IN5520 Mandatory project 1

Espen Lønes

October 8, 2021

Task A:

Mosaic 1:
Top left / subimage 1:
Quite course texture, texel 'blobs' of varying size 1-10 pixels across. No distinct direction of texture so isotropic.

Top right / subimage 2:
Some darker and some lighter areas. More blurry than the others. Looks like there is a white/light grid across the whole texture. This grid consists of thin (2-4 pixel wide) horizontal and vertical lines.

Bottom left / subimage 3:
Mostly almost completely white or black. White background with mostly, almost vertical black lines. Most lines are at about 110/-70 degrees. The lines are 1-4 pixels thick.

Bottom right / subimage 4:
Similar to top left, but finer texture (smaller 'blobs'). The 'blobs' here look like a mix of different intensity lines going over each other in all different directions, so isotropic.

Mosaic 2:
Top left / subimage 5:
Consists of periodically alternating lines at 45 and -45 degrees. The lines are 3-4 pixels wide

Top right / subimage 6:
A mosaic of horizontal rectangles. Each rectangle consists of more than one gray level. All the rectangles are the same size 5x19, excluding the dark lines between them.

Bottom left / subimage 7:
Vertical lines. With small disjoint horizontal lines across the vertical lines.

Bottom right / subimage 8:
Very similar to mosaic 1 top left, course, isotropic, 1-10 pixel wide 'blobs'.

Task B:

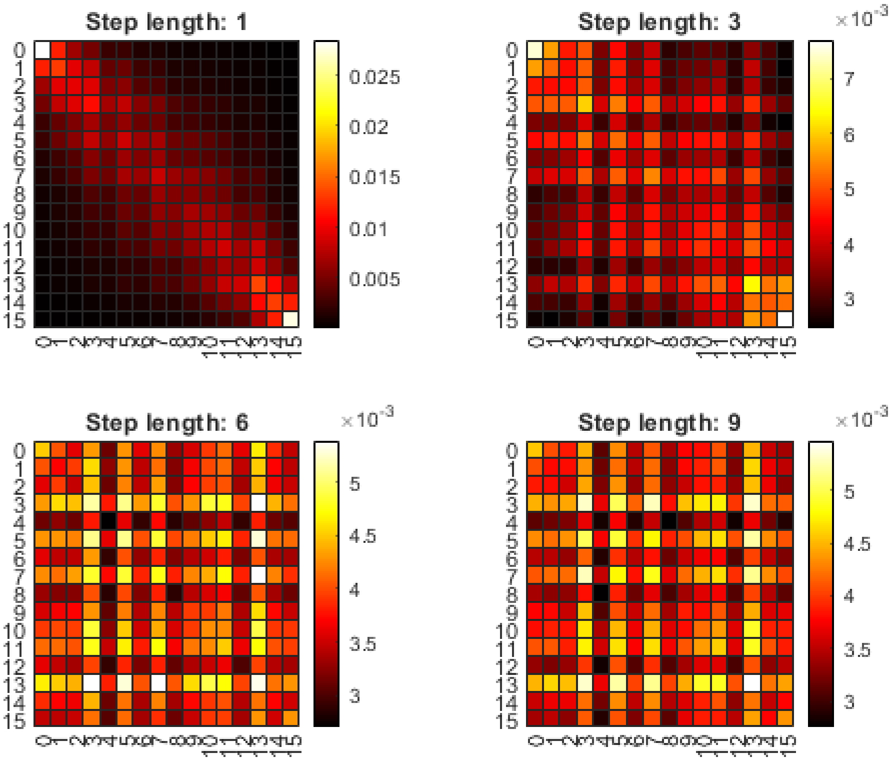
I chose to do histogram equalization on the sub images to enhance contrast. This makes parts of the textures with similar gray levels to less often fall on the diagonal of the glcm. I think this will help create more dissimilar glcms. Since the glcm looks at changes in gray level, and the equalization makes those changes larger.

I use MATLAB's build in glcm function (graycomatrix). This takes in the parameters (d, θ) on the form (dy, dx), so i will use this format for the parameters/offset in the report.

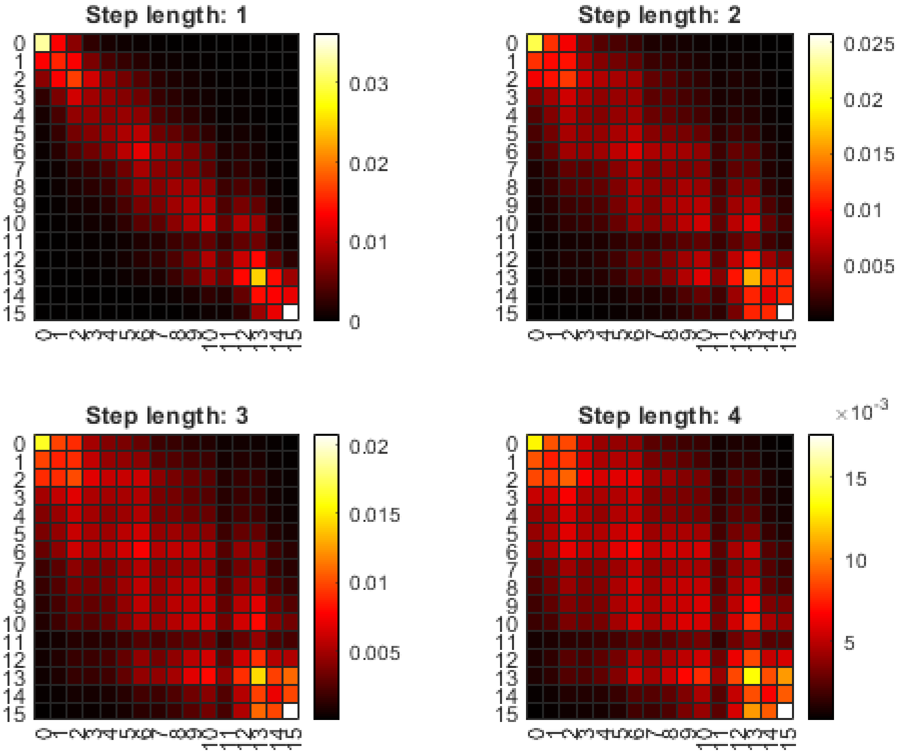
Subimage 1, is has isotropic texture so we should use an isotropic GLCM. This will be calculated by taking symmetric glcms for parameters $[0\ 1; -1\ 1; -1\ 0; -1\ \ -1]$, (0, 45, 90 and 135 degrees).

