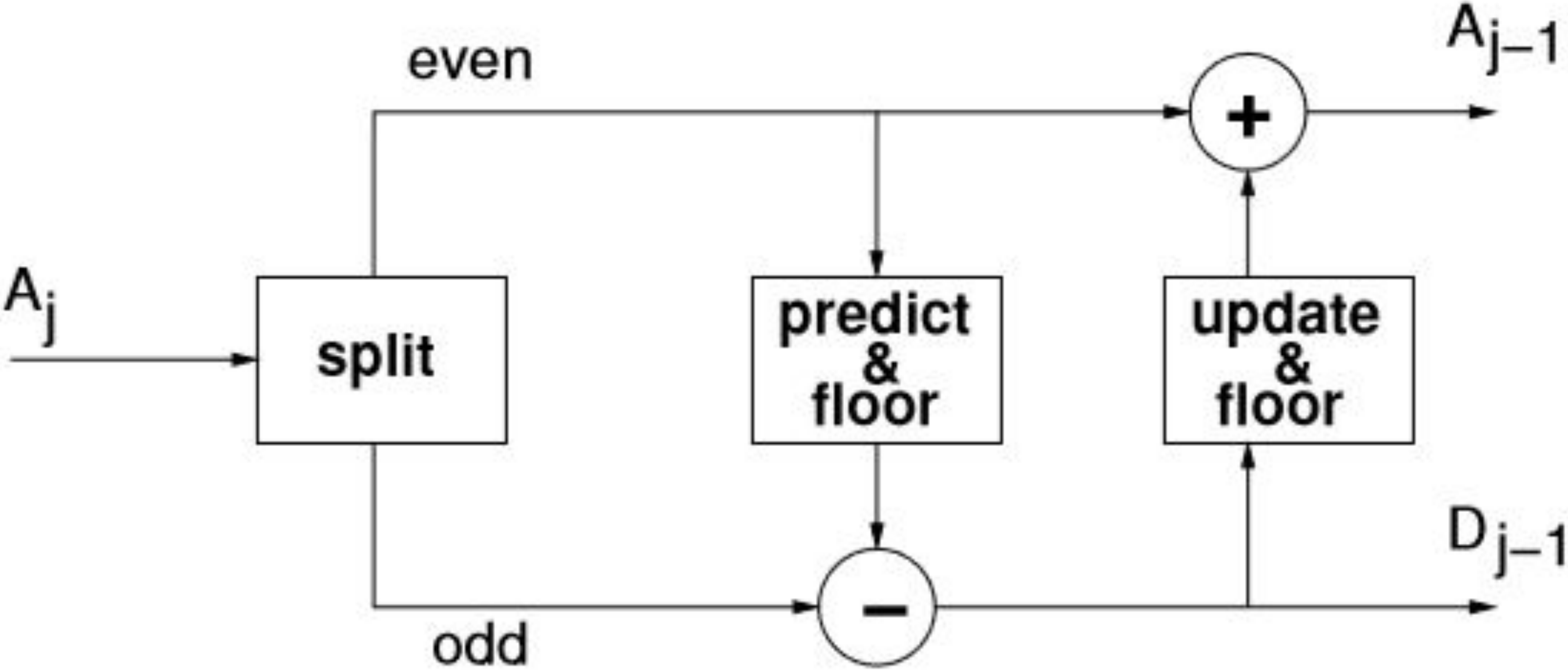
Adaptive Lifting Scheme

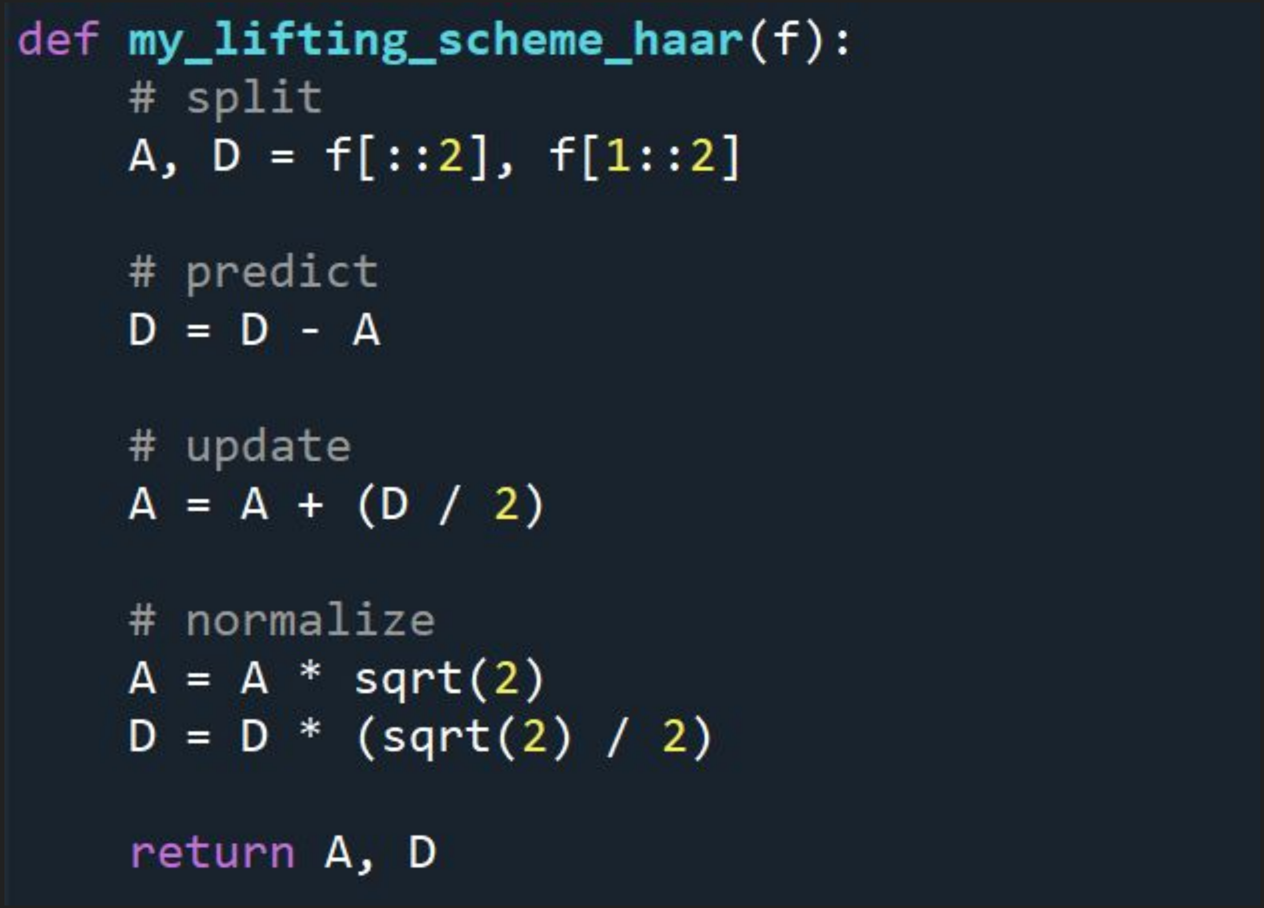
Attila Zsíros

> Design and implement a 2D integer lifting scheme that recursively decomposes the image into the level of individual pixels. The splitting is based on variance. Visualize the output.

