CSE585: Digital Image Processing II
Computer Project 4
Texture Segmentation

Aishwarya Mallampati, Mrunmayi Ekbote, Wushuang Bai

Date:04/10/2020

---

1 Objectives

The main objective of this project is to segment textures in an input image using gabor filter. Also, to implement gabor filter in MATLAB without using built-in functions.

2 Methods

2.1 Texture Segmentation

2.1.1 Theory and Algorithms

Texture segmentation is the process of partitioning an image into regions with different textures containing similar group of pixels. In image processing, a Gabor filter is a linear filter used for texture analysis, which means that it basically analyzes whether there are any specific frequency content in the image in specific directions in a localized region around the point or region of analysis. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.
Gabor filter is applied on images in this project to perform texture segmentation. The algorithm for texture segmentation is as follows:

- Read an input image I(x,y)
- Choose appropriate values for frequency(F), θ and standard deviation(σ) depending on the properties of the input image.
- Compute 2-D circularly symmetric guassian (g(x,y)) using F, θ and σ. 2-D guassin g(x,y) can be computed using 1-D guassian distributions g(x) and g(y).

$$g_1(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{-\frac{x^2}{2\sigma^2}\right\} \tag{2.1a}$$

$$g_2(y) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{-\frac{y^2}{2\sigma^2}\right\} \tag{2.1b}$$

$$g(x,y) = g_1(x).g_2(y) \tag{2.1c}$$

$$g(x,y) = \frac{1}{2\pi\sigma} exp\left\{-\frac{(x^2+y^2)}{2\sigma^2}\right\} \tag{2.1d}$$

- Next step is to compute 2-D Gabor Elementary Functions(GEF-h(x,y)).

$$h(x,y) = g(x,y) exp\left\{j2\pi F(xcos\theta + ysin\theta)\right\} \tag{2.2}$$

The GEF h(x,y) is seperable in x and y(provided that g(x,y) is symmetric)

$$h(x,y) = h_1(x).h_2(y) \tag{2.3a}$$
$$h_1(x) = g_1(x)exp\{j2\pi Fxcos\theta\} \tag{2.3b}$$
$$h_2(x) = g_2(y)exp\{j2\pi Fysin\theta\} \tag{2.3c}$$

- Apply the Gabor filter to the input image.

$$m(x,y) = |i(x,y) * *h(x,y)| \tag{2.4}$$

$**$ implies 2-D convolution. This 2-D convolution on the input image can be replaced with (*)1-D convolutions with the help of above equations using the following three steps:

$$i_1(x,y) = i(x,y) * h_1(x) \tag{2.5a}$$
$$i_1(x,y) = \sum_{k=-2\sigma}^{2\sigma} i(x-k,y)\hat{h}_1(k) \tag{2.5b}$$
$$i_2(x,y) = i_1(x,y) * h_2(y) \tag{2.5c}$$
$$i_2(x,y) = \sum_{k=-2\sigma}^{2\sigma} i(x,y-k)\hat{h}_2(k) \tag{2.5d}$$
$$m(x,y) = |i_2(x,y)| \tag{2.5e}$$

Eq2.5band2.5c are convolutions in x and y respectively. $\hat{h}_1(.)$ is the truncated Gaussian function modulated by sinusoid over a $[-2\sigma; +2\sigma]$ range. $\hat{h}_1(.)$ is an array of complex numbers which is precomputed only once.

- Compute smoothing filter $g\prime(x,y)$ using a different σ using equation2.1d. Then, apply this smoothing filter to the gabor filter output m(x,y) (if required).

$$m\prime(x,y) = m(x,y) * g\prime(x,y) \tag{2.6}$$

- Segment different textures present in $m\prime(x,y)$ by choosing a threshold that produces good segmentation results.
- In the final step, superimpose the image segmentation result on the original image and display its result.

2.1.2 Matlab

All the code used for this problem is placed in main.m file. Texture segmentation is performed on four images namely texture2.gif, texture1.gif, d9d77.gif and d4d29.gif.
The following are the matlab files used for this problem:

- main.m: The following steps are performed in this file:
  - On running this file, the program asks the user to select an input image to perform texture segmentation.