Summing them and dividing by 4.

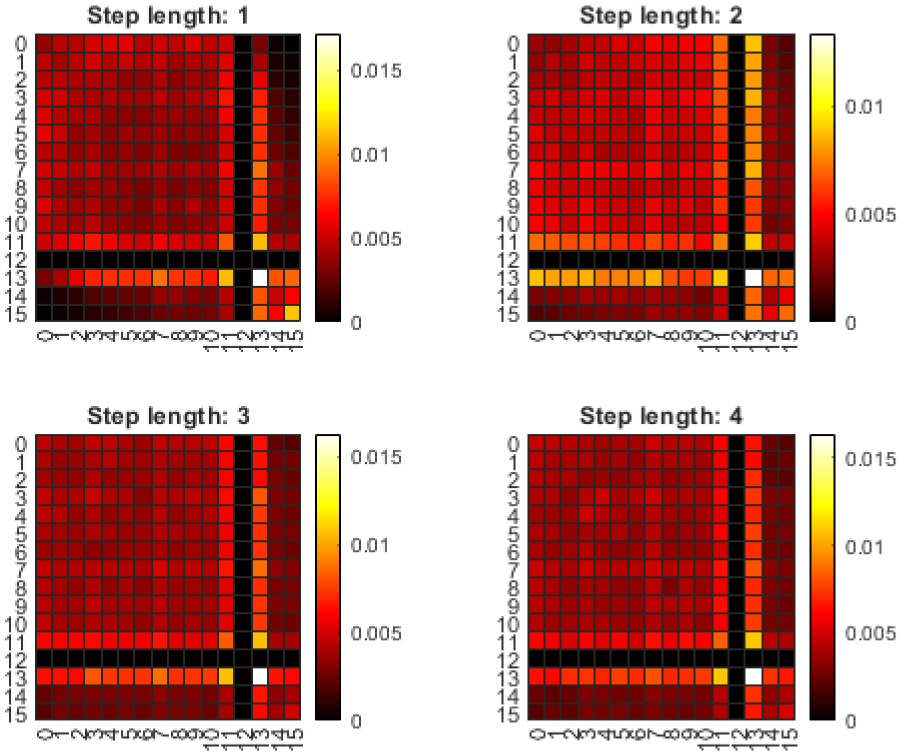
I will try with step lengths of 1, 3, 6 and 9. to see what gives the best result.



Subimage 2, has this underlying grid structure, i think using [0 1; -1 0] 0 and 90 degrees. (sum then divide by 2). With step length around the size of the grid lines 1-4.

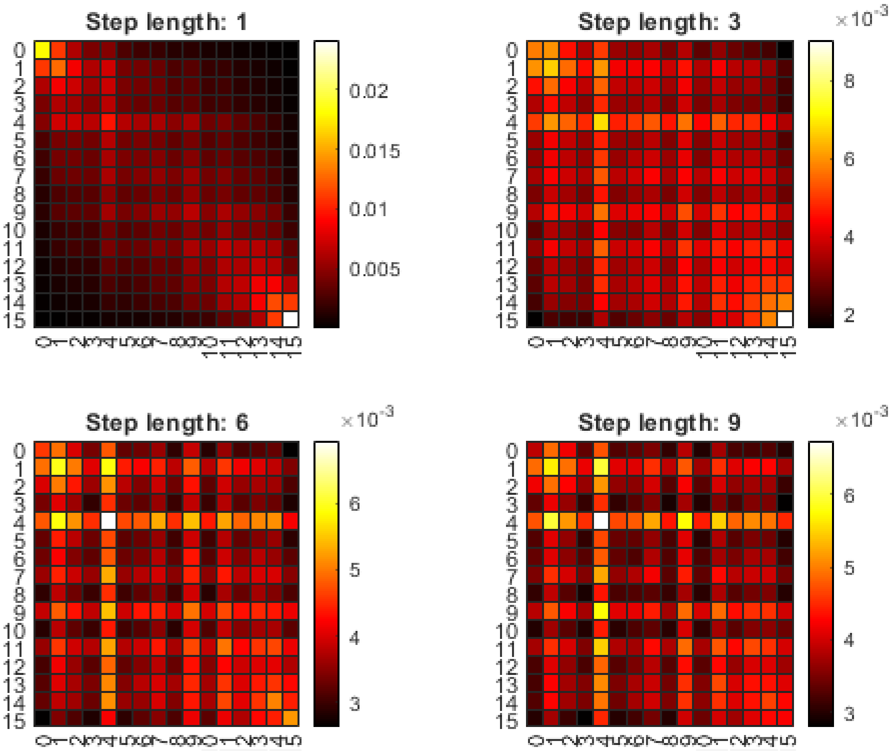


Subimage 3, has these black lines going through it. Using [1 3] (18 degrees). With step length 1-4 should give good results.



