Advances on Multimedia Forensics

Compression-based traces

Prof. Alessandro Piva
Dept. of Information Engineering
University of Florence (Italy)
alessandro.piva@unifi.it

Coding-based techniques

- Forensic techniques that detect tampering in compressed images
- They explicitly leverage statistical correlations introduced by the JPEG lossy compression scheme.
- Consecutive applications of JPEG introduce different fingerprint with respect to a single compression.

JPEG Image Compression Standard

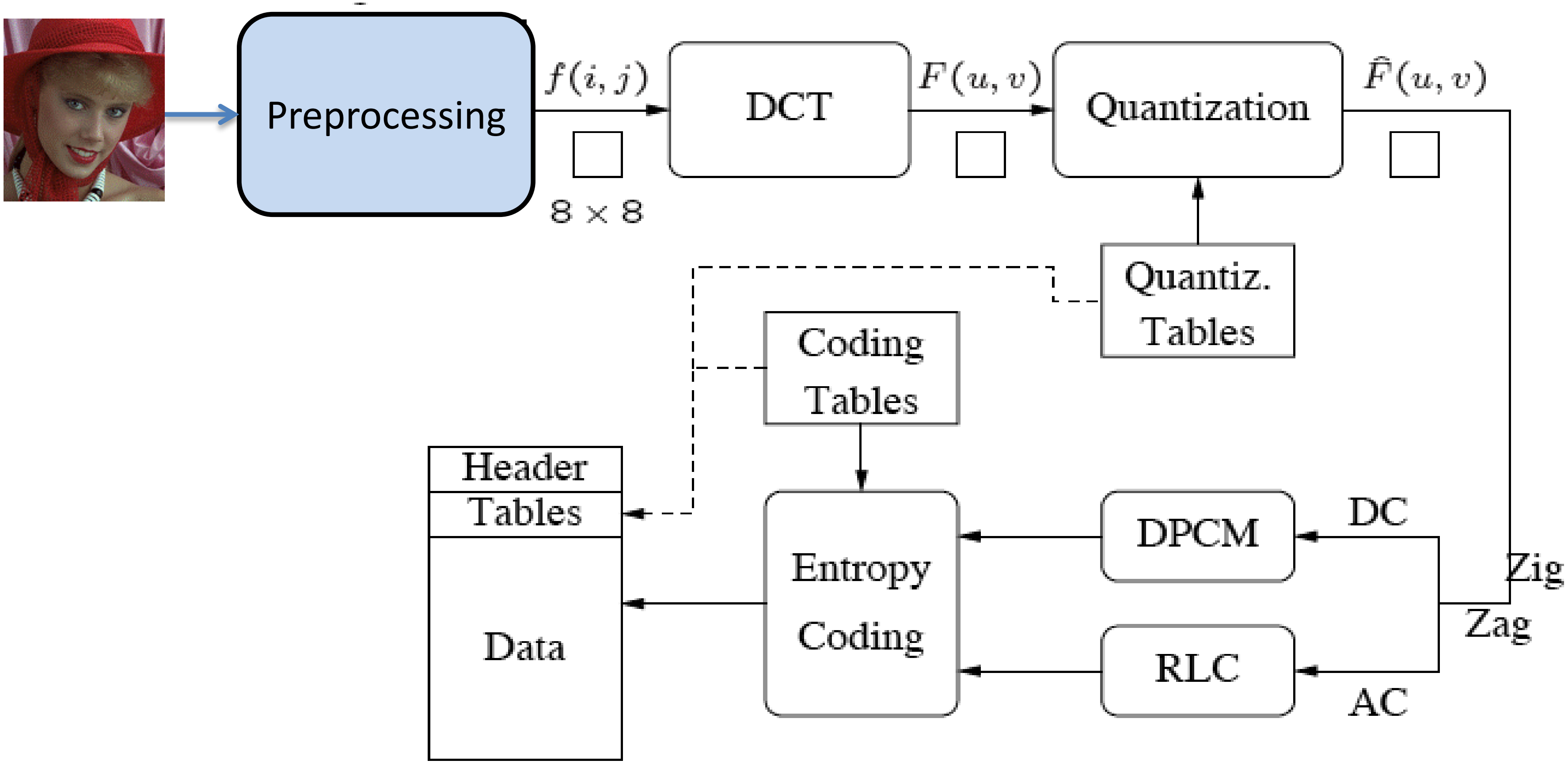
- JPEG - Joint Photographic Experts Group
  - a joint ISO/CCITT committee, organized in 1986, worked toward establishing the first international digital image compression standard for continuous-tone (multilevel) still images, both grayscale and color.
  - Became an international standard in 1992
- Allow for lossy and lossless encoding
  - Part-1 DCT-based lossy compression
    - average compression ratio 15:1
  - Part-2 Predictive-based lossless compression

JPEG Baseline System

- A JPEG compliant decoder has to support a minimum set of requirements, the implementation of which is referred to as ***baseline implementation.***

- It is lossy
- It must be able to decompress image using sequential DCT-based mode.
- For baseline compression the bit depth must be B=8. ; as a more general situation, the image samples are assumed to be unsigned quantities in the range [0, 2^{B-1}].
- (Additional features are supported in the extended implementation of the standard).

JPEG Baseline encoder



JPEG Baseline - Preprocessing

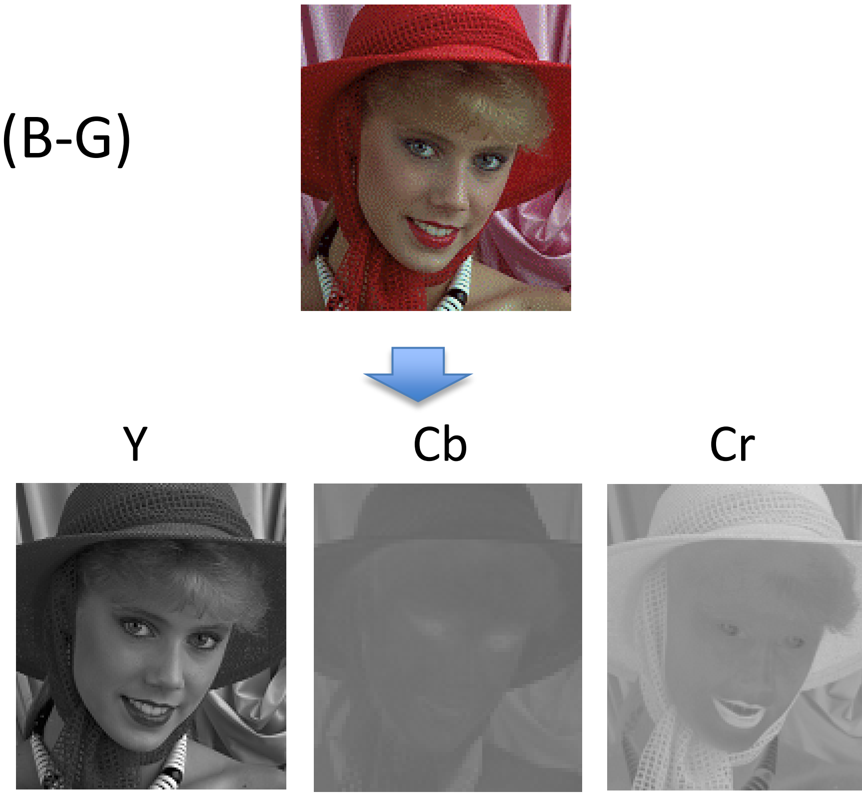
Color Space conversion

- The uncompressed image is usually stored in 24 bit/pixel RGB -- that is, 8 bits each of Red, Green and Blue.
- There is usually a clear visual correlation between R, G and B subchannels the 3 pictures.
- To achieve better compression ratios, it is common to decorrelate the RGB fields into separate luminance (Y) and chrominance (Cb, Cr) components .

JPEG Baseline - Preprocessing

Color Space conversion

- Y=0.299(R-G) + G + 0.114(B-G)
- Cb=0.564(B-Y)
- Cr=0.713(R-Y)



- An unfiltered image with subpixels arranged as {Y,Cb,Cr,Y,Cb,Cr,Y...) is called 4:4:4 format, since there are 4 Y's for every 4 Cb's and 4 Cr's:
- e.g. for a 720x480 pixel image, 4:4:4 format implies that each of the 3 components is 720x480 bytes.

JPEG Baseline - Preprocessing

Color Spatial Downsampling

- Human eye is more sensitive to luminance than chrominance: size can be reduced by subsampling the (Cb,Cr) fields.

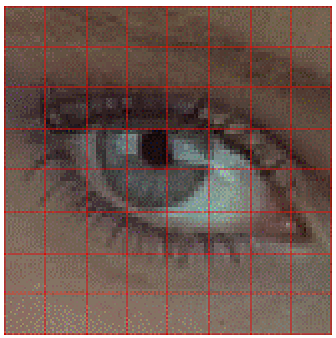
Y Cb Cr

JPEG Baseline - Preprocessing

Level offset

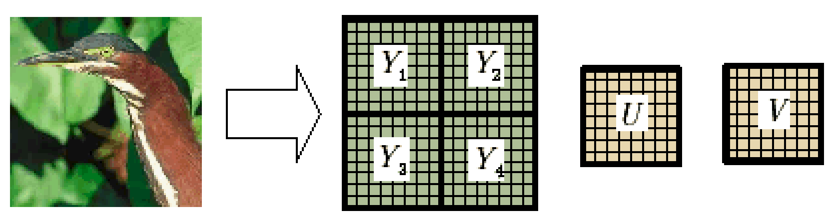
- The image samples are assumed to be unsigned quantities in the range [0, 2^{B-1}]. The level offset subtract $2^{\wedge B-1}$ from every sample value
- Produce signed quantities in the range [-2^{B-1}, $2^{B-1}-1$].
- The purpose of this is to ensure that all the DCT coefficients will be signed quantities with a similar dynamic range.

JPEG Baseline - Preprocessing

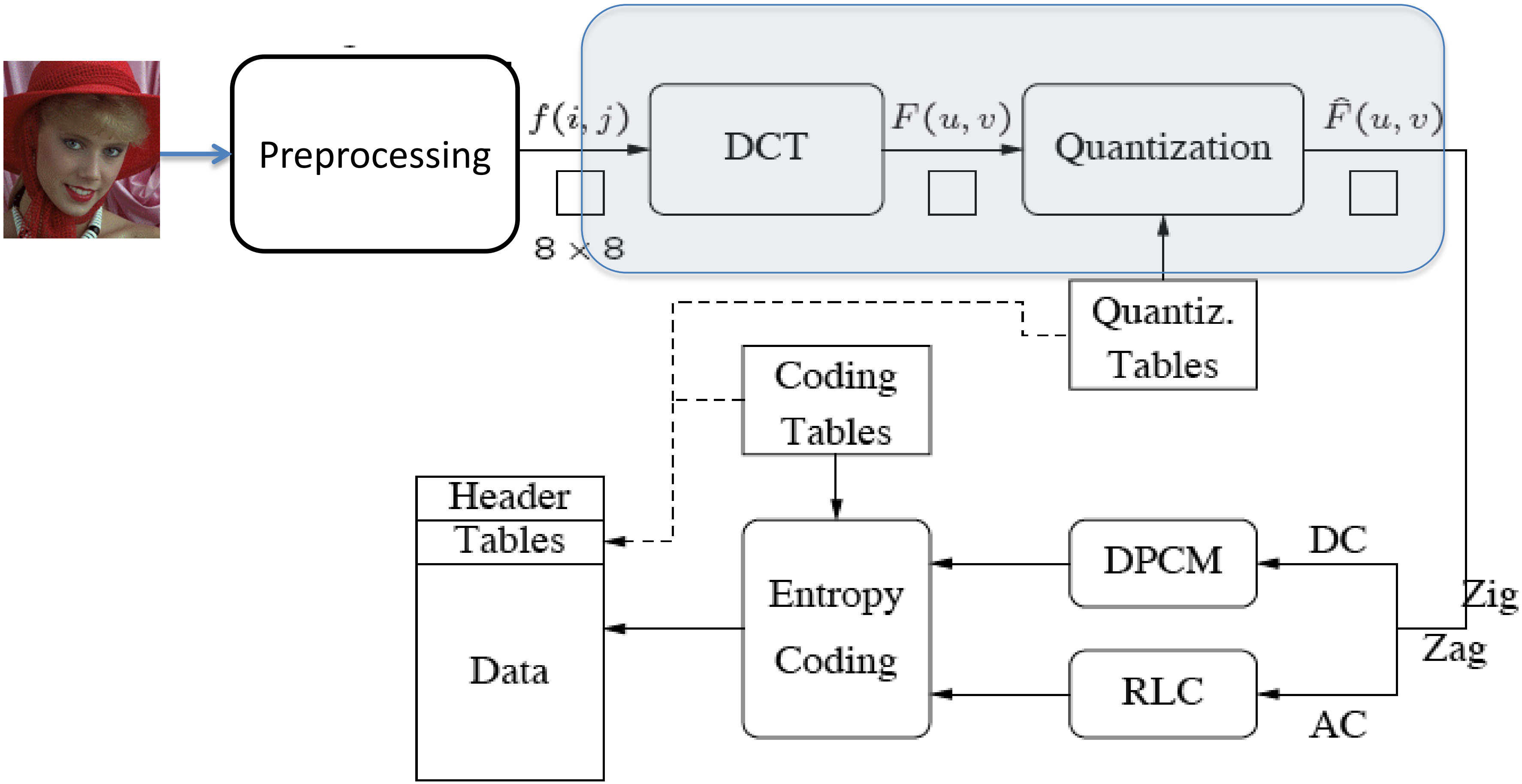


Partitioning

- In each image buffer, the data is partitioned into 8x8 blocks, from left to right and top to bottom.
- These blocks do not overlap, and if the image dimensions are not multiples of 8, the last row and/or column of the image is duplicated as needed.
- Minimum Coded Unit (MCU) contains four Y 8x8 blocks followed by one Cb 8x8 block and one Cr 8x8 block:



JPEG Baseline encoder



JPEG Baseline - DCT

- 2D DCT is then performed on each 8x8 block
  - The DCT is performed independently for each block: this is why, when a high degree of compression is requested, JPEG gives a "blocky" image result

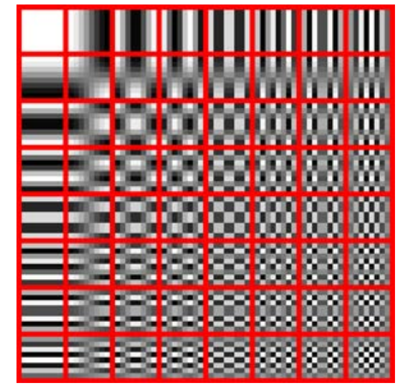
$$F(u,v) = \frac{C(u)C(v)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) cos\left(\frac{2(x+1)u\pi}{16} \right) cos\left(\frac{2(y+1)v\pi}{16} \right)$$

