Institut
Mines-Telecom

Transform coding and JPEG

G. Valenzise

TSIA207 – Intro Multimedia

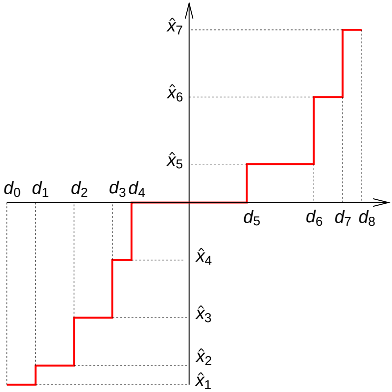
Plan

Plan

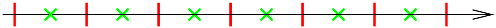
TELECOM ParisTech

Recall: Scalar quantization

$$Q : x \in \mathbb{R} \rightarrow y \in \mathcal{C} = \{\hat{x}_1, \hat{x}_2, \ldots \hat{x}_L\} \subset \mathbb{R}$$



Uniform Quantization



- Simple
- Minimizse maximum error
- Optimal for uniform random variables (RV's)

RD curve for UQ of uniform RV:

$$D = \sigma_X^2 2^{-2R}$$

RD curve for UQ of non-uniform RV in *high resolution* :

$$D = K_X \sigma_X^2 2^{-2R}$$

Uniform Quantization

Example

Image Originale, 24 bpp

Uniform Quantization

Example

Débit 21 bpp PSNR 47.19 dB TC 1.143

Uniform Quantization

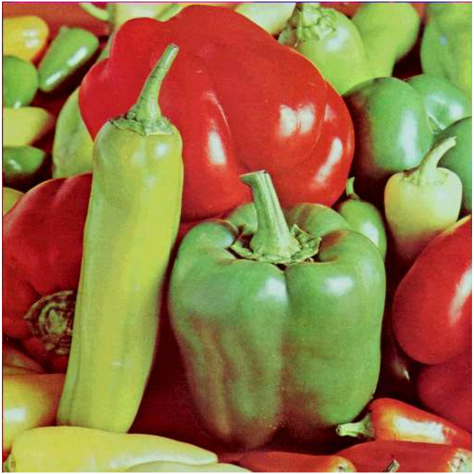
Example

Débit 18 bpp PSNR 42.38 dB TC 1.333

Uniform Quantization

Example

Débit 15 bpp PSNR 36.97 dB TC 1.600



Uniform Quantization

Example

Débit 12 bpp PSNR 31.40 dB TC 2.000

Uniform Quantization

Example

Débit 9 bpp PSNR 29.26 dB TC 2.667

Uniform Quantization

Example

Débit 6 bpp   PSNR 27.83 dB   TC 4.000

Uniform Quantization

Example

Débit 3 bpp   PSNR 25.75 dB   TC 8.000

Quantization

Conclusion

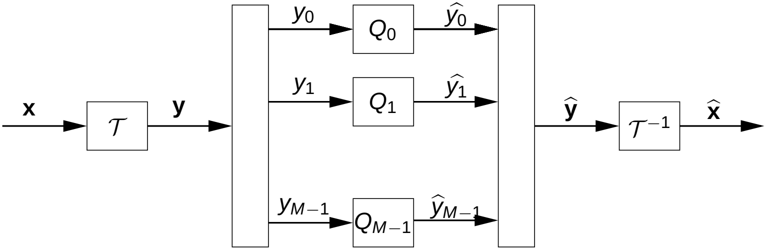
- Quantization: at the core of lossy coding
- Irreversible operation
- Centralin the rate-distortion trade-off
- Approximations in HR:
  - $D \sim 2^{-2R}$
  - $D \sim \sigma^2$
- Quantization alone is not sufficient to give good RD performance

Linear transform

- Linear transform is a base changement
  - Separation of data between important and irrelevant (energy concentration)
  - HSV relevant information
  - Resource allocation
- Correlation reduction

Linear transform

Paradigm of transform coding



From $\mathbf{x}$ to $\mathbf{y} = \mathcal{T}\mathbf{x}$: we want an “easier” vector for coding, i.e. with a few large coefficients and many small ones

Orthogonal Transformas

It means that $\mathcal{T}^{-1} = \mathcal{T}^T$

advantages :

1. immediate inversion
2. norm conservation $\|\mathbf{Y}\| = \|\mathbf{X}\|$

$\Rightarrow$ Distorsion sur **Y** = distorsion sur **X** :

