortion analysis. The fourth approximation method (see Section 5.3.1.2) compares the original and the disturbed image using the Kolmogorov Structure Function. As always, our tests were conducted with different quality factors.

5.3.1 Different Approaches for Rate-Distortion Function

In this section, we present the four approximations addressed above. The first three approaches use the *SNCD* criterion to quantify distortion (see below), while the fourth method uses *KSF* (see below).

5.3.1.1 Complexity-to-Error Migration

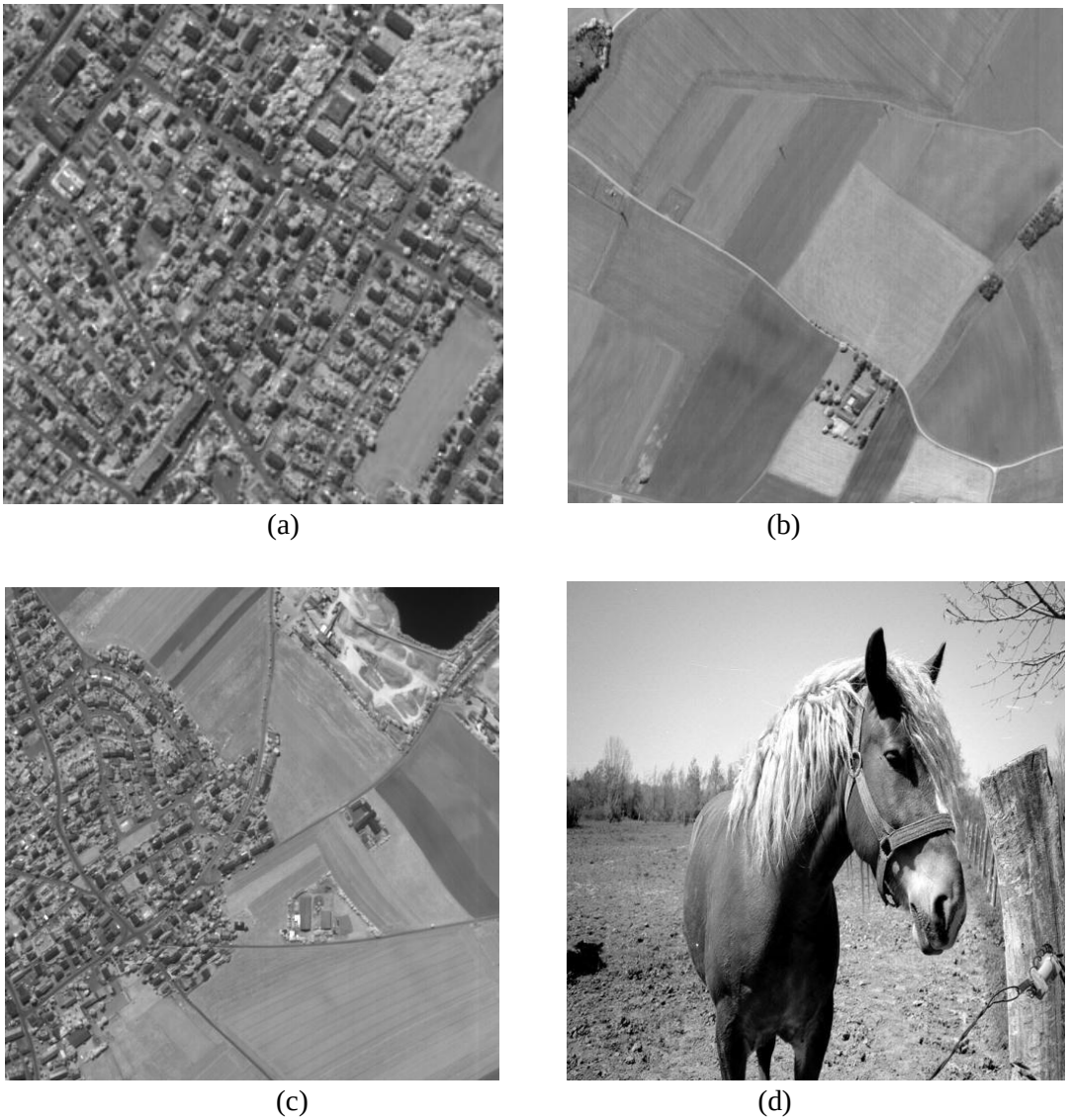
We analyzed the Normalized Compression Distance *(NCD)* that should be an approximation of the Normalized Information Distance *(NID)* in more detail (see Section 4.4). The *NID* is a symmetric measure as the Kolmogorov Complexity $K(x, y) = K(y, x)$. However, we observed experimentally that the *NCD* is not symmetrical, $NCD(x, y) \neq NCD(y, x)$. Therefore, we use a Symmetric Normalized Compression Distance (*SNCD*) defined as the arithmetic mean of $NCD(x, y)$ and $NCD(y, x)$. The *SNCD* is given by:

$$SNCD(x, y) = \frac{1}{2}\left[NCD(x, y) + NCD(y, x)\right], \tag{5.1}$$

$$SNCD(x, y) = \frac{C(x, y) + C(y, x) - 2 \times \min\{C(x), C(y)\}}{2 \times \max\{C(x), C(y)\}}. \tag{5.2}$$

As an approximation to the Rate-Distortion Function, we apply the Complexity-to-Error Migration (*CEM*) analysis. In order to obtain *CEM* curves, we need a lossy compressor that allows us to vary the rate by different quality factors, and a lossless compressor that allows us to estimate the Kolmogorov complexity when calculating the *SNCD*. We used two image compressors, the baseline JPEG lossy compressor− based on the discrete cosine transform *DCT*−and the JPEG 2000 lossy compressor, based on a wavelet transform. Both compressors produce different effects in the compressed images; these effects can be seen in the following examples.

Taking the images of Figure 5.25, we applied both compressors. With similar *ICRs* we obtained the different results shown in Figs. 5.26 to 5.29.



(a) (b)

(c) (d)

Figure 5.25: (a) Satellite image of a city, (b) satellite image of an agricultural field, (c) image with different land cover types, and (d) image of a horse.

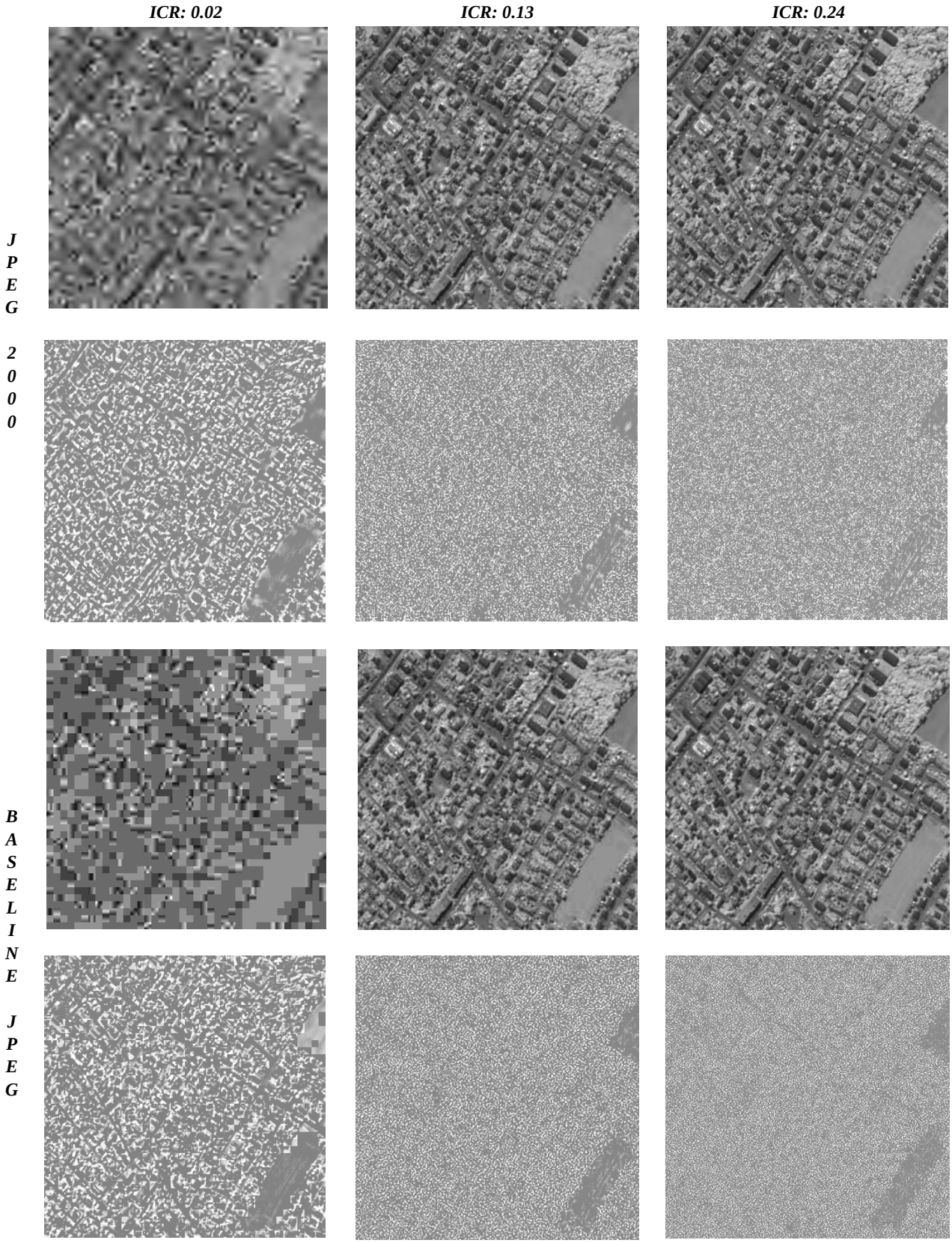


Figure 5.26 Top row: JPEG 2000 compressed-decompressed image with three compression settings. Second row: JPEG 2000 error map images. Third row: baseline JPEG compressed-decompressed image with three compression settings. Fourth row: baseline JPEG error map images.

The results shown in Figure 5.26 used the image of Figure 5.25 (a).

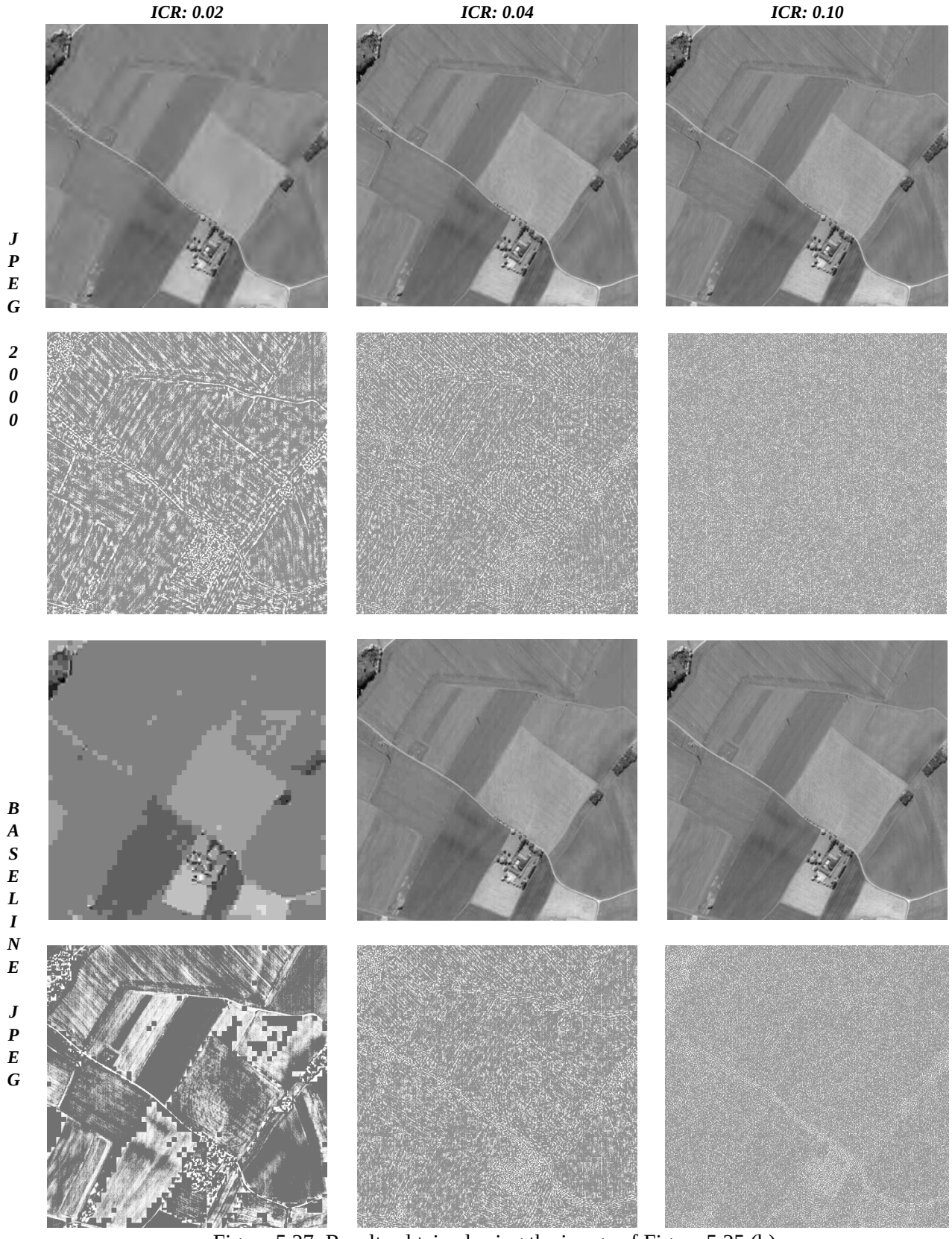


Figure 5.27: Results obtained using the image of Figure 5.25 (b).

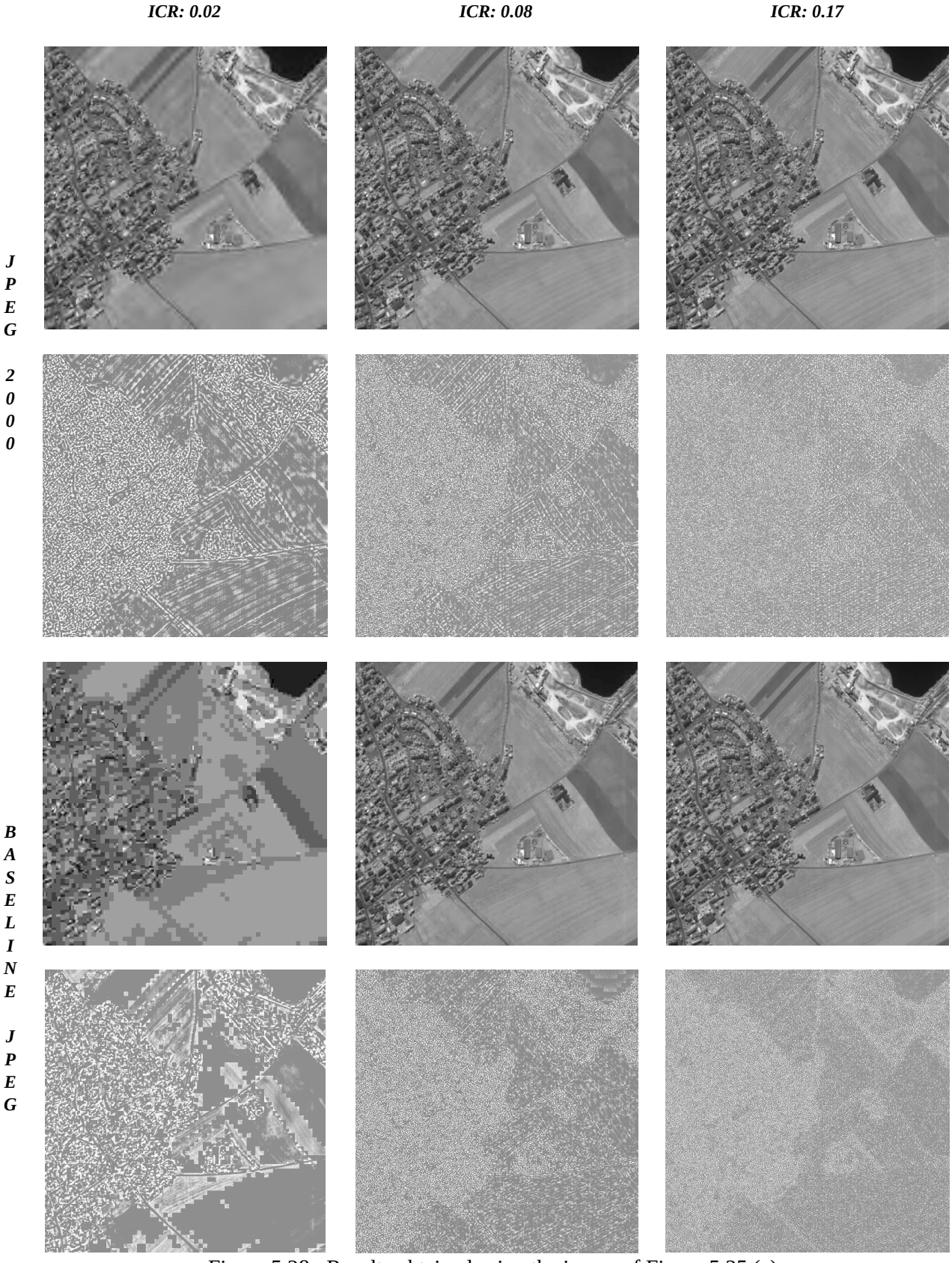


Figure 5.28: Results obtained using the image of Figure 5.25 (c).

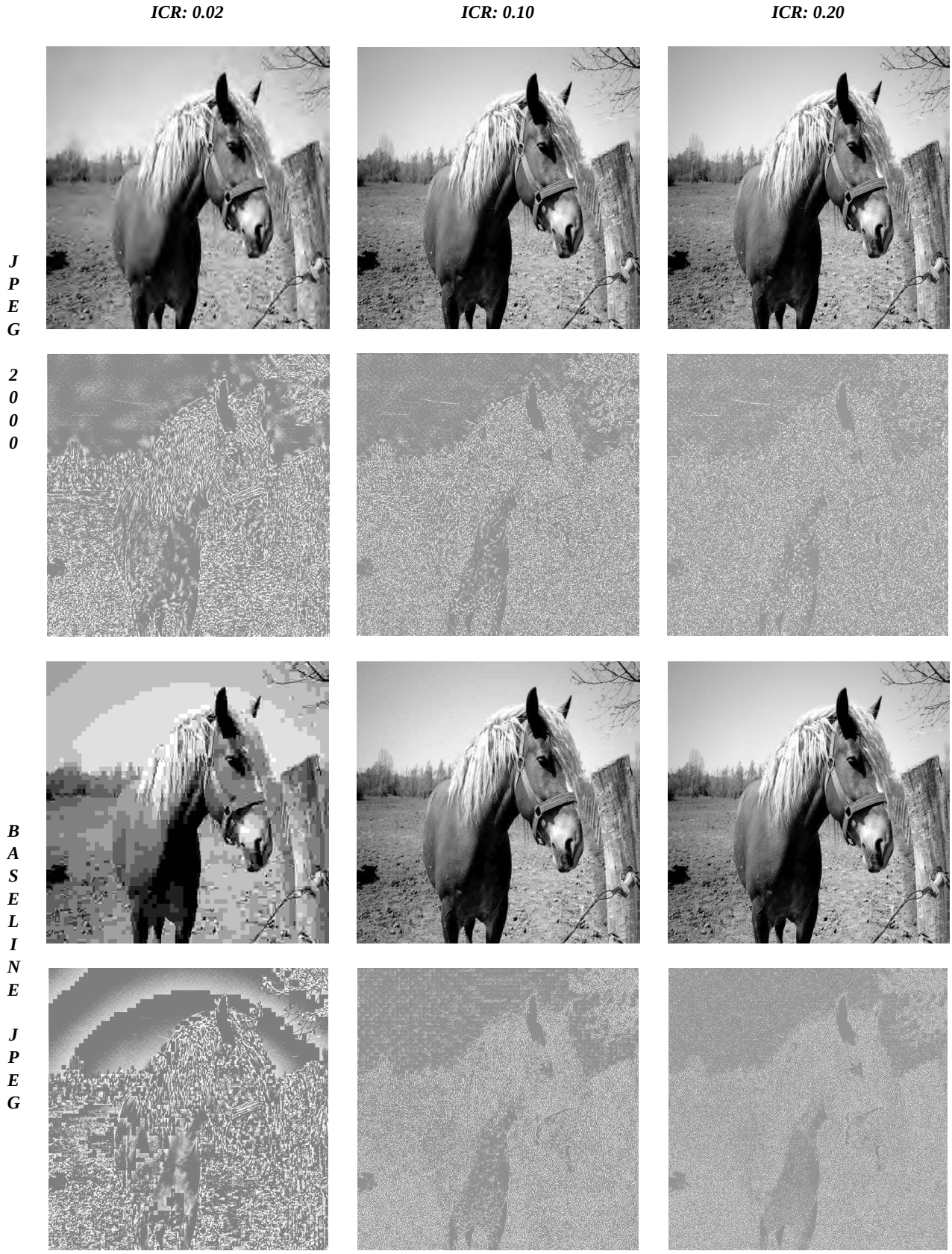


Figure 5.29: Results obtained using the image of Figure 5.25 (d).

The errors *E* between the original image and the compressed-decompressed image are the absolute difference values between the original image *X* and the compressed-decompressed image *Y*, $E_q = abs\ (X - Y_q)$, where *q* is the JPEG quality / rate factor. Different factors lead to a variation of the error maps. This variation of the errors can be very important because we can see that the information content is reflected in the error maps, especially when an image has a well-defined structure as in the case of the horse. We can also see that the error differs between the baseline JPEG and the JPEG 2000 compression.

The error maps are very important for artifact detection since we want to analyze how an artifact passes (or "migrates") into the image error maps. We can see that the image content passes better to the error map when using the baseline JPEG option; as JPEG 2000 provides better performance in compression, the image error map contains less information. Thus, our preferred choice for our application is the baseline JPEG option.

In order to validate the error map behavior explained above, we plot the curves for

- the *CEM* curve comparing (for varying compression settings) the original image *X* and the compressed-decompressed image *Y*
- the *CEM* curve for the original image *X*, and the error map *E*
- the *CEM* curve for the error map *E* and the compressed-decompressed image *Y*
- a 3-dimensional view of these three curves.

The abscissa of the *CEM* curve is the *ICR*, and the ordinate is the *SNCD*. To calculate the *SNCD*, we need a lossless compressor to approximate the Kolmogorov complexity; for comparison, we used the JPEG-LS compressor and a ZIP compressor (see Chapter 6).

The *CEM(X,Y)* curve of the original image *X* and the compressed-decompressed image *Y* should be a normal Rate-Distortion curve. In the case of high compression, the *SNCD* value will be close to one, representing the greatest distance between the original image *X* and the compressed-decompressed image *Y*; in case of lower compression, the *SNCD* value will be closer to zero, meaning that the original image *X* is very similar to the compressed-decompressed image *Y*.

The *CEM(X,E)* curve between the original image *X* and the error map image *E* should represent a steadily rising curve (*E* increases, while *X* always remains the same). A higher inverse compression ratio has a near zero error image with a big distance to the original (non-zero) image.

The *CEM(E,Y)* curve between the error map *E* and the compressed-decompressed image *Y* could have a particular behavior as shown in Figure 5.30 due to the varying *Y* and *E*. At low inverse compression ratios, we have considerable compression and an error map with noticeable entries. When the inverse compression ratio increases, the compression errors become smaller until we reach a point that contains less information in the error map (see the red circle in Figure 5.30). When the inverse compression ratio continues to increase, the compressed-decompressed image *Y* becomes almost equal to the original image *X*. Hence, the error map *E* contains little information which makes the distance between *E* and *Y* larger again. An interesting analysis will be to look at the different slopes and the minimum shown in Figure 5.30.

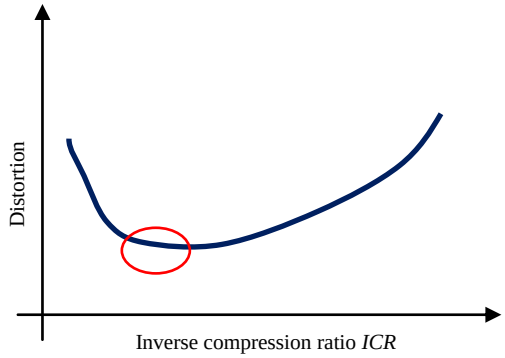


Figure 5.30: *CEM* curve, hypothetical relationship between the error map *E* and the compressed-decompressed image *Y*.

Figure 5.31 shows the different *CEM* curves for the image in Figure 5.25 (a).

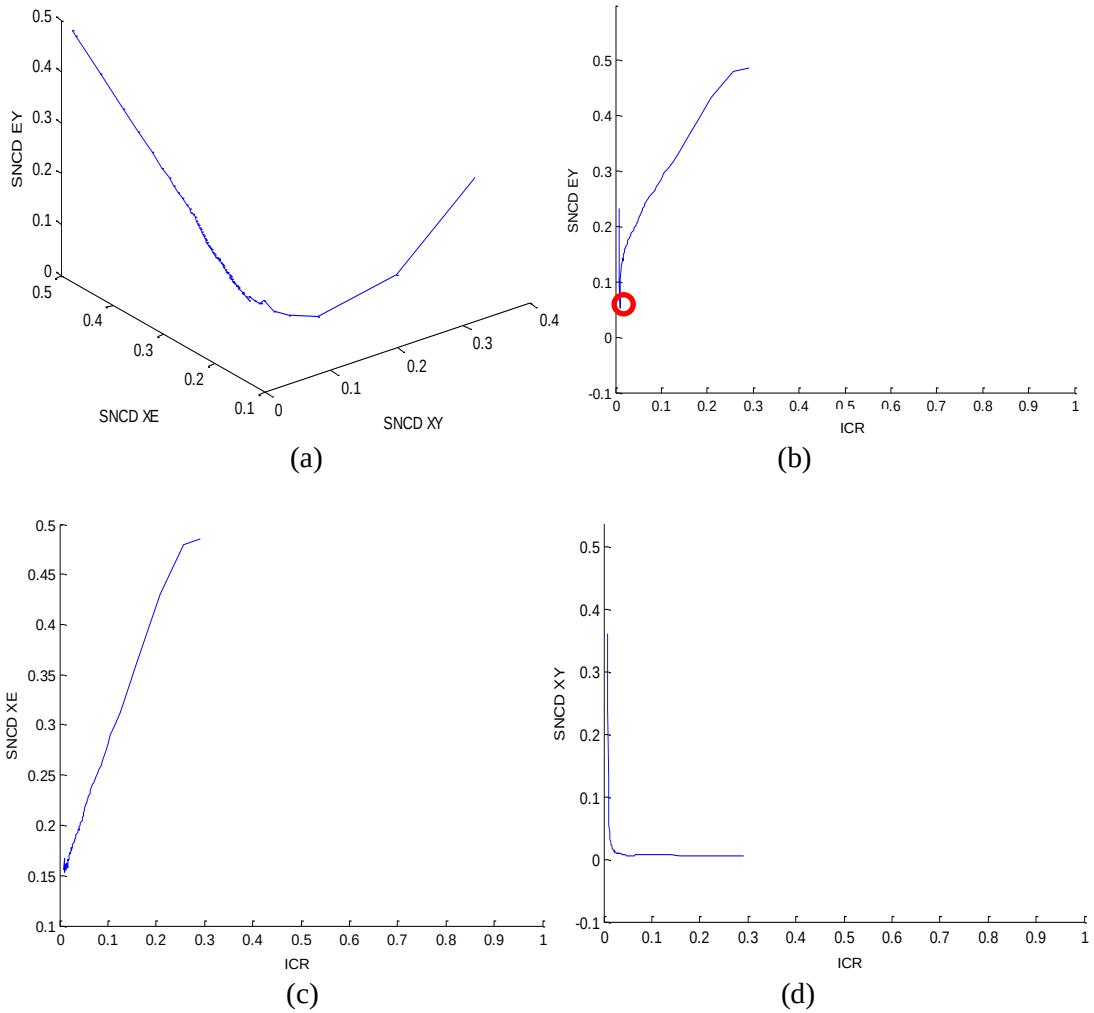


Figure 5.31: *CEM* curves: (a) curve in three dimensions, (b) relationship between the error map *E* and the compressed-decompressed image *Y*, (c) relationship between the original image *X* and the error map *E* (d) relationship between the original image *X* and the compressed-decompressed image *Y*.

