



UNIVERSITA' DEGLI STUDI DI CATANIA

DOTTORATO DI RICERCA IN MATEMATICA APPLICATA – XXIV ciclo

Tesi di Dottorato
**Applied Image Processing Algorithms
for Forensic Applications**

Arcangelo Ranieri BRUNA

Tutor

Ch.mo Prof. Sebastiano Battiato

Coordinatore

Ch.mo Prof. Giovanni Russo

ANNO ACCADEMICO 2011-2012

To my wife Gabriella, Mattia and Gabriel.

Acknowledgements

I would like to take this opportunity to thank the whole Image Processing Lab (Catania University), in particular prof. Battiato for his patience and to trust on me.

A special thank is also dedicated to Fausto Galvan and Giovanni Puglisi for the excellent job: thanks to their technical background they allowed the research to proceed faster and in the right direction.

I cannot forget also to thank my family whose patience allowed me to dedicate some of the time I should have spent with them for this PhD.

Contents

Preface	11
Essential bibliography	15
PART I IMAGE FORGERY DETECTION USING COMPRESSION ARTIFACTS ANALYSIS	17
1. Abstract.....	18
2. The driving application.....	22
3. The DCT – Discrete Cosine Transform	23
4. The JPEG file format.....	26
5. DCT artifacts.....	31
6. Cropping (copy and paste) detection	35
6.1. Li et al approach	36
6.2. Proposed system	41
6.3. Experimental results.....	47
7. Double quantization detection	51
7.1. State of the art.....	52
7.2. Scientific background	54
7.3. Proposed solution (formula based).....	57
7.4. Proposed solution (analysis of the histogram of DCT bins)	64
8. Conclusions and future works	86
PART II Banknotes Counterfeit Detection.....	87

9. Abstract.....	88
10. Introduction.....	89
11. State of the art	91
12. Validation and classification.....	95
12.1. Calibration.....	95
12.2. Training	96
12.3. Banknote authentication	98
13. HW prototype.....	101
14. Experimental results.....	104
15. Conclusions and future work	106
References	107

Preface

The research activities described in this thesis have been mainly focused on the analysis and development of techniques able to detect image forgeries. Image Forgery is not only the process to modify an image, but it has a more subjective meaning. A definition could be the following: image forgery is the process of making, adapting, or imitating images with the intent to deceive. It means that an image can become a forgery based upon the context in which it is used. Image manipulation to enhance the visual appearance or to obtain funny images usually cannot be considered forgery, while similar images used to create ambiguous scenario or to discredit someone is a forgery.

The thesis is organized in two main parts: Forensics Image Processing mainly focused on detecting tampering by using compression artifacts analysis and banknote counterfeit detection.

Tampering detection through compression artifacts analysis

Image forgery is not new, but was already performed also in the past with ‘analog’ photography, but it was not simple to obtain realistic images with such techniques, so only few experts were able to create forgeries. Conversely, the digital imaging and the availability of several image processing softwares allow also to inexperienced people to obtain good results. There are several kinds of digital forgeries [1]:

- Images created with computer graphics tools (i.e. graphic rendering): these are generated through tools and they are entirely virtuals. Although these tools are very powerful, it is not simple to obtain a ‘realistic’ image, hence only experts are usually able to create deceiving images (i.e. forgeries).

- Images obtained by altering the content of a real image: in this case the modifications are obtained by altering some parameters available in any image processing software, e.g., contrast, colors, brightness, etc. The modification can be global or local, just to enhance some particular.
- Images obtained by adding, removing or moving objects. The easiest way is to copy one object from an image and insert it into another image. By manipulating the content, the meaning of the image can drastically change.

There are two methodologies to detect tampered images: active protection methods and passive detection methods [2]. The active protection methods make use of a signature inserted in the image [3]. If the signature is no more detectable, the image has been tampered. These techniques are used basically to assess ownership for artworks and/or related copyrights. Passive detection methods make use of ad-hoc image analysis procedures to detect forgeries. Usually the presence of peculiar artifacts is properly investigated and, in case of anomalies, the image is supposed to be counterfeit.

There is no complete tool helping investigators to detect forgeries, since several techniques counterfeit methods are possible. So the detectives must rely on their experience and apply (usually) semi-automatic methods to retrieve if an image has been tampered.

The first part of the thesis has been devoted to detecting tampering by using compression artifacts analysis. JPEG and MPEG, i.e. the two most used still and video compression standards make use of Discrete Cosine Transform (DCT) engines followed by quantization. It allows obtaining high compression ratio, but, conversely, produces a peculiar artifact. The investigation focused on the characterization of such artifacts and on exploiting such

anomalies to detect counterfeits. The most important artifact is the *blocking artifact*, i.e. a sort of grid superimposed to the original image. This effect is visually visible at higher compression rate, but it can be measured also with low compression ratio. The grid is aligned with the image and has usually the size of 8x8 pixels (in both the directions). Driven by an application aiming to retrieve the original camera model used to shoot the image, two research topics have been targeted: the cropping detection, i.e. the operation that allows obtaining a subpart of the image, and the double quantization detection, i.e. the fact that an image, previously compressed, is opened, modified and hence re-compressed. Moreover techniques aiming to retrieve the first quantization coefficients were also been analyzed.

Image cropping is the technique used to remove unwanted portion of the image. The main effect of the cropping is a misalignment of the regular grid (blocking artifact). Starting from the state of the art algorithms, a new technique has been developed in order to detect if the regular grid is not aligned to the block size (8x8 pixels). This algorithm is very reliable at higher compression ratio while the reliability decreases according to the bit rate, since the blocking artifact is also reduced.

Banknotes counterfeit detection

The detection of counterfeit banknotes is one of the most important tasks in billing machines. Usually it is performed by employing several image analysis techniques (transmittance analysis, reflectance analysis, etc.) on different light spectra (visible, infrared and ultraviolet). Unfortunately these systems are very expensive and are used only for ATM machines where a high degree of reliability is required since those machines are fully

unsupervised by humans. In the last years, cheaper systems for validation and classification of banknotes have been commercialized. They are usually based on motors aimed to let the banknote pass through light emitters and sensors in a dark area.

The second part of the thesis is devoted to describe a new system (both hardware and software) for banknote counterfeit detection. The target was to develop a very cheap system without any moving parts (e.g. motors). Users should simply lean the banknote to be analyzed in a flat glass and the system detects forgery, as well as recognizes the banknote value. We used an infrared camera to acquire the banknote enlighten by six infrared Light Emitter Diodes (LED). The infrared light has been used because there are two different inks used in the banknotes with different behavior: one is reflective and one is absorptive. It allows obtaining peculiar images difficult to be counterfeit. In order to obtain a system with high reliability, a lot of problems were handled. First of all the IR light of the LED produces a non-uniform light distribution. It required performing a calibration and a generation of a correction map to be used before image analysis. Moreover the dust in the glass and the high level of noise due to the low light conditions produce very noisy image. Also the banknote position in the glass is a concern, since it produces roto-translation of the acquired image. The last but not the least was the ambient light: since there is no cover in the proscenium, the ambient light (direct light and reflections) cause several problems to the acquired image and, hence, to the image recognition system.

Overall we had also to face with performance constraints: reliability and response time were very tough. It was required to obtain 100% of counterfeit detection, while an amount of false positive were accepted. All the validation and recognition phases should provide results in 2-3 seconds.

Unfortunately there were few papers in literature related to banknotes and very few about euro-banknotes. A technique has been conceived using a template matching approach: a training phase allows retrieving a template for each banknote (seven templates for all the banknotes). In order to reduce the effect of the light of the LEDs, a calibration phase allows, for each sample, to detect the light distribution map. During the acquisition process, a pre-processing phase allows compensating for the illuminant and to remove Gaussian noise. The validation phase allows to detect the validity of the banknote and the denomination phase allows obtaining the value of the valid banknote. A hardware prototype has also been developed for real time usage.

Essential bibliography

A. R. Bruna, G. Messina, S. Battiato, "Crop Detection Through Blocking Artefacts Analysis" - International Conference on Image Analysis and Processing (ICIAP) 2011

S. Battiato, A. R. Bruna, G. Farinella, C. Guarnera, "Counterfeit Detection and Value Recognition of Euro Banknotes" accepted in 8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2013)

S.Battiato, A.R.Bruna, F. Galvan, G. Puglisi, "A new method to extract the first quantization coefficient in double compressed images" submitted to IEEE International Conference on Multimedia and Expo (ICME 2013), 15-19 July, 2013

S.Battiato, A.R.Bruna, F. Galvan, G. Puglisi, "Double compressed images detection through histogram analysis", to be submitted to IEEE International Conference on Image Processing (ICIP 2013), 15-18 Sep, 2013

S.Battiato, A.R.Bruna, G.Messina, G.Puglisi, "Image Processing for Embedded Devices" - Applied Digital Imaging ebook series (book), eISBN: 978-1-60805-170-0, 2010

S. Battiato, A. R. Bruna, G. Farinella, C. Guarnera, "Counterfeit Detection and Value Recognition of Euro Banknotes" to be submitted to MDPI Sensors — Open Access Journal

PART I

IMAGE FORGERY DETECTION USING COMPRESSION ARTIFACTS ANALYSIS

1. Abstract

Image Forgery is the process of making, adapting, or imitating images with the intent to deceive. It means that an image can become a forgery based upon the context in which it is used. As example, funny modifications to an image in general cannot be considered a forgery until it is not used, e.g., to extort money. There are several examples in internet. One of the most interesting is <http://www.fourandsix.com/photo-tampering-history/>.

As example, in Figure 1, the intent of the image modification is to hide body imperfections [1]. It cannot be considered a forgery. Conversely, the image in Figure 2 is probably a forgery. Figure 3 is also a forgery since the image was used by a Swiss tabloid to deceive the and to fascinating the readers. Figure 4 instead is not a forgery since in this case the aim is not to deceive someone but to show the customers the benefits of a surgical treatment.

In the last years, the number of forged images has drastically increased due to the spread of image capture devices, especially mobile phones, and the availability of image processing software. Several techniques may be used to tamper an image: copy and paste, filtering, collage of multiple image, etc. are just few examples.

In image forensics it is important to understand (without any doubts) if an image has been modified after the acquisition process.

There are two methodologies to detect tampered images: active protection methods and passive detection methods [2]. The active protection methods make use of a signature inserted in the image [3]. If the signature is no more detectable, the image has been tampered. These techniques are used basically to assess ownership for artworks and/or related copyrights. Passive detection methods make use of ad-hoc image analysis

procedures to detect forgeries. Usually the presence of peculiar artifacts is properly investigated and, in case of anomalies, the image is supposed to be counterfeit.

There is no complete tool helping investigators to detect forgeries, since several techniques counterfeit methods are possible. So the detectives must rely on their experience and apply (usually) semi-automatic methods to retrieve if an image has been tampered.

In this thesis are shown methods able to detect tempering by analyzing artifacts due to the compression algorithms. In particular, since the JPEG and MPEG compression schemes are the most used in the world, we focalized on analyzing the effect of the block based DCT transform and quantization of the coefficients. The methodologies are basically stand-alone systems and may be integrated in more complex pipeline, as shown in Figure 5.



Figure 1: Funny image that cannot be considered as forgery.



Figure 2: This image manipulation, probably, should be considered as forgery.



Figure 3: After 58 tourists were killed in a terrorist attack at the temple of Hatshepsut in Luxor Egypt in Nov 1997, the Swiss tabloid Blick digitally altered a puddle of water to appear as blood flowing from the temple.

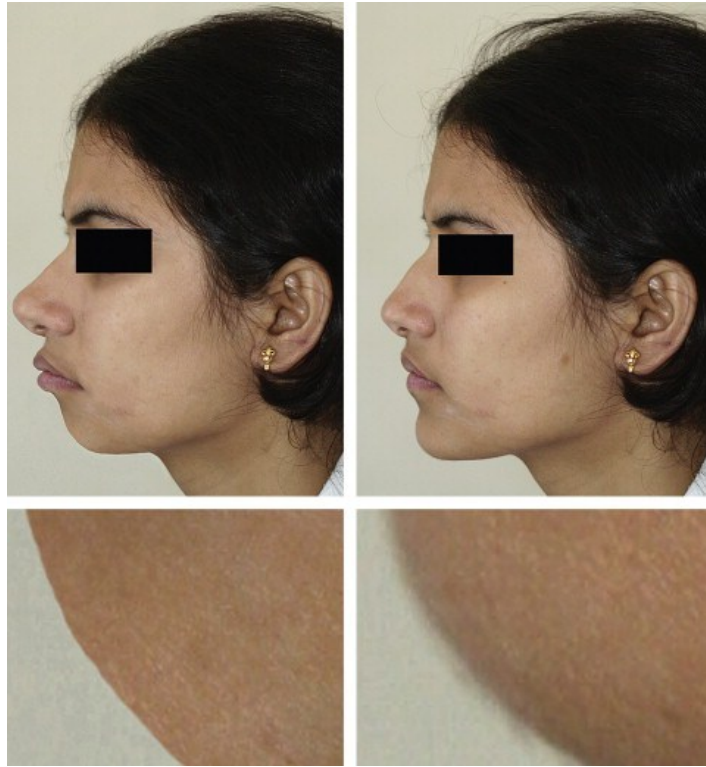


Figure 4: In this case the image was modified digitally to show the changes that can be achieved with surgical-orthodontic treatment.



Figure 5: Block based schema of the pipeline used to retrieve the camera model from an image.

2. The driving application

Although the research has been driven to obtain stand-alone systems to detect forgeries in order to be used to detect the single tampering, also a driven application has been chosen by cascading different blocks in order to obtain a complete solution. As example, in Figure 5 there is a block based schema of a system able to retrieve the camera model from an image, after a copy/paste and re-encoding counterfeit process. The proposed solution aimed to handle images without any further compression. In this case, in fact, the further compression may introduce blocking artifact that may deceive the algorithm. In this case, the algorithms described in [4] and [5] can be used for the quantization detection, while the study proposed in [6], [7] can be used for the signature detection block. In particular the image cropping and the re-quantization will be described in this thesis. They could be inserted in a pipeline for complex tampering detection using compression artifacts analysis.

The research was oriented, in particular, to detect forgeries using the file format commonly used by internet and DSC users. Since JPEG is the most used algorithm for still images and MPEG for video sequences [8], we focused on these algorithms and exploit their peculiarities to detect forgeries. Both JPEG and MPEG use Discrete Cosine Transform (DCT) to obtain a high compression ratio preserving visual quality. This is the reason why, in the next chapters, the DCT, the JPEG algorithm and the DCT artifacts are deeply described.

3. The DCT – Discrete Cosine Transform

The Discrete Cosine Transform (DCT) [9], [10] and [11] is one of the most used image transformation. It is used to de-correlate the image data in the spatial frequencies. It is similar to the Discrete Fourier Transform (DFT), but has several advantages. It is real (while the DFT is complex), it is simpler to be computed, etc.

There are eight standard DCT variants, of which four are common. The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT"; its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". These transformations (DCT and IDCT) are used in some images and video standard compression engines, e.g.: JPEG, MJPEG, MPEG, DV. Moreover there is the modified discrete cosine transform (MDCT), based on the DCT type IV, which is based on a DCT of overlapping data and it is used mostly in audio compression standards, e.g. AAC, Vorbis, WMA, and MPEG-1 audio Layer 3 (known as MP3).

We focused on the DCT and IDCT (i.e. type-II and type-III), since they are used in image and video compression standards.

Formally, the discrete cosine transform is a linear, invertible function $f: R^N \rightarrow R^N$ (where R denotes the set of real numbers), or equivalently an invertible $N \times N$ square matrix. The N real numbers x_0, \dots, x_{N-1} are transformed into the N real numbers y_0, \dots, y_{N-1} according to the formula:

$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad k = 0, \dots, N-1. \quad \text{Eq. 1}$$

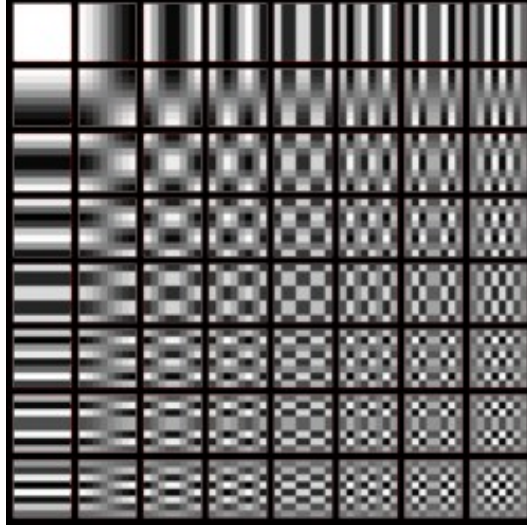


Figure 6: Two dimensional DCT spatial frequencies.

Since an image is a 2D matrix, the 2D transform is obtained as a separable product (equivalently, a composition) of DCTs along each dimension, i.e.:

$$\begin{aligned}
 X_{k_1, k_2} &= \sum_{n_1=0}^{N_1-1} \left(\sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right] \right) \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \\
 &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right]. \quad \text{Eq. 2}
 \end{aligned}$$

The kernel is separable, i.e. it is obtained as two application of the 1D DCT twice: first by row and then by column.

It allows reducing the implementation complexity.

The image in Figure 6 shows combination of horizontal and vertical spatial frequencies for an (8x8) two-dimensional DCT. Each step from left to right and top to bottom is an increase in frequency by 1/2 cycle. For example, moving right one from the top-left square yields a half-cycle increase in the horizontal frequency. Another move to the right yields two half-

cycles. A move down yields two half-cycles horizontally and a half-cycle vertically. The source data (8x8) is transformed to a linear combination of these 64 frequency squares.

4. The JPEG file format

JPEG is acronym of Joint Photographic Expert Group [12]. It is the most used still image compression standard algorithm for both low quality images (i.e. the images used in internet) and high quality images (i.e. the images acquired by the digital still cameras) [8], and resisted also to 'attacks' of other standards, e.g., the JPEG2000 [13].

The Figure 7 shows the block based schema of the algorithm.

The image is supposed to be described in *YCrCb* format (i.e. luminance and chrominance). The chrominance components may also be subsampled, since the visual impact of the lossy information is negligible.

Every component (both luminance and chrominance) is divided into 8x8 pixels blocks. Each block is transformed in the DCT domain according to a modified formula described in the previous Section 3:

$$G_{u,v} = \sum_{x=0}^7 \sum_{y=0}^7 \alpha(u) \alpha(v) g_{x,y} \cos \left[\frac{\pi}{8} \left(x + \frac{1}{2} \right) u \right] \cos \left[\frac{\pi}{8} \left(y + \frac{1}{2} \right) v \right] \quad \text{Eq. 3}$$

where

- u is the horizontal spatial frequency, for the integers $0 \leq u < 8$.
- v is the vertical spatial frequency, for the integers $0 \leq v < 8$.

- $\alpha(u) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } u = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$ is a normalizing scale factor to make the transformation orthonormal

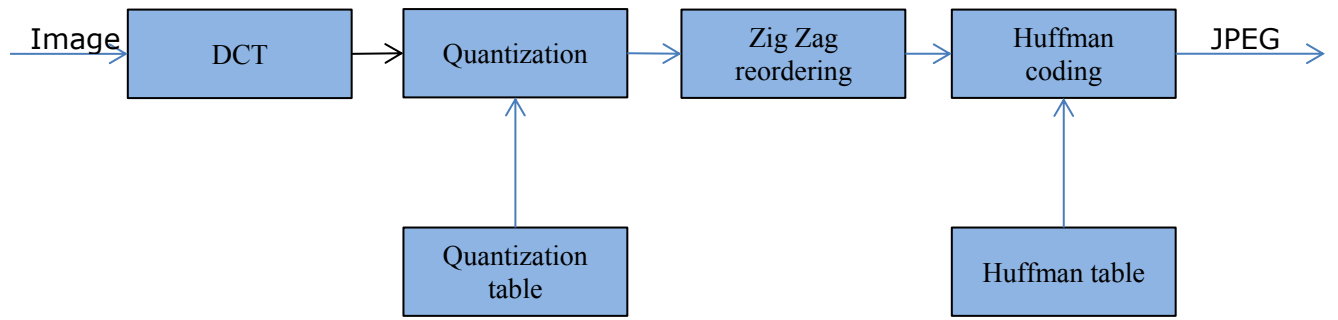


Figure 7: Block based schema of the JPEG standard.

- $g_{x,y}$ is the pixel value at coordinates (x, y)
- $G_{u,v}$ is the DCT coefficient at coordinates (u, v) .

Compared to the DCT in Section 3, this formula makes the DCT-II matrix orthogonal (i.e. the transpose matrix is the inverse matrix), but breaks the direct correspondence with a real-even DFT.

It is used to de-correlate the 64 coefficients in the spatial frequencies, so each one of the coefficient can be elaborated independently using a visually weighted quantization matrix. The quantization coefficients are the main cause of the loss of information. In Figure 8 is shown the quantization matrix for the luminance and chrominances proposed in the Annex K of the JPEG standard [22]:

$$Q_y = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$Q_c = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

Figure 8: Standard proposed quantization matrix used for the luminance (Y) and the chrominance (C).

$$Q_y = \begin{bmatrix} 7 & 7 & 33 & 27 & 46 & 48 & 49 & 50 \\ 55 & 60 & 39 & 44 & 46 & 48 & 48 & 38 \\ 33 & 40 & 43 & 13 & 47 & 48 & 50 & 59 \\ 35 & 28 & 46 & 47 & 48 & 48 & 48 & 52 \\ 47 & 41 & 48 & 48 & 32 & 51 & 53 & 64 \\ 49 & 49 & 49 & 48 & 50 & 55 & 54 & 61 \\ 50 & 81 & 49 & 36 & 58 & 45 & 69 & 29 \\ 53 & 50 & 56 & 32 & 64 & 69 & 78 & 44 \end{bmatrix}$$

Figure 9: Example of quantization table optimized for landscapes.