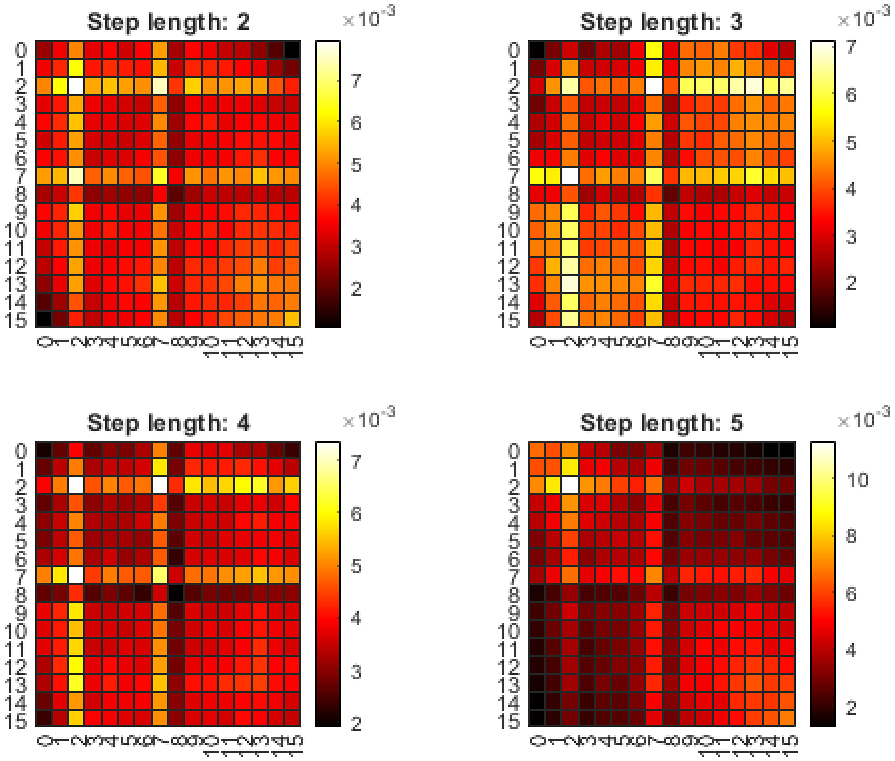
I think i have some problem with the requantization making none of the values go to 12. Creating the black cross. I am unfortunately unable to fix this.

Subimage 4, is quite similar to subimage 1, i will use isotropic GLCM with the same step lenghts as 1. And see if the different size of their texels will make the GLCMs different.

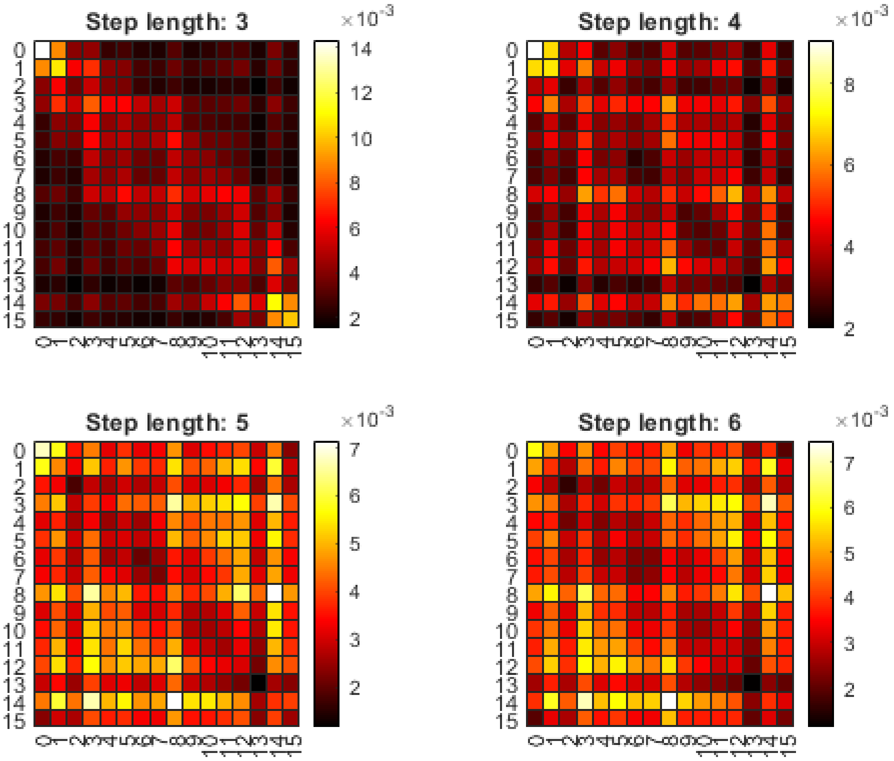


Comparing their GLCMs, step length 1 and 3 are quite similar but for 6 and 9 they are very different.

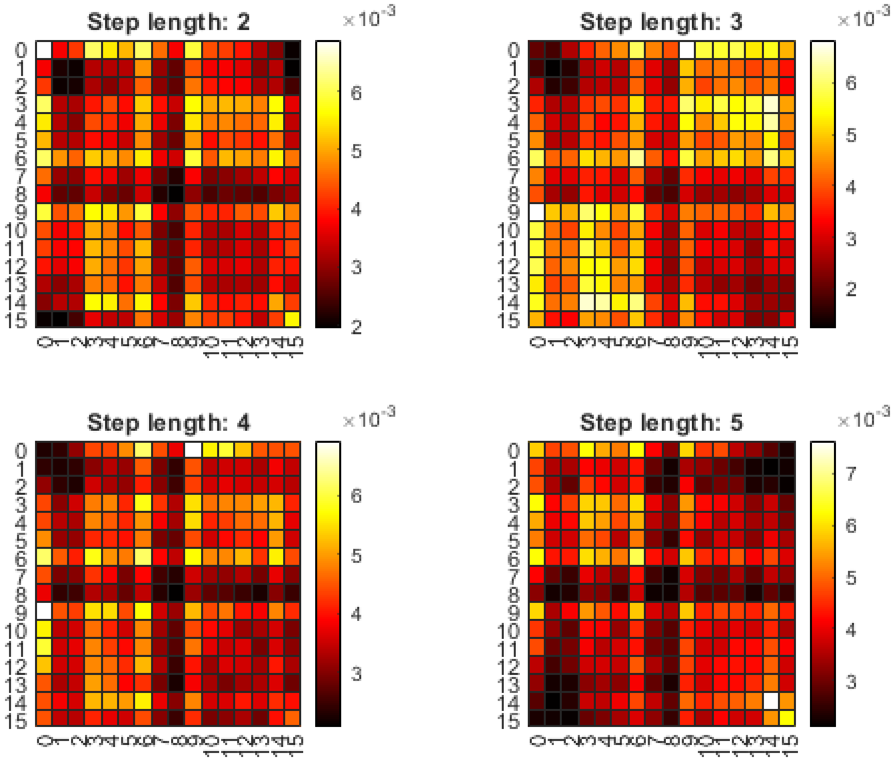
Subimage 5, had periodic lines at 45 and -45 degrees, i then chose to make a GLCM using offset going on the normal of these lines [1 1;-1 1] (-45 and 45 degrees). Then sum them and divide by 2. Due to the lines being 3/4 pixels wide i will test for step length 2,3,4,5.