  - f(x,y): 2-D sample value,
  - F(u,v): 2-D DCT coefficient

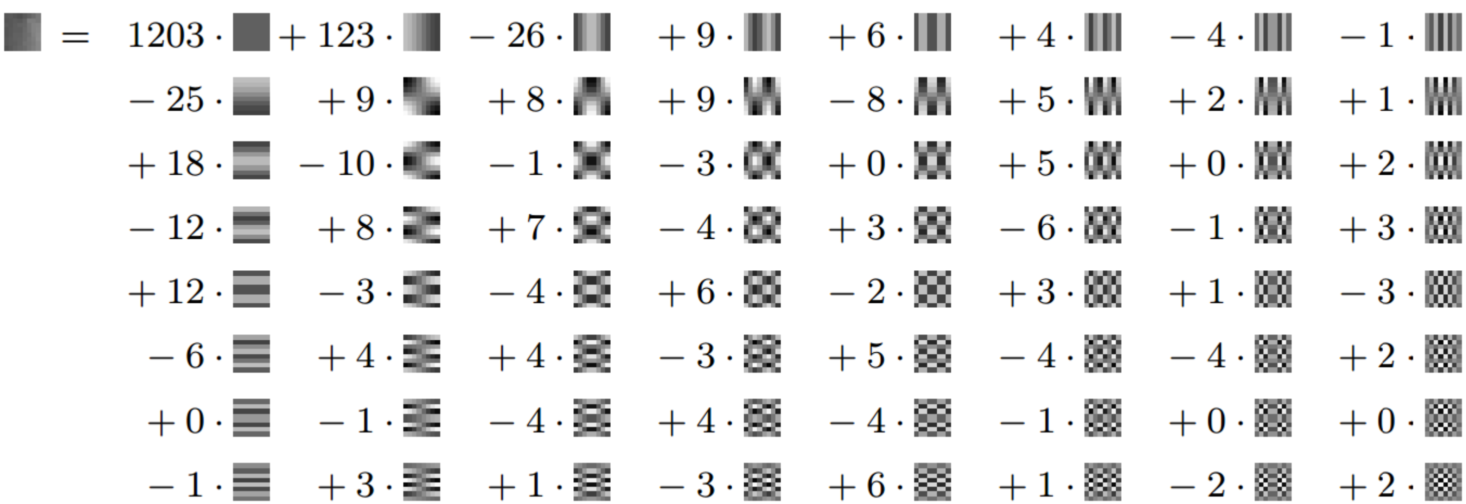
$$C(x) = \begin{cases} 1/\sqrt{2} & x = 0 \\ 1 & otherwise \end{cases}$$

the DCT was chosen because of its decorrelation features, image independence, efficiency of compacting image energy, and orthogonality (which makes the inverse DCT very straightforward).

JPEG Baseline - DCT

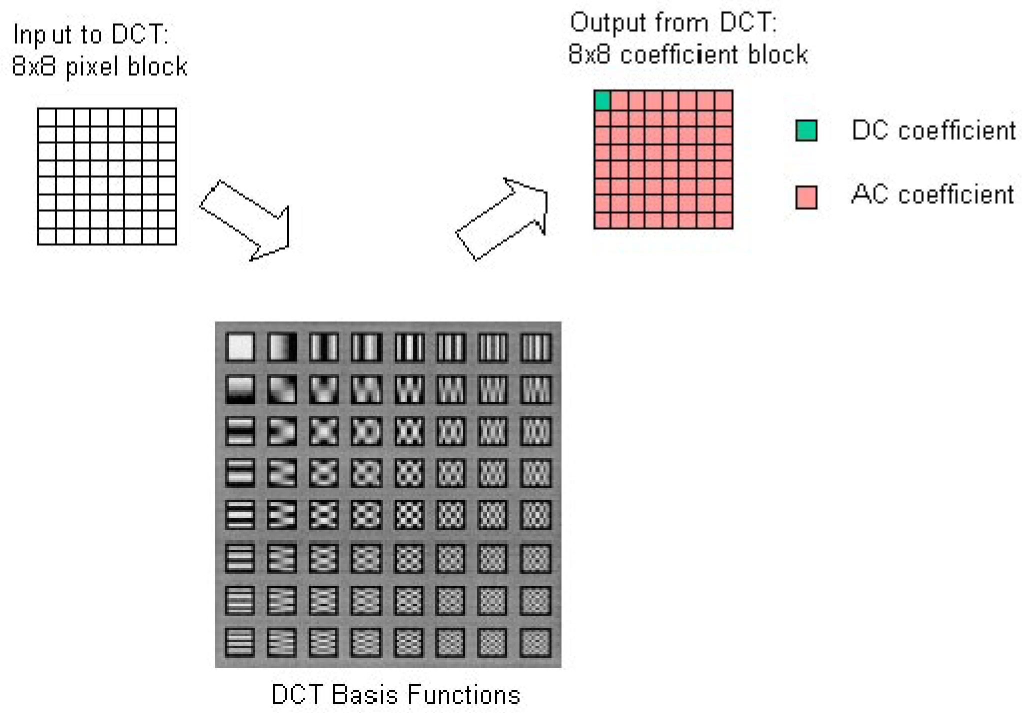


- FDCT takes the 8x8 image block, i.e. a 64-point discrete signal, and decomposes it into 64 orthogonal basis signals.
- ouput of FDCT is the set of 64 "DCT coefficients" whose values can be regarded as the amount of 2D spatial frequencies contained in the input:



JPEG Baseline - DCT

- In performing a DCT on an 8x8 image block, we are correlating the block with each of the 64 DCT basis functions and recording the relative strength of correlation as coefficients in the output DCT matrix.
- For example, the coefficient in the output DCT matrix at (2,1) corresponds to the strength of the correlation between the basis function at (2,1) and the entire 8x8 block.



JPEG Baseline - DCT

- The coefficients corresponding to high-frequency details are located to the right and bottom of the DCT block, and usually have zero or near-zero amplitude.
- it is precisely these weights which we try to nullify with Quantization -- the more zeroes in the block, the higher the compression that is achieved.

Image pixels

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

DCT coefficients

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

JPEG Baseline - DCT

- Example of block with low activity

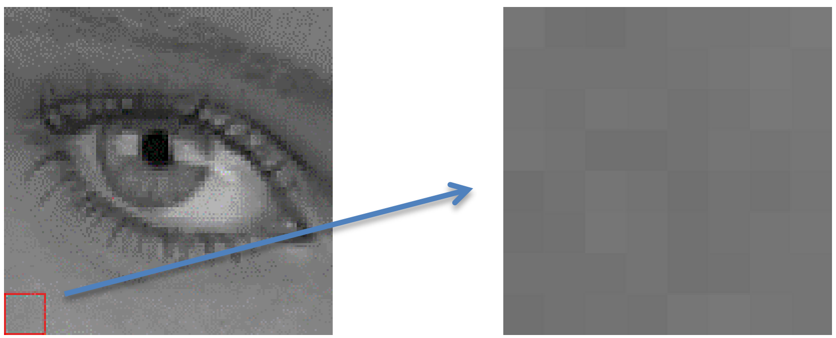


Image pixels

925.5	-11.1	4.4	-1.8	-1.0	1.9	1.4	-0.3
4.3	-1.3	4.4	2.5	2.6	1.0	-0.7	0.6
4.0	-3.6	-0.7	3.5	1.7	-0.2	-1.6	1.0
-2.1	0.3	-0.1	-3.5	1.7	0.6	0.6	-0.6
-1.2	-0.6	0.2	3.3	1.3	-3.6	1.3	-0.8
-0.9	1.2	3.8	2.0	3.5	0.7	-0.9	-0.6
1.9	-1.0	-0.9	-1.3	0.1	0.9	1.4	1.2
-0.9	1.0	-0.8	-3.0	0.3	1.3	0.0	-0.4

(a)

DCT coefficients

JPEG Baseline - DCT

- Example of block with high activity

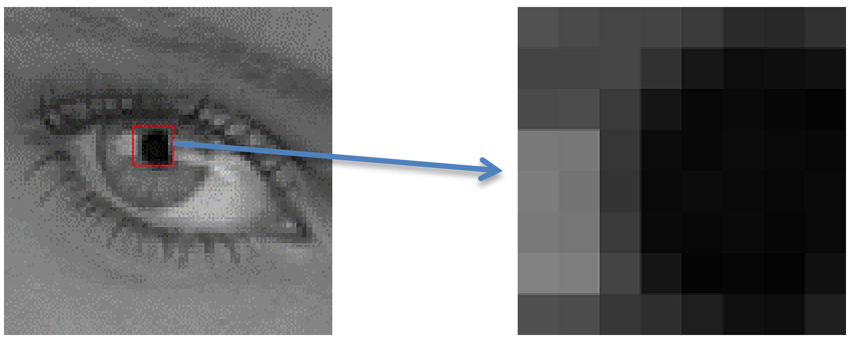


Image pixels

350.5	251.0	109.0	0.8	-17.7	-26.1	-6.7	-3.7
7.0	-47.2	-43.6	-11.3	4.5	15.1	5.1	5.1
30.8	-58.4	-52.0	-26.3	25.0	14.2	6.4	-1.7
35.7	14.0	17.1	12.4	0.3	-6.1	-3.0	1.4
26.0	-19.7	-3.5	10.7	13.7	-3.1	-5.1	-1.1
20.0	18.6	20.0	7.5	-5.7	-6.5	-2.5	-3.1
-6.0	-23.6	-12.4	2.4	1.6	0.5	1.7	2.2
-3.0	-1.8	0.3	0.0	1.3	1.2	1.9	1.2

(b)

DCT coefficients

JPEG Baseline - DCT

- From the 64 DCT coefficients it is possible to reconstruct a 64-point signal through the inverse DCT (or IDCT).

$$f(x,y)=\sum_{u=0}^{7}\sum_{v=0}^{7}\frac{C(u)C(v)}{4}F(u,v)cos\left(\frac{2(x+1)u\pi}{16}\right)cos\left(\frac{2(y+1)v\pi}{16}\right)$$

- If FDCT and IDCT could be computed with perfect accuracy and if the DCT coeffs were not quantized, the original signal could be exactly recovered.
- **In principle, the DCT introduces no loss to the source image samples**; it merely transforms them to a domain in which they can be more efficiently encoded.

JPEG Baseline - Quantization

- In lossy compression coefficients are mapped into a smaller set of possible values, thus discarding information which is not visually significant.
- The quantization unit performs this task of a many-to-one mapping of the DCT coefficients (is not reversible!)
  - possible outputs are limited in number, and are integers, no longer reals: we can use fewer bits to encode them
  - Many quantized DCT coefficients are zero, making them suitable for efficient coding.
- **principal source of lossiness in DCT-based encoders**

JPEG Baseline - Quantization

- Each of the 64 DCT coeffs is quantized with a corresponding element of a 8x8 Quantization Table **Q**, which must be specified by the application (or user) as input to the encoder.
- Same quantization matrix for all blocks of the image

$$F^{Q}(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

$F(u,v)$ original DCT coefficient
$F^{Q}(u,v)$ DCT coefficient after quantization
$Q(u,v)$ quantization value

JPEG Baseline - Quantization

- Q(u,v) can be any integer value from 1 to 255.
- Smaller Q(u,v) means a smaller step size and hence more precision and less compression
- Different quantization step sizes for different frequencies
  - larger entries in Q for higher frequencies
- Color more compressed than luminance.
- Standard tables chosen according to psychophysical studies:

The Luminance Quantization

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

The Chrominance Quantization

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

JPEG Baseline - Quantization

- Default quantization table
  - "Generic" over a variety of images
- Proprietary quantization tables
  - Many image processing tools (e.g. Photoshop) adopt custom Q tables
- scaled versions of "default" quantization table
  - Q(u,v) values multiplied with a scaling factor.

$$F^{Q}(u,v) = Round\left(\frac{F(u,v)}{Scale_{factor} * Q(u,v)}\right)$$

  - Usually the $Scale_{factor}$ is derived from a quality factor Q specified by the user.

A simple example

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

8-bit pixel values of the 8x8 image block

DCT

915.6	451.3	25.6	-12.6	16.1	-12.3	7.9	-7.3
216.8	19.8	-228.2	-25.7	23.0	-0.1	6.4	2.0
-2.0	-77.4	-23.8	102.9	45.2	-23.7	-4.4	-5.1
30.1	2.4	19.5	28.6	-51.1	-32.5	12.3	4.5
5.1	-22.1	-2.2	-1.9	-17.4	20.8	23.2	-14.5
-0.4	-0.8	7.5	6.2	-9.6	5.7	-9.5	-19.9
5.3	-5.3	-2.4	-2.4	-3.5	-2.1	10.0	11.0
0.9	0.7	-7.7	9.3	2.7	-5.4	-6.7	2.5

DCT values of the 8x8 image block

Quantization

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

Standard quantization table for luminance

57	41	2	0	0	0	0	0
18	1	-16	-1	0	0	0	0
0	-5	-1	4	1	0	0	0
2	0	0	0	-1	0	0	0
0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized DCT values using the quantization table on the left

A simple example

57	41	2	0	0	0	0	0
18	1	-16	-1	0	0	0	0
0	-5	-1	4	1	0	0	0
2	0	0	0	-1	0	0	0
0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized DCT values using the quantization table on the left

