

Basics of Image and Video Compression

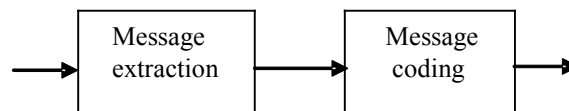
Prof. Dr. Touradj Ebrahimi



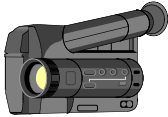
1

Compression

- All compression schemes can be decomposed into two high level modules:
 - Message extraction (symbols forming)
 - Message coding



2



For example:

Two hours of uncompressed standard definition video is equal to:

- 8'000 hours of phone conversation
- 150 CDs
- 250'000 pages of fax
- 100'000 books of 250 pages each

Compression schemes can be clustered in many ways

1) Based on the type of the data :

- Generic data
- audio
- images
- video
- 3D models
- ...

2) Based on the nature of the compression :

- **Lossless.** Decoded data is identical to the original (low compression ratios).
- **Lossy.** Decoded data is not identical to the original (higher compression ratios).



Information theory defines a theoretical framework for design of compression schemes.

- **Entropy of a source:**

It is not possible to compress a given data, losslessly, with a bitrate lower than the entropy of its source.

Measuring the entropy of a source in practice is a difficult task, mainly because of interdependencies between different symbols it can produce and non-stationarities. It is often estimated from the entropy of order zero:

$$H_0 = -\sum_{i \in S} p_i \cdot \log_2(p_i)$$



Lossless compression is often used in generic data coding

Three popular approaches for lossless compression are:

- Huffman coding
- Arithmetic coding
- Dictionary coding (Substitutional coding)



Huffman codes represent each symbol of a source by a binary code whose length is inversely proportional to the frequency (probability) of that symbol.

Mechanism : A tree is built step by step starting from the leaves to the root. At each step, the two least frequent symbols are grouped, and the sum of their probabilities assigned to that node. The code corresponding to a given symbols is then the path reaching that symbol from the root.



Example

symbol	A	B	C	D	E
frequency	15	7	6	6	5

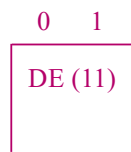


Example

A (15) B (7) C (6) D (6) E (5)



Example



A (15) B (7) C (6) D (6) E (5)



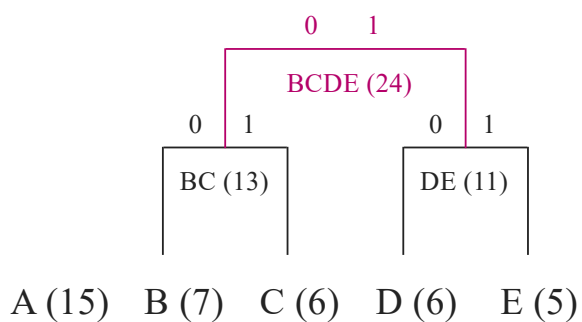
Example



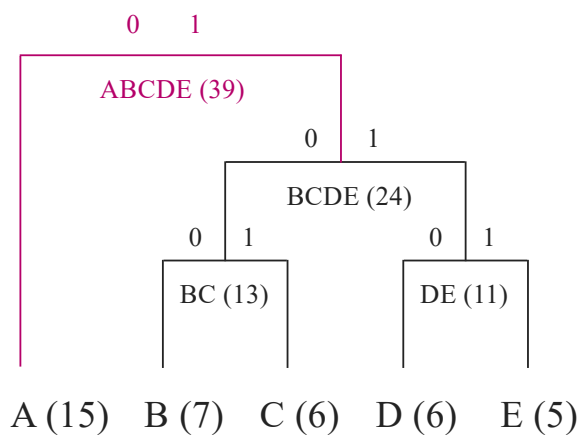
A (15) B (7) C (6) D (6) E (5)



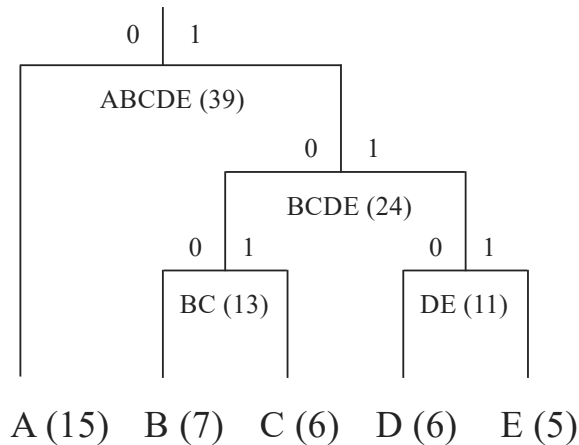
Example



Example



Example



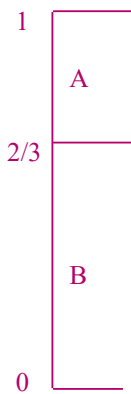
symbol	code
A	0
B	100
C	101
D	110
E	111

Arithmetic coding represents a suite of symbols by a suite of bits corresponding to an interval representing the probability of the suite.