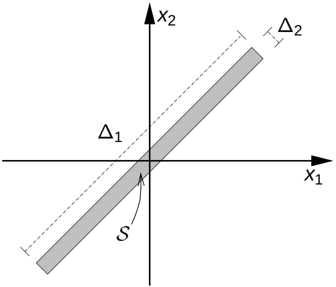
$$\mathrm{E}\left[\|\mathbf{Y}-\widehat{\mathbf{Y}}\|^2\right] = \mathrm{E}\left[(\mathbf{Y}-\widehat{\mathbf{Y}})^T(\mathbf{Y}-\widehat{\mathbf{Y}})\right] = \mathrm{E}\left[(\mathbf{X}-\widehat{\mathbf{X}})^T\mathcal{T}^T\mathcal{T}(\mathbf{X}-\widehat{\mathbf{X}})\right]$$

$$= \mathrm{E}\left[\|\mathbf{X}-\widehat{\mathbf{X}}\|^2\right]$$

Transform coding

Example

Couple of highly correlated RV's



$$f_{X_1,X_2}(x_1,x_2) = \begin{cases} \frac{1}{\Delta_1\Delta_2} & \text{si } (x_1,x_2) \in \mathcal{S} \\ 0 & \text{si } (x_1,x_2) \notin \mathcal{S} \end{cases}$$

$$\Delta_1 \gg \Delta_2$$

$$X_1 \sim X_2 \sim \mathcal{U}\left[-\frac{\Delta_1}{2\sqrt{2}}, \frac{\Delta_1}{2\sqrt{2}}\right]$$

Transform coding

Example

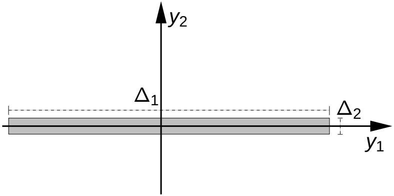
- Quantization of uniform variables $\Rightarrow$ UQ
- $D_i(R_i) = \sigma_i^2 2^{-2R_i}$ for each RV
- $D = \sum_i D_i$
- $\sigma_1^2 = \sigma_2^2 = \sigma^2 = \left(\frac{\Delta_1}{\sqrt{2}}\right)^2 \frac{1}{12} = \frac{\Delta_1^2}{24}$

bits	R	D_1	D_2	D
0	0	σ^2	σ^2	$2\sigma^2$
1	0.5	$\sigma^2/4$	σ^2	$\frac{5}{4}\sigma^2$
1	0.5	σ^2	$\sigma^2/4$	$\frac{5}{4}\sigma^2$
2	1	$\sigma^2/4$	$\sigma^2/4$	$\frac{1}{2}\sigma^2$
2	1	$\sigma^2/16$	σ^2	$\frac{17}{16}\sigma^2$
3	1.5	$\sigma^2/16$	$\sigma^2/4$	$\frac{5}{16}\sigma^2$
4	2	$\sigma^2/16$	$\sigma^2/16$	$\frac{1}{8}\sigma^2$

Transform coding

Transform: 45-degres rotation

After the transform the RV are independent



$$f_{Y_1,Y_2}(y_1,y_2) = \begin{cases} \frac{1}{\Delta_1\Delta_2} & \text{si } (y_1,y_2) \in \mathcal{S} \\ 0 & \text{si } (y_1,y_2) \notin \mathcal{S} \end{cases}$$

$$Y_1 \sim \mathcal{U}\left[-\frac{\Delta_1}{2}, \frac{\Delta_1}{2}\right]$$

$$Y_2 \sim \mathcal{U}\left[-\frac{\Delta_2}{2}, \frac{\Delta_2}{2}\right]$$

Transform coding

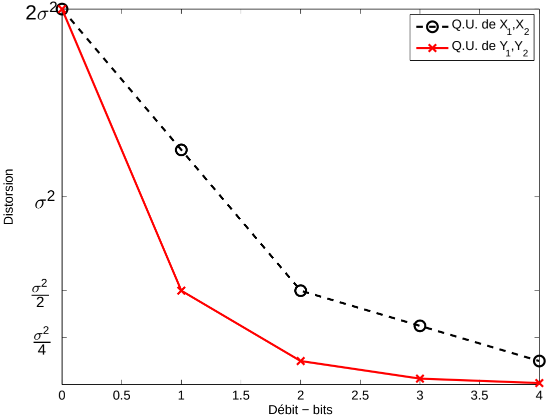
Example

- Quantization of Y_1 and Y_2 : $D(R)$ curvw
- $\sigma_1^2 = \frac{\Delta_1^2}{12} = 2\sigma^2$
- $\sigma_2^2 = \frac{\Delta_2^2}{12} \ll \sigma^2$

bits	R	D_1	D_2	D
0	0	$2\sigma^2$	$\ll \sigma^2$	$2\sigma^2$
1	0.5	$\sigma^2/2$	$\ll \sigma^2$	$\frac{1}{2}\sigma^2$
2	1	$\sigma^2/8$	$\ll \sigma^2$	$\frac{1}{8}\sigma^2$
3	1.5	$\sigma^2/32$	$\ll \sigma^2$	$\frac{1}{32}\sigma^2$