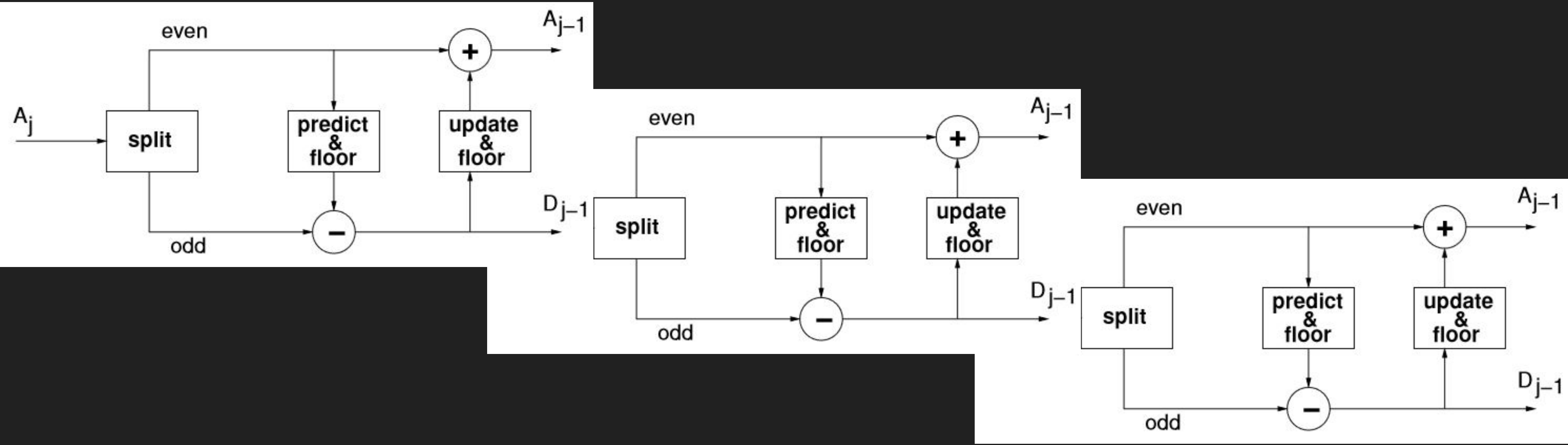
Step 1

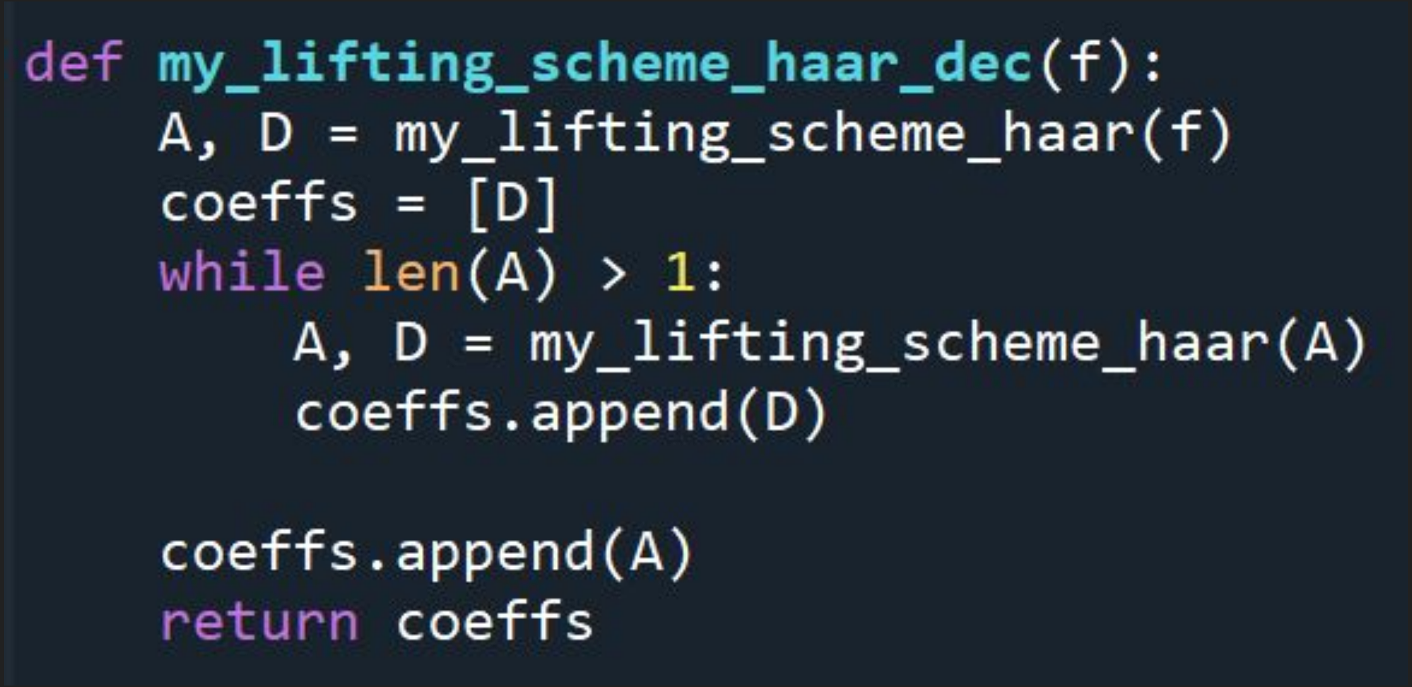
1D DWT using Lifting Scheme


A_j
split
even
odd
predict & floor
update & floor
+
−
A_{j−1}
D_{j−1}



```python
def my_lifting_scheme_haar(f):
    # split
    A, D = f[::2], f[1::2]

    # predict
    D = D - A

    # update
    A = A + (D / 2)

    # normalize
    A = A * sqrt(2)
    D = D * (sqrt(2) / 2)

    return A, D
```


A$_j$
split
even
odd
predict & floor
−
update & floor
+
A$_{j-1}$
D$_{j-1}$
split
even
odd
predict & floor
−
update & floor
+
A$_{j-1}$
D$_{j-1}$
split
even
odd
predict & floor
−
update & floor
+
A$_{j-1}$
D$_{j-1}$

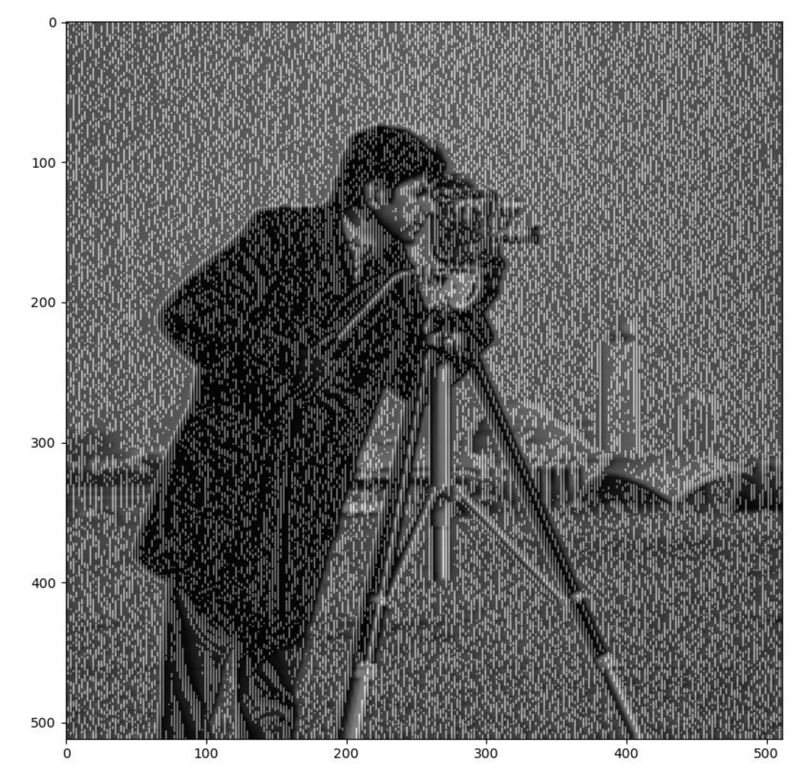


```python
def my_lifting_scheme_haar_dec(f):
    A, D = my_lifting_scheme_haar(f)
    coeffs = [D]
    while len(A) > 1:
        A, D = my_lifting_scheme_haar(A)
        coeffs.append(D)

    coeffs.append(A)
    return coeffs
```

