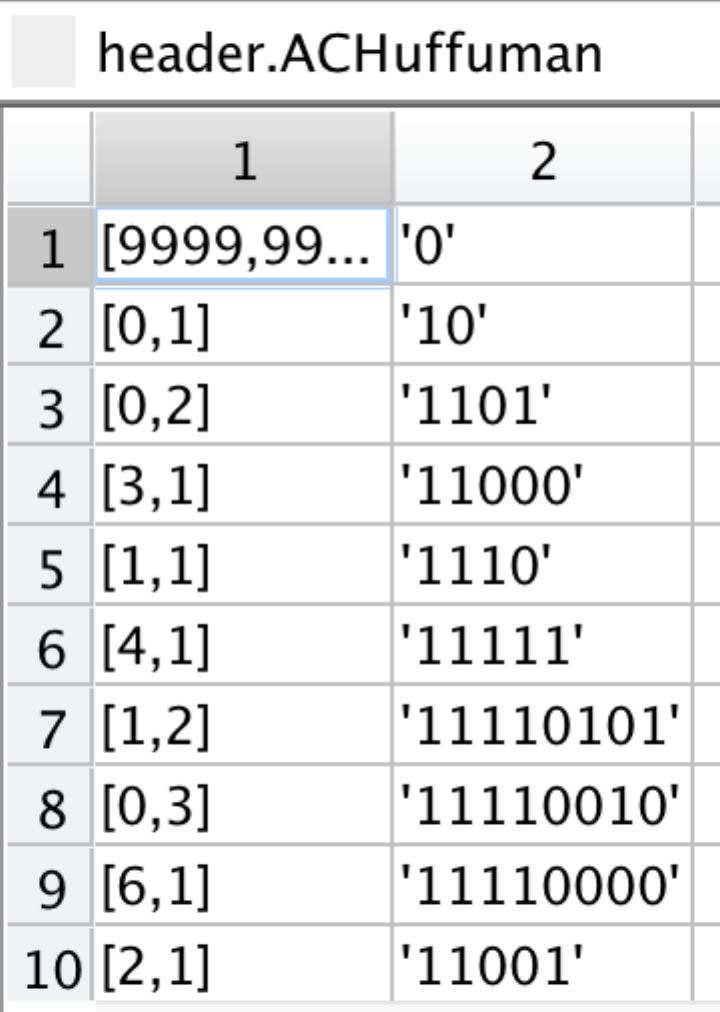
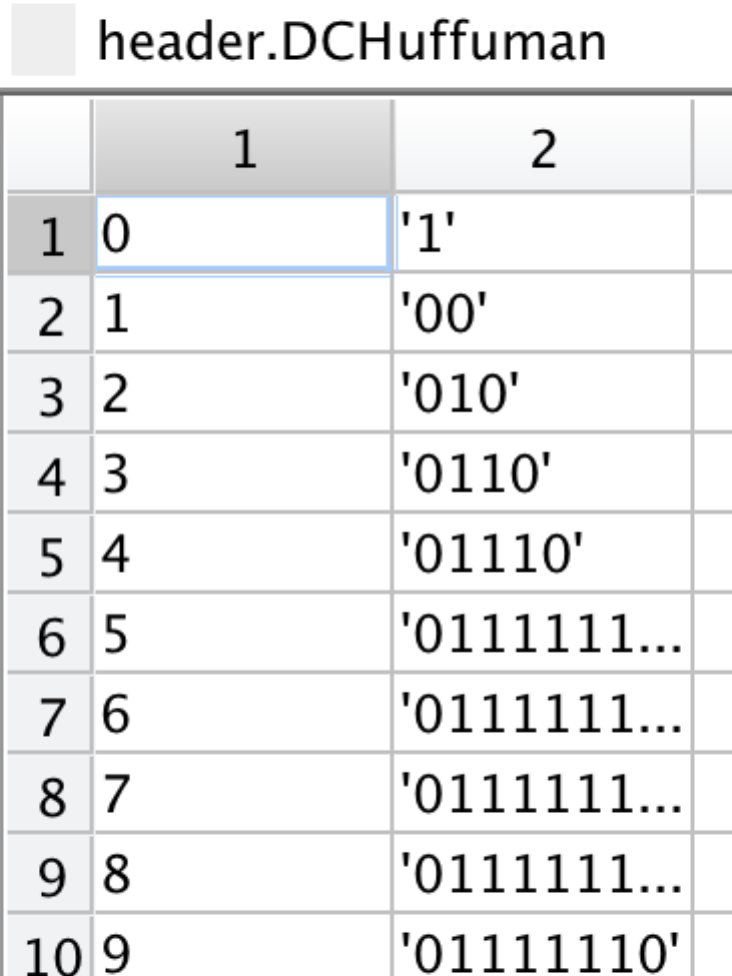
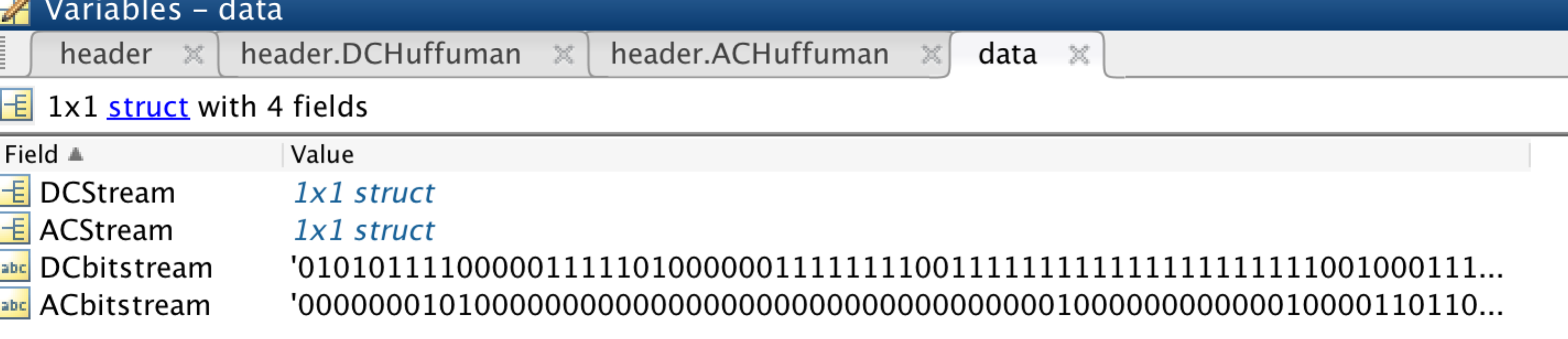
Data Compression Assignment 4 Report

Yixin Yin

In order to calculate the compression ratio and snr accurately, we need to get the real bit stream for dc terms and ac terms. I implement the JPEG fully based on the steps in our course site. My code is attacked with my submission. You just need to run the Homework4.m and Homework4\_plot.m to get the following results. Note that you need about 40 seconds to get the results of the Homework4.m because I keep all the needed information in MyJPEG function and it needs some time to calculate. In my code, I use 9999 to represent EOB for ac terms. Here is the Huffman table structure.



Here is the data structure:



I output the ac term and dc terms stream into two .txt files so that it is clear.

Here is what I ‘ve done in this assignment.