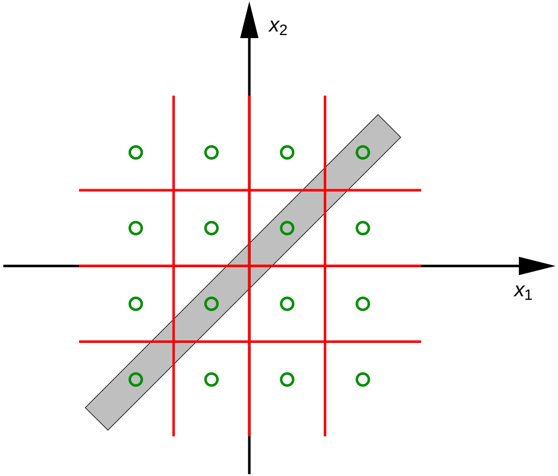
Transform coding

Performances RD: T+Q



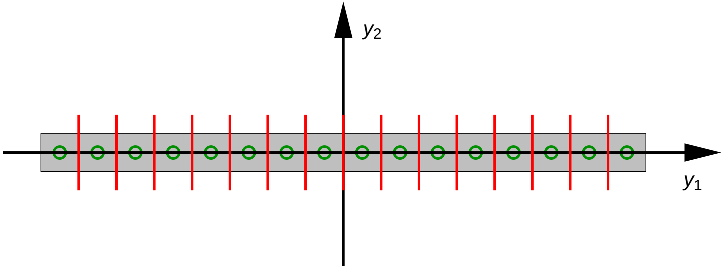
Transform coding

Performance: Q



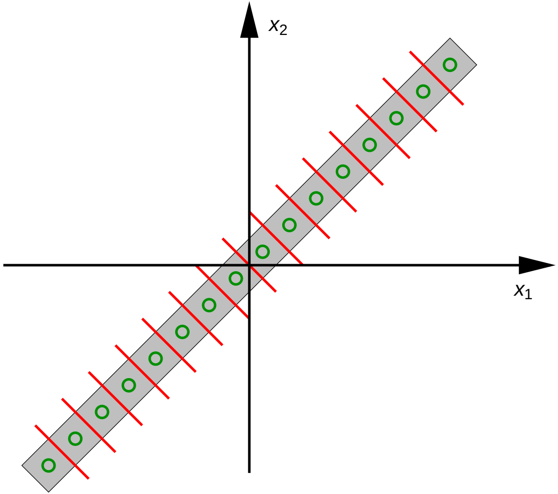
Transform coding

Performance: SQ+T



transform coding

Performance: VQ



Plan

Introduction

Transform coding
- Resource allocation
- TCD
- Discrete cosine transform

JPEG

Resource allocation

- Components of $\mathbf{y} = \mathcal{T}\mathbf{x}$ interpreted as realisations of M random staionary processes with variance σ_k^2
- Case of ortogonalt linear transform: $\mathcal{T}$ is a square matrix and the distortion is the same in the two domain (original and transformed)

$$\mathcal{D} = \mathrm{E}\left[\|\mathbf{X} - \widehat{\mathbf{X}}\|^2\right] = \mathrm{E}\left[\|\mathbf{Y} - \widehat{\mathbf{Y}}\|^2\right]$$
$$= \mathrm{E}\left[\sum_{k=0}^{M-1}(Y_k - \widehat{Y}_k)^2\right] = \sum_{k=0}^{M-1}\mathrm{E}\left[(Y_k - \widehat{Y}_k)^2\right]$$
$$= \sum_{k=0}^{M-1}\mathcal{D}_k$$

- we know the relation between $\mathcal{D}_k$ and the quantization rate b_k (HR hypothesis) :

$$\mathcal{D}_k = c_k\sigma_k^2 2^{-2b_k}$$

Resource allocation

Problem formulation

- Minimise $\mathcal{D}$ uunder a rate constraint

$$\min \mathcal{D}(\mathbf{b}) = \sum_{k=0}^{M-1} c_k \sigma_k^2 2^{-2b_k} \qquad \text{subject to} \sum_{k=0}^{M-1} b_k \leq B$$

- Lagrange method. Minimise:

$$J(\mathbf{b}, \lambda) = \sum_{k=0}^{M-1} c_k \sigma_k^2 2^{-2b_k} + \lambda \left(\sum_{k=0}^{M-1} b_k - B \right)$$