The quantization step is performed according to the following formula:

$$out_{i,j} = round\left(\frac{in_{i,j}}{q_{i,j}}\right), \quad i, j = 0, \dots, 7 \quad \text{Eq. 4}$$

The standard does not impose to use the quantization table shown above, so the quantization values can be chosen depending on the desired compression ratio [14], [15],

[16]. Optimization of the coefficients may be performed depending on several aspects: image acquisition sensor, target bit rate, image category (e.g. portrait and landscape), etc [17]. In Figure 9 an example of quantization table optimized for landscapes is shown.

It is important to remark that every image acquisition system uses, in general, its own quantization tables. This information can be used to retrieve the camera model even if it is not directly shown in metadata [6], [7].

The coefficients of the quantized blocks are then reordered according to a zig-zag ordering method shown in Figure 10. It allows obtaining a 1D vector of coefficients ordered according to the spatial frequency, in particular from the lowest frequency (called DC) to the highest frequencies (called AC).

The particular scanning allows obtaining, statistically, nonzero values at the beginning of the vector and a long run of zero coefficients. Finally the chain is encoded using a run-length/variable length encoder through a Huffman table. The Huffman table is not standardized, but can also be optimized, i.e., according to the image content [18]. This process allows modifying the bit rate, but it is a time consuming methodology and the benefit in terms of compression ratio improvement is not enough, so it is not usually used by commercial systems in real time.

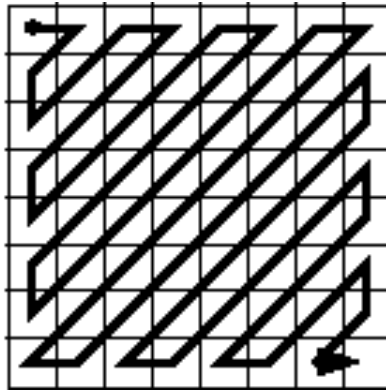


Figure 10: Zig zag scanning order.

5. DCT artifacts

The DCT, as stated above, is the widely used image transformed domain for image and video compression since it provides better tradeoffs between image quality and compression ratio. Increasing the compression ratio, the quality, of course, decreases. Several artifacts are generated at lower quality, since they are the result of discarding a relevant amount of information represented by DCT coefficients once they have undergone the lossy quantization process. Usually these are side effects. But, since they are well known and widely studied, they can also be used to detect if a compression schema has been applied to an image.

The most known artifacts are [44], [45] and [46]:

- Blocking
- Ringing
- Color bleeding

Blocking artifact is the most known artifact. It appears as small square blocks all over the images. It is due to the fact that JPEG and MPEG compression engines work with 8x8 blocks, one at a time. An example is shown in Figure 11.



Figure 11: Example of blocking artifact.

Ringling artifact is a distortion of edge appearance. DCT based compression schemes, in fact, operate in spectral domain, trying to represent the image as a sum of smooth oscillating waves. Spectral domain is appropriate for capturing relatively smooth color gradients, but not particularly appropriate for capturing edges. So, instead of capturing an edge as it is, the edge is fitted with wave functions. Further quantization step produces loss of detail and, especially in strong edges, produces this artifact. An example is shown in Figure 12.

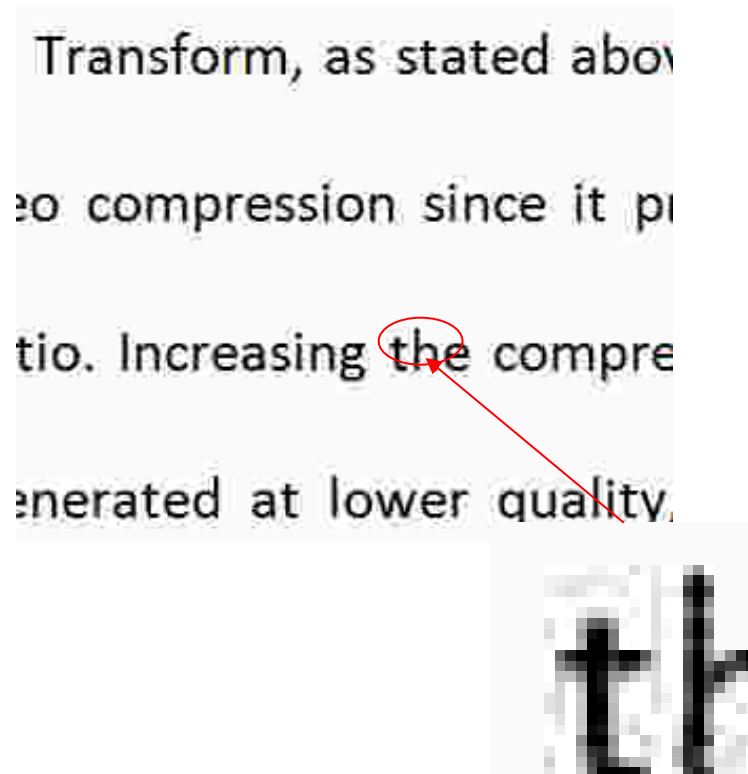


Figure 12: Example of ringing artifact.

Colour bleeding artifact appears as a smearing of the colour between adjacent areas of strongly contrasting chrominance. This phenomenon can be observed in both still images and videos. The phenomenon is due to the combination of quantization and decimation (mainly 4:2:0 sub-sampling), which introduces spurious information on the reconstructed signal; hence no bleeding is usually observed when the two signal degrading processes are not simultaneously applied. An example of such artifact is visible in the Figure 13.



Figure 13: Example of color bleeding artifact.

Usually these artifacts are unwanted effects and a lot of algorithms have been developed to reduce them. In image forensic they can be used to detect if tampering methods have been used.

6. Cropping (copy and paste) detection

Copy and paste is the simplest technique used to counterfeit images. It can be used to obtain a completely new image by cropping the interest part or to cover unwanted details.

Several algorithms exist in literature as reported in a recent survey [4]. Among others, a lot of methods in the field consider the possibility to exploit the statistical distribution of DCT coefficients in order to reveal the irregularities due to the presence of a superimposed signal over the original one [4], [5], [20], [21]. The DCT artifacts analysis have a further advantage due to the fact that it is the most used compression technique; JPEG [22] for still images and MPEG compression family [23] for video sequences make use of block based DCT data compression (usually 8x8 pixel size non-overlapping windows).

While there are a lot of algorithms in literature for the estimation of the quantization (and compression) history [4], [5], there are only a few approaches for cropping detection [21]. In this Section a pipeline is suggested aiming to merge both techniques in order to obtain a more reliable system. In fact, blocking artifacts analysis works well when the image is aligned to the block boundary. But, in case the image has been cropped, this assumption is no more valid and the detection may fail. In the Figure 5 the quantization detection block is preceded by a cropping detection in order to align the image to the block boundary. It allows to increase the reliability of the detection and to better understand if only some parts of the image are tampered.

In Li et al. [2] the grid extraction is realized by extrapolating the Block Artefact Grid (BAG) embedded in the image where block artefact appears, i.e. the grid corresponding to the blocks boundaries. The authors introduce a measure of blockiness just considering the ratio between the sum of AC components of the first row (and column) with respect to the DC

coefficient in the DCT domain. One of the main drawbacks of the method is related to the fact that it requires to compute 8x8 pixels DCT again (in a dense way) over the image under detection, just to locate properly the correct alignment. Also the detection of tampered regions corresponding to misaligned grid is demanded to a visual inspection of the resulting filtering operation without a clear and objective measure.

The same authors propose in [21] an interesting approach based on spatial consideration devoted to locate the BAG just combining together a series of derivative and non-linear filters to isolate blockiness avoiding the influence of textures or strong edges present in the input image. Unfortunately, both the techniques were not properly evaluated just considering a proper dataset (high variability with respect to resolution size) and a sufficient number of misalignments cropping with respect to the 8x8 grid. Also an exhaustive comparison just considering the overall range of JPEG compression factors is lacking.

As described in [4], [24] the new emerging fields of Digital Forensics require to provide common benchmarks and datasets that are needed for fair comparisons among the numerous proposed techniques and algorithms published in the field. The same authors suggested a third approach in [25] that overcome some problems of the previous and will be described deeply in the next Section.

6.1. Li et al approach

One of the most interesting approaches related to the detection of copy and paste detection of pre-quantized images is the Li et al approach [25]. They define “Block Artifact Grid” (BAG) as the grid embedded in an image where block artifact appears.

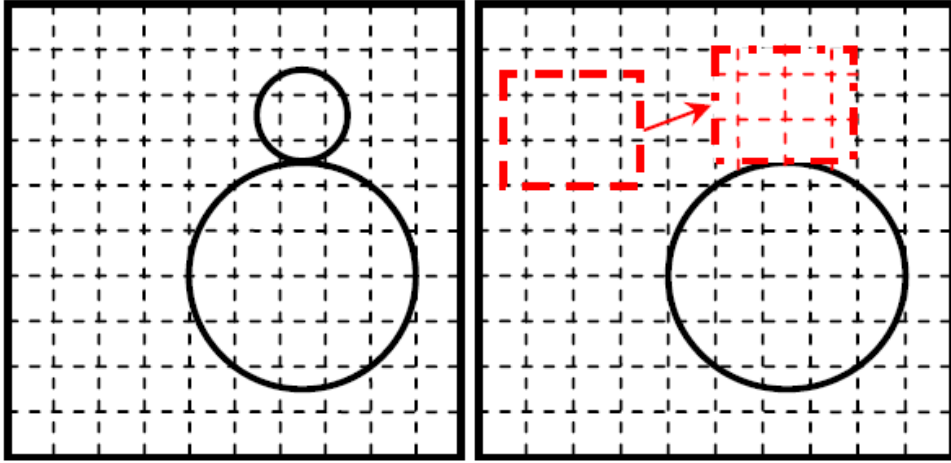


Figure 14: Example of BAG misalignment.

The DCT grid and BAG match together in undoctored images. When an image slice is moved, the BAG within it also moves. In Figure 14 there is an example showing the mismatch of the BAG with the DCT grid when a slice is copied and pasted in another image area.

Li proposed an algorithm to detect the BAG misalignment. It locates the BAG firstly, and then checks whether the BAG mismatches or not with the DCT grid. Once a BAG mismatch is detected, then the image can be considered doctored. It is based on the assumption that the high frequency AC coefficients of every 8x8 DCT block are usually zero after quantization in a pre-compressed image. When no copy and paste operation occurred, high frequencies coefficients of all the blocks are zero, otherwise some values are not zero, since BAG is not aligned. It is basically an anomaly and it is a cue of copy and paste operation. Even when the texture is complex and some high frequency coefficients are not zero, the misaligned BAG may be also detected.

As example a 1D example will be considered. To locate the BAG, an error measure was defined, as shown in the following formula:

$$E_i = |p_{i+7} - s_{i+7}| \quad \text{Eq. 5}$$

Where p_{i+7} is a predictive value while s_{i+7} is the signal and i is the index in the range $[0, \text{length}(\text{signal})]$. Since high frequency DCT coefficients are usually zero, the prediction can be obtained by imposing the 7th coefficient to be zero too, as shown in the following formula:

$$S_7 = \frac{1}{2} \left[\sum_{n=0}^6 \cos \frac{7(2n+1)\pi}{16} + p_{i+7} \cos \frac{7*15\pi}{16} \right] = 0 \quad \text{Eq. 6}$$

Hence the prediction value can be computed as:

$$p_{i+7} = - \frac{\sum_{n=0}^6 \cos \frac{7(2n+1)\pi}{16}}{\cos \frac{7*15\pi}{16}} \quad \text{Eq. 7}$$

Sliding the window along the sequence, the error effect is calculated. An example is shown in Figure 15. It is visible that the minimum of the values appear at the block edges.

The extension to the 2D case is straightforward: assuming the block to be 8x8 (as usual for JPEG standard), suppose the luminance signal s_{ij} and $S_{u,v}$ are the corresponding DCT coefficients:

$$S_{u,v} = \sqrt{\frac{\alpha_u \alpha_v}{8}} \sum_{i=0}^7 \sum_{j=0}^7 s_{ij} \cos \frac{u(2i+1)\pi}{16} \cos \frac{v(2j+1)\pi}{16} \quad \text{Eq. 8}$$

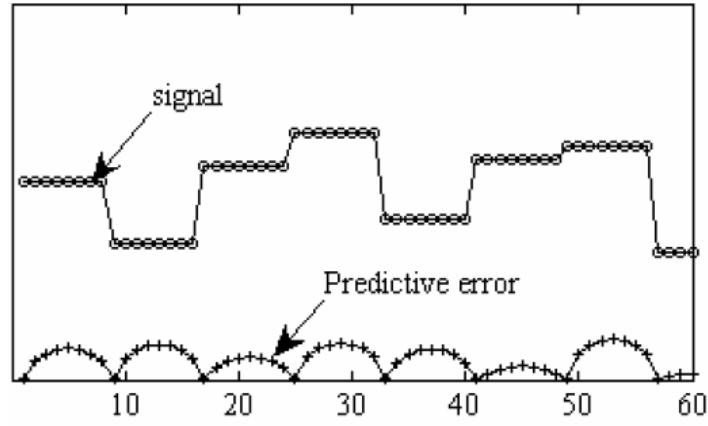


Figure 15: Example of 1D BAG signal and correspondent error function.

Then the local effect is defined by using the last column and row AC coefficients of each block:

$$\begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} & s_{04} & s_{05} & s_{06} & p_{07} \\ s_{10} & & & & & & & \\ s_{20} & & & & & & & \\ s_{30} & & & & & & & \\ s_{40} & s_{41} & & & & & & \\ s_{50} & & & & & & & \\ s_{60} & s_{61} & s_{62} & s_{63} & s_{64} & s_{65} & s_{66} & p_{67} \\ p_{70} & p_{71} & p_{72} & p_{73} & p_{74} & p_{75} & p_{76} & p_{77} \end{bmatrix}$$

Since the 2D DCT is composed by:

$$\begin{aligned} X_{k_1, k_2} &= \sum_{n_1=0}^{N_1-1} \left(\sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right] \right) \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \\ &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right]. \end{aligned} \quad \text{Eq. 9}$$

There are 15 undetermined coefficients. By imposing:

$$P_{u,v} = 0; \quad (u = 7 \text{ or } v = 7) \quad \text{Eq. 10}$$

We obtain 15 equations with 15 undetermined coefficients.

The formula can also be expressed as a matrix equation:

$$[A_{uv,ij}][p_{uv}] = [B_{uv}]; \quad (u \text{ or } v = 7, i \text{ or } j = 7) \quad \text{Eq. 11}$$

Where:

$$A_{uv,ij} = \cos \frac{u(2i+1)\pi}{16} \cos \frac{v(2j+1)\pi}{16} \quad \text{Eq. 12}$$

And

$$B_{u,v} = -\sum_{i=0}^6 \sum_{j=0}^6 s_{ij} \cos \frac{u(2i+1)\pi}{16} \cos \frac{v(2j+1)\pi}{16} \quad \text{Eq. 13}$$

Thus the predictive coefficients can be calculated as:

$$[p_{uv}] = [A_{uv,ij}]^{-1} [B_{uv}] \quad \text{Eq. 14}$$

Finally, the local effect can be defined as the root-mean square prediction error:

$$E_{i,j} = \sqrt{\frac{\sum_{k=0}^7 (p_{i+k,j+7} - s_{i+k,j+7})^2 + \sum_{h=0}^6 (p_{i+7,j+h} - s_{i+7,j+h})^2}{15}} \quad \text{Eq. 15}$$

Sliding the window in the whole image, the map of error E can be obtained as the BAG. The lower values (darkest point if showed as image) correspond to the BAG. In the Figure 16 an example of the process is shown.

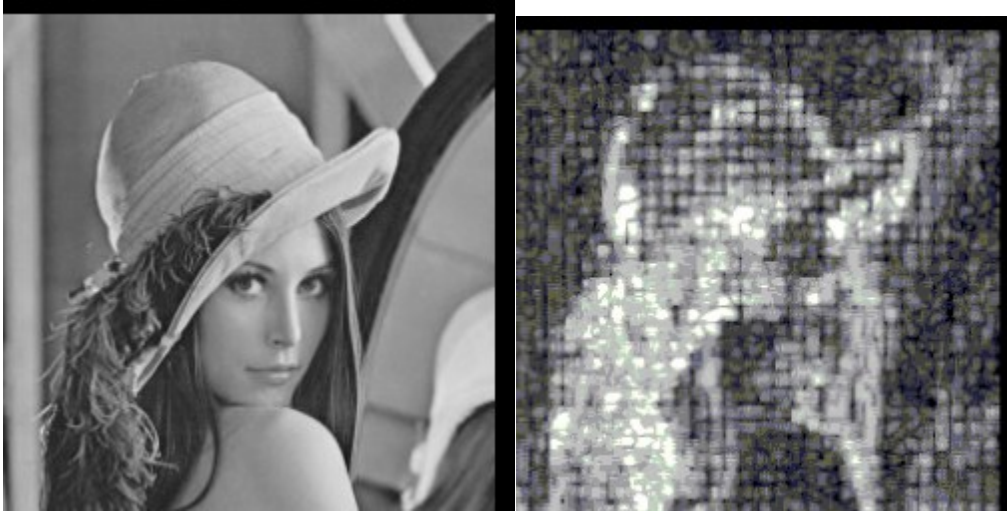


Figure 16: Lena image and the corresponding Block Artifact Grid (BAG) image.

6.2. Proposed system

The DCT codec-engines (e.g., JPEG, MPEG, etc.) typically apply a quantization step in the transform domain just considering non-overlapping blocks of the input data. Such quantization is usually achieved by a quantization table useful to differentiate the levels of quantization adapting its behavior to each DCT basis. The JPEG standard allows using different quantization factors for each of the bi-dimensional DCT coefficients. Usually standard quantization tables are used and a single multiplicative factor is applied to modify the compression ratio [26],[27], obtaining different quality levels. As the tables are included into the image file, they are also customizable as proved by some commercial codec solutions that exploit proprietary tables. Images compressed by DCT codec-engines are affected by annoying blocking artifacts that usually appear like a regular grid superimposed to the signal. In the following, we discuss a simple example based on the Lena image [43]. The picture has been compressed using the *cjpeg* [26] software with a properly managed

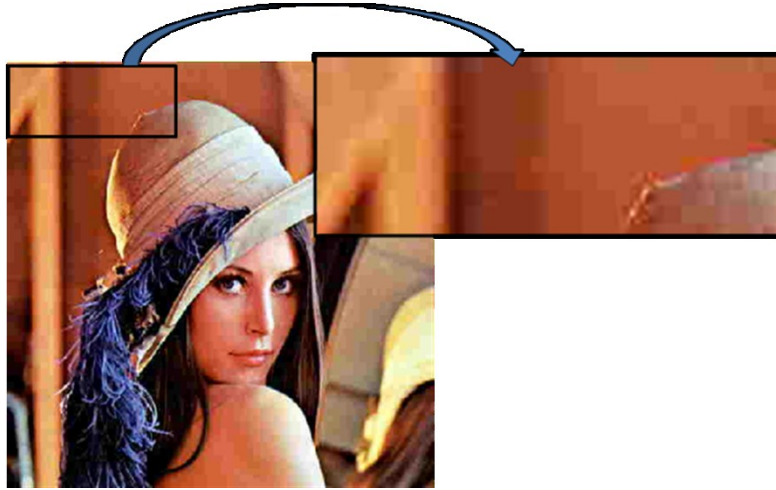


Figure 17: Blocking artefact example in a JPEG compressed image (quality factor = 40).

compression ratio, obtained modifying the quantization tables through the variation of the quality parameter in the range {10, 90} (see Figure 17).

It is an annoying artefact visible especially in flat regions. It is also regular, since it depends on the quantization of the DCT coefficients of every 8x8 blocks.

Unfortunately, this kind of artefact is not simple to be characterized (i.e., detectable) in the Fourier domain, since image content and the effect of the quantization step of the encoding pipeline mask the regular pattern, as shown in Figure 18.

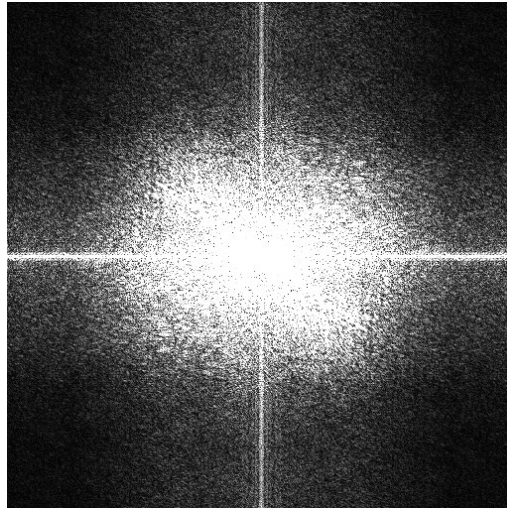


Figure 18: Results of the visualization of the Fourier spectrum applied to the compressed Lena image shown in Figure 17.

We established to work directly in the spatial domain and, in particular, in the luminance component. The blocking artefact is basically a discrepancy in all the borders between adjacent blocks. It is regularly spaced (8x8 for JPEG and MPEG) and it also affects the image in only two perpendicular directions (horizontal and vertical if the image has not been rotated).

The straightforward way to detect such artefact exploits a derivative filter along the horizontal and vertical direction. The proposed strategy could be easily generalized to consider all possible malicious rotation of the cropped image, just iterating the process at different rotation angles. In Figure 19 the overall schema of the proposed algorithm is depicted.