1. G = G -128;
2. Dct(G);
3. Quantize G by divide the quantization\_matrix\*scalar;
4. Encode the dc terms and ac terms separately (need to calculate for some time);
5. Get (k,s,t) for DC terms and (d,k,s,t) for AC terms;
6. Build huffuman tree for DC terms and AC terms;
7. Get the (h,s,m) for both DC terms and AC terms, h is the codewords , s is the sign bit and m is the binary representation for t;
8. Get the bitstream for AC terms and DC terms by connect all the (h,s,m);
9. Then decode the bitstream and get the rebuild image.

(10) calculate compression ratio and snr according to different scalars.

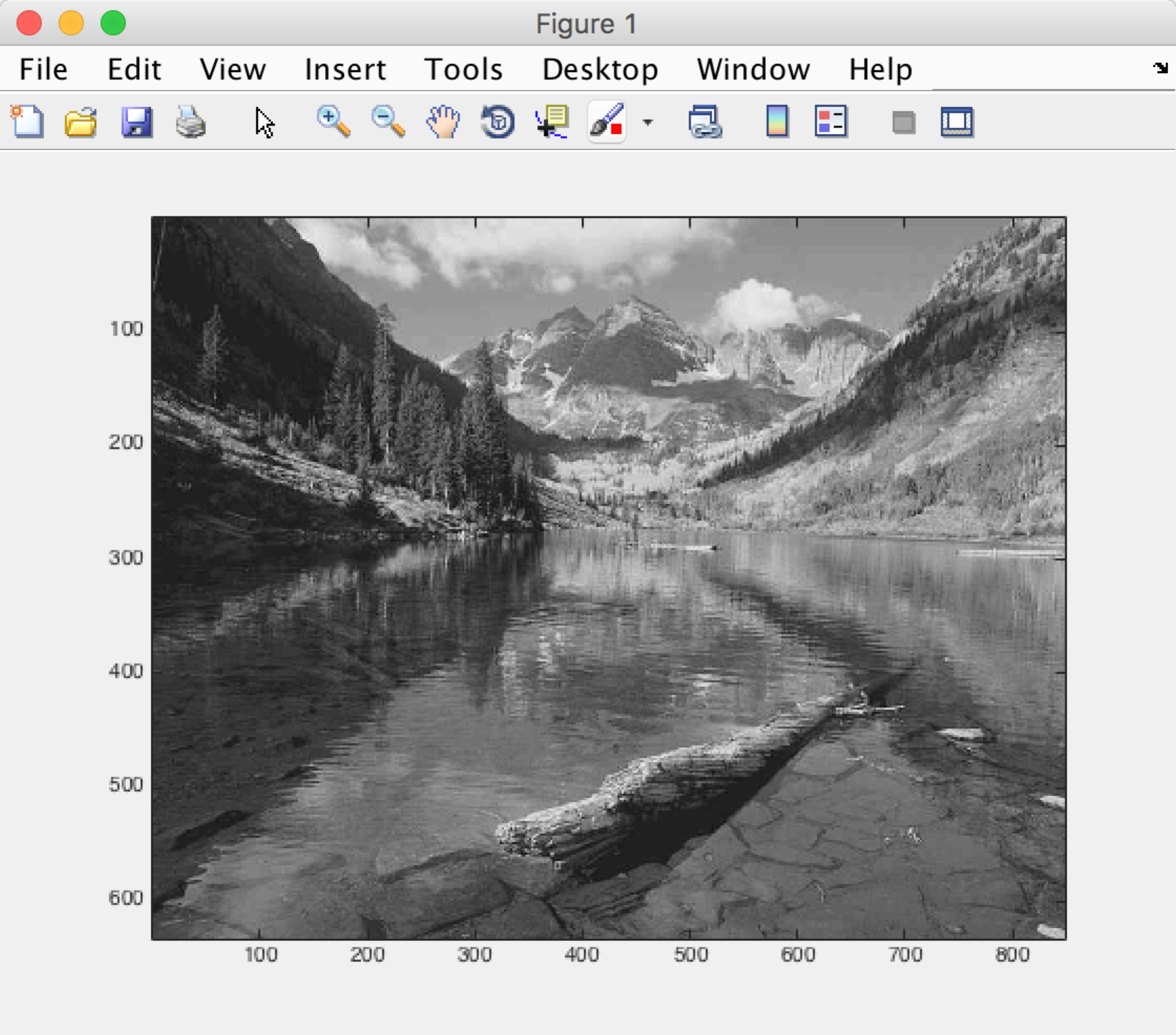
For river.gif

Compression ratio = 12:

When scalar = 1.666, we get:

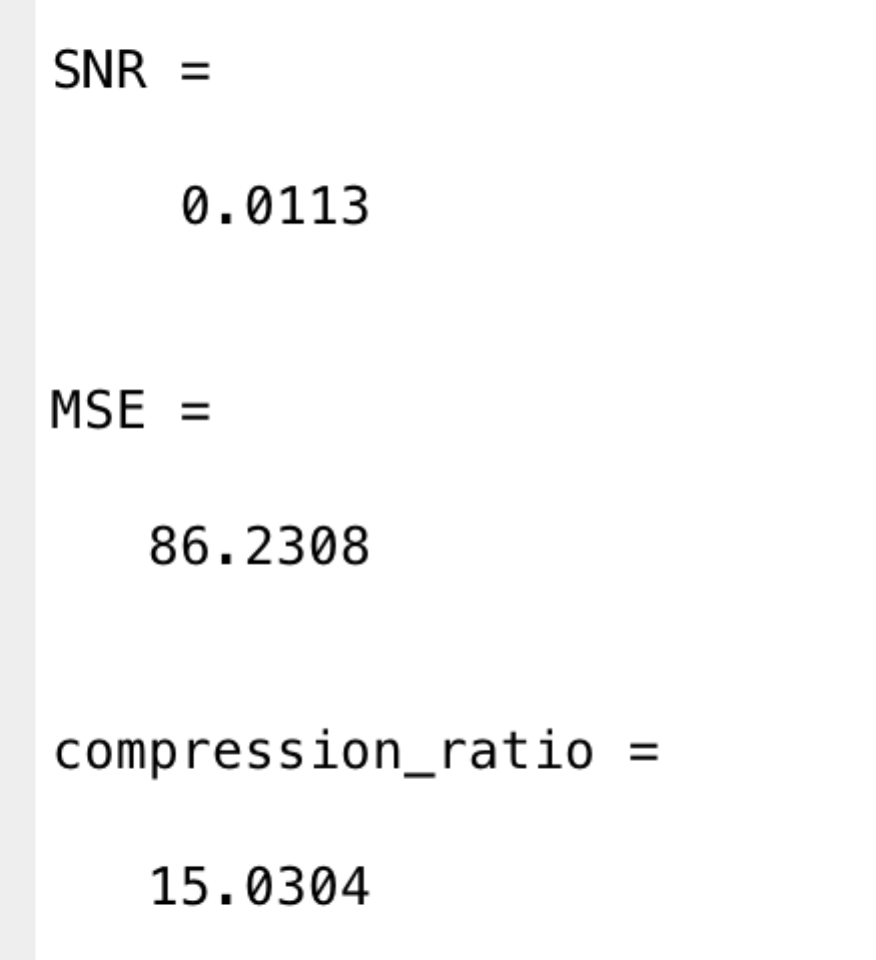


the effect of reconstruct image:

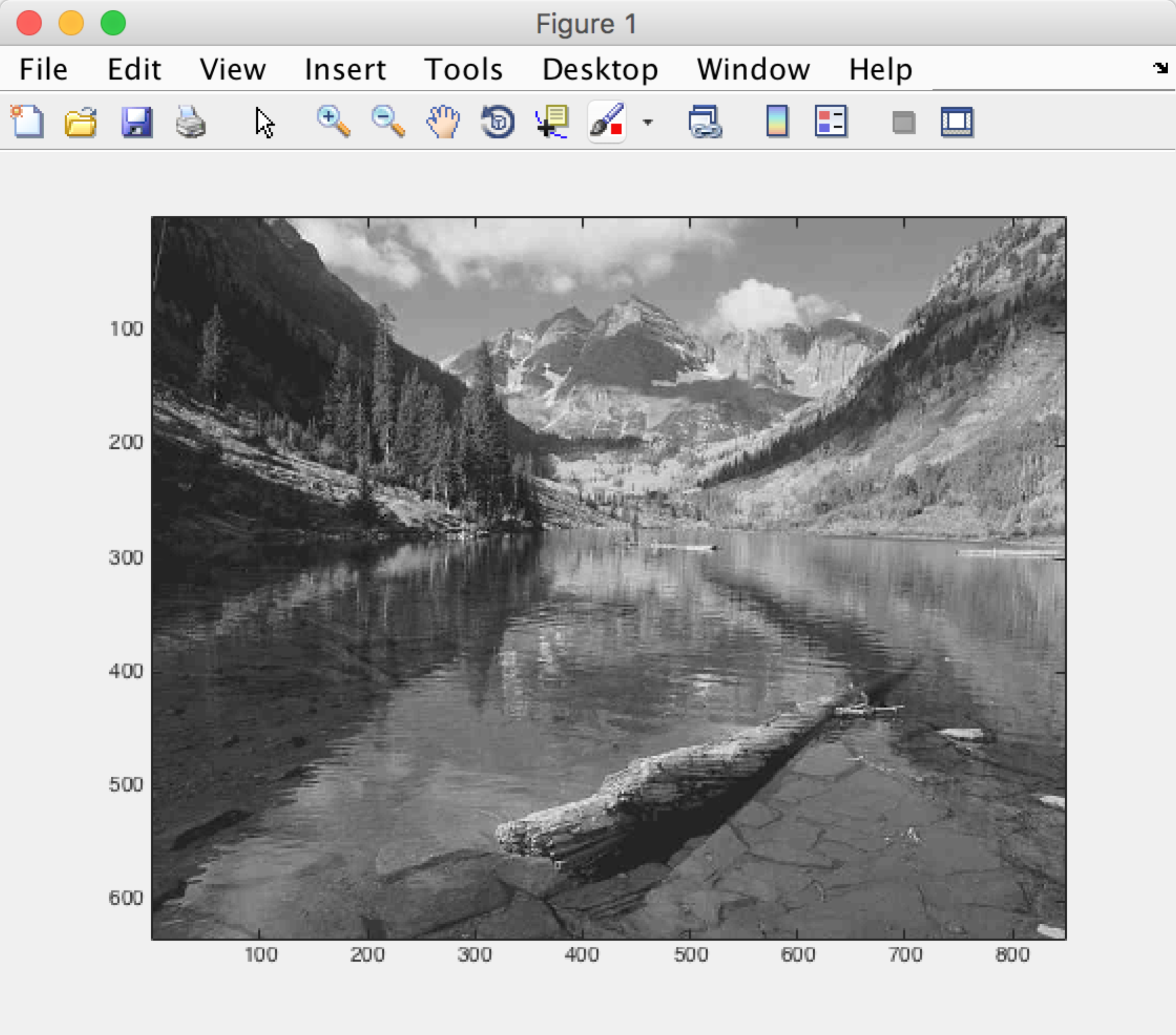


Compression ratio = 15:

When scalar = 2.21

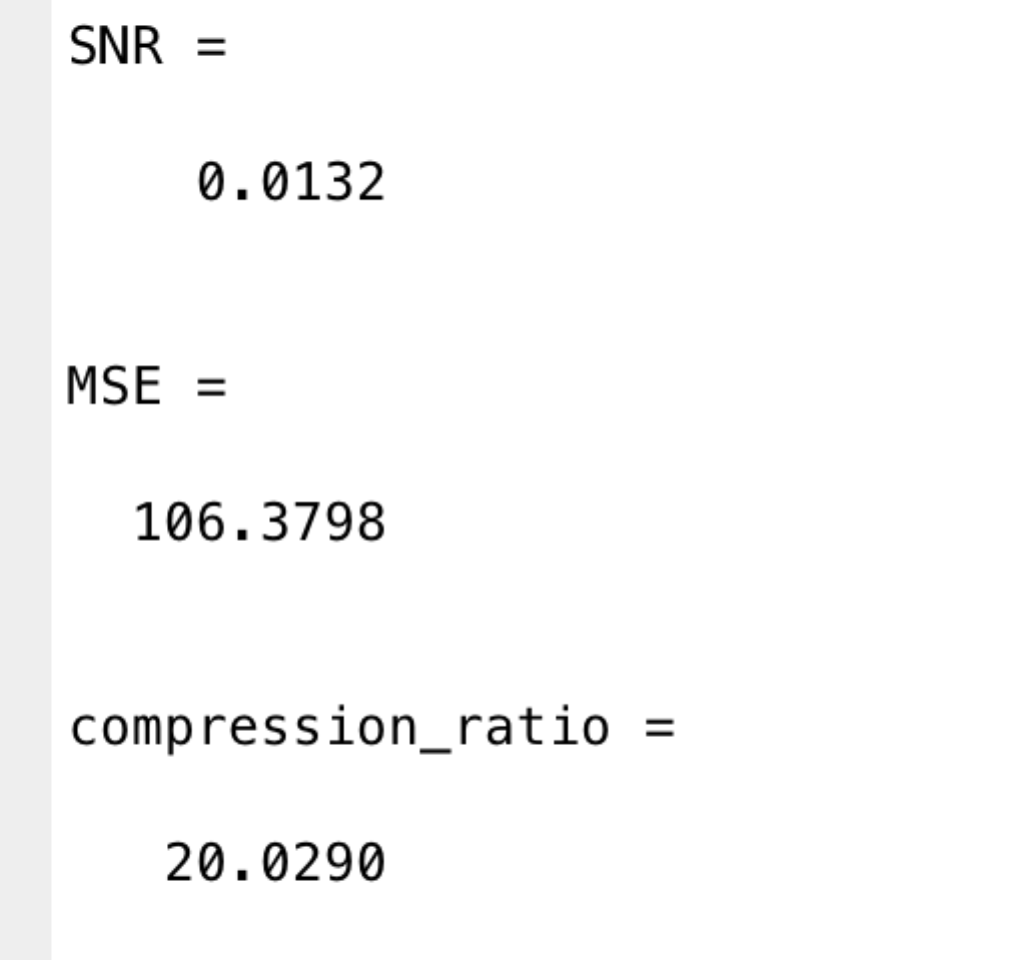


the effect of the reconstructed image:



Compression ratio = 20:

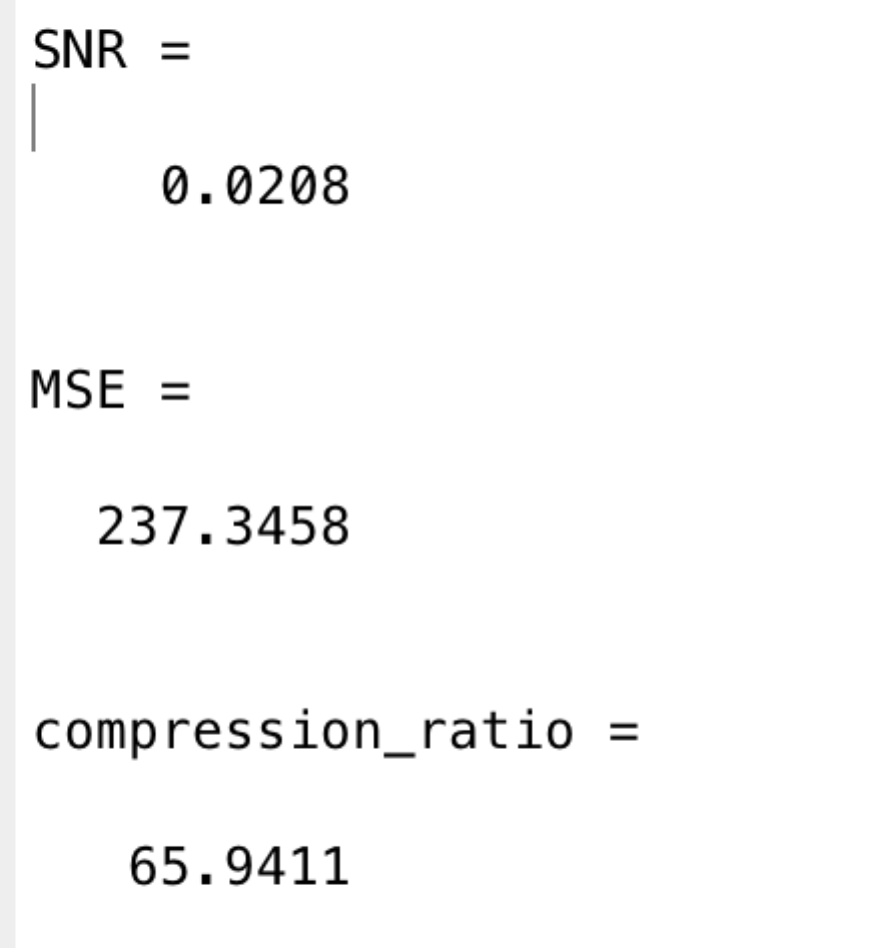
When scalar = 3.08



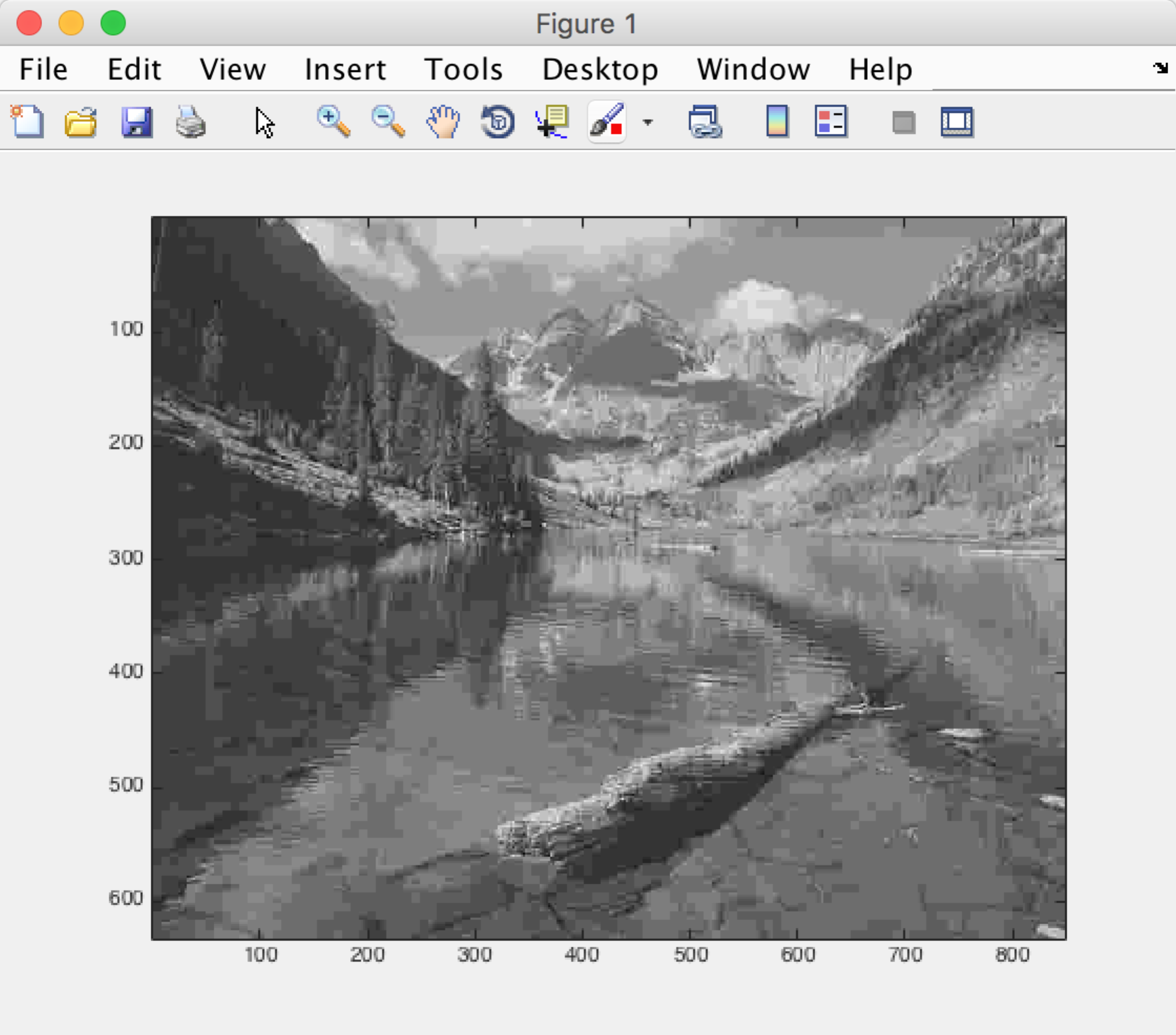
the effect is :



when scalar = 10:

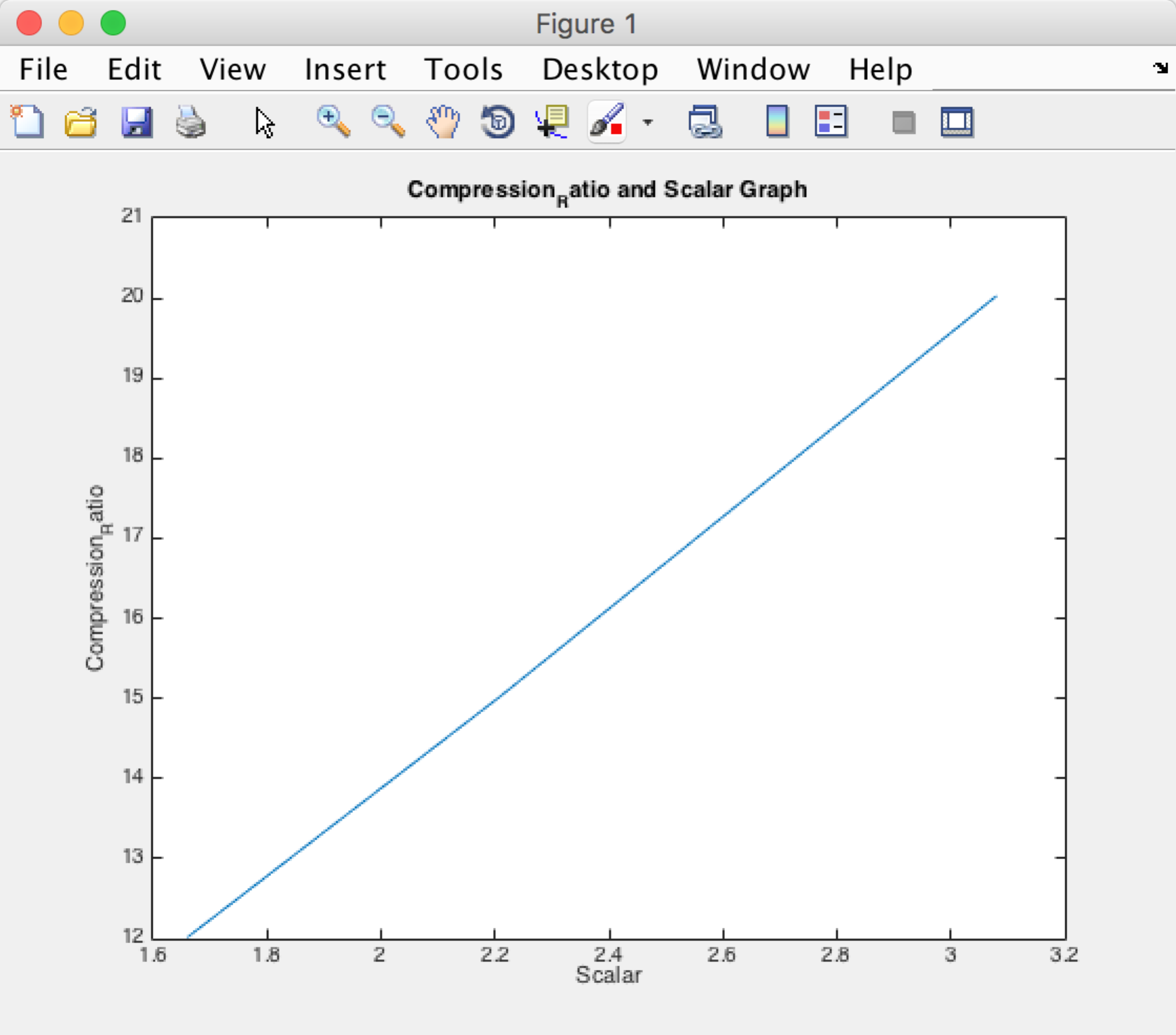


effect:

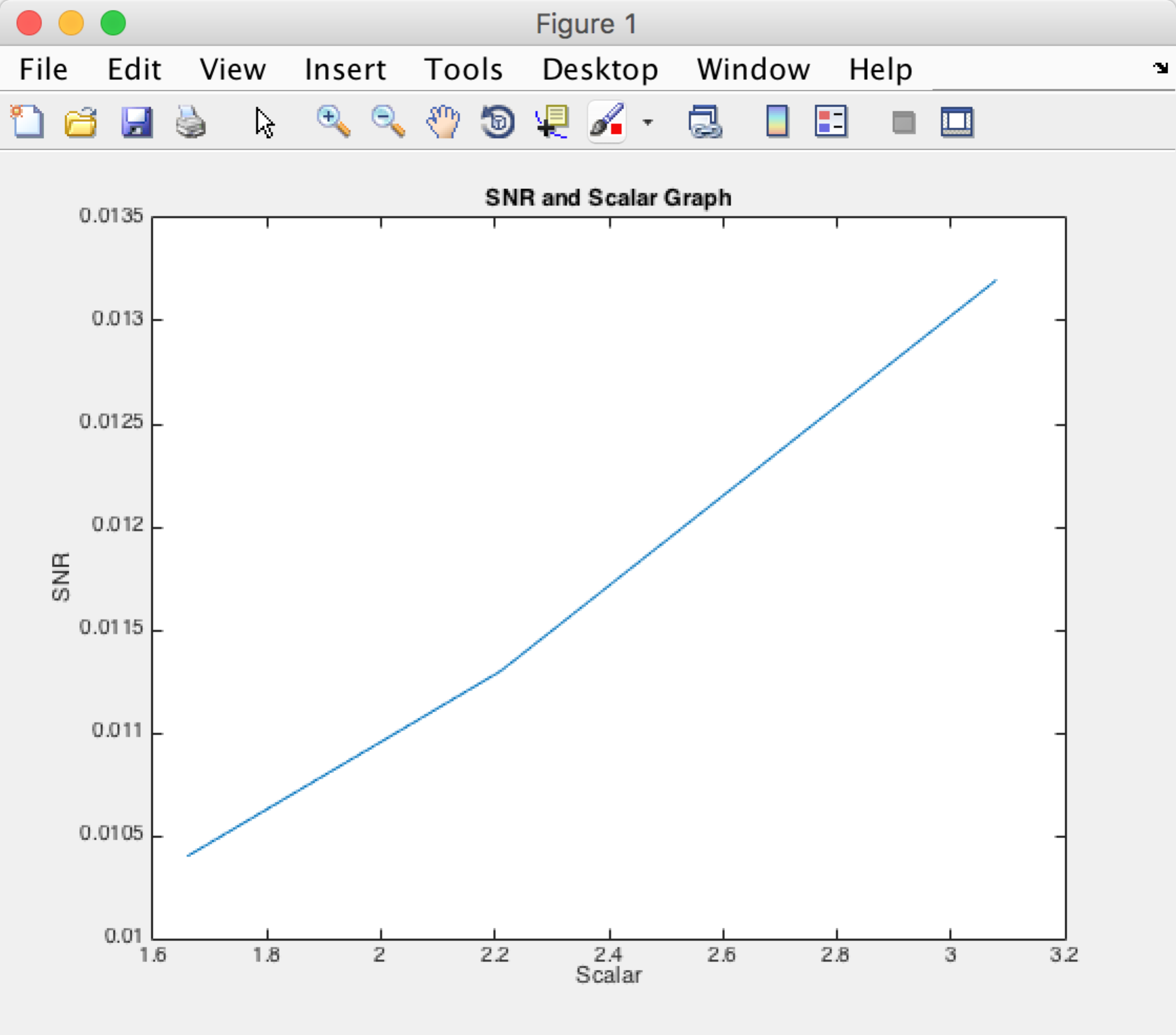


not clear.

Comprssion\_Ratio and Scalar Graph. The three point seems like liner.