Dequantization & IDCT

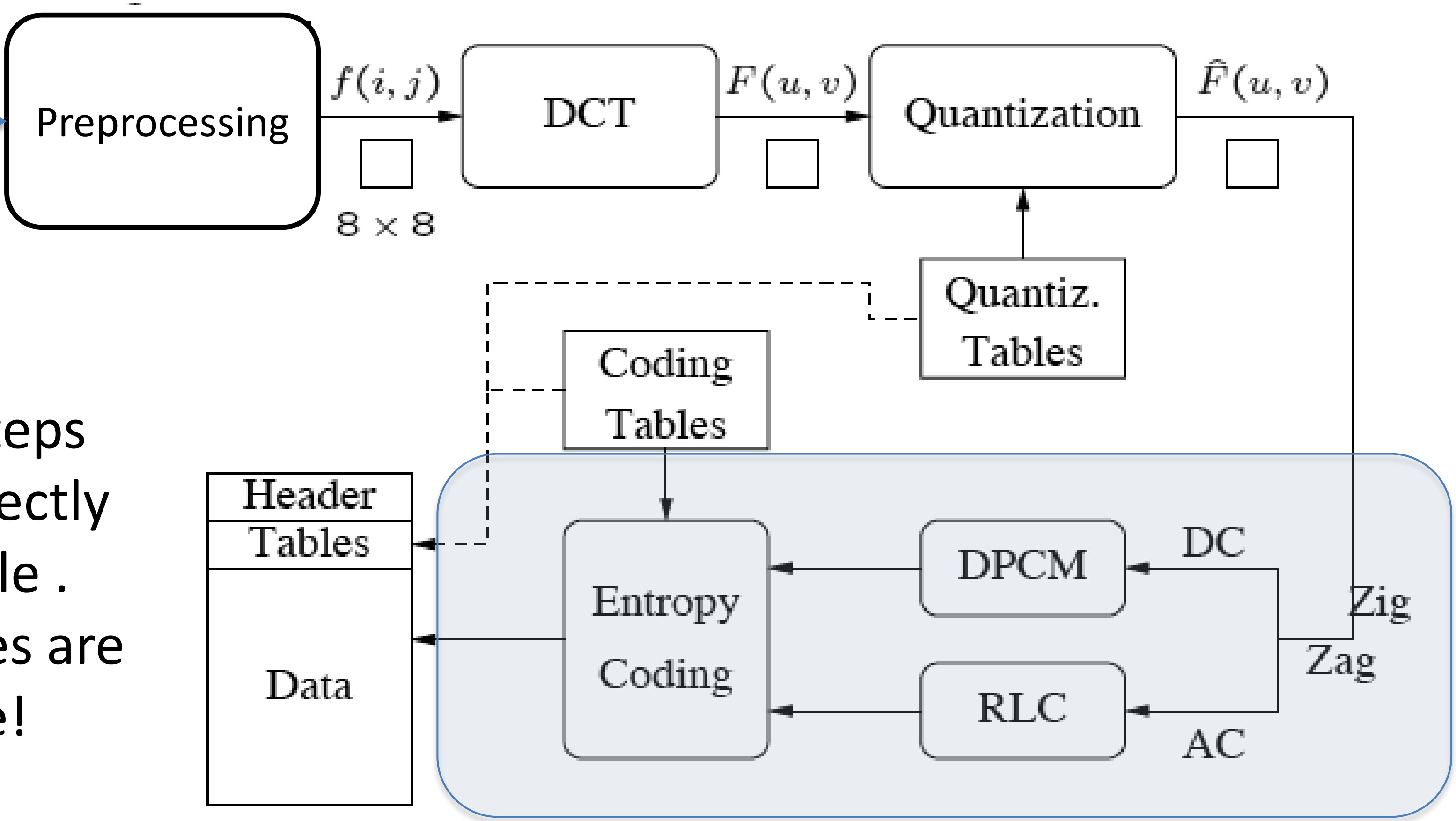
181	185	196	208	203	159	86	27
191	189	197	203	178	118	58	25
192	193	197	185	136	72	36	33
184	199	195	151	90	48	38	43
185	207	185	110	52	43	49	44
201	198	151	74	32	40	48	38
213	161	92	47	32	35	41	45
216	122	43	32	39	32	36	58

8x8 image block recovered after decoding. Note the difference with the original image block: the compression was lossy !

187	188	189	202	209	175	66	41
191	186	193	209	193	98	40	39
188	187	202	202	144	53	35	37
189	195	206	172	58	47	43	45
197	204	194	106	50	48	42	45
208	204	151	50	41	41	41	53
209	179	68	42	35	36	40	47
200	117	53	41	34	38	39	63

Original 8x8 image block

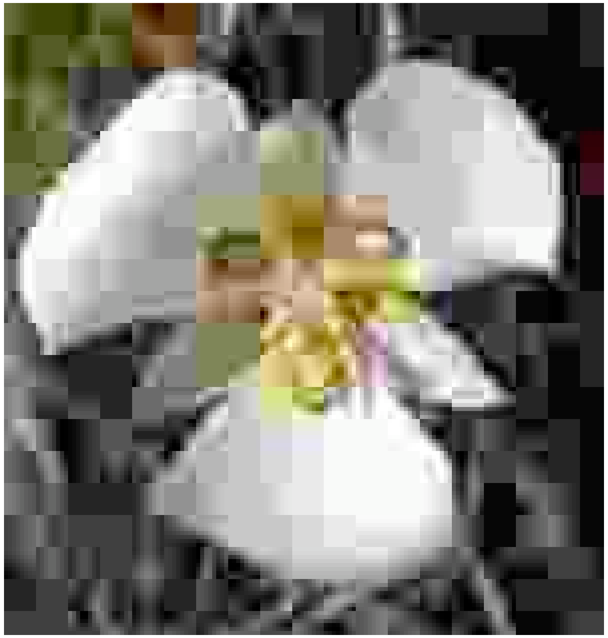
JPEG Baseline encoder



These steps are perfectly reversible .
No traces are left here!

Compression artifacts in spatial domain

- Baseline JPEG works with 8x8 image blocks, individually transformed and quantized;
- artifacts appear at the border of neighboring blocks in the form of horizontal and vertical edges.
- Even "light" compression may leave small but consistent discontinuities across block boundaries



- A manipulation can perturbs these blocking artifacts

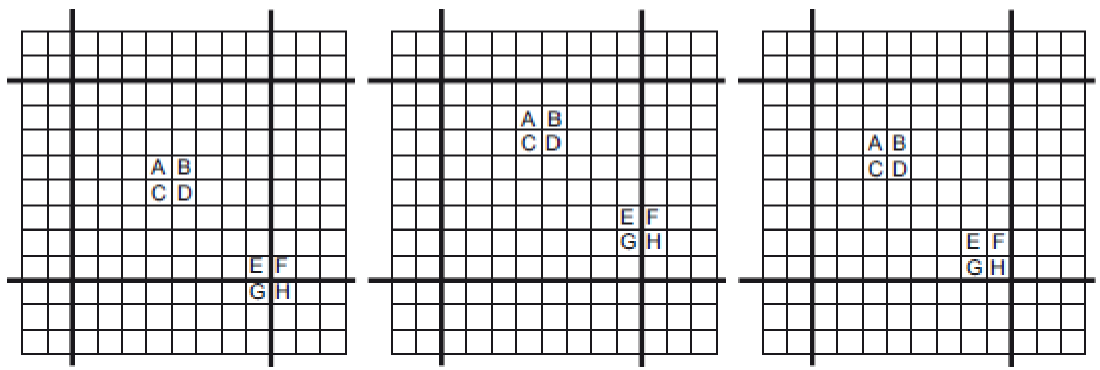
Compression artifacts in spatial domain

- Idea: if there is no compression the pixel differences across 8x8 blocks should be similar to those within blocks, whereas if the image is JPEG-compressed, the differences across blocks should be different due to blocking artifacts.
- assume the block grid is known.

•Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Trans. on Image Processing*, vol. 12, no. 2, Feb 2003.

Compression artifacts in spatial domain

- For each block, compute:
- Z'(x,y)=|A+D-B-C|, Z"(x,y)=|E+H-F-G|,
- where A ~ H are the values of the pixels, and the (x, y) is the coordinate of A in each block.
  - E.g. the coordinate of E: P(E) = P(A) + (4, 4).
  - Examples with (x, y) = (4, 4), (2, 4) and (3, 3) respectively:



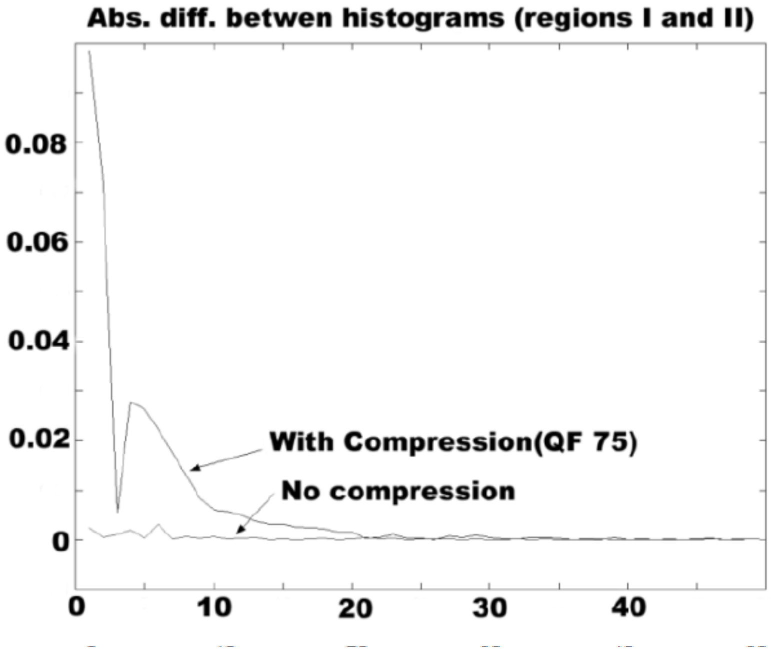
(a) $(x, y) = (4, 4)$ (b) $(x, y) = (2, 4)$ (c) $(x, y) = (3, 3)$

Compression artifacts in spatial domain

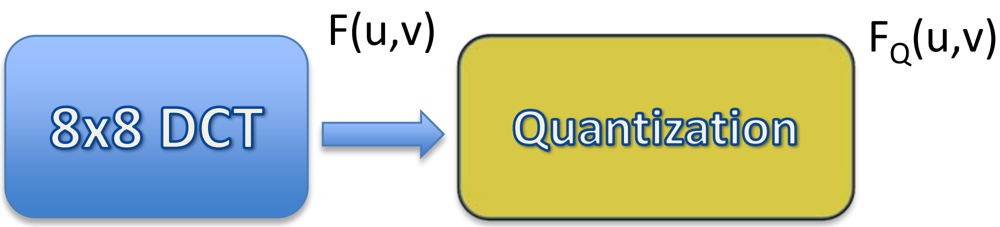
- compute the histograms H_I , H_{II} of $Z'_{(x,y)}$ and $Z''_{(x,y)}$, then energy K of the difference between H_I and H_{II} :

  $K_{(x,y)}(n)=| H_I(n)- H_{II}(n)|$
- where $n \in [0,255*2]$.

- differences are larger across a JPEG block boundary;
- biggest values always occur when $P(A) = (4, 4)$, and when $x = 4$ or ($y = 4$).
- When the coordinates of A to D and E to H are all inside a block, then the difference is small.
- K can be compared to a threshold or given as a confidence parameter.



Compression artifacts in frequency domain



$$F_Q(u,v) = Round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

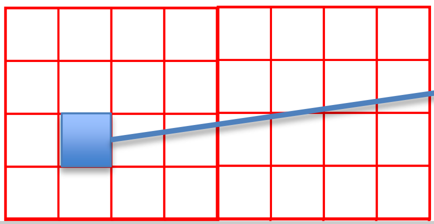
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

forces the value of each DCT coefficient to be an integer multiple of Q(u,v)

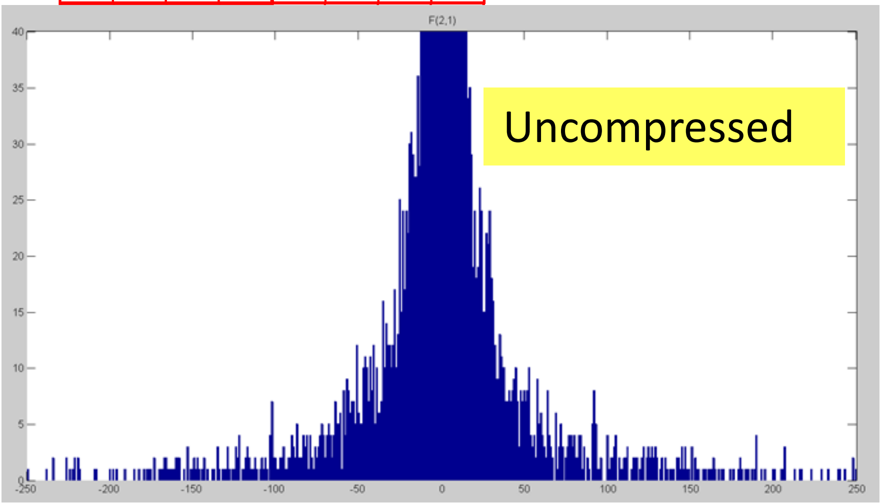
- Though the process of rounding and truncating the decompressed pixel values perturbs the DCT coefficients, their values typically remain clustered around integer multiples of Q(u,v).

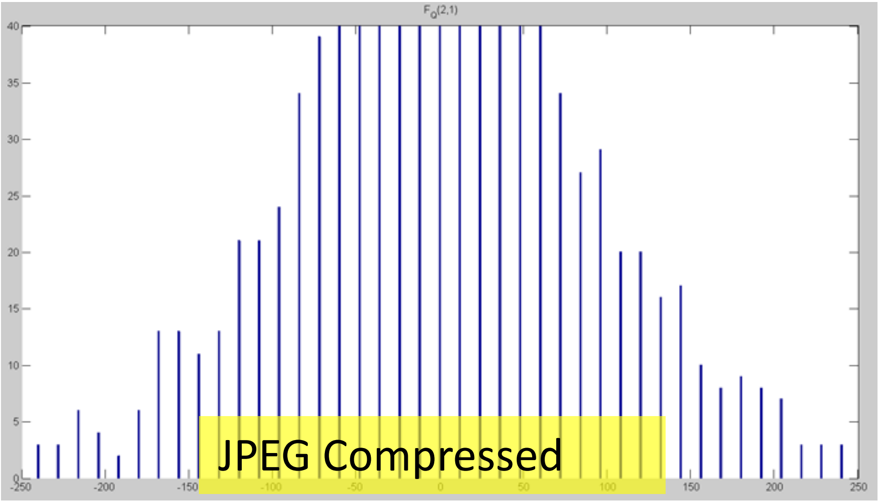
Single Compression

- It leaves traces we can find in the histogram computed by collecting from each 8x8 block the DCT coefficients having same frequency (u,v)



$$F_Q(2,1) = Round\left(\frac{F(2,1)}{12}\right)$$





The distortion of such a behaviour can be used to detect the presence of tampering.

Tampering of JPEG images

- Any digital manipulation requires that an image be loaded into a photo-editing software program and resaved.
- Since most images are stored in the JPEG format, it is likely that both the original and manipulated images are stored in this format.
- In this scenario, the manipulated image is compressed twice.