In Figure 5.31 (a), we can see the three-dimensional curve (*SNCD(X,Y)* vs. *SNCD(X,E)* vs. *SNCD(E,Y)*) which has a distinct turning point supposedly due to the behavior of the compressor and the image content. In (b) we can easily identify the minimum between the error map *E* and the compressed-decompressed image *Y*. The curves (b), (c) and (d) present what was predicted.

Now we want to investigate whether the content of an image affects these curves. Figure 5.32 shows a small database of images that was taken in order to verify this dependence. The results are shown in Figure 5.33 and corroborate our assumption.

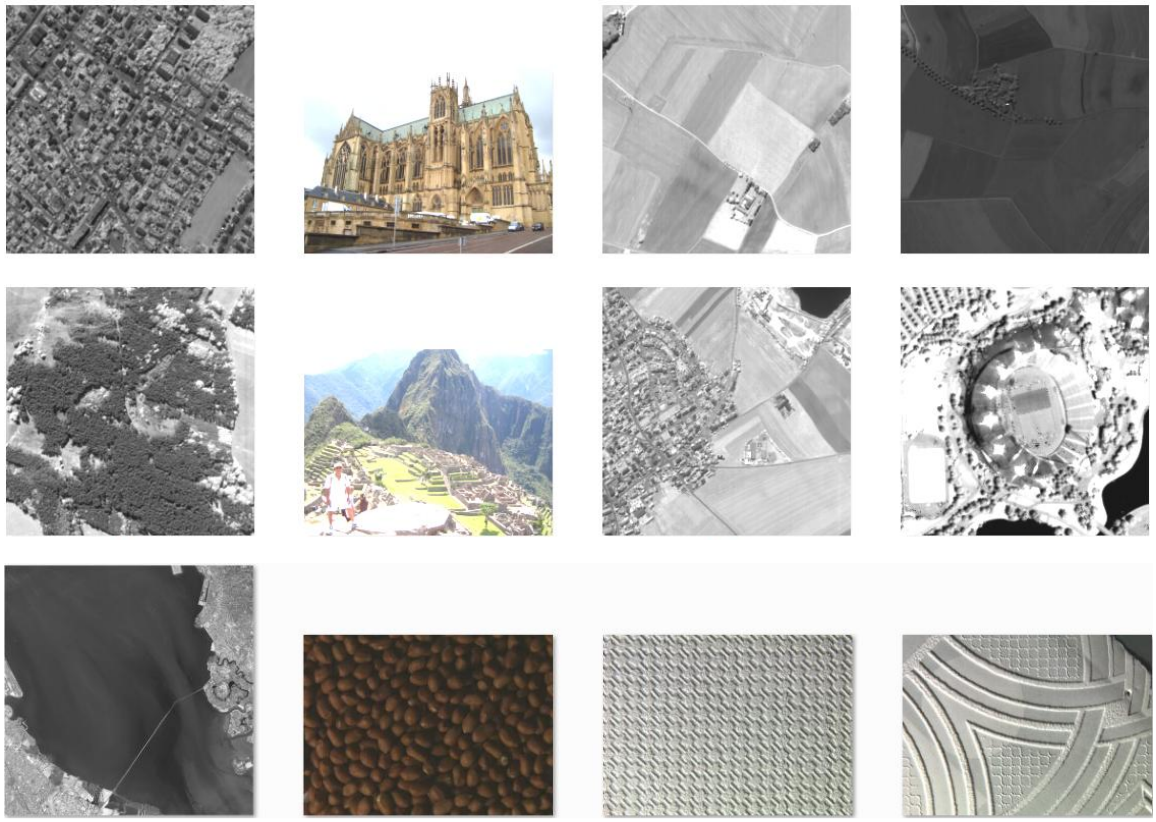


Figure 5.32: Small image database containing Earth observation and media images as well as textures.

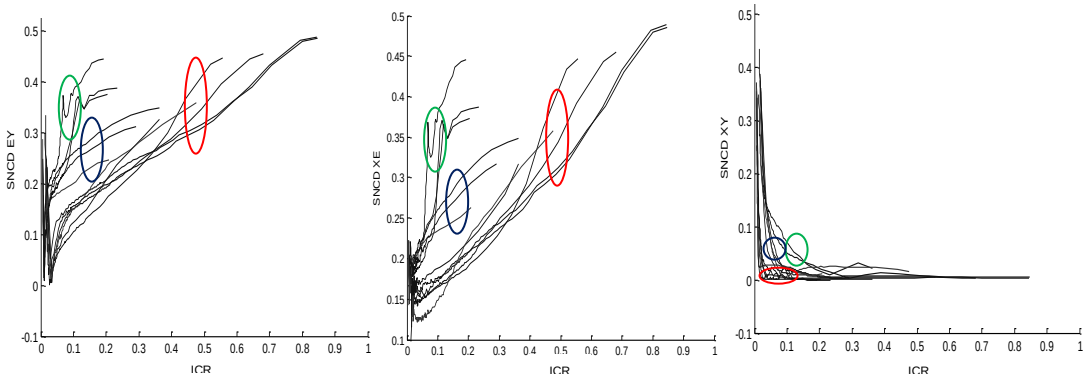
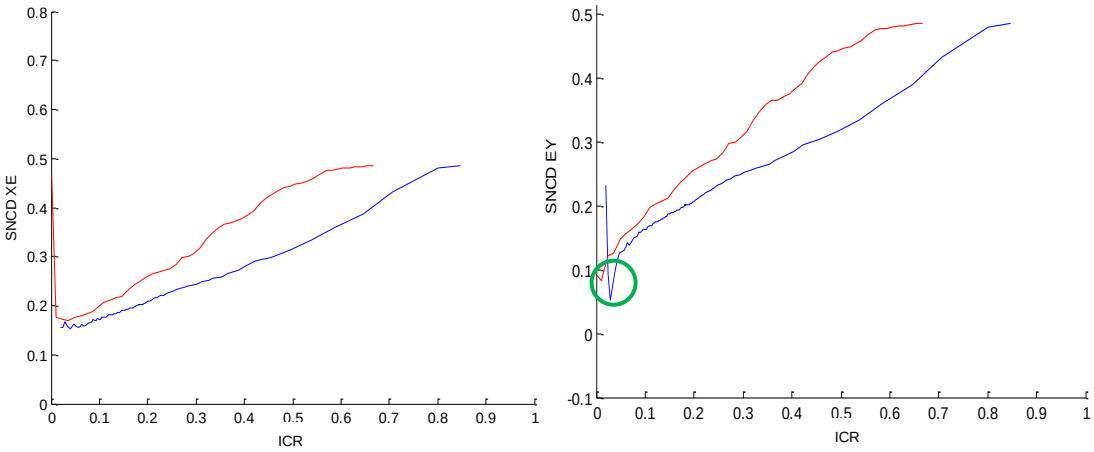


Figure 5.33: *CEM* curves of the images in Figure 5.32.

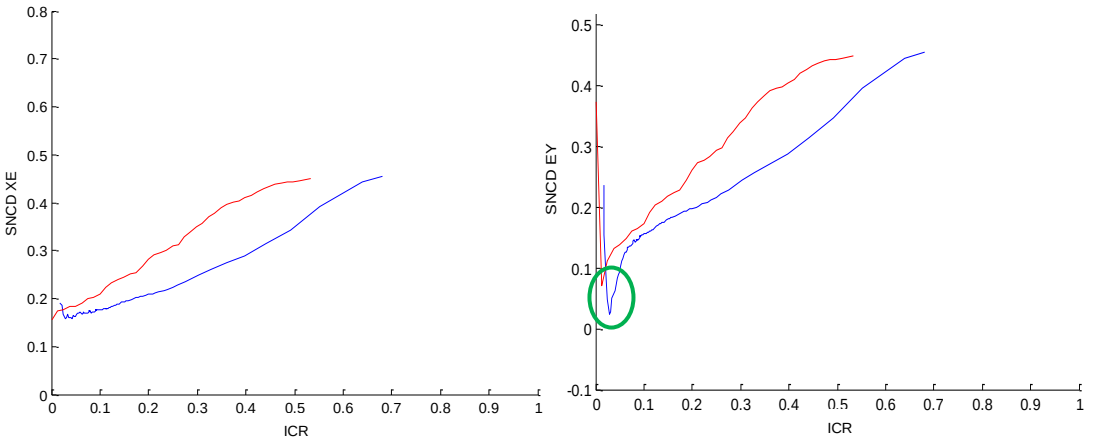
As shown in Figure 5.33, the *CEM* curves can discriminate between different types of images; in this particular case, we can distinguish three groups of curves consisting of a set of Earth observation images, a second group of multimedia images, and a third group for textures. All these curves of Figure 5.33 were calculated using the baseline JPEG and show different shapes. The curves marked in green correspond to more complex images, the curves marked in blue refer to texture images, and the final group marked in red represents satellite images.

To analyze the behavior of the baseline JPEG lossy compressor and the JPEG 2000 lossy compressor, we plot both *CEM* curves with both compressors on a single image, as shown in Figure 5.34 and Figure 5.35 referring to the images in Figure 5.25 (a) and (d).



(a) (b)

Figure 5.34: CEM curves for the image in Figure 5.21 (a); the red curve stands for the JPEG 2000 compressor, the blue curve stands for the baseline JPEG compressor.



(a) (b)

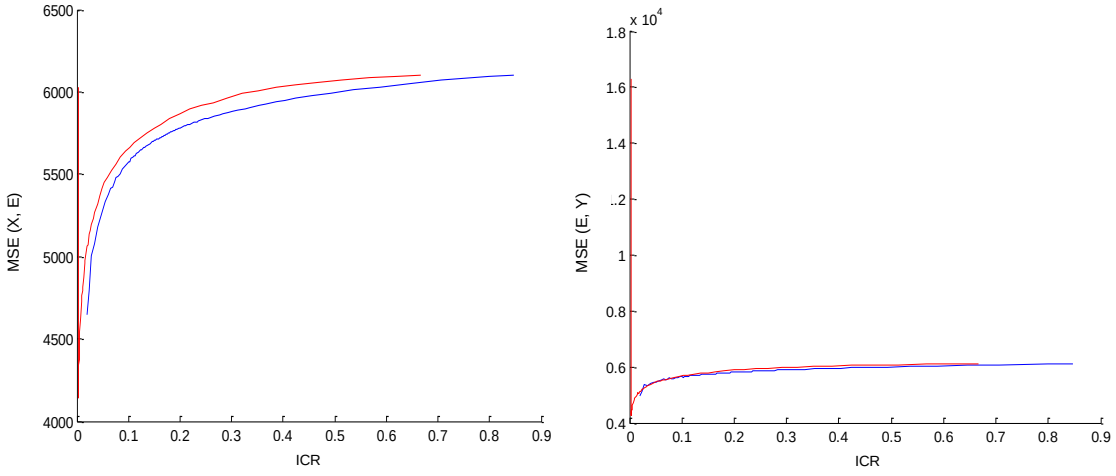
Figure 5.35: CEM curves for the image in Figure 5.25 (d); the red curve stands for the JPEG 2000 compressor, the blue curve stands for the baseline JPEG compressor.

In these curves (with the red curve corresponding to the JPEG 2000 compressor and the blue curve to the baseline JPEG compressor), we can see that

- the JPEG 2000 compressor has a better performance in terms of compression, all points that represent good quality are to the left of the points obtained by baseline JPEG compressor.
- the comparison between the original image *X* and the error map *E* gives smaller *SNCD* values for the baseline JPEG compressor than for the JPEG 2000 compressor. This means that the error map *E* is closer to the original image *X*, so for all compression ratios, the transfer of information or image content to the error map is better reflected by the baseline JPEG compressor; thus, this compressor is better suited to analyze residues during artifact detection. In contrast, the JPEG 2000 compressor transfers less information to the error map because it has a better overall performance.

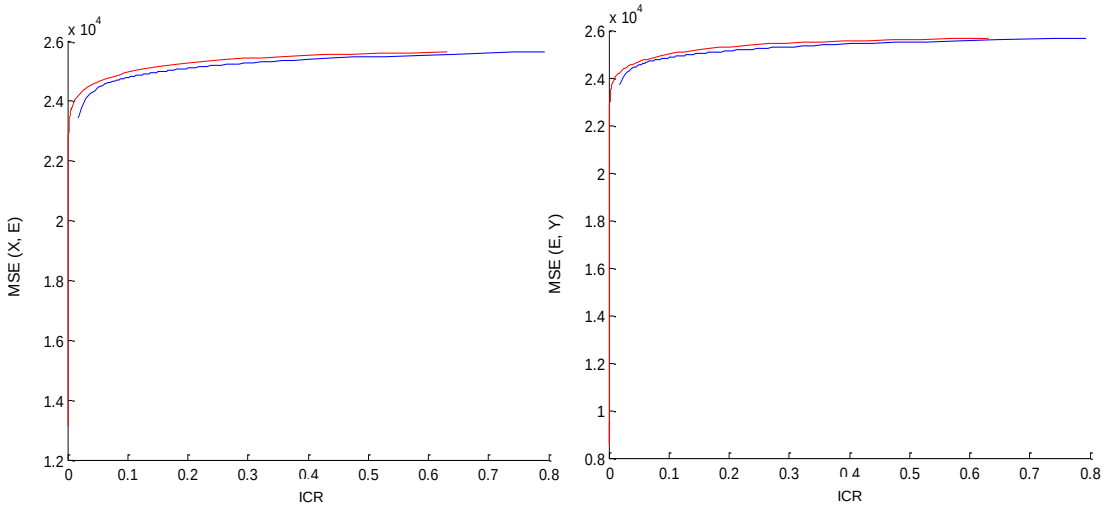
Also, we can see the minimum in the curve when comparing the error map *E* with the compressed-decompressed image *Y*, although the shape of the curve is somewhat different from Figure 5.30. The baseline JPEG method produces a minimum level being lower than that of the JPEG 2000 compressor, suggesting once again that the former produces more image content related errors. On the other hand, the minimum point produced by the JPEG 2000 compressor is shifted to the left, indicating that the performance of this compressor is better.

In order to validate the experiment and the conclusions, we repeated the experiment by re-calculating the Rate-Distortion values using the well-known Mean Squared Error (*MSE*) quality metric; in Figures 5.36 and 5.37 we can see the curves for the images of Figures 5.25 (a) and (d).



(a) (b)

Figure 5.36: Rate-Distortion curves for the image in Figure 5.25 (a), where the red curve stands for the JPEG 2000 compressor, and the blue curve stands for the baseline JPEG compressor.



(a) (b)

Figure 5.37: Rate-Distortion curves for the image in Figure 5.25 (d), where the red curve stands for the JPEG 2000 compressor, and the blue curve stands for the baseline JPEG compressor.

We obtain the same conclusion again: the red curve which corresponds to the JPEG 2000 compressor is always greater than the blue curve which corresponds to the baseline JPEG compressor, since the performance of the JPEG 2000 compressor is better than that of the baseline JPEG compressor. The visual differences between both curves in Figures 5.36 and 5.37 are smaller because the *RD* values are rather high.

5.3.1.2 The Kolmogorov Structure Function

The Kolmogorov Structure Function (*KSF)* is non-computable since the underlying Kolmogorov complexity is also a non-computable function; that is the reason why we use the compression factor as an approximation to complexity. The idea is to use the Kolmogorov structure function as an approximation to the rate-distortion analysis in order to describe the behavior of potential artifacts and to develop a parameter free artifact detection method.

In this sub-section, we are going to present another experiment and its results in order to evaluate the behavior of the *KSF* for different textures, and to check whether we can discriminate them and whether we can use *KSF* for artifact detection.

The first experiment using *KSF* is to apply it to texture discrimination. To evaluate the *KSF* behavior for different textures, we use the Brodatz image databases (USC-SIPI). In Figure 5.38, we show the textures used for this experiment, and in Figure 5.39 we present the results using *KSF*.

We can observe in Figure 5.39 that the *KSF* can discriminate the different structures partially; for all texture groups the *KSF* curve has a similar shape, but their fall-off is different.

The second experiment is to apply *KSF* for artifact detection; here, the first step is to study the behavior of the *KSF* curves for images with or without artifacts.

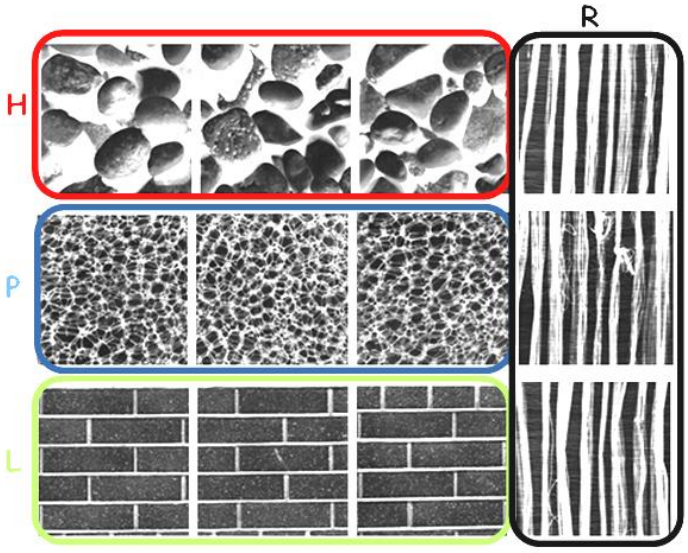


Figure 5.38: Textures used for the experiment represented by the letters H, P, L, and R.

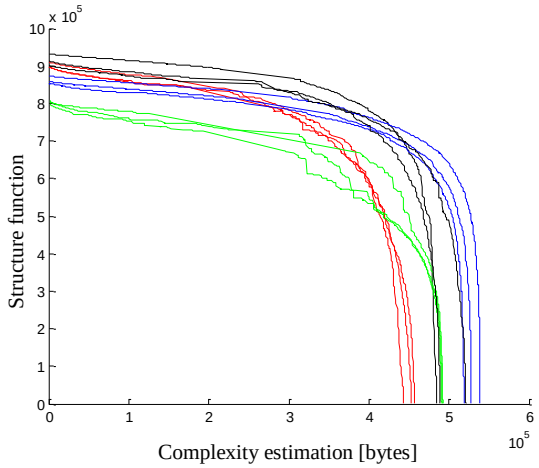
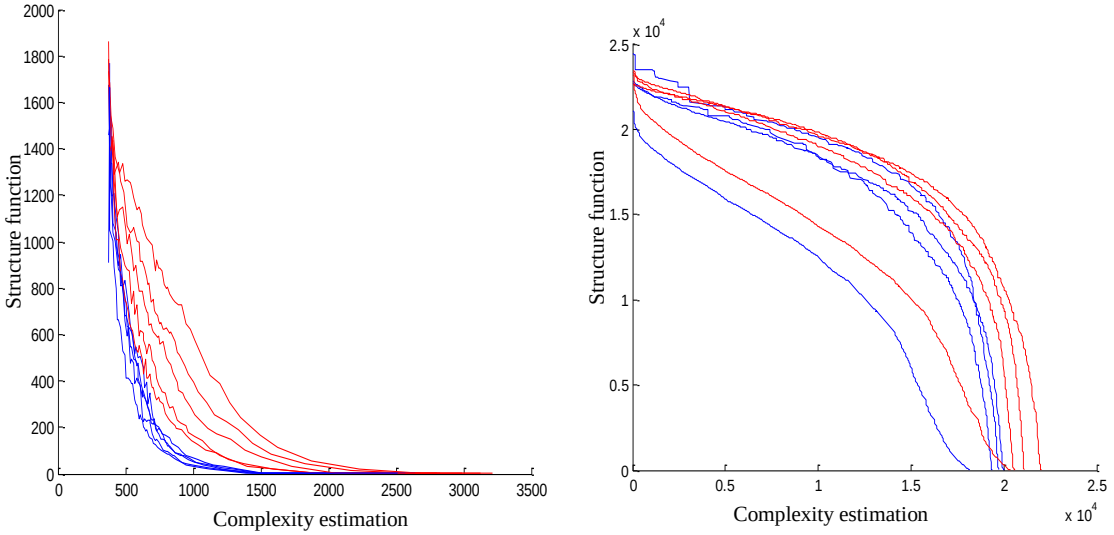


Figure 5.39: Kolmogorov Structure Functions for the textures presented in Figure 5.38; the red curves refer to the texture *H*; the blue curves refer to the texture *P*, the green curves refer to the texture *L*, and the black curves refer to the texture *R*.

One aspect to consider is how to generate the candidates for the necessary space *S* as indicated in Section 4.5. For this purpose, we have generated candidates by using two methods: candidate generation by baseline JPEG lossy compression and candidate generation by a genetic algorithm as in (De Rooj & Vitanyi 2011).

In Figure 5.40, we show the *KSF* curve results for different patches of a satellite image containing urban land cover with aliasing in some parts. In Figure 5.40(a) we use baseline JPEG lossy compression, and in Figure 5.40(b) we use a genetic algorithm. The red curves represent patches with artifacts while the blue curves represent patches without artifacts.



(a) *KSF* using baseline JPEG lossy compression (b) *KSF* using a genetic algorithm

Figure 5.40: Kolmogorov Structure Function: the abscissa shows the compressed size of the images (as an approximation to Kolmogorov Complexity) and the ordinate represents the structure function. The red curves stand for patches with artifacts, while the blue curves represent patches without artifacts.

We can observe that a better discrimination can be reached when we generate the candidates for the space *S* using baseline JPEG lossy compression. In this case, the approximation to the Rate-distortion function is better, too.

Considering this, we used the baseline JPEG lossy compression for candidate generation and the computation of the Kolmogorov Structure Function for each patch of a satellite image to detect potential artifacts. For this experiment we used an image with manually introduced aliasing. In Figure 5.41 we can see the result for aliasing detection.

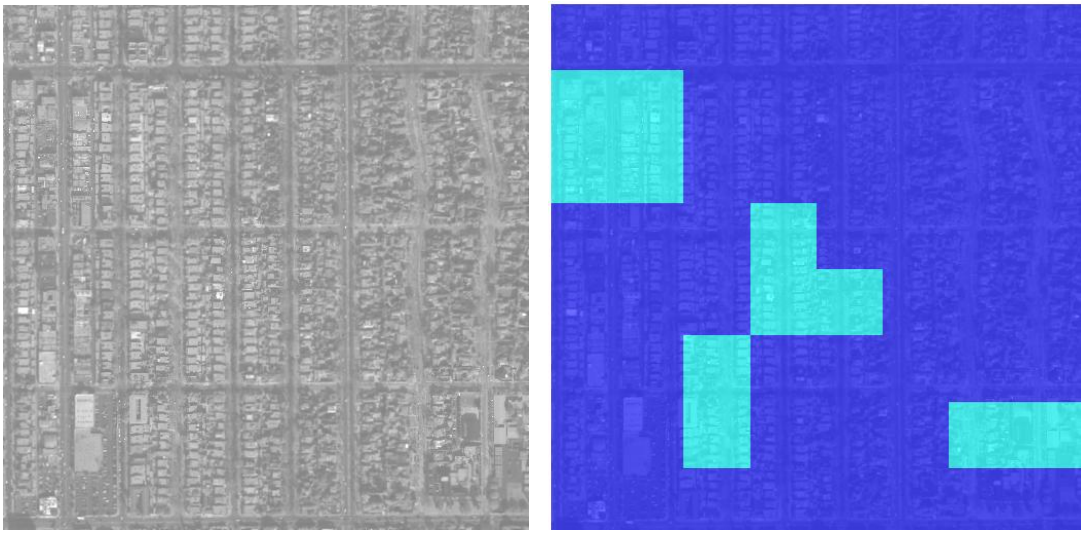
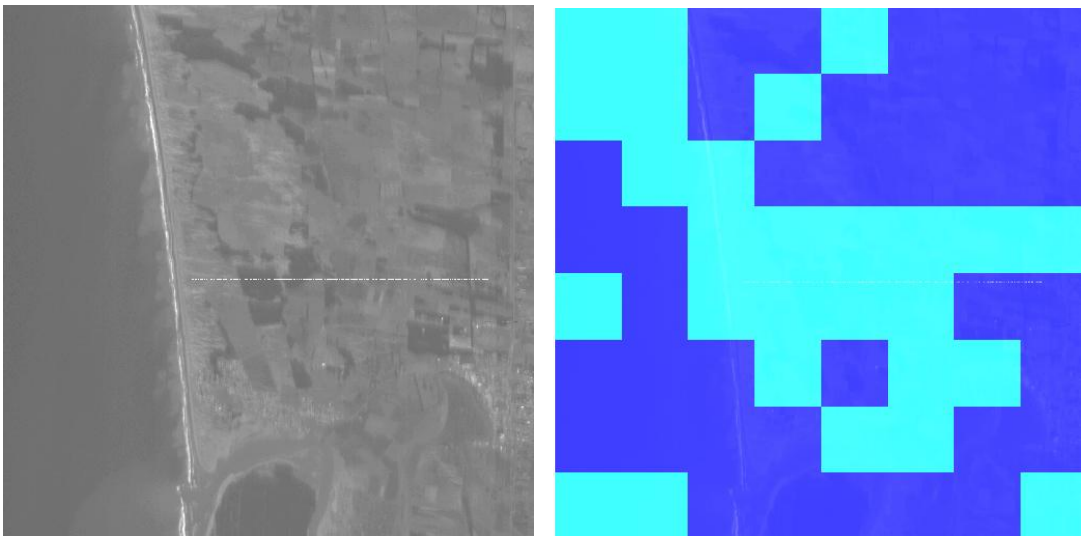


Figure 5.41: Aliasing detection for urban land cover using *KSF* and candidate generation with baseline JPEG lossy compression.

The artifact detection presented in Figure 5.41 was made correctly; the satellite image is an image of urban land cover with manually introduced synthetic aliasing artifacts. For further details how to simulate aliasing in images, see Chapter 6.