- Solution (Huang-Schulteiss formula) :

$$b_k^* = \frac{B}{M} + \frac{1}{2} \log \left[\frac{c_k \sigma_k^2}{c_{\mathrm{GM}} \sigma_{\mathrm{GM}}^2} \right]$$

Resource allocations

Interpretation

- A uniform resource allocation ($\bar{b} = B/M$) is refined using variances
- Per-component distortion:

$$\mathcal{D}_k^* = c_k \sigma_k^2 2^{-2b_k^*} \qquad = c_k \sigma_k^2 2^{-2\bar{b}} \frac{c_{\mathrm{GM}} \sigma_{GM}^2}{c_k \sigma_k^2} = c_{\mathrm{GM}} \sigma_{GM}^2 2^{-2\bar{b}}$$

$$\mathcal{D}_{\mathcal{T}} = \sum_{k=0}^{M-1} \mathcal{D}_k^* \qquad = M c_{\mathrm{GM}} \sigma_{\mathrm{GM}}^2 2^{-2\bar{b}}$$

- Gaussian case: $c_{\mathrm{GM}} = c_k = c_{\mathcal{N}}$ et donc

$$b_k^* = \bar{b} + \frac{1}{2} \log \left[\frac{\sigma_k^2}{\sigma_{GM}^2} \right] \qquad \mathcal{D}_k^* = c_{\mathcal{N}} \sigma_{\mathrm{GM}}^2 2^{-2\bar{b}}$$

$$\mathcal{D}_{\mathcal{T}} = M c_{\mathcal{N}} \sigma_{\mathrm{GM}}^2 2^{-2\bar{b}}$$

Coding gain

- Let X be a random vector of M components (sound, image...)
- Hypothesis : X components are sont i.d., e.g.m Gaussian $\mathcal{N}(0, \sigma_X^2)$
- Without transform, we have SQ and resource allocation. This case is referred to as PCM. The distortion is:

$$\mathcal{D}_{\mathrm{PCM}} = Mc_{\mathcal{N}}\sigma_X^2 2^{-2\bar{b}}$$

Coding gain

- The *Coding gain* of $\mathcal{T}$ the ratio between the PCM distortion and the transform coding distortion:

$$G_{\mathcal{T}} = \frac{\mathcal{D}_{\text{PCM}}}{\mathcal{D}_{\mathcal{T}}} = \frac{\sigma_X^2}{\sigma_{\text{GM}}^2} = \frac{\sigma_{\text{AM}}^2}{\sigma_{\text{GM}}^2}$$

- The transform should make the vector components as different as possible
- The geometrical mean of a set of positive numbers is alwayes less or equal than their arithmetical mean

Resource allocations

Practical Algorithms

Huang-Schulteiss' formula

- might provide negative values
- might provide non-integer values

Sub-optimal algorithms

- Modified HS (Huang-Schulteiss) Algorithm
- *Greedy* Algorithme

Resource allocation

Modified HS Algorithm

1. We compute b_k^* with Huang-Schulteiss ;
2. If some b_k^* are negative, we remove the variances of the concerned components and we repeat the computation. The removed variables are not coded (or coded with zero bits)
3. This step is repeated as long as there are negative allocation values.
4. The results are floored
5. The eventual residual rate is allocated to coefficients with the largest error

Greedy algorithm

1. Initialization
   - $b_k = 0 \quad \forall k \in \{0, 1, \ldots M-1\}$.
   - $D_k = \quad \sigma_k^2 \forall k \in \{0, 1, \ldots M-1\}$.
2. While $\sum_k b_k \leq B$
   - $\ell = \arg\max_k D_k$
   - $b_\ell \leftarrow b_\ell + 1$
   - $D_\ell \leftarrow D_\ell / 4$

2D linear transforms

- Asymptotically optimal when $(N, M \to \infty)$

DFT

$$t(k,\ell,n,m) = \frac{1}{\sqrt{NM}} \exp\left[-\imath 2\pi \left(\frac{(n-1)(k-1)}{N} + \frac{(m-1)(\ell-1)}{M}\right)\right]$$

DCT

$$t(k,\ell,n,m) = \frac{c_k\, c_\ell}{\sqrt{NM}} \cos\left(\pi \frac{(2n-1)(k-1)}{2N}\right) \cos\left(\pi \frac{(2m-1)(\ell-1)}{2M}\right)$$

