CSE585/EE555: Digital Image Processing

Computer Project 4

Yifei Xiao, Ling Zhang, Jiaming Chai

April 10, 2020

A Objectives

We conduct the following topics in this project:

1. Learn the detail of the Gabor filter and implement the filter
2. Apply smoothing filter to certain images
3. Segmentation result to discriminate one texture from another
4. Discussion and evaluation on our results

B Method

Gabor elementary function(GEF) defined as:

$$h(x,y) = g(x',y')exp\{j2\pi(Ux+Vy)\}$$

It can separate into two parts.

First is Gaussian part:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}exp\{-\frac{1}{2}[(\frac{x}{\sigma_x})^2 + (\frac{y}{\sigma_y})^2]\}, which\ centered\ at\ (0,0)$$

$$(x',y') = (x\cos\phi + y\sin\phi, -x\sin\phi + y\cos\phi)$$

And Sinusoidal part:

$$exp[j2\pi(Ux+Vy)] \equiv exp[j2\pi F(x\cos\theta + y\sin\theta)]$$

In image processing, Gabor filter is a linear filter used for texture analysis. It is defined as:

$$m(x,y) = |i(x,y) * h(x,y)|$$

In the project, $\sigma_x = \sigma_y$, so we can separate the formula into two parts. In this way, we use three steps to get Gabor filter:

1. $i_1(x,y) = i(x,y) * h_1(x)$
   $= i(x,y) * g_1(x)exp\{j2\pi Fxcos(\theta)\}$
2. $i_2(x,y) = i_1(x,y) * h_2(x)$
   $= i_1(x,y) * g_2(y)exp\{j2\pi Fysin(\theta)\}$
3. $m(x,y) = |i_2(x,y)|$

And for $i_1(x, y)$, we have

$$i_1(x, y) = \sum_{x'=-2\sigma}^{2\sigma} i(x - x', y)\hat{h}_1(x')$$

Here as we also implemented a smoothing filter:

$$m'(x, y) = m(x, y) * g'(x, y)$$

So given parameter F, θ, σ, we can create a Gabor filter and a smoothing filter for texture segmentation.

C Result

C.1 texture1

We show the texture segmentation for *texture1*. Figure 1 shows the original image. From the original image, we can know that textures in the top part are "×" and textures in the bottom part are "L". The parameters for the Gabor filter and the smoothing filter are list in Table 1. σ_1 is for Gabor filter and σ_2 is for smoothing filter. Figure 2 shows the middle step after applying Gabor filter. We can find that using Gabor filter can already find out the obvious level. Figure 3 shows the middle step after applying smoothing filter. The level is more obvious. We select 0.023 as our filter threshold. As shown in Figure 4, the green background part is segmented texture, which meets the expectation. Some part not in green background is because Gabor filter cannot apply at boarder.

F	0.059
θ	135
σ_1	8
σ_2	24

Table 1: Parameter for texture1.

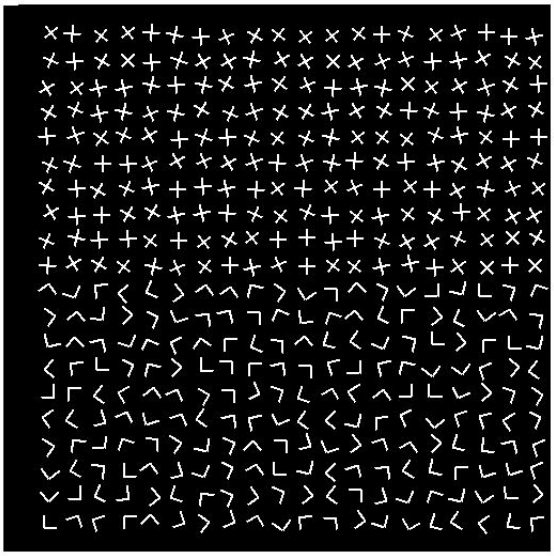

Figure 1: Original Image.

Figure 2: Gabor filtered of texture1.

