Introduction to Scientific Computing
Report Practical : Tomography for binary pictures
Denis Garabajiu S4142551
Constantin Cainarean S4142152
Feb 16, 2021

Assignment 1 Computation of the horizontal and vertical projections

a. The requirements for this part are pretty clear and can be easily done being familiar with some functions of the Matlab. To compute the sum of the columns we have used the function *sum* with the 2D array **A** as parameter, and to compute the sum of the rows we have used the same function, but with the flag **2**, and the same 2D array as parameter.

b. Saving the above sums into a file was done using the *save* function in Matlab, specifying the file name (*rectangle.png*) and the parameters that need to be saved (colsum and rowsum in our case)

c. The before-mentioned step was done in the subpoint c of the assignment as well, but this time for the other two images : *oval.png* and *rock.png*. We decided to use for separate images separate 2D arrays (**B** and **C**) in order not to overwrite the information in a single array.

Disclaimer : The code for both the assignments 1 and 2 can be found in .mlx file inside the same archive!

Assignment 2 Computation of the horizontal and vertical projections

In the second part of the assigment our task is to implement the Kacmarz reconstruction algorithm for binary images in Matlab

a. The main task is implementing Algorithm 2.2 from the slides or in the reader of the course. Having a pseudo-code for the algorithm and the tips, made the implementation pretty straight-forward, however we would like to mention some design decissions that we have chosen :

   - Because we had to reconstruct 3 images, we decided to implement the algorithm in a function so when we will need it we can just call that function.
   - When reconstructing the image we work with 2 matrices, one is f_now which represents the state of the image at iteration x and f which will be the next state of our reconstruction
   - We also made some small adjustments to the algorithm that doesn't influence it's potential: in the line where rounding takes place we excluded the unused additional variable and we worked directly with f_now, we were able to do that due to the fact that matlab is a variation of C programming language.

b. In the second step we applied the algorithm implemented before on the image *rectangle.png*. Using the Matlab function *imwrite* we are saving the result of this operation into a file with PNG format. To make it convenient when we carry out several experiments with different values, we also add the final value of **k** (iteration number) and the used value of epsilon (variable **E** in the program) - threshold of accuracy to the file name.

c. In this subpoint of the assignment we compute the relative error E between the input array A and the reconstruction *f_init*. This is done using the given formula and the functions *sum* and *abs* in Matlab.

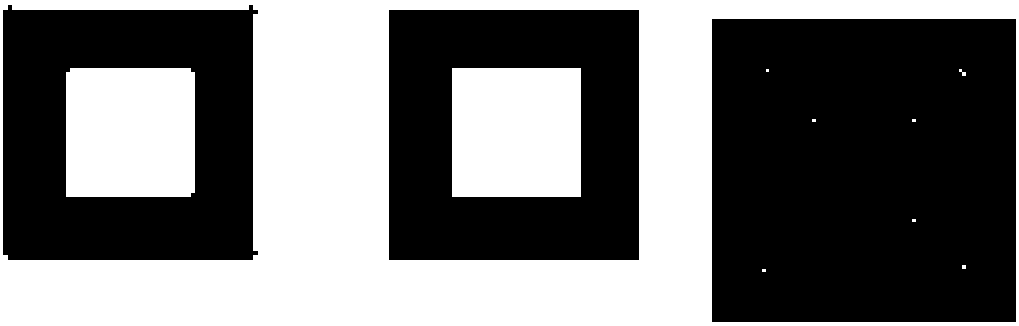
d. In this subpoint of the assigment our task is to see at which pixel locations the differences between input and reconstruction occur, for convenience we compute a difference picture. Since we are working with boolean values 0 and 1 (white and black respectively), if we follow the given formula we might run into the issue that we will have a new value -1, which upon conversion to PNG file will be 0, and will mislead us that there is no difference when there actually is. Therefore we use the absolute value function in Matlab (*abs*) in order to avoid a possible bug.

e. Due to the fact that we are using a function we can easily repeat the above steps for the images *oval.png* and *rock.png*, all we have to do is to call the function *reconstructImage* with the name of the file parameter.

f. Our experiments consisted of two parts : the first one was the computation of the horizontal and vertical projections that will later be used as input in the second part which is the Kaczmarz algorithm. At the beginning we had 3 images a rectangle, rock and oval, each of them was converted into a 2D array based on the color of the pixels. The array was filled with 0 if the color of the pixel was white, and with 1 if the color was black. After the conversion, we have calculated the sum of the columns and of

the rows.Our next step was implementing and applying the Kaczmarz algorithm for reconstruction of a 2D binary image. During the experiment we have also calculated the relative error for each image. It ranges from 0 to 1, the lower it is the more the reconstructed image is identical to the original. Below we will attach the original, reconstructed and difference pictures for each of the given files. These were generated with the accuracy threshhold (epsilon) of **0.0001**.