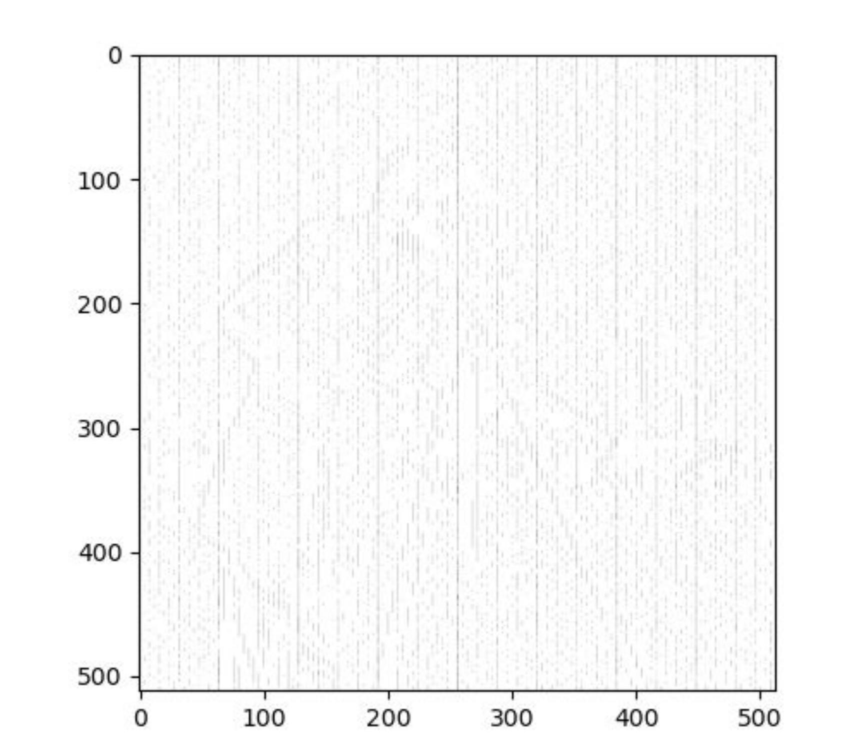
Step 2

2D DWT using Lifting Scheme

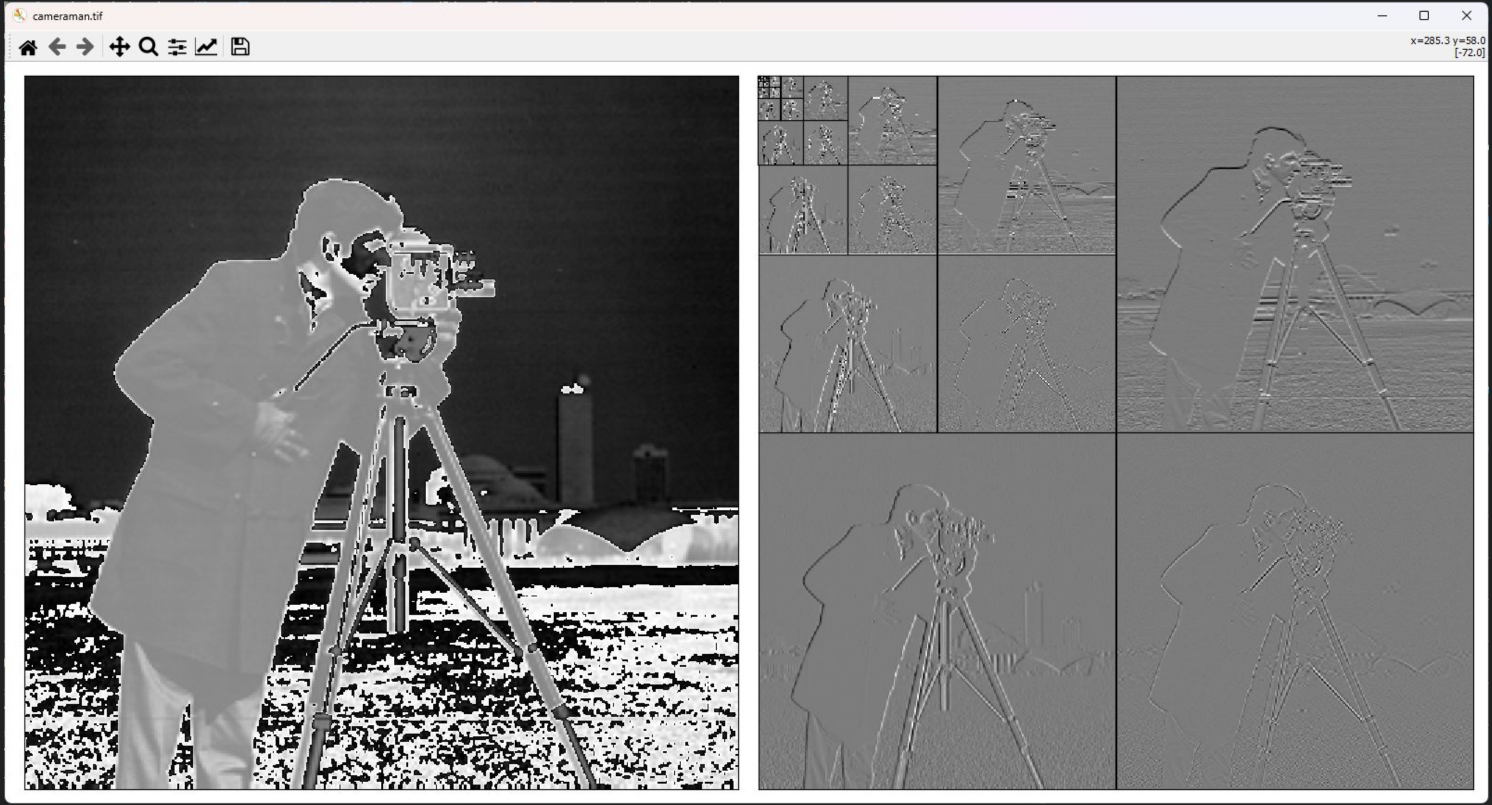
Type overflows