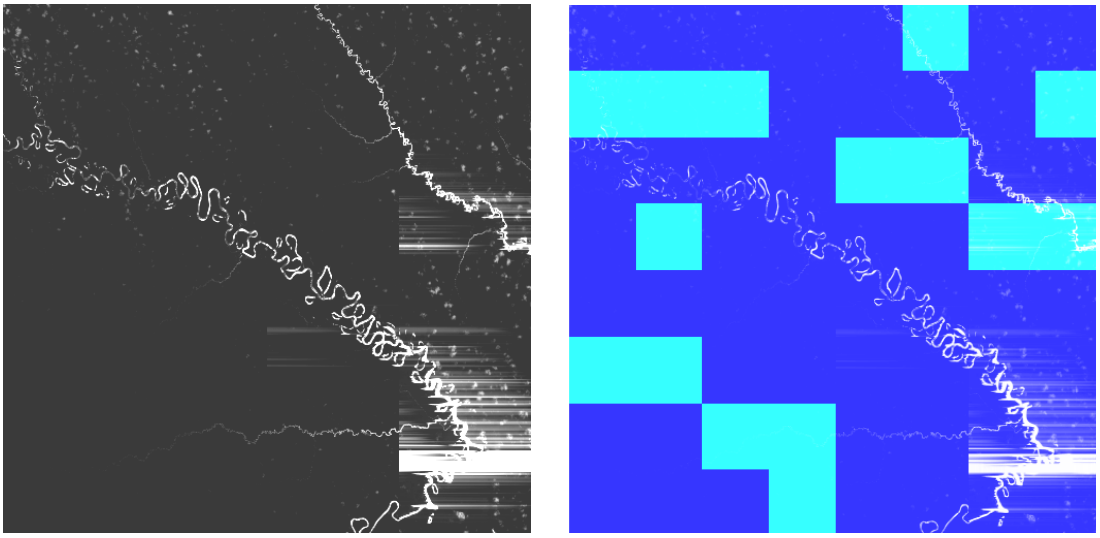
Another example of the application of the method based on *KSF* is presented in Figure 5.42; we have again the image where a stuck A/D converter bit appears.



(a) (b)

Figure 5.42: Results of stuck bit detection using the *KSF* method.

A third example for applying the proposed *KSF* method is shown in Figure 5.43 where we have an image with trailing charge problems.



(a) (b)

Figure 5.43: Results of trailing charge detection using the *KSF* method.

5.3.2 Artifact Detection with *CEM*

For artifact detection, we propose to use the *CEM* function obtained by compression of the image with different compression parameters and a multidimensional analysis of the similarities between *X*, *Y*, and *E*. The *CEM* analysis is made as shown in the block diagram of Figure 5.44. We will also use the *Rate-Complexity* (*CV – complexity variation*) curve to make the same analysis, as shown by the block diagram in Figure 5.45.

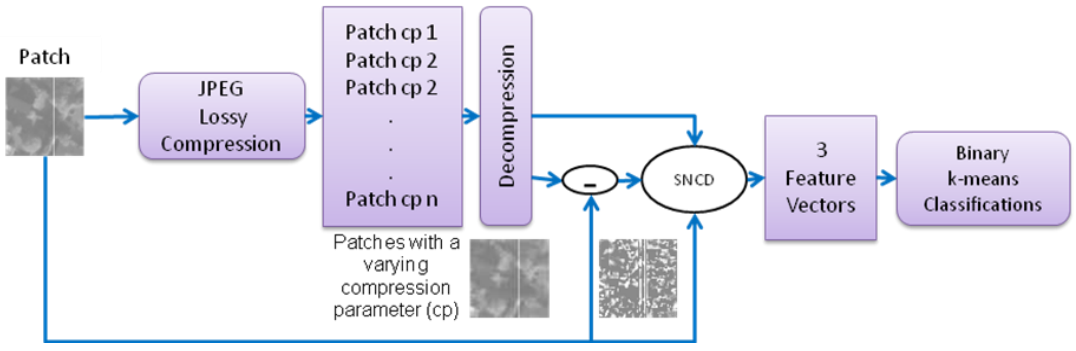


Figure 5.44: Block Diagram for *CEM* analysis: we take the image under test; we compress image patches with varying compression parameters, then decompress the patches and calculate the similarity using the *SNCD* for each compression parameter. Based on these values we compose a feature vector and then apply a classification method.

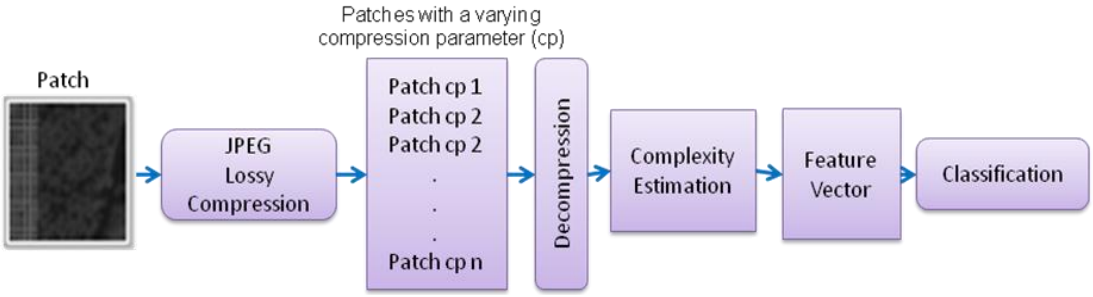


Figure 5.45: Block diagram for *Rate-Complexity* analysis: we take the image under test; we compress image patches with varying compression parameters, then decompress the patches and estimate the complexity using a lossless compressor for each compression parameter. Based on these values, we compose a feature vector and then apply a classification.

In both cases, we take the image under test I, we cut again the image I into n different patches X_i of 64×64 pixels. For each patch X_i, we compress it with a varying quality factor q using a lossy compression. After that, we decompress the patches and we obtain decompressed patches Y_{iq}.

For the method shown in Figure 5.44, the next step is to calculate the similarity between the original patches X_i and the compressed-decompressed patches Y_{iq} using the *SNCD* metrics; thus, we obtain feature vector elements $F_{iq} = SNCD(X_i, Y_{iq})$.

In contrast, for the method shown in Figure 5.45, the corresponding step is to estimate the complexity of Y_{iq} using a lossless compressor, and we obtain a $F_{iq} = ICR(Y_{iq})$.

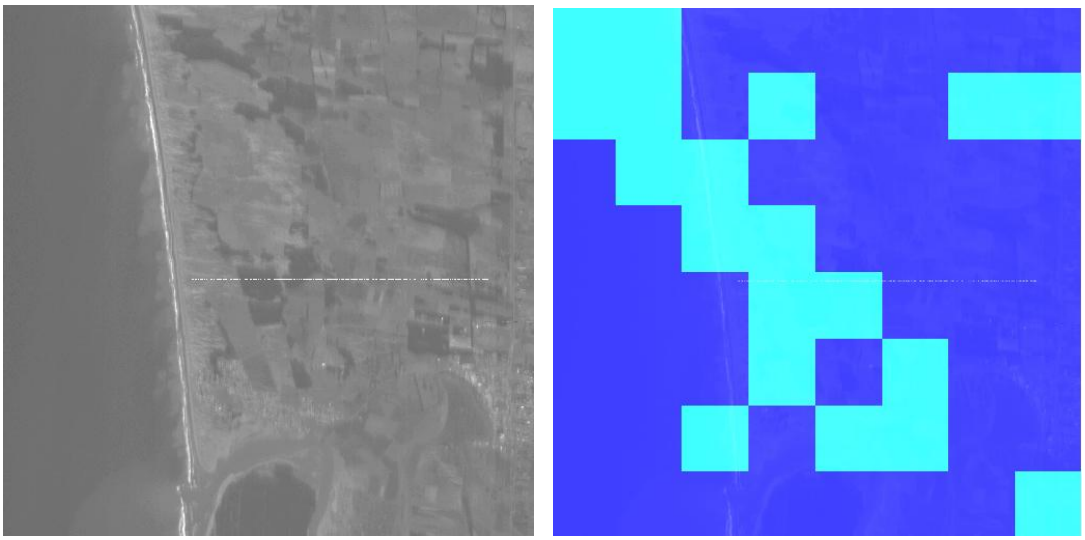
For both cases, based on the F_{iq} values, we compose a feature vector $V_i = [F_{i1}, F_{i2}, \ldots F_{iq} \ldots F_{iQ}]$ and we obtain the matrix:

$$V = \begin{bmatrix} F_{11} & F_{12} & \ldots & F_{1q} & \ldots & F_{1Q} \\ F_{21} & F_{22} & \ldots & F_{2q} & \ldots & F_{2Q} \\ . & . & \ldots & . & \ldots & . \\ F_{i1} & F_{i2} & \ldots & F_{iq} & \ldots & F_{iQ} \\ . & . & \ldots & . & \ldots & . \\ F_{n1} & F_{n2} & \ldots & F_{nq} & \ldots & F_{nQ} \end{bmatrix}.$$

Finally, with this matrix V, we perform a non-supervised *K-MEANS* classification. Again, we take a value of $K = 2$.

For the results shown in Figure 5.46, we used the baseline JPEG lossy compressor with a quality factor variation between *0* and *100* resulting in 101 different settings ($Q = 101$), and the JPEG-LS compressor for lossless compressions.

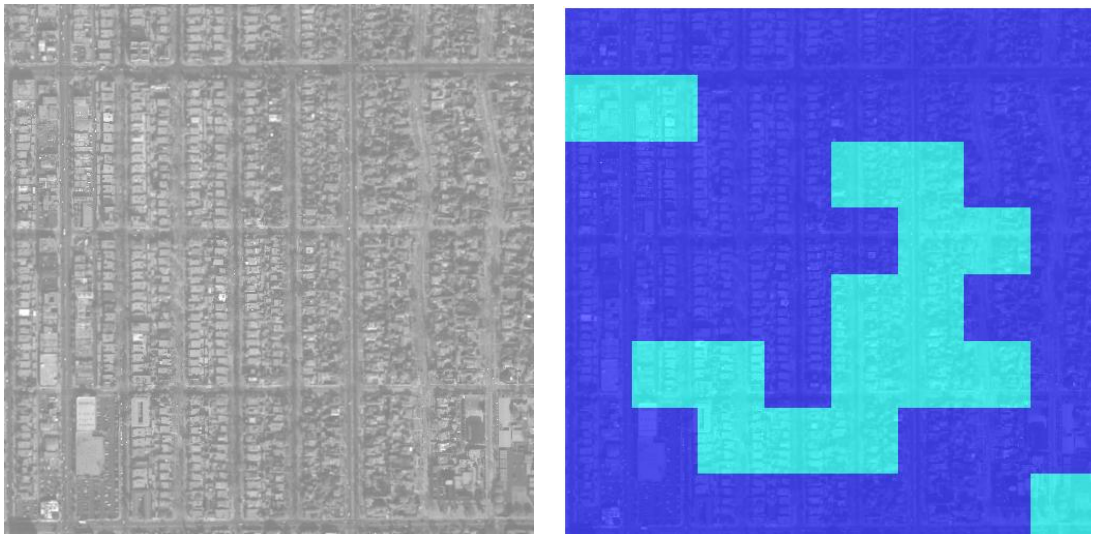
When we try to detect artifacts using *CEM*, we can see the result for A/D stuck bit detection in Figure 5.46. This result will be discussed in Chapter 6.



(a) (b)

Figure 5.46: Results of stuck bit detection using the *CEM* method.

Another example for applying the proposed *CEM* method is shown in Figure 5.47 where we have again our *city* image with simulated aliasing.



(a) Image with aliasing (b) Aliasing detection

Figure 5.47: (a) Satellite image with artificial aliasing. (b) Results of patch-wise aliasing detection using *CEM* analysis.

A third example for applying the proposed *CEM* method is shown in Figure 5.48 where we have the image with trailing charge problems.

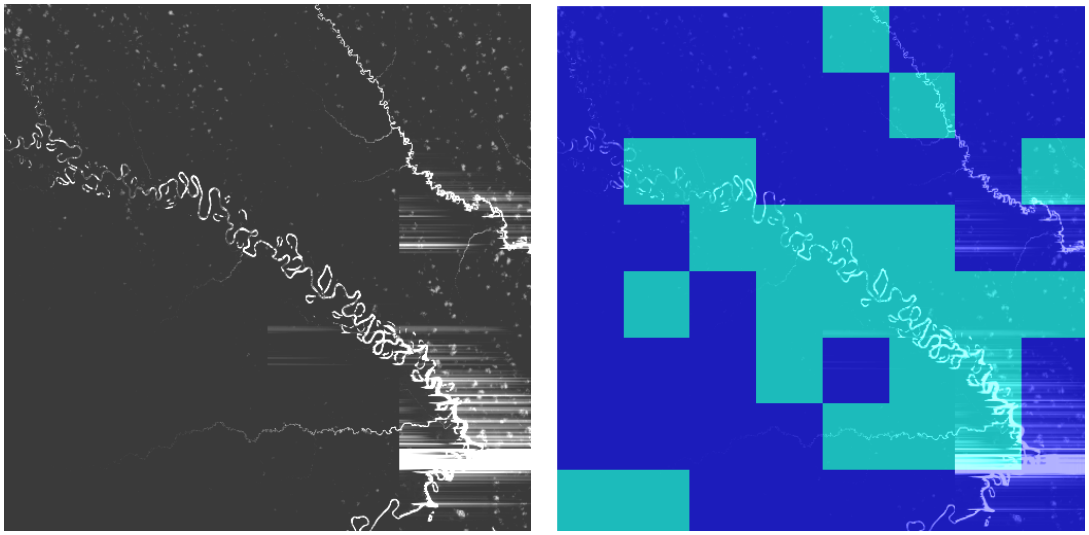


Figure 5.48: Results of trailing charge detection using the *CEM* approach.

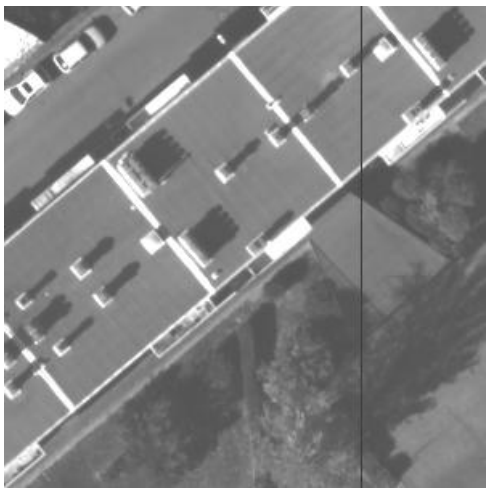
5.4 Artifact Detection Using Image Quality Metrics

In this section we will describe an existing reference method for artifact detection in optical satellite images that follows a steganalysis approach by (Avcibas et al. 2003); the method is based on the analysis of the error produced by a Gaussian filter; this error is analyzed using different quality metrics described in Section 3.3.1. The objective is to evaluate the behavior of the error between the original image and the filtered image produced by a Gaussian filter and analyzed with different quality metrics. We first present a description of the principle. Then we will make a description of the process for artifact detection; the analysis will take place in the error or distortion map of the image. Finally, we apply this method to some examples to show their effectiveness in detecting artifacts.

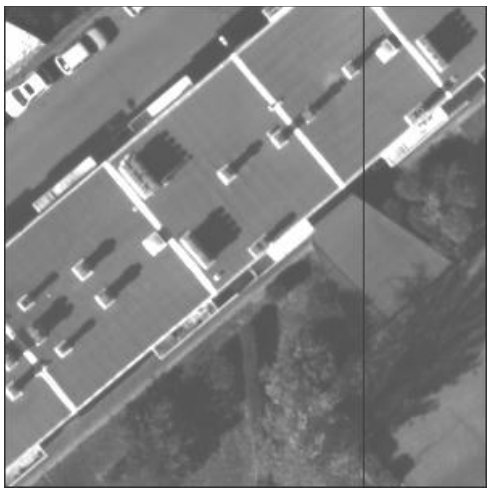
5.4.1 Empirical Analysis of Quality Metrics for Images with Artifacts

The Quality Metrics (*QM*) are the equations that allow us to compare and evaluate the quality of an image using a full-reference approach, meaning that we need access to the original or reference images.

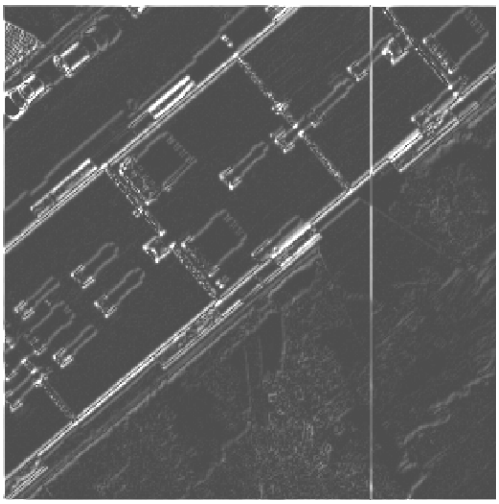
In order to detect the presence of artifacts in satellite images, we make a small experiment. In Figure 5.49, we show two images; Figure 5.49 (a) presents an airborne image with a dead column; Figure 5.49 (b) shows the filtered image after using a Gaussian filter; Figure 5.49 (c) presents the error between images (a) and (b); the error was computed as the absolute value of the difference between images (a) and (b), $E = abs\ (a - b)$; Figure 5.49 (d) shows the results of the *QM* values based on Equations (3.1), (3.2), (3.3), (3.5), (3.6) and (3.7) shown in Chapter 3.



(a) Original image



(b) Filtered image



(c) Error map

$M_1 = 3.1687$
$M_2 = 11.2510$
$M_3 = 0.7515$
$M_5 = 0.5023$
$M_6 = 0.9885$
$M_7 = 1342769.9387$

(d) Selected quality metrics

Figure 5.49: Empirical example of *QM* for artifact detection. (a) shows the original image containing a dead column; (b) shows the filtered image of (a) after using a Gaussian filter; (c) shows the absolute error between the original and the filtered image; (d) shows the quality metrics results.

Figure 5.49 (c) shows the absolute error obtained after comparing the original image with the filtered image; we can see that the principal variations are located along the edges of the structures; the *QM* values are listed in Figure 5.49 (d) and should characterize the error behavior. Therefore, the idea is to analyze the presence of artifacts in the error map produced by taking the difference between the original and the filtered image.

5.4.2 Artifact Detection by Quality Metrics

The approach for artifact detection based on image quality is a method related to the work presented in (Avcibas et al. 2003) where the authors present a technique for steganalysis using well-known image quality metrics. We use this steganalysis method because the problem to detect some hidden information in images is very similar to artifact detection; hidden information changes some statistical parameters in the images and the artifacts produce some changes in the images, too; the quality of the image is affected by the presence of hidden information or the presence of artifacts; since image quality is affected, we can use the image quality metrics to evaluate the presence of hidden information or the presence of artifacts and to detect them. We change the image quality by applying a Gaussian filter, and we evaluate how the behavior of artifacts is affected by filtering. This method based on image quality metrics has a basic concept being similar to *RD* analysis; in both cases, we analyze our results in the error domain.

The process that we propose for artifact detection is to cut the satellite image I again into n patches X_i of 64×64 pixels. For each patch X_i we apply a Gaussian filter and obtain a filtered patch Y_i; after that, we compare the original patch X_i to the patch after applying a Gaussian filter Y_i with different image quality metrics. The Gaussian filter was chosen as:

$$H(m,w) = K \times g(m,w) \tag{5.3}$$

where $g(m,w)$ is the 2-D Gaussian kernel and the normalizing constant is defined by:

$$K = \left(\sum_m \sum_w |g(m,w)|^2\right)^{-1/2}. \tag{5.4}$$

The width of the Gaussian filter was set to $\sigma = 0.5$ with a mask size of 3×3 pixels. The image quality metrics that we used for comparison are:

- the Mean Absolute Error (M_1)
- Mean Squared Error (M_2)
- Czekanowski Distance (M_3)
- Image Fidelity (M_5)
- Normalized Cross-Correlation (M_6)
- Spectral Magnitude Distortion (M_7)

With the values resulting from the comparison, we create a feature vector

$$V_i = [M_{i1}, M_{i2}, M_{i3}, M_{i5}, M_{i6}, M_{i7}]$$

where

$M_{i1} = M_1(X_i, Y_i)$ $\quad$ $M_{i2} = M_2(X_i, Y_i)$
$M_{i3} = M_3(X_i, Y_i)$ $\quad$ $M_{i5} = M_5(X_i, Y_i)$
$M_{i6} = M_6(X_i, Y_i)$ $\quad$ $M_{i7} = M_7(X_i, Y_i)$.

Then, we obtain the matrix:

$$V = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{15} & M_{16} & M_{17} \\ M_{21} & M_{22} & \dots & \dots & \dots & M_{27} \\ . & . & \dots & . & \dots & . \\ M_{i1} & M_{i2} & \dots & \dots & \dots & M_{i7} \\ . & . & \dots & . & \dots & . \\ M_{n1} & M_{n2} & \dots & \dots & \dots & M_{n7} \end{bmatrix}.$$

Finally, with this matrix *V*, we apply a non-supervised classification method; we use the *K-MEANS* classification method and we take a value of $K = 2$ that represents one group for images with artifacts and other group for images without artifacts.

Figure 5.50 depicts the block diagram for artifact detection using image quality metrics. The complete process was already described above.

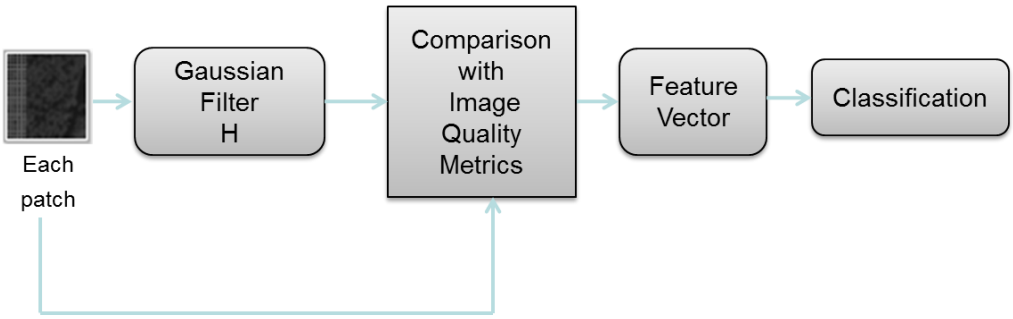
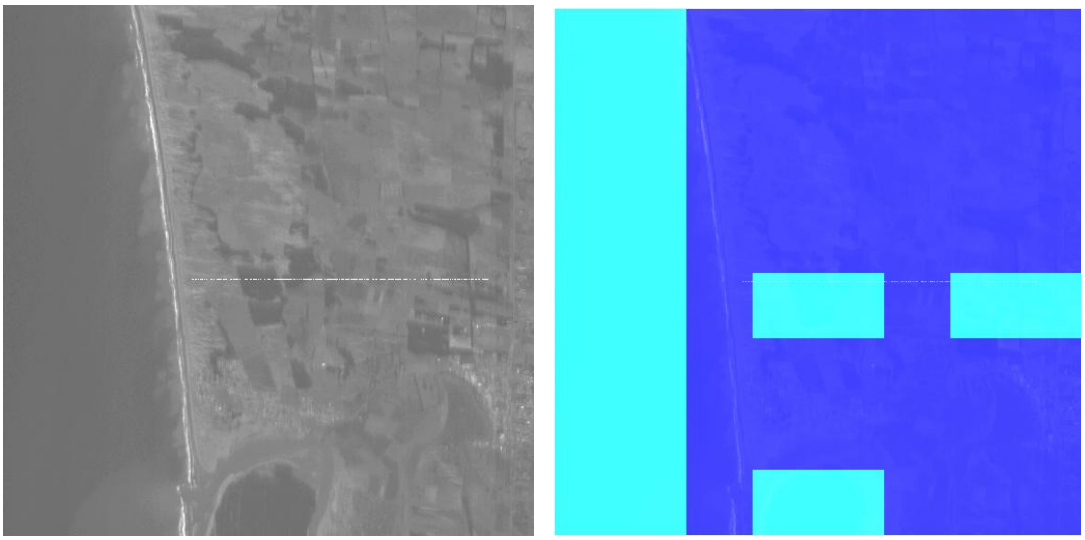


Figure 5.50: Artifact detection using image quality metrics.

5.4.3 Typical Examples

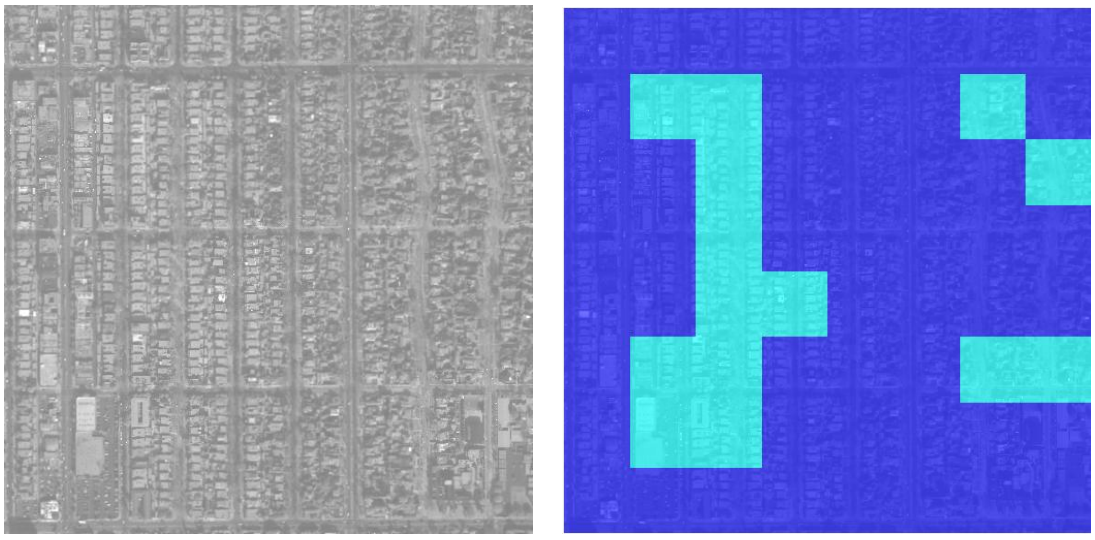
An illustrative example of this method is shown in Figure 5.51; in this case, we have a stuck bit (A/D conversion problem).



(a) (b)

Figure 5.51: Results of stuck bit detection using the *QM* method.

Another example for applying this method based on quality metrics is shown in Figure 5.52 where we have again our *city* image with simulated aliasing.



(a) Image with aliasing (b) Results of aliasing detection

Figure 5.52: (a) Satellite image with artificial aliasing. (b) Results of patch-wise aliasing detection using *QM* analysis.

A third example is shown in Figure 5.53; in this case, we have a SPOT image containing actual sensor artifacts.

