Problem 2: Using Haar wavelets to compress 2D photos of size = 512×512

Task #1: Load your photo !

1. Using matlab's *imread* function, read in the *8-bit* black & white image file "Meghan_Markle_BW.tif". Then, invert the black & white tone, where:

$$\underline{\text{Old tone}} \quad \begin{cases} Black &= 0 \\ White &= 255 \end{cases} \xrightarrow{\text{We will change this to:}} \underline{\text{New tone}} \quad \begin{cases} Black &= 127.5 \\ White &= -127.5 \end{cases}$$

Task #2: Compress your photo once by a factor of 2 $\left(J = 9 \xrightarrow{reduce} J = 8 \right)$

2a. Suppose your photo matrix was named "A." Using the high-school algorithm:

- Perform 1 round of column rastering on our original phot matrix A.
- Let's call the resulting matrix = $B = W_9{}^{-1} A$ (This the $J = 9$, post-column raster result)
- Using *mat2gray* and *imshow*, plot the image equivalent of $W_9{}^{-1} A$.
- For the title of this figure, label it "J = 9, post-columns only"

2b. Now, we're ready to operate on the rows of the above matrix ${W_9}^{-1}A$, where we:

- Perform 1 round of row rastering on matrix B (This is equivalent to column-rastering of B^T)
- Let's call the resulting matrix = ${W_9}^{-1}A\ \left({W_9}^{-1}\right)^T$ (This the $J = 9$, post-column & row raster result)

- Using *mat2gray* and *imshow*, plot the image equivalent of ${W_9}^{-1}A\ \left({W_9}^{-1}\right)^T$
- For the title of this figure, label it "Proper J = 8 photo (small upper-left corner)"

You should be able to see a smaller version of Meghan in the upper left-hand corner of your plot.

Remember – the smaller image (shaded in yellow in Figure 1) is the *low-frequency*, *J =8* scale data from 1 round of wavelet transform (ie. After one round of row and column averaging and ½ differences).