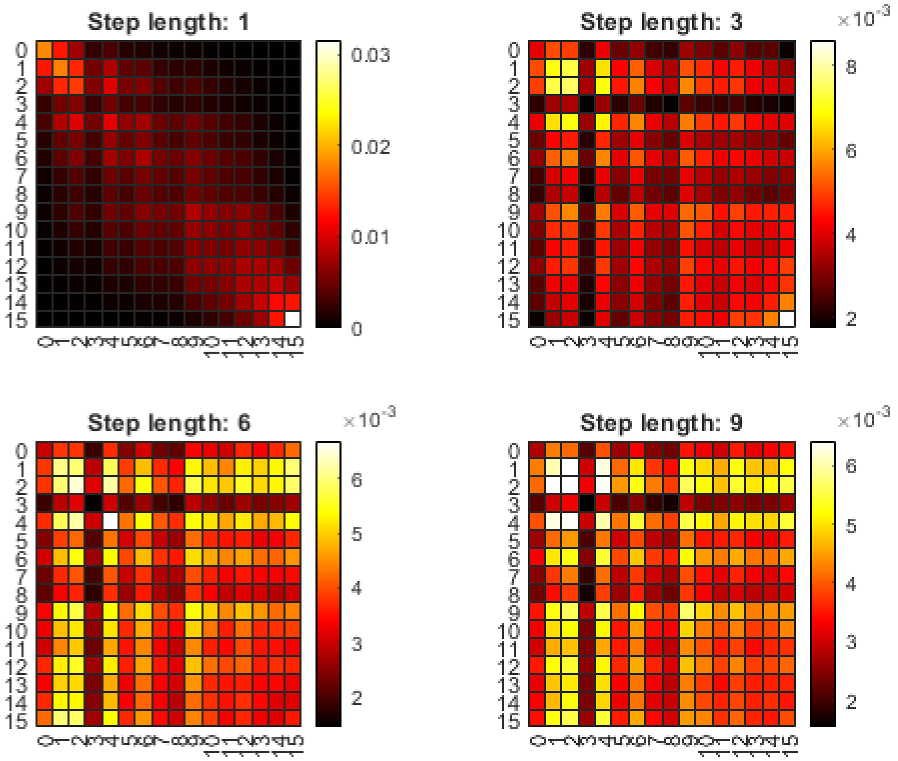
Subimage 6, due to each rectangle having a large intensity gradient within it. Making it change about as rapidly in each direction. I will use [0 1; 1 0], with step lengths 3,4,5,6. In an attempt to capture the information inside each rectangle and the change from one to another.



Subimage 7, i will go normal to the vertical lines, [0 1]. With step length corresponding to line with, 2,3,4,5.



Subimage 8, is very much like subimages 1 and 4. Therefore i will use the same parameters as for them. isotropic with 1,3,6,9 step length.

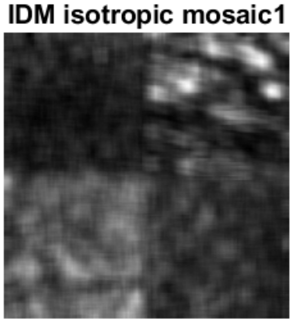


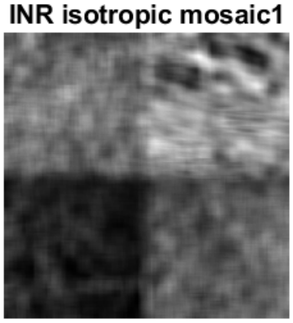
When looking at the GLCMs we see that independent of direction, for step larger than 1. All except subimage 2 are very non diagonal. This is probably due to subimage 2 being the only blurry image. Making the gray level transitions smother, resulting in a more diagonal GLCM. If we now look at the homogeneity/idm we see that GLCMs with concentrated diagonals will have higher idm values. A completely diagonal GLCM will have idm = 1. This makes sense with subimage 2 having a largely diagonal GLCM. As blurry images will naturally be very homogeneous.

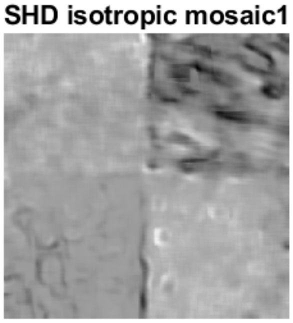
We also see subimage 2 and 6 had the same offset and was a little different but very similar fro step length 3. especial on having high values at the corners. We can see that this is because of a similar feature in both images. They both often go from very bright to very bright or very dark to very dark. We see this trend of high values in the diagonals corners for most of the subimages when they use a low step length.

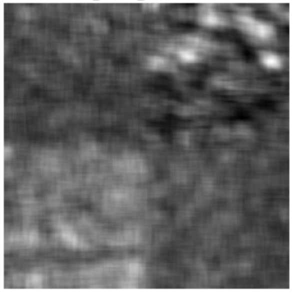
Task C:
I chose a window size of 31x31. Since we are trying to separate the textures, texture position is important so i chose a relatively small window in orderer to preserve spacial information.

For mosaic 1 i will use isotropic with step length 6 to separate subimages 1 and 4. Offset [2 6] for subimage 3. And [0 1;1 0] for subimage 2.