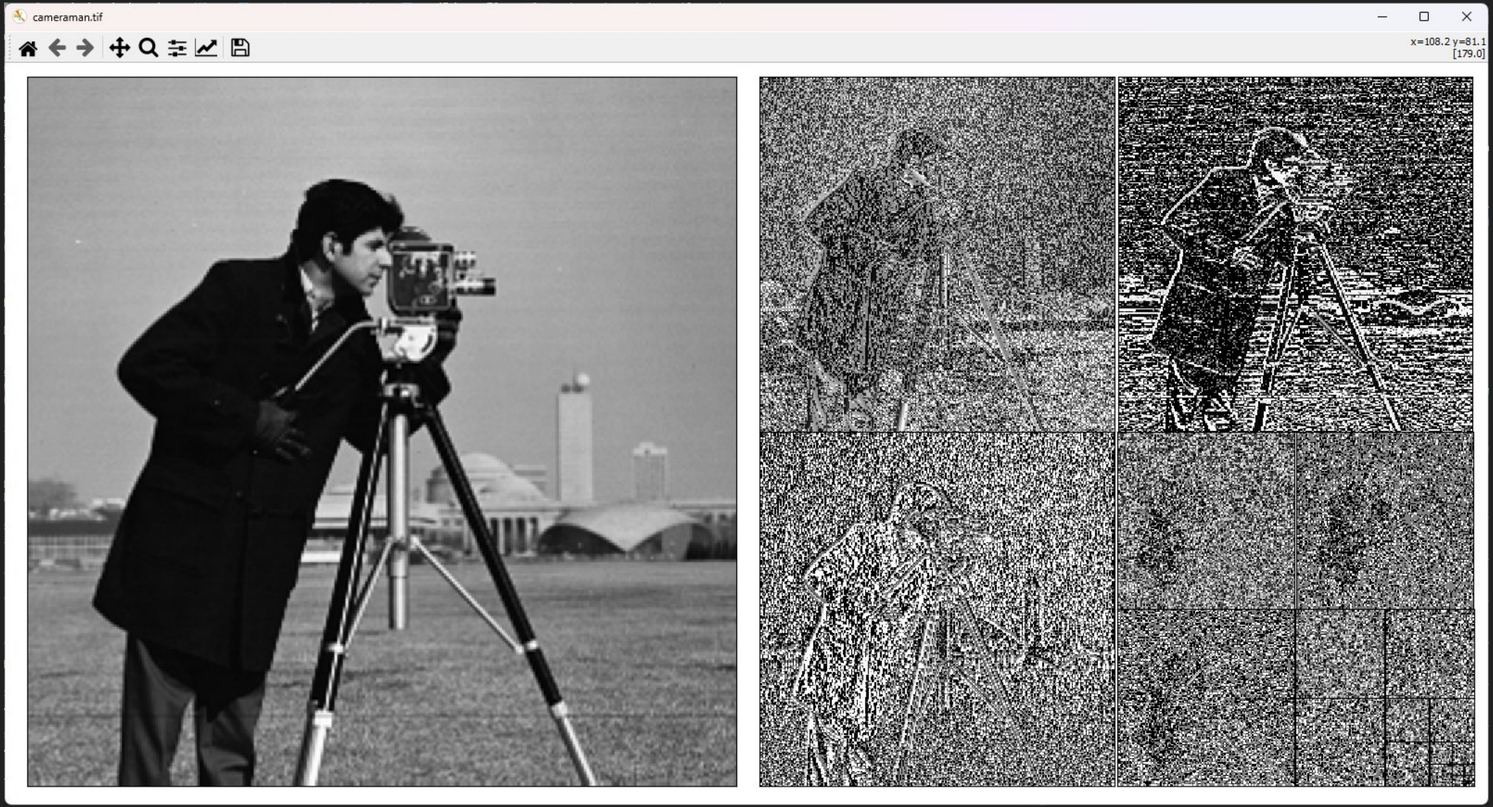
Type overflows



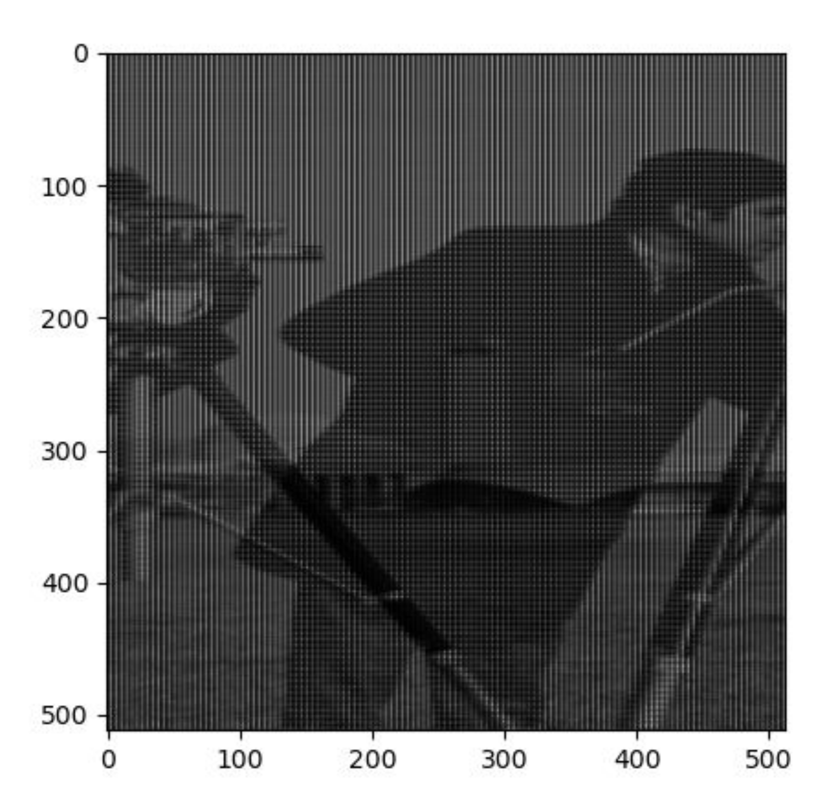
int8



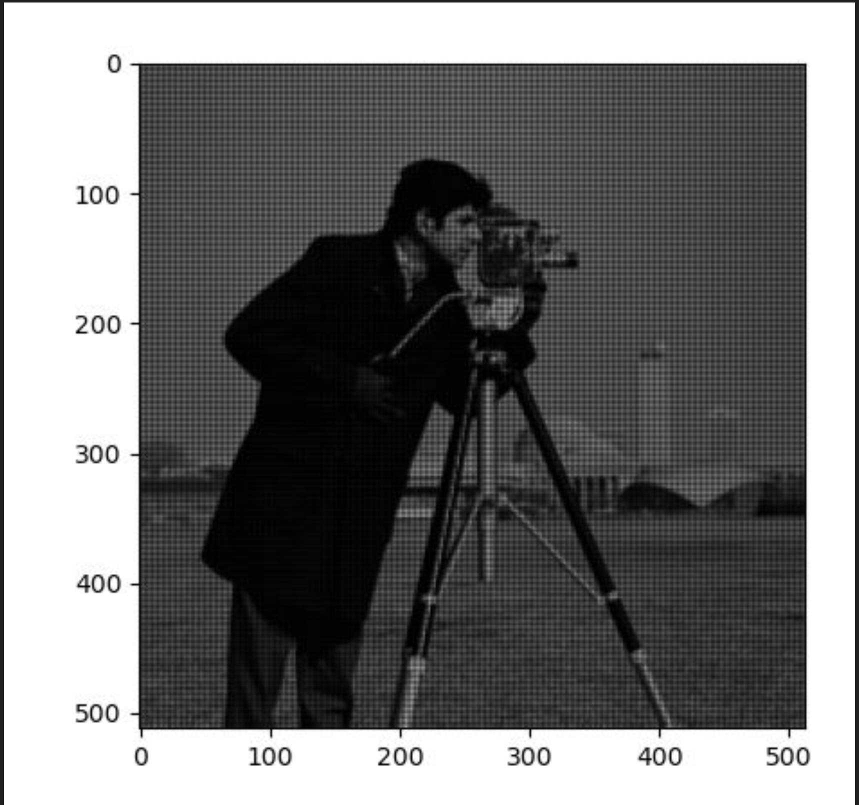