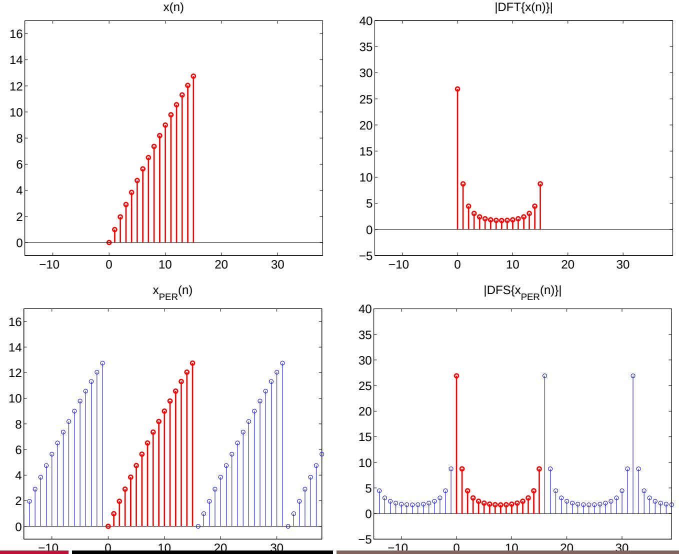
$$c_k = \begin{cases} 1 & \text{si } k = 1 \\ \sqrt{2} & \text{sinon} \end{cases}$$

2D linear transforms

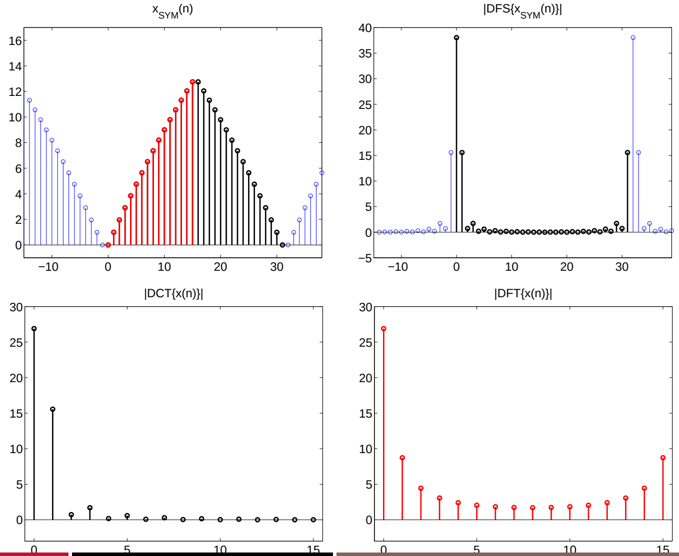
DCT and DFT

- Common characteristics
  - separable : $t(k,\ell,n,m) = t_1(k,n)\,t_2(\ell,m)$
  - fast computations
  - frequency interpretation
    k : vertical frequency
    ℓ : horizontal frequency
- DCT characteristics
  - Real
  - Better energy concentration than DFT for images

DFT vs DCT

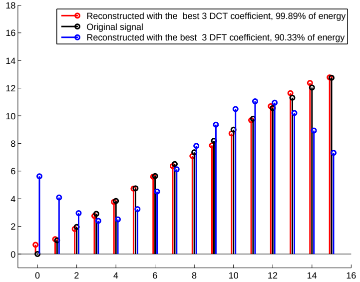

x(n)
|DFT{x(n)}|
xPER(n)
|DFS{xPER(n)}|
16
14
12
10
8
6
4
2
0
40
35
30
25
20
15
10
5
0
−5
−10
0
10
20
30

DFT vs DCT



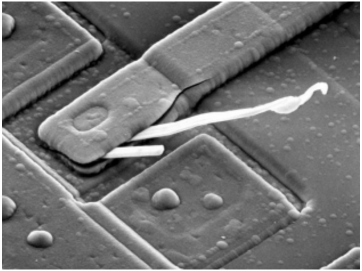
DFT vs DCT

DCT		DFT	
Coeff	% Energy	Coeff	% Energy
1	74.61	1	74.61
2	24.98	2	7.86
3	0.05	3	2.04
4	0.30	4	0.97
5	0.00	5	0.60
6	0.04	6	0.43
7	0.00	7	0.35
8	0.01	8	0.31
9	0.00	9	0.30
10	0.00	10	0.31
11	0.00	11	0.35
12	0.00	12	0.43
13	0.00	13	0.60
14	0.00	14	0.97
15	0.00	15	2.04
16	0.00	16	7.86