- The image chosen by user is read.
- If the input image is binary image, then it is converted to gray scale image.
- Gabor filter is generated and is applied on the image. The output of gabor filter are displayed and stored in the form of 2-D gray scale image and 3-D plots.
- Smoothing is applied to the gabor filter output. The output of smoothing filter are displayed and stored in the form of 2-D gray scale image and 3-D plots.
- Finally, the smoothing filter output is thresholded.
- The thresholded image is superimposed on original image. The superimposed image is displayed and stored in the working directory.

- GEF.m: Generates Gabor Elementary Functions along x and y direction to perform texture segmentation on the input image

- circular_symm_gaussian.m: Compute circular symmetric gaussian function along x and y directions separately.

- convolution1D.m: Performs 1-D convolution between the given input image and given filter along x or y direction.

- adjust_image.m: Adjusts given image using sigma and range

- threshold_image.m: Segments given image using input threshold value.

- superimpose_image.m: Superimposes final thresholded image on the original input image.

The results obtained are displayed in section3.
Note: Run main.m file to get the results The flowchart of the algorithm is displayed in fig1.

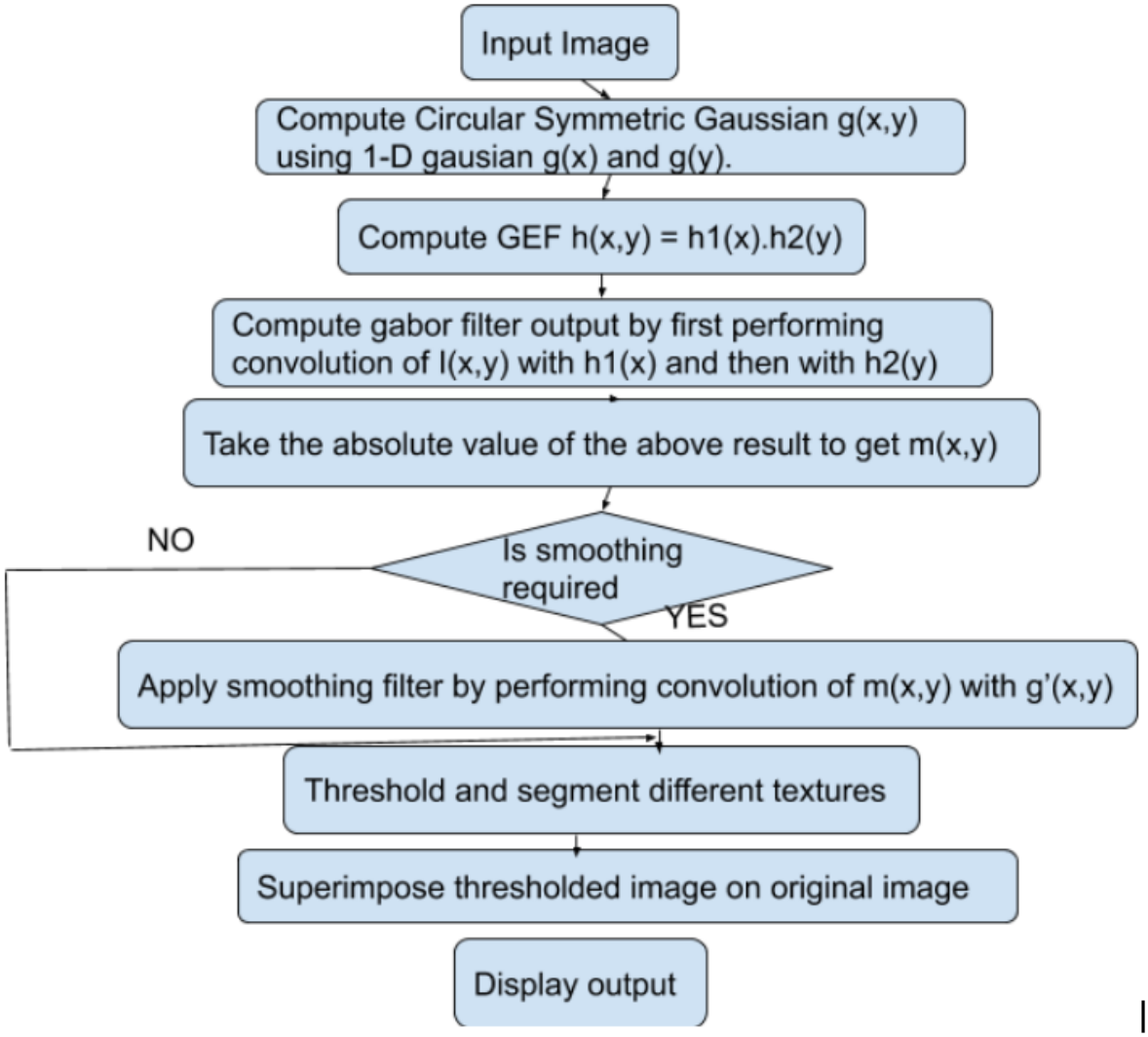

Figure 1: FLOWCHART

3 Results

This section displays the results after applying gabor and smoothing filter to the input images texture2.gif, texture1.gif, d9d77.gif and d4d29.gif in sec 3.1,3.2,3.3 and 3.4 respectively.

3.1 texture2.gif

The original image that is considered for this question is as shown in Figure 2. The gabor filter is applied by using the parametric values, F = 0.059 cycles/pixel; $\theta = 135$ and $\sigma = 8$. Figure3 displays the result obtained after applying gabor filter to the image in the form of 2-D gray scale image and Figure4 shows the gabor filtered output as a 3-D plot. Then, Smoothing filter with $\sigma = 24$ is applied to gabor filter outputs. Figure5 displays the result obtained after applying smooth filter in the form of 2-D gray scale image and Figure6 shows the smooth filtered output as a 3-D plot. The outputs shown

in Figures3 and 5 has been rescaled for the sake of display. Figure7 shows the image after applying the threshold to the smoothened image. A threshold of 12 was used for this image. Figure8 shows the segmentation of different textures.