uint8

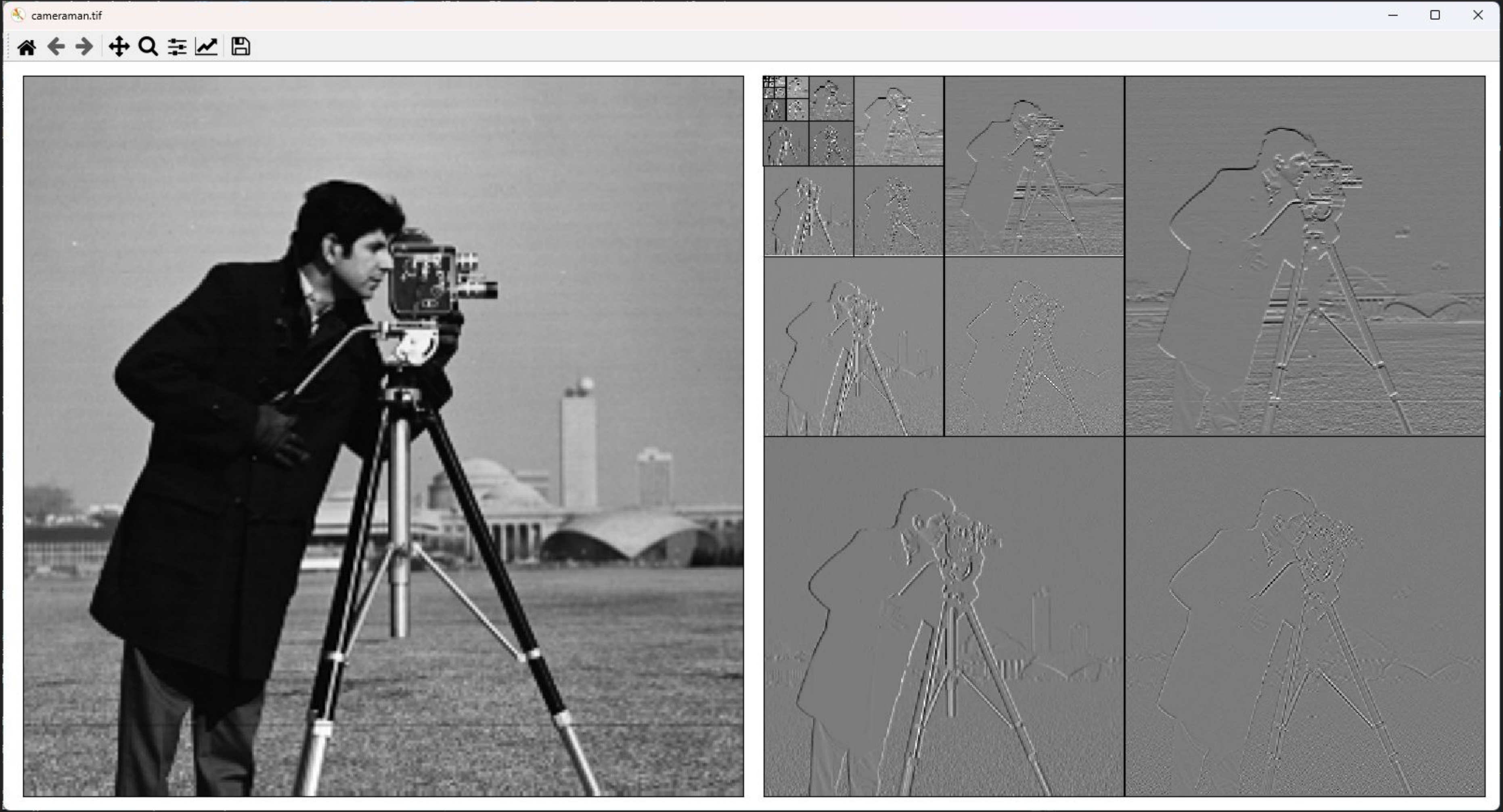


Incorrect reshaping axes



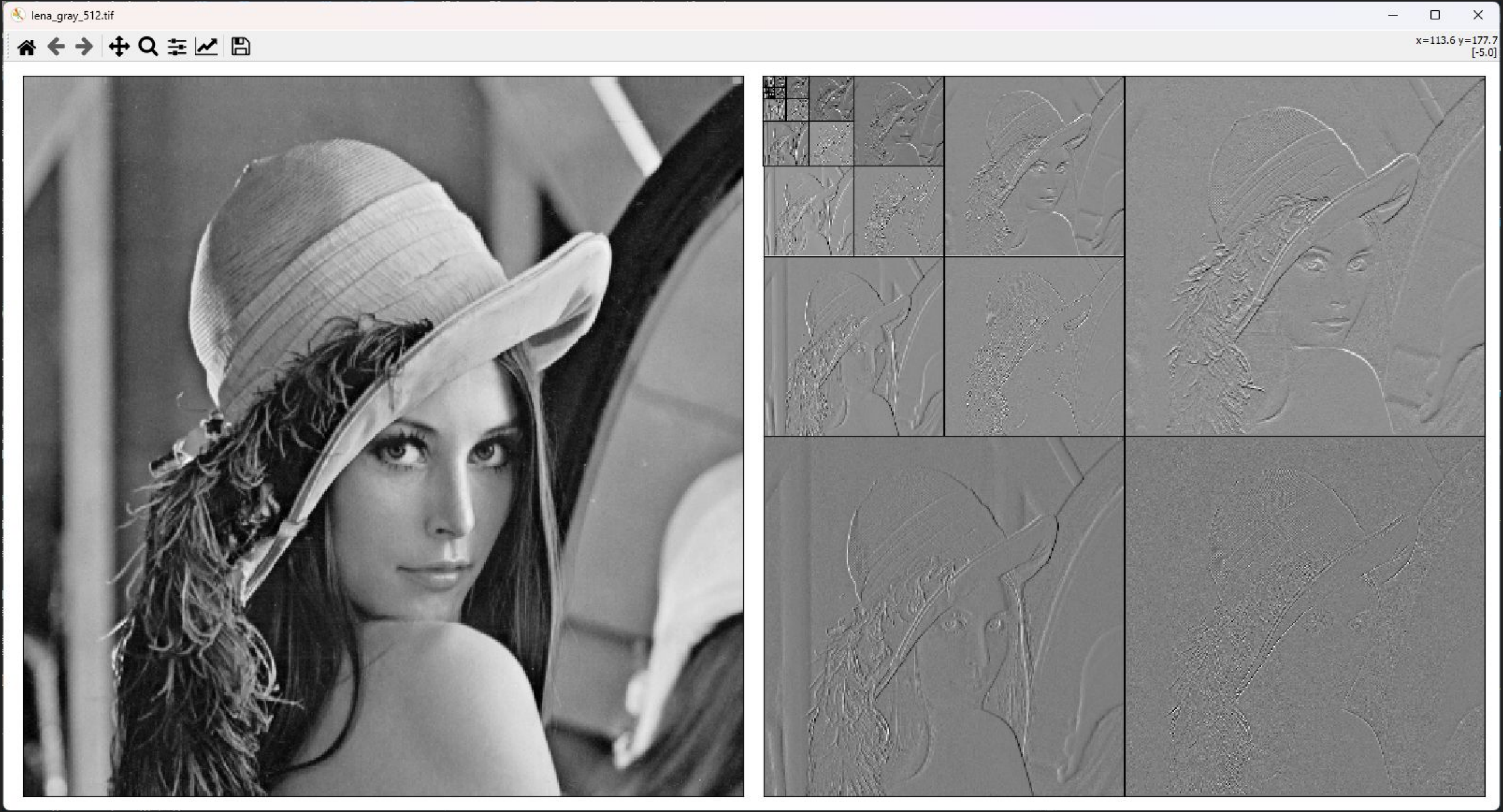
Using only DC (typo)