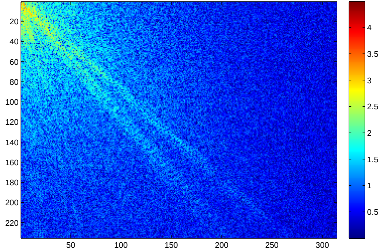


TELECOM ParisTech

DCT examples

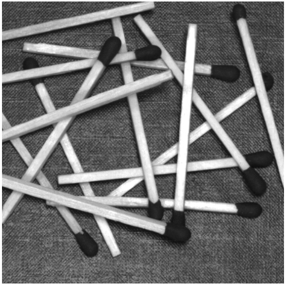


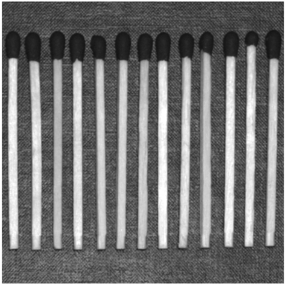
Image

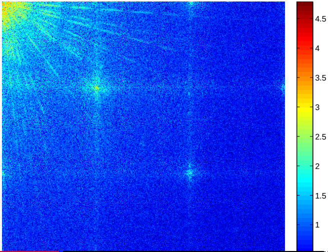


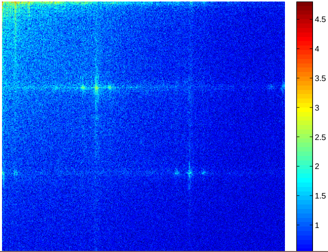
Logarithme des coefficients DCT

DCT examples



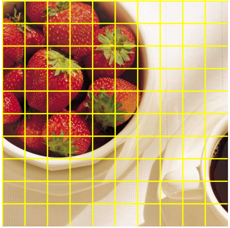


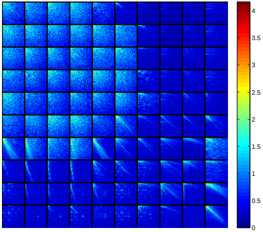




DCT par blocs

- Large-size non-stationary image
  $\Rightarrow$ divided into $I \times J$ rectangular blocks $(\mathcal{B}_{i,j})_{0 \leq i < I, 0 \leq j < J}$, of size $K \times L$ ($K = L = 8$)



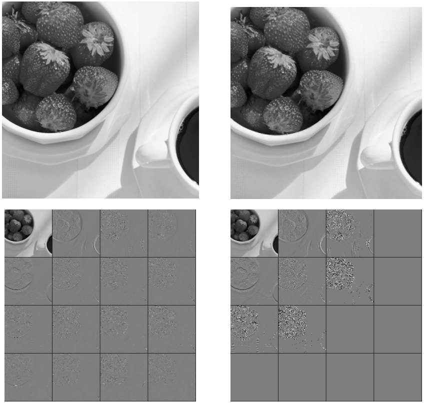


$$N = I\,K \qquad M = J\,L$$

Resource allocation for DCT coefficients

- Huang-Schulteiss formula, or
- Greedy Algorithm, or
- Fixed allocation (independent from data)
  - Very simple solution
  - Very common in practice (e.g., quantization matrix in JPEG)

Resource allocation for DCT coefficients



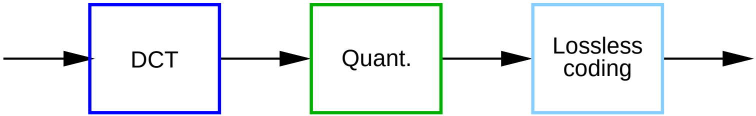
sb	variance	lev	rate	dist
1	141.35	32	3.90	0.86
2	32.92	15	2.74	0.98
3	6.80	5	1.50	1.24
4	1.77	1	0.00	1.77
5	38.07	15	2.74	1.14
6	10.27	7	1.81	1.12
7	3.89	3	1.15	1.41
8	1.19	1	0.00	1.19
9	7.47	5	1.50	1.37
10	3.76	3	1.15	1.37
11	2.07	1	0.00	2.07
12	0.71	1	0.00	0.71
13	1.95	1	0.00	1.95
14	1.27	1	0.00	1.27
15	0.74	1	0.00	0.74
16	0.32	1	0.00	0.32

Plan

JPEG Standard

- Image compression standard based on DCT
- Speicified in 1991, adopted in 1992
- It describes *only the decoder*
- It can work on multi-component (color) images, but we will describe its operation on one component, the other being identical

JPEG Standard: Scheme



- The image is decomposed in 8×8 pixels blocks
- The value 128 is subtracted from the image
- Blocks are (almost) independently encoded

JPEG Standard: Blocks

Example of 8x8 block

173	171	171	143	109	100	91	96
171	169	150	137	112	101	94	96
184	158	139	120	110	107	94	100
170	156	134	119	117	104	98	99
157	147	125	127	103	109	90	98
149	146	132	120	113	107	101	93
147	141	119	119	111	101	100	92
160	122	117	116	115	116	102	95

