TDs Master SSTIM

YEAR 2016-2017

"SYSTÉMES DE COMPRESSION : DE L'IMAGE FIXE AUX NOUVEAUX MÉDIAS 3D"

Professor: Marc Antonini (am@i3s.unice.fr)

Lab assistant: Effrosyni Doutsi (doutsi@i3s.unice.fr)

Download the following grayscale images (8bits/pixel) : "baboon.jpg", "cameraman.jpg" and "lena.jpg".

INTRODUCTION IN CODING PRINCIPLE

The conventional principle in coding is illustrated in Figure 1. A compression method could be **lossy**, meaning that there are some data of the original signal which cannot be restored and this probably is going to effect the quality of the reconstructed signal, or **lossless**, meaning that there is no loss of information. A lossy method is necessary when there is redundancy of information i.e a high resolution image includes data which are not distinguished by the human visual system. As a result, when this image needs to be sent or stored it is efficient to eliminate this set of coefficients.

The input signal is first **transformed** in order to be more compressible. This is a necessary step to extract hidden information of the input signal which are not represented by the amplitude value of its coefficients. There are different kind of transformations like discrete Fourier transform (DFT), Discrete Cosine Transform (DCT), wavelets, Gaussian filters, Gabor filters, Laplacian Pyramids, etc. The process of reducing the number of coefficients is called **quantization**. The transform domain is convenient to decide which are the coefficients which are less involved and eliminate them. Last but not least, the quantized signal is sequenced and losslessly packed into the output bit-stream (**entropy coding**).

The decoding chain consists of the reverse steps, expect of the quantization process which is irreversible.

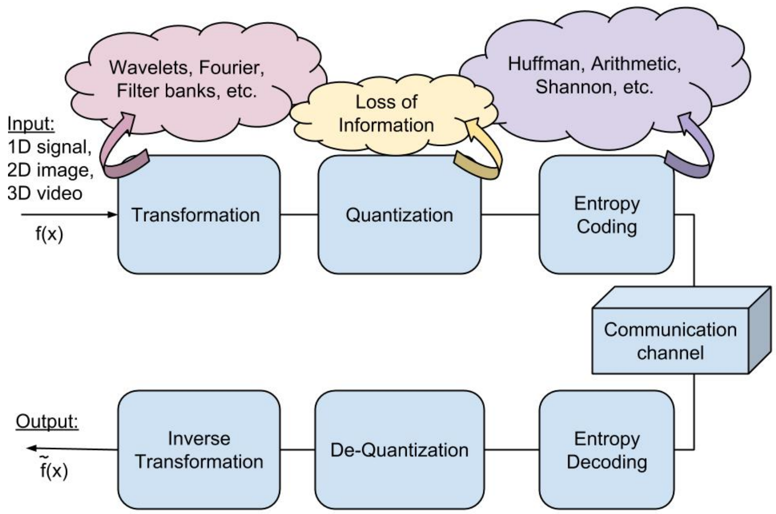

FIGURE 1 – Coding Principle.

TD 1. Transformation

— Perform a function of a wavelet transform (analysis/synthesis) using the **Haar filter**.
— Apply the function to “baboon.jpg". Explain.
— Perform a **Laplacian Pyramid** transform (analysis/synthesis).
— Apply the function to “cameraman.jpg". Explain.

1.1 Haar Wavelet

The Haar filter was proposed in 1909 by Alfréd Haar. It is the simplest possible wavelet transform. Haar filter relies on averaging and differencing values of a matrix making a bid for producing another sparse or nearly sparse matrix. A sparse matrix is a matrix in which a large portion of its entries are 0. A sparse matrix can be stored in an efficient manner, leading to smaller file sizes.

Example 1D signal analysis :

1. Input signal :

$$A = [88\ 88\ 89\ 90\ 92\ 94\ 96\ 97]$$

2. Group the coefficients :

$$A_g = [88\ 88]\ [89\ 90]\ [92\ 94]\ [96\ 97]$$

3. Compute the average of each group
4. Compute the difference of each group
5. Store in a new vector :

$$A_h = [88\ 89.5\ 93\ 96.5\ 0\ \ -0.5\ \ -1\ \ -0.5]$$

For a multiscale decomposition we iterate steps (a)-(c) using as an input the averaged values.

The above process could be implemented as a convolution between the input signal and a set of lowpass and highpass filters which result in averaging and differencing the values of each group.

For a 2D signal we apply the low-pass and high-pass filter to each row and column of the input image.