cameraman.tif

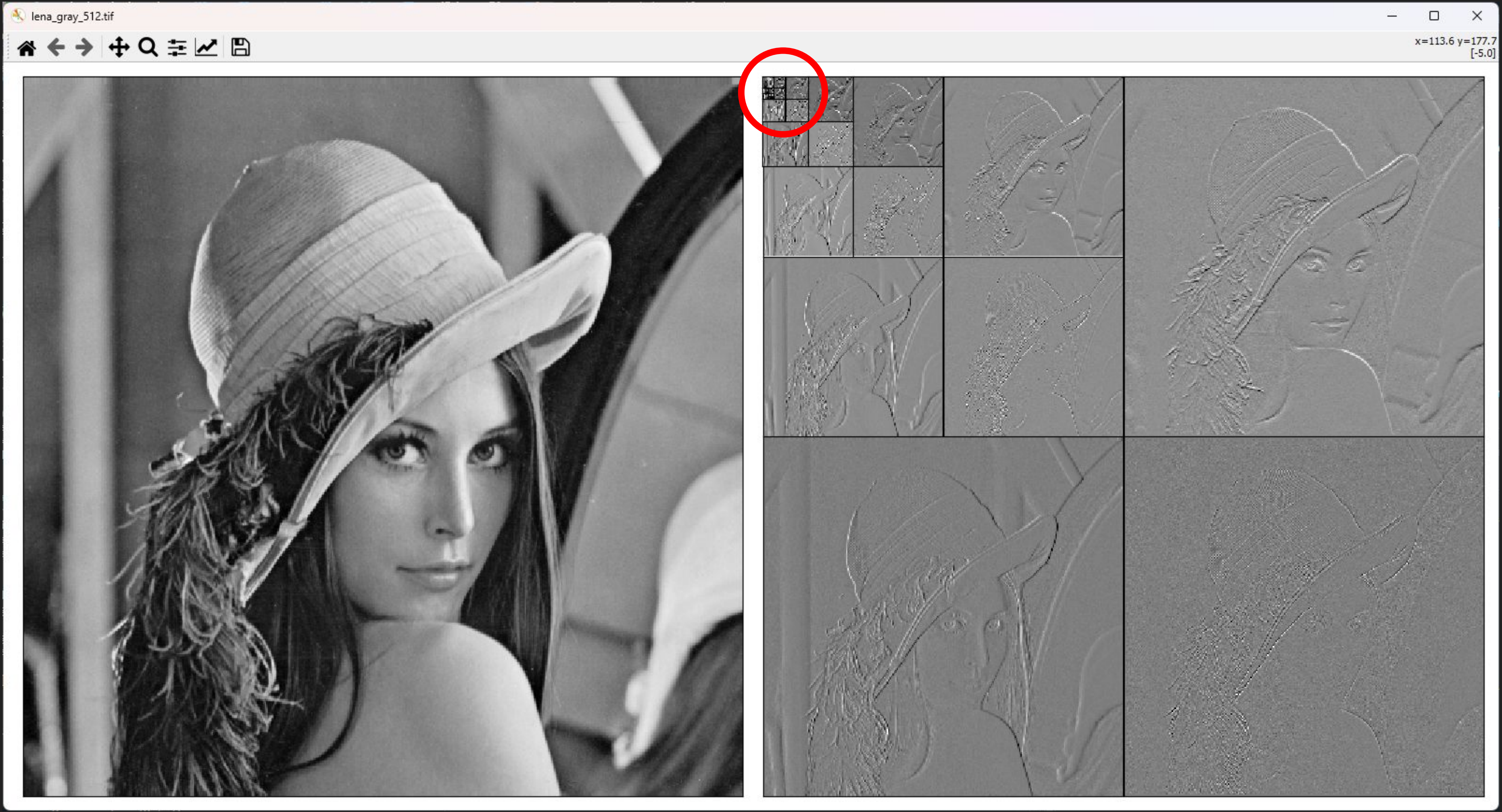
Step 3

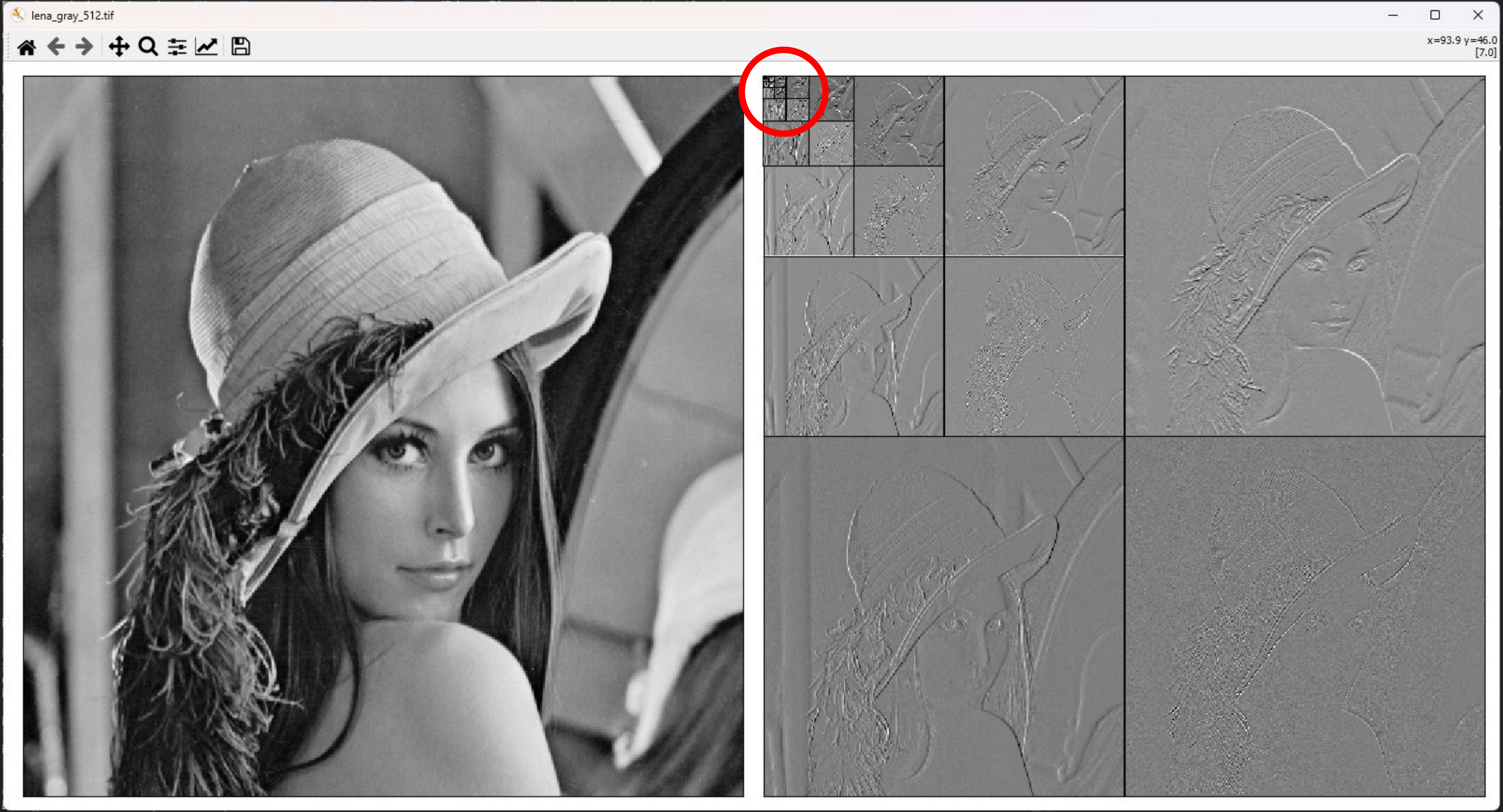
2D Adaptive Integer DWT using Lifting Scheme