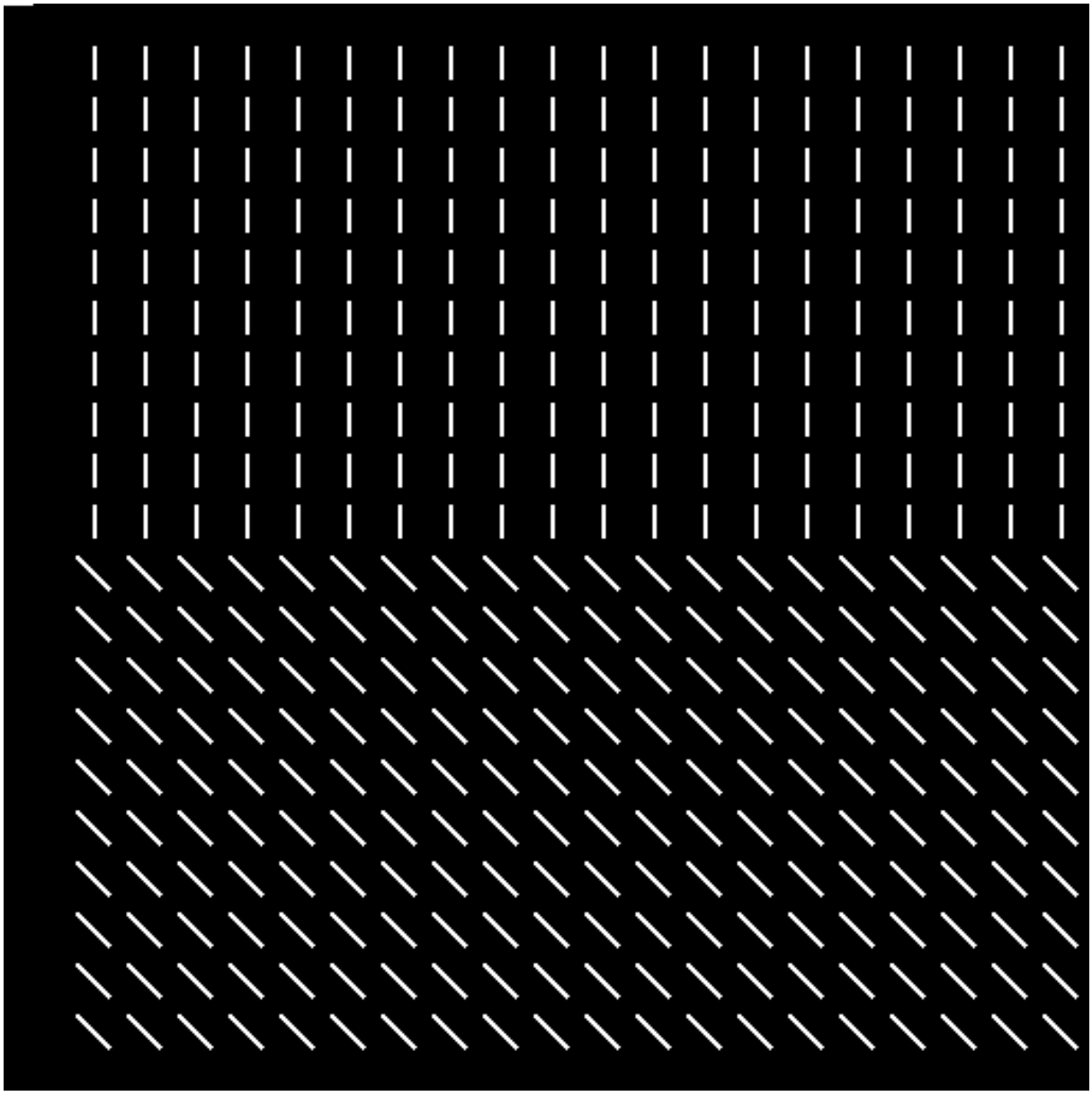

Figure 2: texture2: Original Input Image

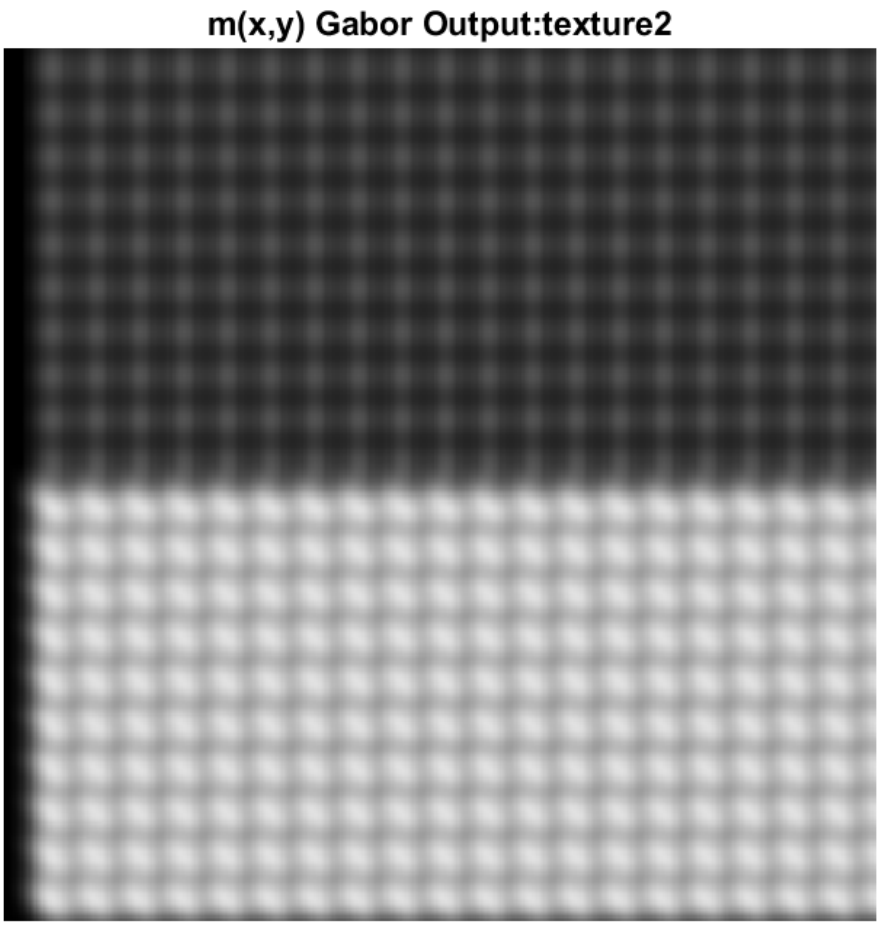