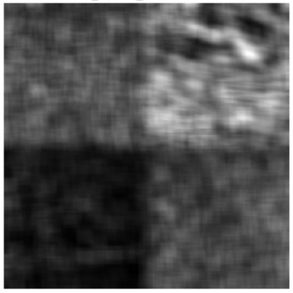

IDM isotropic mosaic1


INR isotropic mosaic1

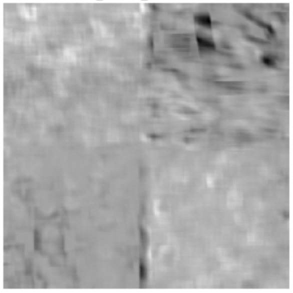

SHD isotropic mosaic1



IDM [2 6] mosaic1

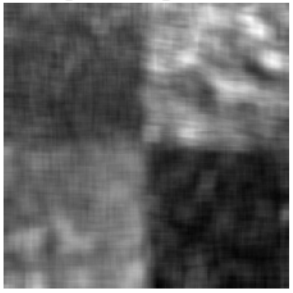


INR [2 6] mosaic1

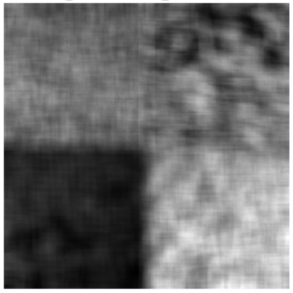


SHD [2 6] mosaic1

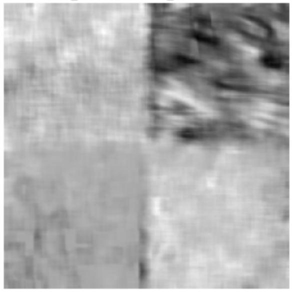
IDM [0 1; 1 0] mosaic1



INR [0 1; 1 0] mosaic1

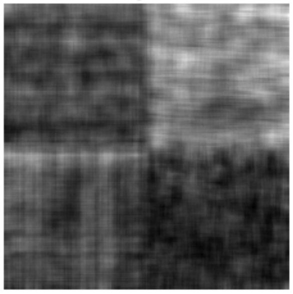


SHD [0 1; 1 0] mosaic1

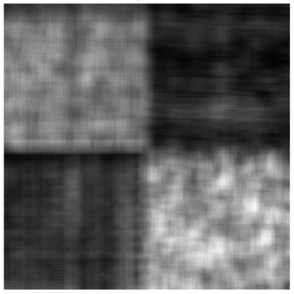


For mosaic 2 i will use isotropic with step length 6 to separate subimage 8. [0 3] for subimage 7. [0 4;4 0] fro subimage 6. And [3 3;-3 3] for subimage 5.