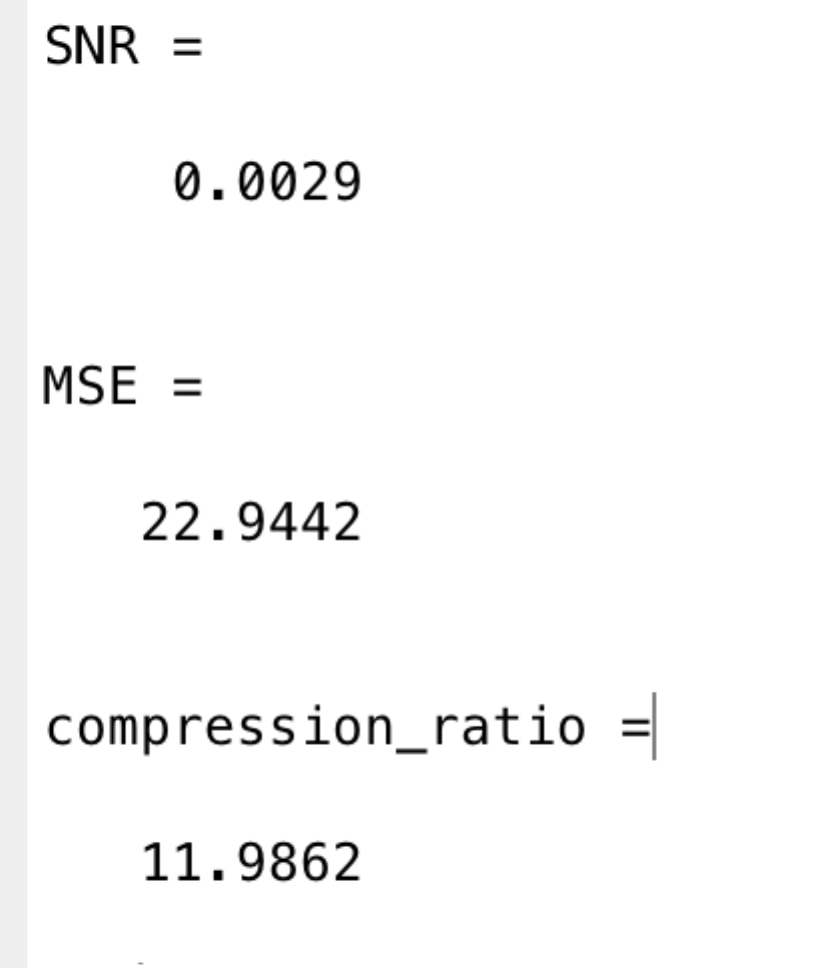


SNR and Scalar Graph

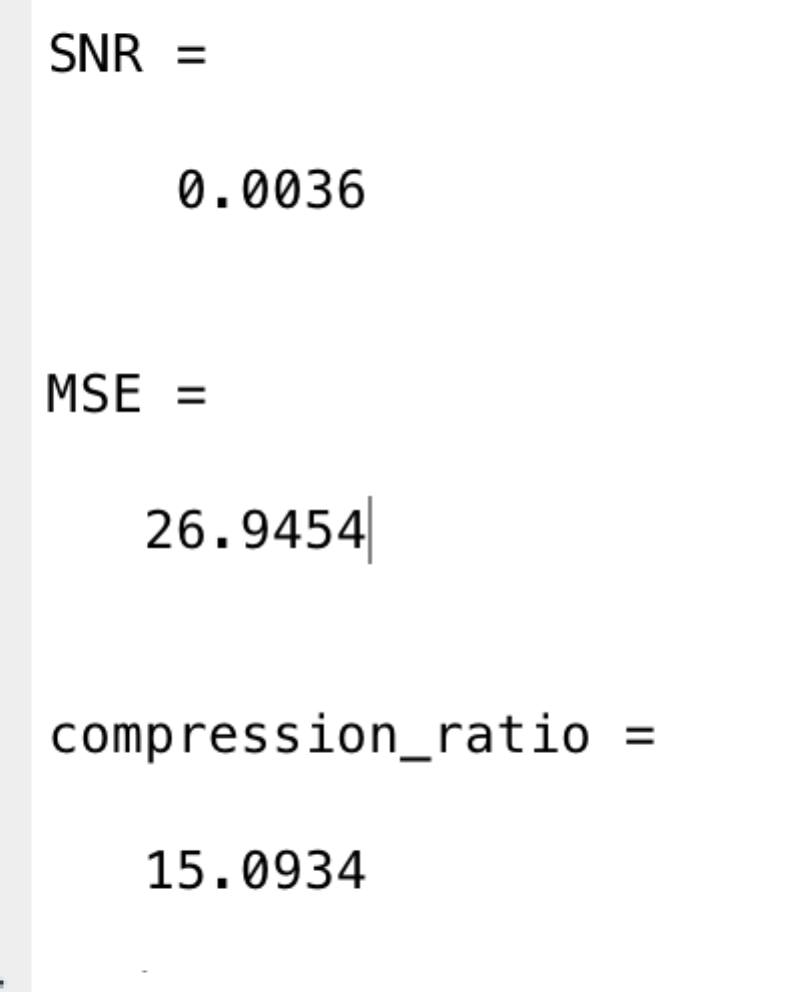


For Lena.gif

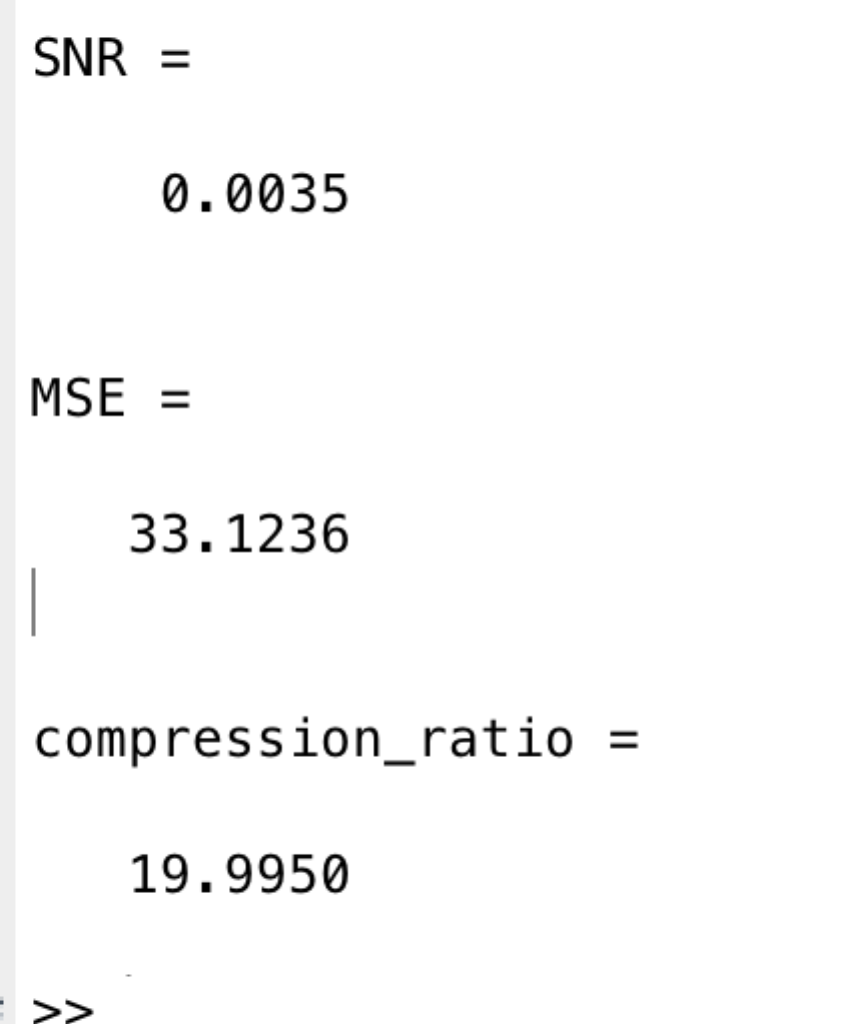
When scalar = 0.89



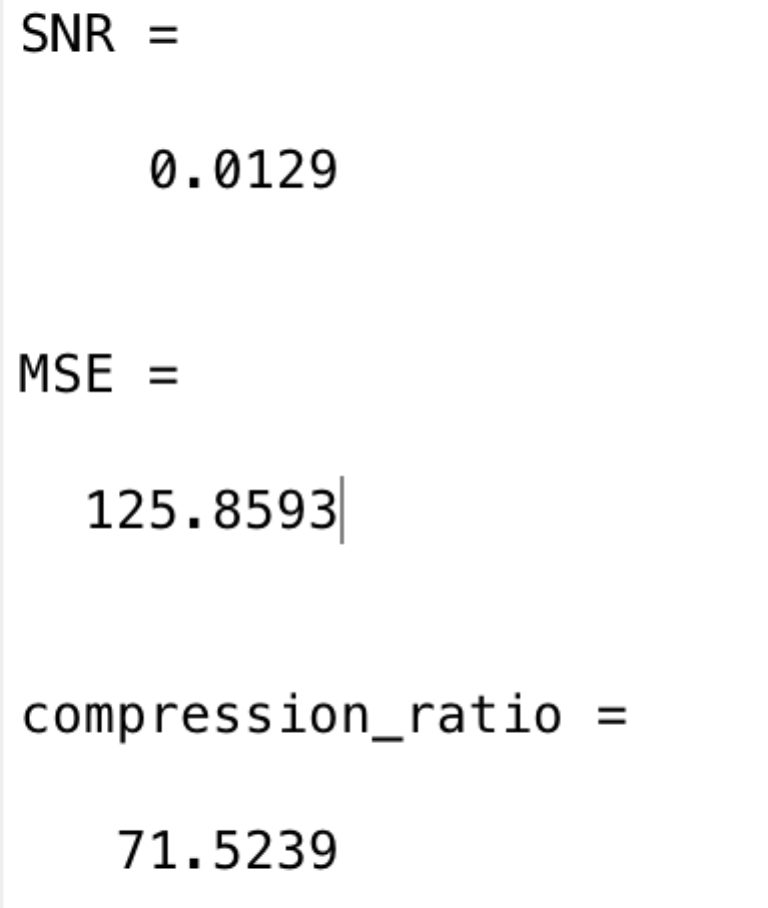