Figure 3: texture2: Gabor Filtered Output Image

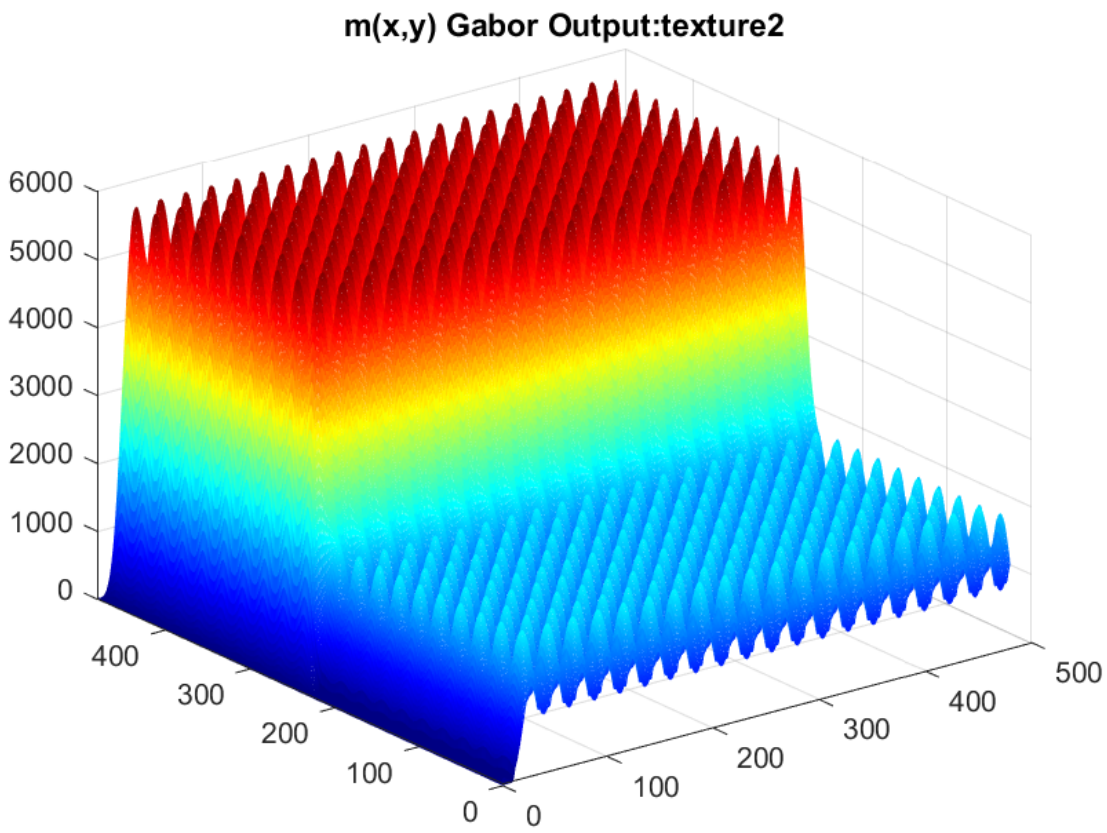

Figure 4: texture2: 3-D Plot of Gabor Filtered Output Image