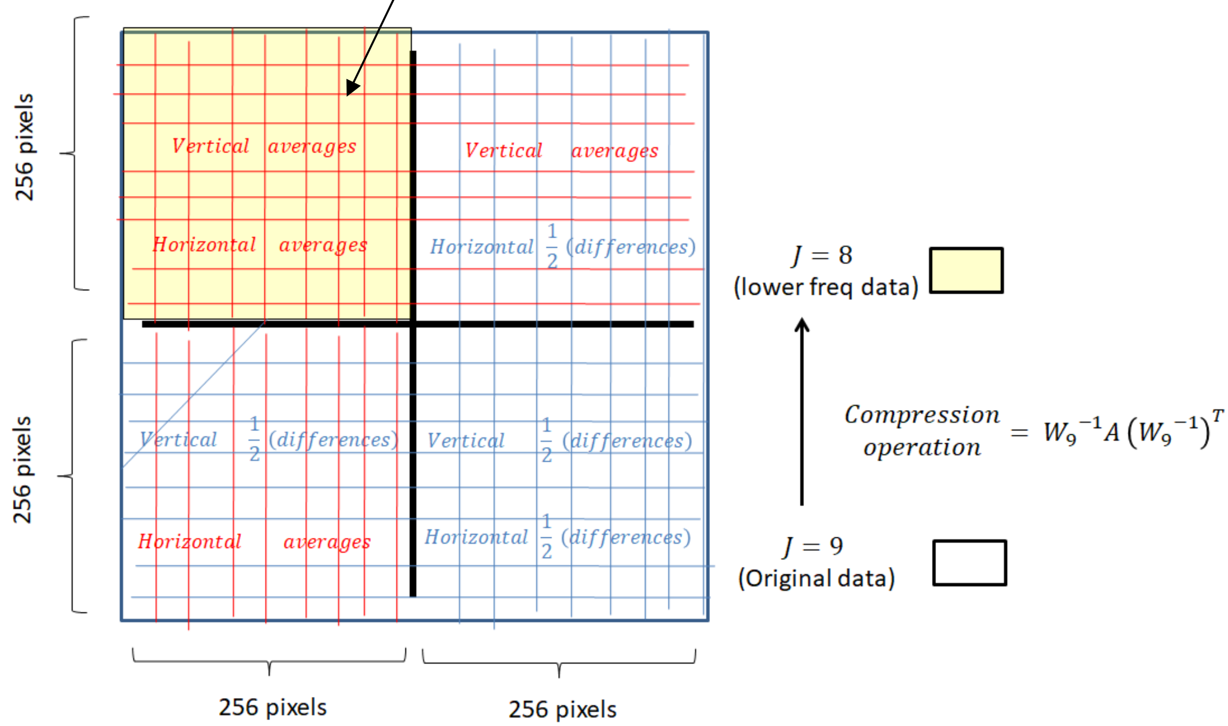


Figure 1: What your plot should look like after 1 whole round of wavelet transform…

Task #3: Compress your photo once more by 2... $\left(J = 8 \xrightarrow{reduce} J = 7 \right)$

3. Now, we would like to compress the upper-left Meghan photo some more !

- Mathematically crop out the top-left 256 x 256 submatrix and call it matrix "C".
- Perform 1 complete round of column / row rastering on C
- Using *mat2gray* and *imshow*, plot the image of the yellow + magenta region in Figure 7
- For the title of this figure, label it "Proper J = 7 photo (upper left corner)"