when scalar = 1.23



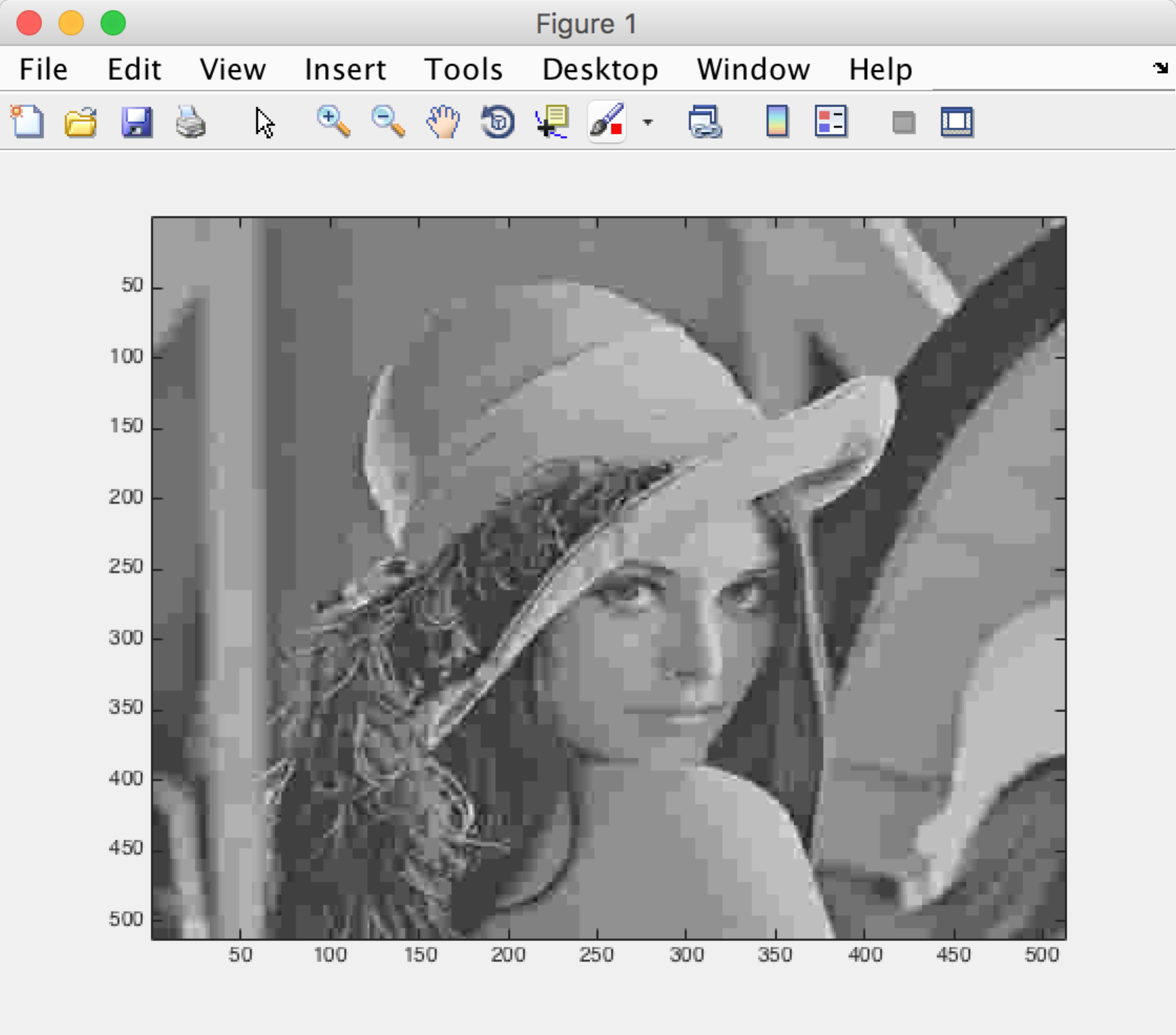
when scalar = 1.82



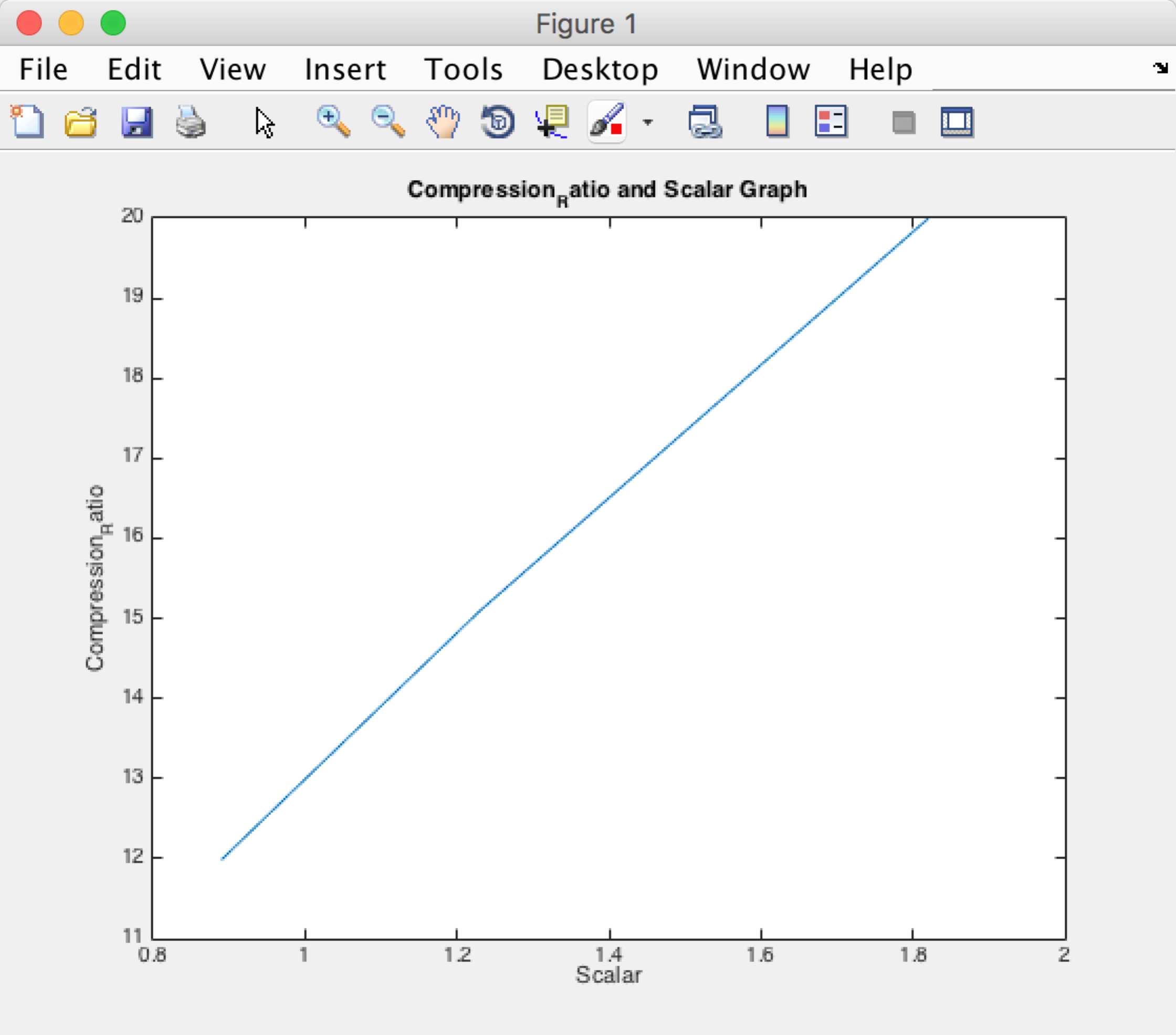
when scalar = 10

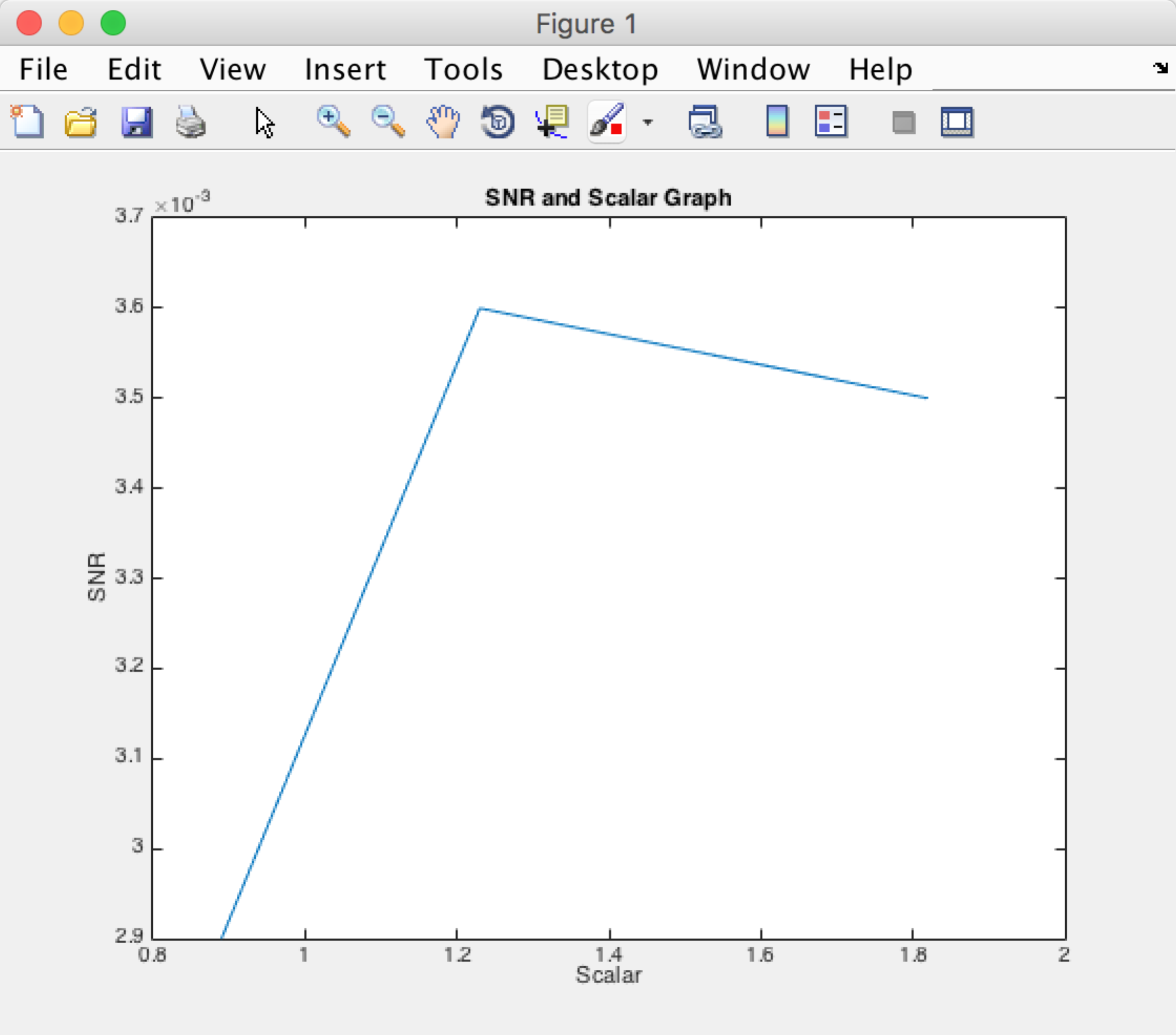


the effect is like:



the two graphs are like:





In the end, I also implement the version without entropy coding. It is easy but I don’t know how to get the real size of the bit stream accurately. So I just submit it.