Double compression chain

- DCT grids of successive compressions aligned

A-DJPG

shift (0,0)

Grid of first compression

shift (0,0)

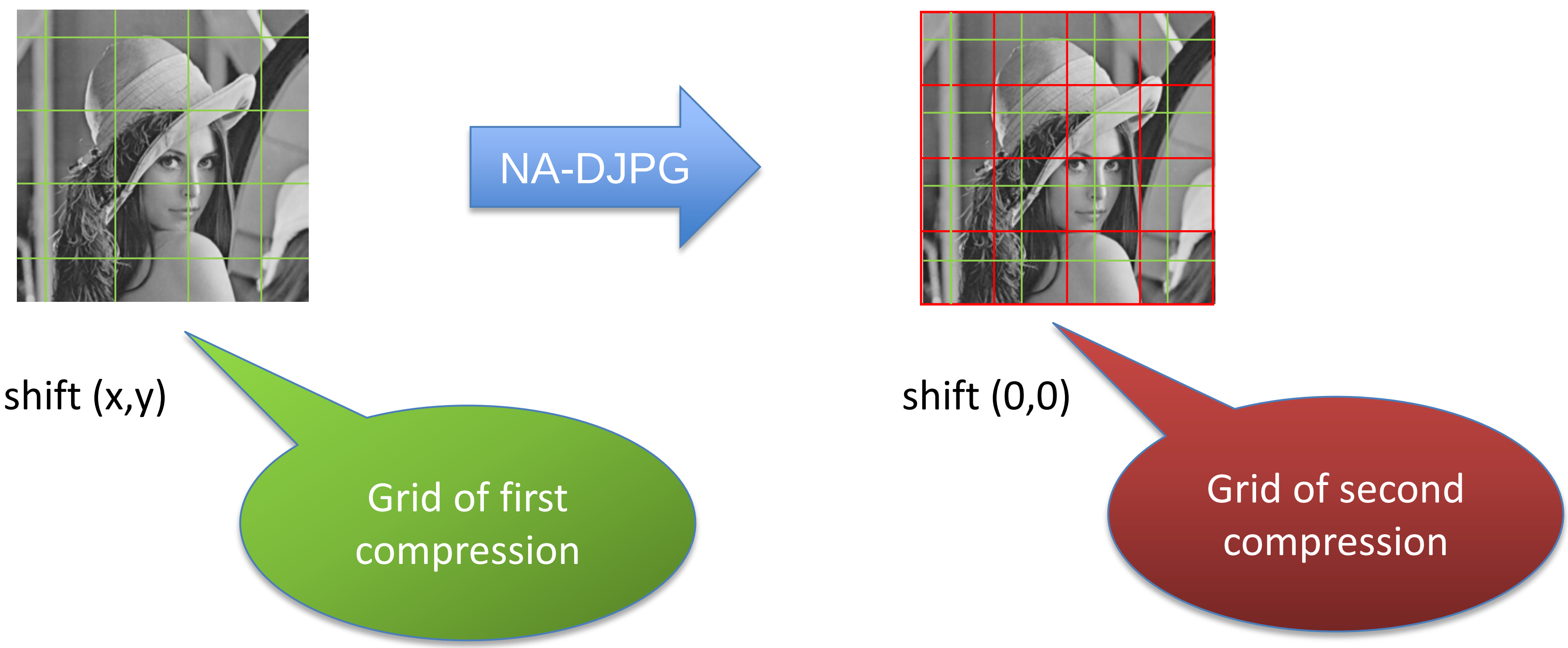
Grid of second compression

8x8 Blocks aligned to upper left corner

8x8 Blocks aligned to upper left corner

Double compression chain

- DCT grids of successive compressions not aligned



8x8 Blocks not aligned to upper left corner

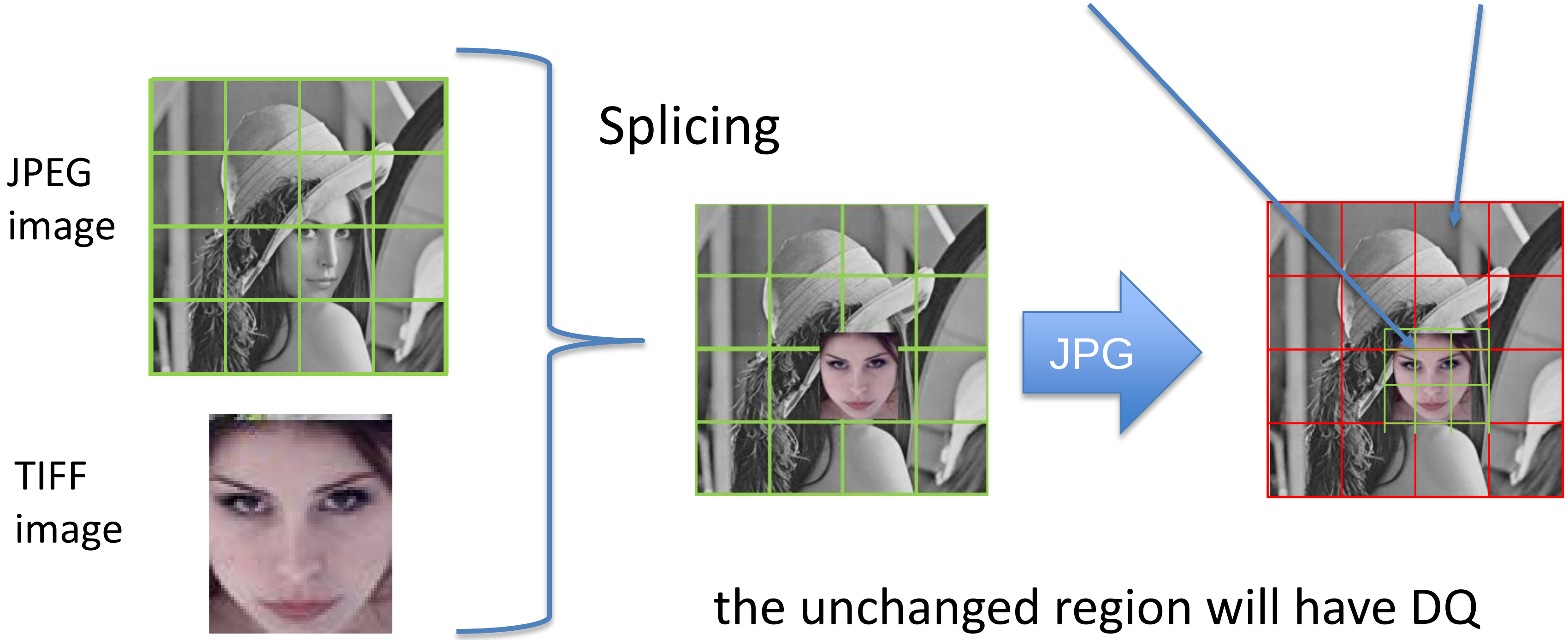
8x8 Blocks aligned to upper left corner

Tampering in JPEG images

- In previous analysis it has been assumed that all the image exhibits A-DJPG or NA-DJPG.
- But in case of manipulation of a JPEG image, this does not hold,
- Depending on the kind of manipulation (here splicing will be assumed) the tampered image will exhibit mixtures of single compression, A-DJPG or NA-DJPG.

A-DJPEG in tampered JPEG images

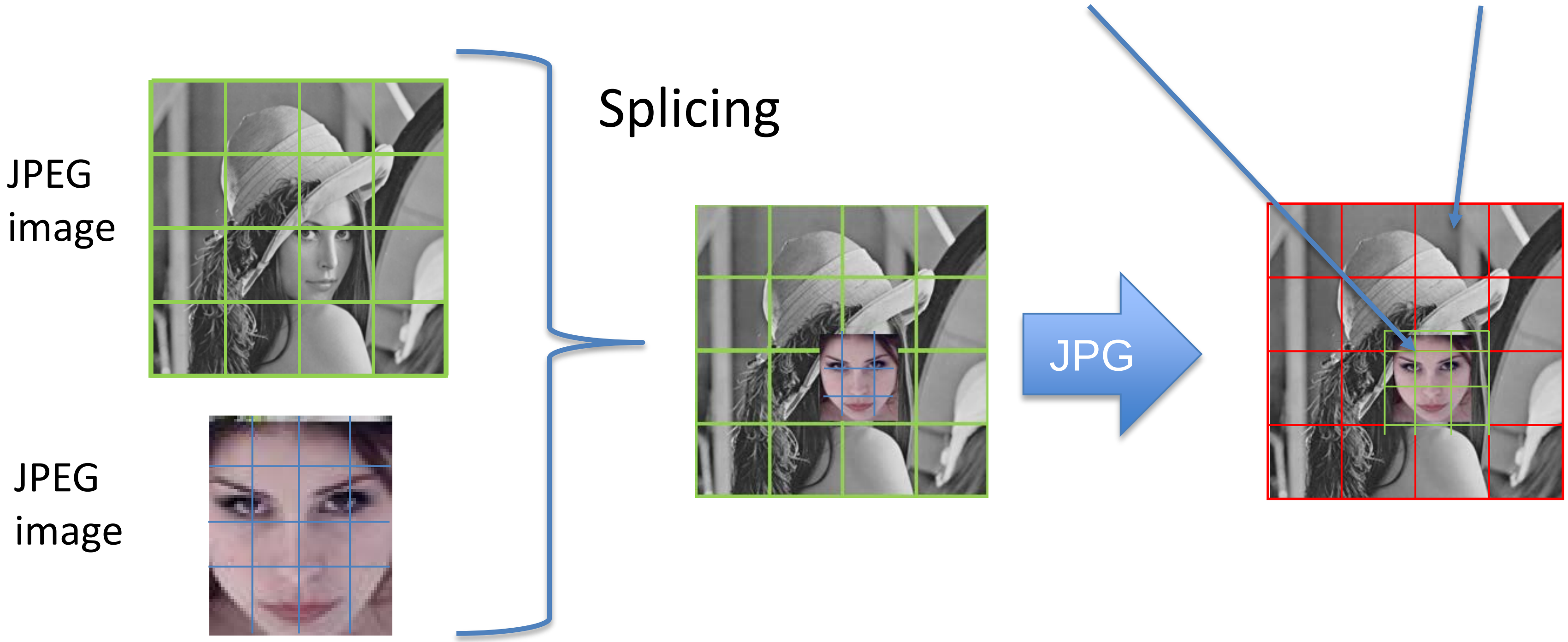
- Due to tampering, some blocks are single JPEG, others A-DJPG.



the unchanged region will have DQ effect, while the tampered region will not

NA-DJPEG in tampered JPEG images

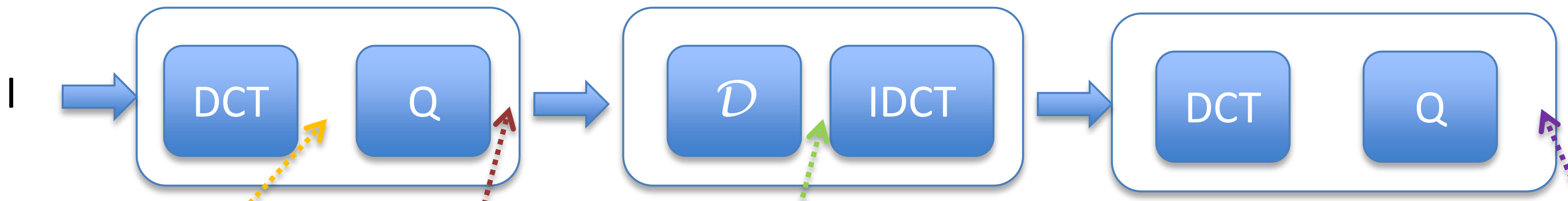
- Due to tampering, some blocks are NA-JPEG, others A-DJPG.



A-DJPG: the DQ effect

When an image is double JPEG-compressed with aligned grids, it will undergo the following steps :

What happens to DCT coefficients:



x

$$Q_{Q_1}(x) = \left[\frac{x}{Q_1}\right]$$

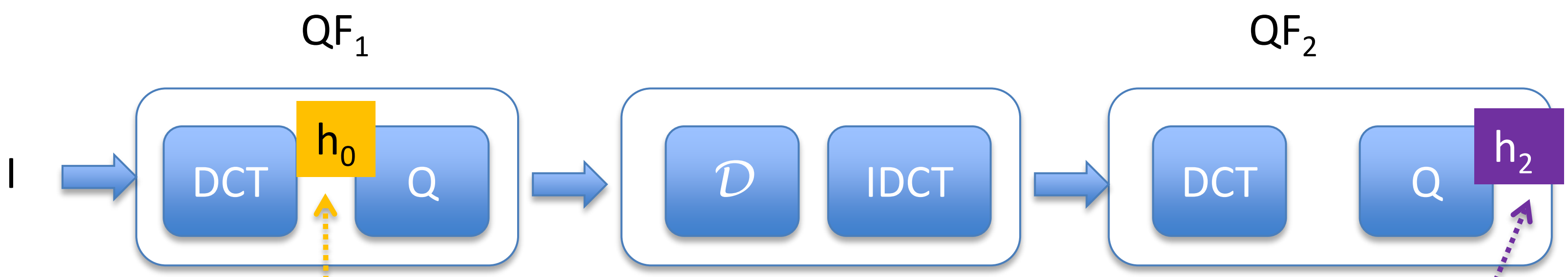
$$Q_{Q_1}^{-1}\left(Q_{Q_1}(x)\right) = \left[\frac{x}{Q_1}\right] Q_1$$

$$Q_{Q_1 Q_2}(x) = \left[\left[\frac{x}{Q_1}\right]\frac{Q_1}{Q_2}\right]$$

Double Quantization effect

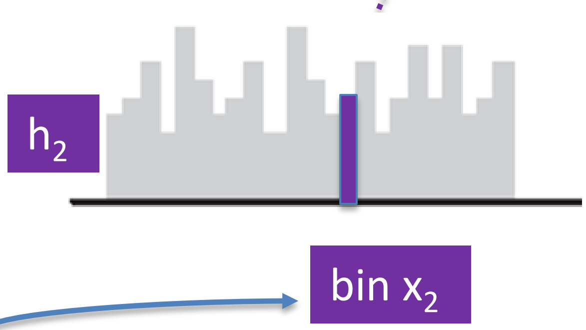
Look at histograms of DCT coefficients of a given frequency :

h_0: histogram before DQ, h_2 histogram afterDQ



Suppose coeffs in bin x_0 of h_0 go in a bin x_2 in h_2 due to DQ:

$$x_2 = \left[\left[\frac{x_0}{Q_1}\right]\frac{Q_1}{Q_2}\right]$$



Double Quantization effect

- $n(x_2)$ number of bins in h_0 contributing to bin x_2 in the DQ histogram h_2 depends on the value x_2 :

$$n(x_2) = R(x_2) - L(x_2)$$

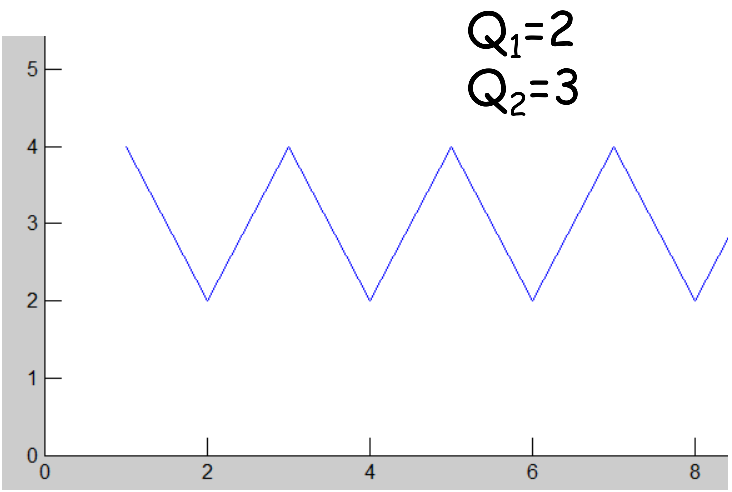
where

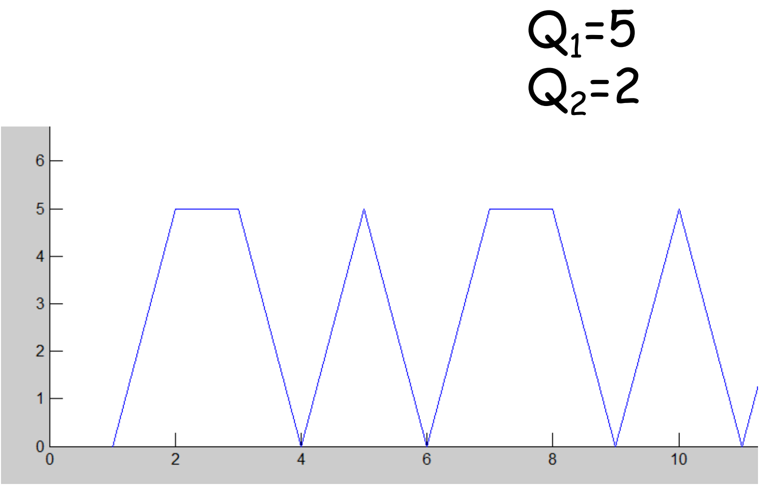
$$\begin{cases} R(x) = Q_1\left(\left\lfloor \frac{Q_2}{Q_1}\left(x+\frac{1}{2}\right)\right\rfloor + \frac{1}{2}\right) \\ L(x) = Q_1\left(\left\lceil \frac{Q_2}{Q_1}\left(x-\frac{1}{2}\right)\right\rceil - \frac{1}{2}\right) \end{cases}$$

