Unidade Curricular

“Teoria Algorítmica da Informação”

António José Ribeiro Neves

an@ua.pt

https://www.ua.pt/pt/uc/ 13643

Outline

- Guiding questions
- Transform Coding
- Subband coding
- Wavelets
- Predictive Coding
- Audio and image predictors

Predictors, Transforms, Subbands, and Wavelets

Importance of source modeling:

- The foundation of any data compression scheme.
- Requires understanding the process or signal from multiple perspectives

Signal representations:

- **Time/Space domain**: Common representation of signals (e.g., voice as a function of time, images as functions of space).
- **Frequency domain**: Alternative representation highlighting frequency components of the signal.

Relationship between representations:

- Time and frequency views are different representations of the same signal.
- The transition between these representations is possible through mathematical transformations.

Guiding questions

What is the purpose of applying transforms like the Discrete Cosine Transform (DCT) in image, video, and audio compression?

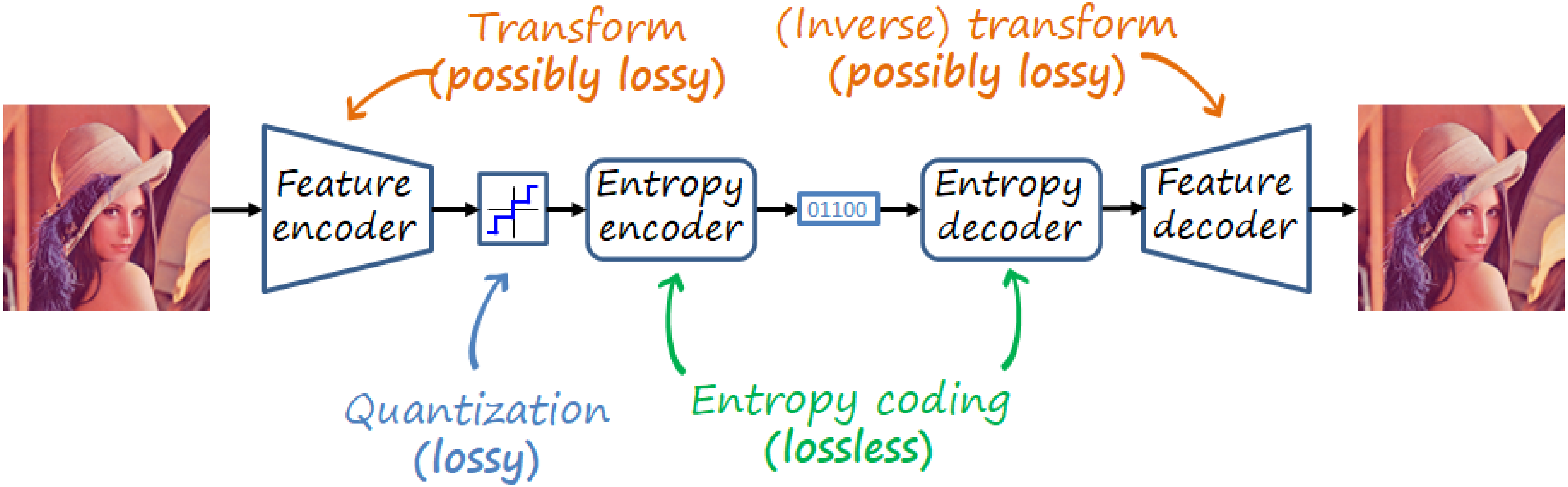
What is subband coding, and how is it applied in audio compression standards like MP3?

How do wavelets differ from DCT in representing audio or image data?

TRANSFORM CODING

- Transform coding uses transforms to represent signals, often in the frequency domain
- Human perception is limited to narrow frequency bands:
  - Spatial frequencies for images.
  - Temporal frequencies for audio.
- Transforms reveal correlation structures in the signal.
- Compact most of the signal energy into a few components, facilitating compression by ignoring low-energy parts.
- Efficient bit allocation ensures focus on the most critical coefficients.
- Transform Coding Process:
  - Transform: Convert data into the transform domain (e.g., frequency domain).
  - Quantization: Reduce precision of transform coefficients to save space.
  - Coding: Encode quantized coefficients efficiently.