- Always more efficient than Huffman
- Bit assignment is totally separate from modeling which makes it more suitable for an adaptive version.
- Reasonable complexity

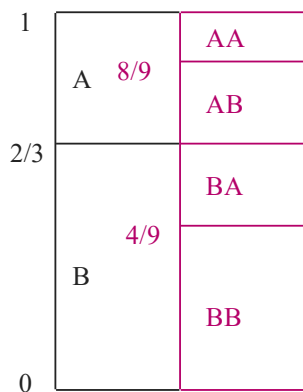
Example

$$P(A) = 1/3 \quad P(B) = 2/3$$



Example

$$P(A) = 1/3 \quad P(B) = 2/3$$



Example

$P(A) = 1/3$ $P(B) = 2/3$

1	A 8/9	AA	AAA
			AAB
2/3	B 4/9	AB	ABA
			ABB
0	BA 16/27		BAA
			BAB
	BB 8/27		BBA
			BBB

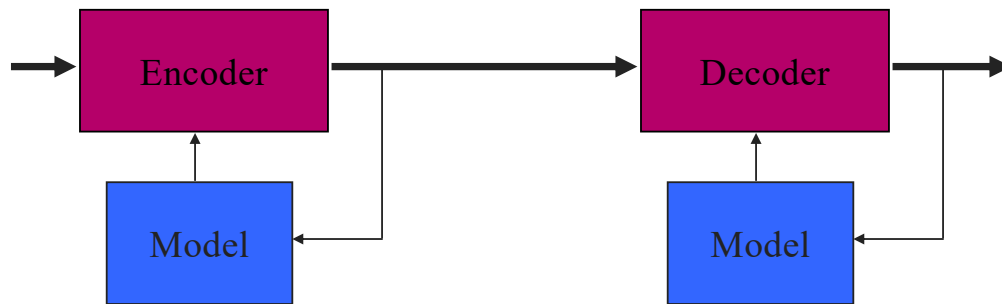


Example

$P(A) = 1/3$ $P(B) = 2/3$
segment code

1	A 8/9	AA	AAA	31/32	.11111
			AAB	15/16	.1111
2/3	B 4/9	AB	ABA	14/16	.1110
			ABB	6/8	.110
0	BA 16/27		BAA	10/16	.1010
			BAB	4/8	.100
	BB 8/27		BBA	3/8	.011
			BBB	1/4	.01





Dictionary coding creates a dictionary of words during the process of compression/decompression, and uses the index of the words in the dictionary to code more complex symbols in form of words

- Efficient compression
- Reasonable complexity

Example:

- Input (text to compress):
"HURLYBURLY"
- Output
7, 9, 11, 13, 5 (HU,RL,YB,URL,Y)

Dictionary

Symbol	Index
H	1
U	2
R	3
L	4
Y	5
B	6
HU	7
UR	8
RL	9
LY	10
YB	11
BU	12
URL	13

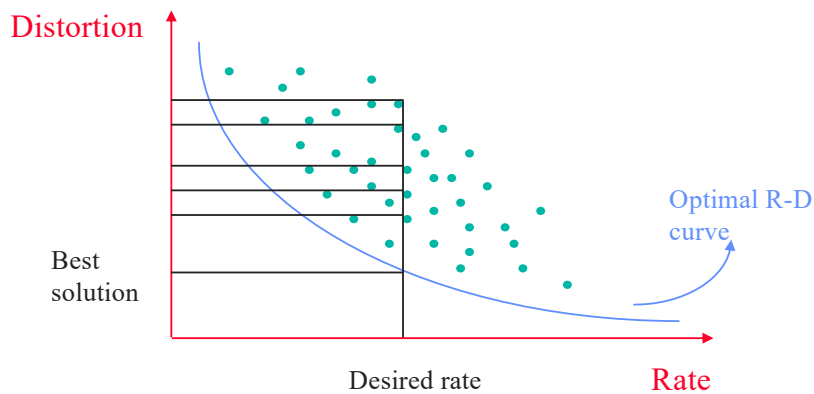
To efficiently design and analyze lossy compression schemes, rate-distortion approach is used :

- **Rate-distortion optimization:**
Find the smallest distortion for a given rate, or the smallest rate for a given distortion

The most popular approach to estimate distortion is to use the MSE (Mean Squared Error):

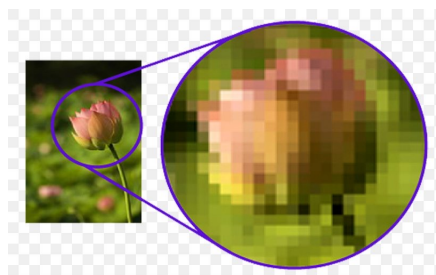
$$MSE = \frac{1}{N} \sum_{i=1}^N [x(i) - \hat{x}(i)]^2$$

MSE is not a good metric to estimate distortions perceived by HVS.

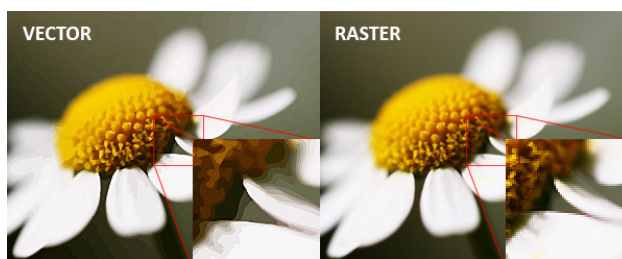
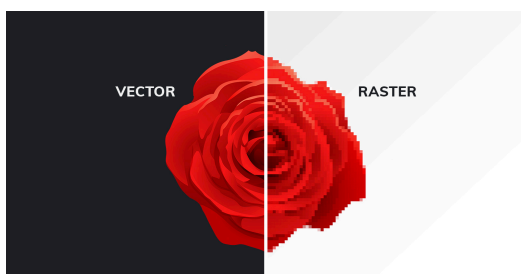
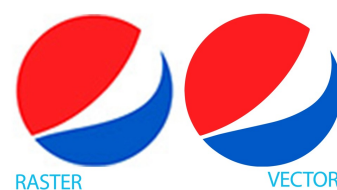
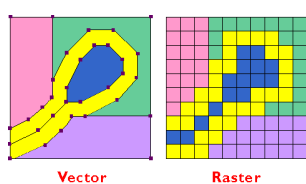


- Images are represented by symbols called **pixels**, each with
 - Unique values representing its shade/color
 - Typically R, G and B
 - But also to represent in other components from color spaces...
 - ... in different number of samples for each components
- Video is images where in addition
 - Each symbol also has a **motion** assigned to it...