Z. Lin et al., "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11.

Double Quantization effect

- $n(x_2)$ is a periodic function, period: $p = Q_1/\mathrm{gcd}(Q_1,Q_2)$
  - Gcd: greatest common divisr.
- This periodicity introduces periodic pattern (peaks / valleys) in histograms of double quantized coeffs: this is the DQ effect

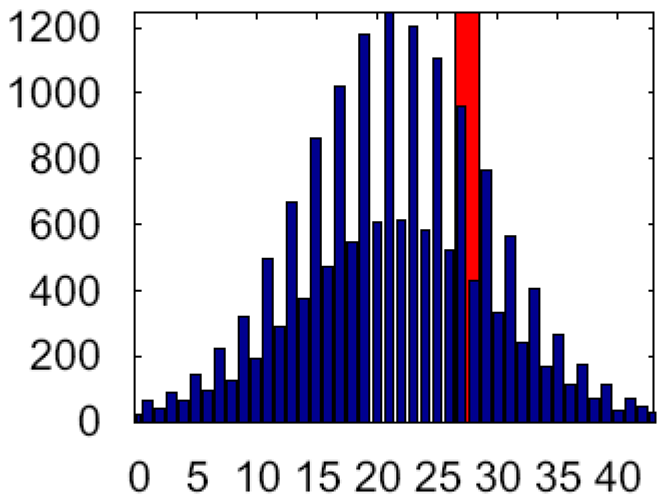




Double Quantization effect

- **Second compression worst quality than the first one**
- When $Q_2>Q_1$ the histogram can exhibit some periodic pattern of peaks and valley:

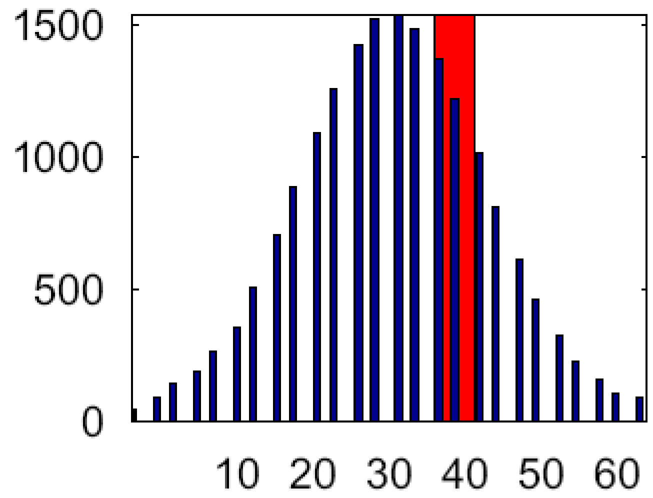
histogram with steps $Q_1=2$ and $Q_2=3$.
The shaded rectangles show one period of the histograms.



Double Quantization effect

- **Second compression better quality than the first one**
- If $Q_2 < Q_1$ then $n(x_2)=0$ for some x_2.
- For example, if $Q_1 = 5$, $Q_2 = 2$, then $n(5k+1)=0$.
- This means that the histogram has periodically missing values:

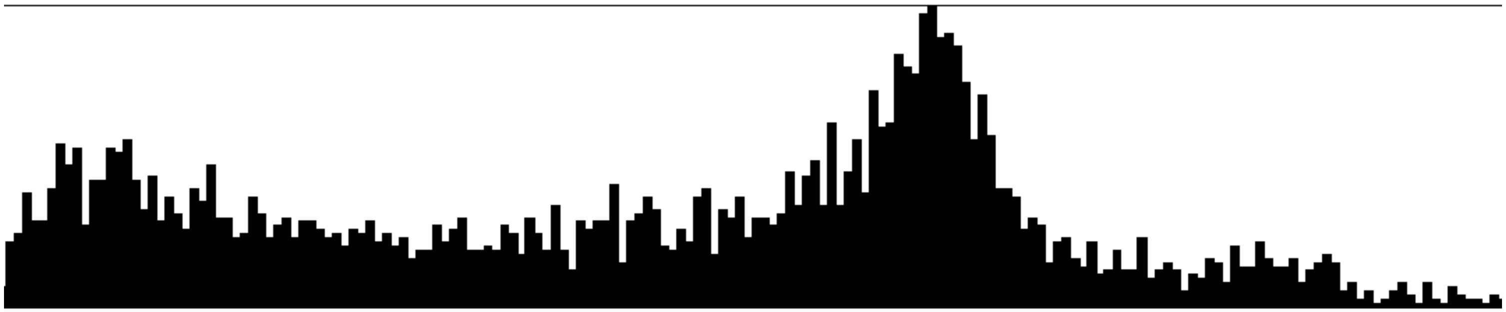
histogram with steps Q_1=5 and Q_2=2.
The shaded rectangles show one period of the histograms.



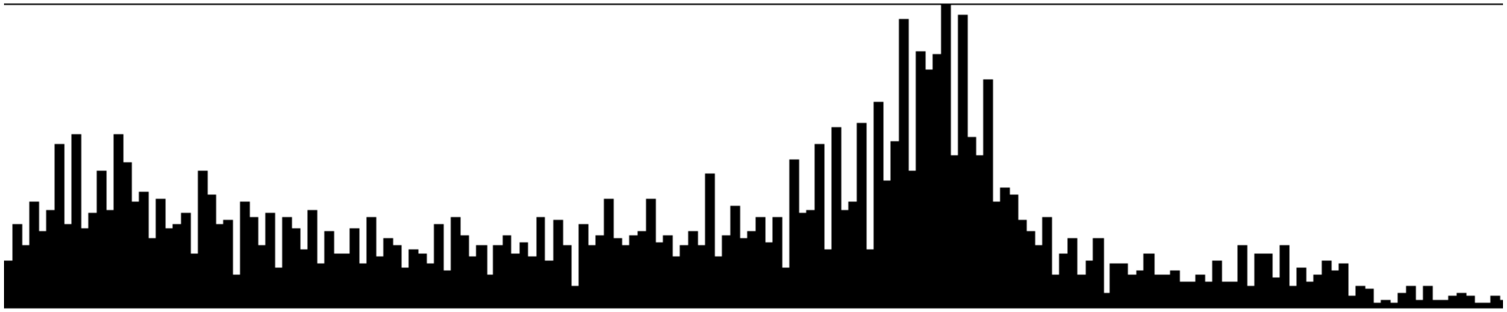
A-DJPG Detection & Estimation

- Given an image in JPEG format, we can detect if the image has been double compressed.
- To this end, the histograms of the DCT coefficients are computed.
- If these histograms contain periodic patterns, then the image is very likely to have been double compressed.
- E.g. consider DCT coefficients correspond to DCT frequencies (1, 1) and their histograms ....

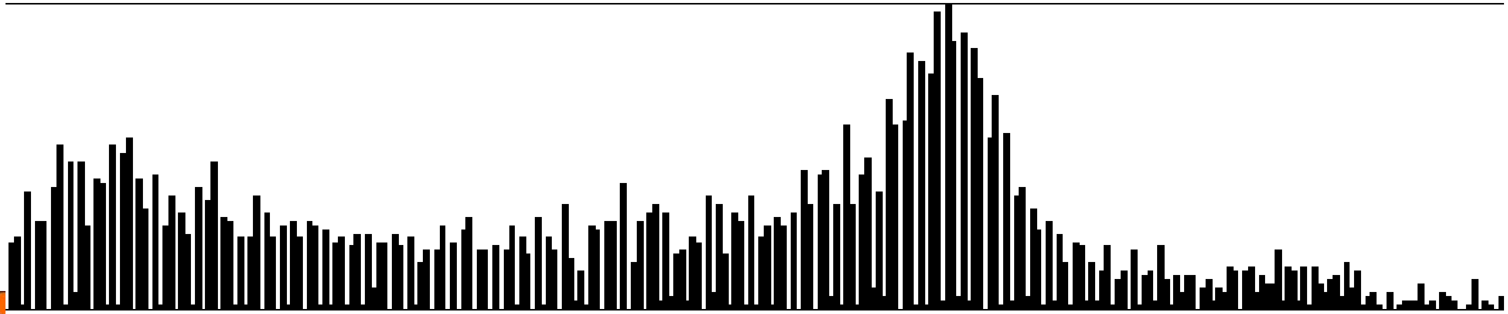
A-DJPG Detection & Estimation



- for an image single JPEG compressed with QF 75



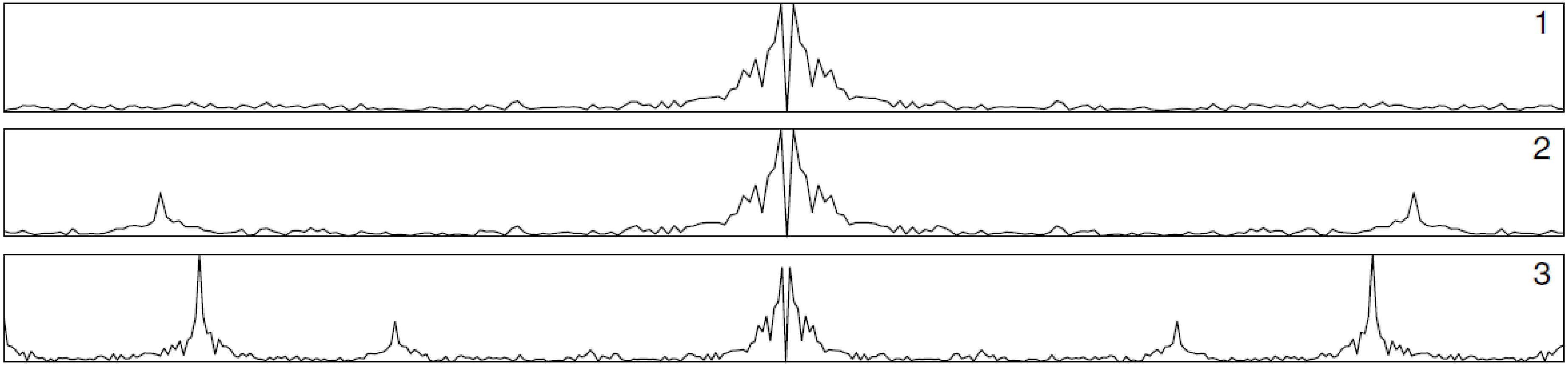
- double JPEG compressed with QF 85 followed by 75



- double JPEG compressed with QF 75 followed by 85

A-DJPG Detection & Estimation

- These periodic artifacts are particularly visible in the Fourier domain as strong peaks in the mid and high frequencies.



- Patterns depend on the quality parameters. As a result, it is possible to estimate the QFs that have been used. Q_2 can be found from the quantization table stored in the JPEG file. Q_1 can be inferred from the location of the frequency peaks in the Fourier transforms of the DCT coefficient histograms.

A-DJPG Localization

Histogram of tampered JPEG image as the superposition of two histograms:

- One shows DQ effect as periodical peaks and valleys (unchanged blocks)
- the other has random bin values (tampered blocks).

- But we don't know *a priori which are tampered and unchanged regions*.
- We only have total histograms !
- So we have to infer the probability of a block being tampered (H_t) or not (H_u)

H_t= Single Q H_u = A-DJPG

A-DJPG Localization

- Idea: use Bayesian inference to assign to each DCT coefficient x a probability of being doubly quantized or not.
- Based on the probability distribution of DCT coefficients conditional to the hypothesis of being tampered $p(x|H_t)$ or of being original $p(x|H_u)$
- These conditional probabilities are estimated from observed histogram of x, h(x), through a proper statistical model .

H_t= Single Q
H_u = A-DJPG

$$p(x/H_t) = \tilde{h}(x)$$

$$p(x/H_u) = \sum_{L(x) \le u \le R(x)} h_0(u)$$

histogram after a single compression with Q_2.

histogram before the first compression

A-DJPG Localization

$\tilde{h}(x)$	Estimated considering the DCT coefficients obtained by recompressing with Q_2 a slightly cropped version of the image under analysis
$h_0(x)$	Not available

- Hence, we propose to introduce the following approximation

$$\frac{1}{n(x)} \sum_{L(x) \leq u < R(x)} h_0(u) \approx \frac{1}{Q_2} \sum_{L'(x) \leq u < R'(x)} h_0(u) \triangleq \tilde{h}(x)$$

where $L'(x) = Q_2 x - Q_2/2$ and $R'(x) = Q_2 x + Q_2/2$, thus bin width $R'(x)-L'(x)=Q_2$

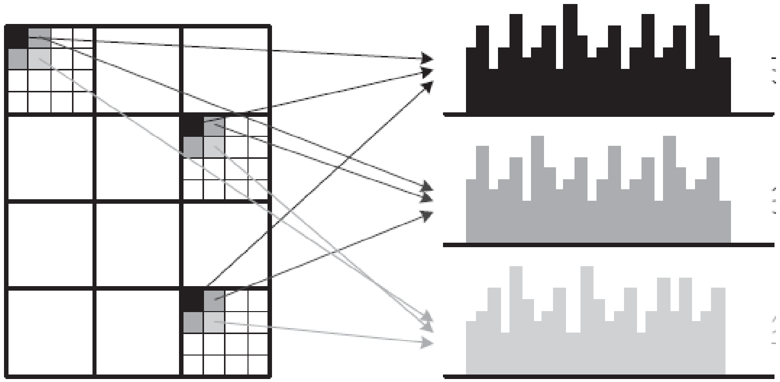
- The above approximation holds whenever n(x) > 0 and the histogram of the original DCT coefficient is locally uniform. In practice, we found that for moderate values of Q2 this is usually true, except for the center bin (x=0) of the AC coefficients, which have a Laplacian like distribution.