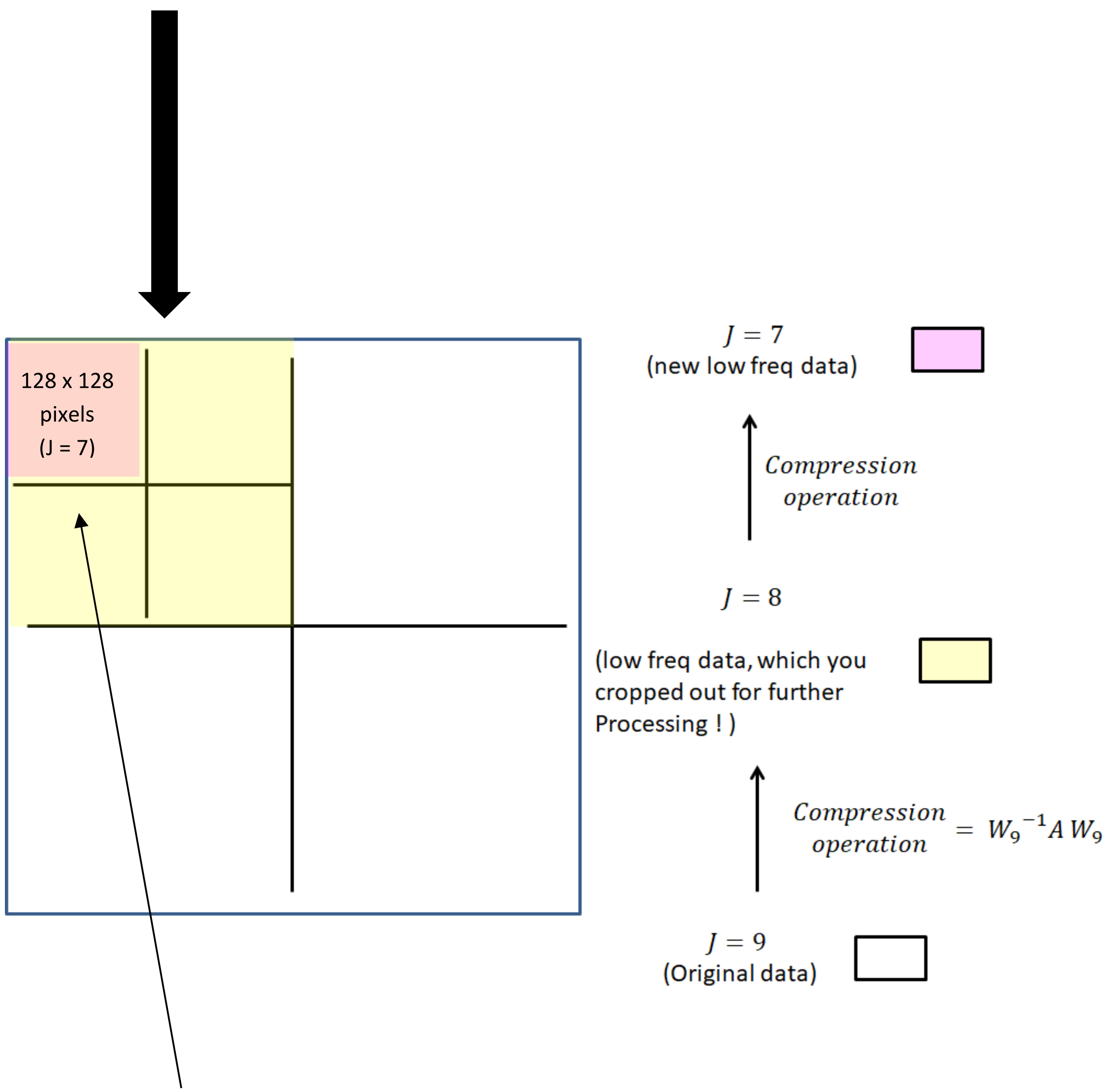


Figure 2: After 1 more round of wavelet transformation, you should get an even smaller Meghan Markle within the magenta region

!

Task #4: Compress your photo 2 more times ! $\left(J = 7 \xrightarrow{reduce\ twice} J = 5 \right)$

4. Creation of proper J = 6 scale image:

   - From the previous image, crop the 128 x 128 upper-left quadrant, perform 1 more round of column / row wavelet transform.
   - Use *mat2gray* and *imshow* to plot the low-frequency data from J = 7 .
   - For the title of this figure, label it "Proper J = 6 photo (upper-left corner)"

5. Creation of proper J = 5 scale image:

   - From the previous image, crop the 64 x 64 upper-left quadrant, perform 1 more round of column / row wavelet transform.
   - Use *mat2gray* and *imshow* to plot the low-frequency data from J = 6 .
   - For the title of this figure, label it "Proper J = 5 photo (upper-left corner)

6. Using matlab's *pcolor* command, plot a pcolor plot of your resulting J = 5 sub-matrix (the data associated with the 32 x 32 tiny Meghan snapshot on the upper-left corner). Then, using the *caxis* command, set the colorbar limits from -127.5 to +127.5