- Advantages
 - Acquisition
 - Restitution
 - Simplicity (matrix representation)
- Drawbacks
 - Different from HVS
 - Not good for interactivity
 - Mature field

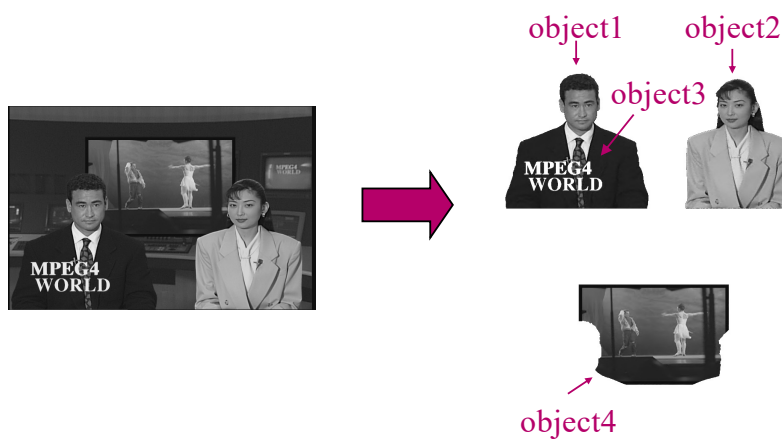


- Vector versus raster (bitmap) representation of an image



Second generation coding

29



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

29

Second generation coding

30

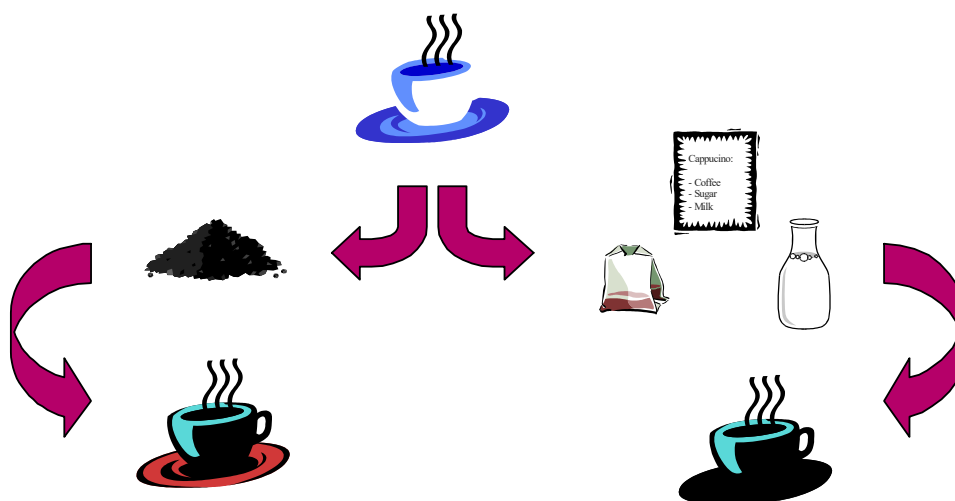


Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

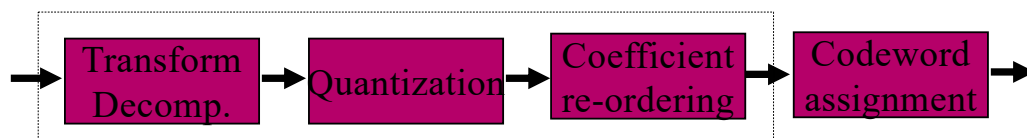
EPFL

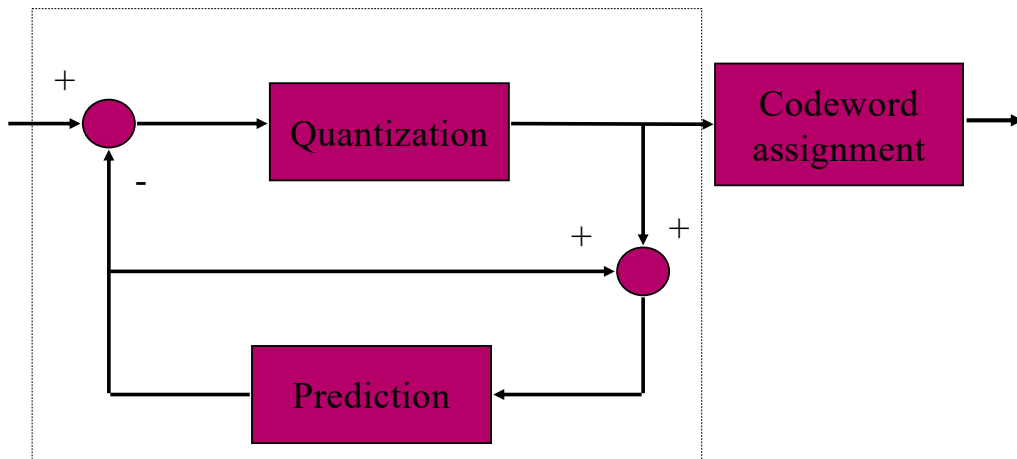
30

- Advantages
 - Similar to HVS
 - High potential for various multimedia applications
- Drawbacks
 - Problem with definition of objects (segmentation, tracking,...)
 - Needs an interface for restitution (composition)
 - Often more complex when compared to first generation schemes