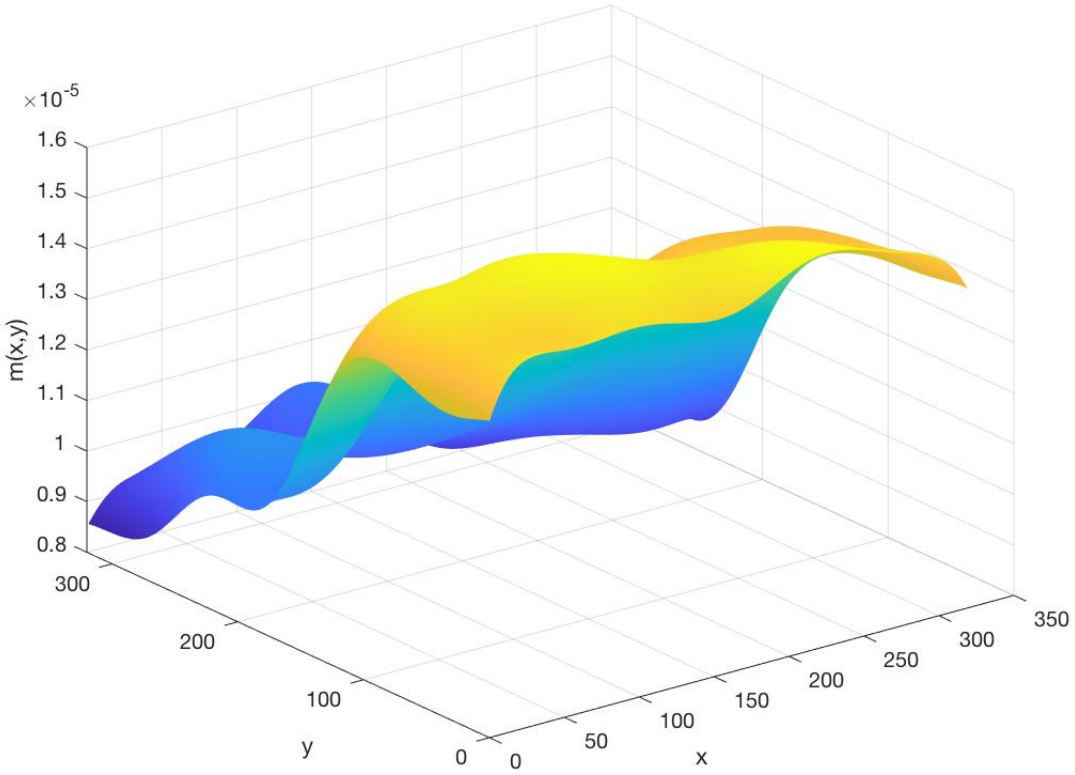

Figure 3: Plot of Gabor filtered of texture1.

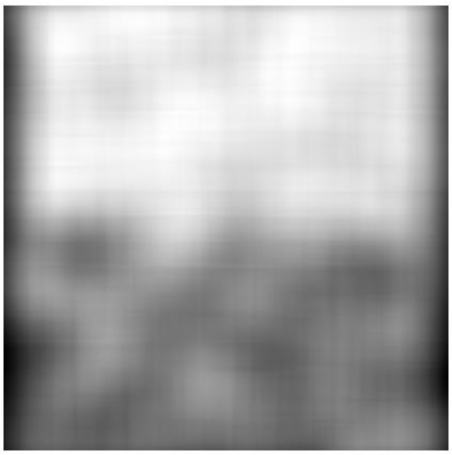

Figure 4: Smoothing filtered of texture1.