TRANSFORM CODING



https://www.lherranz.org/2022/08/24/neural-image-compression-in-a-nutshell-part-1-main-idea/

Transforms of interest

Discrete Fourier transform (DFT)

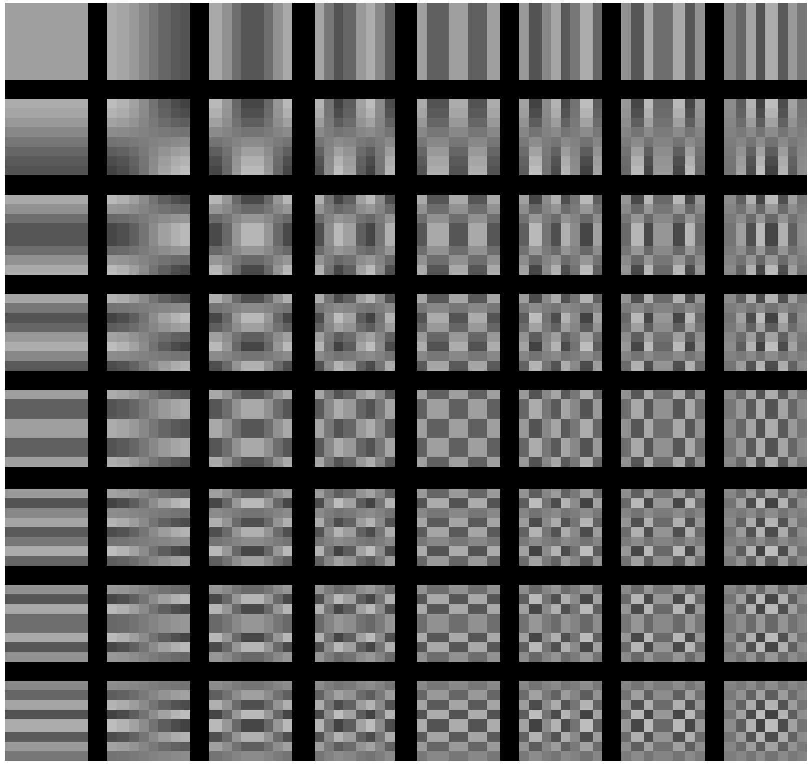
Karhunen–Loéve transform

Discrete cosine transform (DCT)

Modified discrete cosine transform (MDCT) - overlap transforms

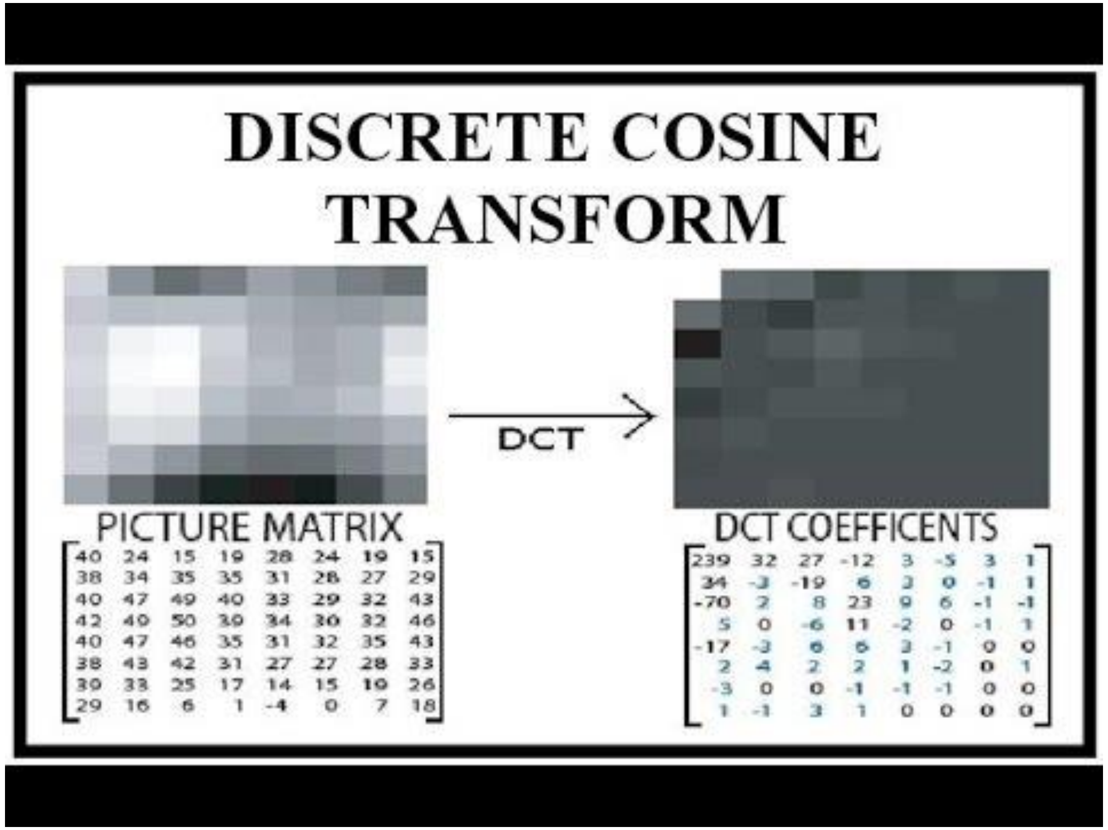
https://www.youtube.com/watch?v=0Kmg1BT9Wxc
https://www.mathworks.com/help/audio/ref/mdct.html