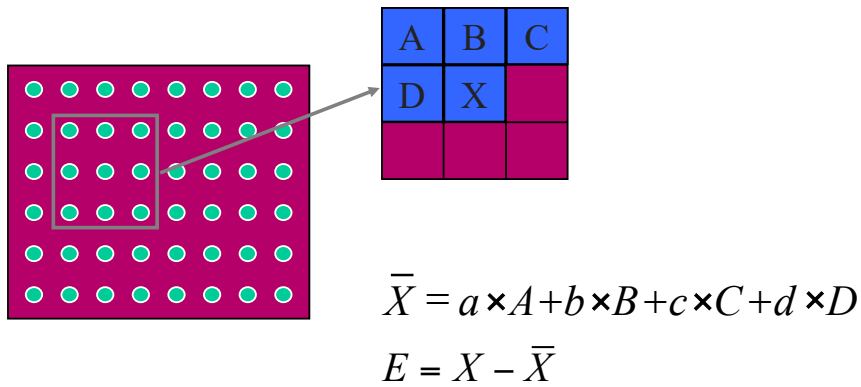
- Spatial redundancies in an image can be reduced using one of the two following approaches:
 - Transform based coding
 - Predictive coding





- Predictive coding
 - Differential Pulse Code Modulation
 - Adaptive DPCM
 - Motion compensation
- Transform based coding
 - Discrete Cosine Transform (DCT)
 - Wavelet Transform
 - Sub-band decomposition
 - Pyramidal decomposition

Linear predictor



- Frames in an image sequence are decomposed into **small size blocks**.
- For **every block in a current frame**, the most similar one in a **reference frame** is searched for.
- **Displacement between the two blocks** provides the **motion of all pixels for that block**.

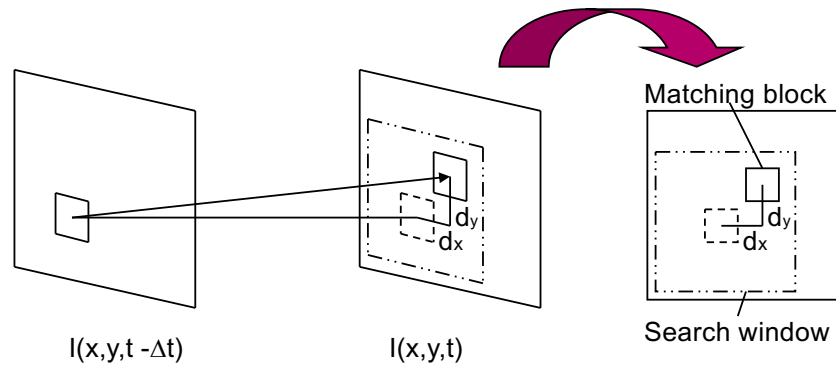


$I(x,y,t - \Delta t)$



$I(x,y,t)$

- Assumptions
 - Each block is viewed as par of a single object.
 - The motion of pixels within the same block is the same.



- Parameters
 - Search strategy: number of candidate blocks (search window), search order
 - Matching function
 - Block size

- Similarity measure or distortion

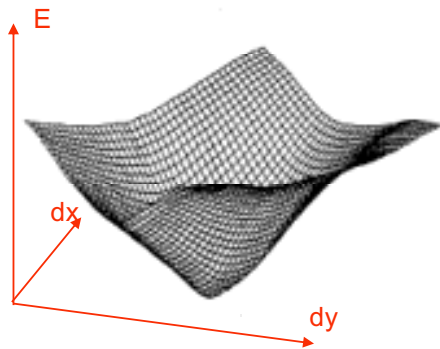
$$\min \sum_B \|I(x, y, t) - I(x - d_x, y - d_y, t - \Delta t)\|$$

- Search methods

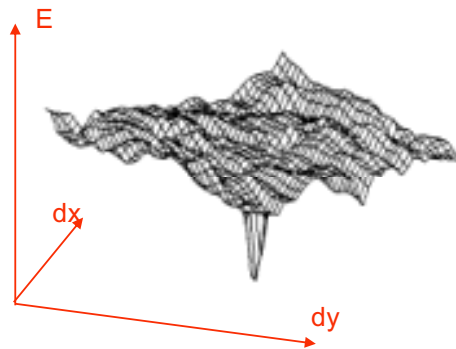
- full-search
- logarithmic
- N-step
- conjugate
- pyramidal
- multi-grid
- ...