Example 1D signal synthesis :

1. Input signal :

$$A_h = [88\ 89.5\ 93\ 96.5\ 0\ \ -0.5\ \ -1\ \ -0.5]$$

2. Upsample the average values and use the differences to reconstruct :

$$A_h = \left[\underbrace{88\ 88}_{\mp 0}\ \underbrace{89.5\ 89.5}_{\mp 0.5}\ \underbrace{93\ 93}_{\mp 1}\ \underbrace{96.5\ 96.5}_{\mp 0.5}\right]$$

3. Output signal identical to the original one :

$$\tilde{A} = [88\ 88\ 89\ 90\ 92\ 94\ 96\ 97] = A$$

1.2 Laplacian Pyramid

The Laplacian pyramid was introduced by Burt and Adelson in 1983. It is very similar to Gaussian pyramid but more efficient from the compression point of view.

Example 2D signal analysis :

— Build a 2D Gaussian Filter $g(x, y)$

— Convolve "lena.jpg" with the Gaussian filter
— Downsample (↓) the rows and columns of the filtered image
— Convolve the filtered image with the Gaussian filter
— Iterate steps (c)-(d) in order to build 5 decomposition layers
— Compute the Laplacian pyramid by subtracting every two sequential layers (upsample the downsampled scales)
— Construct 4 Laplacian decomposition layers $L_1.L_2, L_3, L_4$.

Example 2D signal synthesis :

1. Upsample the last scale which consists of the smoothed version of the input signal L_4
2. Add the L_3
3. Iterate the above process until you get the $f_1(x, y)$ filtered image.

TD 2. Quantization

— Construct a uniform **scalar quantizer** with $L = 2^R$ quantization levels.
— Plot the input/output characteristic function of this quantizer.
— For all the possible values of R quantize the "lena.jpg" image.
— Measure the **distortion** D computing the **Mean Square Error (MSE)**. Explain.
— Plot a graph $D(R)$ with $R = log_2 L$ bits/sample. Explain.

TD 3. Image Codec

— Read the "lena.jpg" image.
— Use Haar filters to obtain 2 decomposition levels.
— Quantize each subband choosing a different quantization step (do not quantize the low frequency image). Explain.
— Calculate the entropy in each subband, and the total entropy to obtain the compression ratio for each configuration of the quantization step.
— Compute the Huffman entropy.
— Compare the two entropies. Explain.
— Compress and reconstruct "lena.jpg" and measure the distortion D using **Peak Signal-to-Noise Ratio (PSNR)** which is defined in

the following way :

$$PSNR_{dB} = 10log_{10}\frac{255^2}{MSE}.$$

TD 4 - Introduction in Motion Estimation

Motion estimation is a process which is used in video compression in order to reduce **temporal redundancy** by determining the motion vectors. It is an ill-possed process since the motion could have 3 dimensions but the images and their motion vectors should be defined from a 3D world to a 2D.

Motion vectors could be related to the whole image or just a small part of the image like rectangular blocks or even pixels. They can approximate the motion of the camera or the motion of an object inside the scene.

Motion estimation could be :

— **Direct :** Related to the pixels.

— **Indirect :** Related to different shapes like contours, etc.

4.1 Block-Matching Algorithm

— Use a bunch of adjacent frames of a video (i.e "xylophone.mp4" or you can visit the following link to download other videos : http ://calvin.inf.ed.ac.uk/datasets/tigdog/).

— Build a macroblock (16x16 block).

— Define a search parameter "p".

— Search the correspondence of the macroblock of a reference frame using an exhaustive method. This method searches for the minimum cost function at each possible location with respect to the search parameter.

— Use as a metric for matching the cost function "Mean Square Error (MSE)".

— Create the motion vectors with the minimum cost function.

4.2 Motion Compensation

— Use the motion vectors to predict a frame in a video stream

— Use the Peak Signal-to-Noise Ratio (PSNR) to measure the distorsion of the frame which is estimated and the predicted one.

TD 5 - Stereoscopy

Stereoscopy is a process which results an illusion of a depth. Usually, it is used a pair of images (2 offsets/perspectives of the same image), each one of which is introduced to a different eye and they are combined into the brain giving the perception of a 3D view.

— Use the proposed stereoscopic images.
— Split them into 2 separate images
— Alternate them such that you will be consistent to the anaglyph 3D (red, cyan).
— Use TD3 to code them using 2 decomposition layers. The quantization step should be R=4 for the 1^{st} layer and R=3 for the 2^{nd}.
— Reconstruct each one of the images
— Superimpose the two images in order to display them. (You need to use a shifting parameter between the red and the cyan version of the same image to superimpose them in case you have a dataset which consists of two offsets of the same image.)
— Explain and comment your results.