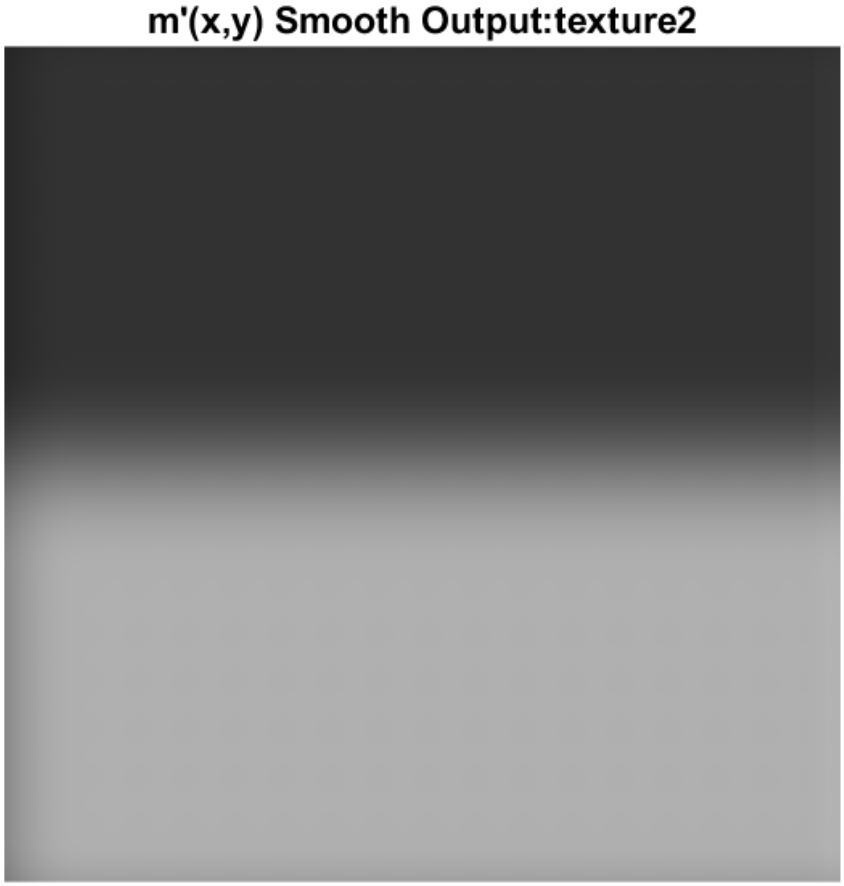

Figure 5: texture2: Smooth Filtered Output Image

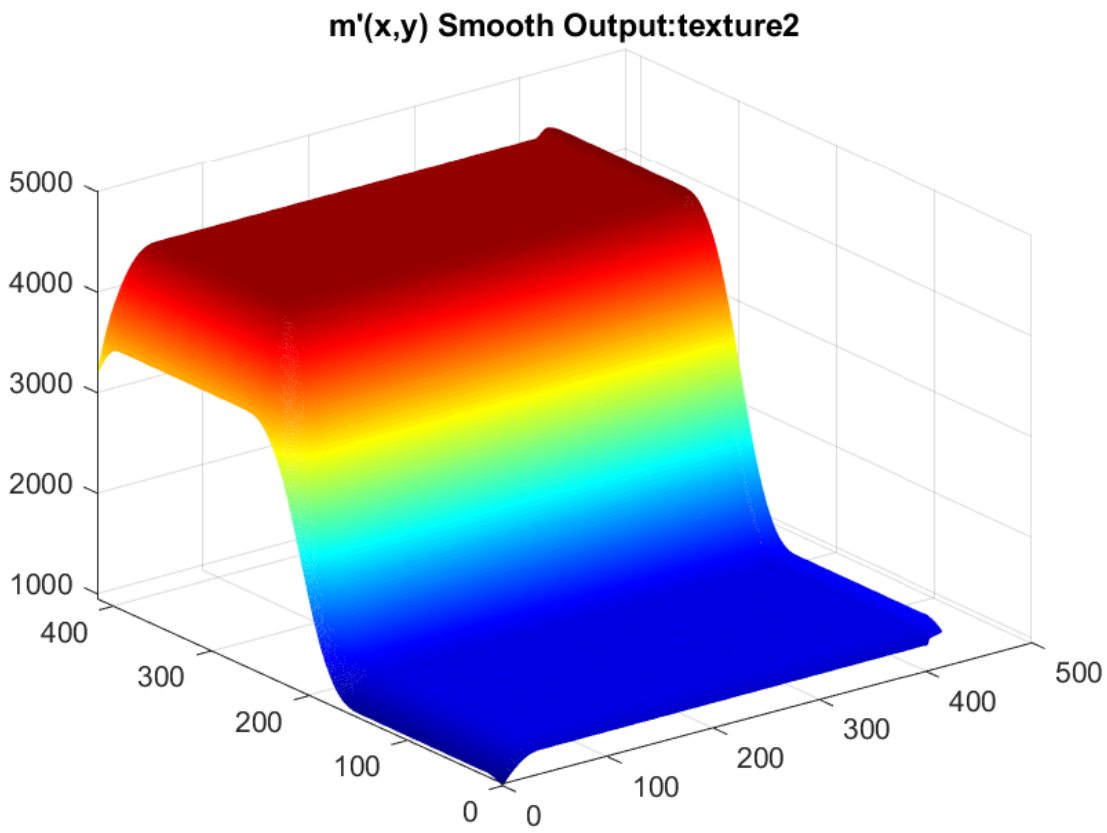