- Difficulty of optimizing similarity measure

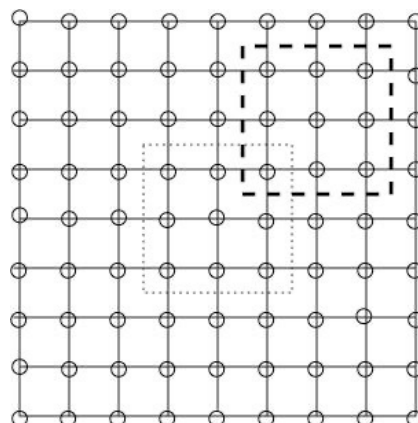
Ideal case: smooth, unimodal

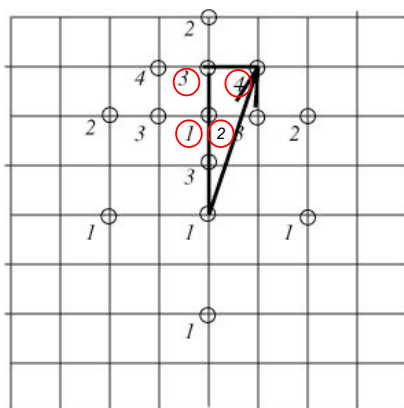


Reality: noisy, multimodal



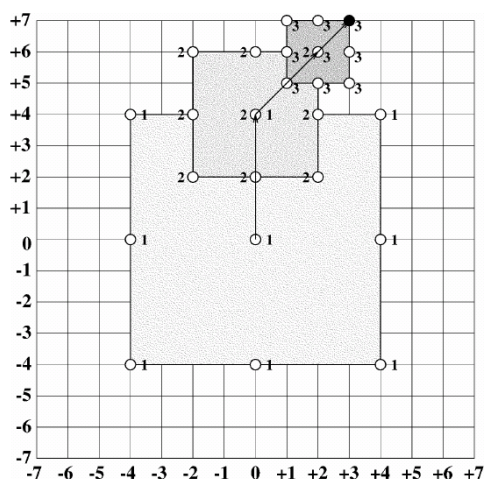
- High computational complexity
- Optimal solution





1. Chose an initial step size s and an initial starting point, (i,j) , of the search.
2. Look at the distortion that would result from choosing the blocks at coordinates (i,j) , $(i-s,j)$, $(i+s,j)$, $(i,j-s)$ and $(i,j+s)$ as prediction for the current block of interest. If the minimum distortion is at (i,j) , the step size is changed to $s/2$. Otherwise, the coordinates (i,j) is change to the coordinates that revealed the minimum distortion.
3. If s is larger than the resolution of the motion vectors then continue at Step 2. Otherwise, examine all 9 neighbors and choose the minimum distortion.

J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Transactions on Communications, vol. COM-29, pp. 1799-1808, 1981.

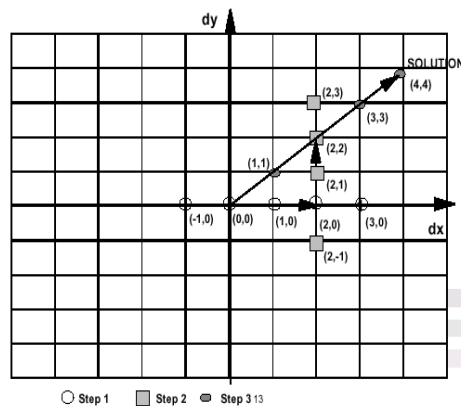


1. Initially only nine locations in the search window are used as seeds for search; they are marked as '1'.
2. After the one that yields the minimum distortion is located, the center of the new search region is moved to it and the step-size is reduced to half.
3. The new nine locations are marked as '2', and so on.

T. Koga, A. Iinuma, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proceedings of the National Telecommunications Conference, 1981, pp. G5.3.1-G5.3.5.



- The algorithm consists of 3 steps: (1) search in x direction, (2) search in y direction, and (3) search in diagonal direction.



R. Srinivasan, K. R. Rao, Predictive Coding Based on Efficient Motion Estimation, IEEE Transactions on Communications, Vol. 33, No. 8, August 1985

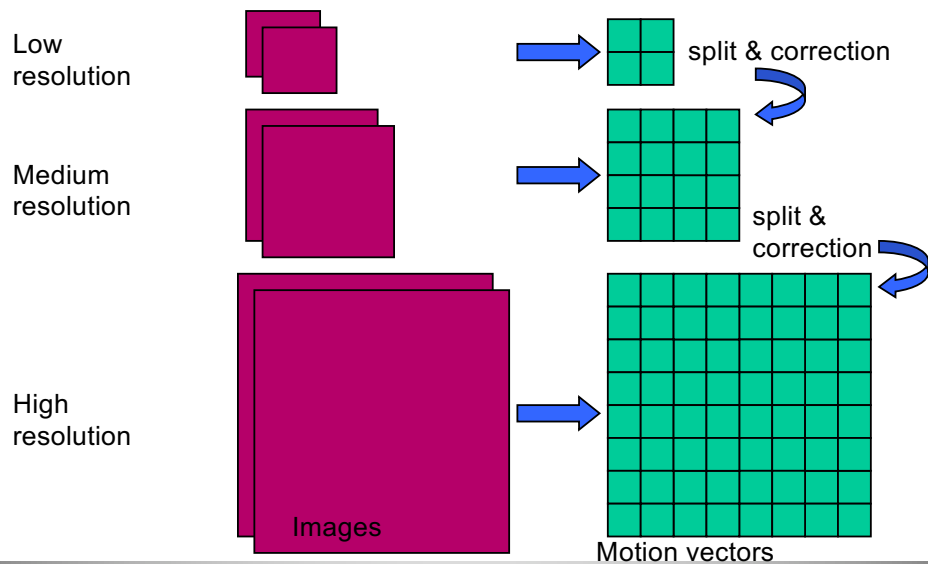


- Problems of fast search at full resolution
 - The solution may not be the global optimum.
 - Small mis-alignment may give high displacement error (especially for texture and edge blocks).
- Hierarchical (multi-resolution) block matching
 - Matching with coarse resolution to narrow down search range
 - Matching with high resolution to refine motion estimation