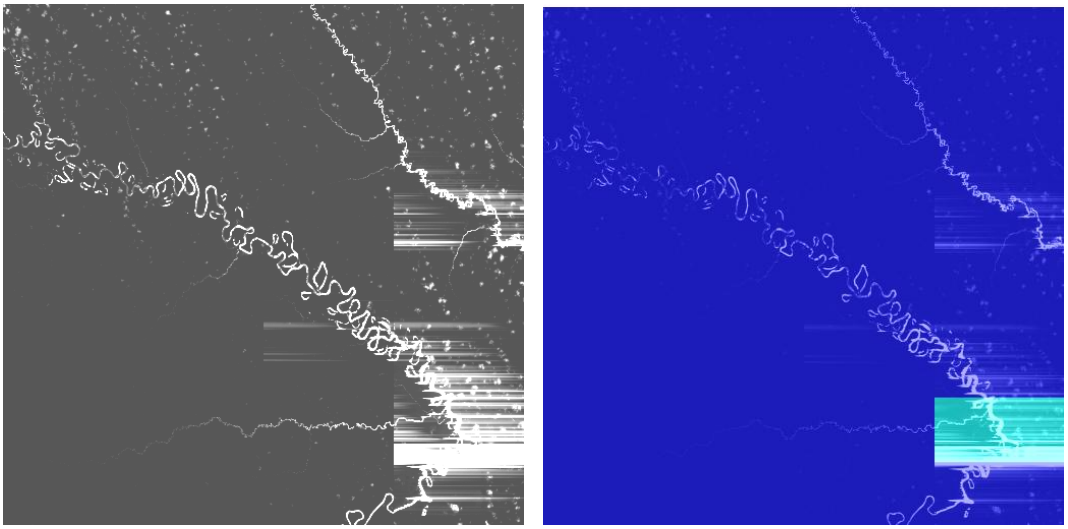


Figure 5.53: Artifact detection using image quality metrics; in this case, we have a SPOT image containing actual sensor artifacts.

We can conclude that the detection was made with an image with real instrumental artifacts; the strongest artifacts with high intensity were detected, while the weaker ones remained undetected. However, we can use this approach for first order artifact detection.

Chapter 6

Analysis of Results and Quality Metrics Applications

This chapter serves several purposes:

- It shall document the results obtained during all test runs when applying the four selected methods of the preceding chapter. In order to have a uniform basis for comparisons, we provide for each set of results
  - a textual description with all test details
  - tables containing the most important quantitative performance results
  - graphical illustrations where we compare different cases and options.

All this is contained in Section 6.2.

- It also shall describe our testing approach where we had to
  - provide and select test images in a database (see Section 6.1)
  - select and insert typical artifacts into images
  - modify the strength of the artifacts
  - learn about the success rates of detecting an artifact
- Finally, we need conclusions for artifact detection (see Section 6.3) and the use of normalized compression distance as a quality metric, especially when we analyze residual errors (see Section 6.4).

Our rationale when organizing the test cases was:

- We needed a small but representative database with sample images. We selected three typical Earth surface images (a city, forest and sea) and inserted synthetic artifacts into these images
- As typical artifacts we selected striping and aliasing. These two cases cover artifacts occurring in instrument electronics as well as artifacts due to on-ground image processing
- Each type of artifacts was applied with different strengths: at least three sub-sampling options for aliasing, and at least three intensity levels for strips were applied

Due to the fact that some of the obtained results seem to be unclear and even contradictory at a first glance, we advise the reader to follow a simple guideline for each test description:

- If a test run is based on rate-distortion analysis (our first method), the results are easily understandable and do not need extra explanations. Therefore, we do not provide additional support for these cases and we can recommend the method due to its simplicity and performance.
- In contrast, if a test run uses *NCD* (as suggested in our second method), we sometimes need additional explanations. These explanations will be provided for the individual test runs and do not represent a formal proof. The main reason seems to be that the artifact patterns are too weak and a clear separation between artifact-free and artifact-affected images patches is not feasible.
- Surprising results can be found with our third method, i.e. the *CEM* approach: in some cases, we obtain results being equivalent to the best results of the rate-distortion method. On the other hand, we also have a number of more or less good results. The details and the suspected causes will be discussed and explained in the sub-sections below.
- The method based on quality metrics is inferior to our *NCD* based results. The reason seems to be that the common quality metrics have not been designed for weak artifacts as they do not consider typical regularities being contained in image artifacts. For instance, the geometrical properties of a striping artifact cannot be described by global statistics and the applied spectral filters seemingly do not comply with the characteristics of our artifacts.

6.1 Synthetic Database Description

For the creation of the synthetic database, we introduced artificial artifacts into satellite images. For this purpose, we simulated aliasing artifacts and introduced strips into the images. The aim of the synthetic artifacts is to control the intensity of the artifacts, to control the influence of the artifacts on image processing, classification or indexing, and to evaluate how the methods can detect the artifacts as a function of their intensity.

For simulating aliasing in an image, we need a spatial down-sampling of the image. When we do the down-sampling, aliasing may occur if we violate the Nyquist criterion (or do not perform appropriate interpolation) and if we want to avoid it, we have to apply a low-pass filter with sufficient strength (Pitas 2000) prior to down-sampling. For our purposes, we want to generate test images with sub-areas with and without aliasing artifacts. Hence, we take an image I comprising $n \times m$ pixels and we choose a positive integer ds for controlling the down-sampling; then the resulting image IA after down-sampling is: $IA = I\,[1{:}ds{:}n,\ 1{:}ds{:}m]$, where the notation $1{:}ds{:}n$ means that we take pixels from position 1 to n with a step size of ds. On the other hand, if we want to down-sample without aliasing, we use a low-pass filter prior to down-sampling; we take the image I, we apply the filtering and obtain a filtered image FI. Then we down-sample the image FI and obtain an image without aliasing $IW = FI\,[1{:}ds{:}n,\ 1{:}ds{:}m]$. Finally, we take the two images IA and IW. Then we insert some randomly

selected patches from *IA* into *IW* and obtain an image *TI*−called test image−where aliasing occurs only in some selectable parts of the image.

The process of the test image *TI* generation and aliasing simulation is shown in the scheme of Figure 6.1.

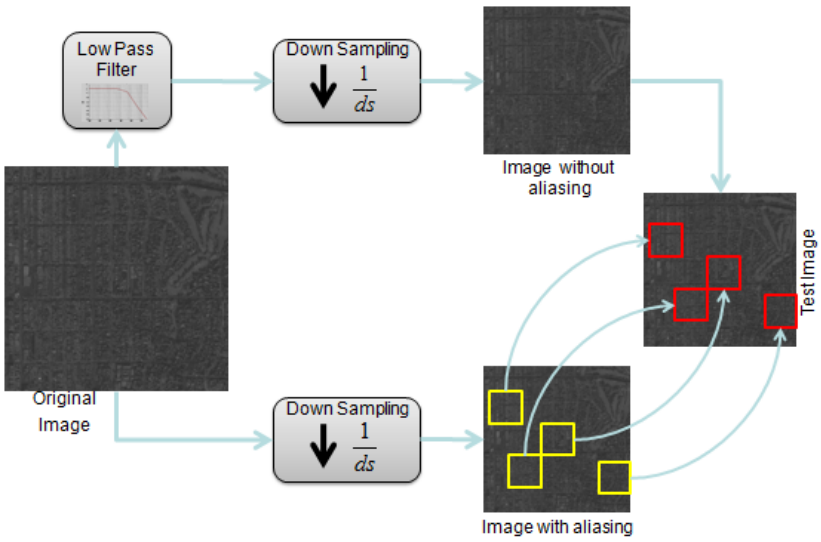


Figure 6.1: Simulation of aliasing: first, we down-sample the image with a selectable step size (e.g., *ds = 3, 5, 7*); during down-sampling, aliasing may occur and, in order to avoid it, we need a low pass filter prior to down-sampling.

Our second simulation process is to introduce strips with varying brightness levels into satellite images. We know that satellite images can be affected by striping that may be due to different reasons. For instance, defective pixels may be "dead", "bleeding", or "hot". In our case, we selected a case that is not too simplistic for artifact detection. We introduced strips by adding them to the grayscale values of the images along some randomly defined columns or lines. The resulting test image *TI* for added strips is obtained by increasing the grayscale value of pixels with selected brightness levels as shown in Figure 6.2, *TI = I(r,s) + k*, for selected positions *(r,s)*.

Figure 6.2: Artifact simulation with various intensities: strip artifacts with different gray levels being added.

Finally, we created a database with different types of synthetic artifacts with varying intensities and for typical Earth surface classes such as *city*, *forest* and *sea*, as shown in the following images of Figure 6.3 in order to evaluate the different methods presented in the previous chapter. These land cover cases were chosen because they are the most used in different applications. The artifacts were inserted into local sub-areas of each image.

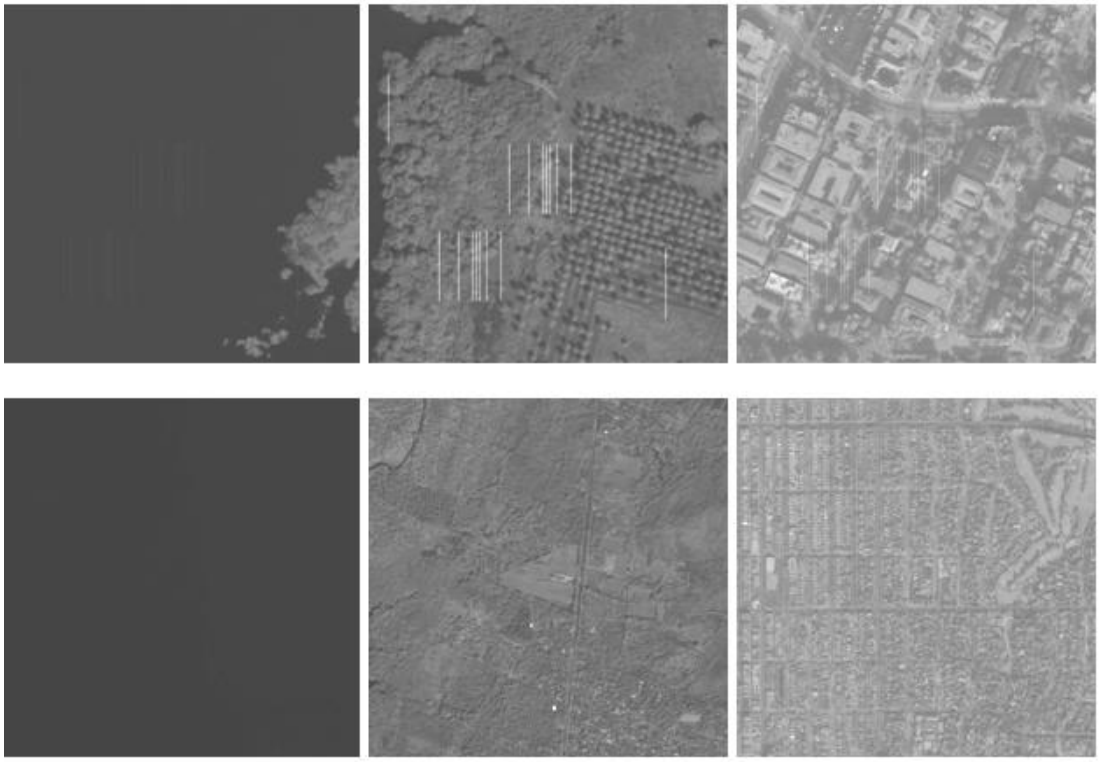


Figure 6.3: Satellite image database with different types of artifacts such as strips, or aliasing with varying intensities and for different Earth surface classes such as *city*, *forest* and *sea*.

6.2 Results

6.2.1 Comparison of *RD* and *NCD* Results

After applying the methods described in Chapter 5 − and detailed in Section 5.1 and 5.2 − to the database we obtained the results shown in Table 6.1. The results are expressed as a percentage of success; this success percentage is calculated from a confusion matrix which classifies two groups: images with artifacts and images without artifacts. The number of elements which are correctly classified divided by the total number of images under analysis multiplied by 100 gives us the success percentage.

SEA (mean value: 19 DN)

Kind of Artifact			**Artifact Detection Method**		
			NCD – baseline JPEG	**NCD – CompLearn/zip**	**Rate-Distortion**
Strips	***k = 1***		64.1 %	50 %	48.4 %
	k = 3		50 %	29.7 %	84.4 %
	k = 6		50 %	39.1 %	89.1 %
	k = 9		62.5 %	34.4 %	89.1 %
	k = 10		62.5 %	34.4 %	84.4 %
Aliasing	***ds = 5***		65.6 %	76.6 %	78.1 %
	ds = 9		68.8 %	79.7 %	70.3 %

FOREST (mean value: 39 DN)

Kind of Artifact			**Artifact Detection Method**		
			NCD – baseline JPEG	**NCD – CompLearn/zip**	**Rate-Distortion**
Strips	***k = 2***		39.1 %	59.4 %	39.1 %
	k = 6		46.9 %	29.7 %	64.1 %
	k = 10		84.4 %	29.7 %	68.8 %
	k = 50		89.1 %	81.3 %	92.2 %
Aliasing	***ds = 5***		40.6 %	37.5 %	65.6 %
	ds = 7		54.7 %	50 %	79.7 %

CITY (mean value: 64 DN)

Kind of Artifact			**Artifact Detection Method**		
			NCD – baseline JPEG	**NCD – CompLearn/zip**	**Rate-Distortion**
Strips	***k = 1***		53.1 %	51.9 %	67.2 %
	k = 10		79.7 %	35.9 %	76.6 %
	k = 30		90.6 %	43.8 %	87.5 %
Aliasing	***ds = 3***		85.9 %	76.6 %	100 %
	ds = 5		90.6 %	79.7 %	84.4 %
	ds = 7		81.3 %	76.6 %	81.3 %

Table 6.1: Results for *RD/NCD* based artifact detection with synthetic artifacts.

Table 6.1 shows the results obtained by the proposed methods with the database described in Section 6.1. The *NCD* calculation was made using the two lossless compressors JPEG-LS and ZIP; the *RD* curve was calculated using the baseline JPEG lossy compressor.

We can observe that the critical intensity for strip detection is lowest for *sea* images; we obtain 62.5% of success using the method based on *NCD* and 89.1% of success using the method based on *RD* analysis for $k = 9$, where k is the intensity of the strips. On the other hand, we need $k = 50$ and $k = 30$ for an artifact detection in the *forest* and *city* Earth surface classes; we obtain an 89.1% success rate using *NCD* and 92.2% of success using *RD* for the *forest* surface class; and 90.6% of success using *NCD* and 87.5% of success using *RD* for the *city* surface class.

For aliasing detection, the best results are obtained for the *city* surface class where we get a 90.6% success rate using the *NCD* approach and 100% using the *RD* approach. For the *sea* and *forest* surface classes the success rate is lower; we obtain 65.6% using the *NCD* approach and 78.1% using the *RD* approach; 40.6% using the *NCD* approach and 65.6% using the *RD* approach respectively. A better detection of aliasing occurs in the *city* because the image bandwidth is wider.

We can observe that the results obtained by the *RD* analysis approach are better than *NCD* in most cases; the *RD* curve can discriminate between an image with artifacts and an image without artifacts.

6.2.2 Results of the Complexity-to-Error Migration Method

For the *CEM* method described in Chapter 5, Section 5.3, the obtained results are shown in Tables 6.2 to 6.5. The results are expressed as their percentage of success. As this method is our most innovative approach, we expanded our investigations into the use of different compressors. For lossless compression we used both JPEG-LS and ZIP, while for lossy compression we used baseline JPEG and JPEG 2000. For an in-depth analysis of the best lossless/lossy combination we used a permutation of all cases.

6.2.2.1 Use of Baseline JPEG and JPEG-LS

We present the baseline JPEG / JPEG-LS summary results for our images in Table 6.2 followed by three illustrations shown in Figures 6.4 to 6.6. In these summary results, we include the five proposed measures, namely

- the complexity variation (*CV*)
- the *CEM* between the original image *X* and the compressed-decompressed image *Y*
- the *CEM* between the original image *X* and the error map *E*
- the *CEM* between the error map *E* and the compressed-decompressed image *Y* and
- the Kolmogorov Structure Function (*KSF*).

In Figures 6.4 to 6.6, we illustrate the relationships contained in Table 6.2 in an easily comprehensible format.

We can observe that the critical intensity for strip detection is lower for the *sea* image; we obtain a success rate of 90.6% when using the *CV* method, 57.8% with *CEM(X,Y)*, 81.3% with *CEM(X,E)*, 81.3% with *CEM(E,Y)*, and 65.6% of success using the *KFS* analysis method for $k = 9$, where k is the intensity of the strips. On the other hand, we need a $k \geq 50$ and $k \geq 30$ for a detection in the *forest* and *city* images; we obtain 54.7% of success using *CV*, 62.5% using *CEM(X,Y)*, 65.6% using *CEM(X,E)*, 59.4% with *CEM(E,Y)* and 92.2% of success using *KSF* for the *forest* image; we get 60.9% of success using *CV*, 87.5% using *CEM(X,Y)*, 56.3% using *CEM(X,E)*, 71.9% with *CEM(E,Y)*, and 89.1% of success using *KSF* for the *city* image.

These data tell us that we have a wide range of success rates. When we try to understand the causes of these diverse success rates, we have to take into account quite a number of parameters. The main factors will be explained in Section 6.3.

SEA (mean value: 19 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 1***		62.5 %	***62.5 %***	70.3 %	59.4 %	64.1 %
	k = 3		60.9 %	***62.5 %***	71.9 %	57.8 %	64.1 %
	k = 6		71.9 %	57.8 %	78.1 %	51.6 %	64.1 %
	k = 9		90.6 %	57.8 %	***81.3 %***	***81.3 %***	***65.6 %***
	k = 10		***92.2 %***	57.8 %	***81.3 %***	54.7 %	***65.6 %***
Aliasing	***ds = 5***		68.8 %	68.8 %	68.8 %	68.8 %	***53.1 %***
	ds = 9		***71.9 %***	***71.9 %***	***70.3 %***	***71.9 %***	***53.1 %***

FOREST (mean value: 39 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 2***		64 %	***64.1 %***	***68.8 %***	***60.9 %***	67.2 %
	k = 6		***64.1 %***	62.5 %	***68.8 %***	***60.9 %***	67.2 %
	k = 10		***64.1 %***	62.5 %	65.6 %	59.4 %	59.4 %
	k = 50		54.7 %	62.5 %	65.6 %	59.4 %	***92.2 %***
Aliasing	***ds = 5***		***54.7 %***	***53.1 %***	***62.5 %***	***54.7 %***	70.3 %
	ds = 7		51.6 %	***53.1 %***	53.1 %	53.1 %	***76.6 %***

CITY (mean value: 64 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 1***		54.7 %	53.1 %	53.1 %	54.7 %	54.7 %
	k = 10		50 %	53.1 %	***56.3 %***	56.3 %	51.6 %
	k = 30		***60.9 %***	***87.5 %***	***56.3 %***	***71.9 %***	***89.1 %***
Aliasing	***ds = 3***		54.7 %	***60.9 %***	***85.9 %***	50 %	***92.2 %***
	ds = 5		51.6 %	51.6 %	53.1 %	53.1 %	85.9 %
	ds = 7		***57.8 %***	57.8 %	53.1 %	***54.7 %***	75 %

Table 6.2: Results for artifact detection using a baseline JPEG lossy compressor and a JPEG-LS lossless compressor.

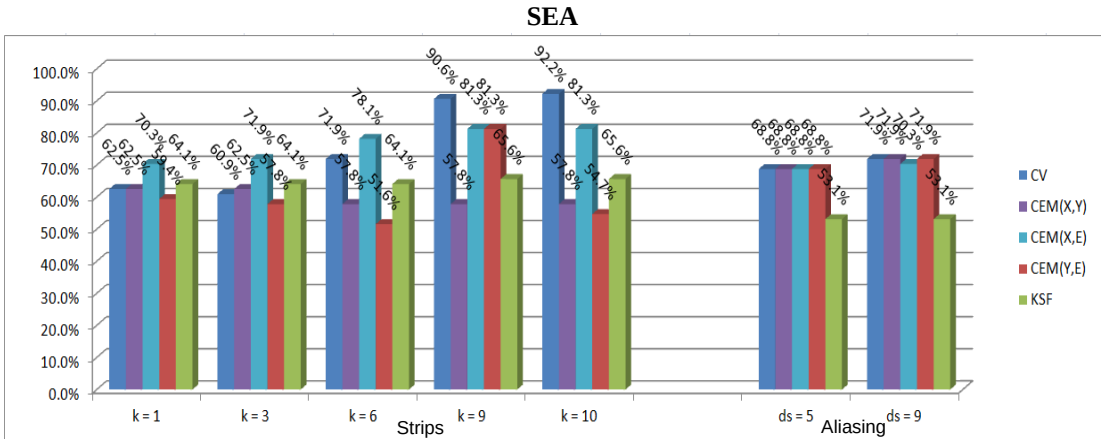


Figure 6.4: Summary results of Table 6.2 for the *sea* image.

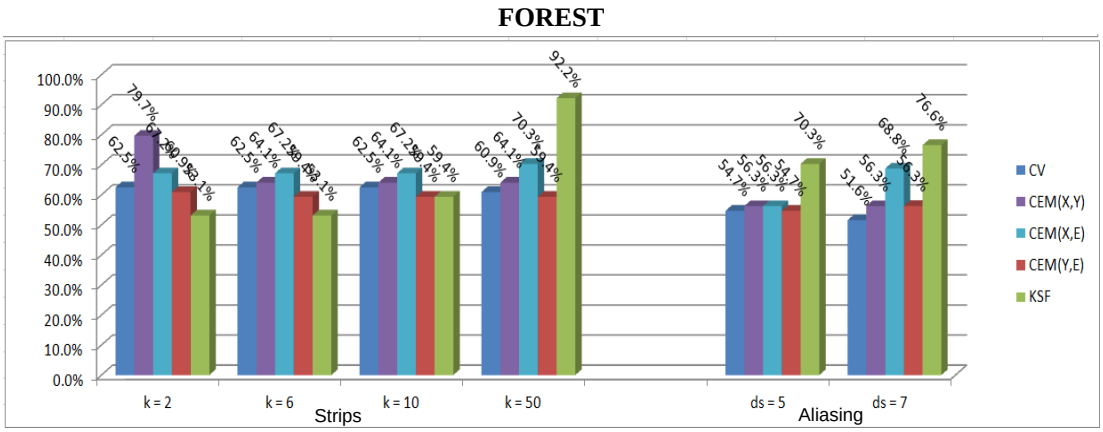


Figure 6.5: Summary results of Table 6.2 for the *forest* image.

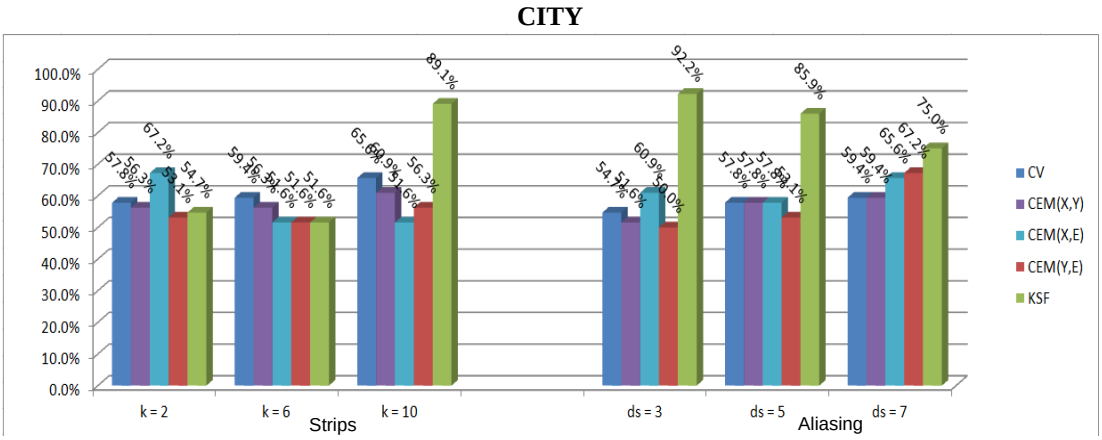


Figure 6.6: Summary results of Table 6.2 for the *city* image.

For aliasing detection within *sea* and *forest* images the percentage of successful detection is lower than for the city image; for the *sea* image, we obtain 68.8% of success using *CV*, 68.8% using *CEM(X,Y)*, 68.8% using *CEM(X,E)*, 68.8% with *CEM(E,Y)*, and 53.1% of success using the *KSF* approach; the *forest* image leads to 54.7% of success using *CV*, 53.1% using *CEM(X,Y)*, 62.5% using *CEM(X,E)*, 54.7% with *CEM(E,Y)*, and 70.3% of success using the *KSF* approach. The best results are obtained for the *city* image where we get a success rate of 54.7% using *CV*, 60.9% using *CEM(X,Y)*, 85.9% using *CEM(X,E)*, 50% with *CEM(E,Y)*, and 92.2% of success using the *KSF* approach. This is due to the fact that the *city* image bandwidth is wider. Again, the diversity of the results calls for more detailed explanations. These are to be found in Section 6.3.

6.2.2.2 Use of Baseline JPEG and ZIP

We present the baseline JPEG / ZIP summary results for our images in Table 6.3 followed by three illustrations shown in Figures 6.7 to 6.9. In these summary results, we include the five proposed measures listed in Section 6.2.2.1.

In Figures 6.7 to 6.9, we illustrate the relationships contained in Table 6.3 in an easily comprehensible format.

We can observe that the critical intensity for strip detection is lowest for the *sea* image; we obtain a success rate of 68.8% using the method based on *CV,* 51.6% with *CEM(X,Y),* 71.9% with *CEM(X,E),* 51.6% with *CEM(E,Y),* and 65.6% of success using the method based on *KSF* analysis for $k = 9$, where k is the intensity of the strips. On the other hand, we need a $k \geq 50$ and $k \geq 30$ for a detection in the *forest* and *city* images; we obtain 60.9% of success using *CV,* 64.1% using *CEM(X,Y),* 70.3% using *CEM(X,E),* 59.4% with *CEM(E,Y),* and 92.2% of success using *KSF* for the *forest* image, as well as 65.6% of success using *CV,* 60.9% using *CEM(X,Y),* 51.6% using *CEM(X,E),* 56.3% with *CEM(E,Y),* and 89.1% of success using *KSF* for the *city* image.

SEA (mean value 19 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 1***		51.6 %	***51.6 %***	***71.9 %***	***51.6 %***	64.1 %
	k = 3		***68.8 %***	***51.6 %***	***71.9 %***	***51.6 %***	64.1 %
	k = 6		***68.8 %***	***51.6 %***	***71.9 %***	***51.6 %***	64.1%
	k = 9		***68.8 %***	***51.6 %***	***71.9 %***	***51.6 %***	***65.6 %***
	k = 10		64.1 %	***51.6 %***	***71.9 %***	***51.6 %***	***65.6 %***
Aliasing	***ds = 5***		65.6 %	65.6 %	67.2 %	65.6 %	51.6 %
	ds = 9		***71.9 %***	***71.9 %***	***71.9 %***	***70.3 %***	***62.5 %***

FOREST (mean value 39 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 2***		***62.5 %***	***79.7 %***	67.2 %	***60.9 %***	53.1 %
	k = 6		***62.5 %***	64.1 %	67.2 %	59.4 %	53.1 %
	k = 10		***62.5 %***	64.1 %	67.2 %	59.4 %	59.4 %
	k = 50		60.9 %	64.1 %	***70.3 %***	59.4 %	***92.2 %***
Aliasing	***ds = 5***		***54.7 %***	***56.3 %***	56.3 %	54.7 %	70.3 %
	ds = 7		51.6 %	***56.3 %***	***68.8 %***	***56.3 %***	***76.6 %***

CITY (mean value 64 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 1***		57.8 %	56.3 %	***67.2 %***	53.1 %	54.7 %
	k = 10		59.4 %	56.3 %	51.6 %	51.6 %	51.6 %
	k = 30		***65.6 %***	***60.9 %***	51.6 %	***56.3 %***	***89.1 %***
Aliasing	***ds = 3***		54.7 %	51.6 %	60.9 %	50 %	***92.2 %***
	ds = 5		57.8 %	57.8 %	57.8 %	53.1 %	85.9 %
	ds = 7		***59.4%***	***59.4 %***	***65.6 %***	***67.2 %***	75 %

Table 6.3: Results for artifact detection using a baseline JPEG lossy compressor and a ZIP lossless compressor.