JPEG Standard: Transform

- DCT is performed on 8×8 blocks
- Small blocks ensures stationarity
- Large blocks better exploit correlation
- Size chosen after experiences
- Coefficients DCT : impact SVH

JPEG Standard: Transform

Coefficients DCT of the block

985.3	186.2	34.1	11.6	7.3	1.6	4.9	-8.2
40.3	47.8	5.7	-26.0	-5.3	-3.5	4.0	-1.0
6.3	4.0	-9.3	-6.7	-1.2	8.1	3.4	4.1
-0.0	4.9	-13.3	-20.8	-10.4	-1.0	-4.5	-5.1
2.1	-1.3	-1.6	0.6	3.6	3.3	8.1	-1.7
1.3	3.7	2.4	-2.7	-2.2	-3.0	-4.1	7.8
5.1	0.4	3.1	4.8	-1.4	2.5	9.8	5.3
-5.6	1.6	4.4	0.1	3.3	2.3	4.3	-8.4

JPEG Standard: Transform

Standard deviation of DCT coefficients for natural images

396.64	100.99	49.26	31.15	19.74	14.57	8.76	7.33
100.23	55.78	37.40	24.77	16.44	11.70	8.44	6.25
49.42	36.39	28.01	20.40	14.64	10.46	7.64	5.88
30.82	24.05	19.73	15.47	11.99	8.88	6.83	5.45
21.09	16.79	14.79	11.54	9.19	7.30	5.90	4.68
15.32	11.91	10.31	8.71	7.15	5.78	4.61	3.91
11.22	8.58	7.66	6.78	5.69	4.64	3.82	3.24
8.21	6.65	5.93	5.52	4.45	3.75	3.15	2.80

JPEG Standard: Quantization

- Uniform dead-zone quantization
- $\widetilde{c}_{i,j} = \left\lfloor \frac{c_{i,j}}{q_{i,j}} \right\rfloor$
- The Quantization q table defines the RD trade-off, but it is not defined by the standard
- Thus q must be encoded
- Quality factor Q

JPEG Standard: Quantization

Quantization table example

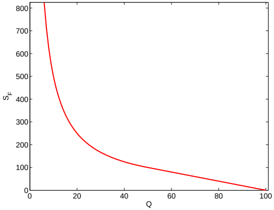
$q^* =$

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	81	61
18	22	37	56	68	109	103	77
24	35	55	64	81	104	111	90
49	63	78	87	101	121	120	100
72	92	95	98	112	100	103	99

JPEG Standard:Quality factor

- Non normative tool
- Q between 1 and 100
- Defines a scaling factor S_F for the quantization matrix

$$S_F = \begin{cases} \frac{5000}{Q} & 1 \leq Q \leq 50 \\ 200 - 2Q & 50 < Q \leq 99 \\ 1 & Q = 100 \end{cases}$$



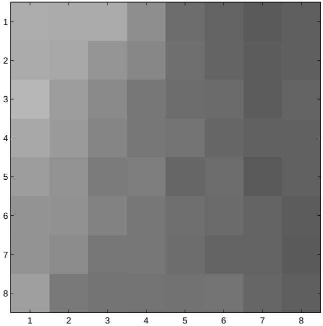
$$q \leftarrow \frac{S_F q^* - 50}{100}$$

JPEG Standard: Quantization

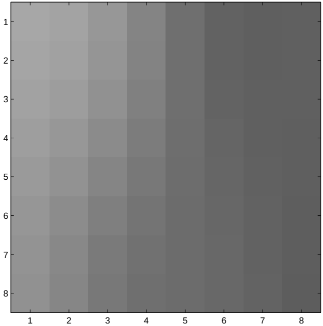
Quantized coefficients

61	16	3	0	0	0	0	0
3	3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

JPEG Standard: Example

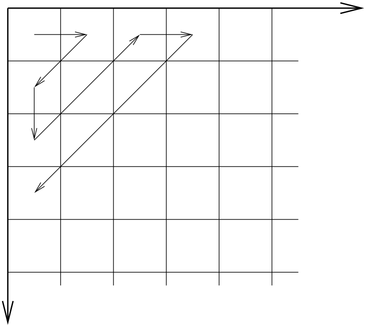


Original



$\mathrm{DCT} \rightarrow Q \rightarrow Q^{-1} \rightarrow \mathrm{DCT}^{-1}$

JPEG Standard: Entropy coding



- Zig-zag scan

TELECOM ParisTech

JPEG Standard: Entropy coding

- DC Coefficient : preidction + Huffman
- AC Coefficients : “run-lenght” + Huffman

coeff$\neq 0$	n. de 0	coeff$\neq 0$	n. de 0	...	EOB	...