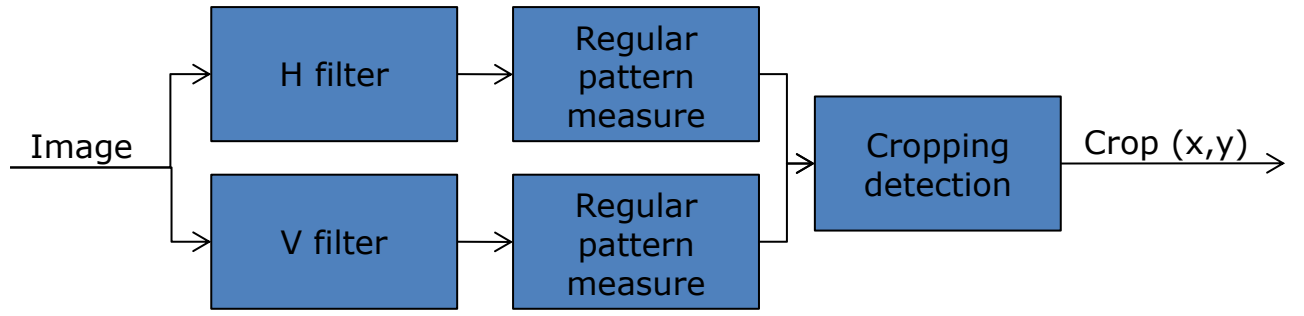


Figure 19: Block based schema of the cropping detection algorithm.



Figure 20: Effect of the Sobel filter (3x3 kernel size) applied to the Lena image.

The blocks “*H filter*” and “*V filter*” are derivative filters. Basically they are High Pass Filters (HPF) usually used to isolate image contours. An example is the Sobel filter, with the following basic kernel:

$$H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}; \quad V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \quad \text{Eq. 16}$$

These filters are able to detect textures, as shown in Figure 20.

It is very useful to retrieve textures of the image, but the blocking artefact effect is also masked. In order to detect only regular pattern and discard real edges, a very long taps directional filter has been used. It was obtained by properly expanding the following 3x3 filters along the horizontal or vertical direction:

$$H = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 0 & 0 & 0 \end{bmatrix}; \quad V = \begin{bmatrix} 1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \end{bmatrix}; \quad \text{Eq. 17}$$

Bigger is the number of taps, better are the results, although computational time also increase. In Figure 21 is shown the result of a long directional filters with different kernel size.

We define the *Regular Pattern Measure* (RPM) that computes a measure of the blockiness effect as defined in the following. Let I a $M \times N$ pixel size image and I^H , I^V the corresponding filtered images obtained by applying a directional HPF as above. For sake of simplicity let suppose to have serialized the image, by simple scan line ordering of the corresponding rows and columns just obtaining the vector I^H , I^V . The RPM values for both directions is obtained as:

$$RPM_H(i) = \sum_{j=0}^{\text{floor}(N/8)} I^{H'}(8 \cdot j + i); \quad i = 1, \dots, 7; \quad \text{Eq. 18}$$

$$RPM_V(i) = \sum_{j=0}^{\text{floor}(M/8)} I^{V'}(8 \cdot j + i); \quad i = 1, \dots, 7; \quad \text{Eq. 19}$$



Figure 21: 30 taps directional HPF applied to the Lena image. Borders are not considered (thus the vertical size in the left image and the horizontal size in the right images are less than the original size).

Experiments have shown that both the RPM_H and RPM_V measures allow discriminating, in a very robust way (e.g., with respect to the main content of the image), the periodicity of the underlying cropping positions. Such values can be extracted by considering a simple order statistic criterion (e.g., the maximum). Figure 22 and Figure 23 show the plot of the two RPM measures, in case of no cropping (e.g., blocking artefact starts with the pixel [1,1]) and in case of malicious cropping at position [6, 5].

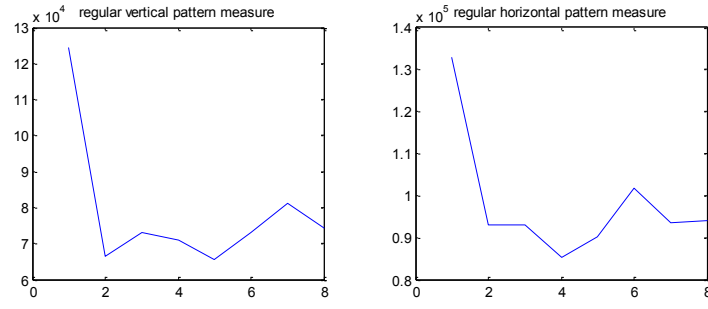


Figure 22: RPM measure without cropping. The blocking artefact starts with the pixel [1,1].

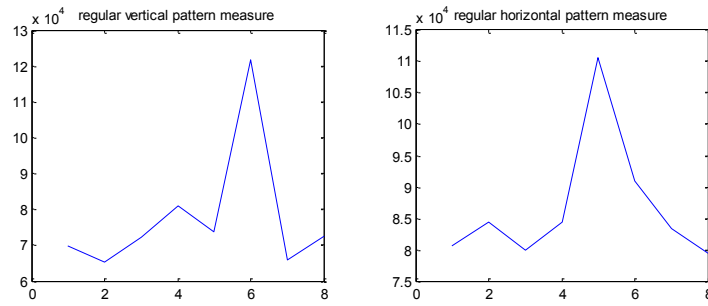


Figure 23: RPM measure with cropping. The blocking artefact starts with the pixel [6,5].

6.3. Experimental results

To assess the effectiveness of any forgery detection technique, a suitable dataset of examples should be used for evaluation. According to [24] the input dataset contains a number of uncompressed images organized with respect to different resolutions, sizes and camera models. Also the standard dataset from Kodak images and from UCID v.2 have been used. The overall dataset can be downloaded from [28]. It is composed by 114 images with different resolution. Experiments were done varying in an exhaustive way the cropping position and the compression rate.

Moreover the *cjpeg* [26], (i.e., the reference code for the JPEG encoder), has been used to compress the images and the flag *-quality* was used to modify the quality from 10 (high compression ratio) to 90 (low compression ratio) with the step of 10. In particular each image has been cropped in order to test every possible cropping position in the 8x8 block, just to consider the possibility to test the method also in presence of real regular patterns in the image that could influence the results. In Table 1 are reported the overall results, described in terms of correct percentage of the cropping position detection, with respect to the involved compression ratio.

Table 1: Results of the proposed method.

Quality factor	Accuracy (%)
10	99
20	91
30	80
40	69
50	58
60	46
70	39
80	28
90	16

Exhaustive tests have been done for every image, every cropping position and quality factor.

Thus the accuracy has been obtained considering 2891 cases.

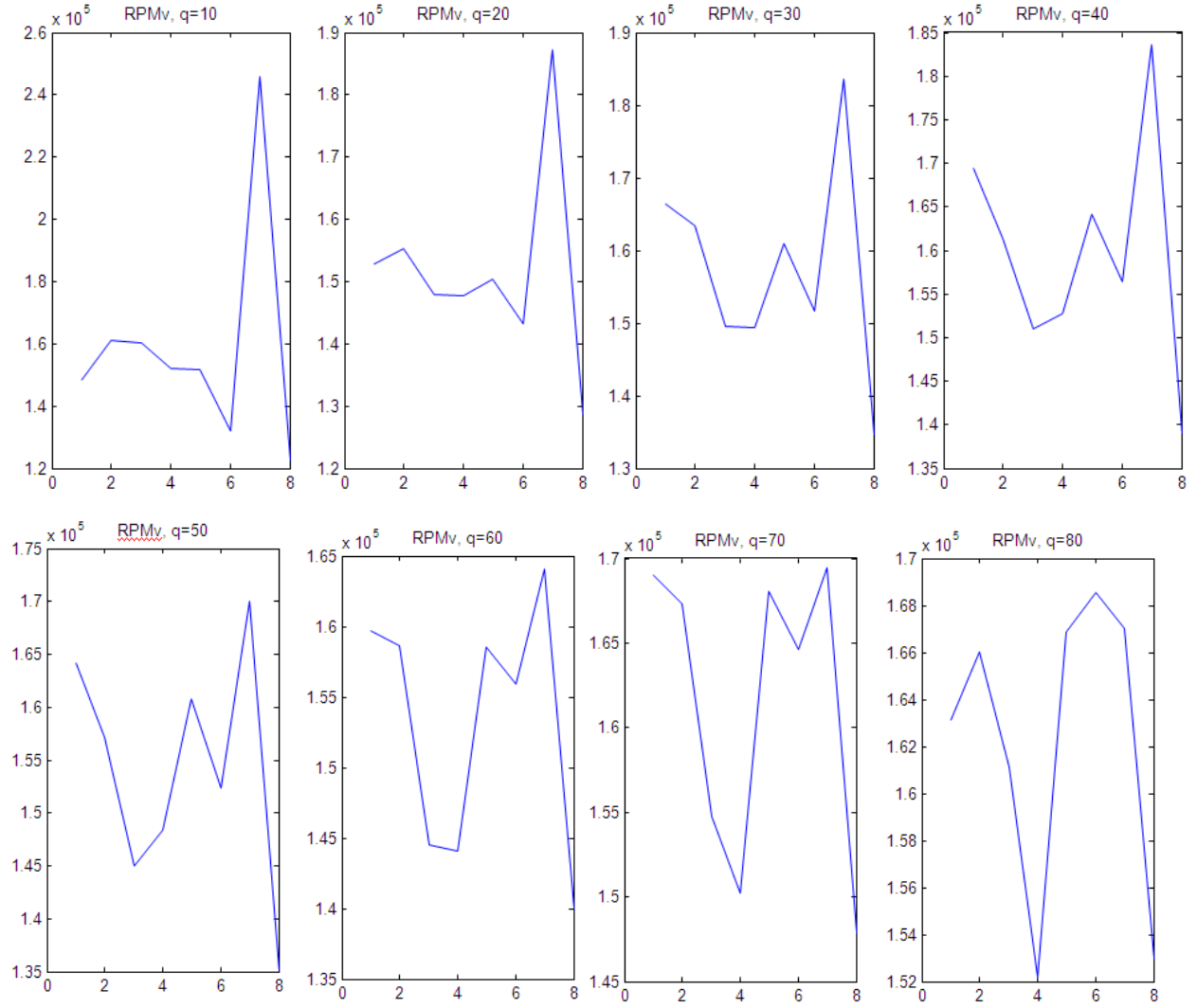


Figure 24: RPM measure obtained at varying the *quality* factor.

Experimental results show that performances increase according to the compression rate. It is reasonable, since the blocking artifact increases at higher compression ratio.

In Figure 24 the RPM (only vertical) measure is shown at varying the *quality* factor (real crop position = 7). Reducing the quantization (i.e., increasing the quality factor), the peak is less evident and, in this example, with the *quality*=80 the estimation fails, since the effect of a real edge becomes predominant.

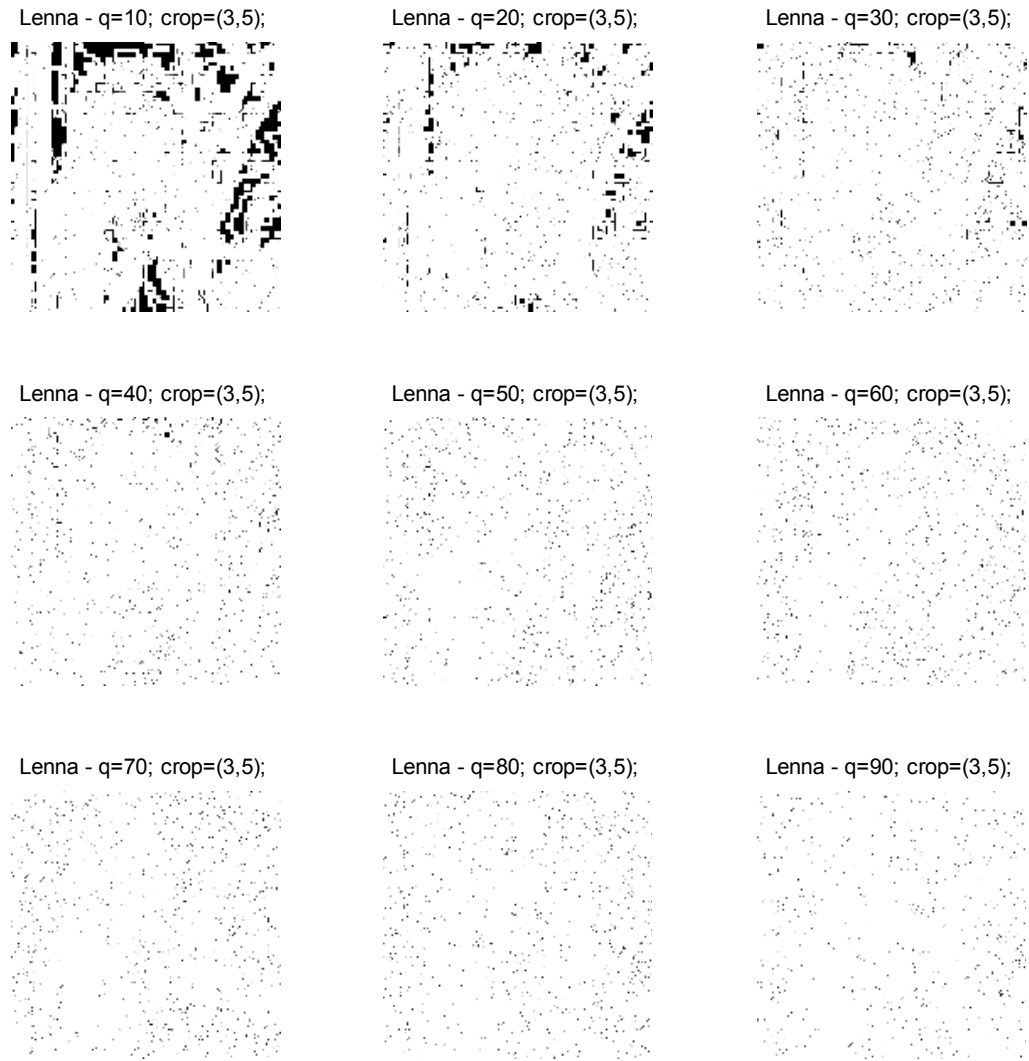


Figure 25: Li's method [1, 7] applied to Lenna image at varying the *quality* factor.

The proposed solution was compared to the method described in [2] [21]. Unfortunately in these papers the cropping detection is not automatic, but it is supposed a visual inspection at the end of the process. In Figure 25 are shown the results of this method at varying the quality factor from 10 to 90 for the Lenna image for a cropping position (3,5). It is evident that the cropping position is detectable up to $q=40$. Above this value it is no more visible. Similar results have been obtained for all the involved dataset.

7. Double quantization detection

In the forensic domain it is important to recover image history, i.e. the camera that acquired the image and any further modification (cropping, filtering, merging with other images, etc.). One of the steps is to detect whether or not an image has been doubly compressed, since every (malicious) modification needs the user to decompress the image, to perform the modification and to save it again.

This is not a clear indication of forgery, but it tells us that the image, probably, is not the one originally saved at the time of the shooting. In literature, algorithms can be found able to detect if double quantization occurs. They are based on the analysis of anomalies in the DCT coefficient distribution. Unfortunately, there are no methods able to retrieve in a robust way the coefficients of the first compression in a double compressed JPEG image. It could be useful, e.g., to detect the camera model used for capturing the image. In the following Sections we report methods able to obtain these coefficients exactly when the second compression is lighter than the first one. In these approaches we do not only use anomalies of the compression engines, but we rely also on the peculiar statistical distribution of the DCT coefficients. It is well known, in fact, that the AC coefficients are distributed according to a Laplacian distribution [20]:

$$p_{i,j}(x) = \frac{\lambda_{i,j}}{2} e^{-\lambda_{i,j}x}, i, j = 0, 1, 2, \dots, 7; \quad \text{Eq. 20}$$

Where:

$$\lambda_{i,j} = \frac{\sqrt{2}}{\sigma_{i,j}}; \quad \text{Eq. 21}$$

$\sigma_{i,j}$ is the (i,j) -DCT standard deviation.

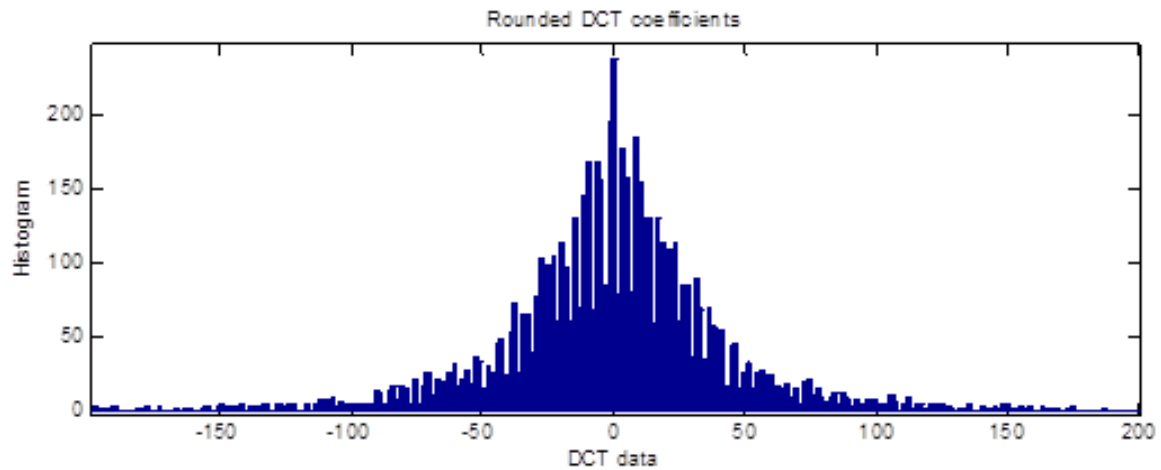


Figure 26: An example of Laplacian distribution.

In Figure 26 there is an example of Laplacian distribution.

7.1. State of the art

The pipeline which leads to ascertain whether an image has undergone to some kind of forgery leads through the following steps: determine whether the image is “original” and, in the case where the previous step has given negative results, try to understand the past history of the image. Discovery that the input image has been manipulated or not is a prelude to any other type of investigation. Regarding the first stage, the EXIF metadata could be examined, but they are not so robust to tampering, so that they can give us indicative but not certain results. To discover image manipulations, many approaches have been proposed in literature (JPEG blocking artifacts analysis [29], hash functions [30], etc.). A lot of works have proved that analyzing the statistical distribution of the values assumed by the DCT coefficient is widely more reliable [20]. In this regard, as reported in [31], [4] and [32], by checking the histogram of these values we are able not only to determinate

whether the image was or not doubly saved, but also if the quantization coefficient, in this case, was greater than or less the one of the first compression. In [5], the authors suggest a method that helps to determine whether an image has been subjected to double compression with the same compression factor. The part of this theory not yet fully developed, at least in our knowledge, concerns the determination of the values of the coefficients of the first quantization. In our approach we focused on the determination of these coefficients. In particular we demonstrate that, when the second quantization factor is lower than the first one, the performed estimation is without doubts, i.e., it also allows detecting the forgery.

From here on, q_1 and q_2 will indicate respectively the coefficient of first and second quantization for a generic frequency in the DCT domain. In [33] the author, to estimate q_1 , proposes to carry out a third quantization and then calculates the error between the coefficients before and after this step. By varying the coefficient of the third quantization he is able to detect two minima, one (absolute) in correspondence of q_2 , and one (local) in correspondence of q_1 . We will discuss in detail the results of this method and its limitations in the remainder of this Section, which is structured as follows: in Section 7.2, after a brief review of the JPEG compression algorithm, we present the mathematical details of our new method, we will explain why we are able to find q_1 when $q_1 > q_2$, and also because this method cannot bring to the same good results in the case where $q_1 \geq q_2$. In Section 7.3 the effectiveness of the proposed solution has been tested considering both synthetic and real data (uncompressed images) reproducing a double quantization in order to simulate the lossy part of the JPEG algorithm.

7.2. Scientific background

The JPEG compression engine [34] works just considering a partition of the input image into 8x8 non-overlapping blocks (for both luminance and chrominance channels). A DCT transform is then applied to each block; next a proper dead-zone quantization is employed just using for each coefficient a corresponding integer value belonging to an 8x8 quantization matrix [20]. The quantized coefficients obtained just rounding the results of the ratio between the original DCT coefficients and the corresponding quantization values are then transformed into a data stream by mean of a classic entropy coding (i.e., run length/variable length). Coding parameters and other metadata are usually inserted into the header of the JPEG file to allow a proper decoding. If an image has been compressed twice (i.e., after having introduced some malicious manipulation) although, of course, the current quantization values are available, the original (initial) quantization factors are lost. It is worth noting that in forensics application, this information could be fundamental to assess the integrity of the input image or to reconstruct some information about the embedded manipulation [35].

