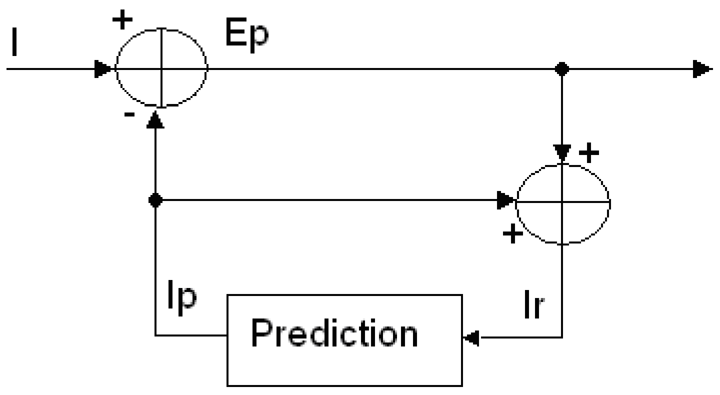
Exercise – Properties of the prediction error

In this exercise, we will show that the error between an image and its prediction has interesting statistical characteristics when we want to use an entropic coding to compress and transmit an image.

Introduction

An image is a signal whose elements are strongly correlated. For example, for a grayscale image, on the homogeneous areas (without contour) a pixel has a gray level close to the levels of its neighbours. There is a spatial redundancy and thus one can predict the neighbouring pixels by using predictive functions. The following predictive system is considered:



where I is the original image, Ip is the predicted image, Ep is the prediction error and Ir is the reconstructed image. The image is scanned with a raster scan (from left to right, top to bottom). The notation of the pixels used for the prediction is given on the figure below (where X is the pixel to be encoded):

B	C
A	**X**

Case of the predictive function P(X) = A

Let us consider the predictive function: **P(X) = A** with the following initial condition: for each row, the first column of the reconstructed image is the same as the first column of the original image I.

1- Load the *Boats_lumi.bmp* image. Scan the pixels with a raster scan by using two imbricated for loops.
For each pixel I(m,n) of the original image, calculate the prediction Ip(m,n). Deduce the prediction error Ep(m,n) and the reconstructed value Ir(m,n).

2- Display the images I, Ir and Ep. What do you notice on the error image ? Interpret these results.

3- The theoretical dynamic range of the prediction error is the range [-255, +255]. Calculate the histogram of the prediction error on this range and plot it.

Display the histogram and ...

4- Calculate the entropy of the original image and the entropy of the prediction error. Compare these values and draw a conclusion about the properties of the prediction error image.

5- Calculate the mean square error from the image Ep. Deduce the peak signal to noise ratio (PSNR) for this kind of predictive function.

Other examples of predictive functions

Let us consider now two other predictions $P_1(X) = C$ and $P_2(X) = (A+C)/2$:

- with the first prediction, we assume that the first row is copied into Ir without error;
- with the second prediction, we assume that the first column is copied into Ir without error and $P_2(X) = A$ for the pixels of the first row.

For each prediction:

- calculate and display the error image,
- calculate the entropy of the prediction error image,
- calculate the PSNR.

Conclude.

Solution of the exercise: Properties of the prediction error

Case of the predictive function P(X) = A

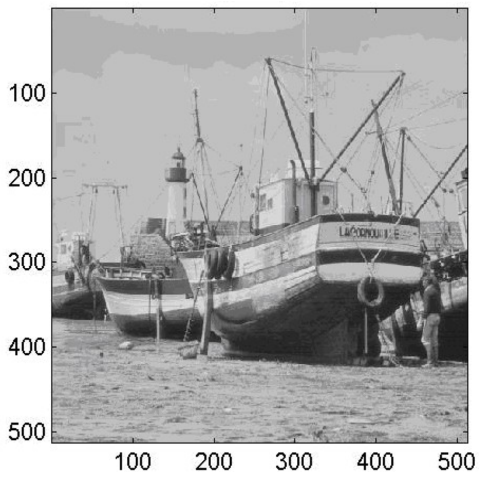
1. Here are the Matlab commands to load the *Boats_lumi.bmp* image and to initialize the matrix of the predicted image, the prediction error and the reconstructed image:

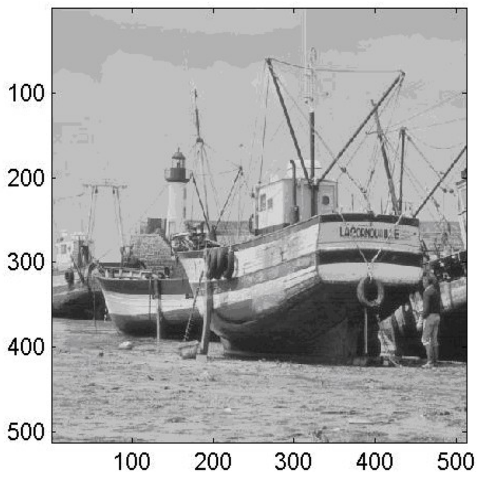
```
% - Load the original image
I = imread('BOATS_LUMI.BMP');
I = double(I);
[m, n] = size(I);

Ip = zeros(m,n);      % prediction image
Ip(:,1) = NaN;        % 1st row not defined

Ep = zeros(m,n);      % prediction error
Ep(:,1) = NaN;        % 1st row not defined

Ir = zeros(m,n);      % reconstructed image
Ir(:,1) = I(:,1);     % 1st row of the reconstructed image
```

Then we can scan the pixels and compute the values of the matrix Ip, Ep and Ir:

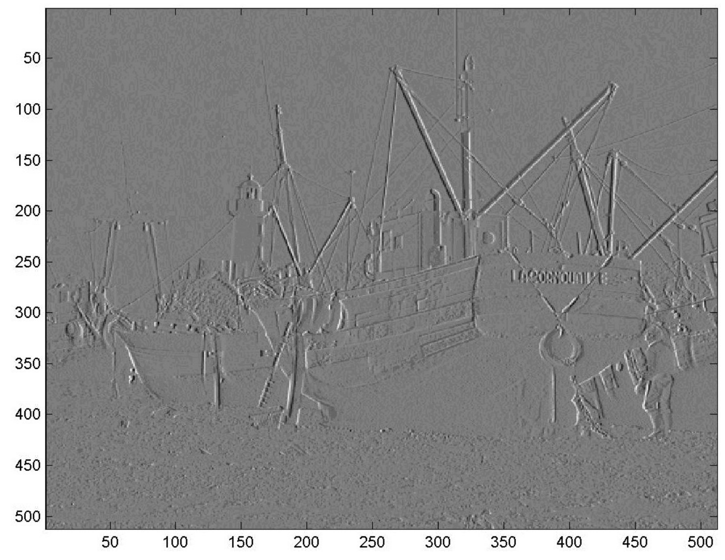
```
for i = 1:m    % row scan
   for j = 2:n % column scan for each row
      Ip(i,j) = Ir(i,j-1);
      Ep(i,j) = I(i,j) - Ip(i,j);
      Ir(i,j) = Ip(i,j) + Ep(i,j);
   end
end
```

2. Here are the original image (on the left) and its prediction (on the right):





Visually there is a masking effect and the human visual system cannot detect the errors. In order to see the errors, we display the prediction error image:

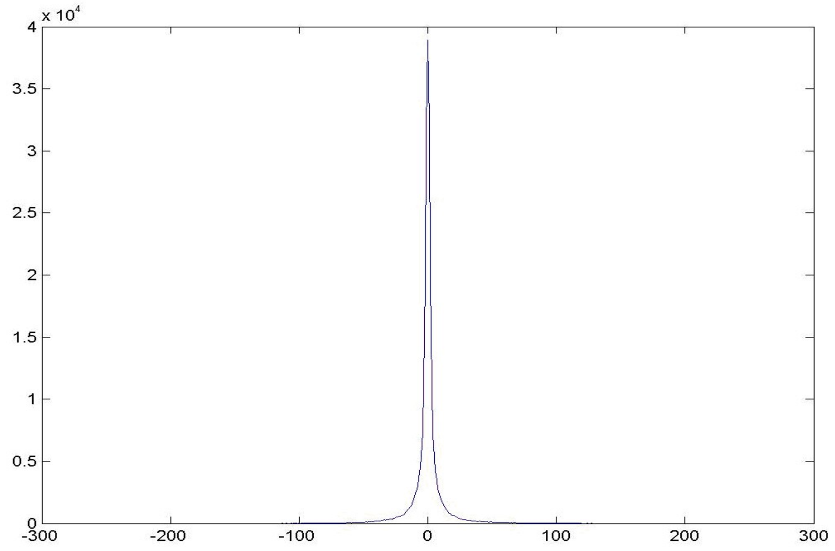


The observation of this image indicates that the errors are higher for the vertical contours. Indeed, the predictive function P(X) = A generates a horizontal gradient between the original image and the predicted image. Vertical contours thus appear in the error image.

3. The dynamic range of the error image is larger than the one of the original image. Theoretically, for an 8-bit original image, the dynamic range can cover the range. To create the histogram **h** of the error image, we must consider the number of pixels for each level in the range [-255, +255]:

```
% - To compute the histogram of Ep
for i = -255:255       % scan the levels in [-255, +255]
   j = find(Ep == i); % look for the pixel values equal to level i
   hist_err(i+256) = length(j); % number of pixels that have the level i
end
% - To display the histogram
niv = -255 : 255 ; %  levels of Ep
plot(niv, hist_err)
```

Here is the histogram obtained:



The normalized histogram according to the image size is the probability density function of the image gray levels. We thus notice that the repartition of this probability density function is more interesting than the one of the original image. Indeed a statistical coding would achieve a better compression by encoding the error image instead of the original image.

4. The entropy H of the source S = {s_1, ..., s_N} is defined by:

$$H(S) = -\sum_{i=1}^{N} p_i \log_2(p_i)$$

where p_i stands for the probability of the symbol s_i.
In the case of a grayscale image, the symbols s_i are the gray levels in the image.
Here is a command example to type to calculate the entropy of the original image:

```
hist_im = imhist(uint8(I));  % original image histogram
i = find(hist_im>0);  %look for the levels that the image contains
pr_im = hist_im(i)/(size(I,1)*size(I,2)); % gray level probabilities
H = -sum(pr_im.*log2(pr_im));  % original image entropy
```

in the same way, knowing the error image histogram hist_err error:

```
i = find(hist_err>0);
pr_err = hist_err(i)/(size(I,1)*size(I,2));
H2 = -sum(pr_err.*log2(pr_err));
```

In the case of the *Boats_lumi.bmp* image, the entropy H of the image is 6.65 bits/symbol. The entropy H2 of the error image is lower: 4.77 bits/symbol. A statistical coding (Huffman, arithmetic, etc.) allows us to encode the prediction error image with less bits than we need to encode the original image. In term of compression rate, it is thus more interesting to encode the prediction error image.

5. To calculate the mean square error MSE and the peak signal to noise ration PSNR1:

```
SE = Ep(:,2:end).^2; % !! the first column is not defined !!
MSE = sum(sum(SE))/(size(I,1)*(size(I,2)-1));
PSNR1 = 10*log10( 255^2/MSE );
```

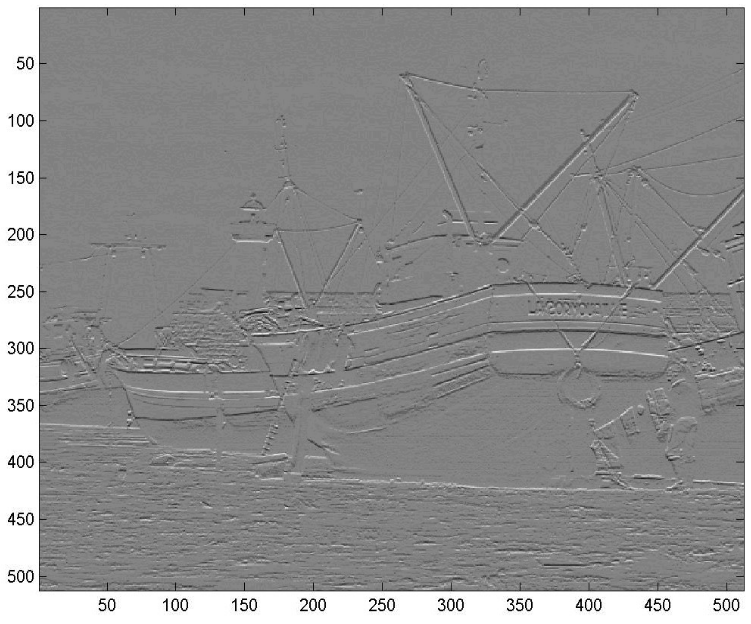
the mean square error is 164.49 and the peak signal to noise ratio is 25.97 dB.