Block matching - Pyramidal search (2/3)

47



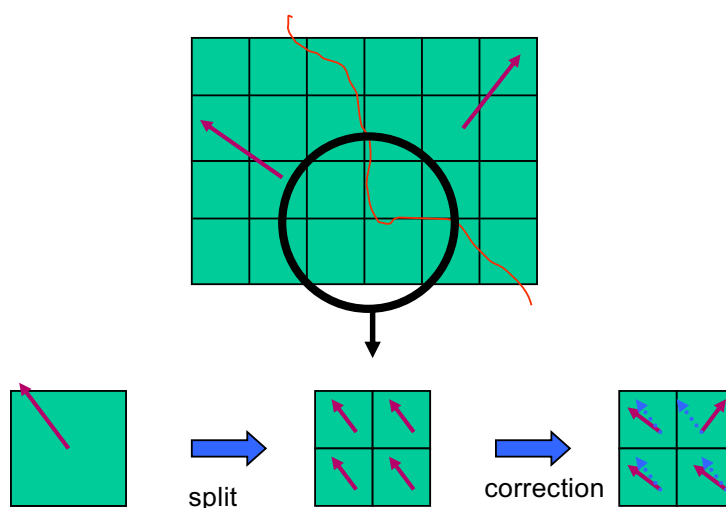
Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne



47

Block matching - Pyramidal search (3/3)

48



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne



48

- Advantages
 - Explicit minimization of prediction errors
 - Regular algorithmic structure (specially in full search): suitable for hardware implementation
 - Low overhead (one motion vector per block)
 - Robustness to noise
- Drawbacks
 - Complex
 - High prediction errors in the border regions of moving objects
 - Blocking artifact



Forward transform (1D)

$$c(0) = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} f(k)$$

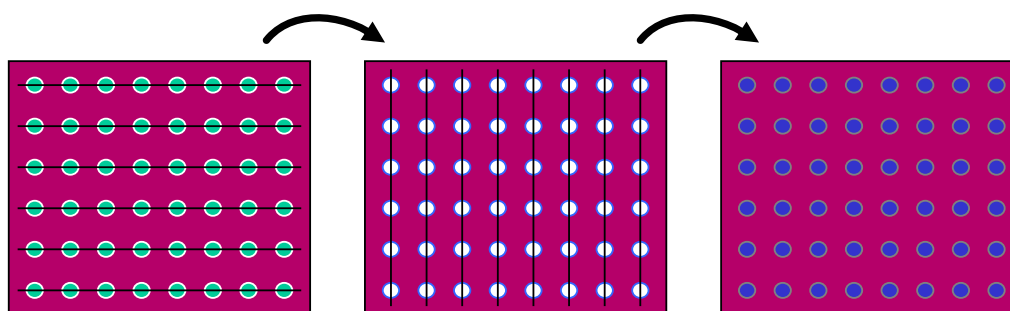
$$c(m) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} f(k) \cos \frac{(2k+1)m\pi}{2N}$$

Inverse transform (1D)

$$f(k) = \sqrt{\frac{1}{N}}c(0) + \sqrt{\frac{2}{N}} \sum_{m=1}^{N-1} c(m) \cos \frac{(2k+1)m\pi}{2N}$$

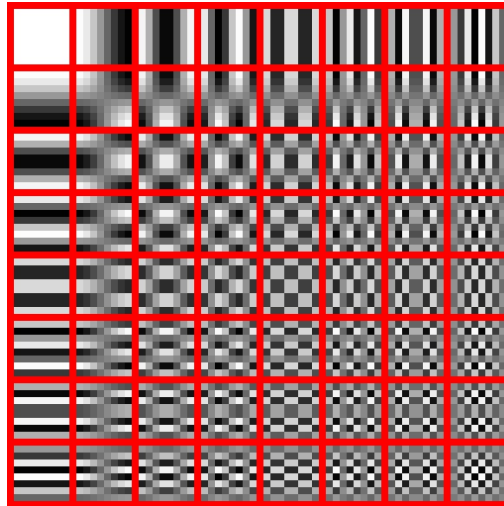
DCT is widely used in image compression thanks to its attractive Energy compaction properties and availability of fast implementation methods

Separable 2D-DCT



2D-DCT basis functions

53



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

53

Example of DCT

54

Separable 2D-DCT

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	160	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	162	163	158	158	158



236	-1	-12	-5	2	-2	-3	1
-23	-17	-6	-3	-3	0	0	-1
-11	-9	-2	2	0	-1	-1	0
-7	-2	0	1	1	0	0	0
-1	-1	1	2	0	-1	1	-1
2	0	2	0	-1	1	1	-1
-1	0	0	-1	0	2	1	-1
-3	2	-4	-2	2	1	-1	0