Example: DCT



https://www.mathworks.com/help/images/discrete-cosine-transform.html

https://www.youtube.com/watch?v=Q2aEzeMDHMA



SUBBAND CODING

- Working with signals in the frequency domain allowed us to use their spectral structure and develop transform coding approaches.
- Combines temporal/spatial and spectral domains for signal processing.
- Decomposes signals into multiple frequency bands, enabling tailored encoding for each band.
- Low-Frequency Bands:
  - Signals tend to be smooth.
  - Exploits sample-to-sample correlation for efficient encoding.
- High-Frequency Bands:
  - Signals are sparse.
  - Enables specialized encoding schemes suited to sparse data.
- Leverages human perception's spectral and temporal limitations - allows selective ignoring or discarding of less important components to improve compression.
- Uses band-specific encoding to maximize efficiency:
  - Coarser quantization for less perceptually critical components.
  - Finer quantization for bands where errors are more noticeable.

Example

Analysis

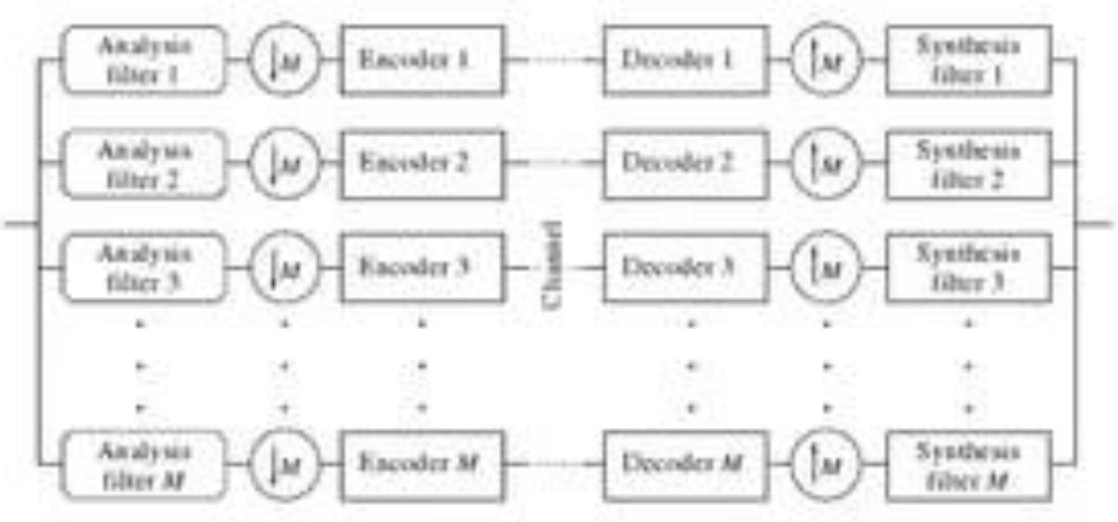
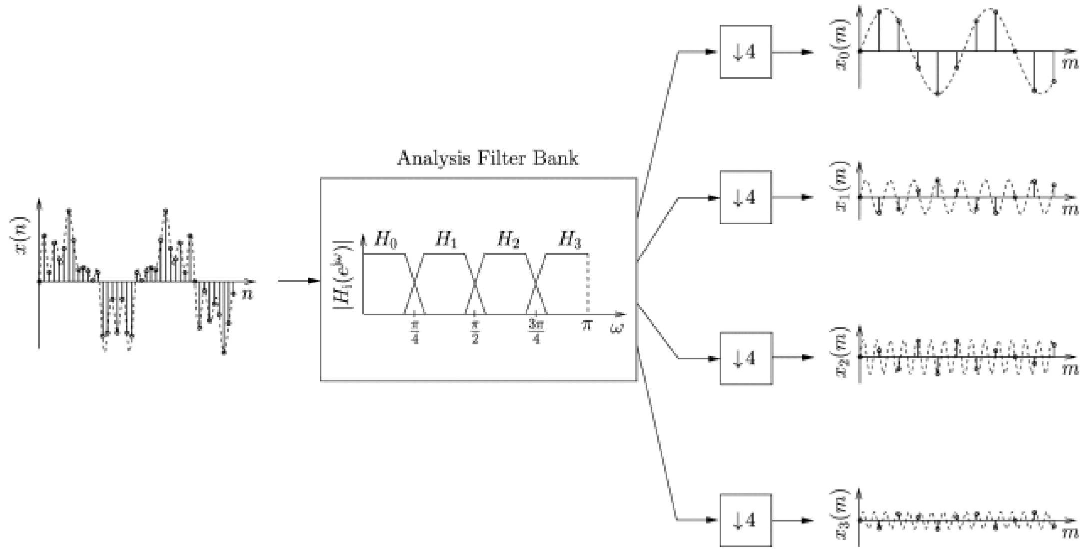