Other examples of predictive functions

1. For the predictive function P(X) = C, here are the commands to obtain the prediction error image:

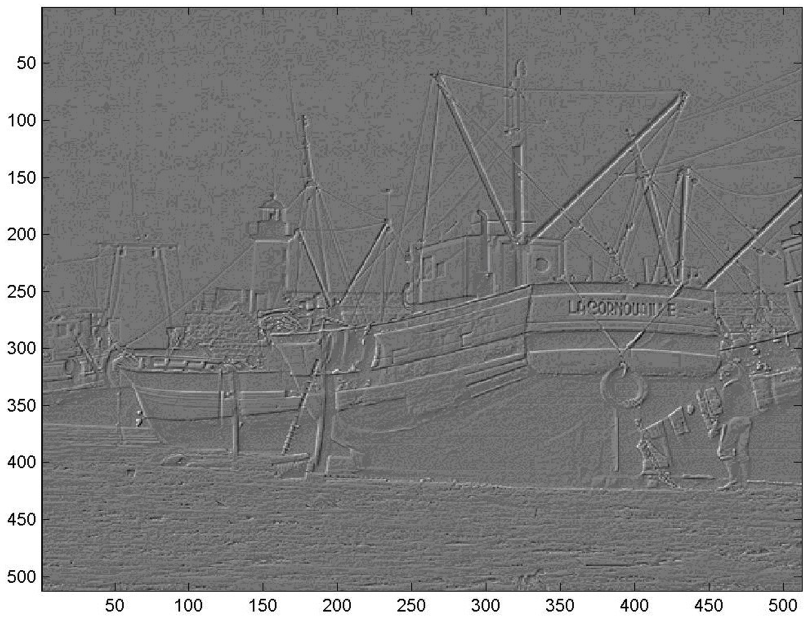
```
[m, n] = size(I);
Ip = zeros(m,n);      % predicted image
Ip(1,:) = NaN;        % 1st row is not defined

Ep = zeros(m,n);      % prediction error
Ep(1,:) = NaN;        % 1st row is not defined

Ir = zeros(m,n);      % reconstructed image
Ir(1,:) = I(1,:); % 1st row of the reconstructed image
for i = 2:m
   for j = 1:n
      Ip(i,j) = Ir(i-1,j);
      Ep(i,j) = I(i,j) - Ip(i,j);
      Ir(i,j) = Ip(i,j) + Ep(i,j);
   end
end
imagesc(Ep)
colormap(gray)
```

The prediction error image is displayed below:



The errors are mainly visible on the horizontal contours because the predictive function generates a vertical gradient between the original image and the predicted image.
The error image entropy is 4.99 bits/symbol and the PSNR is 25.51dB. The predictive function P(X) = C is thus less efficient in terms of compression than the predictive function P(X) = A for the *Boats_lumi* image. However, the PSNR is higher. In both cases, the errors on the predicted image are not perceptible by the human visual system.

Here is the image obtained with the predictive function P(X) = (A+C)/2 :



As in the previous cases, the prediction error is more visible on the image contours. The errors are there located around vertical and horizontal contours. The prediction however reduces these errors: the PSNR is 28.1dB. The entropy is 2.82 bits/symbol. A statistical coding allows us to obtain a better compression rate with this type of prediction.