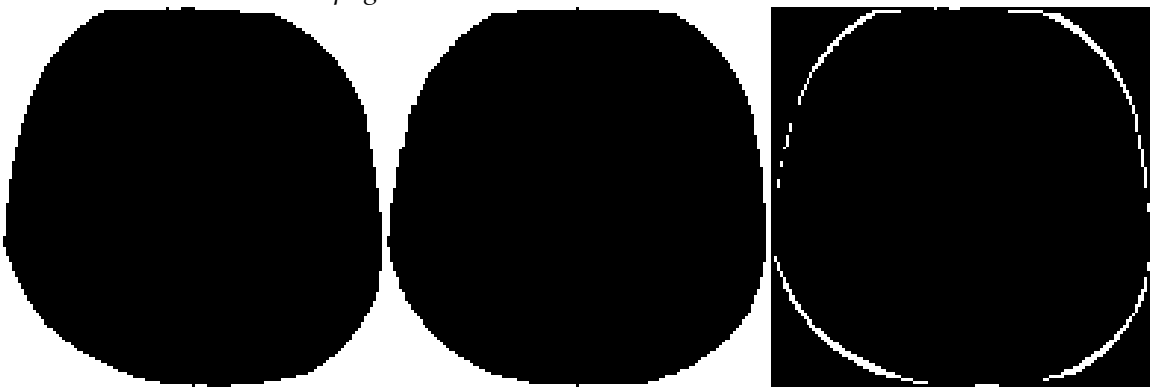
Rectangle :
The relative error for *rectangle.png* is **1.107266e-03**



Original Reconstructed Difference

Looking at the generated images and on the value of relative error we can conclude that there is little to no difference between the original and the reconstructed image, we can notice that the reconstructed image is even a clearer representation of a rectangle, the error pixel in the original image from the corners have been ”fixed”. Looking at the difference picture, we have observed that only a few pixels are white, meaning that there is very little difference.

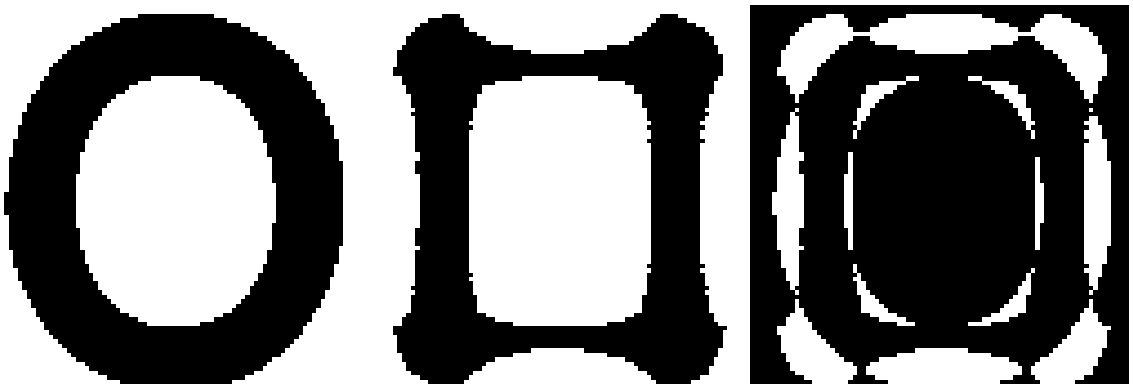
Rock :
The relative error for *rock.png* is **2.636719e-02**



Original Reconstructed Difference

Observing the generated images and the value of relative error of this file, we conclude that there is little difference, between the original and the reconstructed image, however more than on previously tested image.

Oval :
The relative error for *oval.png* is **2.412457e-01**



Original Reconstructed Difference

From the start, without looking at the generated images, we notice that the relative error for this file is larger than for the previous two files. When we take a look on the output images we clearly see that there is a significant difference between the reconstructed image and the original one. Taking into consideration the fact that we had 46 iterations at the given epsilon value and also the fact that each iteration differed a bit from the other ones, we made the conclusion that this is only one of many more possible reconstructions of the original images. Also we noticed that the shape of the rock and oval is similar, but there is a huge difference between the outputs of these two files, this made us think that such a difference is caused by the fact that the oval image is not a solid black figure, instead it has white pixels in the center, while the rock doesn't.