For aliasing detection within the *sea* and *forest* images the percentage of success detection is low; for the *sea* image, we obtain 65.6% of success using *CV*, 65.6% using *CEM(X,Y)*, 67.2% using *CEM(X,E)*, 65.6% with *CEM(E,Y)*, and 51.6% of success using the *KSF* approach; for the *forest* image, we get 54.7% of success using *CV*, 56.3% using *CEM(X,Y)*, 56.3% using *CEM(X,E)*, 54.7% with *CEM(E,Y)*, and 70.3% of success using the *KSF* approach. The best results are obtained for the *city* image with a wide bandwidth where we obtain a success rate of 54.7% by using *CV*, 51.6% using *CEM(X,Y)*, 60.9% using *CEM(X,E)*, 50% with *CEM(E,Y)*, and 92.2% of success using the *KSF* approach.

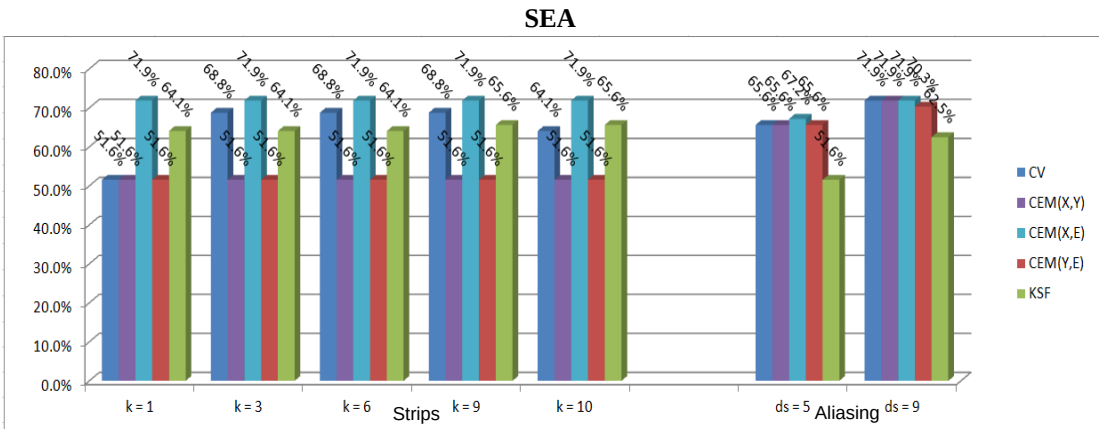


Figure 6.7: Summary results of Table 6.3 for the *sea* image.

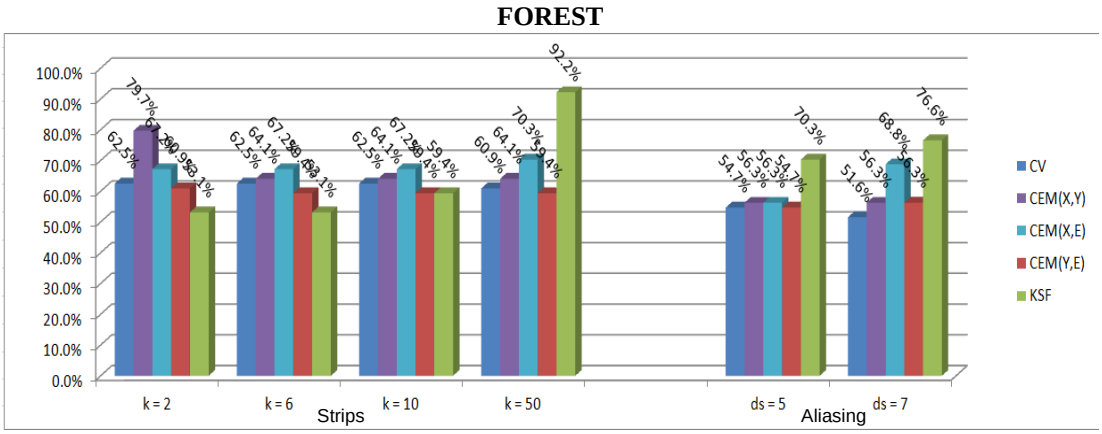


Figure 6.8: Summary results of Table 6.3 for the *forest* image.

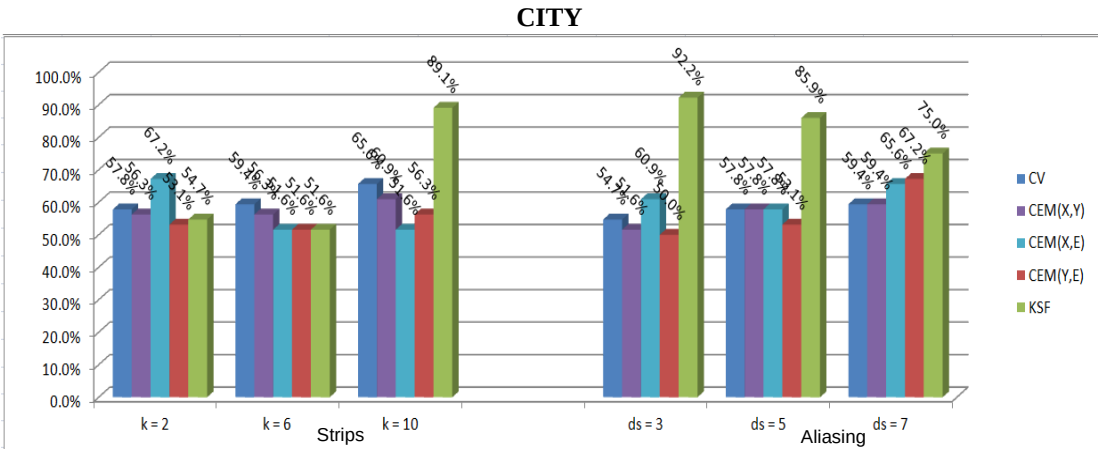


Figure 6.9: Summary results of Table 6.3 for the *city* image.

As a conclusion, we can say that the use of ZIP instead of JPEG-LS does not lead to better results when we combine them with the baseline JPEG lossy compressor.

6.2.2.3 Use of JPEG 2000 and JPEG-LS

Now we present the JPEG 2000 / JPEG-LS summary results for our images in Table 6.4 followed by three illustrations shown in Figures 6.10 to 6.12. Again, we include the five proposed measures listed in Section 6.2.2.1.

In Figures 6.10 to 6.12, we illustrate the relationships contained in Table 6.4 in an easily comprehensible format.

We can observe that the critical intensity for strip detection is lowest for the *sea* image; we obtain a success rate of 65.63% by using the method based on *CV*, 65.6% with *CEM(X,Y)*, 79.7% with *CEM(X,E)*, 76.6% with *CEM(E,Y)*, and 71.9% of success using the method based on *KSF* analysis for $k = 9$, where k is the intensity of the strips. On the other hand, we need a $k \geq 50$ and $k \geq 30$ for a detection within the *forest* and *city* images; we obtain 65.6% of success using *CV*, 68.8% using *CEM(X,Y)*, 59.4% using *CEM(X,E)*, 67.2% with *CEM(E,Y)*, and 90.6% of success using *KSF* for the *forest* image; for the *city* image we get 70.3% of success using *CV*, 54.7% using *CEM(X,Y)*, 62.5% using *CEM(X,E)*, 56.3% with *CEM(E,Y)*, and 76.6% of success using *KSF*.

For aliasing detection within the *sea* and *forest* images the success rate is low; for the *sea* image we obtain 67.2% of success using *CV*, 67.2% using *CEM(X,Y)*, 67.2% using *CEM(X,E)*, 67.2% with *CEM(E,Y)*, and 57.8% of success using the *KSF* approach; the *forest* image leads to 64.1% of success using *CV*, 67.2% using *CEM(X,Y)*, 68.8% using *CEM(X,E)*, 71.9% with *CEM(E,Y)*, and 68.8% of success using the *KSF* approach. The best detection of aliasing occurs in the *city* image because its bandwidth is wider. We get a success rate of 51.6% by using *CV*, 51.6% using *CEM(X,Y)*, 93.8% using *CEM(X,E)*, 92.2% with *CEM(E,Y)*, and 85.9% of success using the *KSF* approach.

SEA (mean value 19 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 1***		57.8 %	56.3 %	71.9 %	70.3 %	***71.9 %***
	k = 3		62.5 %	62.5 %	76.6 %	71.9 %	***71.9 %***
	k = 6		64.1 %	64.1 %	76.6 %	75 %	***71.9 %***
	k = 9		***65.6 %***	***65.6 %***	***79.7 %***	***76.6 %***	***71.9 %***
	k = 10		64.6 %	***65.6 %***	76.6 %	75 %	***71.9 %***
Aliasing	***ds = 5***		67.2 %	67.2 %	67.2 %	67.2 %	***57.8 %***
	ds = 9		***71.9 %***	***71.9 %***	***70.3 %***	***71.9 %***	53.1 %

FOREST (mean value 39 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 2***		***73.4 %***	***68.8 %***	***71.9 %***	***68.8 %***	60.9 %
	k = 6		***73.4 %***	***68.8 %***	67.2 %	67.2 %	60.9 %
	k = 10		***73.4 %***	***68.8 %***	57.8 %	67.2 %	62.5 %
	k = 50		65.6 %	***68.8 %***	59.4 %	67.2 %	***90.6 %***
Aliasing	***ds = 5***		64.1 %	67.2 %	***68.8 %***	***71.9 %***	68.8 %
	ds = 7		***68.8 %***	***68.8 %***	65.6 %	68.8 %	***75 %***

CITY (mean value 64 DN)

Kind of Artifact			Artifact Detection Method				
			CV	CEM(X,Y)	CEM(X,E)	CEM(E,Y)	KFS
Strips	***k = 1***		53.1 %	50 %	51.6 %	51.6 %	73.4 %
	k = 10		50 %	51.6 %	54.7 %	54.7 %	75 %
	k = 30		***70.3 %***	***54.7 %***	***62.5 %***	***56.3 %***	***76.6 %***
Aliasing	***ds = 3***		51.6 %	51.6 %	***93.8 %***	***92.2 %***	***85.9 %***
	ds = 5		***57.8 %***	***57.8 %***	65.6 %	62.5 %	79.7 %
	ds = 7		***57.8 %***	***57.8 %***	65.6 %	60.9 %	65.6 %

Table 6.4: Results for artifact detection using a JPEG 2000 lossy compressor with compression rates from 0.001 to 13 and a JPEG-LS lossless compressor.

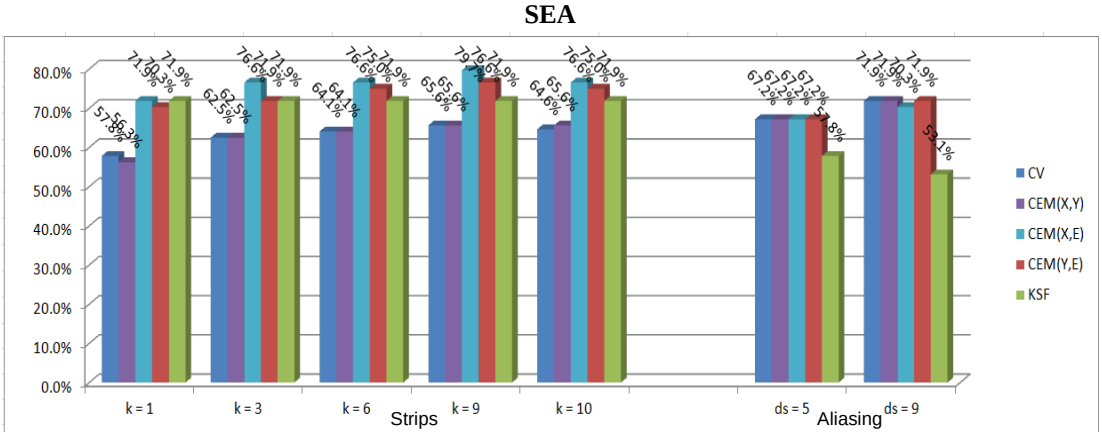


Figure 6.10: Summary results of Table 6.4 for the *sea* image.

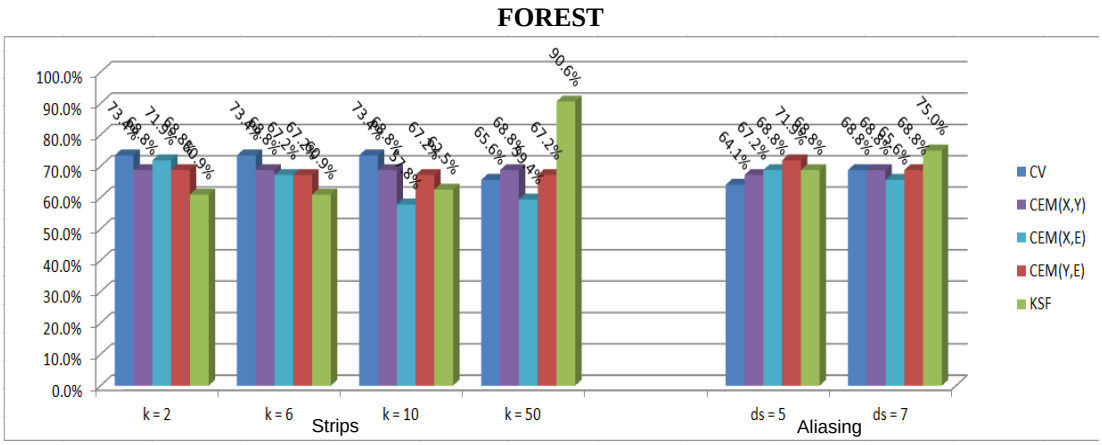


Figure 6.11: Summary results of Table 6.4 for the *forest* image.

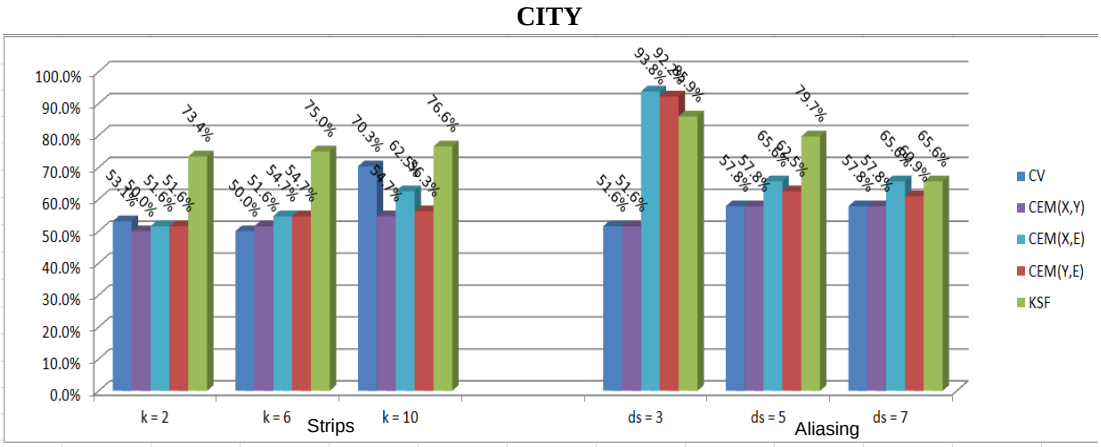


Figure 6.12: Summary results of Table 6.4 for the *city* image.

In summary, one can see that the combination of JPEG 2000 and JPEG-LS will lead to a few good results; however, the majority of results are of mid-level quality. The reason for this behavior can be explained as follows. On the one hand, JPEG 2000 is a very powerful compression method; on the other hand, it is not optimized for distinct error mapping (and thus artifact detection).

6.2.2.4 Use of JPEG 2000 and ZIP

Finally, we present the JPEG 2000 / ZIP summary results for our images in Table 6.5 followed by three illustrations shown in Figures 6.13 to 6.15. Again, we include the five proposed measures listed in Section 6.2.2.1.

In Figures 6.13 to 6.15, we illustrate the relationships contained in Table 6.5 in an easily comprehensible format.

We can observe that the critical intensity for strip detection is lowest for the *sea* image; we obtain 57.8% of success using the method based on *CV*, 71.9% with *CEM(X,Y)*, 71.9% with *CEM(X,E)*, 71.9% with *CEM(E,Y)*, and 71.9% of success using the method based on *KSF* analysis for $k = 9$, where k is the intensity of the strips. On the other hand, we need a $k \geq 50$ and $k \geq 30$ for a detection in the *forest* and *city* images; for the *forest* image, we obtain 73.4% of success using *CV*, 73.4% using *CEM(X,Y)*, 70.3% using *CEM(X,E)*, 73.4% with *CEM(E,Y)*, and 90.6% of success using *KSF*; for the *city* image, we get 62.5% of success using *CV*, 53.1% using *CEM(X,Y)*, 54.7% using *CEM(X,E)*, 51.6% with *CEM(E,Y)*, and 76.6% of success using *KSF*.

For aliasing detection within the *sea* and *forest* images the success rate is low; for the *sea* image, we obtain 65.6% of success using *CV*, 65.6% using *CEM(X,Y)*, 67.2% using *CEM(X,E)*, 65.6% with *CEM(E,Y)*, and 57.8% of success using the *KSF* approach; for the *forest* image, we get 50% of success using *CV*, 60.9% using *CEM(X,Y)*, 53.1% using *CEM(X,E)*, 59.4% with *CEM(E,Y)* and 68.8% of success using the *KSF* approach. The best detection of aliasing occurs in the *city* because the image bandwidth is wider. Here, we obtain a success rate of 50% by using *CV*, 93.8% using *CEM(X,Y)*, 71.9% using *CEM(X,E)*, 92.2% with *CEM(E,Y)*, and 85.9% of success using the *KSF* approach.

SEA (mean value 19 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 1***		56.3 %	***71.9 %***	***71.9 %***	***71.9 %***	***71.9 %***
	k = 3		56.3 %	***71.9 %***	***71.9 %***	***71.9 %***	***71.9 %***
	k = 6		56.3 %	***71.9 %***	***71.9 %***	***71.9 %***	***71.9 %***
	k = 9		***57.8 %***	***71.9 %***	***71.9 %***	***71.9 %***	***71.9 %***
	k = 10		***57.8 %***	***71.9 %***	***71.9 %***	***71.9 %***	***71.9 %***
Aliasing	***ds = 5***		65.6 %	65.6 %	67.2 %	65.6 %	***57.8 %***
	ds = 9		***70.3 %***	***70.3 %***	***71.9 %***	***70.3 %***	53.1 %

FOREST (mean value 39 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 2***		***73.4 %***	68.8 %	***70.3 %***	***73.4 %***	60.9 %
	k = 6		***73.4 %***	70.3 %	***70.3 %***	***73.4 %***	60.9 %
	k = 10		***73.4 %***	68.8 %	***70.3 %***	***73.4 %***	62.5 %
	k = 50		***73.4 %***	***73.4 %***	***70.3 %***	***73.4 %***	***90.6 %***
Aliasing	***ds = 5***		50 %	60.9 %	53,1 %	59.4 %	68.8 %
	ds = 7		***68.8 %***	***68.8 %***	***68.8 %***	***68.8 %***	***75 %***

CITY (mean value 64 DN)

Kind of Artifact			**Artifact Detection Method**				
			CV	**CEM(X,Y)**	**CEM(X,E)**	**CEM(E,Y)**	**KFS**
Strips	***k = 1***		59.4 %	***59.4 %***	***56.3 %***	***53.1 %***	73.4 %
	k = 10		51.6 %	53.1 %	***56.3 %***	51.6 %	75 %
	k = 30		***62.5 %***	53.1 %	54.7 %	51.6 %	***76.6 %***
Aliasing	***ds = 3***		50 %	***93.8 %***	71.9 %	***92.2 %***	***85.9 %***
	ds = 5		51.6 %	59.4 %	***75 %***	65.6 %	79.7 %
	ds = 7		***62.5 %***	68.8 %	67.2 %	65.6 %	65.6 %

Table 6.5: Result for artifact detection using a JPEG 2000 lossy compressor and a ZIP lossless compressor.

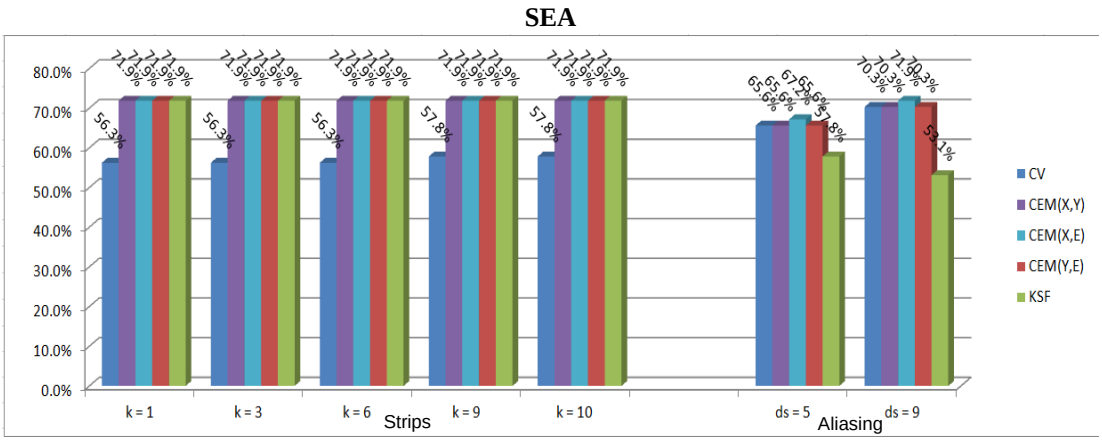


Figure 6.13: Summary results of Table 6.5 for the *sea* image.

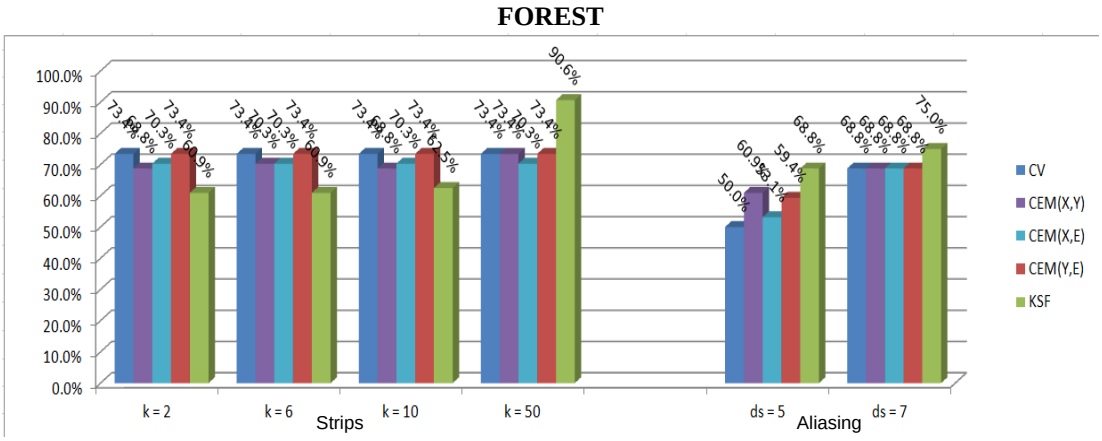


Figure 6.14: Summary results of Table 6.5 for the *forest* image.

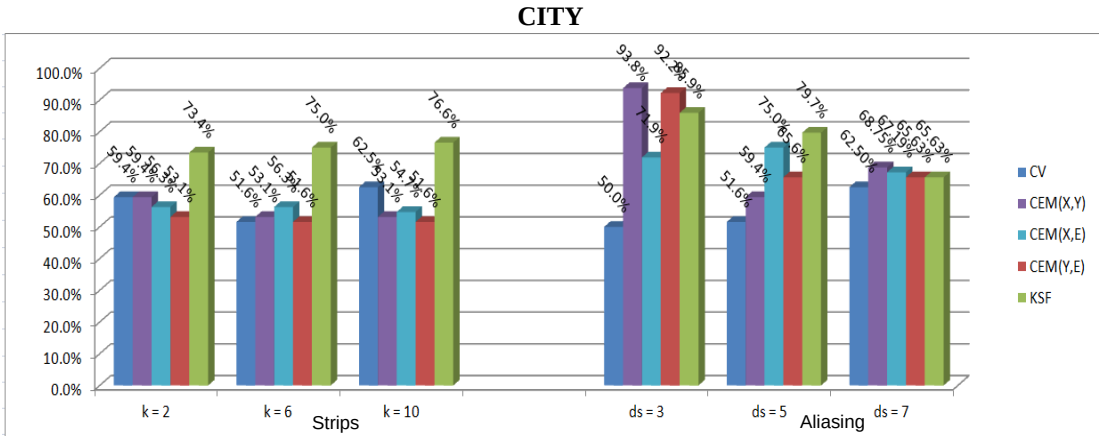


Figure 6.15: Summary results of Table 6.5 for the *city* image.

When we try to summarize the use of JPEG 2000 and ZIP, we again face a few good results and a majority of mid-performance results. See Section 6.2.3.3 for further explanations.

6.2.3 Results of Existing Methods Based on Quality Metrics

In contrast, for the method based on image quality metrics described in Chapter 5, Section 5.4, the obtained results are shown in Table 6.6. As always, the results are expressed as their percentage of success.

SEA (mean value: 19 DN)

Kind of Artifact			**Artifact Detection Using Image Quality Metrics**
Strips	***k = 1***		23.4 %
	k = 3		23.4 %
	k = 6		23.4 ***%***
	k = 9		23.4 ***%***
	k = 10		23.4 %
Aliasing	***ds = 5***		***51.6 %***
	ds = 9		46.9 %

FOREST (mean value: 39 DN)

Kind of Artifact			**Artifact Detection Using Image Quality Metrics**
Strips	***k = 2***		65 %
	k = 6		65 %
	k = 10		65 %
	k = 50		***73.4 %***
Aliasing	***ds = 5***		***51.6 %***
	ds = 7		32.8 %

CITY (mean value: 64 DN)

Kind of Artifact			**Artifact Detection Using Image Quality Metrics**
Strips	***k = 1***		***57.8 %***
	k = 10		54.7 %
	k = 30		50 %
Aliasing	***ds = 3***		***78.1 %***
	ds = 5		65.6 %
	ds = 7		56.3 %

Table 6.6: Results for artifact detection with synthetic artifacts using image quality metrics.

We can recognize that for strip detection is not possible to detect any artifact within the *sea* image. For the *forest* image, we obtain 73.4% of success for $k = 50$, while for the *city* image, we get a 57.8% success rate for $k = 1$.

For aliasing detection, it is also not possible to detect any artifact within the *sea* image. However, we obtain 51.6% of success for the *forest* image and 78.1% of success for the *city* image.