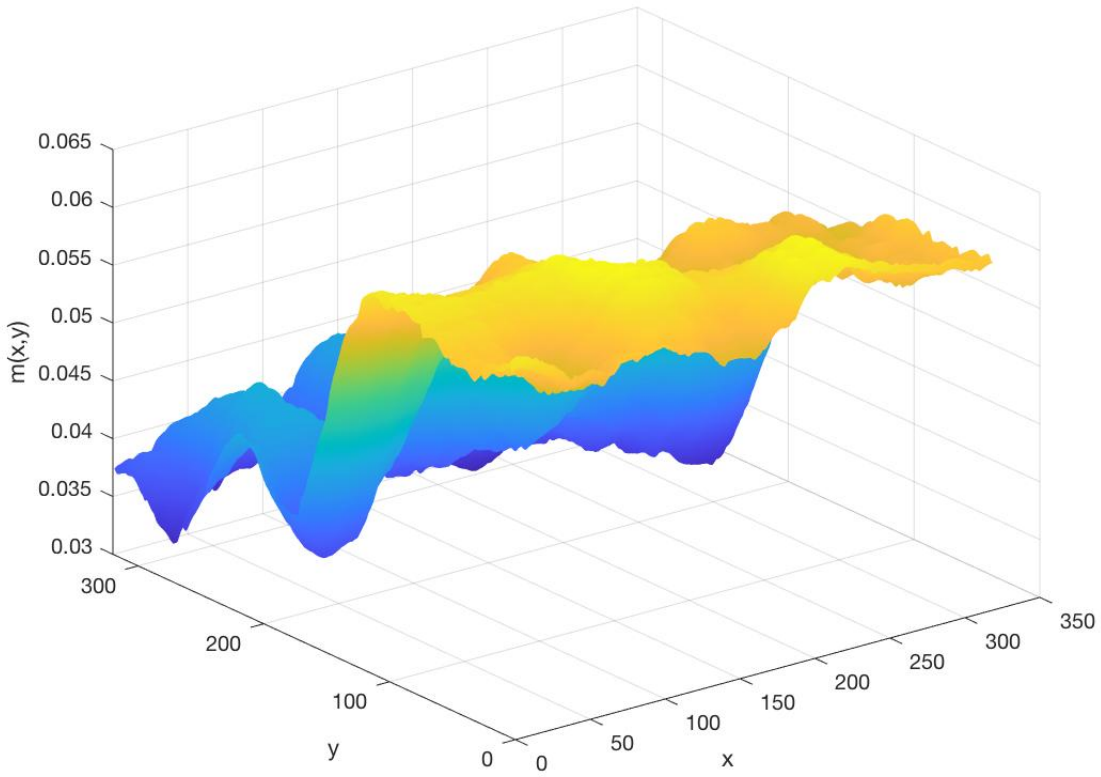

Figure 5: Smoothing filtered of texture1.

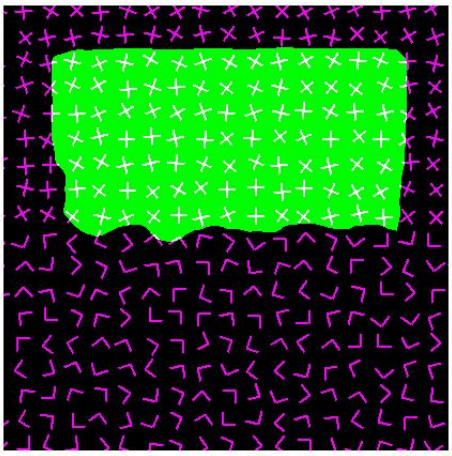

Figure 6: Segmentation of texture1.

C.2 texture2

We show the texture segmentation for *texture2*. Figure 5 shows the original image. From the original image, we can know that textures in the top part are "|" and textures in the bottom part are "\". The parameters for the Gabor filter and the smoothing filter are list in Table 4. σ_1 is for Gabor filter and σ_2 is for smoothing filter. Figure 6 shows the middle step after applying Gabor filter. We can find that using Gabor filter can already find out the obvious level, and it is better than *texture1*. This is because in this case we don't have rotation, all textures are in the same direction. Figure 7 shows the middle step after applying smoothing filter. The level is more obvious. We select 0.0273 as our filter threshold. As shown in Figure 8, the green background part is segmented texture, which meets the expectation.

F	0.042
θ	0
σ_1	24
σ_2	24

Table 2: Parameter for texture2.

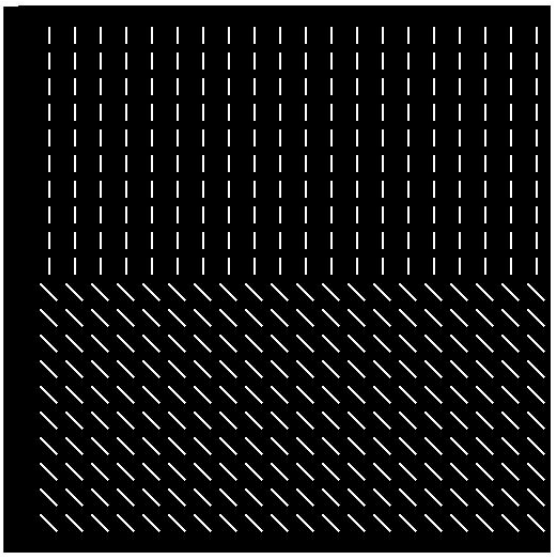

Figure 7: Original Image.