Fig-1. Block diagram of subband coding system

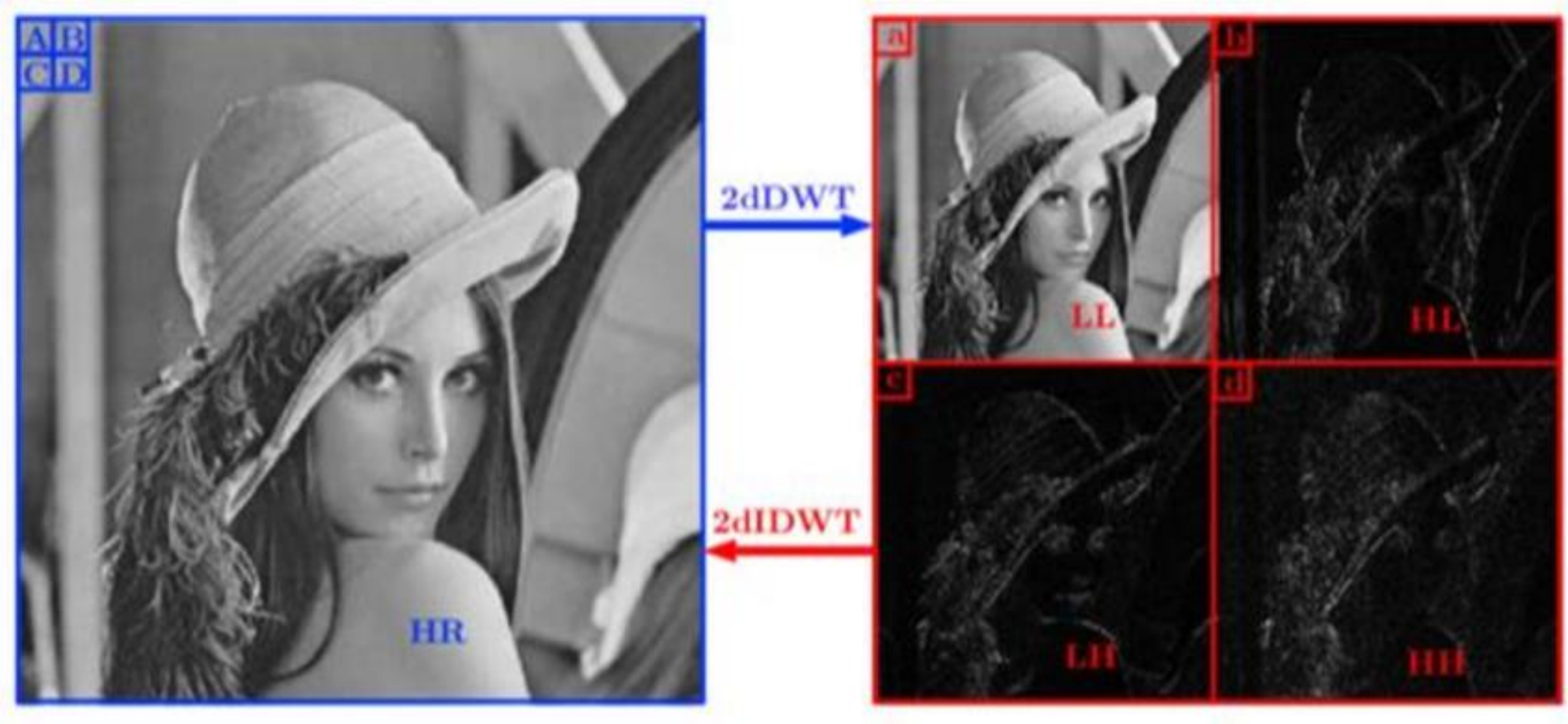


https://www.sciencedirect.com/topics/computer-science/synthesis-filter-bank

WAVELETS

- Wavelets enable time-frequency decomposition, capturing both temporal and spectral structures
  - Decomposes a signal into coarse and fine details at different scales.
  - Allows precise representation of transient signals and localized features.
- Useful for signals with non-stationary characteristics, where properties vary across time or space.
- Transform Coding: Assumes stationarity; averages over the entire signal, losing temporal resolution.
- Subband Coding: Separates frequency bands but lacks fine-grained temporal localization.
- Wavelet Transform: Provides a multiresolution analysis, adapting to both time and frequency variations.
  - Signal decomposition using analysis filter banks.
  - Downsampling, quantization, and encoding of filter outputs.
  - Decoding involves upsampling and synthesis filter banks to reconstruct the signal.

WAVELETS



https://www.siue.edu/~msong/Research/article.pdf

https://eeweb.engineering.nyu.edu/~yao/EL5123/lecture11_wavelet_JPG2K.pdf

Guiding questions

What is predictive coding, and how does it work?

How does predictive coding help in audio compression?

How is predictive coding applied in image compression?

How is predictive coding used in video compression (intra- vs. inter-frame prediction)?

What are some challenges in predictive coding?

How does quantization affect predictive coding in lossy compression?

Predictive coding

- Let $x^n = x_1 x_2 \dots x_n$ be the sequence of values (scalars or vectors) produced by an information source until time n.
- Predictive coding is based on encoding sequence $r^n = r_1 r_2 \dots r_n$, instead of the original sequence x^n, where

$$r_n = x_n - \hat{x}_n$$

and

$$\hat{x}_n = p(x^{n-1}) = p(x_1 x_2 \dots x_{n-1})$$

- The $\hat{x}_n$ are the estimates and the values of the sequence r^n are the residuals.
- Function $p()$ is the estimator or predictor.
- The aim of predictive coding is to have $H(r^n) < H(x^n)$.