6.3 Conclusions for Artifact Detection

In this section, we will provide an overview about the performance of the various artifact detection methods, while an overall summary of the entire thesis will be presented in Chapter 7.

When we take the approaches presented in Chapter 5, Sections 5.1, Section 5.2, and Section 5.4 we realize that strips have a good chance to be detected in a *sea* environment with an intensity of $k \geq 10$; however, in the *forest* and the *city* environment we need $k \geq 30$. In contrast, aliasing can be detected in a *city* environment, but not for a *sea* or a *forest* image, because the *city*'*s* bandwidth is wider than that of *sea* and *forest*.

An analysis of the approach presented in Section 5.3 shows that acceptable results are obtained for the *sea* environment with strips of $k = 10$ with the *complexity variation* method (*CV*). For the *forest* environment we got acceptable results when we have strips of $k \geq 50$ with the *Kolmogorov structure function* (*KSF*). In the *city* environment for aliasing with $ds = 3$ and $ds = 5$, we obtained acceptable results using the *KFS* method. When comparing baseline JPEG to JPEG 2000, we obtained better artifact detection results analyzing the errors produced by the baseline JPEG method because the baseline method produces larger structural errors when depicting artifacts than JPEG 2000.

This can also serve as a more general explanation to the effects seen in Section 6.2.2.1. Here we face the situation of mixed quality results that have not yet been explained. We assume that the *SNCD* is no ideal tool for image quality assessment. This will be explained further in Section 6.4.

In Figures 6.16 to 6.18, we present three summary results, demonstrating the performance of the different methods.

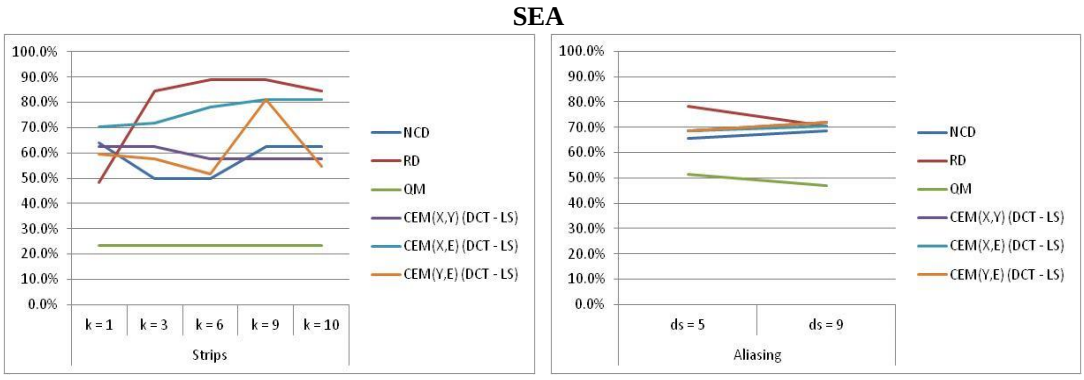


Figure 6.16: Summary results for the *sea* images using the different approaches proposed in Chapter 5.

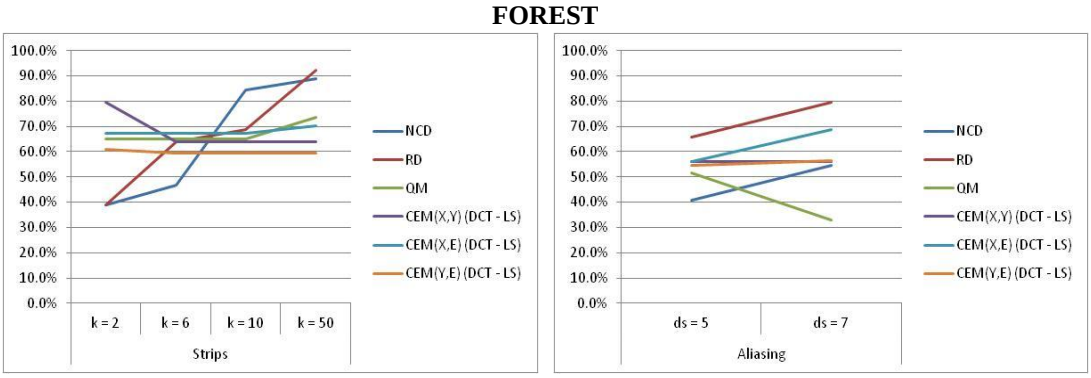


Figure 6.17: Summary results for the *forest* images using the different approaches proposed in Chapter 5.

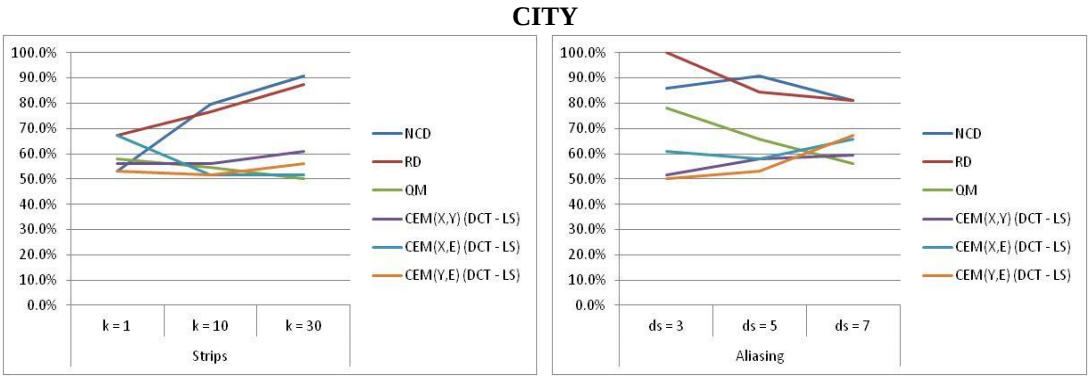


Figure 6.18: Summary results for the *city* images using the different approaches proposed in Chapter 5.

In these summary results, we can recognize that, in most cases, the *RD* analysis approach produces better results than the other approaches. The main reason for this effect seems to be an inadequate description of complexities for *NCD*-based tools when we compare images with and without artifacts.

Finally, we have developed a tool for artifact detection using the previously explained methods; its interface is shown in Figure 6.19. This interface supports the calculation of *NCD* and also the use of the Rate-Distortion function; we can also to vary the size of the image patches.

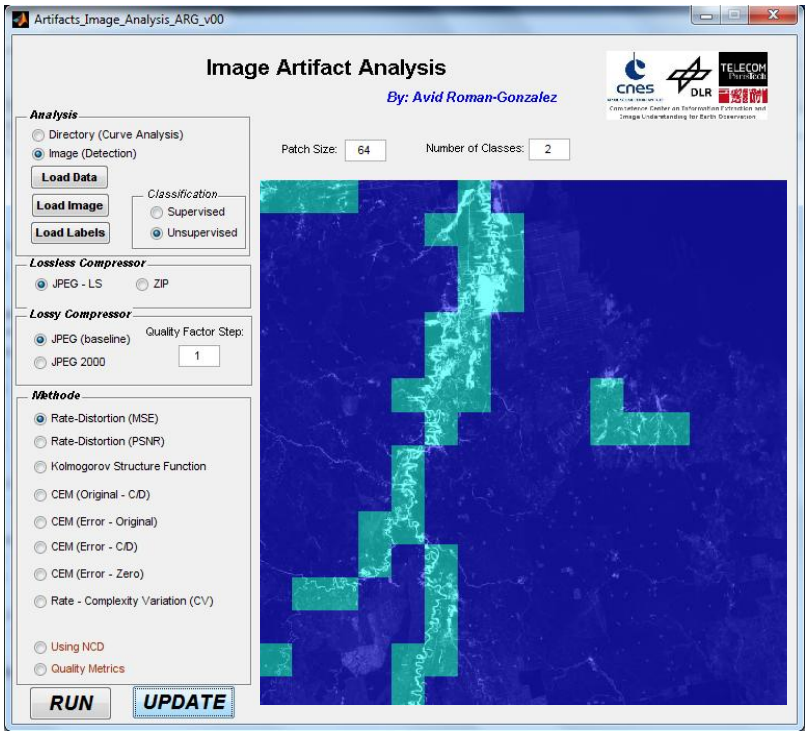


Figure 6.19: Tool for artifact detection: via this interface we can select the calculation of *NCD*, or use Rate-Distortion; one can also vary the image patch size.

Using this interface we can evaluate the performance the different methods in a straightforward manner; it can also be used for educational purposes, and as a testing tool.

6.4 The *SNCD* as a Metric for Image Quality Assessment

In this section, we propose the *SNCD* as a metric for the measurement of image quality, especially when we analyze residual errors. Here we do not consider the detection of artifacts but we concentrate on the quality of full images where we show the performance of this metric comparing it with other metrics that we can found in the literature as in (Sheikh et al. 2006a; Wang & Li 2010; Sheikh & Bovik 2006b).

6.4.1 Database Description

In order to perform experiments with the *SNCD* metric and to make appropriate comparisons, we use a database that has already been used by other researchers and is available on the Internet. The database that we use is the Cornell-A57 collection (Simoneclli & Adelson 1989), consisting of three original images (*baby*, *harbor*, and *horse)* as shown in Figure 6.20 and which also includes distorted images. For each original image, we have six types of distortion:

- blurring by a Gaussian filter (*BLR*)
- blurring by a Gaussian filter (*BLR*)
- JPEG 2000 compression with the dynamic contrast-based quantization algorithm (*DCQ*)
- quantization of the LH (L = Low and H = High) sub-bands of a 5-level discrete wavelet transform, where the sub-bands were quantized via uniform scalar quantization (*FLT*)
- JPEG 2000 compression without visual frequency weighting (*JP2*)
- baseline JPEG compression (*JPG*)
- additive white Gaussian noise (*NOZ*)

For each type of distortion, we have 3 intensities; thus we have a database of 54 images (3 images × 6 distortion types × 3 distortion parameters) plus three original (undistorted) images.

Figure 6.20: Original images of the Cornell-A57 database.

Each image has a size of 512×512 pixels; we can see that the *baby* picture and the *horse* picture contain a predominant object that we will use to analyze the behavior of our selected compression methods together with the existing metrics.

6.4.2 Metrics for Image Quality Assessment

To assess the quality of images as already defined in Section 3.1, we have three approaches: the "*full reference*" approach, the "*non-reference*" approach, and finally, the "*reduced reference*" approach.

For comparison, in the present work, from the many existing metrics in the literature with a full reference approach (where we compare image pairs), we use the *PSNR* and *SSIM* metrics that are also used and evaluated in (Sheikh et al. 2006a; Wang & Li 2010).

The *PSNR* (Peak Signal-to-Noise Ratio) is given by (see Section 3.1):

$$PSNR = 10\log_{10}\left(\frac{L^2}{MSE}\right) \tag{6.1}$$

where *MSE* is the Mean Squared Error and *L* is the maximum dynamic range; for gray-scale images with 8 bits/pixel *L = 255*.

Another metrics is the *SSIM* (Structural Similarity Index, see Section 3.1) that has three independent components: luminance, contrast, and structure. The *SSIM* is given by:

$$SSIM = f(l(x,y), c(x,y), s(s,y)) \tag{6.2}$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{6.3}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{6.4}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{6.5}$$

where μ_x , σ_x and σ_{xy} represent the global mean, the standard deviation, and the cross-correlation of images *x* and *y*. C_1, C_2 and C_3 are selectable constants.

6.4.3 Comparison of Metrics

In order to compare the *PSNR*, *SSIM*, and the *SNCD*, we use three correlation coefficients. These correlation coefficients are calculated from the results obtained by a subjective evaluation of images of the database and the results obtained by the three metrics (when applying them in comparisons between the original images and their distorted versions). This subjective assessment was performed by a group of experts who evaluated the quality of each image in the database.

The correlation measures we will use are :

- *The Pearson correlation coefficient (PCC)* is an index that measures the linear relationship between two quantitative random variables. Unlike the covariance, Pearson correlation is independent of the scale of the measured variables. To calculate the *PCC*, we use the following MATLAB instruction: *corr(MOS, RG, 'type', 'Pearson')*, where *MOS* is the result for the subjective evaluation, and *RG* is the result using the image quality metrics.

- *The Spearman correlation coefficient (SCC)* is a measure of correlation (association or interdependence) between two continuous random variables. To calculate it, the data is sorted and replaced by their ordered indices. We used the following MATLAB instruction: *corr(MOS, RG, 'type', 'Spearman')*, where *MOS* is the result for the subjective evaluation, and *RG* is the result using the image quality metrics.

- *The Kendall correlation coefficient (KCC)* is another non-parametric correlation measure. To calculate the *KCC*, we used the following MATLAB: *corr(MOS, RG, 'type', 'Kendall')*, where *MOS* is the result for the subjective evaluation, and RG is the result using the image quality metrics.

We used all three of these correlations as they are most often similar, but differ on a few occasions.

6.4.4 Analysis of Results

In a first sequence of tests, we calculated the *RG* quality measures of the images of the entire database, and compared them with the *MOS* subjective evaluation using the correlation coefficients explained above. The *MOS* subjective evaluation was obtained from seven imaging experts by using a continuous rating system; greater values represent a greater distortion.

The final correlation coefficient results are shown in Table 6.7.

CORRELATION COEFFICIENTS			
	PCC	***SCC***	***KCC***
PSNR	0.6347	0.6189	0.4309
SSIM	**0.7528**	**0.8066**	**0.6058**
SNCD XY JPEG-LS	0.0967	0.1501	**0.1287**
SNCD XE JPEG-LS	0.2943	**0.1860**	0.1217
SNCD EY JPEG-LS	0.1245	0.1273	0.1063
SNCD XY ZIP	0.0929	0.0448	0.0518
SNCD XE ZIP	**0.295**	0.0278	0.0154
SNCD EY ZIP	0.0196	0.0789	0.0686

Table 6.7: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the entire database.

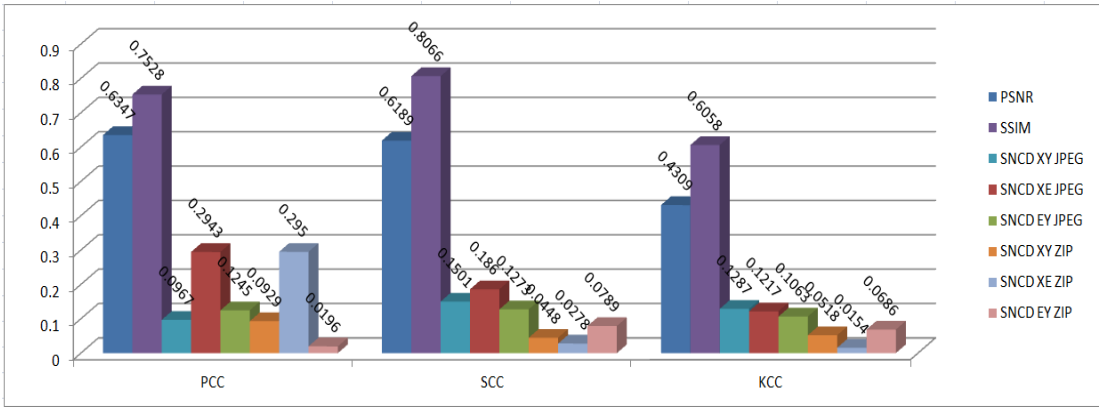


Figure 6.21: Summary results of Table 6.7.

We can see that the best results are obtained by the classical metrics; we obtain a *Pearson* correlation of 0.7528 using *SSIM* metrics, a *Spearman* correlation of 0.8066, and a *Kendall* correlation of 0.6058. The values obtained by the *SNCD* are really very low, indicating that it is not a good representation of the subjective assessment of quality; we obtained for the *SNCD* between the *X* image and the *E* map the following values: a *Pearson*

correlation of 0.2943, a *Spearman* correlation of 0.1860, and a *Kendall* correlation of 0.1287 using a JPEG-LS compressor.

A second experiment we conducted was to sub-divide the database for each given parent image since, as mentioned above, the database contains two parent images with a predominant structure, and another parent image that does not have a predominant structure; then we could see how they behave with respect to the metrics. The results are shown in Tables 6.8 to 6.10 and in graphical representation in Figures 6.22 to 6.24.

CORRELATION COEFFICIENTS FOR "BABY"			
	PCC	***SCC***	***KCC***
PSNR	**0.7786**	0.6925	**0.5686**
SSIM	0.7559	**0.7152**	0.5556
SNCD XY JPEG	0.1983	0.3024	0.1895
SNCD XE JPEG	0.3102	0.2239	0.1111
SNCD EY JPEG	0.1503	**0.5501**	**0.4510**
SNCD XY ZIP	**0.4109**	0.4613	0.3595
SNCD XE ZIP	0.0975	0.0072	0.0458
SNCD EY ZIP	0.2105	0.3664	0.2680

Table 6.8: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the 18 degraded *baby* images.

CORRELATION COEFFICIENTS FOR "HARBOUR"			
	PCC	***SCC***	***KCC***
PSNR	0.6137	0.7438	0.5461
SSIM	**0.7375**	**0.8182**	**0.6382**
SNCD XY JPEG	0.2312	0.0723	0.0066
SNCD XE JPEG	0.3629	**0.2965**	**0.1645**
SNCD EY JPEG	**0.4642**	0.1560	0.0724
SNCD XY ZIP	0.2773	0.0465	0.1118
SNCD XE ZIP	0.1945	0.0031	0.0461
SNCD EY ZIP	0.3626	0.1829	0.0855

Table 6.9: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the 18 degraded *harbour* images.

CORRELATION COEFFICIENTS FOR "HORSE"			
	PCC	***SCC***	***KCC***
PSNR	**0.7968**	0.6863	0.4771
SSIM	0.7779	**0.7936**	**0.5948**
SNCD XY JPEG	**0.3282**	**0.3230**	**0.2941**
SNCD XE JPEG	0.2917	0.0423	0.0196
SNCD EY JPEG	0.0595	0.1538	0.0980
SNCD XY ZIP	0.3099	0.1950	0.1373
SNCD XE ZIP	0.0829	0.1373	0.1111
SNCD EY ZIP	0.1605	0.1889	0.1111

Table 6.10: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the 18 degraded *horse* images.

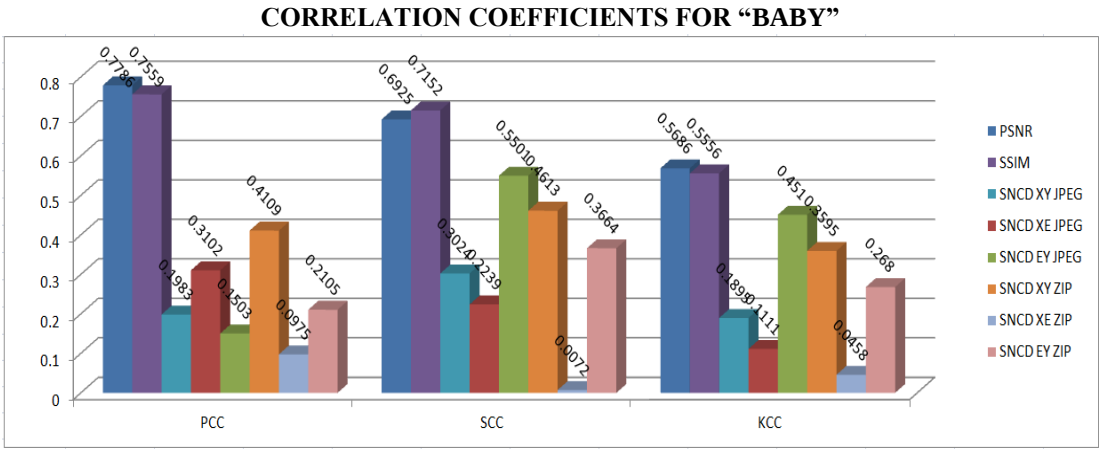


Figure 6.22: Summary results of Table 6.8.

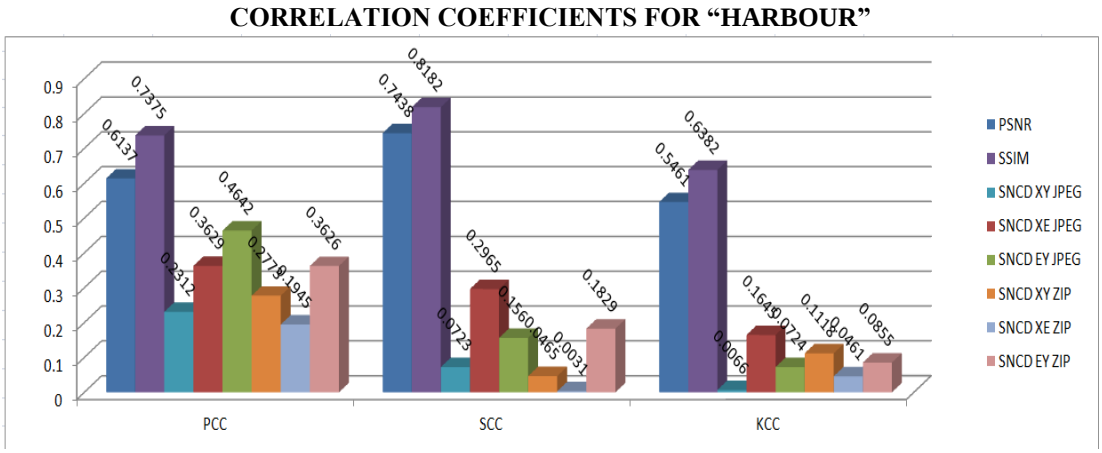


Figure 6.23: Summary results of Table 6.9.

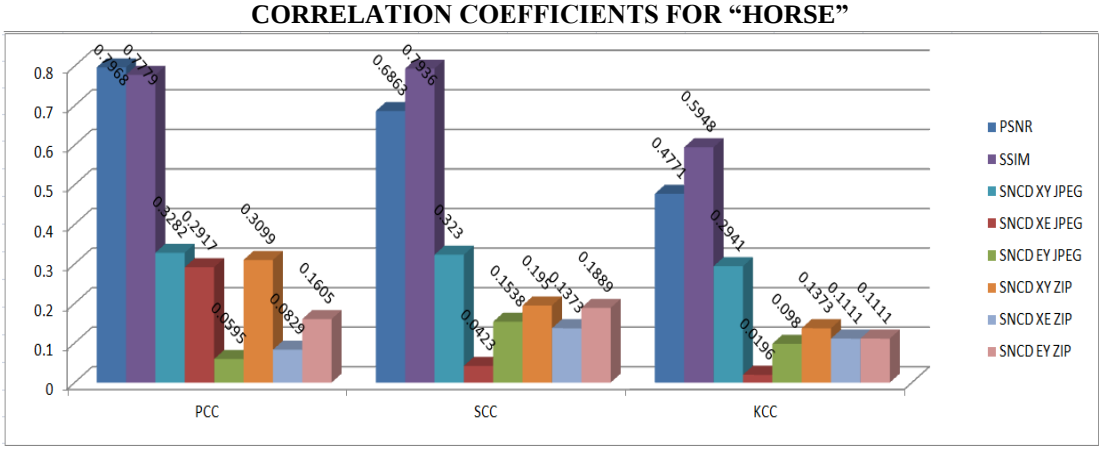


Figure 6.24: Summary results of Table 6.10.

When we sub-divide the database into smaller databases for each parent image, we see that the traditional metrics for image quality still show a better performance (see Figures 6.22 to 6.24). For *baby*, our best correlation results are a *Pearson* correlation of 0.7786 when using *PSNR*, a *Spearman* correlation of 0.7152 using *SSIM*, and a *Kendall* correlation of 0.5686 using *PSNR*. For the *harbour* image, we obtain as our best correlation results a *Pearson* correlation of 0.7375 when using *SSIM*, a *Spearman* correlation of 0.8182 using *SSIM*, and a *Kendall* correlation of 0.6382 using *SSIM*. Finally, for the *horse* image we obtain as best results a *Pearson* correlation of 0.7968 when using *PSNR*, a *Spearman* correlation of 0.7936 using *SSIM*, and a *Kendall* correlation of 0.5948 using *SSIM*.

We also see that the performance of the *SNCD* has improved somewhat compared to our first full database experiment, although is still not comparable with the classical metrics. We obtain the following best *SNCD* results: for the *baby* image, we obtain a *Pearson* correlation of 0.4109 when using *SNCD XY*, a *Spearman* correlation of 0.5501 using *SNCD EY*, and a *Kendall* correlation of 0.4510 using *SNCD EY*. For the *harbour* image, we get a *Pearson* correlation of 0.4642 when using *SNCD EY*, a *Spearman* correlation of 0.2965 using *SNCD XE*, and a *Kendall* correlation of 0.1645 using *SNCD XE*. Finally, for the *horse* image we obtain a *Pearson* correlation of 0.3282 when using *SNCD XY*, a *Spearman* correlation of 0.3230 using *SNCD XY*, and a *Kendall* correlation of 0.2941 using *SNCD XY*.

We could expect that *SNCD* can improve the comparison performance for images with a predominant structure, but experience shows that it is not.

Therefore, the third experiment to perform is to sub-divide the database according to the type of distortion. In this case, we have 6 types of distortion with 9 distorted images for each one. The results are shown in Tables 6.11 to 6.16 and in graphical representation in Figures 6.25 to 6.30.

CORRELATION COEFF. FOR *"BLR"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	0.5904	0.4667	0.3889
SSIM	**0.9421**	**0.8000**	**0.6667**
SNCD XY JPEG-LS	**0.8243**	**0.5167**	0.3889
SNCD XE JPEG-LS	0.7199	0.5000	0.3889
SNCD EY JPEG-LS	0.5738	0.3833	0.2778
SNCD XY ZIP	0.3872	0.2833	0.2778
SNCD XE ZIP	0.5924	0.4333	0.2778
SNCD EY ZIP	0.6477	0.5333	**0.4444**

Table 6.11: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *blurring* distorted 9 images.

CORRELATION COEFF. FOR *"DCQ"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	0.5637	0.5000	0.3889
SSIM	**0.9369**	**0.9667**	**0.8889**
SNCD XY JPEG-LS	**0.9472**	0.8833	0.7778
SNCD XE JPEG-LS	0.4522	0.3833	0.2222
SNCD EY JPEG-LS	0.9115	0.8500	0.7222
SNCD XY ZIP	0.5940	0.7333	0.5556
SNCD XE ZIP	0.2456	0.2333	0.1667
SNCD EY ZIP	0.9051	**0.9667**	**0.8889**

Table 6.12: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *JPEG 2000 + DCQ* distorted 9 images.

CORRELATION COEFF. FOR *"FLT"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	**0.9100**	**0.9000**	**0.7222**
SSIM	0.8982	0.8500	0.6667
SNCD XY JPEG-LS	0.4327	0.3333	0.2222
SNCD XE JPEG-LS	0.9533	0.9167	0.7778
SNCD EY JPEG-LS	0.9432	0.9000	0.8333
SNCD XY ZIP	0.4342	0.2667	0.1667
SNCD XE ZIP	0.9519	0.9500	0.8333
SNCD EY ZIP	**0.9803**	**0.9667**	**0.8889**

Table 6.13: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *FLT allocation* distorted 9 images.