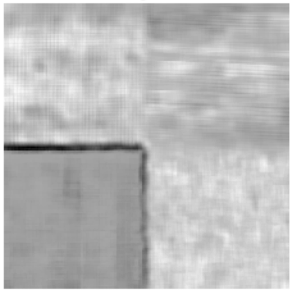
IDM isotropic mosaic2

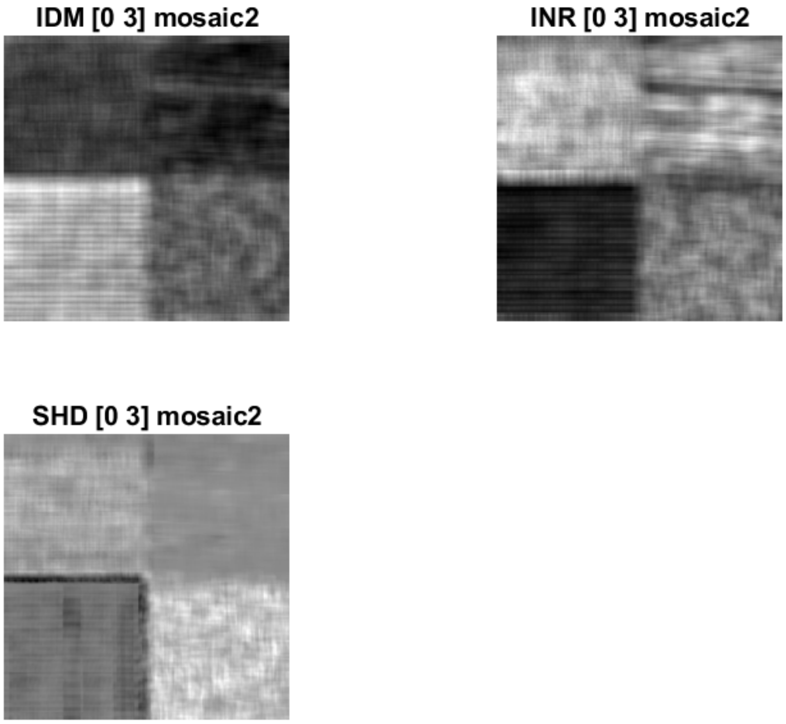


INR isotropic mosaic2



SHD isotropic mosaic2

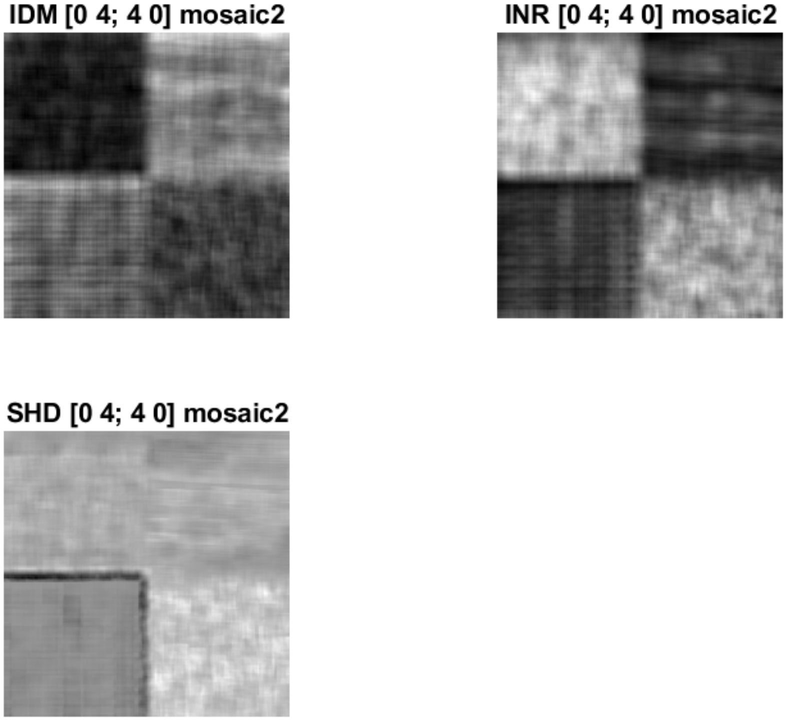




IDM [0 3] mosaic2

INR [0 3] mosaic2

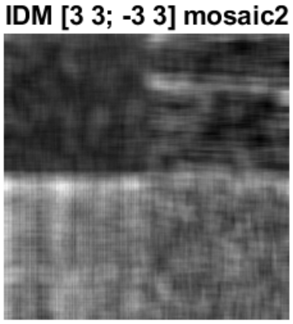
SHD [0 3] mosaic2

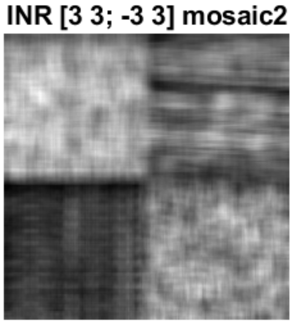


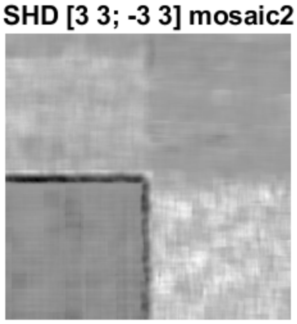
IDM [0 4; 4 0] mosaic2

INR [0 4; 4 0] mosaic2

SHD [0 4; 4 0] mosaic2


IDM [3 3; -3 3] mosaic2

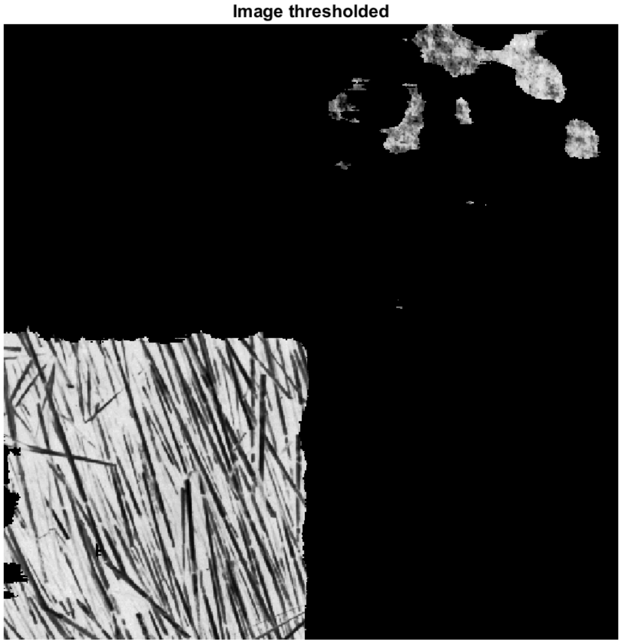

INR [3 3; -3 3] mosaic2


SHD [3 3; -3 3] mosaic2

Task D:

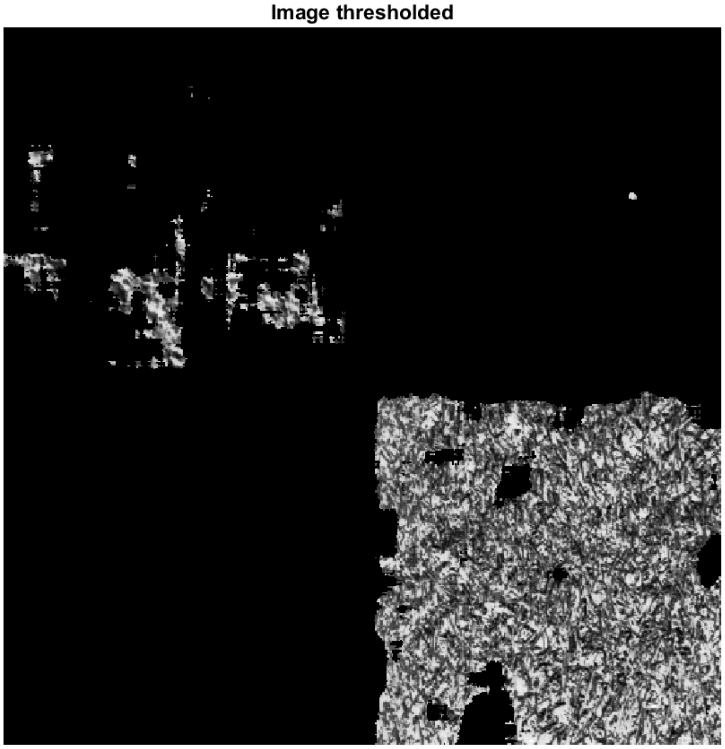
Mosaic 1:
Subimages 3 and 4 were fairly easy to segment out because they where very distinct in some feature images. 2 was harder but easier after having removed 3 and 4 from the picture. Segmenting out subimage 1 was hard and i could not do it without having a lot of noise. Even after removing most of the other subimages. How i did this is described below.

We see the INR images are well suited for isolating subimage 3. Trough some experimenting i found a good compromise in loss to noise ratio, (loss being parts the subimage not selected and noise being selected parts that are not the subimage). Using a threshold of 0.3 on INR [1 0;0 1]. Giving.