JPEG Standard: Entropy coding

Encoded quantized coefficients

61	16	3	0	0	0	0	0
3	3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

61-dc_{k-1}	0,16	0,3	1,3	0,3	7,-1	EOB

JPEG Standard: Entropy coding

DC Coefficients :

- represented by the the category/amplitude couple
- There are 12 categories, named $0, \ldots, 11$, coded on 4 bit
- Category k contains 2^k values: $\{\pm 2^{k-1}, \ldots, \pm 2^k - 1\}$; each value is coded on k bits

AC Coefficients:

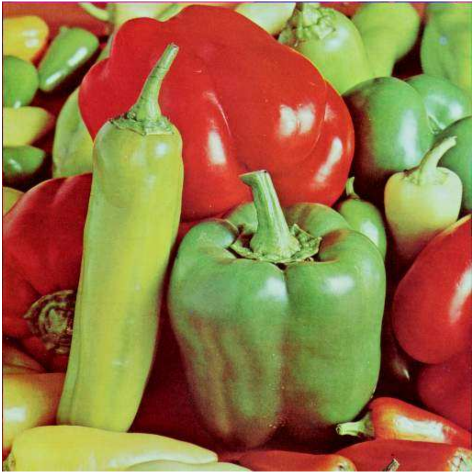
- represented by run-length, category, amplitude.
- Categories and run-lengths are coded on 4 bit each: 8 bits in total for (R,C)
  - Special symbol 1: (15,0) means "at least 15 zeros before next non-zero coefficient"
  - Special symbol 2: (0,0) means "end of block"
- As in the DC case, the category k contains 2^k values, each coded on k bits

JPEG Standard: Coding example

Image Originale, 24 bpp

JPEG Standard: Coding example

Débit 1.0198674.2 bpp   PSNR 33.92 dB   TC 23.532



JPEG Standard: Coding example

Débit 0.7481384.2 bpp PSNR 33.45 dB TC 32.080

JPEG Standard: Coding example

Débit 0.5017404.2 bpp PSNR 32.70 dB TC 47.834

JPEG Standard: Coding example

Débit 0.3081364.2 bpp PSNR 31.31 dB TC 77.888

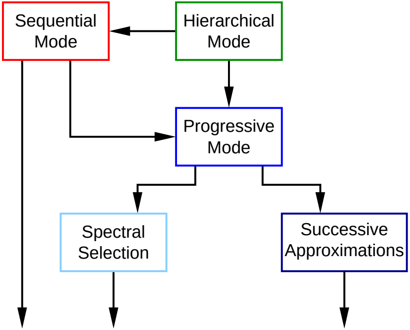
JPEG Standard: Coding example

Débit 0.2069704.2 bpp PSNR 29.50 dB TC 115.959

JPEG Standard: Parts

- JPEG baseline (sequential DCT mode)
- JPEG progressive
- JPEG hierarchical
- JPEG sequential lossless mode
- JPEG partie 3
  - Variable quantization
  - Tiling
- Standard JPEG-LS
- Motion JPEG

JPEG Modes


Sequential
Mode
Hierarchical
Mode
Progressive
Mode
Spectral
Selection
Successive
Approximations

Progressive JPEG

Spectral selection

Example with four quality layers

1. All the DC coefficients
2. The first three AC coefficients of all blocks (zig-zag scan)
3. AC coefficients 4 to 7 of all blocks
4. The other AC coefficients

JPEG syntax allow to define arbitrary layers, provided that they are made up of consecutive transform coefficients in the zig-zag scan order

The quality is the same as baseline JPEG, the rate might be slightly increased.

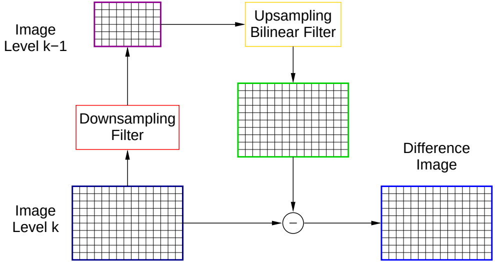
Progressive JPEG

Successive approximation

- First layer : all DC coefficients (as in Spectral Selection)
- Next layers: bit-plane coding of AC coefficients
  - An 8×8 matrix of coefficients, represented on bits $\Rightarrow$ 16 8×8 matrices of bits
  - Entropy coding of bitplanes
- Increased complexity
- RD performence slightly improved

Hierarchical JPEG

Pyramidal scheme



Each image is coded with baseline or progressive JPEG