RRY025-- Image Processing

Exercises 5 – Lecture 5 – Wavelet I

EX 1: Getting familiar with computing the Wavelet decomposition.

Exercise with basic patterns to develop understanding of how the DWT is calculated by hand and by () in Matlab.

Construct a test image that has vertical and horizontal lines on it. Use the following code:

```
interval = 10.
array = zeros(100, 100);
array(:, 1:interval:end) = 1.;
array(1:interval:end, :) = 1.;
```

Perform level 1 DWT. Inspect the approximation and detail images.

Approximation: Why are there bright spots at the locations in which the lines cross? (Examine the situation via a by-hand calculation).

ANSWER: One should see this directly from the by-hand calculation. The bright spots are where the approximation is averaged over the largest number of '1's.

Vertical details: Why are the vertical lines not constant in brightness? Examine the situation again via a by-hand calculation. Can you also explain qualitatively why the lines are not constant brightness, even though the vertical lines in the original image are constant in brightness?

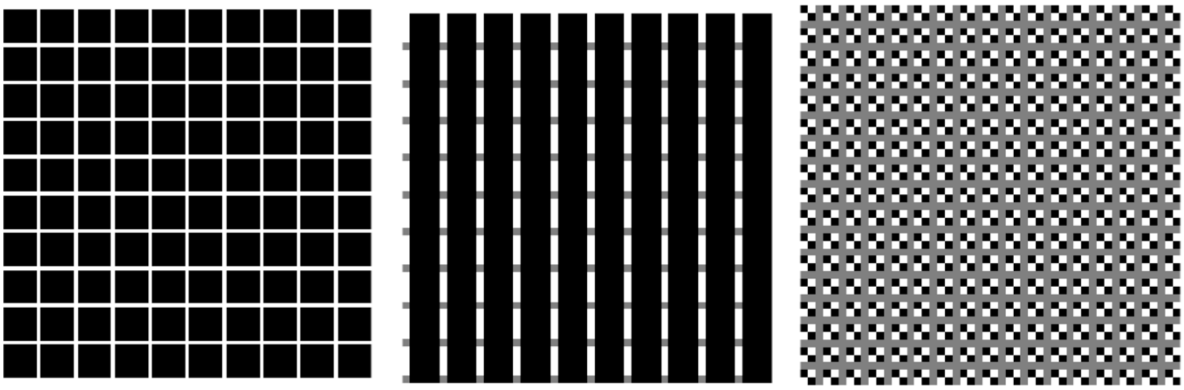
ANSWER: One should see this directly from the by-hand calculation. Since the detail is calculated as a difference to the mean (approximation), the detail is small in pixels in which the approximation is large.

Change the *interval* to 3 and inspect the diagonal details. Can you relate the pattern in the diagonal details image (see the right panel above) to how we calculated the 2D FWT on the whiteboard?

ANSWER: Again, one should see this directly from the by-hand calculation. Applying the 2D kernel of the diagonal details to each 2x2 pixel area results in the shown pattern.

Change the *interval* to 2. Why do all details vanish even though there is still a pattern in the original image?

Reference images for help:



Left: Original image with interval=10; Center: Vertical details with interval=10; Right: Diagonal details with interval=3.

EX 2. Wavelet decomposition by hand

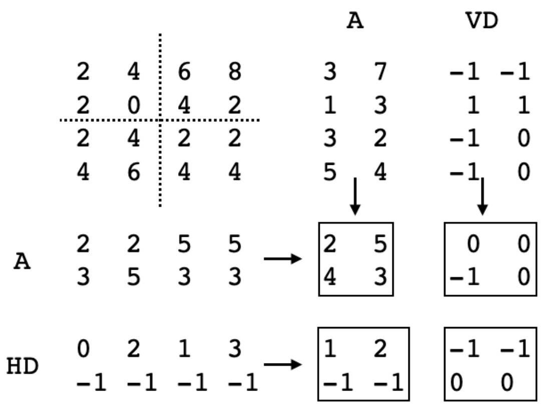
Perform a DWT by hand for the array:

```
2 4 6 8
2 0 4 2
2 4 2 2
4 6 4 4
```

Check your result with Matlab.

Warning: Beware of the indexing direction in Matlab. Commonly horizontal and vertical details are mixed due to x/y direction confusion... Also, Matlab's *dwt2*() scales the coefficients with 2*n* (*n* is the scale; see the help of *dwt2*()).

ANSWER:




```
Diagonal details
a = 1/4 x (1x2 -1x4 -1x2 +1x0) =  -1
b = 1/4 x (1x6 -1x8 -1x4 +1x2) =  -1
c = 1/4 x (1x2 -1x4 -1x4 +1x6) =   0
d = 1/4 x (1x2 -1x2 -1x4 +1x4) =   0
```

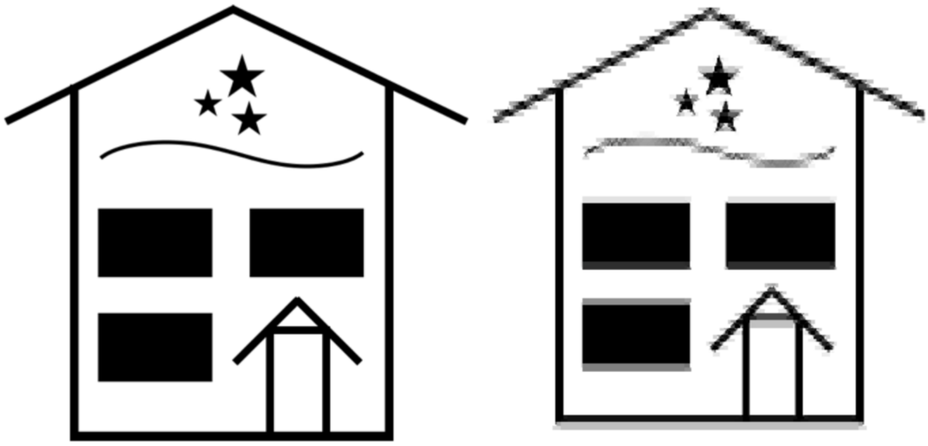
In Matlab:

```
wavelet_type = 'haar';

I = [2 4 6 8; 2 0 4 2; 2 4 2 2; 4 6 4 4]

[a1, hd1, vd1, dd1] = dwt2(I, wavelet_type);
```

```
disp("Approximation:")
a1    / 2.^1
disp("Horizontal Details:")
hd1   / 2.^1
disp("Vertical Details:")
vd1   / 2.^1
disp("Diagonal Details:")
dd1   / 2.^1
```

EX 3. Application of wavelet filtering: selective smoothing

A simplistic example of how wavelets can be used for selective filtering is shown below. The image on the right is wavelet filtered three levels down and then reconstructed by setting all horizontal details to zero. In other words, the horizontal details have been removed from scales 1, 2, and 3. As you can see, the vertical details are sharp as in the original.



Demonstrate the above effect in case of a more realistic photograph. Use the image 'cameraman.tif'. Write a code that 1) performs the FWT down to third level 2) reconstructs the image, but sets the horizontal and diagonal coefficients to zero at all levels.

Hint for the result: The tripod of the cameraman has a vertical pole in the middle. If your code works correctly, you should see a smoothed image with the pole still sharp. See the picture below. This demonstrates the principle of selective filtering by manipulating the wavelet coefficients.

ANSWER: Example given as .mlx code.