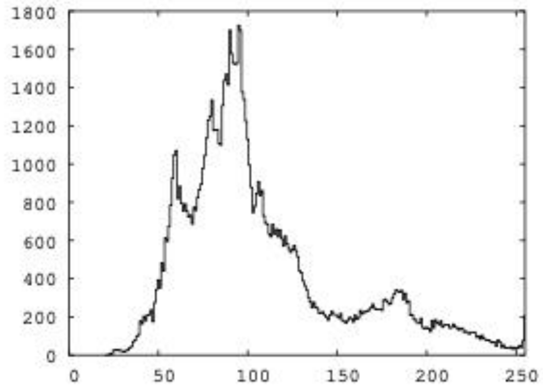
Examples

- Simple polynomial predictors used in some audio encoders:

$$
\begin{cases}
\hat{x}_n^{(0)} = 0 \\
\hat{x}_n^{(1)} = x_{n-1} \\
\hat{x}_n^{(2)} = 2x_{n-1} - x_{n-2} \\
\hat{x}_n^{(3)} = 3x_{n-1} - 3x_{n-2} + x_{n-3}
\end{cases}
$$

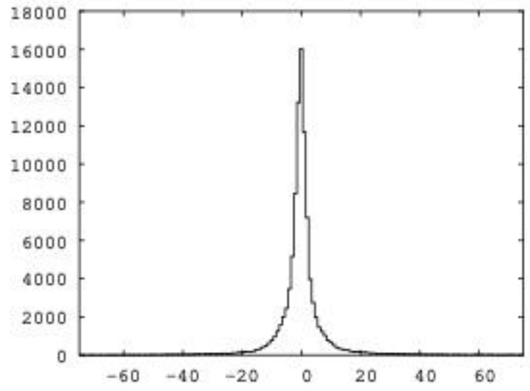
and the corresponding residuals, computed efficiently:

$$
\begin{cases}
\hat{r}_n^{(0)} = x_n \\
\hat{r}_n^{(1)} = r_n^{(0)} - r_{n-1}^{(0)} \\
\hat{r}_n^{(2)} = r_n^{(1)} - r_{n-1}^{(1)} \\
\hat{r}_n^{(3)} = r_n^{(2)} - r_{n-1}^{(2)}
\end{cases}
$$



Original

$H = 7.26$ bits/symbol



Predictor 1 JPEG

$H = 4.49$ bits/symbol

Predictive coding

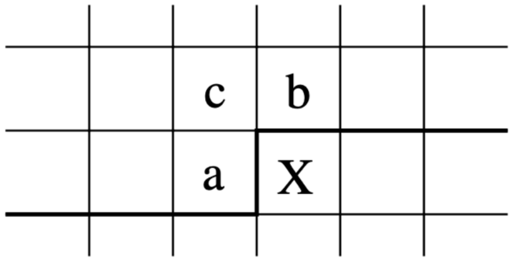
- Good predictors are fundamental in predictive coding.
- For efficient encoding, the estimated values should be as close as possible to the real values, i.e., the r_k values should be small.
- The decoder must be able to generate the same sequence, $\hat{x}^n$, of estimated values.
- In other words, the predictor cannot introduce any error during encoding / decoding.
- Therefore, the predictor must be causal, and, in lossy coding, the predictor at the encoder must use the reconstructed values, $\tilde{x}^{n-1}$, instead of the original values, x^{n-1}.

Predictive coding

- One of the main advantages of predictive coding is allowing a simple design of lossless encoders.
- In fact, most of the lossless encoders for audio and image rely on predictive coding techniques.
- However, for lossless coding, there is an additional constraint regarding the predictor: the estimates generated must be platform independent.
- Generally, this constraint implies that the predictor can use only integer arithmetic.

Predictive coding

- The lossless mode of JPEG (ISO/IEC 10918-1, ITU-T T.81, 1992) provides seven linear predictors:

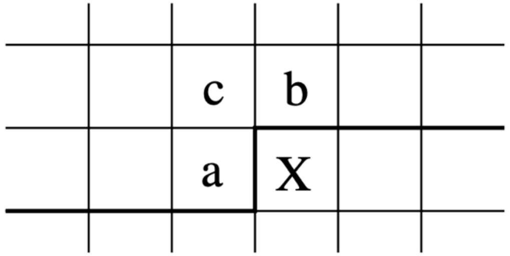


Mode	Predictor
1	a
2	b
3	c
4	$a+b-c$
5	$a+(b-c)/2$
6	$b+(a-c)/2$
7	$(a+b)/2$

- Generally, the performance of the several predictors may vary considerably from image to image.
- If encoding time is not a problem, then all of them can be tested and the one with the best compression rate chosen.