A-DJPG Localization

- Thus for each coeff.

$$p(x/H_t) = \tilde{h}(x)$$
$$p(x/H_u) = n(x)\tilde{h}(x)$$

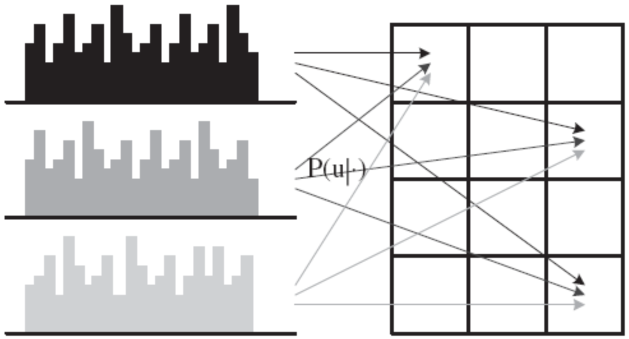
This holds for the histograms of each frequency coefficient



A-DJPG Localization

- From the naive Bayesian approach (equiprobable H_t and H_u),
- If a block contributes to the (x)-th bin, then the posterior probability of it being a tampered block is:

$$p(H_t/x) = \frac{p(x/H_t)}{p(x/H_t) + p(x/H_u)} = \frac{1}{1 + n(x)}$$



- With all the available histograms, we can accumulate the probabilities to give the posterior probability of the whole block
- Accumulated probability means:
  - Under the hypothesis that DCT coeffs in block are mutually independent
  - corresponding prob. values are multiplicated

A-DJPG Localization

- The probability of a block being tampered is then:

$$p = \frac{1}{1 + \prod_{i \mid x_i \neq 0} n_i(x_i)}$$

  - $n_i(x_i)$ indicates function n(x) related to the i-th DCT coefficient
  - computation does not take into account coeffs equal to zero.
- only the first low frequency coefficients are used in practice.
- in order to compute *p*, we need a reliable estimate of the quantization table used by the first JPEG compression.

A-DJPG Localization

- If we assume that the probability distribution of the observed coefficients for x ≠ 0 can be modeled as a mixture

$$p(x; Q_1, \alpha) = \alpha \cdot n(x; Q_1) \cdot \tilde{h}(x) + (1 - \alpha) \cdot \tilde{h}(x)$$

where α is the mixture parameter . Q_1 can be estimated (by trying every possible Q_1 in a limited set of possible values) as

$$\hat{Q}_1 = \arg\min_{Q_1} \sum_{x \neq 0} [h(x) - p(x; Q_1, \alpha_{opt})]^2$$

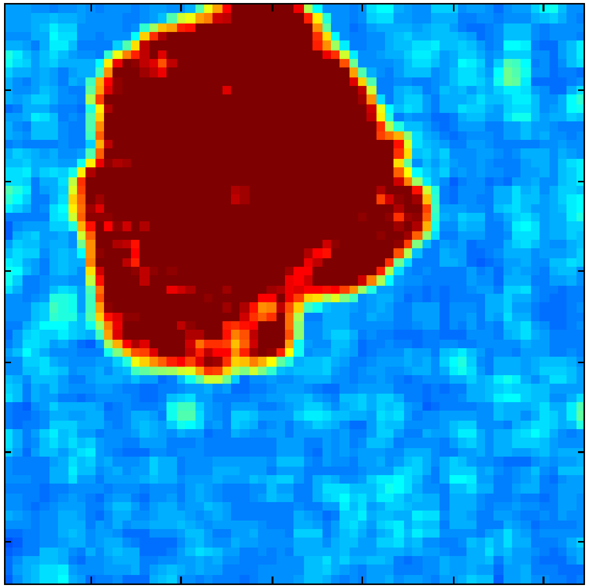
for each Q_1, α_{opt} is the optimal parameter in the least square sense:

$$\alpha_{opt} = -\frac{\sum_{x \neq 0} \tilde{h}(x)[n(x; Q_1) - 1] \cdot [\tilde{h}(x) - h(x)]}{\sum_{x \neq 0} \tilde{h}(x)^2 [n(x; Q_1) - 1]^2}.$$

To estimate the complete quantization matrix, the above minimization problem is separately solved for each of the 64 DCT coefficients.

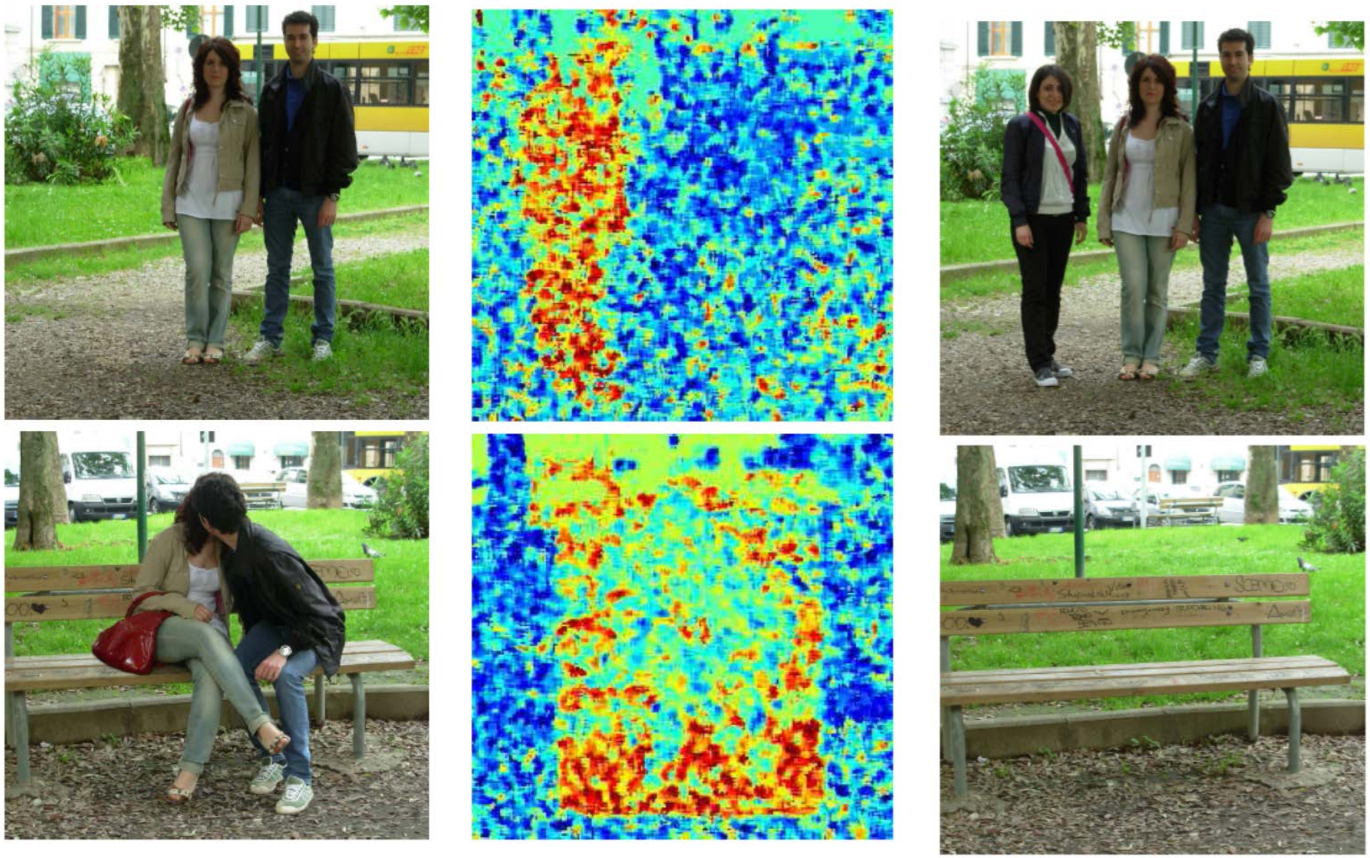
A-DJPG Localization

- Such probabilities, accumulated over each 8 × 8 block, will provide a DQ probability map allowing us to tell original areas (high DQ prob.) from tampered areas (low DQ prob.).



A-DJPG Localization

- Application to realistic forgeries: (a) images under analysis; (b) probability maps (c) original images



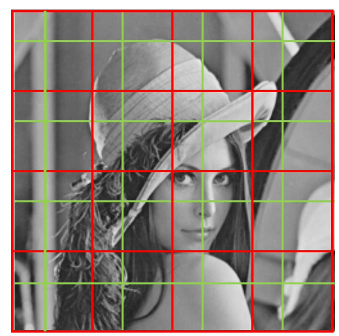
NA-DJPG Detection & Estimation

- Based on a single feature which depends on the integer

Input : JPEG image I

Hypothesis testing

I is single compressed shift (0,0)



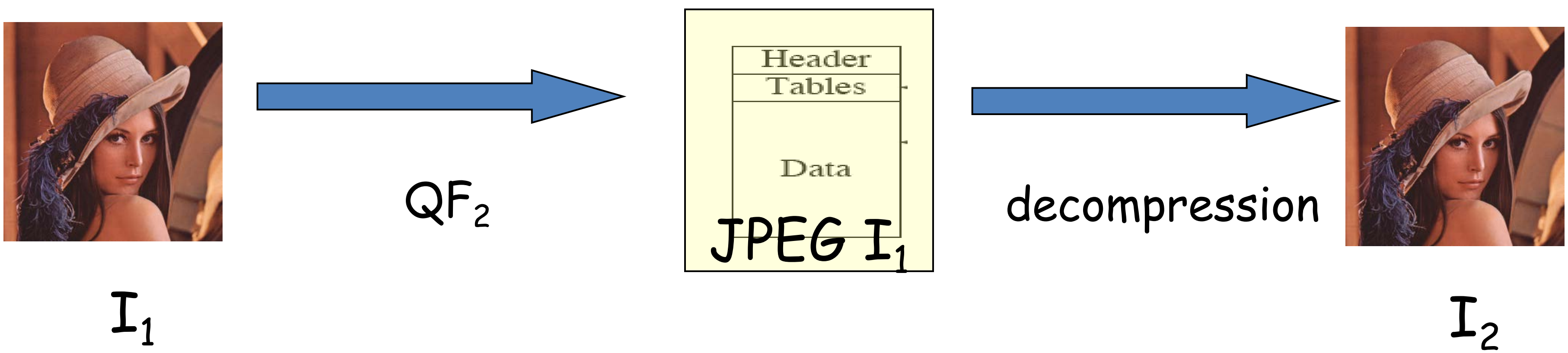
I is double compressed with first compression shift (x,y) and Quality factor QF_1 to be estimated

Quality factor QF : known from header
Blocks with shift (0,0)

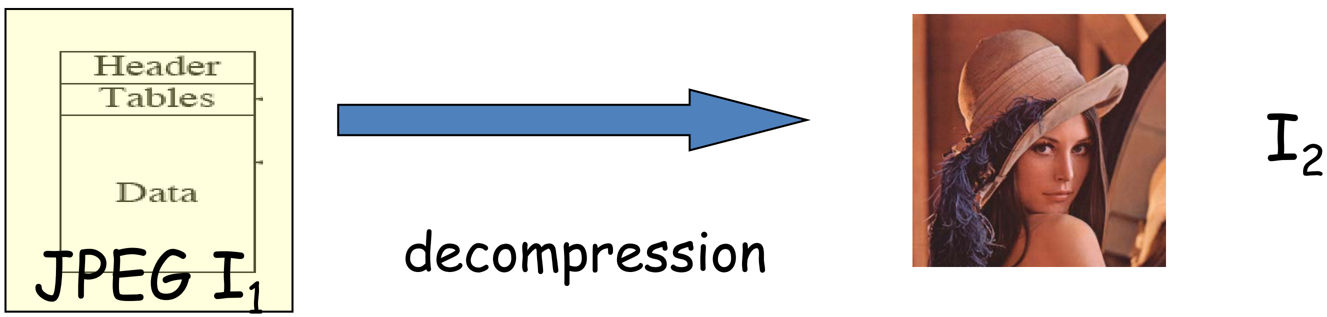
T.Bianchi, A.Piva, "Detection of Nonaligned Double JPEG Compression Based on Integer Periodicity Maps", IEEE Transactions on Information Forensics & Security, vol. 7, no. 2, April 2012.

NA-DJPG Detection & Estimation

- Let us assume that an original image I_1 is JPEG compressed with QF_2, grid aligned with the upper left corner of the image
- then decompressed to obtain the image I_2 :



NA-DJPG Detection & Estimation



Decompression includes:

- Entropy decoding & Runlength decoding
  - lossless – we dont'care
- Dequantization of DCT coefficients
- IDCT

NA-DJPG Detection & Estimation

- I_2 can thus be modeled as:

$$\mathbf{I}_2 = \mathbf{D}_{00}^{-1} Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{E}_2 = \mathbf{I}_1 + \mathbf{R}_2$$

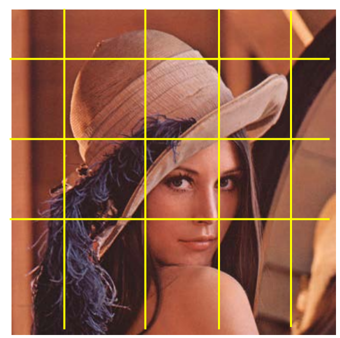
I_2

- I_1 : original image;
- D_{00} : 8x8 DCT with grid aligned to upper left corner
- $Q_2(.)$: quantization / dequantization with QF_2
- E_2 : error due to Rounding/Truncating values to 8 bit;
- R_2 : overall error introduced by JPEG w.r.t. the original.