For sake of simplicity, considering a single DCT coefficient c and the related quantization factors q_1 (first quantization) and q_2 (second quantization), the value of each coefficient is given by:

$$\mathit{round}\left(\mathit{round}\left(\frac{c}{q_1}\right) \times \frac{q_1}{q_2}\right) \quad \text{Eq. 22}$$

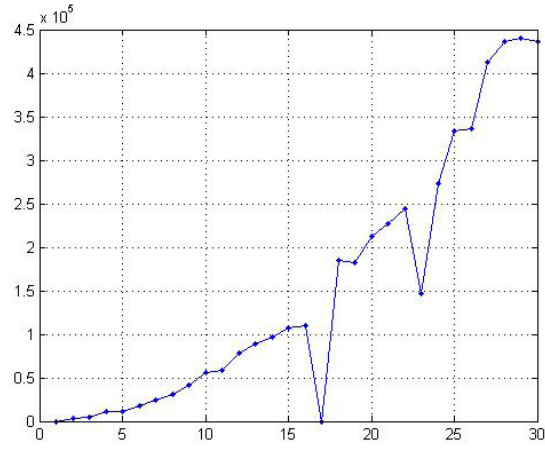
To infer the value of q_1 , previous works suggest to quantize again such value with a quantization coefficient (q_3) varying in a certain range [33] and to evaluate an error function defined as follows:

$$\begin{aligned}
f_e(c, q_1, q_2, q_3) &= \text{abs} \left(\text{round} \left(\text{round} \left(\text{round} \left(\frac{c}{q_1} \right) \times \frac{q_1}{q_2} \right) \times \frac{q_2}{q_3} \right) \times q_3 \right. \\
&\quad \left. - \text{round} \left(\text{round} \left(\frac{c}{q_1} \right) \times \frac{q_1}{q_2} \right) \times q_2 \right)
\end{aligned}$$

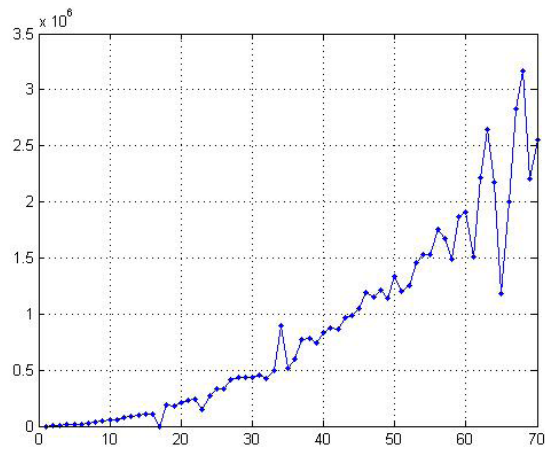
Eq. 23

For examples, taking an image from [36], the outcome, considering the DC term when $q_1=23$ and $q_2=17$, is shown in Figure 27 (a). In this particular case (both coefficients are prime numbers) we obtain interesting results: both q_1 and q_2 can be easily found, since they correspond to the two evident local minima. Hence, in this specific example, the first quantization coefficient q_1 can be retrieved (q_2 , as mentioned before, is already available). Unfortunately, in real cases, the original quantization coefficient cannot be easily inferred as reported in the following:

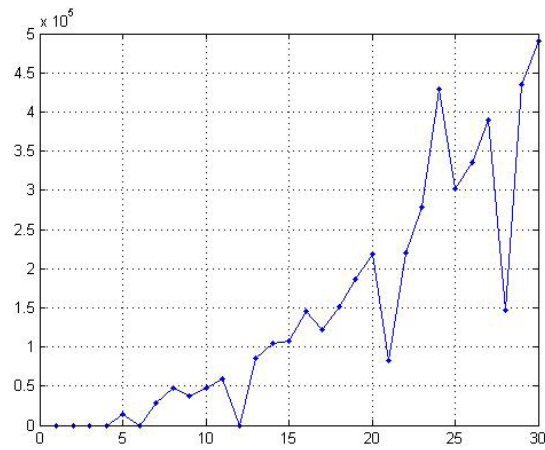
- Considering the quantization values used before ($q_1=23$, $q_2=17$) and the same input image, but varying q_3 in the range [1:70], the outcome is more complex to analyze than before. Several local minima arise and a strong one can be found in 65 (see Figure 27 (b)).
- Considering the same input image and q_3 in the range [1:30], but with different quantization coefficients $q_1=21$ and $q_2=12$ (they are both not prime numbers), the outcome reported in Figure 27 (c) is obtained. Without additional information about the input image, a wrong estimation could be performed (for example $q_1=28$).



(a)



(b)



(c)

Figure 27: Examples of error function values vs q_3 for the triple quantization: (a) $q_1 = 23$, $q_2 = 17$ and q_3 is in the range $[1:30]$; (b) $q_1 = 23$, $q_2 = 17$ and q_3 is in the range $[1:70]$; (c) $q_1 = 21$, $q_2 = 12$ and q_3 is in the range $[1:30]$.

When $q_3 = q_2$, the error function Eq. 23 is equal to zero. This motivates the absolute minima found in the previous examples.

Finally, the Eq. 23 allows estimating q_2 , but this information is not useful since q_2 is already known from the bitstream. Obtaining reliable q_1 estimation from Eq. 23 is a difficult task because too many cases must be considered. In the following an alternative strategy able to sensibly increase the reliability of the estimation is then proposed.

7.3. Proposed solution (formula based)

As explained above, Eq. 23 allows obtaining q_2 (already present in the header data) but is not able to provide a reliable estimation of q_1 . To solve this problem we have designed a new error function:

$$f'_e(c, q_1, q_2, q_3) = \text{abs} \left(\text{round} \left(\text{round} \left(\text{round} \left(\text{round} \left(\frac{c}{q_1} \right) \times \frac{q_1}{q_2} \right) \times \frac{q_2}{q_3} \right) \times \frac{q_3}{q_2} \right) \times q_2 - \text{round} \left(\text{round} \left(\frac{c}{q_1} \right) \times \frac{q_1}{q_2} \right) \times q_2 \right) \quad \text{Eq. 24}$$

To properly understand the behavior of Eq. 24, especially when $q_3 = q_1$, we have to better analyze the effect of a single quantization and de-quantization step. If we examine the behavior of the following function:

$$\text{round} \left(\frac{c}{q} \right) \times q \quad \text{Eq. 25}$$

We can note that if q is odd, all integer numbers in $\left[nq - \text{floor} \left(\frac{q}{2} \right) : nq + \text{floor} \left(\frac{q}{2} \right) \right]$ will be mapped in nq (with n a generic integer number). If q is even it maps in nq all the integer

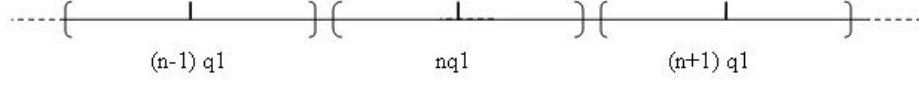


Figure 28: The effect of quantization and de-quantization for coefficient q_1 .

numbers in $\left[nq - \frac{q}{2} : nq + \frac{q}{2} - 1\right]$. From now on we call this range ‘space attributable to nq ’.

We can see an example of this in Figure 28 when q is even.

As a consequence of this behavior, we can say that Eq. 23 “groups” all integer numbers of its domain in multiples of q .

It is very important to observe that the maximum distance between a generic coefficient c and the corresponding \hat{c} , obtained by the quantization and de-quantization function, is $\frac{q}{2}$ if c is even, $\text{floor}\left(\frac{q}{2}\right)$ if c is odd. Based on the above observation, we analyze three rounds applied in sequence, when $q_2 < q_1$ and $q_3 = q_1$:

$$\text{round}\left(\text{round}\left(\text{round}\left(\frac{c}{q_1}\right) \times \frac{q_1}{q_2}\right) \times \frac{q_2}{q_1}\right) \times q_1 \quad \text{Eq. 26}$$

- $c_1 = \text{round}\left(\frac{c}{q_1}\right) \times q_1$ for the above observations leads to the situation shown in Figure 28;
- $c_2 = \text{round}\left(\frac{c_1}{q_2}\right) \times q_2$ maps multiples of q_1 in multiples of q_2 . It is worth noting that, being $q_2 < q_1$, a generic nq_1 will be mapped in a multiple of q_2 (for example mq_2) whose distance from nq_1 will be less than or equal to $\frac{q_2}{2}$ (or $\text{floor}\left(\frac{q_2}{2}\right)$ if q_2 is odd), then in the space attributable to nq_1 ;

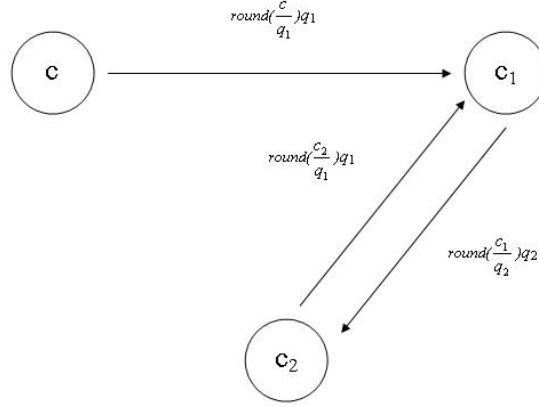


Figure 29: Scheme describing the effect of three quantization and de-quantization with coefficients q_1 , q_2 and $q_3 = q_1$ again.

- at this point, $\text{round}\left(\frac{c_2}{q_1}\right) \times q_1$ maps c_2 in nq_1 again, since, as pointed out in the preceding paragraph, c_2 is in the space attributable to nq_1 .

With the three steps above, we demonstrated that, as shown graphically in Figure 29:

$$\text{round}\left(\text{round}\left(\text{round}\left(\frac{c}{q_1}\right) \times \frac{q_1}{q_2}\right) \times \frac{q_2}{q_1}\right) \times q_1 = \text{round}\left(\frac{c}{q_1}\right) \times q_1$$

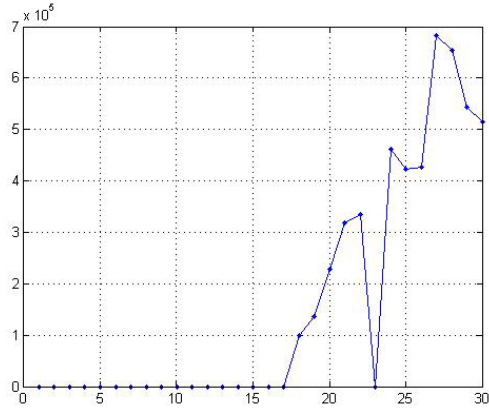
Eq. 27

Therefore the error function in Eq. 27 is zero when $q_3 = q_1$ regardless the c value.

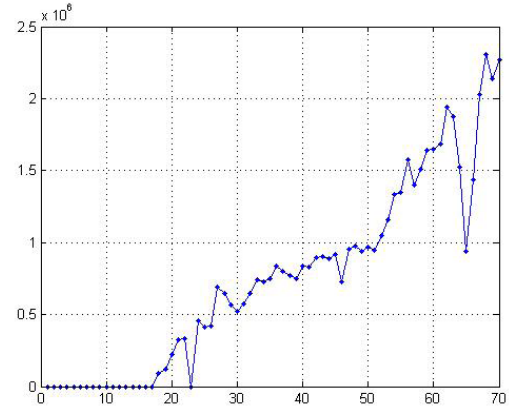
7.3.1. Experimental results

In order to prove the effectiveness of the proposed approach several tests and comparisons have been performed. A first test has been conducted considering artificial data. Specifically, a random vector of 5000 elements has been built by using a uniform

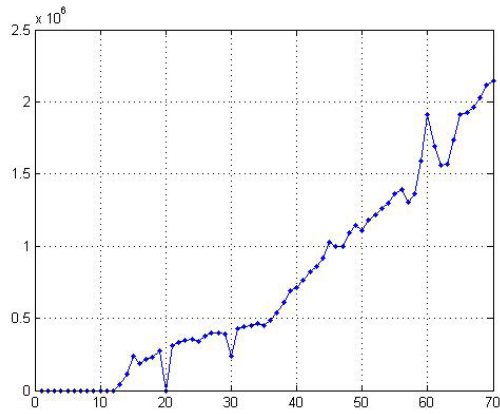
distribution in the range $[-1023:1023]$. The range corresponds to an input image within the range $[0:255]$ in the spatial domain. In fact, the output of the DCT transform, as it is defined, is three bits wider of the input bit depth and it is centered in zero. These simulated DCT coefficients are then used as input of the error function (Eq. 27) by considering several pairs of quantization coefficients with $q_1 < q_2$. As can be easily seen from Figure 30 (a, b and c), Eq. 27 has a global minimum (equal to zero) when $q_3 = q_1$. Moreover, q_1 value can be found in the range $q_3 \in]q_2, +\infty[$. It is worth noting that, sometimes, more than one local minimum can be found in the range $q_3 \in]q_2, +\infty[$. For example, this behavior can arise when q_1 is a multiple of q_2 (see Figure 30 (d)). However, even in this case, the correct q_1 value can be easily found. In fact, the additional minima are always between q_1 and q_2 and their positions depend on the common divisors between q_2 and q_3 . Hence, in this case, q_1 is a multiple of q_2 the correct value is the maximum one.



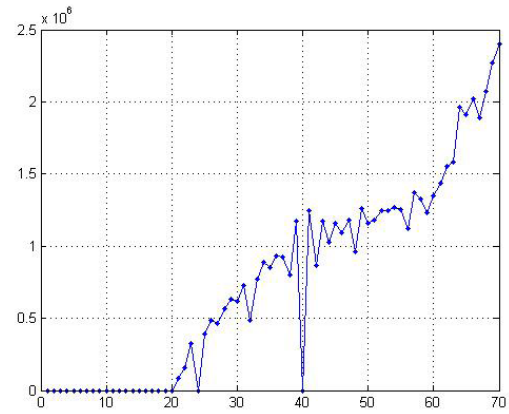
(a)



(b)



(c)



(d)

Figure 30: Examples of error function values vs q_3 , computed by using the proposed solution Eq. 24: (a) $q_1 = 23$, $q_2 = 17$ and q_3 is in the range $[1:30]$; (b) $q_1 = 23$, $q_2 = 17$ and q_3 is in the range $[1:70]$; (c) $q_1 = 20$, $q_2 = 12$ and q_3 is in the range $[1:70]$; (d) $q_1 = 40$, $q_2 = 20$ and q_3 is in the range $[1:70]$.

Further tests and comparisons have been performed by simulating a double compression process on real uncompressed images. Specifically, we randomly selected a photo from the Kodak image dataset [36] composed by uncompressed images, we divided it into 8×8 blocks of pixels and, for each block, we computed the DCT. Later, considering all the computed blocks of DCT coefficients, we collect in a vector the elements with the same relative

position in the block. The obtained vector has been then double quantized and dequantized, with two values q_1 and q_2 (where $q_1 > q_2$), randomly generated in a range of values $[1:30]$. Finally, starting from the double quantized vector, we evaluated both the error function Eq. 23 related to [33] and the proposed one Eq. 24. As can be easily seen from Figure 31 the proposed error function outperforms the one proposed in [33] providing, in all the considered cases, a clear global minimum (equal to zero) in correspondence of $q_3 = q_1$. On the contrary, obtaining q_1 values from Eq. 23, analyzing their local minima, is a difficult task (see, for example, Figure 31 (g)). Moreover, in some cases, the correct solution does not correspond to a minimum of Eq. 23 as in Figure 31 (c).

As already stated before, the proposed error function has a local minimum (equal to zero) when $q_3 = q_1$ and the correct solution has to be looked for in the range $]q_2, +\infty[$ (q_2 is already available). However, in some conditions, other minima could appear. For example, we have already described the behavior reported in Figure 31 (b and f) (q_1 is a multiple of q_2), that can be easily solved considering the maximum q_3 element with zero value. Other conditions can also produce additional zeros. Specifically, the distribution of the DCT coefficients, depending on the image content, can influence the final results (see Figure 31 (e, i and l)). Although in these cases we obtain more than a candidate (at most 3 in our experiments), the correct q_1 value corresponds to the maximum one.

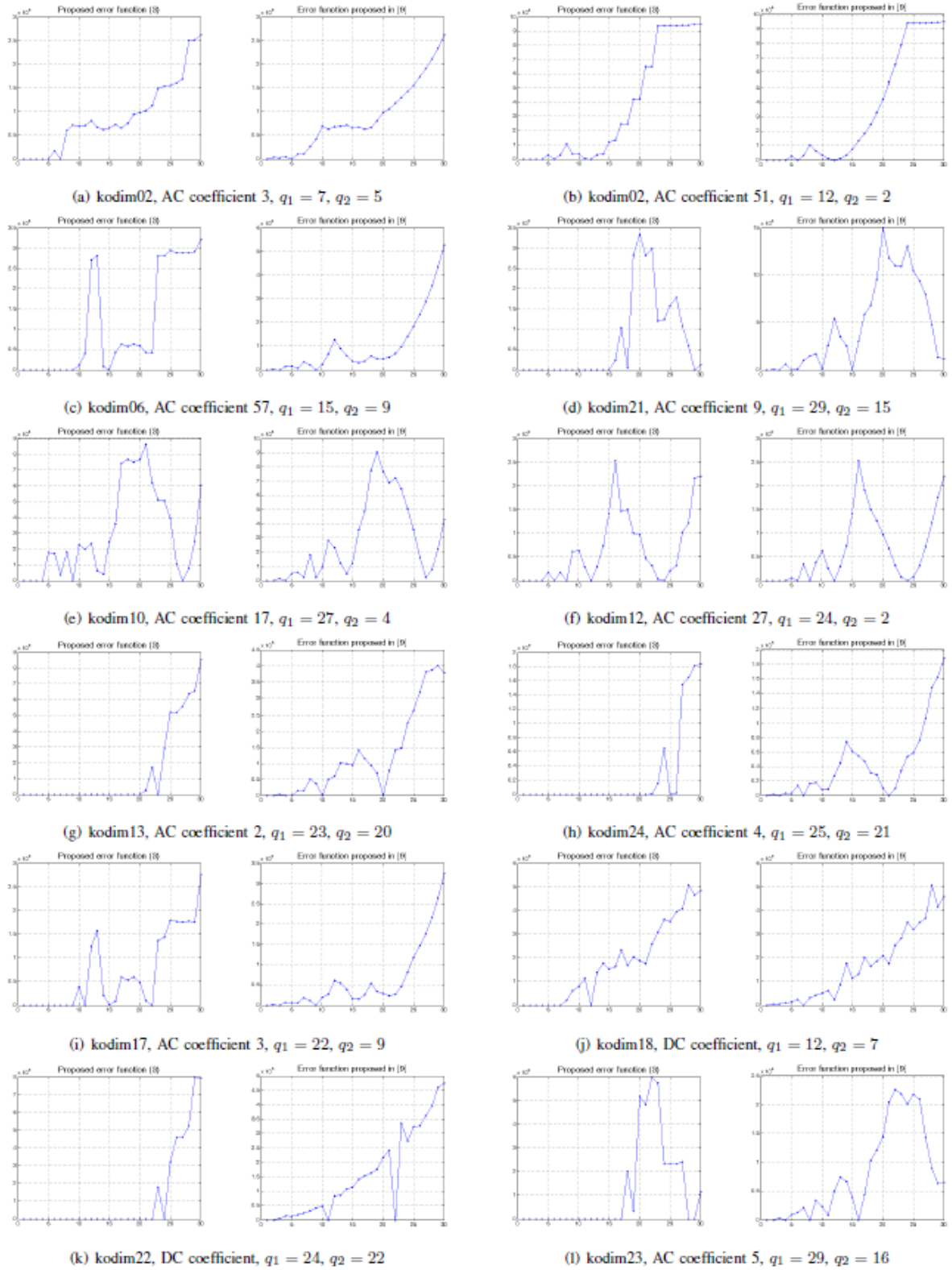


Figure 31: Test on real image dataset. The proposed function Eq. 24 is able to better estimate the q_1 quantization factor than Eq. 23 previously proposed in [33].

7.4. Proposed solution (analysis of the histogram of DCT bins)

In this paragraph a methodology to detect double quantization is explained. It is also able to reconstruct some coefficients of the original quantization matrix. It is based on the analysis of the histogram of bins of the dequantized DCT coefficients.

In particular it is based on the assumption that the quantization step projects the original space of the original DCT data in a subspace composed by a lower number of bins. A further quantization (i.e. double quantization) reduces more and more the cardinality of the new subspace.

In the Figure 32 an example of the histogram before and after a first and a second quantization is shown.

The same concepts explained in 7.3 are used for this method: the original DCT values span in the integer values; the values after the first quantization span in a subset (multiple of the quantization step); the values after the second quantization span in the subset depending on both the first and second quantization values. After Huffman decoding and dequantization steps, the histogram can be used to detect the first quantization coefficient.

In order to better explain the process, the graphs shown in the Figure 32 will be better described: The first graph shows the histogram of the DCT data after the DCT and rounding operation. All the integer bins are filled, i.e. all the integer values spanning theoretically in the range $[-\infty; +\infty]$. It is also important to observe that the histogram has a Laplacian distribution. It is a well-known behavior of the distribution of the AC coefficients [20], as described in Section 7. The second graph shows the histogram of the same image after the

first quantization and de-quantization. All the available bins are multiple of the quantization step (10 in this case). Each non zero bin (t) contains all the original values spanning in the range $[t-q_1/2, t+q_1/2]$. The third graph shows the histogram after the second quantization and de-quantization. In this case the bins are not all the multiples of the second quantizer (in this case 6), but some values are missed. As example, the bin 24 is zero because it should contains the bins, before the second quantization, in the range $[21, 27]$, but these values are not multiple of the first quantization table (i.e. 10), so the bin is empty.

The proposed solution makes use of the available non zero bins and the analysis of the anomalies in the histogram of the decompressed image to detect if double quantization occurred. Some values of the original quantization matrix can also be reconstructed. Since the bins are symmetrical respect to the zero value, the absolute value allows increasing their numbers and, hopefully, filling empty bins (see Figure 33).