CORRELATION COEFF. FOR *"JP2"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	0.7957	0.8000	**0.6667**
SSIM	**0.8641**	**0.8167**	**0.6667**
SNCD XY JPEG-LS	0.6422	0.6833	0.5000
SNCD XE JPEG-LS	**0.7495**	0.7000	0.5000
SNCD EY JPEG-LS	0.7470	0.7000	**0.5556**
SNCD XY ZIP	0.1047	0.1500	0.1667
SNCD XE ZIP	0.6645	0.6333	0.5000
SNCD EY ZIP	0.6742	**0.7167**	0.5556

Table 6.14: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *JPEG 2000 Compression* distorted 9 images.

CORRELATION COEFF. FOR *"JPG"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	0.7008	0.6333	0.5000
SSIM	**0.9178**	**0.9333**	**0.7778**
SNCD XY JPEG-LS	0.6659	0.7167	**0.6111**
SNCD XE JPEG-LS	0.1015	0.4167	0.1667
SNCD EY JPEG-LS	0.6852	0.7333	0.5556
SNCD XY ZIP	**0.7225**	**0.7833**	0.6111
SNCD XE ZIP	0.0300	0.0667	0.0556
SNCD EY ZIP	0.0163	0.0833	0.1111

Table 6.15: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *baseline JPEG Compression* distorted 9 images.

CORRELATION COEFF. FOR *"NOZ"* DISTORTION			
	PCC	***SCC***	***KCC***
PSNR	**0.9340**	**0.9500**	**0.8333**
SSIM	0.8834	**0.9500**	**0.8333**
SNCD XY JPEG-LS	0.3986	0.2500	0.2222
SNCD XE JPEG-LS	0.3254	0.2833	0.2222
SNCD EY JPEG-LS	0.4414	0.5000	0.3889
SNCD XY ZIP	0.5715	0.3333	0.3333
SNCD XE ZIP	0.8552	0.8333	0.7222
SNCD EY ZIP	**0.9194**	**0.9167**	**0.8333**

Table 6.16: Correlation results between *MOS* and *RG* values of the different metrics to evaluate image quality using the *Gaussian Noise* distorted 9 images.

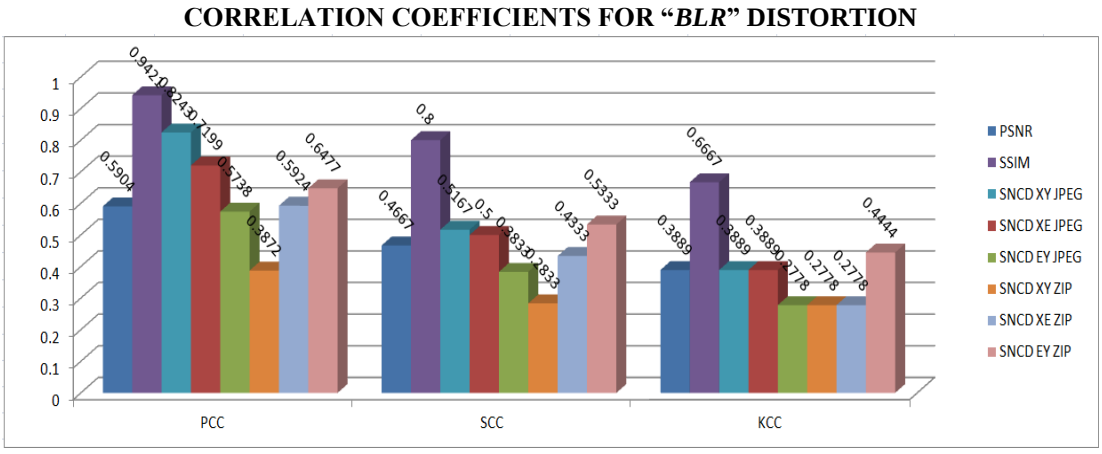


Figure 6.25: Summary results of Table 6.11.

CORRELATION COEFFICIENTS FOR "*DCQ*" DISTORTION

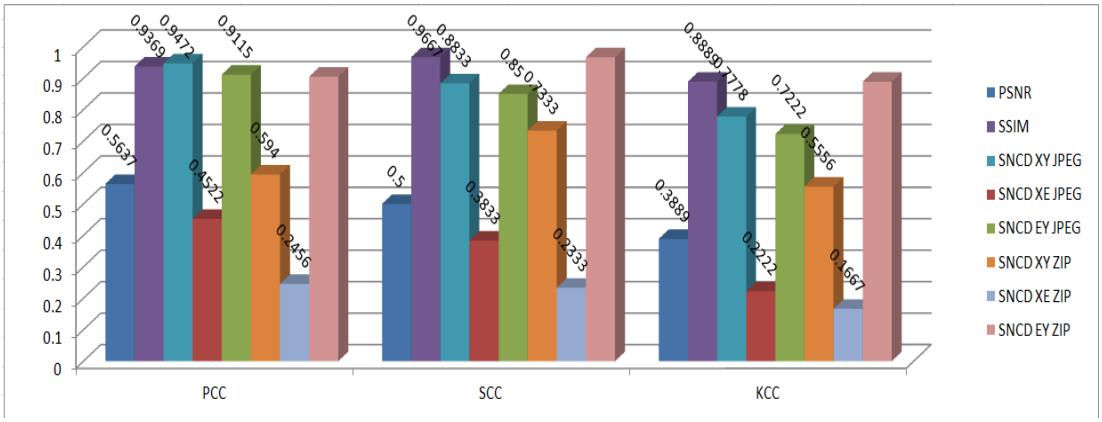


Figure 6.26: Summary results of Table 6.12.

CORRELATION COEFFICIENTS FOR "*FLT*" DISTORTION

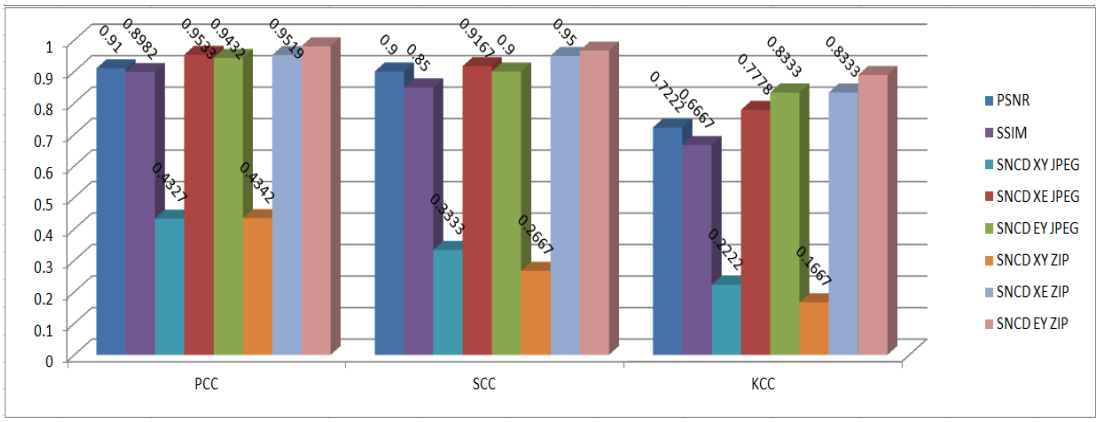


Figure 6.27: Summary results of Table 6.13.

CORRELATION COEFFICIENTS FOR "*JP2*" DISTORTION

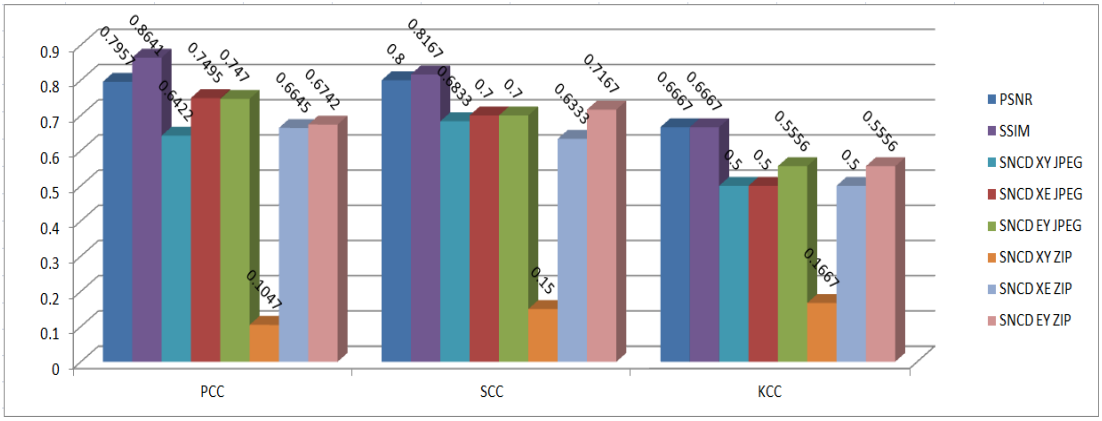


Figure 6.28: Summary results of Table 6.14.

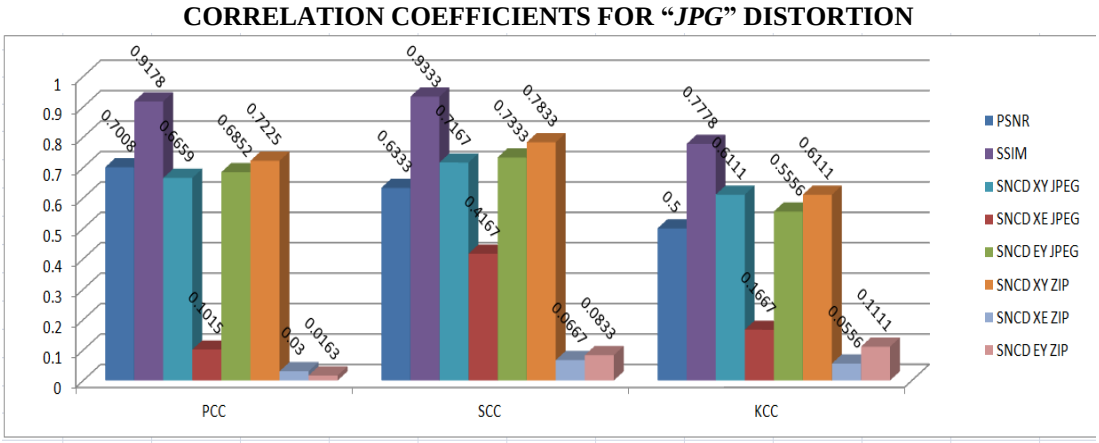


Figure 6.29: Summary results of Table 6.15.

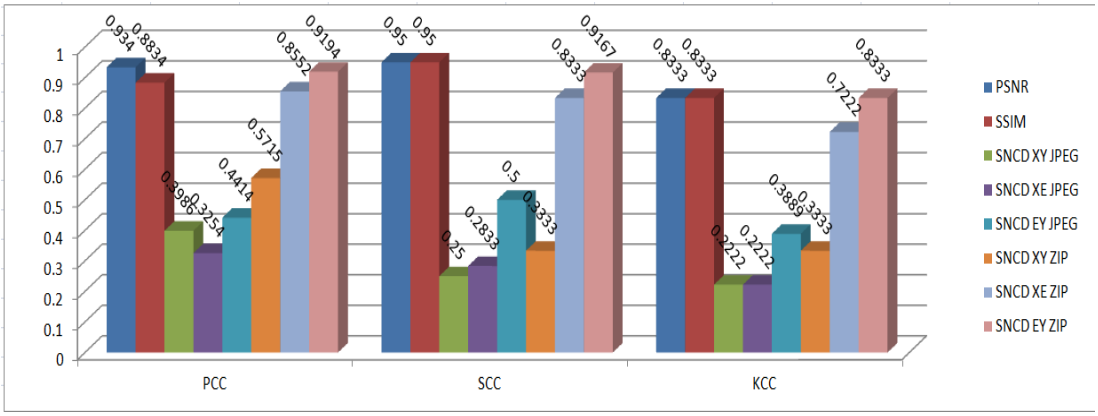


Figure 6.30: Summary results of Table 6.16.

The results of this third experiment grouped by the type of distortion are very interesting. We have encouraging results for the *SNCD*. The performance of the *SNCD* has improved considerably in all cases. It outperforms the traditional metrics *SSIM* and *PSNR* for the *DCQ* case and for the *FLT* case; however, for the remaining distortion cases, the obtained values are quite comparable to the classical metrics (see Figures 6.25 to 6.30).

In addition, for all distortions cases based on compression, the performance of *SNCD* is inferior as this method is based on data compression, and therefore, cannot identify the compression distortions, but still shows very comparable values.

In the third experiment where we sub-divided the database by type of distortion we have good results for *SNCD*. Why we do not have the same results when we work with the database sub-divided per parent image, or when working with the entire database? A reason may be that the *SNCD* method properly evaluates the distortion or quality of the images, but does not consider the magnitude of the type of distortion for the entire database. This means

that for the subjective assessment, some kind of distortion is more influential than another. In contrast, during *SNCD* computation, the sequence of distortion types can be rearranged; however, the *SNCD* determines with good approximation the intensity of the type of distortion. This holds for all results shown in the different tables.

We ran a fourth experiment to generate and analyze distorted images with the same or about the same mean squared error *MSE*. For this experiment, we take the original images of Figure 6.20 and create distorted images. We calculate the measure of quality of the images of the new database (9 distorted images for the three original images shown in Figure 6.20; the distortions for each original image are: *baseline JPEG compression*, *JPEG 2000 compression* and *Added Noise*; all distortions have about the same *MSE* values between 3200 and 3400 and *PSNR* values between 22 and 24; these distorted images are shown in Figure 6.31) and compared them using the correlation coefficients explained above. The results are shown in Table 6.17 and in Figure 6.32.

Figure 6.31: Distorted images of the Cornell-A57 database with nearly the same *MSE* and *PSNR*.

CORRELATION RESULTS OF THE 4th EXPERIMENT			
	PCC	***SCC***	***KCC***
PSNR	**0.7722**	**0.9160**	**0.8003**
SSIM	0.0576	0.0672	0.0572
SNCD XY JPEG-LS	0.2219	0.1092	0.1143
SNCD XE JPEG-LS	0.7333	0.6555	0.5145
SNCD EY JPEG-LS	0.0536	0.4034	0.1715
SNCD XY ZIP	**0.5946**	**0.6471**	**0.5145**
SNCD XE ZIP	0.7496	0.7311	0.5717
SNCD EY ZIP	0.0345	0.2185	0.1143

Table 6.17: Correlation results between *MOS* and *RG* values of the different metrics to evaluate the 9 distorted images of Figure 6.31.

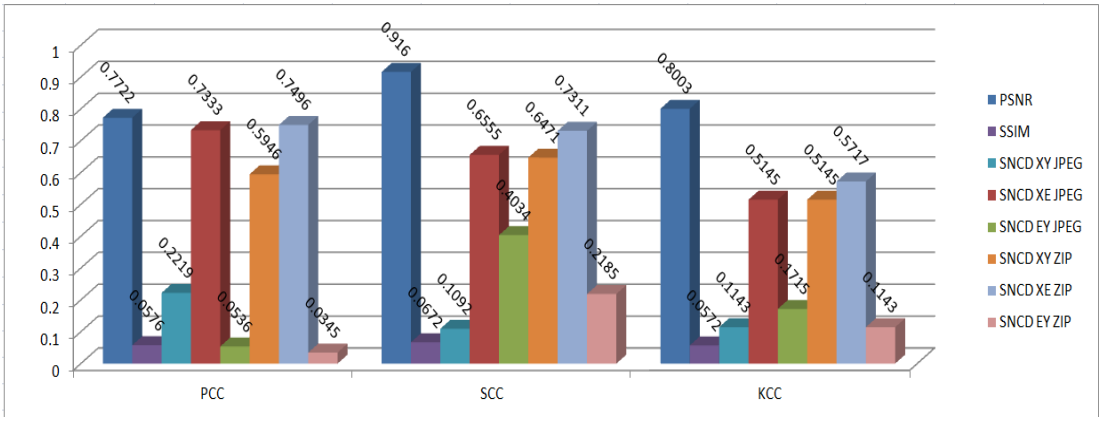


Figure 6.32: Summary results of Table 6.17.

In order to evaluate this fourth experiment we need a subjective assessment; this subjective assessment was obtained from six imaging experts. The subjective evaluation leads to values between zero and one. Values closer to one are for highly distorted images, values closer to zero are for images with low distortion. We can see that *PSNR* yields the best results; we obtain a *Pearson* correlation of 0.7722, a *Spearman* correlation of 0.9160, and a *Kendall* correlation of 0.8003. The results obtained for the *SNCD* are not so good but better than those obtained for the complete database of 54 images; for *SNCD XY* we obtain a *Pearson* correlation of 0.5946, a *Spearman* correlation of 0.6471, and a *Kendall* correlation of 0.5145. Although we can imagine that *PSNR* would not work properly because all images have about the same *PSNR*, the small variations in the *PSNR* values are sufficient to reflect the subjective evaluation of images; that is why we obtain a better result using the *PSNR* metric (see Figure 6.32).

Chapter 7

Conclusions and Discussion

7.1 Conclusions

In this thesis, we proposed new methodologies and approaches for artifact analysis and artifact detection in optical satellite images. We also proposed approaches based on Kolmogorov complexity and information theory to better understand the behavior of artifacts and their influence on the classification and indexing process.

After making a study of the acquisition chain of a satellite image and the different processes through which an image passes, we have identified potential sources of artifact generation. By analyzing the satellite image quality and considering its importance for a good classification or indexing process, we could see that artifacts have a negative influence on them. Artifacts mainly affect the statistical parameters of an image so we established a relationship between information theory, Kolmogorov complexity, and the different methods for message insertion in multimedia data such as watermarking, steganography, and image fakery since they create effects similar to the presence of artifacts in satellite images. Based on these findings, we developed three new approaches for artifact analysis and detection; the first approach uses an experimentally derived *RD* curve to study error maps; the second approach uses the *NCD* to study similarities; our third approach uses different distortion measures to obtain new experimental curves to analyze error maps. The new experimental curves are similar to *RD* curves; for instance, we developed the *CEM* curve, where we used the *SNCD* as a measure of similarity.

We applied the methods to two databases, one for images with synthetic artifacts, and another one with actual instrumental artifacts. The former is based on images of three land cover categories (*city*, *forest*, and *sea*) and contains striping and aliasing artifacts. The latter contains images of the MERIS, IKONOS, SPOT and ROSIS instruments with a variety of artifacts. In order to handle all data efficiently in conjunction with all artifact detection methods, we also developed a toolbox supporting all methods.

For artifact detection, the methods that we propose yield better results than an already existing method which uses well-known image quality metrics. When we compare rate-distortion (*RD*) analysis with normalized compression distance (*NCD*), rate-distortion analysis provides better results. The detection of artifacts depends on the environment we are

working in. In *sea* images it is easier to detect strip artifacts but not aliasing artifacts, while in the *city* and the *forest* images it is easier to detect aliasing.

The results with hyperspectral images that have a high radiometric resolution are not good when striping artifacts are contained in the less significant bits and cannot be detected. To improve this result, we converted images into bit planes and reformatted them using horizontal or vertical scanning; acceptable results for hyperspectral images are found with vertical (column-wise) scanning when the striping artifacts have the same orientation.

The approximation of the Rate-Distortion curve based on the Kolmogorov complexity, using the Symmetric Normalized Compression Distance *SNCD*, is a good approximation and, together with the residuals or error map analysis gives us the ability to analyze the behavior of the compressor with different compression parameters. The results help us to discriminate between different types of images as each group has a different behavior when calculating the *SNCD*.

When we look at *SNCD* for image quality assessment, our initial expectations are not fulfilled: the traditional image quality assessment metrics show a better performance than our *SNCD* method. The reason for this is that the good classification performance of *SNCD* cannot be exploited by details created by typical distortions.

When we analyze the performance of the *SNCD* and *NCD* by comparing an original image with a degraded version of this image, the obtained values for *SNCD* and *NCD* are almost identical. For other applications where different images are compared, the variation between the value of the *SNCD* and *NCD* can be significant. The *SNCD* as a metric for assessing image quality is limited to a single type of distortion with different levels of intensity.

7.2 Discussion

The selection of the compressor to be used for approximating the Kolmogorov Complexity needs some additional discussion. In principle, it could be any compressor, supposing that a better compressor gives a better approximation to Kolmogorov Complexity; however, it would be better to use a compressor being adapted to the type of data to be analyzed. Then we can fully exploit the type of redundancy that can occur in the data. If we use a ZIP compressor, we exploit the sequential redundancy in a string, while a JPEG compressor can exploit the two-dimensional spatial redundancy in an image. Data in general may have different types of redundancy. Ideally, one could exploit all kinds of redundancy and get a more reliable approximation to the Kolmogorov Complexity. In Figure 7.1, we see an illustration of this idea where the data contain several types of redundancy *R1*, *R2*, *R3*, … and the remaining part would be random data *A*. In reality, however, we are more or less limited to the use of an existing established method.

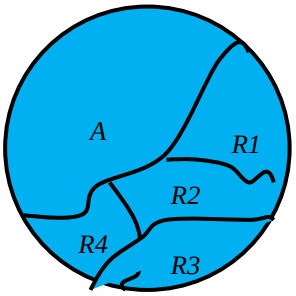


Figure 7.1: Redundancy in the data.

Another important aspect to be discussed is the calculation of the *C(x,y)* function for *SNCD* or *NCD* where one uses a concatenation of the data *x* and *y*. As we have a number of options how to concatenate *x* and *y* (e.g., along lines or columns, or by interspersing data), the concatenation process should be related to the type of redundancy to be exploited. If we use a ZIP compressor, a sequential concatenation by would be most appropriate; however, if we use a JPEG compressor that exploits spatial two-dimensional redundancy, a sequential concatenation is no longer suitable, and we have to find a more appropriate process for the calculation of *C(x, y)*. This can be done by the following equation that introduces a neutral element.

$$SNCD(x, y) = \frac{C(x, y) + C(y, x) - 2 \times \min\{C(x, \phi), C(y, \phi)\}}{2 \times \max\{C(x, \phi), C(y, \phi)\}} \tag{7.1}$$

where: ϕ is a neutral element.

The neutral element for the concatenation process would be an empty string; but if we extract a dictionary from data *y* to compress data *x* in order to calculate *C(x,y)*, the neutral element would be *x* or *y*, leaving the expression as follows:

$$SNCD(x, y) = \frac{C(x, y) + C(y, x) - 2 \times \min\{C(x, x), C(y, y)\}}{2 \times \max\{C(x, x), C(y, y)\}}. \tag{7.2}$$

The process explained above is presented in Figure 7.2. The *C(x, y)* function receives the data *x* and *y*. The first step is to extract a dictionary d_y from data *y*. After that, we do the encoding of data *x* using the dictionary d_y giving as a result a compressed version *x'*. Finally, we calculate the inverse compression ratio. We divide the size of the compressed version by the size of the original version ($|x'| / |x|$).

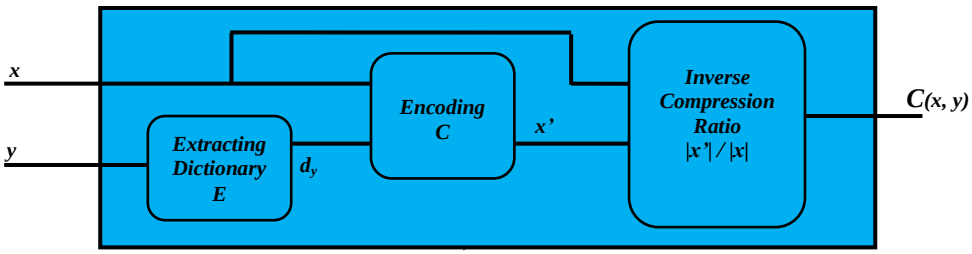


Figure 7.2: Concatenation process.

This topic, however, is still open for further research.

Bibliography

Alber, E., Farber, M., Yeager, N., Xiong, Z. & Pottenger, W. M. (2001), Retrieval of Multi and Hyperspectral Images Using an Interactive Relevance Feedback form of content-based image retrieval, *Proceedings of SPIE AeroSense Conference*, Orlando, Florida.

Avcibas, I., Memon, N. & Sankur, B. (2003). Steganalysis Using Image Quality Metrics, *IEEE Transactions on Image Processing*, vol. 12, N° 2, pp. 221-229.

Avcibas, I., Sankur, B. & Sayood, K. (2002). Statistical Evaluation of Image Quality Measures, *Journal of Electronic Imaging*, vol. 11, N° 2, pp. 206-223.

Avcibas, I. (2001). Image Quality Statistics and Their Use in Steganalysis and Compression, *PhD Thesis – Bogazici University*, Istanbul/Istanbul Province, Turkey.

Betancourt, G. A. (2005). Las Maquinas de Soporte Vectorial (SVMs), *Scientia et Technica Año XI*, No 27, UTP.

Blahut, R. (1972). Computation of Channel Capacity and Rate Distortion Functions, *IEEE Transactions on Information Theory*, vol. 18, N°4, pp. 460-473.

Blanchet, G. (2006). Study of the Blur, Ringing and Aliasing Artifacts in Numerical Imaging, Application to Restoration, *PhD Thesis - Ecole Normale Supérieur de Cachan*, Cachan, France.

Blanchet, G., Moisan, L. & Rougé, B. (2009). A-Contrario Ringing Detection and Shannon-Compliant Images, *preprint MAP5*.

Blanchet, G., Moisan, L. & Rougé, B. (2010). Automatic Detection of Well Sampled Images Via a New Ringing Measure, *IEEE International Conference on Acoustics, Speech and Signal Processing – ICASSP*, Dallas, Texas, United State of America, pp. 1030-1033.

Brooks, A. C., Zhao, X. & Pappas T. N. (2008). Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortion, *IEEE Transactions on Image Processing*, vol. 17, N° 8, pp. 1261-1273.

Brunet, D., Vrscay E. R. & Wang, Z. (2012). On the Mathematical Properties of the Structural Similarity Index, *IEEE Transactions on Image Processing*, vol. 21, N° 4, pp. 1488-1499.

Burkovski, A., Klenk, S. & Heidemann, G. (2011). Similarity Calculation with Length Delimiting Dictionary Distance, *IEEE International Conference on Tools with Artificial Intelligence*, Florida, United States of America, pp. 856-864.

Cachin, C. (2004). An Information-Theoretic Model for Steganography, *Information and Computation*, vol. 192, pp. 41-56.

Campana, B. J. L. & Keogh, E. J. (2010). A Compression Based Distance Measure for Texture, *Statistical Analysis and Data Mining*, vol. 3, N° 6, pp. 381-398.

Campedel, M., Moulines, E., Maitre, H. & Datcu, M. (2005), Features Selection for Satellite Image Indexing, *ESA-EUSC: Image Information Mining*, Frascati, Italy.

Cebrian, M., Alfonseca, M. & Ortega, A. (2005). Common Pitfalls Using the Normalized Compression Distance: What to Watch out for in a Compressor, *Communication in Information and Systems*, vol. 5, N° 4, pp. 367-384.

Celik, U., Sharma, G., & Murat, A. (2004). Universal Image Steganalysis Using Rate Distortion Curves, *SPIE Proceedings of Security, Steganography, and Watermarking of Multimedia Contents VI*, vol. 5306, pp. 19-22.

Cerra, D. & Datcu, M. (2011). Algorithmic Relative Complexity, *Entropy*, vol. 13, pp. 902-914.

Cerra, D. (2010). Pattern Oriented Algorithmic Complexity Towards Compression Based Information Retrieval, *PhD. Thesis - TELECOM ParisTech*, Paris, France.