NA-DJPG Detection & Estimation

- If the original image I_1 was previously JPEG compressed with QF_1 and a grid shifted by (y,x) w.r.t. the upper left corner, starting from an uncompressed image I_0:

compression



I_1 compressed with shift (y,x) and decompressed

Uncompressed I_0

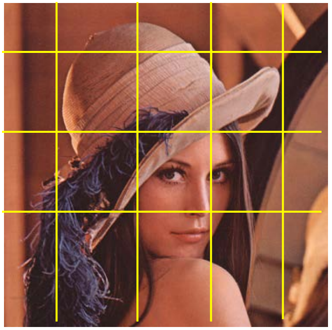
$$\mathbf{I}_1 = \mathbf{D}_{yx}^{-1} Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \mathbf{E}_1$$

$0 \leq x \leq 7$
$0 \leq y \leq 7$

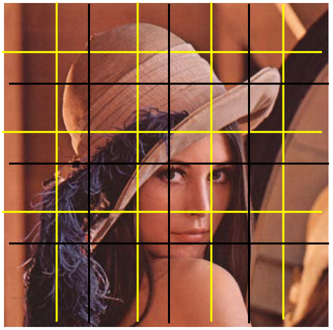
NA-DJPG Detection & Estimation

- Then I_2 has been doubly compressed and finally decompressed:

I_0



I_1



I_2

$$\mathbf{I}_2 = \mathbf{I}_1 + \mathbf{R}_2 = \mathbf{D}_{yx}^{-1} Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \mathbf{E}_1 + \mathbf{R}_2$$

NA-DJPG Detection & Estimation

- We want now to analyze I_2
- We take I_2 and we apply to it a block DCT with a grid shift (i,j).

$0 \leq i \leq 7$
$0 \leq j \leq 7$

- Blue: grid of second compression with QF_2, shift (0,0)
- Yellow: grid of first compression with QF_1, shift (y,x)
- Green: grid of new block DCT, with shift (i,j)

NA-DJPG Detection & Estimation

What happens to the statistics of DCT coefficients present in these blocks ? 3 cases:

Case a)

(i,j)= (0,0)

Blocks aligned with last compression

Case b)

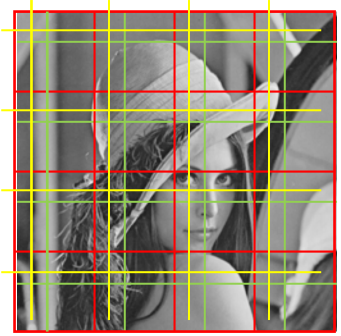
(i,j)= (y,x)

Blocks aligned with *possible* first compression

Case c)

(i,j) ≠(y,x)
(i,j) ≠(0,0)

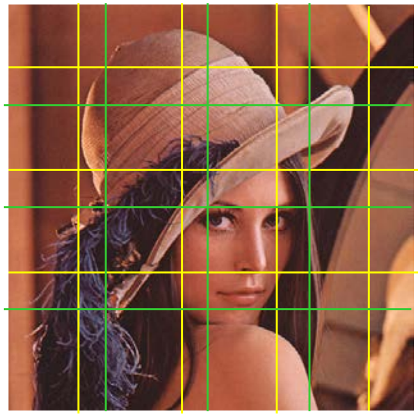
grid is aligned to neither of the two JPEG compressions



NA-DJPG Detection & Estimation

- grid aligned to the one of the last compression, i.e. i = 0, j = 0:

$$\mathbf{D}_{ij}\mathbf{I}_2 = \mathbf{D}_{00}\mathbf{I}_2 = \mathbf{D}_{00}\left(\mathbf{D}_{00}^{-1}Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right)+\mathbf{E}_2\right)= Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right)+\mathbf{D}_{00}\mathbf{E}_2$$

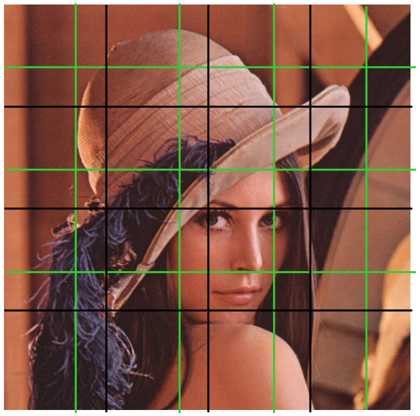


i.e. the green grid is aligned with the blue one

NA-DJPG Detection & Estimation

- grid aligned to the one of the first compression, i.e. i = y, j = x:

$$\mathbf{D}_{ij}\mathbf{I}_2 = \mathbf{D}_{yx}\mathbf{I}_2 = \mathbf{D}_{yx}(\mathbf{I}_1 + \mathbf{R}_2) = Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2)$$



i.e. the green grid is aligned with the yellow one

NA-DJPG Detection & Estimation

- Grid is aligned to neither of the two compressions, i.e. i ≠ 0,y, j ≠ 0,x:

$$\mathbf{D}_{ij}\mathbf{I}_2 = \mathbf{D}_{ij}\mathbf{D}_{00}^{-1}Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{D}_{ij}\mathbf{E}_2$$

i.e. the green grid is not aligned with the yellow and blue ones

NA-DJPG Detection & Estimation

- In summary, if a block DCT with grid shift (i,j) is applied to I_2, we can have 3 cases:

$$\mathbf{D}_{ij}\mathbf{I}_2 = \begin{cases} Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{D}_{00}\mathbf{E}_2 & \mathrm{i}=0, \mathrm{j}=0 \\ Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2) & \mathrm{i}=y, \mathrm{j}= \\ \mathbf{D}_{ij}\mathbf{D}_{00}^{-1}Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{D}_{ij}\mathbf{E}_2 & \text{elsewhere} \end{cases}$$

If the grid is aligned to the one of the last compression

If the grid is aligned to the one of the first compression

if the DCT grid is aligned with neither of the two compressions

NA-DJPG Detection & Estimation

- when the grid is aligned with the one of the last/first compression, coefficients tend to cluster around the points of the lattices defined by$Q_2()$ and $Q_1()$
- with a spread due to presence of the error terms []:

$$\mathbf{D}_{ij}\mathbf{I}_2 = \begin{cases} Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \left[\mathbf{D}_{00}\mathbf{E}_2\right] & \mathrm{i} \ldots \\ Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \left[\mathbf{D}_{yx}\left(\mathbf{E}_1 + \mathbf{R}_2\right)\right] & \mathrm{i} = \mathrm{y}, \mathrm{j} = \mathrm{x} \end{cases}$$

- spread fixed and quite limited, approx. Gaussian distributed with 0 mean and variance 1/12.

- Spread depends on power of R_2, i.e., on second compression quality.: approx. Gaussian distributed with 0 mean and variance $(Q^2_2+1)/12$

NA-DJPEG Bianchi's method

- When the DCT grid is aligned with neither of the two compressions, coeffs usually do not cluster around any lattice:

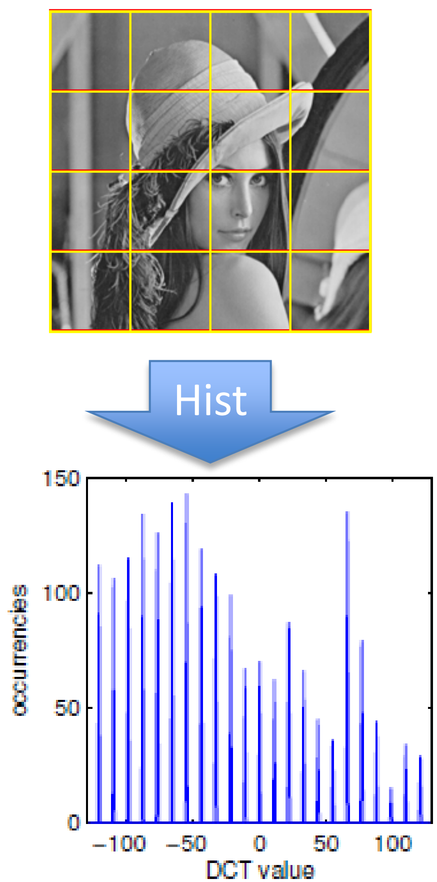
- No clustering !

$$\mathbf{D}_{ij}\mathbf{I}_2 = \begin{cases} Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{D} \cdots {}_2 & \mathrm{i}=0, \mathrm{j}=0 \\ Q_1\left(\mathbf{D}_{yx}\mathbf{I}_0\right) + \mathbf{D}_{yx}(\mathbf{E}_1 + \mathbf{R}_2) & \mathrm{i}=\mathrm{y}, \mathrm{j}=\mathrm{x} \\ \mathbf{D}_{ij}\mathbf{D}_{00}^{-1}Q_2\left(\mathbf{D}_{00}\mathbf{I}_1\right) + \mathbf{D}_{ij}\mathbf{E}_2 & \text{elsewhere} \end{cases}$$

We can see this behavior if we make the histogram of the coefficients of same frequency (e.g. the DC coefficient) taken from all the blocks in which we have divided I_2 .....

NA-DJPG Detection & Estimation

- Histogram of the DC coefficients of each block



Case a)

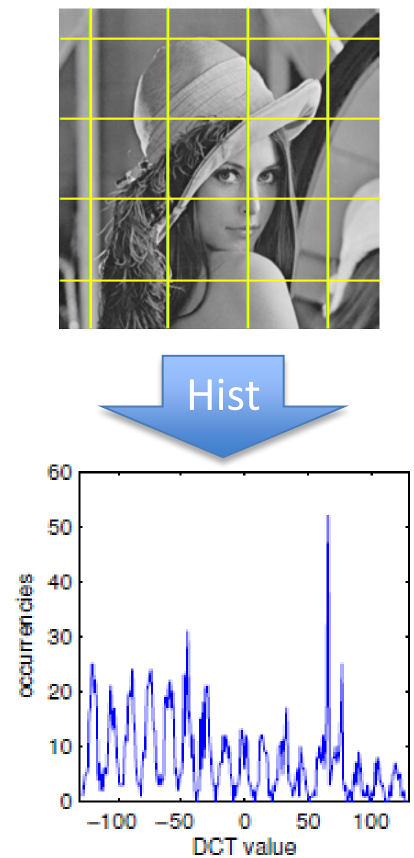
(i,j)= (0,0)

Blocks aligned with last compression

DCT coefficients cluster around the points of a lattice defined by Q_2 , with a spread due to presence of error terms due to the last compression:

NA-DJPG Detection & Estimation

- Histogram of the DC coefficients of each block



Case b)

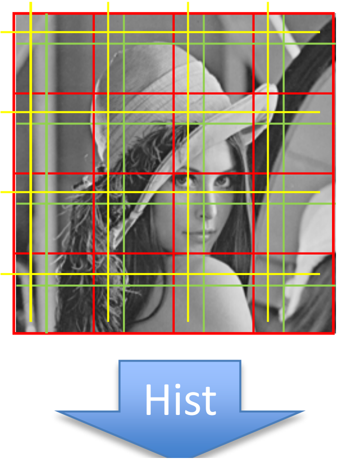
(i,j)= (x,y)

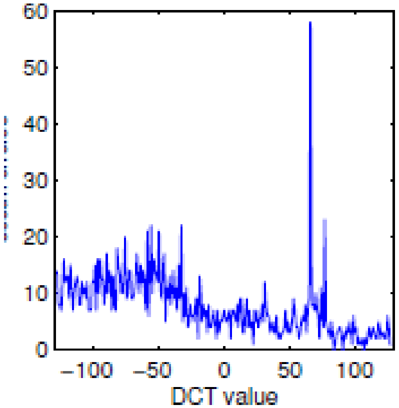
Blocks aligned with *possible* first compression

DCT coefficients cluster around the points of a lattice defined by Q_1 , with a spread due to presence of error terms due to the first and last compression (higher noise)

NA-DJPG Detection & Estimation

- Histogram of the DC coefficients of each block





Case c)

(i,j) ≠(x,y)

(i,j) ≠(0,0)

grid is aligned to neither of the two JPEG compressions

DCT coefficients usually do not cluster around any lattice

NA-DJPG Detection & Estimation

- Idea: detect NA-DJPEG by measuring how DCT coefficients cluster around a given lattice for any possible grid shift.
- When NA-DJPEG is detected, the parameters of the lattice also give Q_1 & shift.

- Since the effect is more evident in the DC coefficient , and to keep the detection simple, only the DC coefficient of each block is studied.

NA-DJPG Detection & Estimation

- clustering around a lattice can be measured by looking at the the periodicity of the histogram for an integer period Q
- Evaluated by considering its Fourier transform at frequencies which are reciprocal of Q:

$$f_{ij}(Q) \triangleq \sum_{k} h_{ij}(k) e^{-j\frac{2\pi k}{Q}}, \quad Q \in \mathbb{Z} \backslash \{0\}$$

  - $H_{ij}(k)$ is the histogram value of DC coeff. for a grid shift (i,j).
  - Q : a possible quantization step of DC coefficient.

NA-DJPG Detection & Estimation

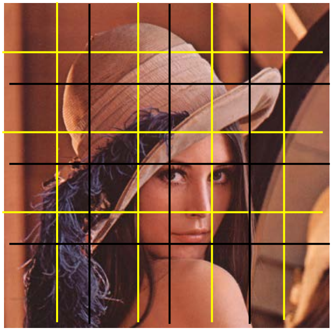
- The Fourier transform will have a maximum in the amplitude if the shift (i,j) and the quantization scale Q are the same of the parameters of a previous compression.

$$f_{ij}(Q) \triangleq \sum_{k} h_{ij}(k) e^{-j\frac{2\pi k}{Q}}, \quad Q \in \mathbb{Z} \setminus \{0\}$$

- What happens to $f_{ij}(Q)$ in presence or in absence of double compression ?

NA-DJPG Detection & Estimation

Presence of NA-DJPEG :



I_2 doubly compressed, first with QF_1 and shift (y,x), next with QF_2 and shift (0,0).

- Both $f_{00}(Q_2)$ and $f_{yx}(Q_1)$ have higher magnitude than other values.

Absence of NA-DJPEG

I_2 singly compressed, with QF_2 and shift (0,0).

- only $f_{00}(Q_2)$ will have higher magnitude.
- for $Q \neq Q_2$, $f_{ij}(Q)$ changes very little with (i, j).