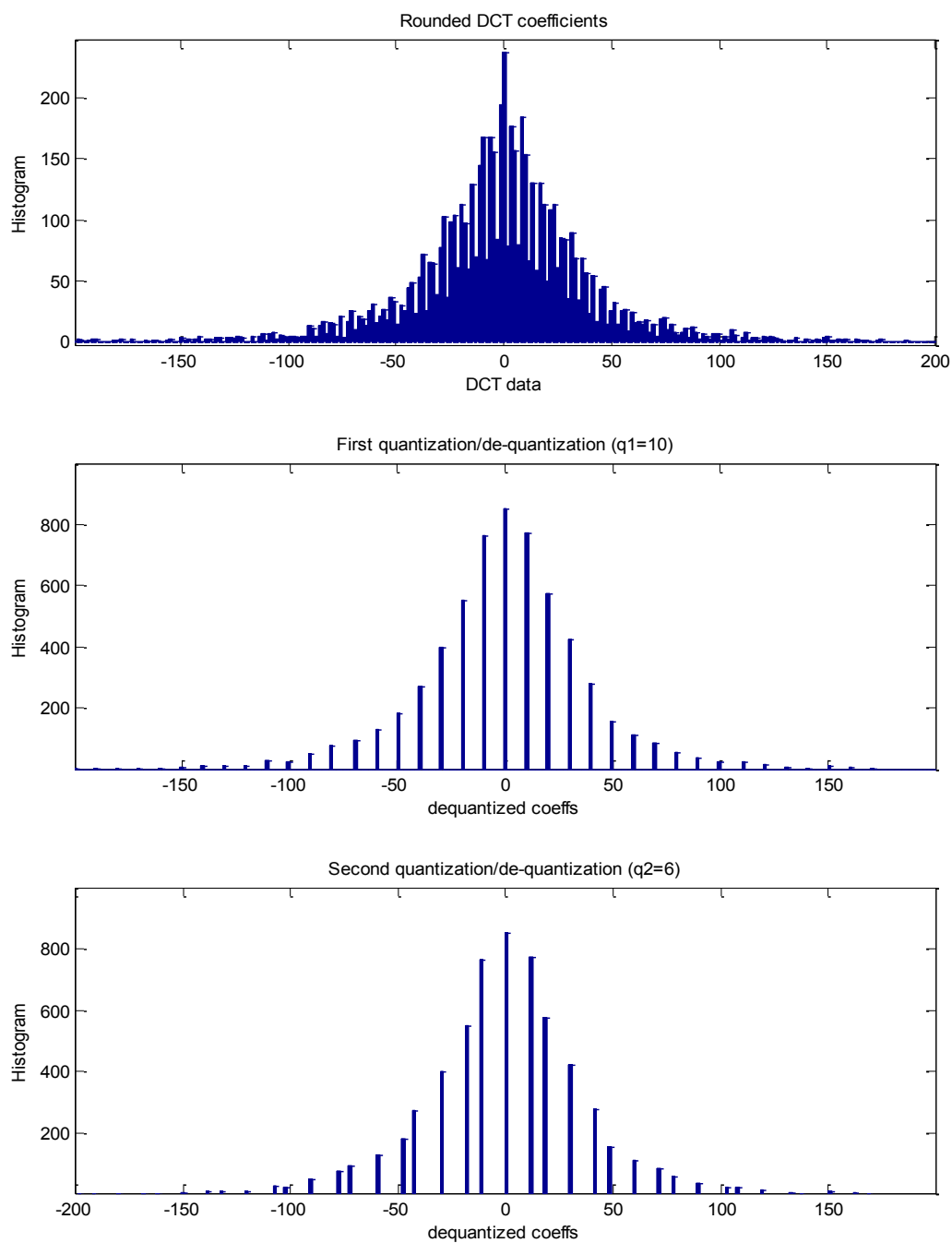


Figure 32: Quantization and double quantization effects on DCT data. Bins are accumulated in few coefficients (multiple of the quantization value). Double quantization produces some anomalies in the dequantized bins.

The idea is based on the following rationales:

1. Each non zero bin (k) is multiple of the second quantization step (see Figure 32);
2. The needed information can be extracted only through non zero bins analysis.
3. The symmetry of the histogram and the algorithm allows to consider the absolute value of each bin (see Figure 33).
4. The zero bin does not provide any information for the quantization value estimation.
5. Every non zero bin (k) collects all the coefficients of the first quantization step laying in the interval $[k-q_2/2, k+q_2/2]$.

Based on these assumptions, it is possible to build an iterative method to retrieve, for every position in the 8x8 grid of the block, the quantization coefficient of the first compression. Since some bins will not be available, especially for the highest frequencies, some coefficients will not be recovered uniquely. This method allows also providing a probability level of each of the reconstructed coefficient. In the following the methodology is explained. Examples will refer to the histogram in Figure 33 obtained by quantizing the DCT coefficients (0,2) with $q_1=10$ and $q_2=6$.

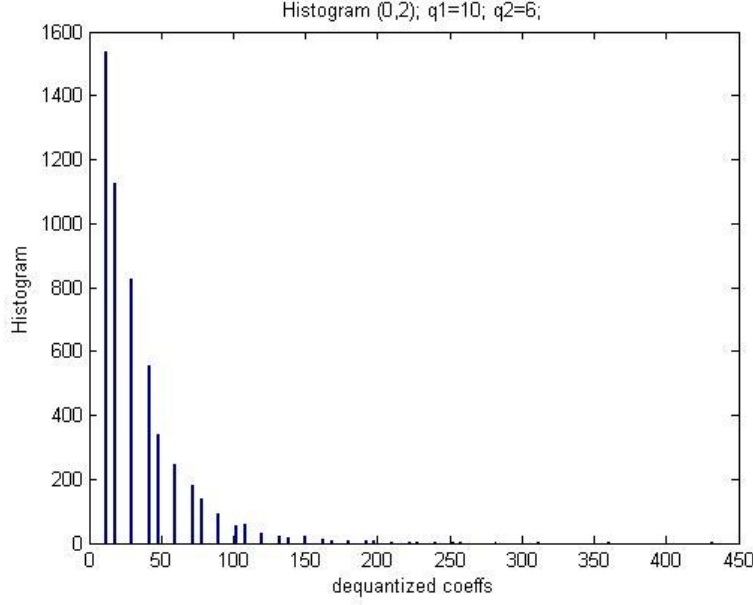


Figure 33: The histogram considered as example for the following paragraphs. Only the absolute value is taken into account.

It considers the absolute values of the bins. Moreover all the nonzero bins are taken into account. They are sorted and the list of each entry is taken into account only once. In this example the list of nonzero bins is:

$\phi =$

$$\left\{ 12, 18, 30, 42, 48, 60, 72, 78, 90, 102, 108, 120, 132, 138, 150, 162, 168, 180, 192, 198, 210, 222, 228, \right. \\ \left. 240, 252, 258, 282, 312, 360, 432 \right\}$$

Every bin $n_i \in \phi, i = 1, \dots, \text{card}(\phi)$ allows detect a list of candidates for the original quantization value. Based on the rationale (5) described previously, we have:

$$\bar{q}(n_i, k) \in \left[\text{ceil} \left(\frac{1}{k} \cdot \left(n_i - \frac{q_2}{2} \right) \right) : \text{floor} \left(\frac{1}{k} \cdot \left(n_i + \frac{q_2}{2} \right) \right) \right] \quad \text{Eq. 28}$$

Where \bar{q} is the estimated quantization value, $k \in \mathbb{N}$ and q_2 is the second quantization value (known). It means that the estimated values are not necessarily unique. As example, for $n_1=12$ and $q_2=6$ we will have:

$$\bar{q} \in \begin{cases} [4, 20] \text{ for } k = 1 \\ [2, 10] \text{ for } k = 2 \\ [2, 6] \text{ for } k = 3 \\ [1, 5] \text{ for } k = 4 \\ \dots \end{cases} \quad \text{Eq. 29}$$

For $n_2=18$ and $q_2=6$ we will have:

$$\bar{q} \in \begin{cases} [15, 21] \text{ for } k = 1 \\ [8, 10] \text{ for } k = 2 \\ [5, 7] \text{ for } k = 3 \\ [4, 5] \text{ for } k = 4 \\ \dots \end{cases} \quad \text{Eq. 30}$$

For $n_3=30$ and $q_2=6$ we will have:

$$\bar{q} \in \begin{cases} [22, 38] \text{ for } k = 1 \\ [11, 19] \text{ for } k = 2 \\ [8, 12] \text{ for } k = 3 \\ [6, 9] \text{ for } k = 4 \\ \dots \end{cases} \quad \text{Eq. 31}$$

This process may be iterated for all the $N = \text{card}(\phi)$ available bins. For any set of $\{k_1, k_2, \dots, k_N\}$ with $k_1 < k_2 < \dots < k_N$, it is possible to estimate if there is one or more quantizer values that is a possible quantizer value as the intersection of the corresponding sets of values. As example, for $k_1 = 1; k_2 = 2; k_3 = 3$; the possible first quantizer coefficient is in the following range:

$$\bar{q} \in \{[4,20] \cap [8,10] \cap [8,12]\} = [8,10]$$

for $k_1 = 1; k_2 = 3; k_3 = 4$; the possible first quantizer coefficient is in the following range:

$$\bar{q} \in \{[4,20] \cap [5,7] \cap [6,9]\} = \emptyset$$

That means that this is not a possible set. By varying all the possible k_i there are several allowed set. The possible values for the first quantization coefficient are the union of all the non-empty set.

It is possible to formalize this process.

A single set of values for a given set of k_i can be formalized as:

$$\bar{q} \in (\bar{q}(n_1, k_1) \cap \bar{q}(n_2, k_2) \cap \dots \cap \bar{q}(n_N, k_N)); k_i \in \mathbb{N}, i = 1, \dots, N; k_i < k_j \forall i < j;$$

Eq. 32

Then, for each set of values obtained with the Eq. 32, the first quantization coefficient belongs to the union of all the non-empty set.

Generalizing:

$$\bar{q} \in \bigcup (\bigcap_{i=1, \dots, N} \bar{q}(n_i, k_i)); \forall k_i \in \mathbb{N}, k_i < k_{i+1} \forall i \quad \text{Eq. 33}$$

This formula was implemented with an iterative method. The result, for each bin, is a collection of one or more candidates.

It is to be noted that the retrieval may be done for each of the 8x8 block position, hence 64 times for luminance and 128 for chrominance. In order to detect if a double quantization occurred, it is enough to detect only one coefficient where \bar{q} is not null.

The methodology may be also used to reconstruct the entire first quantization table, as described in the following Section.

7.4.1. Quantization matrix reconstruction

The method described above allows retrieving a value or a set of values for each of the 8x8 quantization coefficient. It allows reconstructing the first quantization value as the collection of \bar{q} values with only one value. It allows detecting without any doubt the original quantizer. The other coefficients, i.e. the set containing more than one value, are due to lack of information (i.e. few nonzero bins). In this case we may choose one of these values according to a heuristic methodology. In case of uncertainty, the probability of the estimation is $1/\text{card}(\bar{q})$.

A possible heuristic is also described in the next Section.

7.4.2. Experimental results

The method described above has been implemented in Matlab. It was tested by using the image database in [36]. In this Section is shown the results obtained with one of the image of the database, in particular kodim13.png.



Figure 34: Kodim13.png, the image used for the tests shown in this Section.

The image was quantized first with the following quantization table:

$$Q_1 = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

The second quantization is the following:

$$Q_2 = \begin{bmatrix} 10 & 7 & 6 & 10 & 14 & 24 & 31 & 37 \\ 7 & 7 & 8 & 11 & 16 & 35 & 36 & 33 \\ 8 & 8 & 10 & 14 & 24 & 34 & 41 & 34 \\ 8 & 10 & 13 & 17 & 31 & 52 & 48 & 37 \\ 11 & 13 & 22 & 34 & 41 & 65 & 62 & 46 \\ 14 & 21 & 33 & 38 & 49 & 62 & 68 & 55 \\ 29 & 38 & 47 & 52 & 62 & 73 & 72 & 61 \\ 43 & 55 & 57 & 59 & 67 & 60 & 62 & 59 \end{bmatrix}$$

It is to be remarked that the Q1 is the standard proposed quantization table, while the Q2 is obtained from the previous with a scalar multiplier (in this case 0.6). It is the usually used methodology to modify the compression ratio of the image (i.e. the visual quality of the compressed image).

Concerning the coefficient reconstruction we will show only few examples:

DC coefficient:

It is the first coefficient in the position (0,0). The histogram is shown in Figure 35.

The main characteristic of this histogram is that it is usually composed by several bins not necessarily centered in the zero value. By applying the methodology described above, the value 16 is the only possible (100% of probability).

AC coefficient, position (0,1):

The histogram (of the absolute values) is shown in Figure 36.

In this case we have a sufficient high number of bins; the reconstruction method provides the value 11 (100% of probability).

AC coefficient, position (0,4):

The histogram (of the absolute values) is shown in Figure 37.

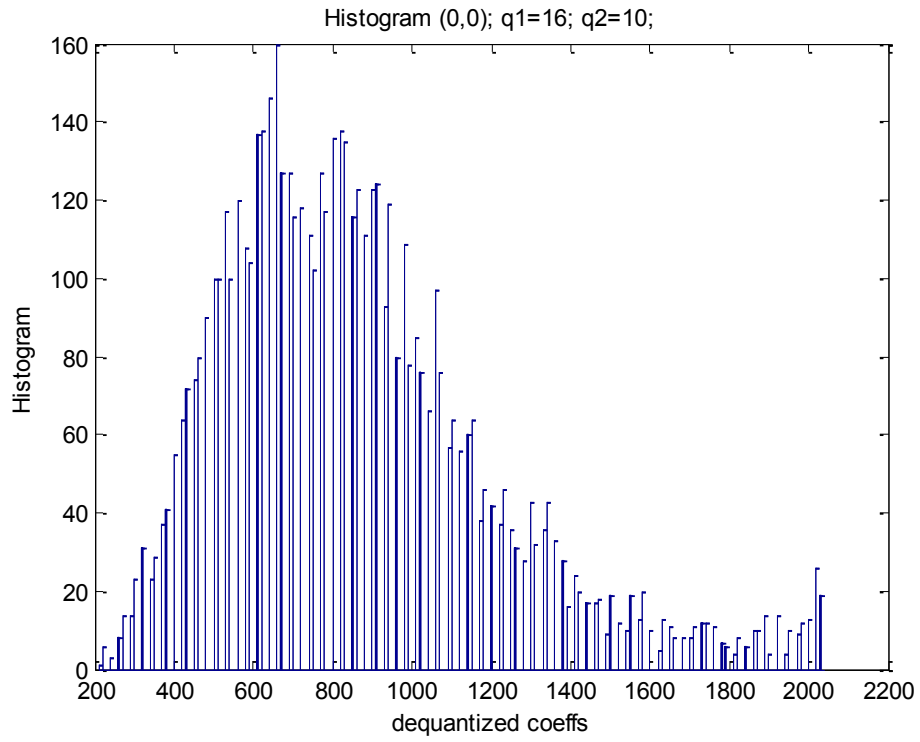


Figure 35: histogram of the DC coefficient (0,0).

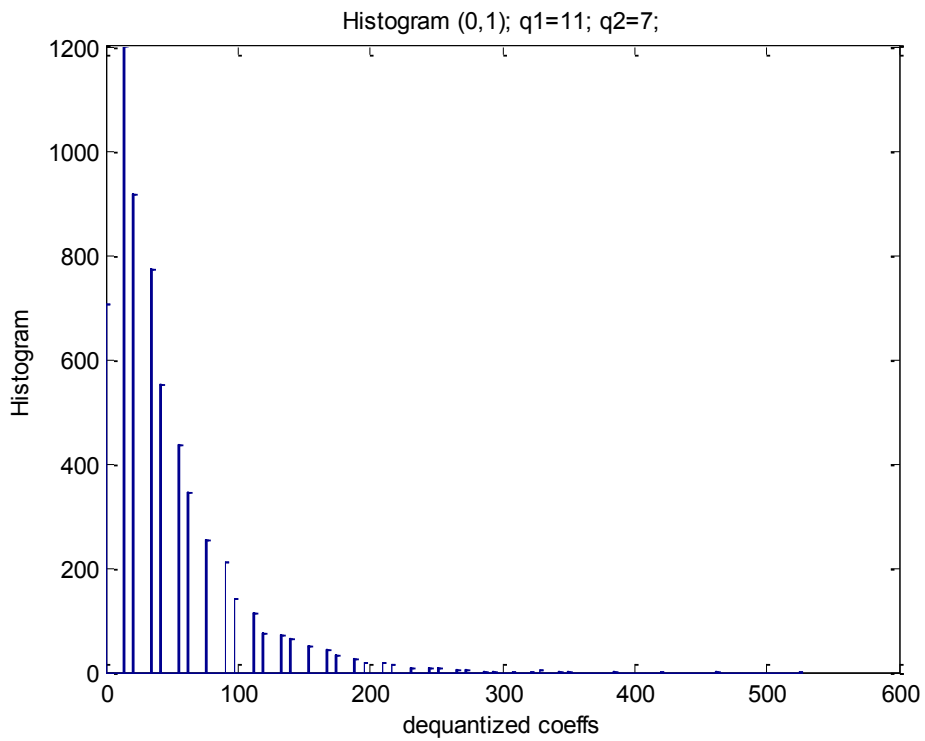


Figure 36: histogram of the AC coefficient (0,1).

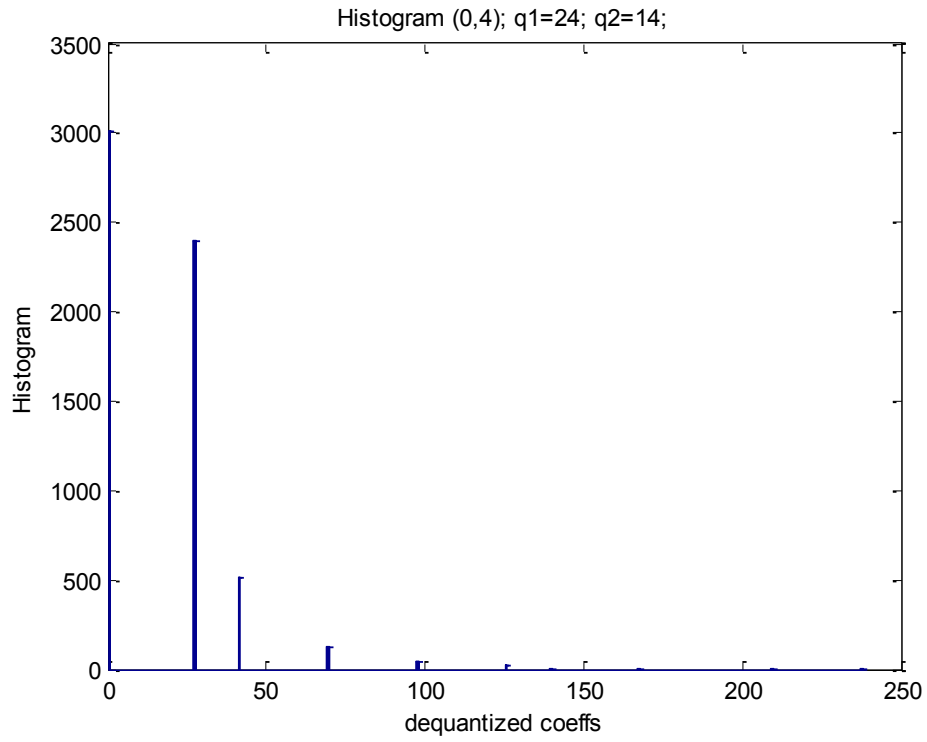


Figure 37: histogram of the AC coefficient (0,4).

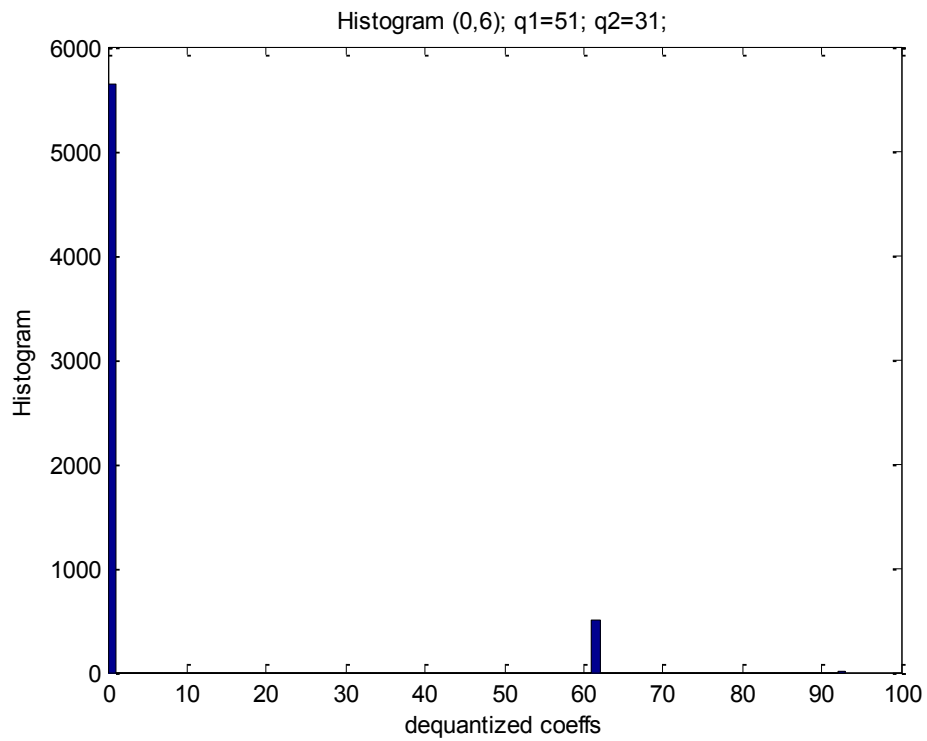


Figure 38: histogram of the AC coefficient (0,6).

It is composed by fewer bins than previously. In this case the retrieving method provides 3 values: $\phi = \{15,21,24\}$. The probability is $1/\text{card}(\phi)=1/3=.333$ (33%) for each result.

AC coefficient, position (0,6):

The histogram (of the absolute values) is shown in Figure 38.

In this case the number of bins is very low and there is a lot of uncertainty. The retrieving method estimates the following values: $\phi = \{32,33,34,35,36,47,48,49,50,51,52,53,54\}$. The probability is $1/\text{card}(\phi)=1/13=0.08$ (8%).