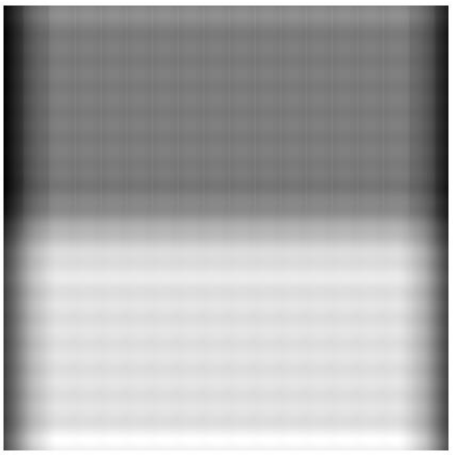

Figure 8: Gabor filtered of texture2.

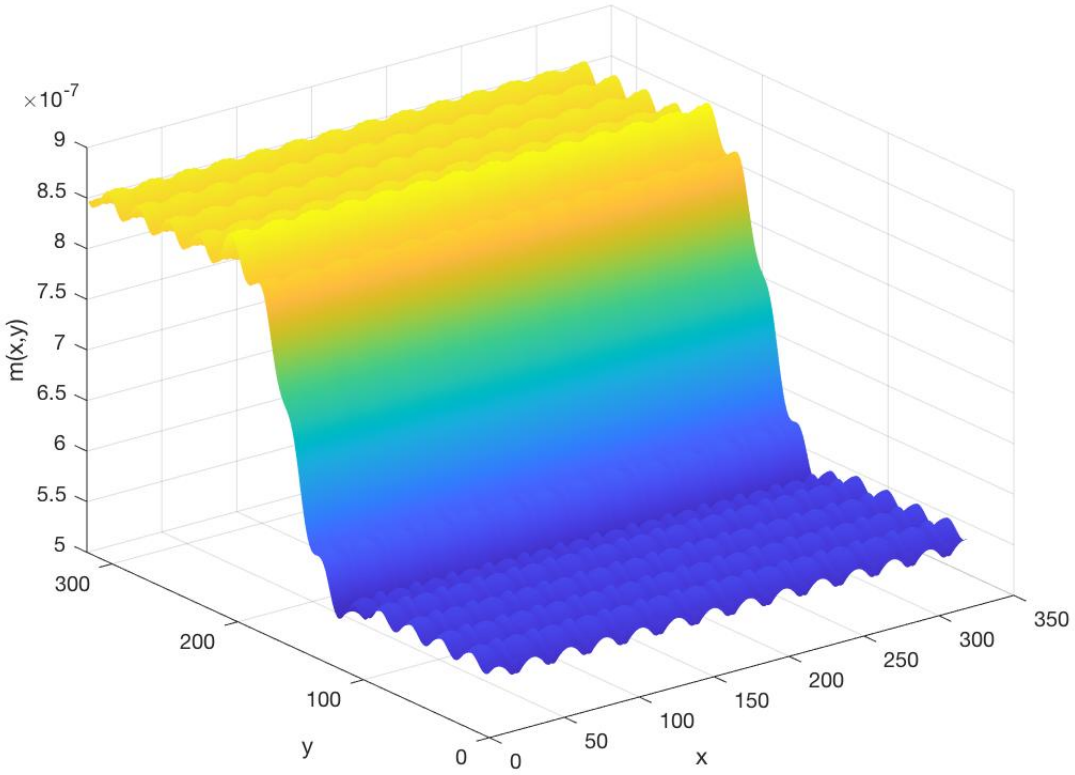

Figure 9: Plot of Gabor filtered of texture2.

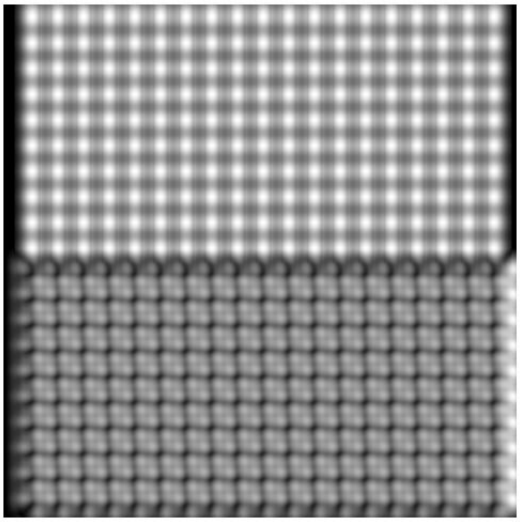

Figure 10: Smoothing filtered of texture2.