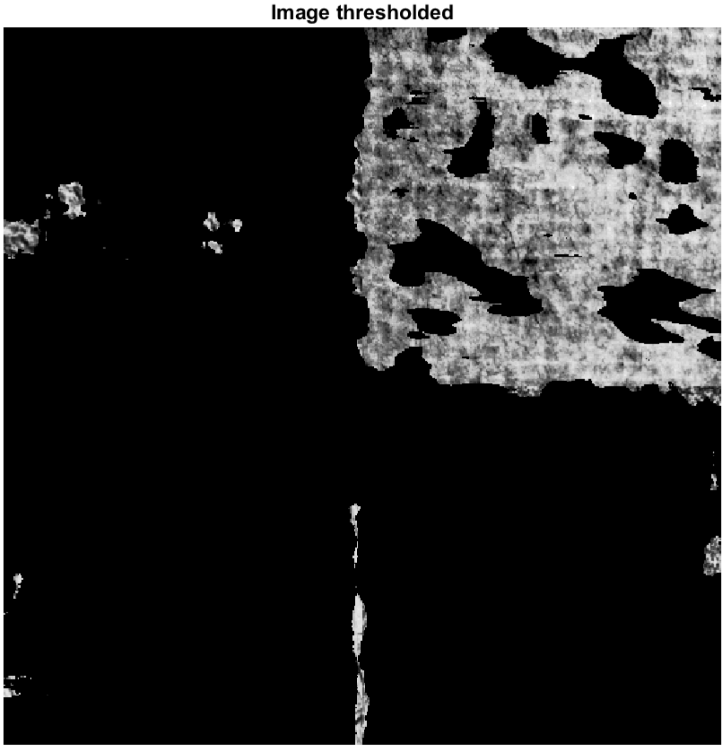
We see that it got most parts of subimage 3 but i had trouble removing all parts of subimage 2.
I then create a new image with this part removed.

It looks like IDM [0 1;1 0] can be used to segment out subimage 4. Again an ok noise to loss ratio with treshold at 0.54 (used on the new image) gives.



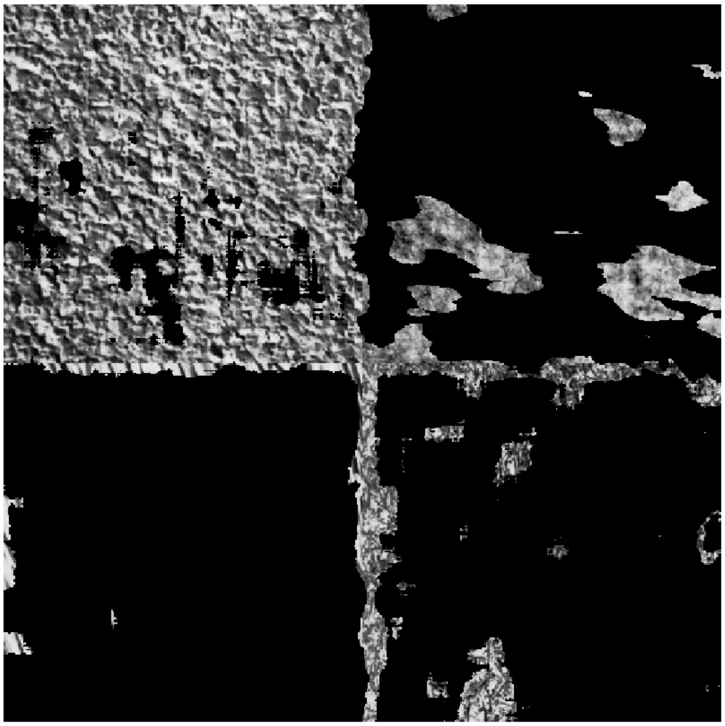
Here we have much more of both noise and loss compared to between 3 and 2.
I then create another new image by removing the subimage 4 part from the previous new image.

I then use SHD [0 1;1 0] with a threshold of -0.15 on this new image to get.



This had fairly little noise but a lot of loss, much because of parts removed along with subimage 3.
I then create a new image as before.

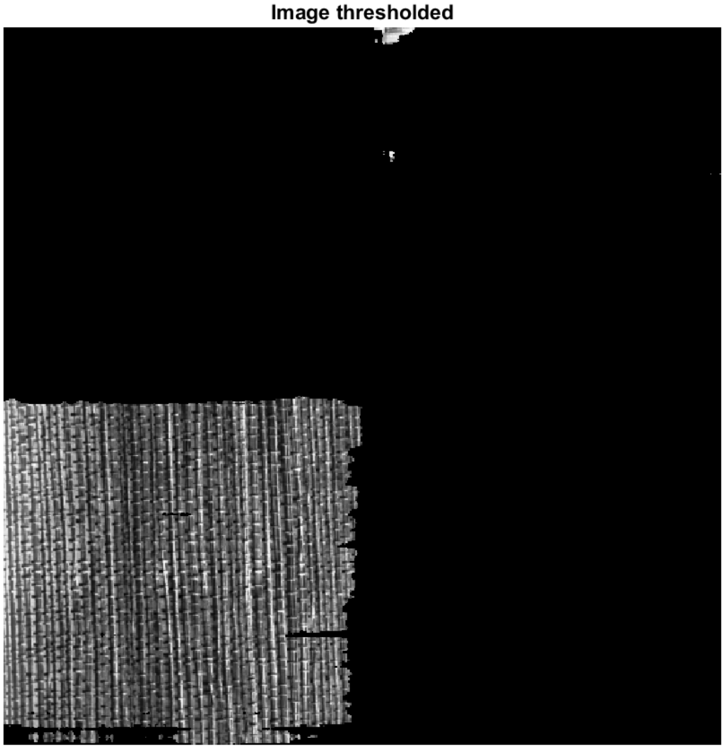
I tried using the different feature images to remove residual noise, but with no luck. So the best i can do for subimage 1 is the current new image, which is mostly subimage 1.



As mentioned in the start this has a lot of noise i could not find a way to remove.

Now for mosaic 2:
I use the same technique as for mosaic 1. Remove the easiest textures first then go for the ones less distinct in the feature images.