Summarizing, the reconstructed quantization matrix (considering only the values with 100% of probability) is:

16	11	10	16	NaN	NaN	NaN	NaN
12	12	NaN	NaN	NaN	NaN	NaN	NaN
14	NaN	16	NaN	NaN	NaN	NaN	NaN
14	NaN	NaN	29	NaN	NaN	NaN	NaN
18	22	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

And the probability mask is:

1.00	1.00	1.00	1.00	0.33	0.11	0.08	0.02
1.00	1.00	0.50	0.50	0.33	0.08	0.06	0.02
1.00	0.50	1.00	0.33	0.17	0.08	0.02	0.02
1.00	0.50	0.50	1.00	0.08	0.05	0.02	0.02
1.00	1.00	0.20	0.11	0.06	0.01	0.01	0.02
0.33	0.25	0.08	0.06	0.05	0.01	0.01	0.01
0.17	0.07	0.05	0.02	0.01	0.01	0.01	0.01
0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01

Where NaN means that the estimated value is not unique. As expected the uncertainty is bigger at higher spatial frequencies for two reasons:

- Higher frequencies are less populated than the lower frequencies (i.e. statistically the higher frequency components are zero)
- The quantization values increase according to the frequency

The previous result is enough to assert that the image has been re-quantized and to understand that, probably, the first quantization table was the standard one. Anyway, in order to increase the number of estimated values, it is possible to use some heuristics, as described below.

Assumption of the biggest value

This assumption is based on the statistical distribution of the AC coefficients. As stated above, the AC data are distributed according to a Laplacian function. It means that,

probably, all the not null coefficients fill entirely all the lowest non-zero quantized values.

This assumption corresponds to impose in the Eq. 33 $\bar{q} \in U(\cap_{i=1,\dots,N} \bar{q}(n_i, k_i)); \forall k_i \in \mathbb{N}, k_i < k_{i+1} \forall i$ that $k_j = j \forall j$. The Eq. 33 can be formalized as following:

$$\bar{q} \in U(\cap_{i=1,\dots,N} \bar{q}(n_i, i)); \quad \text{Eq. 34}$$

Since the assumption is almost true for the lower coefficients but can be false for the higher (when the probability value is very low and some intermediate coefficients' probability can be zero), the index i must be stopped when the set ϕ contains only one number.

It is possible to determine directly from the results of the Eq. 33 the same results of Eq. 34 without running two separate processes: the assumption correspond to state that, statistically, the biggest value of the ϕ set resulting from Eq. 33 is the most probable quantization factor (hence the name of the assumption). Moreover it is true if the difference between the latest and the previous one is bigger than one, otherwise all the consecutive values (starting from the biggest one) are equally probable. The theory behind is that the biggest value represents the number having lower and consecutives iterations. The others are spurious (e.g. submultiples etc.). If there are more than one consecutive value, it means that there is uncertainty in the reconstruction value, i.e., the Eq. 34 produces more than one quantization factor, while if there is only one value and the others are with difference bigger than one, means that it is the best candidate to be the quantization factor. As example, in the AC coefficient, position (0,4) shown above, the possible values are $\phi = \{15, 21, 24\}$. The 24 fits the rules (it is the biggest value and the difference from the previous $24 - 21 > 1$). Since there is not the absolute certainty that this coefficient is the real one, (the assumption is probably true but not necessarily true) we consider the probability 0.5 (50%).

Conversely, the AC coefficient, position (0,6) provides the following quantization values:
 $\phi = \{32,33,34,35,36,47,48,49,50,51,52,53,54\}$. Since $54-53=1$, the biggest value is not
probably the quantization coefficient (but probably it is in the range [47-54]).

Thanks to this assumption, the estimated quantization table can be probably filled more and
more with other coefficients. For the example used in this Section, the estimation using also
the assumption is:

16	11	10	16	24	NaN	NaN	NaN
12	12	14	19	26	NaN	NaN	NaN
14	13	16	24	NaN	NaN	NaN	NaN
14	17	22	29	NaN	NaN	NaN	NaN
18	22	NaN	NaN	NaN	NaN	NaN	NaN
24	35	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

And the probability mask is:

1.00	1.00	1.00	1.00	0.50	0.11	0.08	0.02
1.00	1.00	0.50	0.50	0.50	0.08	0.06	0.02
1.00	0.50	1.00	0.50	0.17	0.08	0.02	0.02
1.00	0.50	0.50	1.00	0.08	0.05	0.02	0.02
1.00	1.00	0.20	0.11	0.06	0.01	0.01	0.02
0.50	0.50	0.08	0.06	0.05	0.01	0.01	0.01
0.17	0.07	0.05	0.02	0.01	0.01	0.01	0.01
0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01

7.4.3. Real use case: JPEG double compression of still images

The aforementioned method is able to detect double quantization. But, in real cases, the compression pipeline is not composed by only the quantization, but other transform domain (e.g. RGB to YCrCb, YCrCb to DCT etc.) and intermediate rounding and clipping operations may deceive the system. In this paragraph we take into account such problems considering, in particular, the JPEG algorithm.

In the Figure 39 and Figure 40 the block based pipeline of both the JPEG encoder and decoder are shown. In these figures the rounding and clipping operations are highlighted.

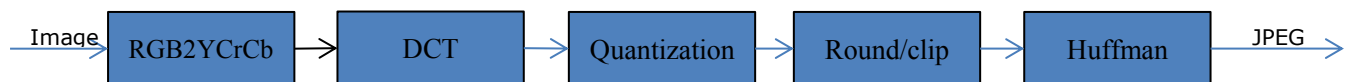


Figure 39: JPEG encoder block based schema with rounding/clipping blocks.

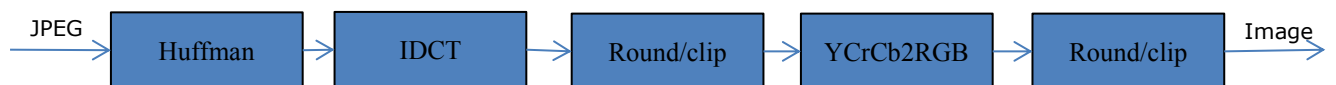


Figure 40: JPEG decoder block based schema with rounding/clipping blocks.

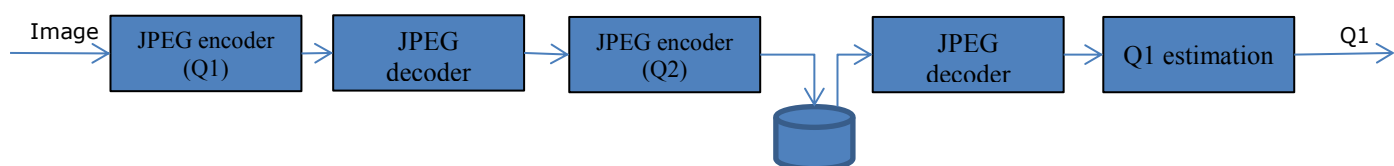


Figure 41: Double JPEG compression schema used for analysis.

It is to be noted that there could be also other rounding and clipping between other blocks (e.g. between RGB2YCrCb and DCT block), depending on the implementation.

One of the most used JPEG implementation is the *cjpeg* [26]. We used this implementation for the analysis. The methodology is described in Figure 41.

The histogram of the DC coefficient is shown in the next Figures. Compared to the theoretical analysis, the histogram contains a sort of 'dirty' bins. It can deceive the iterative algorithm, but these bins can be removed by using a threshold. The threshold can be hard threshold or adaptive. As example, for the AC coefficients, since their distribution is usually Laplacian, a threshold taking into account the Laplacian distribution can be used. It is to be noted that also the number of peaks are reduced. It means that the reliability of the method decreases compared to the theoretical analysis. Ongoing analysis is devoted to converge versus a stable computational strategy.

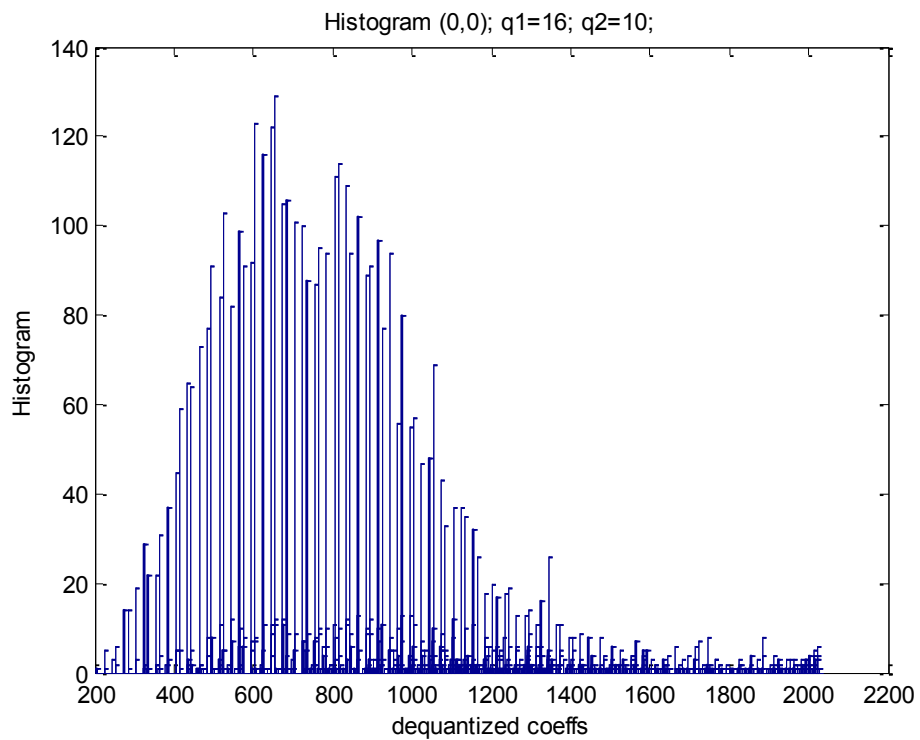
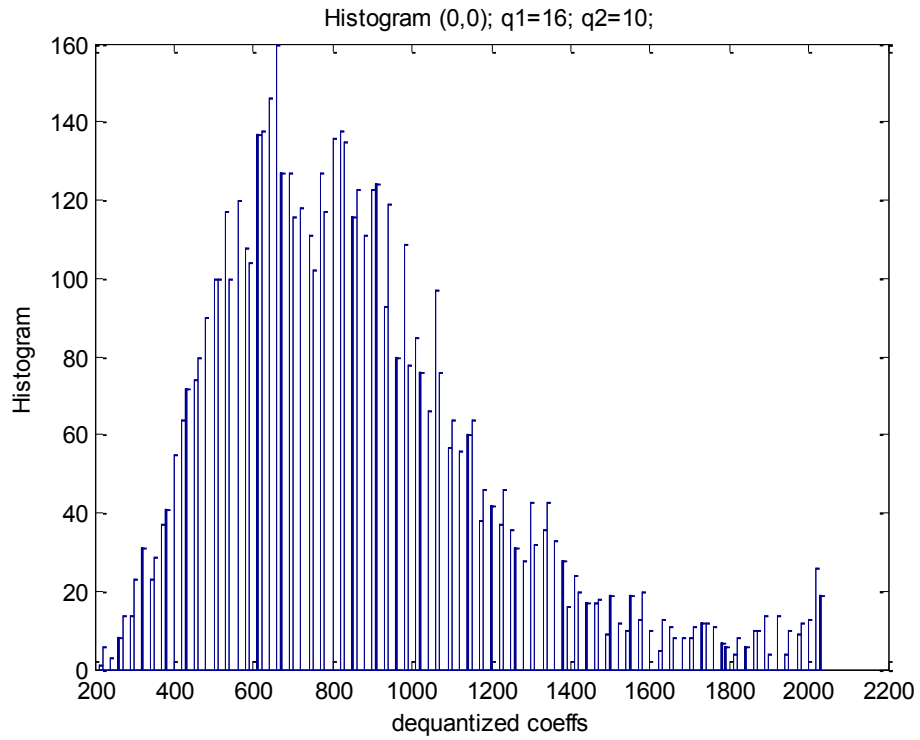


Figure 42: DC coefficient (theoretical approach vs double JPEG compression).

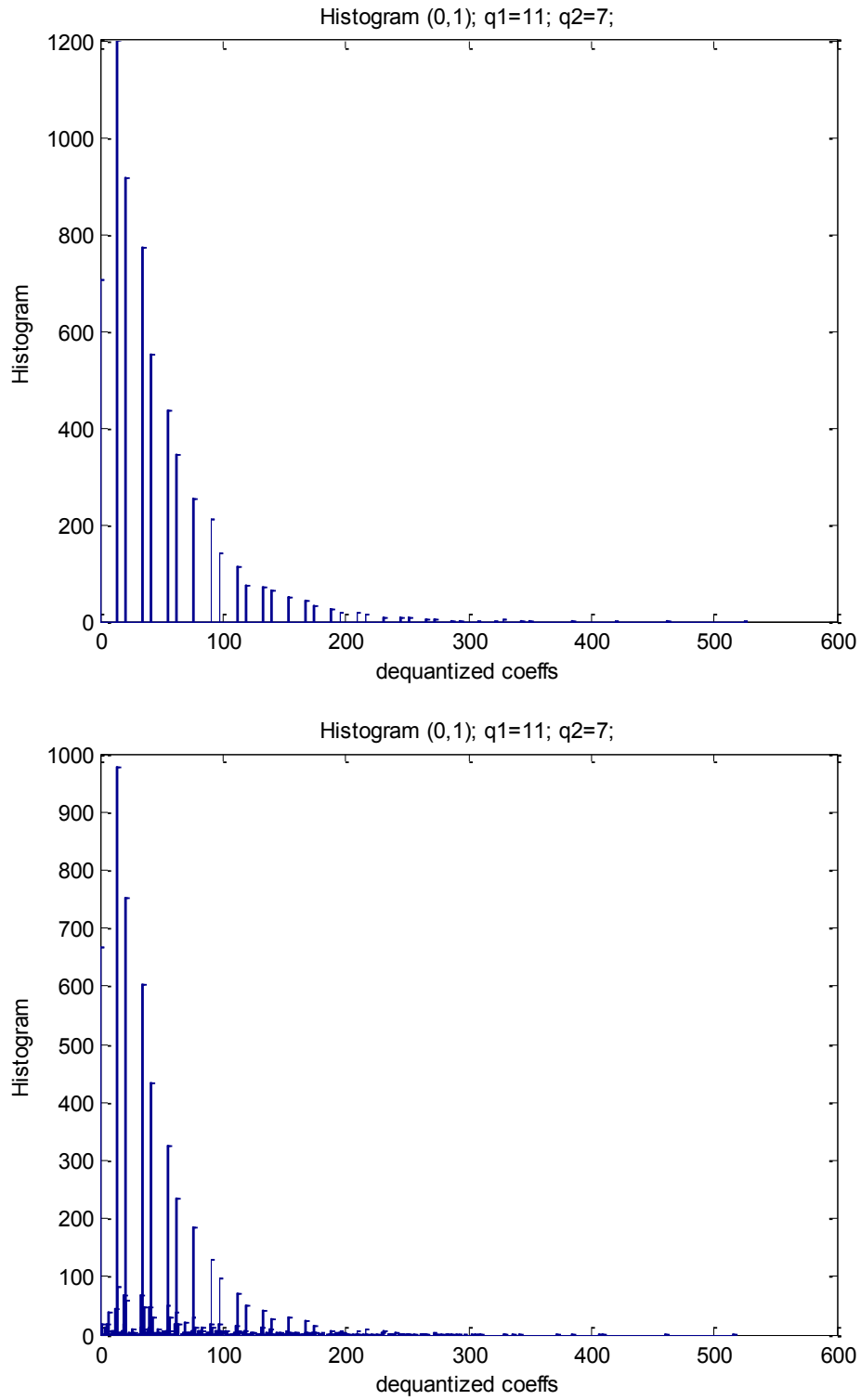


Figure 43: AC(0,1) coefficient (theoretical approach vs double JPEG compression).

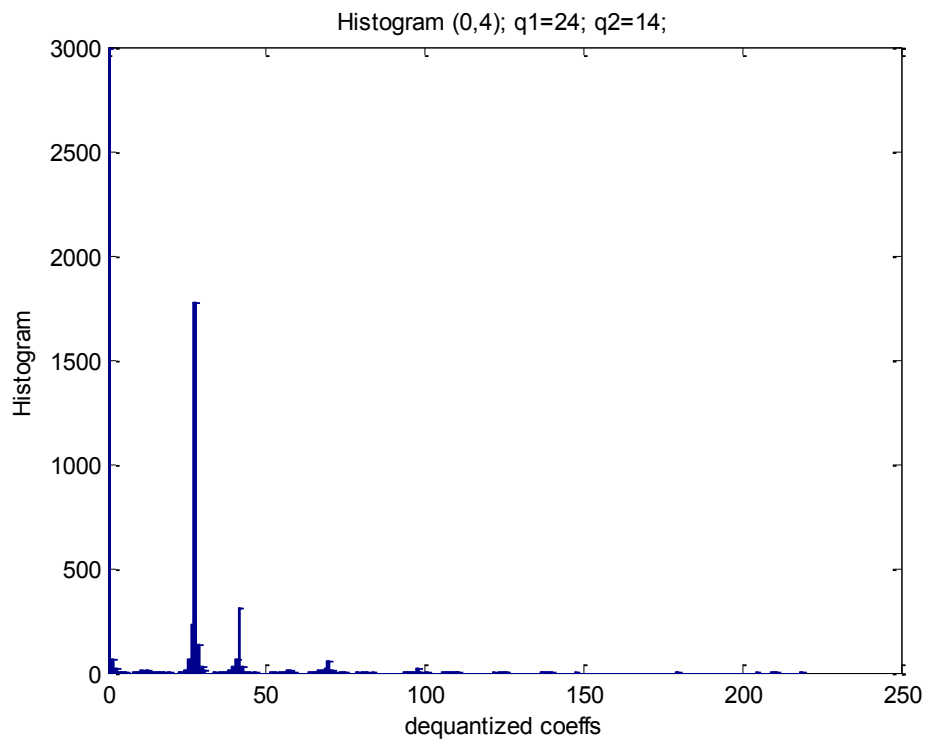
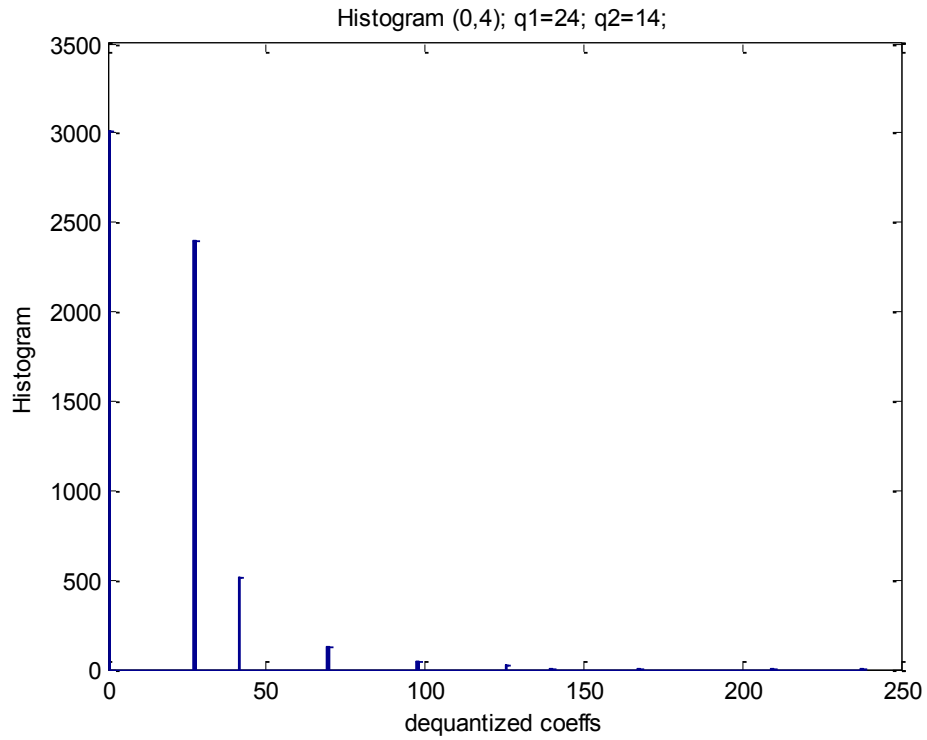


Figure 44: AC(0,4) coefficient (theoretical approach vs double JPEG compression).