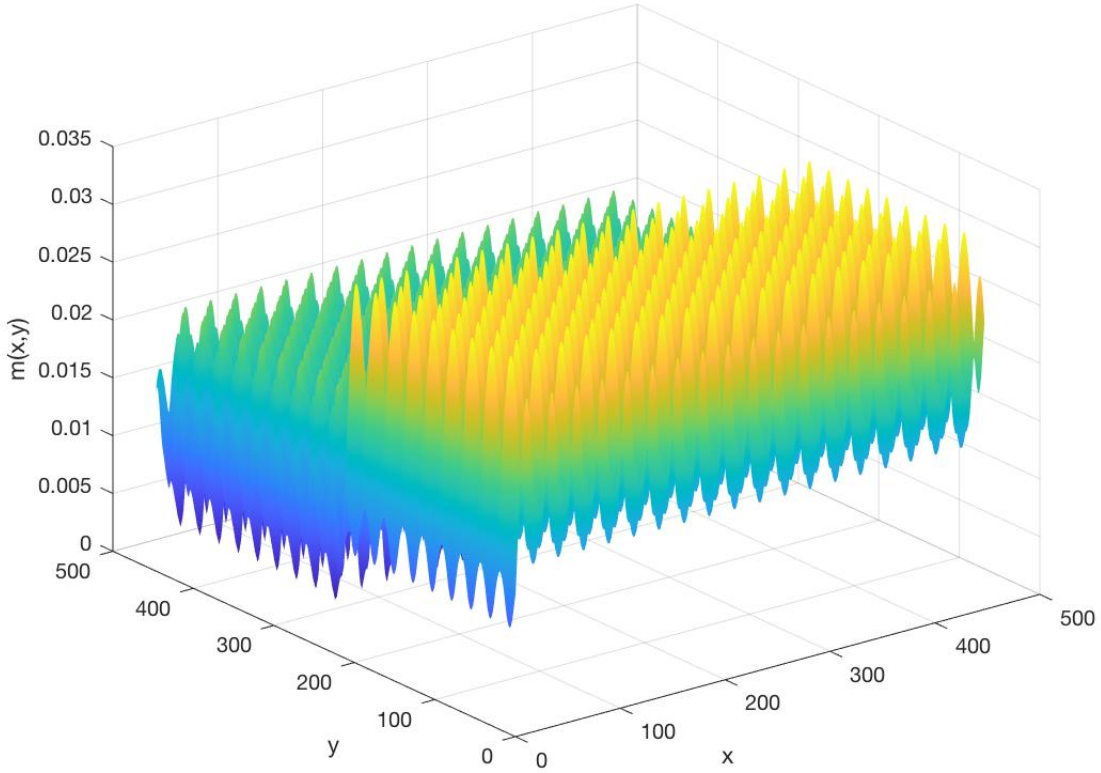

Figure 11: Smoothing filtered of texture2.

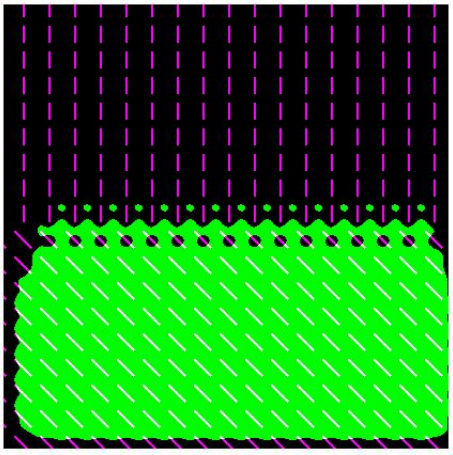

Figure 12: Segmentation of texture2.

C.3 d9d77

We show the segmentation for a picture that has disordered Brodatz textures that consists of grass lawn and cotton canvas. In this part no smoothing filter is applied.