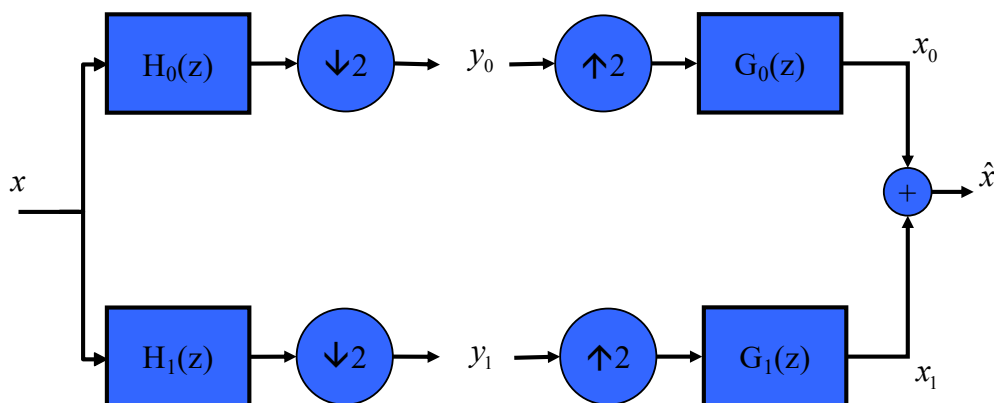


Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

54

• Two sub-band decomposition (1D)



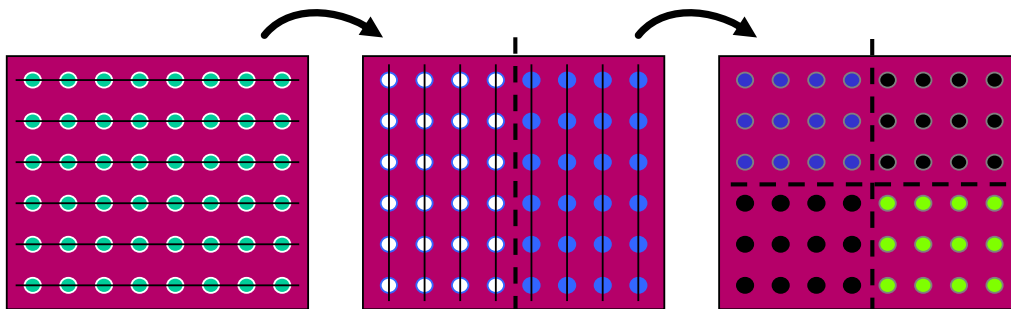
$$\begin{aligned}
 y &\rightarrow \downarrow 2 \rightarrow y' & \Rightarrow & Y'(z) = \frac{1}{2} [Y(z^{1/2}) + Y(-z^{1/2})] \\
 y' &\rightarrow \uparrow 2 \rightarrow y'' & \Rightarrow & Y''(z) = Y'(z^2) & \Rightarrow & Y''(z) = \frac{1}{2} [Y(z) + Y(-z)]
 \end{aligned}$$

$$\hat{X}(z) = \frac{1}{2} G_0(z) [H_0(z)X(z) + H_0(-z)X(-z)] + \frac{1}{2} G_1(z) [H_1(z)X(z) + H_1(-z)X(-z)]$$

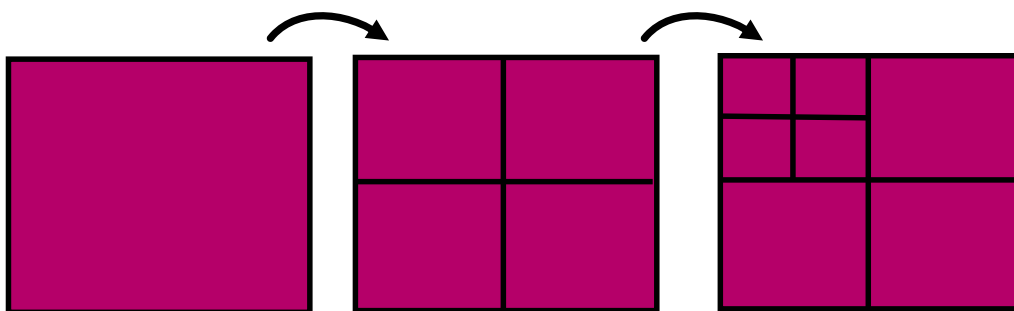
$$\hat{X}(z) = \frac{1}{2} [G_0(z)H_0(z) + G_1(z)H_1(z)]X(z) + \frac{1}{2} [G_0(z)H_0(-z) + G_1(z)H_1(-z)]X(-z)$$

$$\hat{X}(z) \equiv X(z) \Rightarrow \begin{cases} G_0(z)H_0(z) + G_1(z)H_1(z) = 2 \\ G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0 \end{cases}$$