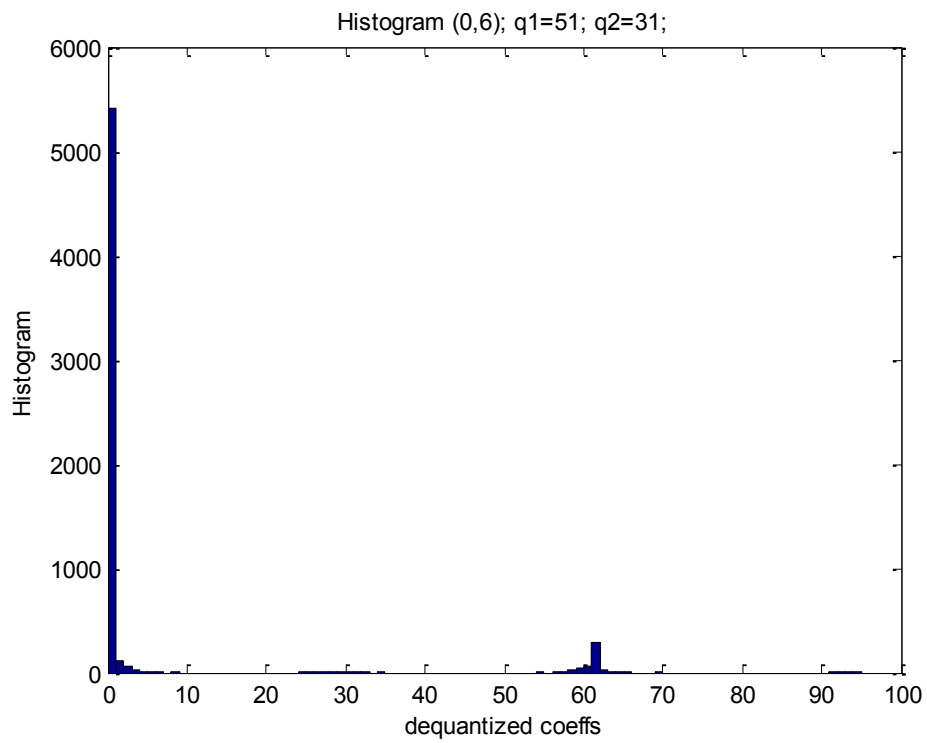
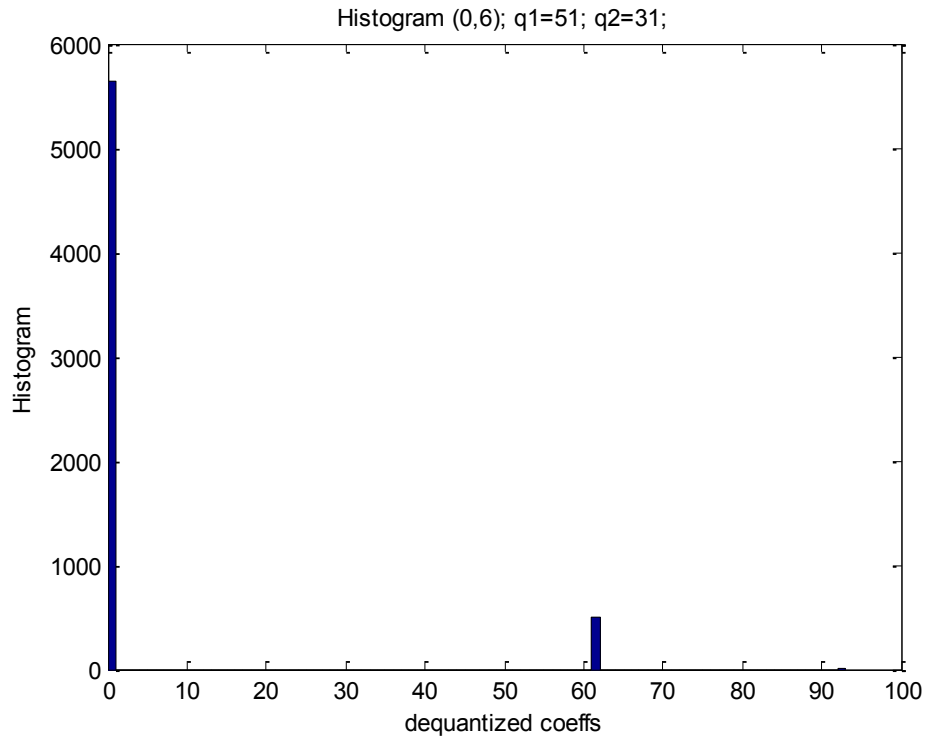


Figure 45: AC(0,6) coefficient (theoretical approach vs double JPEG compression).

8. Conclusions and future works

In this thesis several methods were analyzed to detect cropping and image recompression. Both state of the art and new methodologies were taken into account. In particular two techniques have been described: the first one analyze a formula able to provide immediate information about the first quantization coefficient. The second one is an iterative algorithm with the same goal. Moreover the additional domain transforms, rounding and clipping of the real JPEG algorithm have been considered for the real usage of the iterative algorithm. Although not fully developed, ideas were developed to be further analyzed. Next works could be to also consider such problems to the formula based algorithm and to increase the performances of the iterative one by adding other heuristics. Another task to be performed is the analysis of the case $q_1 \leq q_2$.

PART II

Banknotes Counterfeit Detection

9. Abstract

In this section both is shown the research activity related to the counterfeits detection of euro banknotes. It is also able to recognize the banknote values. The system is composed by both hardware and software components. The proposed method makes use of images acquired with near infrared camera and works without motors or any other moving parts. This makes the overall system low cost.

The proposed system is reliable to ambient lights and banknote positioning. Users should simply lean the banknote to be analyzed in a flat glass and the system detects forgery, as well as recognizes the banknote value. The effectiveness of the proposed solution has been properly tested on a dataset composed by genuine and fake euro banknotes.

10. Introduction

The detection of counterfeit banknotes is one of the most important tasks in billing machines. Usually it is performed by employing several image analysis techniques (transmittance analysis, reflectance analysis, etc.) on different light spectra (visible, infrared and ultraviolet). Unfortunately these systems are very expensive and are used only for ATM machines where a high degree of reliability is required since those machines are fully unsupervised by humans. In the last years, cheaper systems for validation and classification of banknotes have been commercialized. They are usually based on motors aimed to let the banknote pass through light emitters and sensors in a dark area.

This allows to have controlled light. On the other hand, the motors are expensive and the mechanical parts frazzle in few years.

In this paper we present both, hardware and software components useful to detect counterfeit banknotes and to recognize their currency value. This is obtained exploiting a low cost system without moving parts. A prototype of the proposed system has been built and tested on a dataset composed by genuine and fake euro banknotes provided by an Italian Bank. The system is composed mainly by an infrared camera, a microprocessor and a software component. A glass is placed in the focal plane of the camera in order to acquire a focused image. The acquired image is then processed through the designed algorithms executed by the micro. The user should just lean a banknote on the glass and the system provides the information on the validity and the related value.

In Section 11 reviews state of the art algorithms in this field. Section 12 summarizes the proposed algorithms for counterfeit detection and currency value recognition of euro

banknotes. In Section 13 the realized hardware prototype is described, whereas experimental results are presented in Section 14.

11. State of the art

It is well known that, in order to avoid forgeries, the security systems of money are typically encoded in the banknotes (in different ways in the different currencies). This fact has induced researchers to develop different counterfeit detection algorithms taking into account the different currencies.

In [37] authors make use of light transmittance and pattern recognition techniques to recognize the value of banknotes. The proposed solution requires the banknote to pass through light emitters (LED) and receivers (photo transistors) placed in opposite side. Hence, the system requires of a motor or a moving part. Moreover, the method can be simply deceived by counterfeit banknotes (not handled by the authors).

In [38] is described an algorithm for Korean won bills classification. It makes use of images acquired with visible light. The recognition is performed by extracting features in the wavelet domain. Banknote images are first processed with a Sobel filter and then the wavelet feature extraction is performed. Also in this case the problem of detecting counterfeit banknotes is not considered.

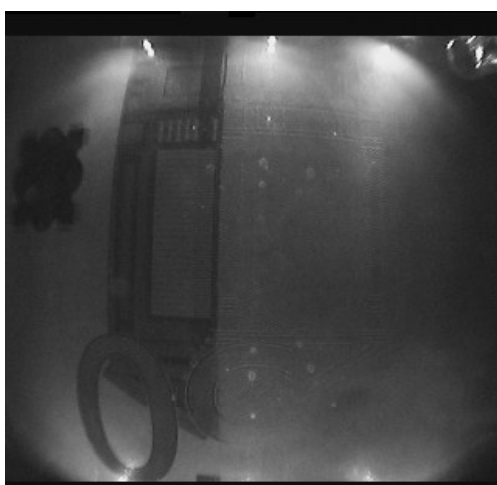
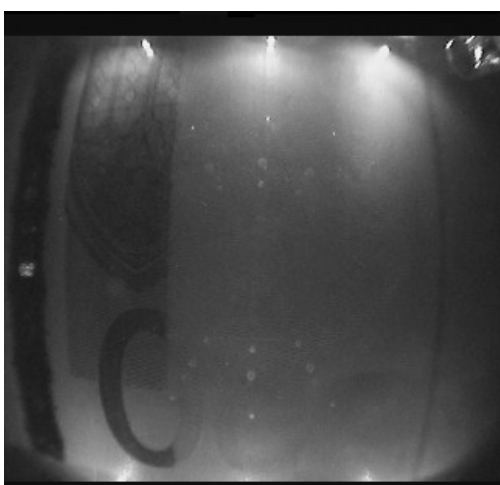
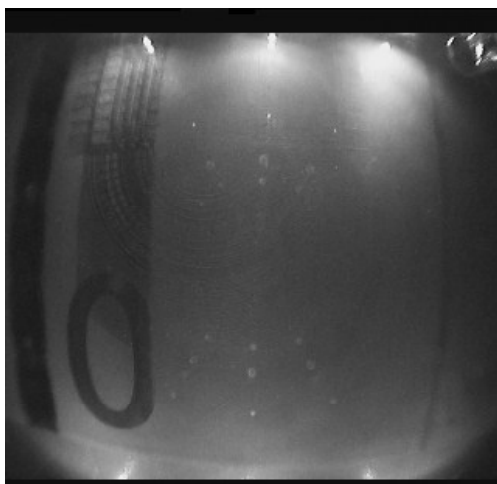
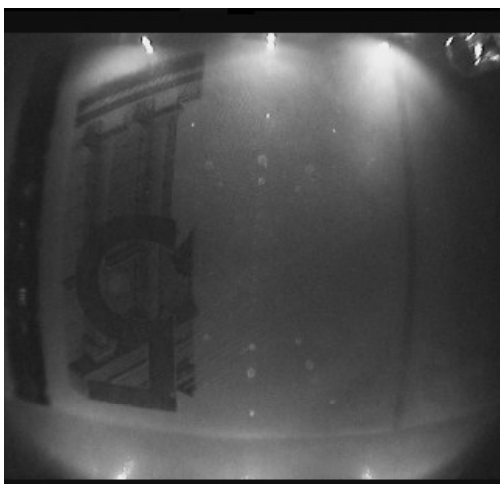
In the approach proposed in [39] the banknote is segmented into different regions and a classifier for each segmented region is employed. A consensus on the results obtained from the different classified regions provides the final decision. Specifically, genetic algorithms were used for classification of both, the segmented regions and fusion blocks.

A neural network and genetic algorithms has been exploited in [40], [41] to address the problem of banknote recognition, whereas in [42] the authors proposed an Intelligent

Banknote Identification System (IBIS) based on neural networks technique. The system is designed for Turkish Lira and Cyprus Pounds identification.

Despite different approaches have been presented in literature, most of them do not take into account the Euro Banknotes. In the Euro banknotes there are different features which make difficult the forgery: the sheet, the watermarks, the particular inks with different behavior in visible, infrared and ultraviolet lights, etc. All the techniques above could be extended to other currencies (e.g., Euro), but their adaptation to counterfeit detection is not straightforward because, in the visible light, the counterfeit banknotes usually looks like the valid ones. In the proposed system the infrared (IR) image has been used since it provide good and robust features for counterfeit detection of Euro banknotes.

An example of a euro banknote IR image is shown in the Figure 46. The two different inks present on the euro banknotes (one reflective to IR and one absorbing IR) allow to obtain useful features for the considered task.



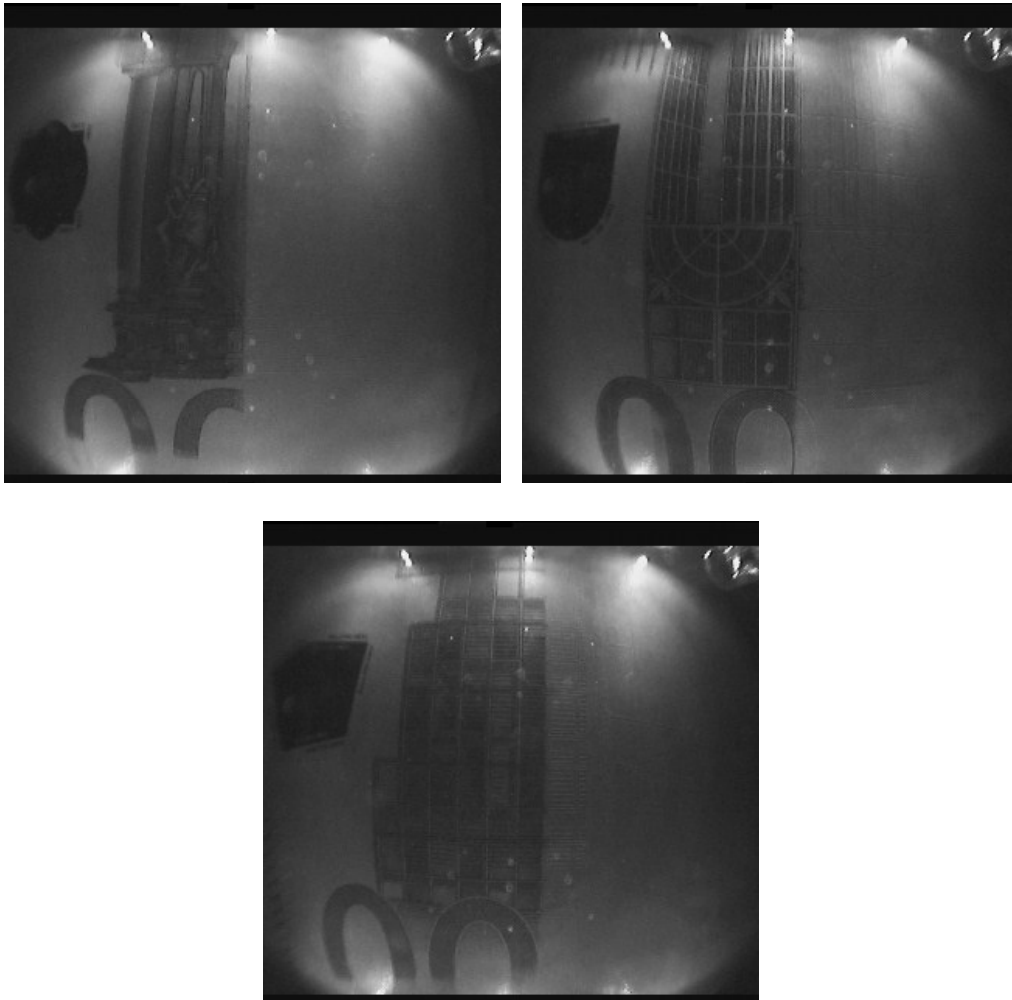


Figure 46: The set of euro banknotes (5, 10, 20, 50, 100, 200 and 500 euro) as seen the IR camera of our system.

12. Validation and classification

The overall scheme of the developed software for infrared based counterfeit detection and currency value recognition is shown in Figure 47. The software consists of three main blocks: Calibration, Training and Use Module.

12.1. Calibration

Since LED illumination is not spatially uniform (see Figure 46), and taking into account its changes from prototype to prototype, a calibration phase is required to remove illuminations variability. It can be considered a foxed pattern noise, since it is constant along the time. Specifically, in this phase a brightness map is acquired and a compressed version of it is stored inside the flash memory. To obtain the brightness map we captured an image of a white sheet of paper under an idealized lighting condition (i.e. dark room, with no source of infrared light). Since the captured frame is very noisy, a multi frame acquisition is performed and the map is obtained as the mean of such frames. It allows reducing the zero mean Gaussian noise [47], [48] and [49].

The map is then used during by Training and Use Modules to normalize the input banknotes images as following:

$$\mathbf{NormalisedImage}(i,j) = \frac{\mathbf{InputImage}(i,j)}{\mathbf{BrightnessMap}(i,j)} \quad \text{Eq. 35}$$

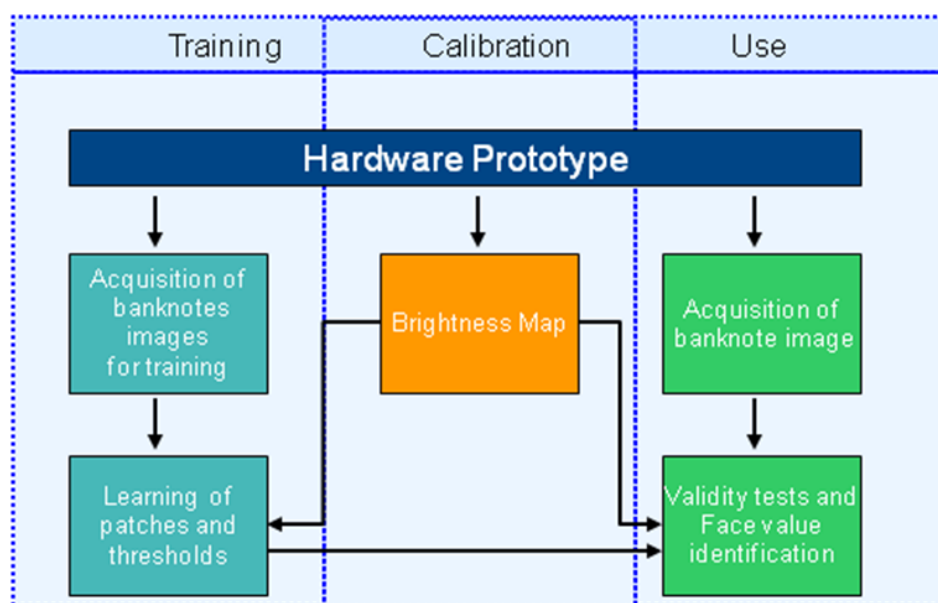


Figure 47: Scheme of the proposed approach.

12.2. Training

The Training block is used to learn optimal features of the banknotes (i.e., patches), which will be used to determine the validity and the value of each banknote. The parameters are learned using a large data set of both genuine and counterfeit banknotes, which come in several face values.

The training dataset of 1000 images has been acquired taking into account typical contexts of use of the apparatus, in which there are very different lighting conditions (e.g. neon, sunlight, incandescent and fluorescent lamps, etc.) and a high degree of misalignment with respect to the proscenium (the system must be robust to slightly translated and or rotated banknotes).

Genuine euro banknotes are made such that only specific visual features are visible under infrared lighting (see Figure 46 as example). In particular banknotes must show darker areas

in different zones depending on their value. Those areas always show characteristic patterns. However, there are overlaps among all banknote values, for both dark and bright areas, which can be used as validity check.

The aim of learning stage was hence to find the best area to check banknote validity by using a training dataset composed by genuine and fake euro banknotes.

Given the set of all genuine banknotes $G = \{G_1, G_2, \dots, G_n\}$, the training block is devoted to search for the largest common dark area F_g , and the largest common bright area T_g .

Let $F_{gi} = IRSignal(G_i)$ be the infrared highlighted dark area of each G_i , then $F_g = \cap_i F_{gi}$.

Let $T_{gi} = \frac{G_i}{F_{gi}}$ be the unresponsive bright area of each G_i , then $T_g = \cap_i T_{gi}$.

The regions T_g and F_g need to be refined in order to be robust to counterfeit banknotes, which in some case might show a slightly similar infrared response. In a similar way, given the set $C = C_1, C_2, \dots, C_n$ of all counterfeit banknotes in our data set, we define $F_{cj} = IRSignal(C_j)$ and $T_{cj} = \frac{C_j}{F_{cj}}$.

Let C^* be the set of all C_j such that $((F_{cj} \cap F_g \neq \emptyset) AND (T_{cj} \cap T_g \neq \emptyset))$.

We can finally define T and F as follows: $T = \frac{T_g}{\cup_j T_{cj}}$ and $F = \frac{F_g}{\cup_j F_{cj}}$.

For each banknote B_k in $G \cup C$, for each possible threshold $s \in [0, 255]$, we computed the percentage $percB_{ks}$ of pixels above s in region T_k and the percentage $percD_{ks}$ of pixels below s in regions F_k . Using those values we computed optimum $percB^*$ and $percD^*$ thresholds to separate genuine and counterfeit classes.

To classify the banknote value, during training phase the software determines the discriminative patterns P_i for each possible banknote value $D_i \in \{5,10,20,50,100,200,500\}$, such that the intra-class distance is minimized whereas the inter-class distance is maximized. This process is very tricky for some banknote values which have similar patterns (e.g., 5 Euro and 20 Euro). The output of this process is a set of patches, with the corresponding position, related to the different banknote values. To take into account slightly translated and/or rotated images, the search area of each patch has the same shape and center coordinates of the patch itself, but it is wider: approximately the search area is 2.5 times the corresponding patch area.

Once the training phase is performed, the selected banknote features (patches, locations and thresholds learned) are stored in the flash memory and used during usage stage (see Figure 47) to infer validity and value of input banknote images.

12.3. Banknote authentication

In the pseudo code related to the authentication and classification is shown.

Pseudo-code

```
img ← captureBanknote();
imgNorm ← brightnessCorrection(img, brightnessMap);
genuineness ← isGenuine(imgNorm, locations, thresholds);
denomination ← identifyFaceValue(imgNorm, locations, patterns);
```

The image is first read from the IR sensor (*captureBanknote()*). Once the image has been corrected for the non-uniform led illumination (see Section 12.1), a proper threshold needs

to be found according to the actual input data. In our experiments we have noticed that the average grey value (indicated as *MeanRef*) of the blue region shown in Figure 48 can be robustly used for this purpose.

To check the genuineness the percentage of pixels inside the green region with a grey value below *MeanRef(percLT)* is calculated, together with the percentage of pixels inside the magenta and red regions with a grey value above *MeanRef(percGT)*.

The banknote is classified as genuine if $((percLT > percD^*) AND (percGT < percB^*))$.

To identify the face value, the training block provides us with a set of patches (templates related to the different banknotes values) and the corresponding search areas coordinates. Search areas are wider than the patches in order to be robust to a small misalignment.

Each template patch is placed at the center of its search area and a correlation measure between the pattern itself and the corresponding pixels on the search area is calculated. The procedure then search if a translation around the neighborhood of the current position could increase the correlation. This is performed by moving the pattern position. This step is then repeated until the correlation reaches a local maximum. The pattern with the highest correlation determines the final value to be assigned to the input banknote. Information about genuineness and face value are then sent to the display, in order to inform the user about the results of the banknote analysis.