7. Finally, echo your your proper J = 5 submatrix (the 32 x 32 guy from part 8) in your diary file.

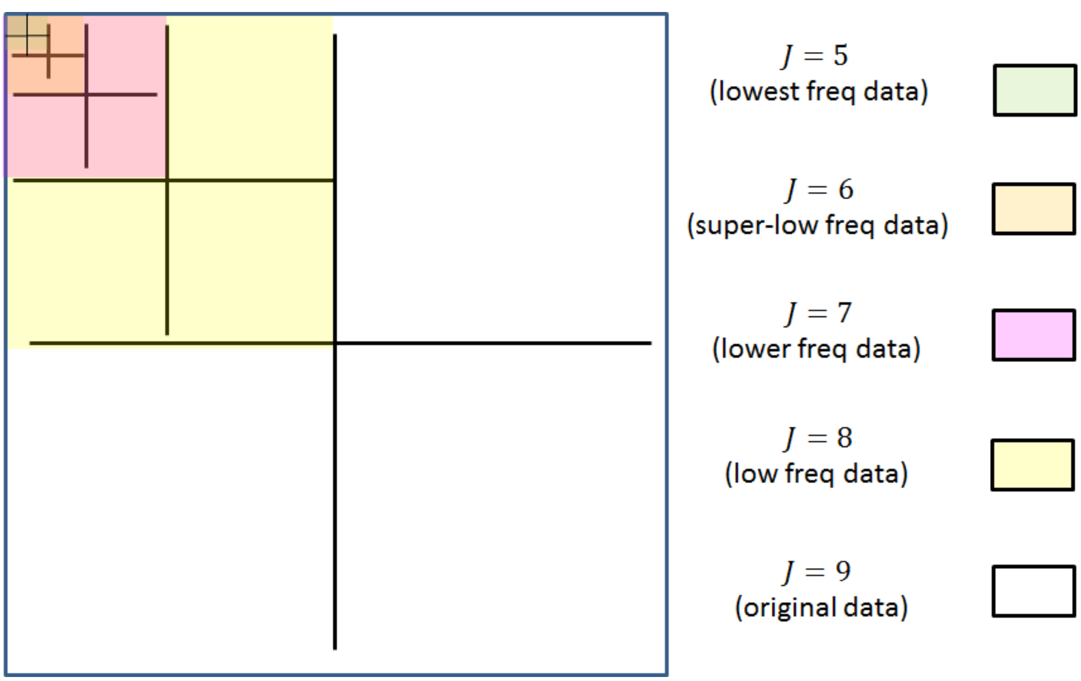


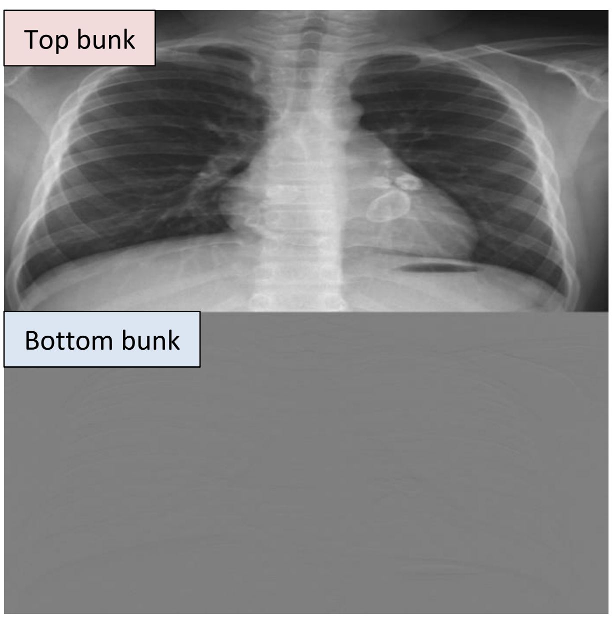
Figure 3: Your Finished product should be a very small image of the dutchess !

Example of 2D wavelet compressions:

Original chest X-ray image

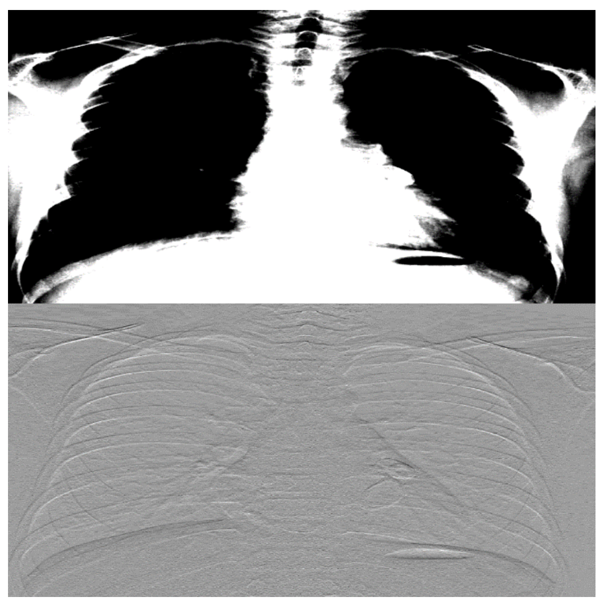
$J = 9 \;\; photo \quad (512 \times 512 \;\; pixels)$

Performing a single round of column-raster operations..