F	0.063
θ	60
σ_1	36
σ_2	36

Table 3: Parameter for d9d77.

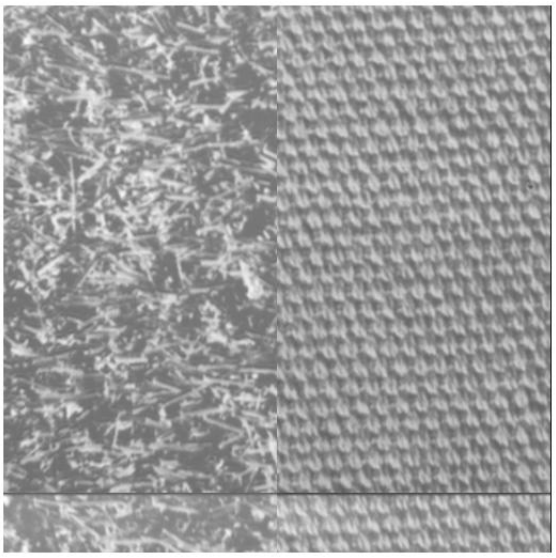

Figure 13: Original Image.

Figure 14: Gabor filtered of d9d77.

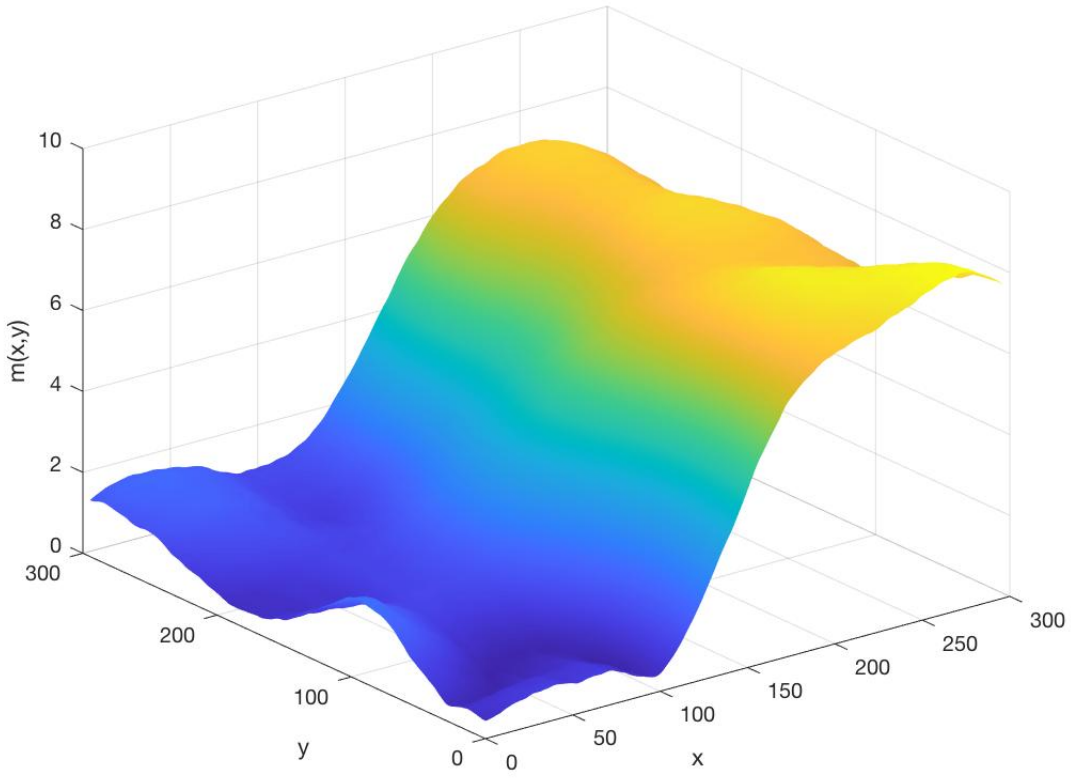

Figure 15: Plot of Gabor filtered of d9d77.