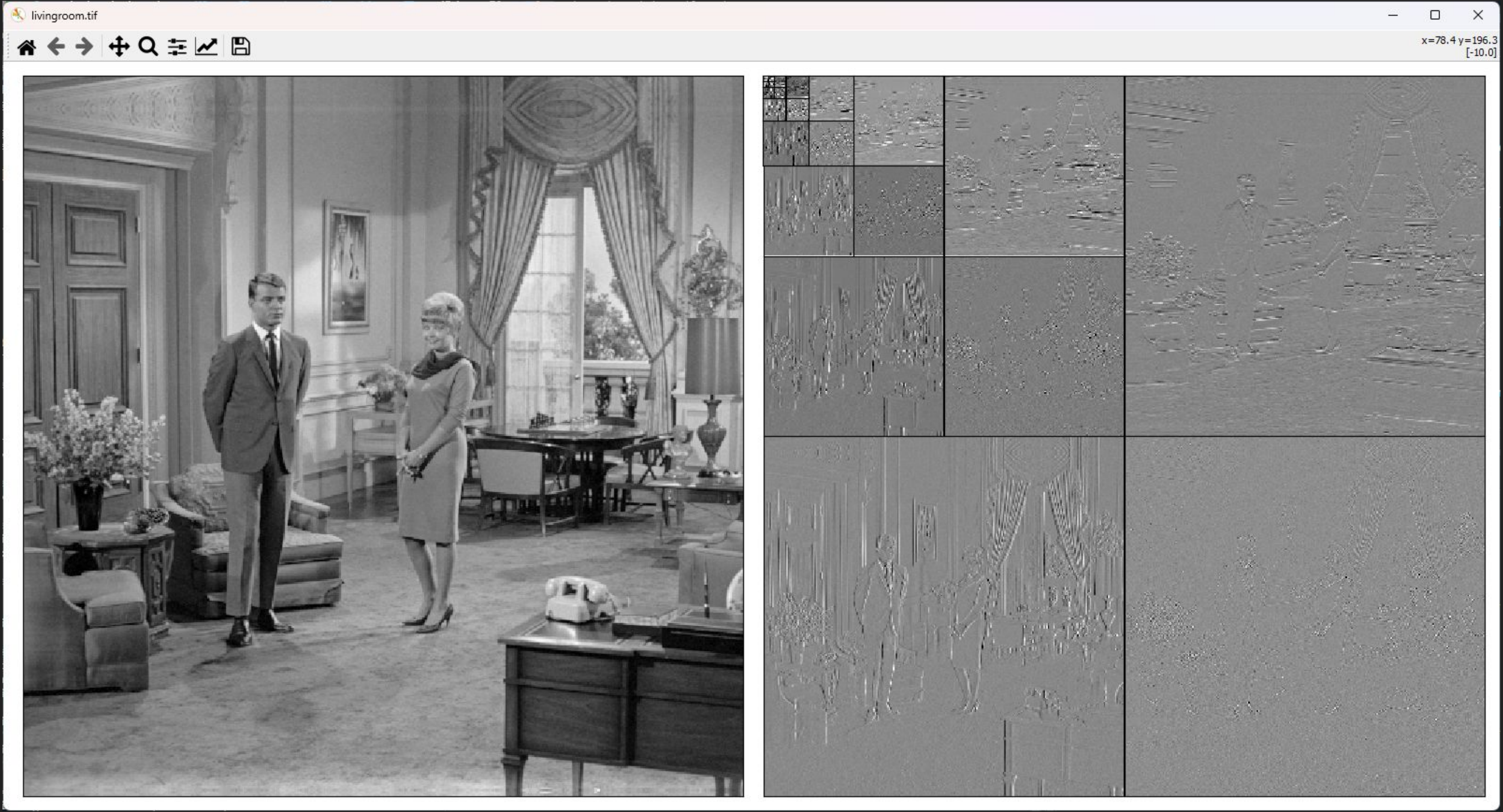


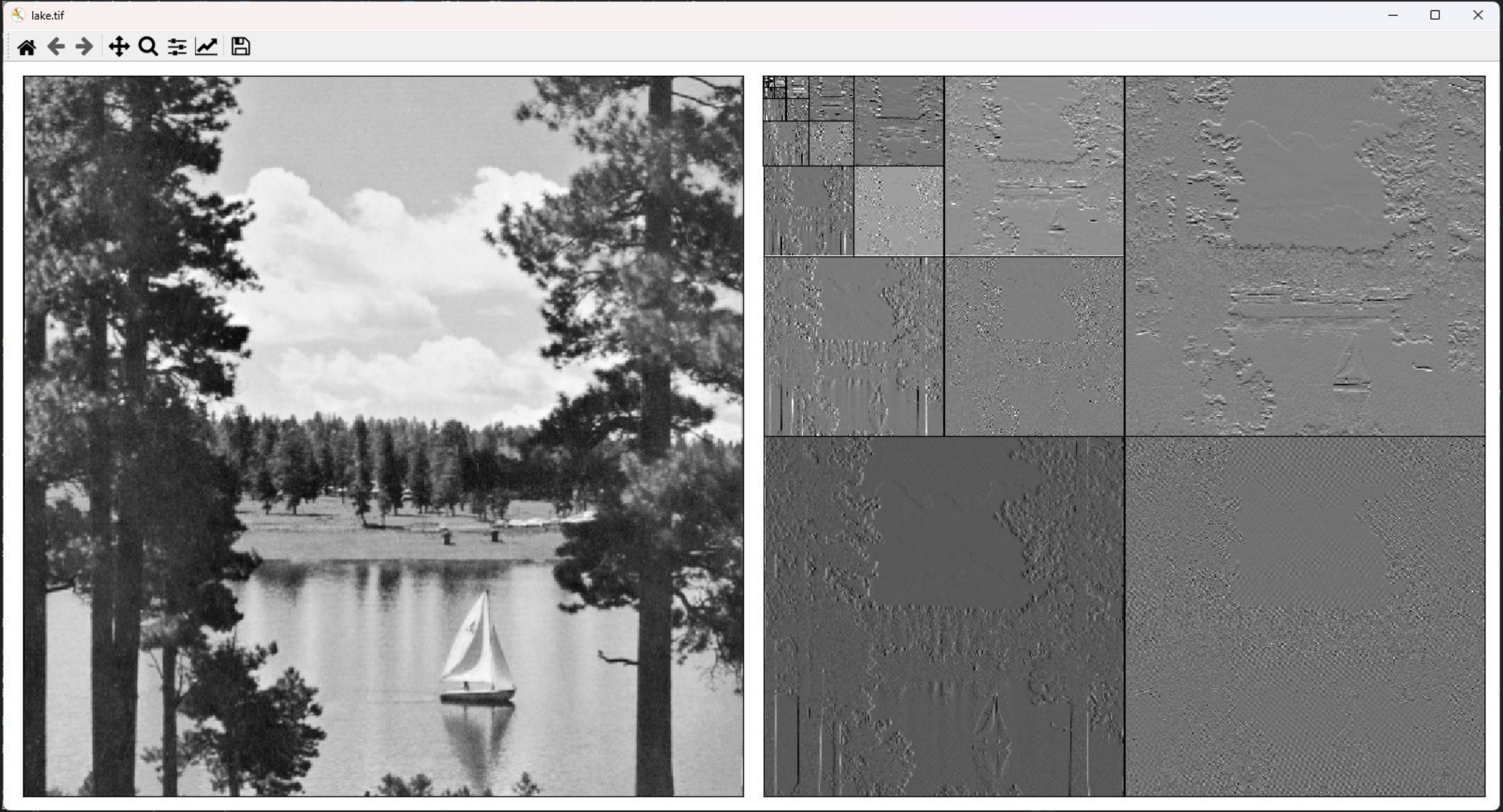
lena_gray_512.tif

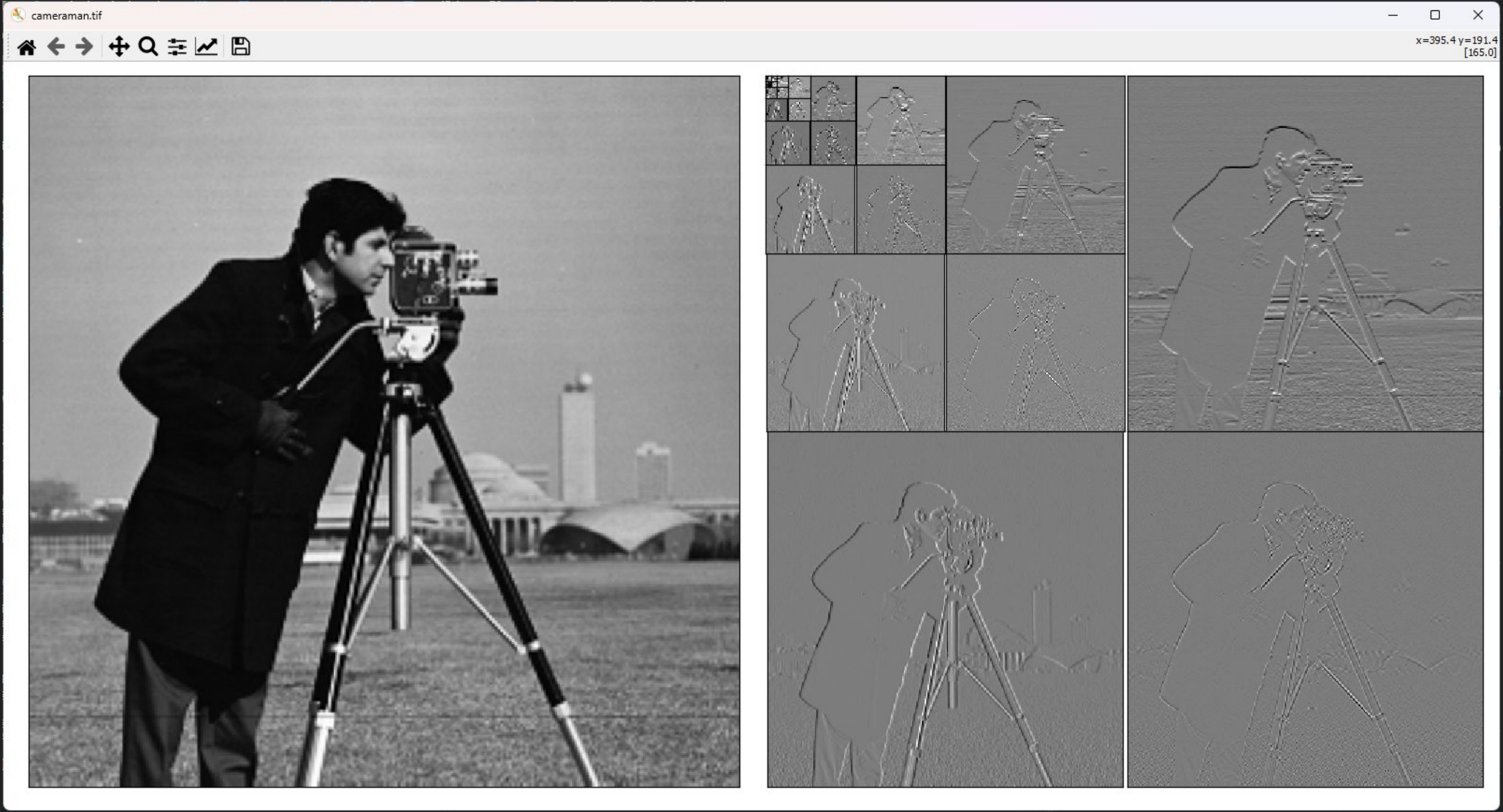
x=113.6 y=177.7
[-5.0]

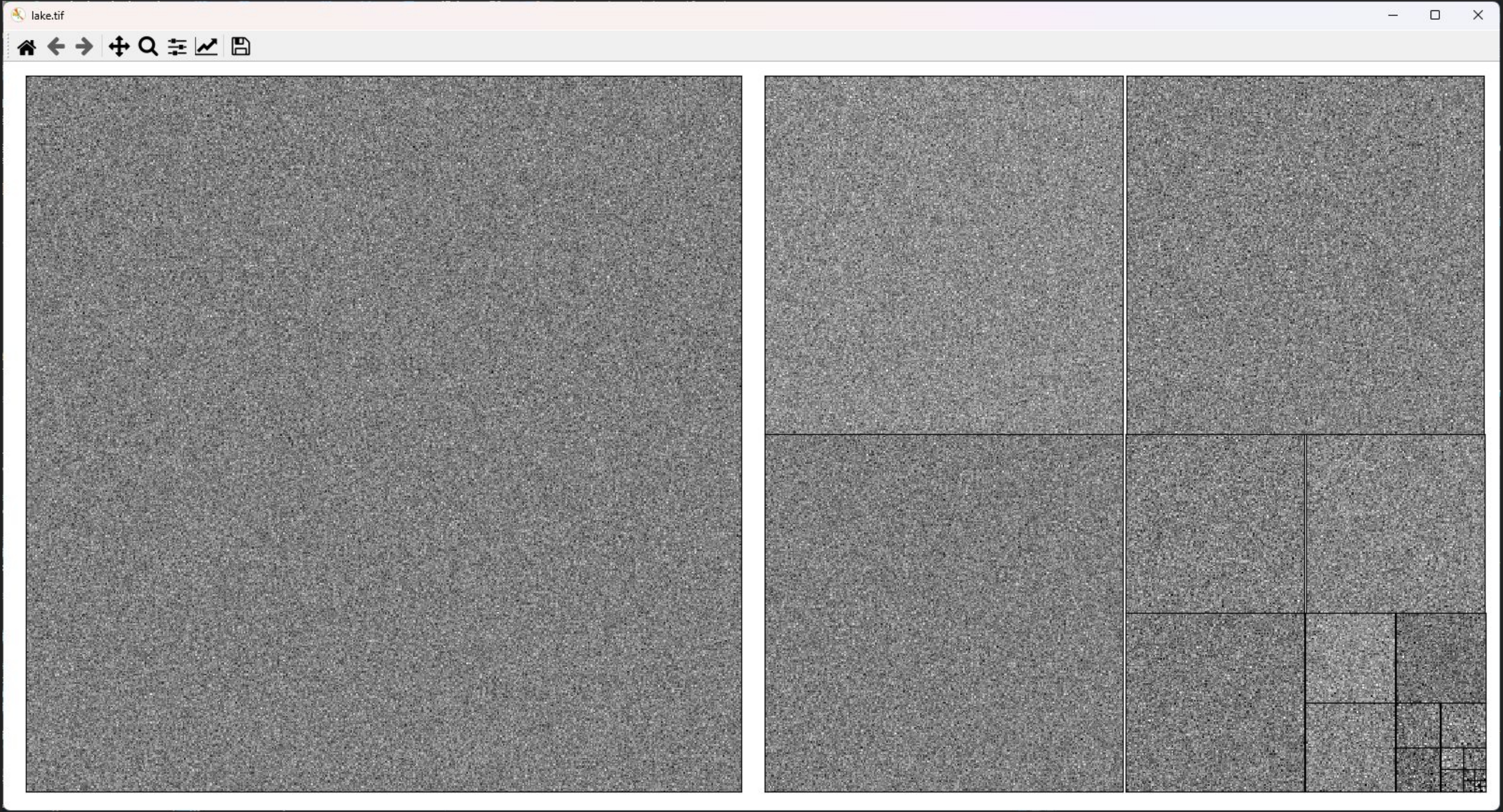

lena_gray_512.tif
x=113.6 y=177.7
[-5.0]


lena_gray_512.tif
x=93.9 y=46.0
[7.0]


livingroom.tif
x=78.4 y=196.3
[-10.0]


lake.tif


cameraman.tif
x=395.4 y=191.4
[165.0]


lake.tif