Post-column raster only

$J = 9 \;\; photo \quad (512 \times 512 \;\; pixels)$

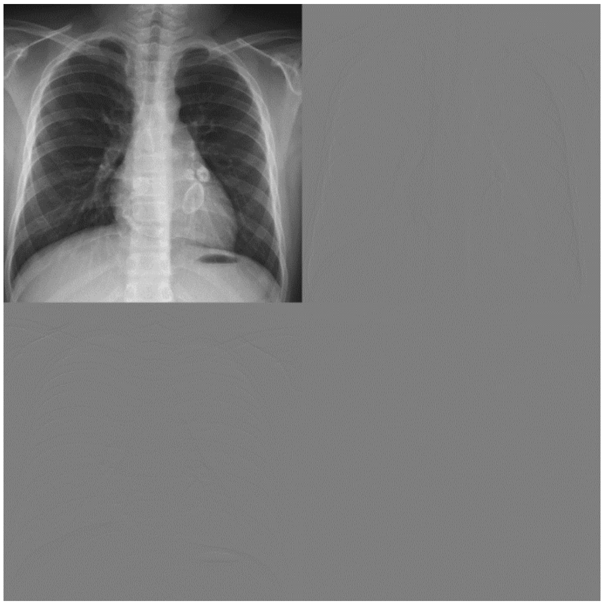


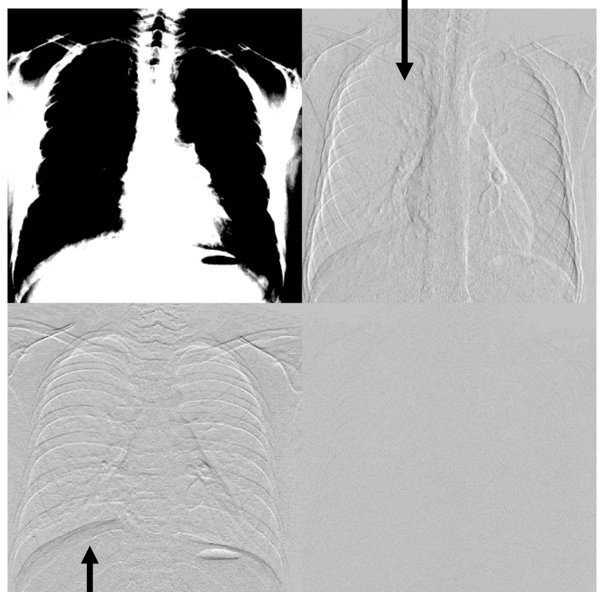
A better look at the high-frequency “vertical” wavelet coefficients in the “bottom bunk” section

(I artificially cranked up the contrast so that they’re actually visible !)

Then, performing a single round of row-raster operations on the previous result

Taking the *averages* in the vertical direction will de-emphasize the ribs





Post-column & row raster result

$J = 8$ *photo on the upper left corner* !!
$(256 \times 256 \ pixels)$

Taking *differences* in the vertical direction will accentuate the ribs

(The "direction of periodic variations" of the ribs actually lies along the $\pm y$-axis)

Epilogue:

Take a good look at your final, tiny snapshot of Meghan…. She is now only 32 x 32 pixels !! Hence, judging only by edge pixels alone, we have:

$$\frac{new\ image\ edge = \ 32}{old\ image\ edge \ = \ 512} = \ 0.0625 \ \sim 6.25\%\ of\ the\ original\ image$$

Now, we have not taken into account to "bit energy compression"… a concept that's similar to the power spectrum of Fourier coefficients, but let's not worry about that detail for now!!