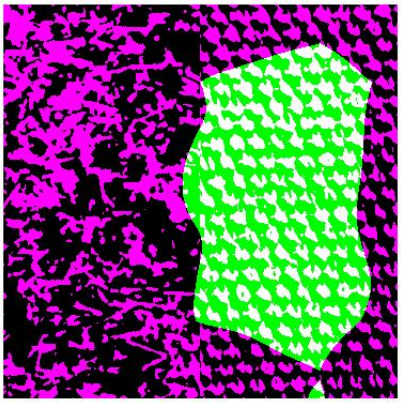

Figure 16: Segmentation of d9d77.

C.4 d4d29

We show the segmentation for a picture that has pressed cork and beach sand with Gabor filter and follow-up smoothing filter.

F	0.6038
θ	-50.5
σ_1	8
σ_2	8

Table 4: Parameter for d4d29.

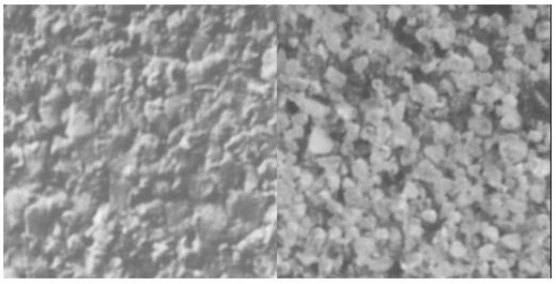

Figure 17: Original Image.

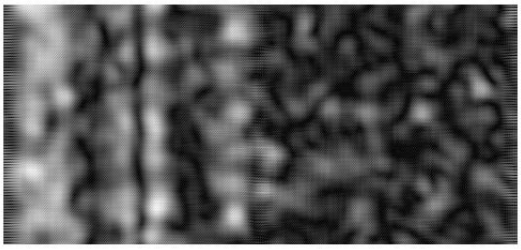

Figure 18: Gabor filtered of d4d29.

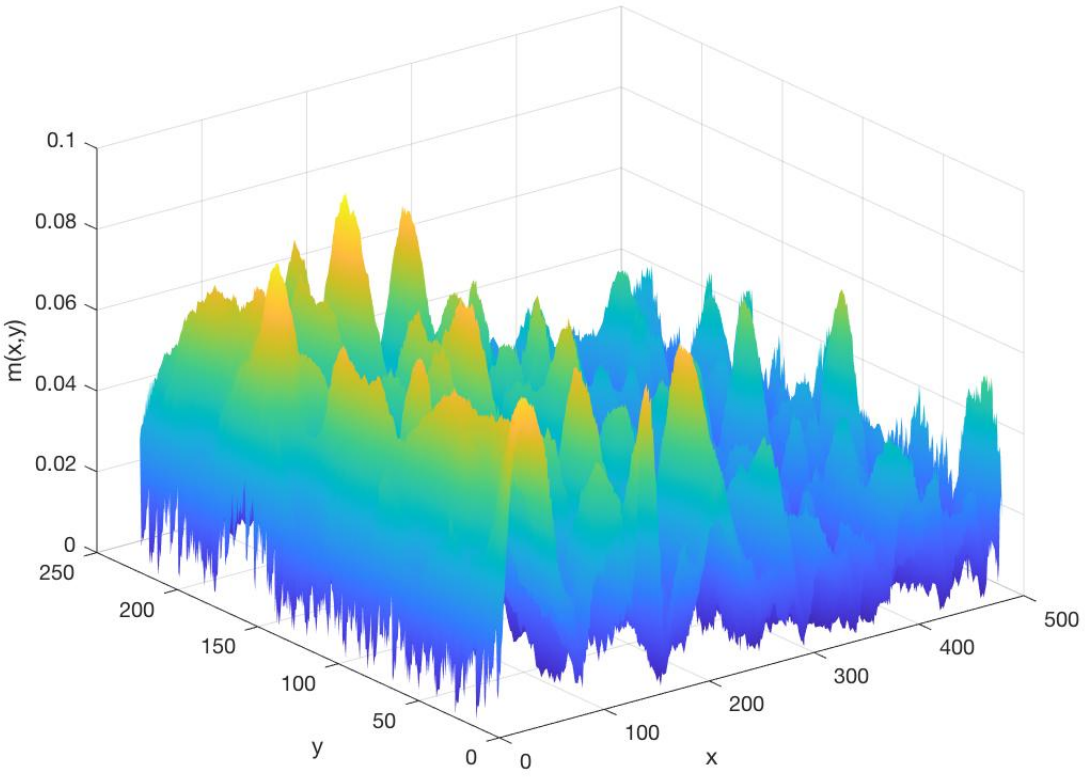

Figure 19: Plot of Gabor filtered of d4d29.