Predictive coding

- JPEG-LS (ISO/IEC 14495-1, ITU-T T.87, 1999) uses a predictor based on the same spatial configuration as that of JPEG:



- However, instead of a linear predictor, it uses the nonlinear predictor

$$\hat{x} = \begin{cases} \min(a, b) & \text{if} \quad c \geq \max(a, b) \\ \max(a, b) & \text{if} \quad c \leq \min(a, b) \\ a + b - c & \text{otherwise} \end{cases}$$

- Note that the linear part of this predictor ($a + b - c$) is the same as predictor number 4 of JPEG.

Predictive coding

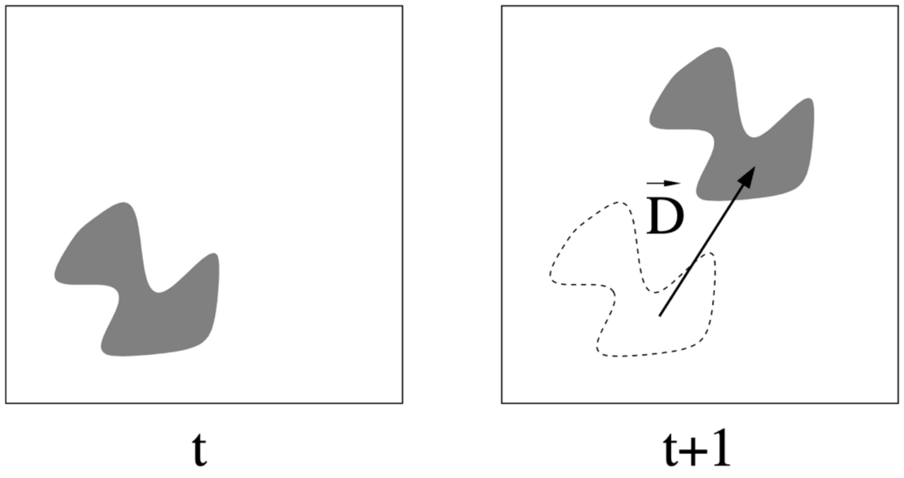
(a)

(b)

Predictor	1	2	3	4	5	6	7	JLS
Entropy (a)	4.49	4.21	4.74	4.18	4.16	4.04	4.10	**3.98**
Entropy (b)	5.60	5.05	5.82	5.19	5.23	4.97	5.15	**4.93**

Motion compensation

- Typically, the differences between one frame and the previous frame of a video sequence are due to the motion of the several elements of the scene.
- Exceptions occur when there are scene changes, zoom-in / zoom-out operations and camera translation.



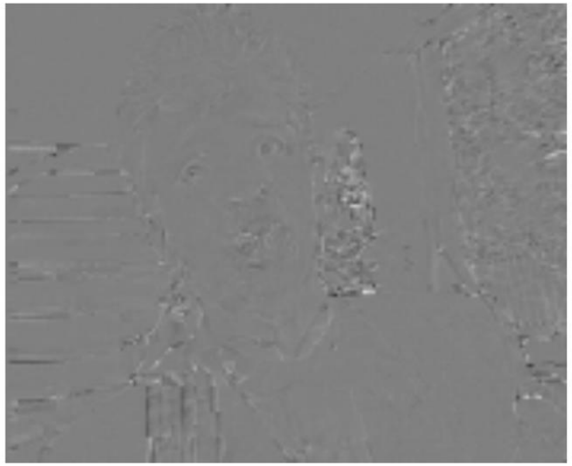
- To exploit this redundancy, it is frequent to use temporal prediction (interframe compression), which relies on motion compensation.

Example

Frame 200

Frame 201

Direct difference
$H = 5.23$ bpp



Motion compensation
$H = 4.38$ bpp