- Separable 2D sub-band decomposition

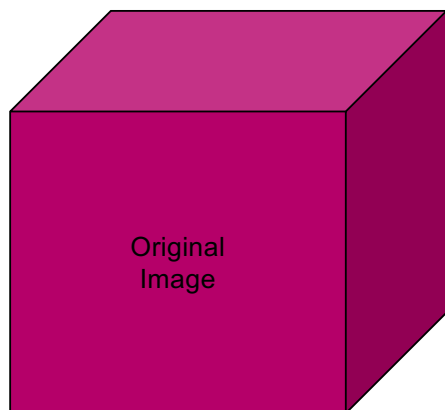


- Dyadic decomposition



Wavelet decomposition example

59



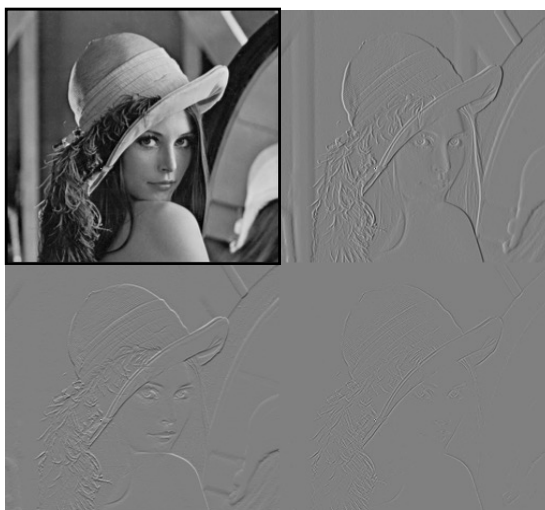
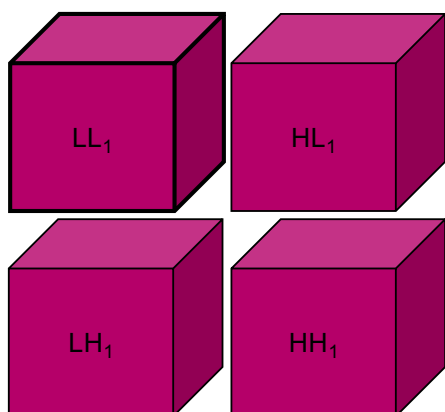
Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

59

Wavelet decomposition example

60



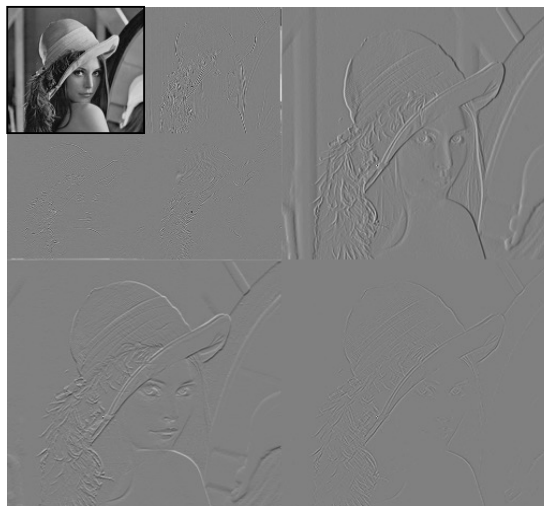
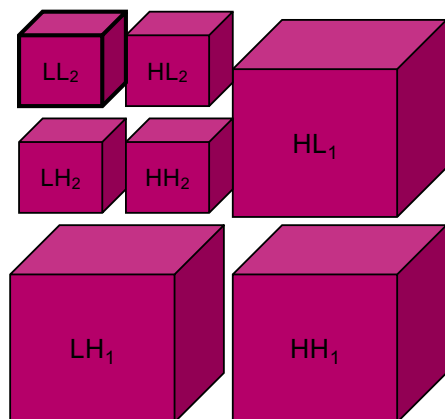
Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

60

Wavelet decomposition example

61



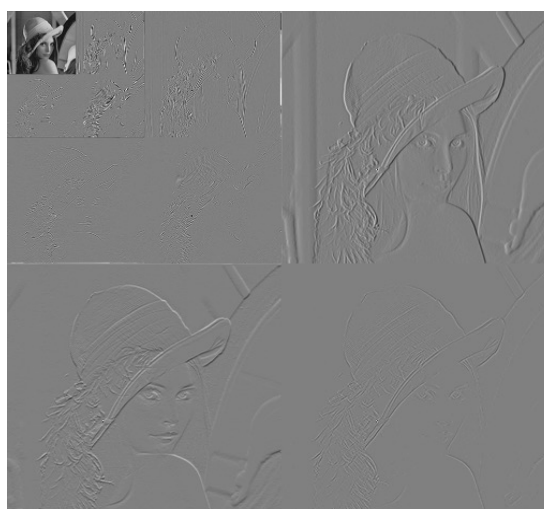
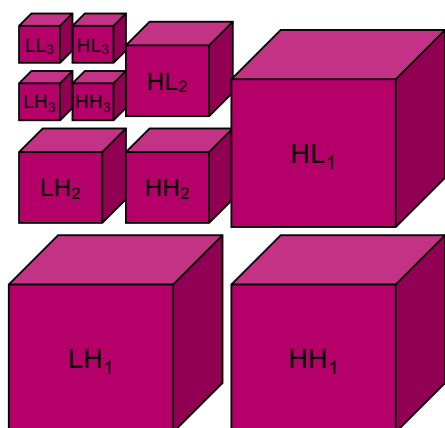
Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

61

Wavelet decomposition example

62



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

62

Quantization techniques:

- **Scalar**
 - Uniform
 - Non-uniform
 - Lloyd-Max
 - Weighted (perceptual)
- **Vector**
 - LBG
 - Adaptive
 - Lattice



Weighted quantization of DCT coefficients

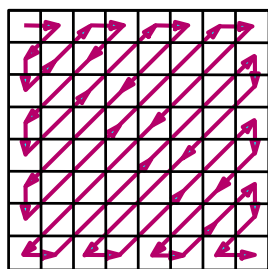
16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Luminance

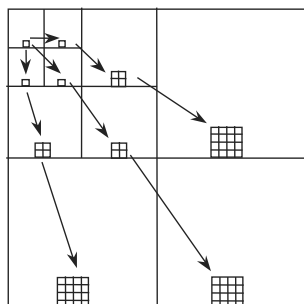
17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Chrominance





Zig-zag



Zero-tree