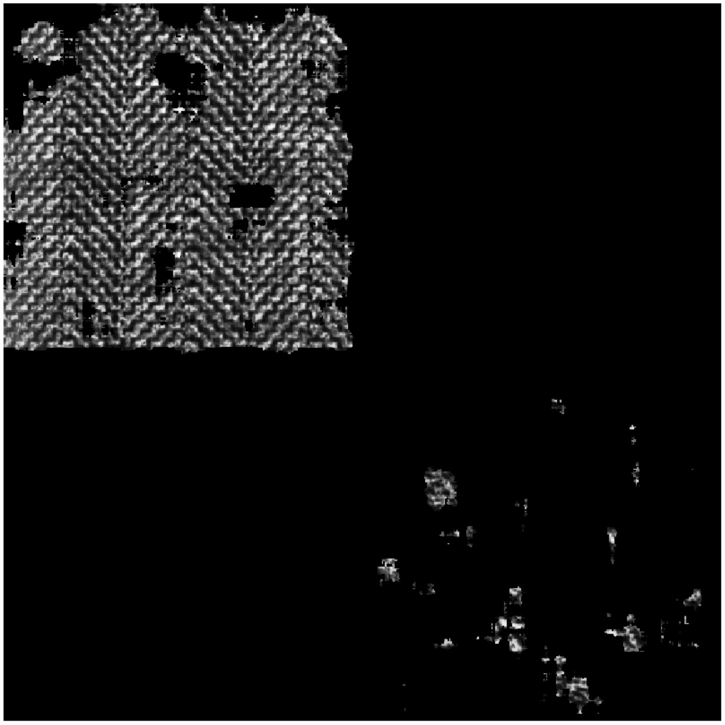
Subimage 7 should be separated well using INR [0 3]. Using a threshold of 0.3 i got very little noise but some loss, especial at the edge to subimage 5.



I then create a new image without the segmented part as like i did with mosaic 1.

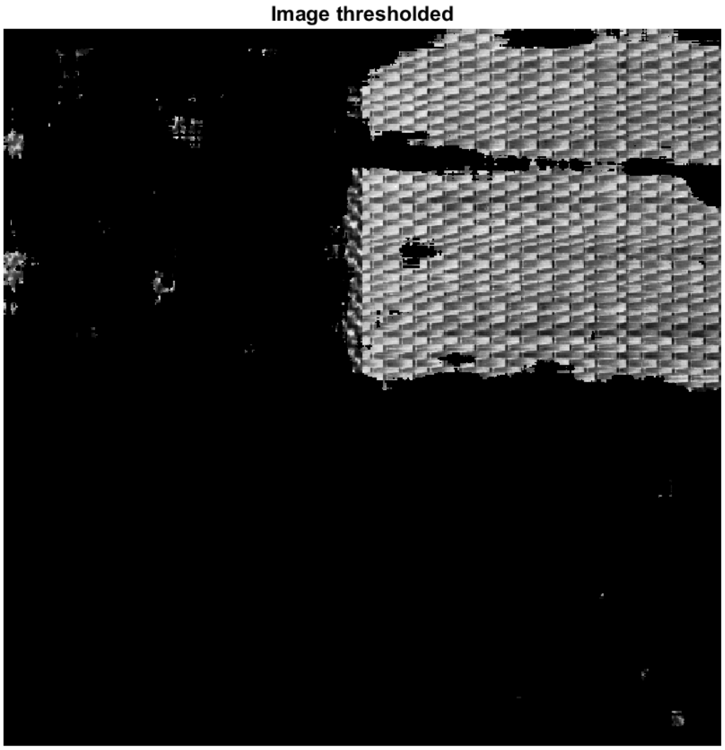
I then use IDM [0 4;4 0] to segment subimage 5. I use threshold 0.43 and get quite a lot of loss and some noise from subimage 8.

Image thresholded



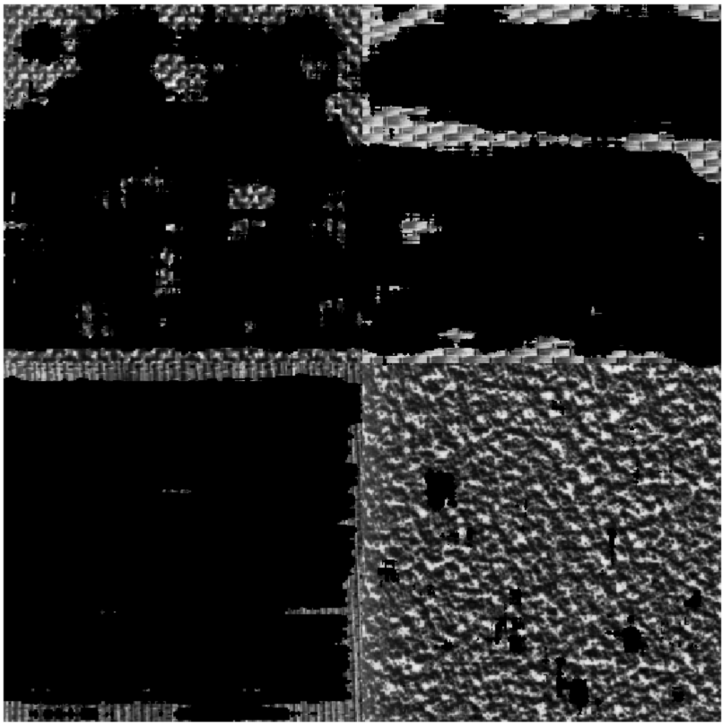
Again i create a new image by removing the segmented parts.

I will try to use IDM [0 3] to segment out subimage 6. A threshold of 0.4 gives quite a lot of loss with almost no noise. Just a little bit from subimage 1 and 8.



Creating the next new image i have mostly subimage 8 left.

Using both IDM and INR isotropic looks like good candidates for removing the noise from the current new image, witch is currently:



Using the INR proved to work poorly. So i used IDM with a threshold of 0.5 witch gave a nice result.