NA-DJPG Detection & Estimation

Problem:

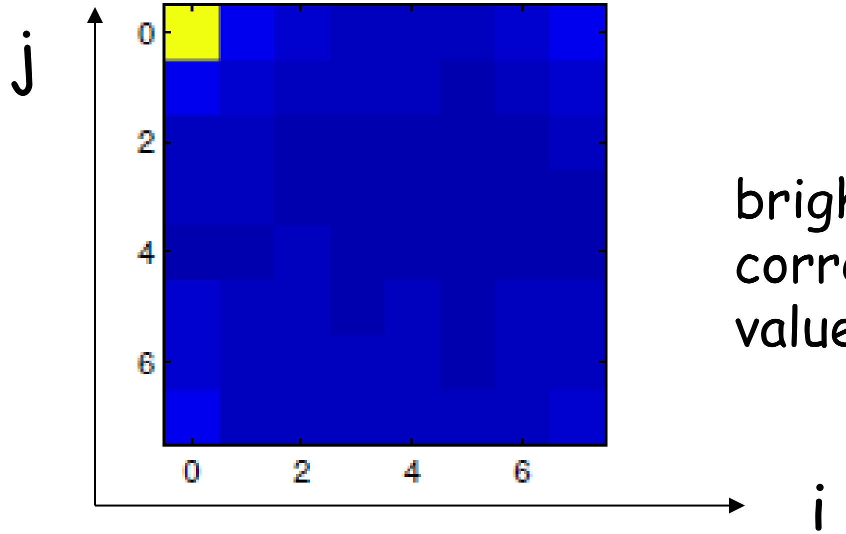
- We don't know which shift was applied in the first – possible – compression, so we have to try all combinations (i,j)
- Then, for all possible grid shifts (i,j):
  - we apply a block DCT
  - we take all DC coefficients and we quantize them by Q.
  - We compute the magnitude of Fourier transforms

- How to analyze all shifts together ?
- We introduced the **Integer Periodicity Map** (IPM)

NA-DJPG Detection & Estimation

- IPM at the quantization step Q allows us to visualize the values of the FT for each possible grid shift:

$$M_{ij}(Q) \triangleq \frac{|f_{ij}(Q)|}{\sum_{i'j'} |f_{i'j'}(Q)|}, \quad 0 \leq i \leq 7,\ 0 \leq j \leq 7.$$



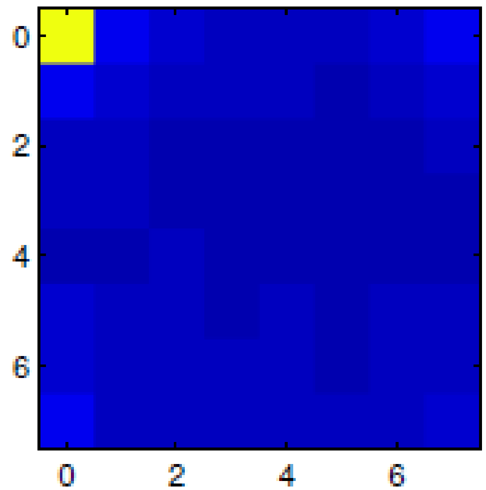
bright/dark points correspond to high/low values of M_{ij}

NA-DJPG Detection & Estimation

- $M(Q_2)$ will always show a peak at the location (0; 0) due to the last compression, since we assume I_2 has been compressed with QF_2 and shift (0,0).

Note that the value of Q_2 is extracted from the quantization matrix stored in the JPEG file.

If $Q = Q_2$



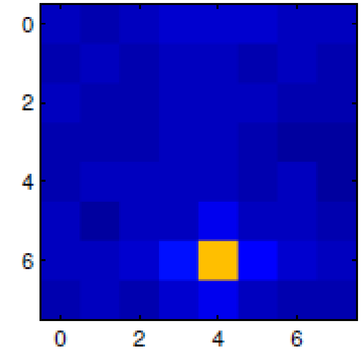
NA-DJPG Detection & Estimation

- We also don't know the value of Q_1 !
- So we can just compute the IPM for each Q in a set of possible values but $\neq Q_2$, and look at it.
- We will have:
  - Presence of NA-DJPEG: we will find a value Q_1 where $M(Q_1)$ has a single entry much greater than the others at the shift of the primary compression (y, x)
  - Absence of NA-DJPEG: M(Q) nearly uniform for every $Q \neq Q_2$, since there is only one compression.

NA-DJPG Detection & Estimation

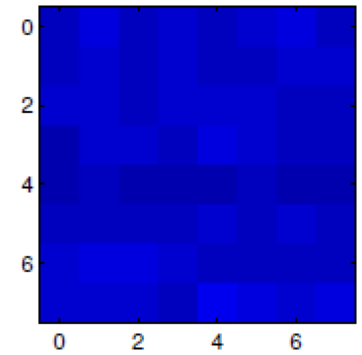
- Presence of NA-DJPEG:
- IPM has a peak

$Q=Q_1 \neq Q_2$, $(y,x) = (6, 4)$;



- Absence of NA-DJPEG:
- IPM is uniform

$Q \neq Q_2$, absence of NA-JPEG



- IPM does not work when $Q_1 = Q_2$, since the peak due to the first compression is hidden by the one in (0,0)

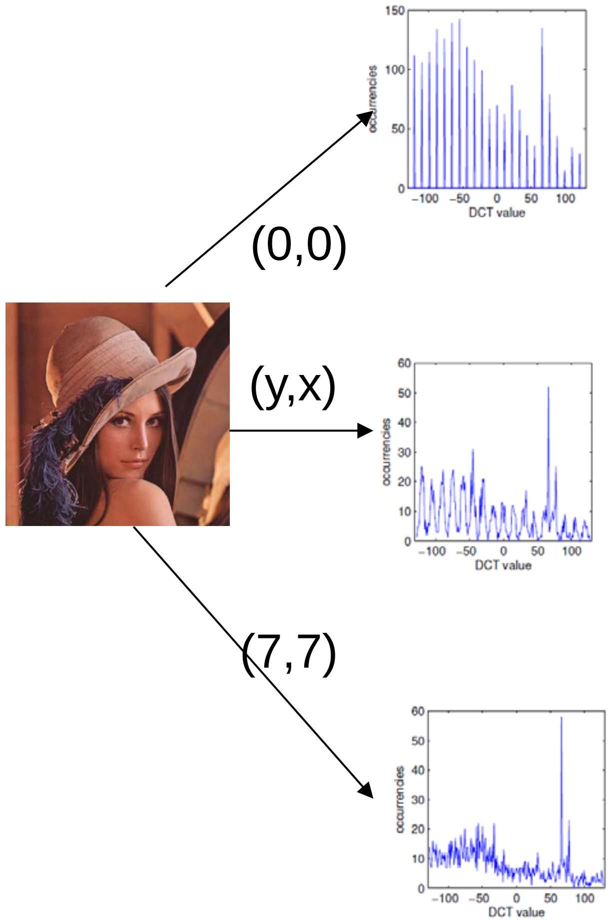
NA-DJPG Detection & Estimation

- Uniformity of IPM can be measured by its min-entropy:

$$H_\infty(Q) \triangleq \min_{ij}(-\log M_{ij}(Q)).$$

- Low $H_\infty(Q)$ $\leftrightarrow$ IPM with high peak $\leftrightarrow$ presence of NA-DJPEG
- High $H_\infty(Q)$ $\leftrightarrow$ mostly uniform IPM $\leftrightarrow$ absence of NA-DJPEG

NA-DJPG Detection & Estimation



DC coefficient histograms

input I_2

for each possible shift (i, j) do

  compute $D_{ij}I_2$

  compute histogram h_{ij}

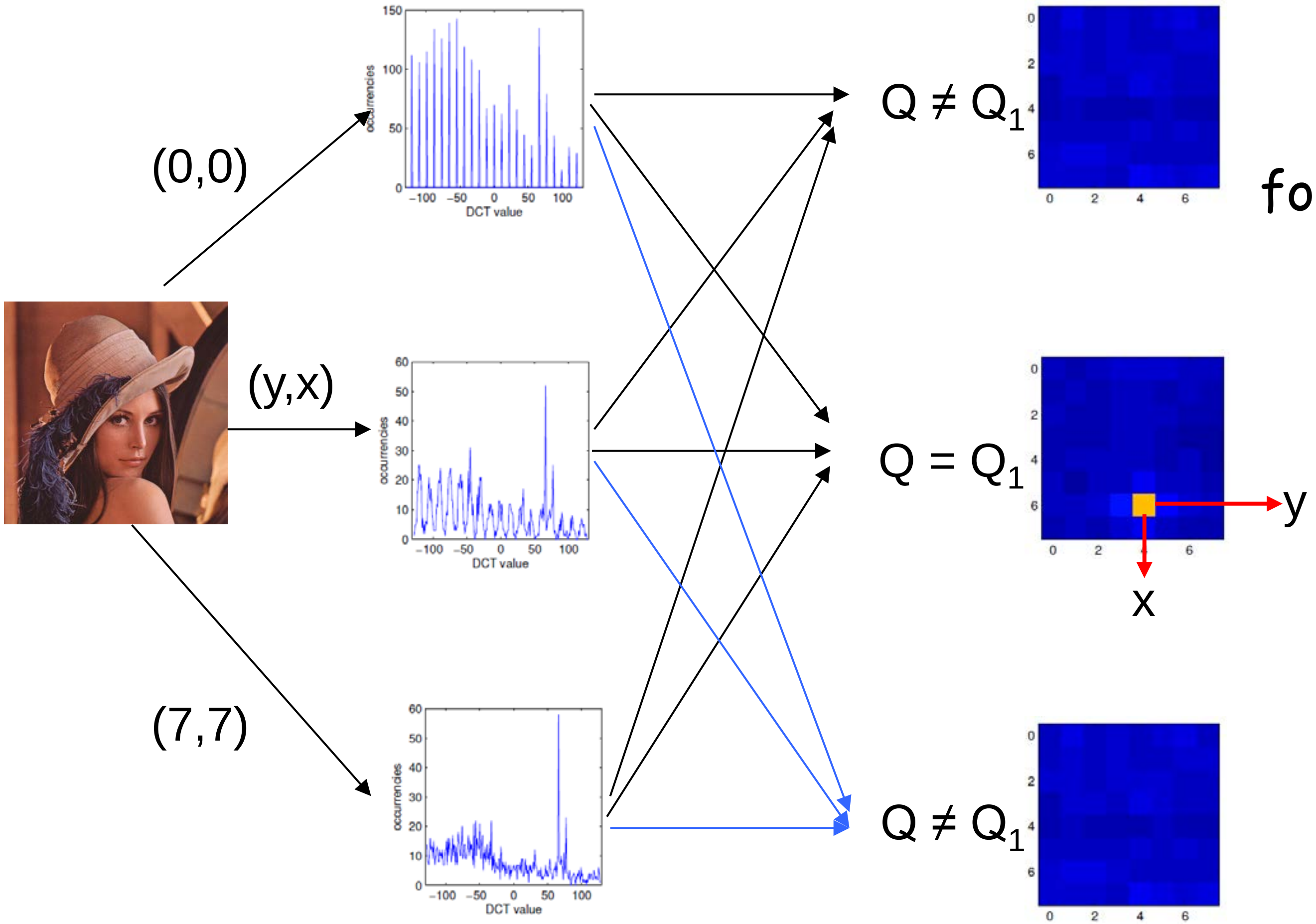
  for $Q = Q_{min}: Q_{max}$ do

    compute $f_{ij}(Q)$

  end for

end for

NA-DJPG Detection & Estimation



DC coefficient histograms

Integer Periodicity Maps

for each $Q = Q_{min}:Q_{max}$

compute $M_{ij}(Q)$

compute $H_{\infty}(Q)$

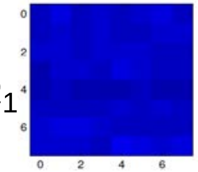
NA-DJPG Detection & Estimation

I_2 classified NA-DJPEG if:

$$\exists\ Q \neq Q_2 : H_\infty(Q) < T$$

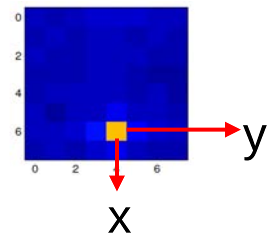
$$(y, x) = \arg\max_{(i,j)} M_{ij}(Q) \neq (0,0)$$

$Q \neq Q_1$



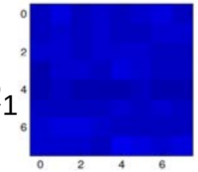
$H_\infty > T$

$Q = Q_1$



$H_\infty < T$

$Q \neq Q_1$



$H_\infty > T$

IPM

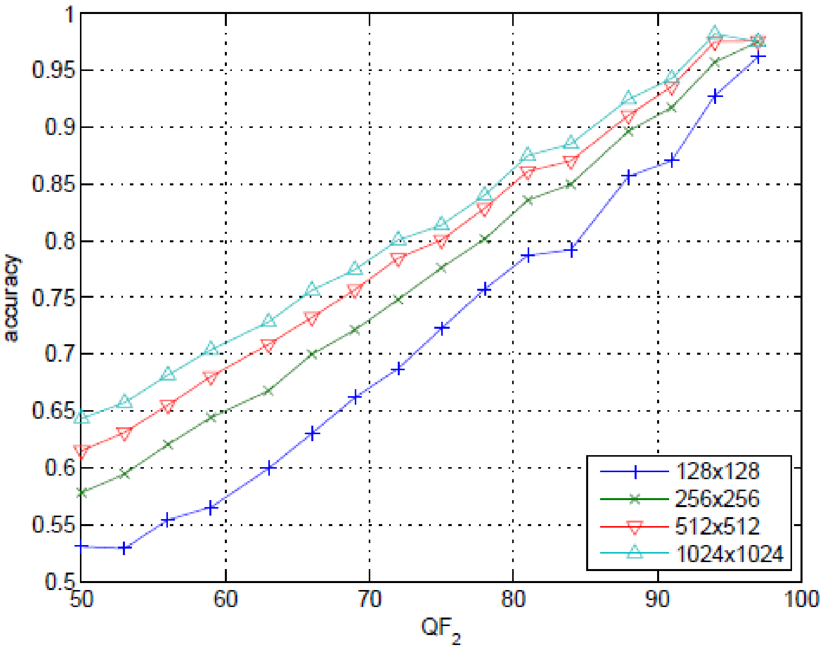
Min-Entropy

NA-DJPG Detection & Estimation

- Experimental Dataset
- 1000 original non-compressed TIFF images, central portion NxN (N=128, 256, 512, 1024) is extracted.
- Dataset without NA-DJPEG
  - Simply compressing the original images with QF_2.
- Dataset with NA-DJPEG
  - each original image is compressed with QF_1, decompressed, cropped by a random shift $(i,j) \neq (0,0)$, and recompressed with QF_2.
- QF_1 / QF_2 chosen so DC coeff. has Q_1=2:16, Q_2:1:16.
- 240000 tampered & 16000 untampered - each size

NA-DJPG Detection & Estimation

- maximum accuracy for different QF_2 and sizes, as % of correctly classified images, av. over all QF_1



- Probability of detection of the proposed detector (%) for a probability of false alarm equal to 1%, for image size 1024x1024.

QF_1 \ QF_2	50-57	58-67	68-76	77-85	86-95	96
50-57	78.1	86.9	88.6	90.4	92.4	92.1
58-67	62.3	82.6	90.5	93.8	95.5	94.1
68-76	20.3	55.9	86.0	94.4	97.1	97.5
77-85	1.5	7.9	34.7	82.3	98.1	98.6
86-95	1.3	1.3	0.8	9.9	71.7	98.6