The overall computational speed is of about 1 second for validity and 2 seconds for value recognition using the microprocessor described in the Section 13.

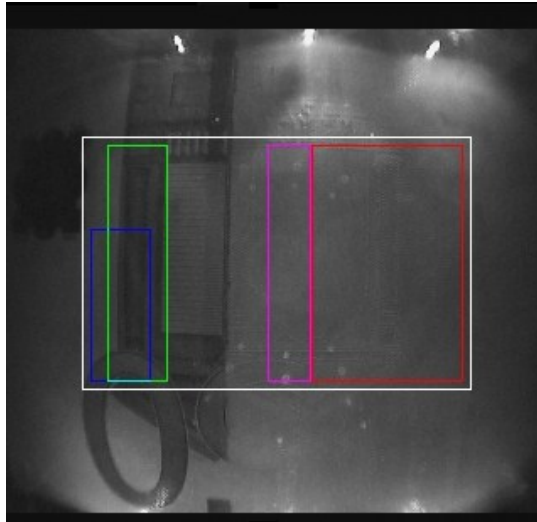


Figure 48: A banknote and the set of regions used to test its genuineness. White: ROI. Blue: the threshold to binarize the image is calculated in this area. Green: when seen under infrared light, this area must be dark. Red: when seen under infrared light, this area must be bright and without noticeable patterns. Magenta: additional area used to check the genuineness of some denominations.

Should be noted that the proposed software has been designed with the aim to work on the low cost hardware and with the low computational resources (e.g., 32Kbyte of memory) described in the following Section.

13. HW prototype

The hardware prototype has been designed to demonstrate the effectiveness of the proposed framework.

The overall system is summarized in the block diagram reported in Figure 49.

The Infrared LEDs (IR leds block) illuminate the scene. It is composed by 6 LEDs placed around the proscenium (the area where the banknote is placed). The illumination is not uniform in the real system. Moreover, it can vary from prototype to prototype. Hence, a calibration is needed to take into account the non-uniformity and the performances decay in the system life (see Section 12.1). An optical filter is inserted to avoid the external light source to influence the image acquisition system. It is placed on top of the infrared camera (IR camera block) that acquires the image. It is basically a common camera with CCTV output. Since the output is analog, an analog to digital converter (A/D converter block) is used to obtain a digital standardized format (i.e., CCIR-656).

The digital image is acquired by the microprocessor. In the prototype we have used an ATMEL AT91SAM9XE256, containing an ARM926EJ-STM processor, 200MHz with 256KB internal high-speed flash memory in which is stored the program. It contains also an Image Sensor Interface (ISI) port able to capture video sequences compliant with the standard ITU-R BT 601/656. The program contains, beside the control logic for the entire subsystems (e.g. IR led, IR camera settings, Display, etc.), also the related algorithms (for both validation and classification) described in the Section 12. The prototype has been also equipped with external SRAM memory, since the microprocessor contains only 32KB of internal SRAM.

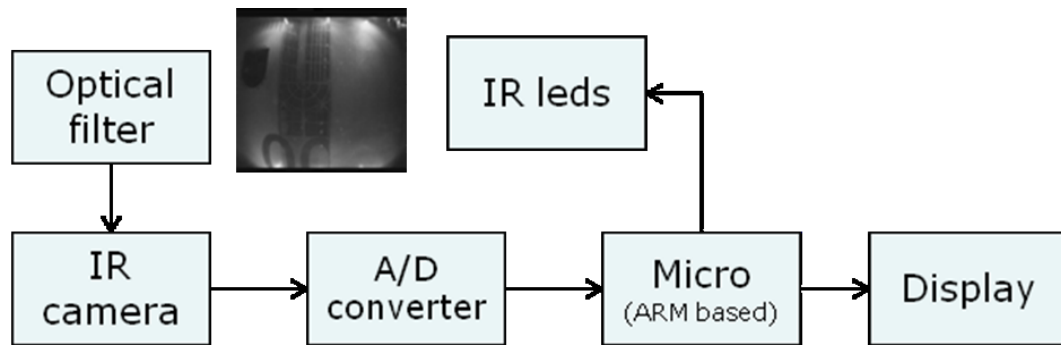


Figure 49: The block based schema of the hardware prototype

In Figure 50 the HW prototype is shown. In the center there is the ATMEL microprocessor; on the right side there is the A/D converter, in the upper part there is the external SRAM memory, in the lower/right side there is the power supply.

The display is also shown in Figure 51. It is composed by several 8-segment displays: 3 are used to show the banknote size, while 8 are used to show the validity and other service messages.

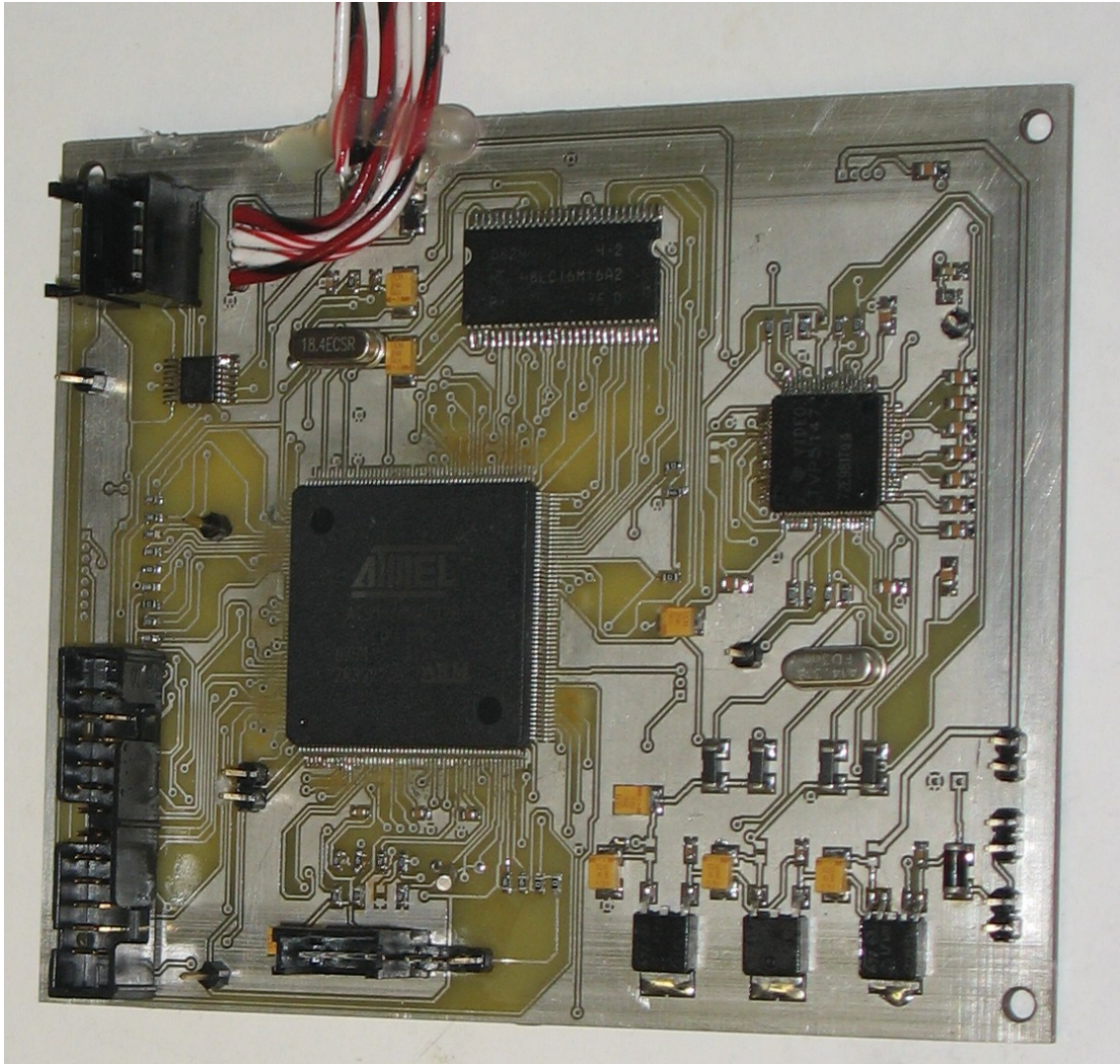


Figure 50: The main board containing the microprocessor (in the center); the A/D converter on the right side; the external SRAM memory in the upper part and the power supply in the lower/right side

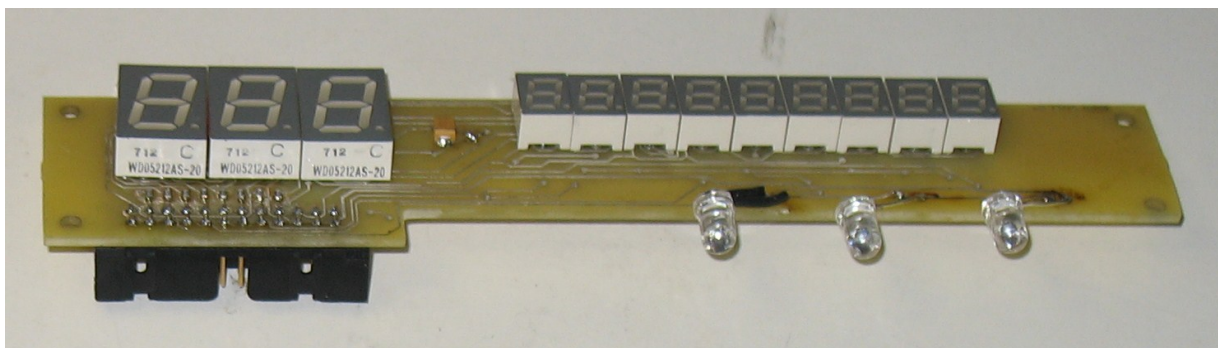


Figure 51: The display board containing 8-segments displays. Also three IR LEDs are visible.

14. Experimental results

To evaluate performances of the proposed technique we used our prototype to acquire a test set of 1750 banknote images, with the same criteria used for the training dataset (i.e. both counterfeit and genuine banknotes have been acquired under several environment lighting conditions, with different illuminants and brightness). In both cases, training and experimental phases, the dataset has been provided by an Italian Bank.

Acquired counterfeit banknotes include also specimens carefully calibrated to mislead digital counterfeit detectors. To deal with special cases (i.e., fake banknotes provided by the bank), additional procedure has been included in our software. The overall processing time is very close to the base algorithm, since it includes a few average computations on very small areas. Table 2 reports the results of the validity assessment.

Table 2: Genuine/Counterfeit classification

True positive (counterfeit banknote correctly classified)	True negative (genuine banknotes correctly classified)
100%	95.7%
False positive	False negative
0%	4.3%

Table 3 shows the output of the banknote value classification.

Most of the false negatives are observed for images acquired under direct sunlight illumination and for worn banknotes. Wear and tear can be identified as the main cause of misclassified samples for 5 and 10 euros banknotes, since those banknotes are widely used in everyday life.

Table 3: Banknote value classification.

	5E	10E	20E	50E	100E	200E	500E	Counterfeit
5E	88,00%	0,00%	0,00%	0%	0,00%	0,00%	0,00%	12,00%
10E	1,00%	91,00%	0,00%	0,00%	0,00%	0,00%	0,00%	8%
20E	0,00%	0,00%	98,00%	0,00%	0,00%	0,00%	0,00%	2%
50E	0,00%	0,00%	0,00%	99,00%	0,00%	0,00%	0,00%	1,00%
100E	0,00%	0,00%	0,00%	0,00%	93,00%	0,00%	0,00%	7,00%
200E	0,00%	0,00%	0,00%	0,00%	0,00%	98,00%	1,00%	1,00%
500E	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	95,00%	5,00%

15. Conclusions and future work

A complete informative system has been developed to obtain a fake euro banknote recognition system. The value of the banknote is recognized too. It is composed by both hardware and software parts. The software part is performed in three steps: a calibration phase, executed only once, in order to calibrate all the system to handle the sample variations; the training phase, performed offline in order to retrieve the model of each banknote; the real time phase composed by the steps of compensation, validity check and value retrieval.

The results are enough for the customer, since the false banknotes are all detected correctly. Anyway the system can be enhanced in several ways, especially in the false alarm rate, i.e. the detection of a false in true banknotes and in the value verification step, in order to avoid false values. Several algorithms exist in literature related to pattern matching, and some of them could produce better results. The problem is to find methodologies that are more robust of the current used but with comparative complexity. The paper "Counterfeit Detection and Value Recognition of Euro Banknotes" has been accepted in 8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP 2013).

References

- [1] S. Battiato, G. Messina, R. Rizzo - Image Forensics - Contraffazione Digitale e Identificazione della Camera di Acquisizione: Status e Prospettive - Chapter in IISFA Memberbook 2009 DIGITAL FORENSICS
- [2] W. Li, N.i Yu and Y. Yuan, "Doctored JPEG image detection" In Proceedings of International Conference on Multimedia and Expo, IEEE ICME (2008)
- [3] W.N. Lie, G.S. Lin and S.L. Cheng, "Dual protection JPEG images based on informed embedding and two-stage watermark extraction techniques" In IEEE Transactions on Information Forensics and Security 1 (2006), Pag. 330-341
- [4] Fan, Z. and De Queiroz, R.L. "Identification of bitmap compression history: JPEG detection and quantizer estimation" IEEE Transactions on Image Processing (2003), Pag. 230-235
- [5] Huang, F., Huang, J. and Shi, Y.Q. "Detecting double JPEG compression with the same quantization matrix" In IEEE Transactions on Information Forensics and Security, 5 (4), art. no. 5560817, Pag. 848-856, December 2010
- [6] H. Farid, "Digital Image Ballistics from JPEG Quantization", Technical Report, TR2006-583, Dartmouth College, Computer Science, 2006
- [7] H. Farid, "Digital image ballistics from JPEG quantization: A followup study," Department of Computer Science, Dartmouth College, Tech. Rep. TR2008-638, 2008.
- [8] S. Battiato, M. Mancuso - An Introduction to the Digital Still Camera Technology - ST Journal of System Research - Special Issue on Image Processing for Digital Still Camera, Vol. 2, No.2, December 2001
- [9] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," IEEE Trans. Computers, 90-93, Jan. 1974
- [10] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications (Academic Press, Boston, 1990)
- [11] http://en.wikipedia.org/wiki/Discrete_cosine_transform
- [12] CCITT-Recommendation T.81: "information technology - Digital compression and coding of continuous tone still images" - Requirements and guidelines (1992)
- [13] S. Battiato, A. Bruna, A. Buemi, A. Castorina – "Analysis and Characterization of JPEG 2000 Standard for Imaging Devices" – IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, November 2003, Page(s): 773- 779
- [14] A. Bruna, M. Mancuso – "JPEG Compression Factor Control: A New Algorithm" - IEEE Proceedings of the International Conference on Computer Electronics (ICCE2001 pp.206-207), Los Angeles, June 2001

- [15] A. Bruna, M.Mancuso, A.Capra. S.Curti – “Very Fast algorithm for Jpeg Compression Factor Control” - Proceedings of SPIE Electronic Imaging 2002 - Sensors, Cameras, and Applications for Digital Photography IV - 20-25 January 2002, San José CA USA
- [16] A. Bruna, S. Smith, F. Vella - "JPEG Rate Control Algorithm for multimedia", IEEE-ISCE, 1-3 September 2004, Reading, United Kingdom
- [17] M. Farina, A. Bruna - "Evolutionary Multiobjective Optimization of Quantization in JPEG Image Compression" - Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO2007)
- [18] S. Battiato, C. Bosco, A. Bruna, G. Di Blasi, G. Gallo – “Statistical Modeling of Huffman Tables Coding”, Springer-Verlag GmbH, Lecture Notes in Computer Science - Volume 3617 / 2005, ISBN: 3-540-28869-4
- [19] Judith A. Redi, Wiem Taktak and Jean-Luc Dugelay, “Digital image forensics: a booklet for beginners” In Multimedia Tools and Applications - Volume 51 no. 1, pp. 133–162, January 2011
- [20] S. Battiato, M. Mancuso, A. Bosco and M. Guarnera - “Psychovisual and Statistical Optimization of Quantization Tables for DCT Compression Engines” In IEEE Proceedings of International Conference on Image Analysis and Processing ICIAP 2001, Pag. 602-606 Palermo, Italy, September 2001
- [21] Li, W., Yuan, Y and Yu, N. “Passive detection of doctored JPEG image via block artefact grid extraction” In Signal Processing (2009), 89 (9), pp. 1821-1829
- [22] <http://www.jpeg.org/jpeg/index.html>
- [23] <http://www.mpeg.org/MPEG/video/>
- [24] S. Battiato and G. Messina “Digital Forgery Estimation into DCT Domain - A Critical Analysis” In Proceedings of ACM Multimedia 2009 – Workshop Multimedia in Forensics (MiFor 2009) - Beijing (China), October 2009, pp. 37–42
- [25] W. Li, Y. Yuan and N. Yu “Detecting copy-paste forgery of JPEG image via block artifact grid extraction” In Proceedings of International Workshop on Local and Non-Local Approximation in Image Processing (IWIP-08), Lausanne, Switzerland, 2008
- [26] cjpeg code can be found in <http://www.ijg.org/>
- [27] A. Bruna, M.Mancuso, A.Capra. S.Curti, “Very Fast algorithm for Jpeg Compression Factor Control” In Proceedings of SPIE Electronic Imaging 2002 - Sensors, Cameras, and Applications for Digital Photography IV - San José CA USA - Jan 2002
- [28] DBForgery 1.0:
http://iplab.dmi.unict.it/index.php?option=com_content&task=view&id=41&Itemid=118
- [29] A. Bruna, G. Messina, and S. Battiato, “Crop detection through blocking artefacts analysis,” in In Proceedings of International Conference on Image Analysis and Processing - ICIAP 2011, ser. Lecture Notes in Computer Science, vol. 6978, 2011, pp. 650–659.

- [30] S. Battiato, G. M. Farinella, E. Messina, and G. Puglisi, "Robust image alignment for tampering detection," *IEEE Transactions on Information Forensics & Security*, vol. 7, no. 4, 2012.
- [31] A. Popescu and H. Farid, "Statistical tools for digital forensics," in *Proceeding of the 6th International Workshop on Information Hiding*, Toronto, Canada, 2004.
- [32] J. Lukas and J. Fridrich, "Estimation of primary quantization matrix in double compressed jpeg images," in *in Proceedings of Digital Forensic Research Workshop (DFRWS)*, 2003.
- [33] —, "Exposing digital forgeries from JPEG ghosts," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, pp. 154–160, 2009.
- [34] G. K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [35] E. Kee, M. K. Johnson, and H. Farid, "Digital image authentication from JPEG headers," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1066–1075, 2011.
- [36] "Dataset Eastman Kodak Company: PhotoCD PCD0992, <http://r0k.us/graphics/kodak/>."
- [37] Hinwood, A., Preston, P., Suaning, G., and Lovell, N. (2006). Bank note recognition for the vision impaired. *Australian Physical and Engineering Sciences in Medicine*, 29(2)
- [38] Choi, E., Lee, J., and Yoon, J. (2006). Feature extraction for bank note classification using wavelet transform. In *International Conference on Pattern Recognition*.
- [39] He, C., Girolami, M., and Ross, G. (2004). Employing optimized combinations of one-class classifiers for automated currency validation. *Pattern Recognition*, 37(6).
- [40] Takeda, F., Nishikage, T., and Omatu, S. (1999). Banknote recognition by means of optimized masks, neural networks and genetic algorithms. *Engineering Applications of Artificial Intelligence*, 12(2).
- [41] Takeda, F., Sakoobunthu, L., and Satou, H. (2003). Thai banknote recognition using neural network and continues learning by dsp unit. In Palade, V., Howlett, R., and Jain, L., editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 2773 of *Lecture Notes in Computer Science*. Springer Berlin - Heidelberg.
- [42] Khashman, A. and Sekeroglu, B. (2005). Multi-banknote identification using a single neural network. In Blanc-Talon, J., Philips, W., Popescu, D., and Scheunders, P., editors, *Advanced Concepts for Intelligent Vision Systems*, volume 3708 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg.
- [43] A. R. Bruna, G. Messina, S. Battiato, "Crop Detection Through Blocking Artefacts Analysis" - *International Conference on Image Analysis and Processing (ICIAP) 2011*
- [44] A. Castorina, G. Spampinato, A. Bruna, A. Capra - "Adaptive color bleeding removal for video and still DCT compressed sequences" - *Proceedings of SPIE Electronic Imaging 2007 - Digital Photography III - 28-1 January 2007, San José CA USA*

- [45] G. Spampinato, A. Castorina, A. Bruna, A. Capra - "JPEG Adaptive Chromatic Post-Processing" - Computational Color Imaging Workshop 2007 (CCIW'07), Modena, September 14, 2007
- [46] G. Spampinato, A. Castorina, A. Bruna, A. Capra - "Chromatic Enhancement Technique for JPEG Image" - 14th International Conference on Systems, Signals and Image Processing (IWSSIP 2007)
- [47] A. Bosco, R. A. Bruna, D. Giacalone, S. Battiato, R. Rizzo – “Signal dependent raw image denoising using sensor noise characterization via multiple acquisitions” - IS&T/SPIE Electronic Imaging 2010 – San Jose (CA)
- [48] A. Bosco, S. Battiato, A. Bruna, R. Rizzo - "Noise Reduction for CFA Image Sensors Exploiting HVS Behaviour" - MDPI Sensors Journal, Sensors 2009, Pages 1692-1713
- [49] A. Bosco, S. Battiato, A. Bruna, R. Rizzo - "Texture Sensitive Denoising for Single Sensor Color Imaging Devices" - Computational Color Imaging (CCIW) 2009. Pages 130-139