Cerra, D., Mallet, A., Gueguen, L. & Datcu, M. (2010). Algorithmic Information Theory-Based Analysis of Earth Observation Images: An Assessment, *IEEE Geoscience and Remote Sensing Letters*, vol. 7, N° 1, pp. 8–12.

Cerra, D. & Datcu, M. (2010). Compression-based Hierarchical Clustering of SAR Images, *Remote Sensing Letters*, vol. 1, N° 3, pp. 141-147.

Cerra, D., Mallet, A., Gueguen, L. & Datcu, M. (2008), Complexity Based Analysis of Earth Observation Imagery, an Assessment, *ESA EUSC*, Frascati, Italy.

Chandler, M. D. & Hemami, S. S. (2007). VSNR: A Wavelet-based Visual Signal-to-Noise Ratio for Natural Images, *IEEE Transactions on Image Processing*, vol. 16, N° 9, pp. 2284-2298.

Chang, I., Yu, J. C. & Chang, C. (2012). A Forgery Detection Algorithm for Exemplar-Based Inpainting Images Using Multi-Region Relation, *Image and Vision Computing*, vol. 33, N° 1, pp. 57-71.

Cheeseman, P. & Stutz, J. (1996), Bayesian Classification (AUTOCLASS): Theory and Results, *Advances in Knowledge Discovery and Data Mining, AAI/MIT Press*, USA, pp. 61-83.

Chen, M., Fridrich, J., Goljan, M. & Lucas, J. (2008). Determining Image Origin and Integrity Using Sensor Noise, *IEEE Transactions on Information Theory*, vol. 3, N° 1, pp. 74–90.

Chen, Y. & Hsu, C. (2011). Detecting Recompression of JPEG Images via Periodicity Analysis of Compression Artifacts for Tampering Detection, *IEEE Transactions on Information Forensics and Security*, vol. 6, N° 2, pp. 396–406.

Cilibrasi, R. (2007). Statistical Inference Through Data Compression, PhD. Thesis - Institute for Logic, *Language and Computation Universiteit Van Amsterdam*, Amsterdam, Netherlands .

Cilibrasi, R. & Vitanyi, P. (2005). Clustering by Compression, *IEEE Transactions on Information Theory*, vol. 51, N° 4, pp. 1523–1545.

Cilibrasi, R., Cruz, A., de Rooij, S. & Keijzer, M. (2003). The CompLearn Toolkit, *Available online at complearn.sourceforge.net*.

Comaniciu, D., & Meer, P. (2002). Mean Shift A Robust Approach Toward Features Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, N° 5, pp. 603-619.

Cozzolino, D., Poggi, G., Sansone, C. & Verdoliva, L. (2012). A Comparative Analysis of Forgery Detection Algorithms, *SSPR & SPR Springer-Verlag Berlin Heidelberg*, pp. 693-700.

Davison, L. D. (1972). Rate-Distortion Theory and Application, *Proceedings of IEEE*, vol. 60, N° 7, pp. 800-808.

De Rooij, S. & Vitanyi, P. M. B. (2012). Approximating Rate-Distortion Graphs of Individual Data: Experiments in Lossy Compression and Denoising, *IEEE Transactions on Computers*, vol. 63, N° 3, pp. 395-407.

Dowman, I. J. & Peacegood, G. (1989). Information Content of High Resolution Satellite Imagery, *Photogrammetria (PRS)*, vol. 43, pp. 295-310.

Duan, F. & Wang, Y. (2010). A Nonlinear Detector for Optimal DCT-domain Watermark Detection, *IEEE International Conference on Electrical and Control Engineering*, Wuhan, China, pp. 176-178.

Dufaux, F., Sullivan, G. J. & Ebrahimi, T. (2009). The JPEG XR Image Coding Standard, *IEEE Signal Processing Magazine*, vol. 26, N° 6, pp. 195-204.

Farid, H. (2009). Image Forgery Detection, *IEEE Signal Processing Magazine*, vol. 26, N° 2, pp. 16-25.

Fridrich, J. (2009). Digital Image Forensics, *IEEE Signal Processing Magazine*, vol. 26, N° 2, pp. 26-37.

Gloe, T. (2011), Demystifying Histograms of Multi-Quantised DCT Coefficients, *IEEE International Conference on Multimedia and Expo - ICME*, Barcelona, Spain, pp. 1-6.

Golomb, S. W. (1966). Run Length Encodings, *IEEE Transactions on Information Theory*, July, pp. 399-401.

Gomez-Chova, L., Zurita-Milla, R., Alonso, L., Amoros-Lopez, J., Guanter, L. & Camps-Valls, G. (2011). Gridding Artifacts on Medium-Resolution Satellite Image Time Series: MERIS Case Study, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, N° 7, pp. 2601-2611.

Gruenwald, P. (2000). Model Selection Based on Minimum Description Length, *Journal of Mathematical Psychology*, vol. 44, N° 1, pp. 133-152.

Gruenwald, P. (2007). The Minimum Description Length Principle, *The MIT Press*, Massachusetts, United State of America.

Gruenwald, P. & Vitanyi, P. (2004). Shannon Information and Kolmogorov Complexity, *arXiv preprint cs.IT/0410002*.

Gruenwald, P. & Vitanyi, P. (2008). Algorithmic Information Theory, *Handbook of the Philosophy of Information*, pp. 281-320.

Gueguen, L. (2007). Extraction d'Information et Compression Conjointes des Séries Temporelles d'Images Satellitaires, *PhD. Thesis - TELECOM ParisTech*, Paris, France.

Haralick, R. M. (1979). Statistical and Structural Approaches to Textures, *IEEE Proceedings*, vol. 67, pp. 786-804.

Haralick, R. M., Shanmugan, K. & Dinstein, I. (1973). Textural Features for Image Classification, *IEEE Transactions on System, Man and Cybernetics*, vol. 6, N° 3, pp. 610-621.

Helder, D. L. & Ruggies, T. A. (2004). Landsat Thematic Mapper Reflective-Band Radiometric Artifacts, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, N° 12, pp. 2704-2716.

Huffman, D. A. (1952), A Method for the Construction of Minimum-Redundancy Codes, *Proceedings of the I.R.E.*, pp 1098–1102.

Hyun Bae, S., Pappas, T. N. & Juang, B. H. (2009). Subjective Evaluation of Spatial Resolution and Quantization Noise Tradeoffs, *IEEE Transactions On Image Processing*, vol. 18, N°. 3, pp. 495-508.

Iwata, M., Hori, T., Shiozaki, A. & Ogihara, A. (2010). Digital Watermarking Method for Tamper Detection and Recovery of JPEG Images, *IEEE International Symposium on Information Theory and its Applications - ISITA*, Taichung, Taiwan, pp. 309-314.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999). Data Clustering, A Review, *ACM Computing Surveys*, vol. 31, N° 3, pp. 264-323.

Jain, A. & Farrokhnia, F. (1991). Unsupervised Texture Segmentation using Gabor Filters, *Pattern Recognition*, vol. 12, N° 24, pp. 1167-1186.

Jegou, H., Schmid, C., Harzallah, H. & Verbeek, J. (2010). Accurate Image Search Using the Contextual Dissimilarity Measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2-11.

Jha, R. K., Biswas, P. K. & Mishra, D. (2010). Improving Watermark Detection Performance Using Suprathreshold Stochastic Resonance, *IEEE TENCON*, Fukuoka, Japan, pp. 691-696.

Jung, H., Won, J. S. & Kang, M. H. (2010). Detection and Restoration of Defective Lines in the SPOT 4 SWIR Band, *IEEE Transactions on Image Processing*, vol. 19, N° 8, pp. 2143-2156.

Keogh, E., Lonardi, S. & Ratanamahatana, Ch. (2004). Towards Parameter Free Data Mining, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, pp. 206-215.

Kiely, A. R. & Klimech, M. A. (2009). Exploiting Calibration-Induced Artifacts in Lossless Compression of Hyperspectral Imagery, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, N° 8, pp. 2672-2678.

King, R. L. Hyperspectral Earth Observation Imaging and Analysis an Agricultural Perspective, *ROSA Summer School*, Sinaia - Romania.

Kirchner, J. W. (2005), Aliasing in $1/f^{\alpha}$ Noise Spectra: Origins, Consequences, and Remedies, *Physical Review E*, vol. 71, N° 6, 066110.

Kitakami, M. & Tai, K. (2009), Lossless Image Compression by PPM-Based Prediction Coding, *Data Compression Conference*, Utah, USA, pp. 452.

Kwok, T., Sheung, H. & Wang, C. C. L. (2010). Fast Query for Exemplar-Based Image Completion, *IEEE Transactions on Image Processing*, vol. 19, N° 12, pp. 3106-3115.

Lempel, A. & Ziv, J. (1976). On the Complexity of Finite Sequences, *IEEE Transactions on Information Theory*, vol. IT 22, N° 1, pp. 75 – 81.

Li, M. & Vitanyi, P. (2008). An Introduction to Kolmogorov Complexity and its Applications, *Springer-Verlag*, New York.

Li, M., Chen, X., Ma, X. B. & Vitanyi, P. M. B. (2004). The Similarity Metric, *IEEE Transactions on Information Theory*, vol. 50, N°. 12, pp. 3250-3264.

Li, W., Yuan, Y. & Yu, N. (2009). Passive Detection of Doctored JPEG Image Via Block Artifacts Grid Extraction, *Signal Processing*, vol. 89, N°. 9, pp. 1821-1829.

Lier, P., Valorge, C. & Briottet, X. (2008). Imagerie Spatiale : Des principes d'acquisition au traitement des images optiques pour l'observation de la Terre, *Cepadues*, France, pp.490.

Liu, Q., & Ying, J. (2012). Grayscale Image Digital Watermarking Technology Based on Wavelet Analysis, *IEEE Symposium on Electrical & Electronics Enginering - EEESYM*, Kuala Lumpur, Malaysia, pp. 618-621.

Luo, W., Wang, Y. & Huang, J. (2010). Detection of Quantization Artifacts and Its Applications to Transform Encoder Identification, *IEEE Transactions on Information Forensics and Security*, vol. 5, N° 4, pp. 810-815.

Lyu, S. & Farid, H. (2006). Steganalysis Using Higher-Order Image Statistics, *IEEE Transactions on Information Forensics and Security*, vol. 1, N° 1, pp. 111-119.

Mackay, D. (2003). Information Theory, Inference, and Learning Algorithms, *Cambridge University Press*.

Mahdian, B. & Saic, S. (2010). Blind Methods for Detecting Image Fakery, *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, N° 4, pp. 18-24.

Mahdian, B. & Saic, S. (2008a). Blind Authentication Using Periodic Properties of Interpolation, *IEEE Transactions on Information Forensics and Security*, vol. 3, N° 3, pp. 529-538.

Mahdian, B. & Saic, S. (2008b). Detection of Resampling Supplement with Noise Inconsistencies Analysis for Image Forensics, *IEEE International Conference on Computational Sciences and Its Applications ICCSA 2008*, Perugia, Italy, pp. 546-556.

Malacara, D. & Thompson, B. J. (2001). Handbook of Optical Engineering, *CRC Press*.

Mallet, A. & Datcu, M. (2008a). Rate Distortion Based Detection of Artifacts in Earth Observation Images, *IEEE Geoscience and Remote Sensing Letters*, vol. 5, N°. 3, pp. 354-358.

Mallet, A. & Datcu, M. (2008b). Model Free Earth Observation Image Artifact Detection, *Geoscience and Remote Sensing Syposium – IGARSS*, Boston, USA, pp. IV-549.

Mallet, A., Gueguen, L. & Datcu, M. (2008). Complexity Based Image Artifact Detection, *Data Compression Conference*, Utah, USA, pp. 534.

Manjunath, B. S. & Ma, W. Y. (1996) Texture Features for Browsing and Retrieval of Image Data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, N° 8, pp. 837-842.

Mannos, J. L. & Sakrison, D. J. (1974). The Effects of a Visual Fidelity Criterion on the Encoding of Images, *IEEE Transactions on Information*, vol. 20, N°. 4, pp. 525-536.

Marsetic, A., Kokalj, Z. & Ostir, K. (2011), The Effect of Lossy Image Compression on Object Based Image Classification – WorldView – 2 Case Study, *International Archives of Photogrammetry, Remote Sensing and Spatial Sciences*, vol. XXXVIII-4/W19, pp. 187-192.

Megalingam, R. K., Nair, M. M., Srikumar, R., Balasubramanian, V. K. & Sarma, V. (2010), Performance Comparison of Novel, Robust Spatial Domain Digital Image Watermarking with the Conventional Frequency Domain Watermarking Technique, *International Conference on Signal Acquisition and Processing*, Bangalore, India, pp. 349-353.

Mittal, A., Moorthy, A. K. & Bovik, A. C. (2011), Blind/Referenceless Image Spatial Quality Evaluator, *IEEE Conference on Signals, Systems and Computers - ASILOMAR*, Pacific Grove, CA, USA, pp. 723-727.

Mittal, A., Muralidhar, G. S., Ghosh, J. & Bovik, A. C. (2012), Blind Image Quality Assessment Without Human Training Using Latent Quality Factors, *IEEE Signal Processing Letters*, vol. 19, N° 2, pp. 75-78.

Moulin, P. & O'Sullivan, J. A. (2003). Information-Theoretic Analysis of Information Hiding, *IEEE Transactions on Information Theory*, vol. 49, N° 3, pp. 563-593.

Moorthy, A. K. & Bovik, A. C. (2010), A Two-Step Framework for Constructing Blind Image Quality Indices, *IEEE Signal Processing Letters*, vol. 17, N° 5, pp. 513-516.

Muramatsu, J. & Kanaya, F. (1994). Distortion-Complexity and Rate-Distortion Function, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. 77, N° 8, pp. 1224-1229.

Nannen, V. (2010). A Short Introduction to Kolmogorov Complexity, *arXiv preprint arXiv:1005.2400*.

Nikvand, N. & Wang, Z. (2010), Generic Image Similarity Based on Kolmogorov Complexity, *IEEE International Conference on Image Processing – ICIP*, Hong Kong, pp. 309-3012.

Nikvand, N. & Wang, Z. (2011), Perceptual Normalized Information Distance for Image Distortion Analysis Based on Kolmogorov Complexity, *International Conference on Applied Mathematics, Modeling and Computational Science - AMMCS*, Ontario, Canada, pp. 321.

Nill, N. B. (1985), A Visual Model Weighted Cosine Transform for Image Compression and Quality Assessment, *IEEE Transactions on Communications*, vol. 33, N° 6, pp. 551-557.

Ostergaard, J., Derpich, M. S. & Channappayya, S. S. (2011), The High-Resolution Rate-Distortion Function Under the Structural Similarity Index, *EURASIP Journal on Advances in Signal Processing*, ID 857959.

O'Sullivan, J. A. (1998). Information Theoretic Image Formation, *IEEE Transactions on Information Theory*, vol. 44, N° 6, pp. 2094- 2123.

Pappas, T. N., Zujovic, J. & Neuhoff, D. L. (2010). Image Analysis and Compression: Renewed Focus on Texture, *Visual Information Processing and Communication*, Proc. SPIE vol. 7543.

Pennebaker, W. B. & Mitchel, J. L. (1992). JPEG: Still Image Data Compression Standard, *Kluwer Academic Pub.*

Pitas, I. (2000), Digital Image Processing Algorithms and Applications, *Wiley*, New York, United State of America.

Popescu, A. C. & Farid, H. (2005). Exposing Digital Foregeries by Detecting Traces of Resampling, *IEEE Transactions on Signal Processing*, vol. 53, N° 2, pp. 758-767.

Presutti, M. (2004). La Matriz de Co-Ocurrencia en la Clasificación Multiespectral: Tutorial para la Enseñanza de Medidas Texturales en Cursos de Grado Universitario, *4ª Jornada de Educação em Sensoriamento Remoto no Âmbito do Mercosul*, São Leopoldo, RS, Brasil.

Quispe-Ayala, M. R., Asalde-Alvarez, K. & Roman-Gonzalez, A. (2010). Image Classification Using Data Compression Techniques, *IEEE 26th Convention of Electrical and Electronics Engineers in Israel - IEEEI*, Eilat, Israel, pp. 349-353.

Rabin, J., Delon, J. & Gousseau, Y. (2011). Removing Artefacts From Color and Contrast Modifications, *IEEE Transactions on Image Processing*, vol. 20, N° 11, pp. 3073-3085.

Rajashekar, U., Bovik, A. C. & Cormack, L. K. (2006). Visual Search in Noise: Revealing the Influence of Structural Cues by Gaze-contingent Classification Image Analysis, *Journal of Vision*, vol. 6, N° 4.

Reyes, M. G., Zhao, X., Neuhoff, D. L. & Pappas, T. N. (2007). Lossy Compression of Bilevel Images Based on Markov Random Fields, *Proc. Int. Conf. Image Processing (ICIP)*, vol. 2, Texas, USA, pp. II-373 – II-376.

Reyes, M. G., Zhao, X., Neuhoff, D. L. & Pappas, T. N. (2008). Structure-preserving Properties of Bilevel Image Compression, *Human Vision and Electronics Imaging XIII*, Proc. SPIE vol. 6806.

Reynar, J. C., Herz, F., Eisner, J., & Ungar, L. (1999), "Lempel-Ziv Data Compression Technique Utilizing a Dictionary Pre-filled with Frequent Letter Combinations, Words and/or Phrases", *Patent and Trademark Office*, U.S. Patent No. 5,951,623. Washington, DC: U.S.

Rigau, J., Feixas, M. & Sbert, M. (2008). Informational Aesthetics Measures, *IEEE Computer Graphics and Applications*, vol. 28, N° 2, pp. 24-34.

Roman-Gonzalez, A. & Datcu, M. (2011a). Satellite Image Artifacts Detection Based on Complexity Distortion, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2011)*, Vancouver, Canada, pp. 1437-1440.

Roman-Gonzalez, A. & Datcu, M. (2011b). Data Cleaning: Approaches for Earth Observation Image Information Mining, *ESA-EUSC-JRC 2011 Image Information Mining: Geospatial Intelligence from Earth Observation Conference*, Ispra, Italy, pp. 117-120.

Roman-Gonzalez, A., Veganzones, M. A., Graña, M. & Datcu, M. (2011c). A Novel Data Compression Technique for Remote Sensing Data Mining, *ESA-EUSC-JRC 2011 Image Information Mining: Geospatial Intelligence from Earth Observation Conference*, Ispra, Italy, pp. 101-104.

Roman-Gonzalez, A. & Datcu, M. (2010). Parameter Free Image Artifacts Detection: A Compression Based Approach, *Proceedings SPIE Remote Sensing*, vol. 7830, 783008, Toulouse, France.

Roman-Gonzalez, A. & Vargas-Cuentas, N. I. (2013). Contenido y Calidad de las Imágenes de Observación Terrestre, *Revista ECIPeru*, vol. 9, N° 2, pp. 15-21.

Roman-Gonzalez, A. (2013a). The SNCD as a Metrics for Image Quality Assessment, *International Journal of Advanced Computer Science and Applications – IJACSA*, vol. 4, N° 7, pp. 40-47.

Roman-Gonzalez, A. (2013b). Compression Techniques for Image Processing Tasks, *International Journal of Advanced Research in Computer Science and Software Engineering – IJARCSSE,* vol. 3, N° 7, pp. 379-388.

Roman-Gonzalez, A., Reynaga-Cardenas, C. J. & Ganvini-Valcarcel, C. (2013). Método General para la Detección de Imágenes Alteradas Utilizando Técnicas de Compresión, *Revista ECIPeru*, vol. 10, N° 1, pp. 14-23.

Roman-Gonzalez, A. & Asalde-Alvarez, K. (2012). Image Processing by Compression: An Overview, *World Congress on Engineering and Computer Science – WCECS,* San Francisco, USA, pp. 650-654.

Roman-Gonzalez, A. (2012a). Clasificación de Datos Basado en Compresión, *Revista ECIPeru,* vol. 9, N° 1, pp. 69-74.

Roman-Gonzalez, A. (2012b). Digital Images Analysis, *Revista ECIPeru,* vol. 9, N° 1, pp. 61-68.

Saad, M. A., Bovik, A. C. & Charrier, C. (2010). A DCT Statistics-Based Blind Image Quality Index, *IEEE Signal Processing Letters*, vol. 17, N° 6, pp. 583-586.

Samuel, S. & Penzhom, W. T. (2004). Digital Watermarking for Copyright Protection, *IEEE AFRICON*, pp. 953-957.

Shannon, C. (1948). A Mathematical Theory of Communication, *The Bell System Technical Journal* 27:379-423, 623-656.

Sheikh, H. R., Sabir, M. F., & Bovik, A. C. (2006a). A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms *IEEE Transactions on Image Processing*, vol. 15, N°. 11, pp. 3441-3451.

Sheikh, H. R. & Bovik, A. C. (2006b). Image Information and Visual Quality, *IEEE Transactions on Image Processing*, vol. 15, N° 2, pp. 430-444.

Simoncelli, E. P. & Adelson E. H. (1989). Nonseparable QMF Pyramids, *Proc. SPIE, Visual Comm and Image Proc. IV*, vol. 1199.3, pp. 1242-1246.

Song, H., Qiu, Z. & Gu, J. (2010). A Novel Semi-fragile Image Watermarking Scheme Based on Wavelet, *IEEE International Conference on Audio, Language and Image Processing*, Shanghai, China, pp. 1504-1510.

Sow, D. M. & Eleftheriadis, A. (2003). Complexity Distortion Theory, *IEEE Transactions on Information Theory*, vol. 49, N° 3, pp. 604-608.

Sow, D. & Eleftheriadis, A. (2000). Approximation of the Resource Bounded Complexity Distortion Function, *IEEE International Symposium on Information Theory*, Sorrento, Italy, pp. 206.

Stein, D. W. J., Beaven, S. G., Hoff, L. E., Winter, E. M., Schaum, A. P. & Stocker, A. D. (2002). Anomaly Detection from Hyperspectral Imagery, *IEEE Signal Processing Magazine*, vol. 19, N° 1, pp. 58-69.

Sullivan, K., Madhow, U., Chandrasekaran, S. & Manjunath, B. S. (2006). Steganalysis for Markov Cover Data with Application to Images, *IEEE Transactions on Information Forensics and Security*, vol. 1, N° 2, pp. 275-287.

Sun, H., Weng, C., Lee, C. & Yang, C. (2011). Anti-Forensics with Steganographic Data Embedding in Digital Images, *IEEE Journal on Selected Areas in Communications*, vol. 29, N° 7, pp. 1392-1403.

Telles, G. P., Minghim, R. & Paulovich, F. V. (2007). Normalized Compression Distance for Visual Analysis of Document Collections, *Computer & Graphics*, vol. 31, pp. 327-337.

Tran, N. (2007). The Normalized Compression Distance and Image Distinguishability, *Proceedings of SPIE*, vol. 6492, p. 64921D.

USC-SIPI Image Database, [online] available at http://spici.usc.edu/database.

Vazquez, P. & Marco, J. (2011). Using Normalized Compression Distance for Image Similarity Measurement: An Experimental Study, *The Visual Computer*, vol. 28, N° 11, pp. 1-22.

Vereshchagin, N. K. & Vitanyi, P. M. B. (2010). Rate Distortion and Denoising of Individual Data Using Kolmogorov Complexity, *IEEE Transactions on Information Theory*, vol. 56, N° 7, pp. 3438-3454.

Vereshchagin, N. K. & Vitanyi, P. M. B. (2004). Kolmogorov's Structure Functions and Model Selection, *IEEE Transactions on Information Theory*, vol. 50, N° 12, pp. 3265-3290.

Vereshchagin, N. K. & Vitanyi, P. M. B. (2006), Algorithmic Rate-Distortion Function, *IEEE International Symposium on Information Theory*, Washington, USA, pp. 798-802.

Vereshchagin, N. K. & Vitanyi, P. M. B. (2002). Kolmogorov's Structure Functions with an Application to the Foundations of Model Selection, *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, Vancouver, Canada, pp. 751-760.

Vitanyi, P. (2006). Meaningful Information, *IEEE Transactions on Information Theory*, vol. 52, N°. 10, pp. 4617-4626.

Wang, Z. & Li, Q. (2011) Information Content Weighting for Perceptual Image Quality Assessment, *IEEE Transactions on Image Processing*, vol. 20, N° 5, pp. 1185-1198.

Wang, Z. & Simoncelli, E. P. (2005). Translation Insensitive Image Similarity in complex Wavelet Domain, *IEEE Inter. Conf. Acoustic, Speech & Signal Processing*, vol. 2, Pennsylvania, USA, pp. 573-576.

Wang, Z., Wu, G., Sgeikh, H. R., Simoncelli, E. P., Yang, E. H. & Bovik, A. C. (2006). Quality Aware Images, *IEEE Transactions on Image Processing*, vol. 15, N° 6, pp. 1680-1689.

Wang, Z. & Bovik, A. C. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing*, vol. 13, N° 4, pp. 600-612.

Wang, Z., Simoncelli, E. P. & Bovik, A. C. (2003), Multi-Scale Structural Similarity for Image Quality Assessment, *IEEE Conference on Signals, Systems and Computers*, pp. 1398-1402.

Wang, Z. & Bovik, A. C. (2002). A Universal Image Quality Index, *IEEE Signal Processing Letters*, vol. 9, N° 3, pp. 81-84.

Watanabe, T., Sugawara, K. & Sugihara, H. (2002). A New Pattern Representation Scheme Using Data Compression, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, N°5, pp. 579 – 590.

Weinberger, M. J., Seroussi, G. & Sapiro, G. (2000). The LOCO-I LossLess Image Compression Algorithm, Principles and Standardization into JPEG-LS, *IEEE Transactions on Image Processing*, vol. 9, N°8, pp. 1309- 1324.

Weinberger, M. J., Rissamen, J. J. & Arps R. B. (1996). Applications of Universal Context Modeling to LossLess Compression of Gray-Scale Images, *IEEE Transactions on Image Processing*, vol. 5, N° 4, pp. 575- 586.

Wilkinson, G. (2005). Result and Implications of a Study of Fifteen Years of Satellite Image Classification Experiments, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, N° 3, pp. 433-440.

Xiang, Y. Li, K. & Zhou, W. (2011). Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics, *IEEE Transactions on Information Forensics and Security*, vol. 6, N° 2, pp. 426-437.

Yongliang, L., Yang, X., Yao, H., Huang, T. & Gao, W. (2005). Watermark Detection Schemes with High Security, *IEEE International Conference on Information Technology: Coding and Computing - ITCC*, Las Vegas, USA, pp. 113-117.

Zeng, W. & Liu, B. (1999). A Statistical Watermark Detection Technique Without Using Original Images for Resolving Rightful Ownerships of Digital Images, *IEEE Transactions on Image Processing*, vol. 8, N° 11, pp. 1534-1548.

Zhang, L., Zhang L., Mou, X. & Zhang, D. (2011). FSIM : A Feature Similarity Index for Image Quality Assessments, *IEEE Transactions on Image Processing*, vol. 20, N° 8, pp. 2378-2386.

Ziv, J., & Lempel, A. (1978), Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory*, vol. 24, N°5, 530-536.

Zhang, Y. & Adjeroh, D. (2008). Prediction by Partial Approximation Matching for Lossless Image Compression, *IEEE Transactions on Image Processing*, vol. 17, N° 6, pp. 924-935.