Figure 20: Smoothing filtered of d4d29.

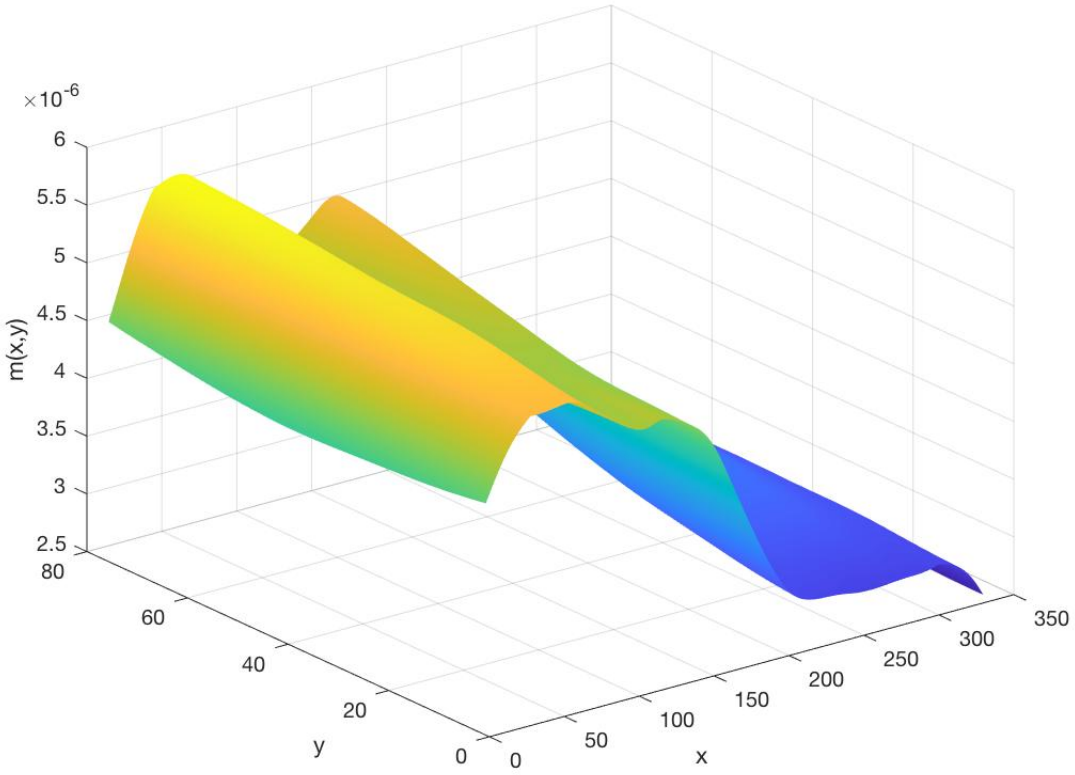

Figure 21: Smoothing filtered of d4d29.

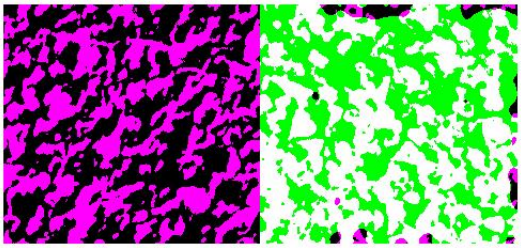

Figure 22: Segmentation of d4d29.

C.5 Discussion

We think these segmentations are pretty good in terms of distinguishing one texture from another. In our segmentations some pixels are not completely included in the segmented version. Overall the segementations can indicate one sub-texture from another.

The portions in the images are shown as the bright green color that covers some area in the image. We labelled both parts as two bright colors. Those two parts are the ones that could be actually segmented.

Also, when we threshold the pixels to segment the images, we do it pixel by pixel as it is the simplest way. We do think that using pooling could eliminate those "various" small pixels, like max pool or min pool.

D Conclusions

In this project, we learned Gabor filters and applied it to certain images with follow-up smoothing filter. We first introduce the method theoretically and then applied to the given gif images. Also we segment the images based on the pixel value by putting a threshold to it. We also believe using pooling can make the segmentation robust.