Figure 6: texture2: 3-D Plot of Smooth Filtered Output Image

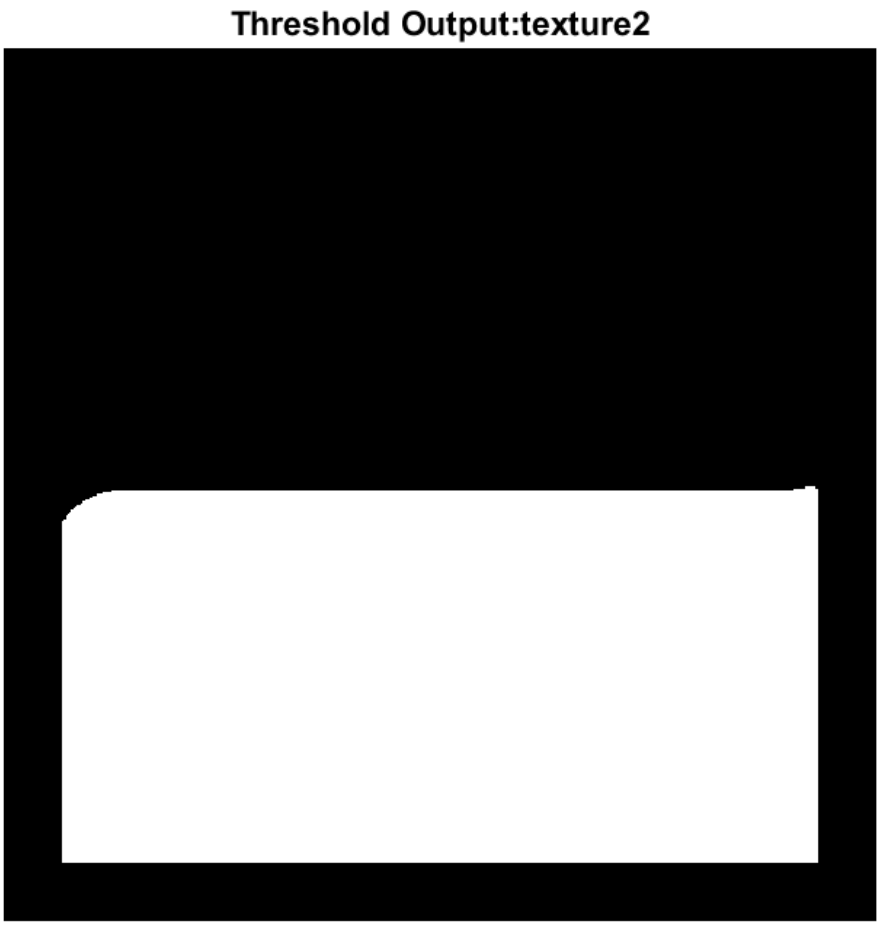

Figure 7: texture2: Thresholded Image

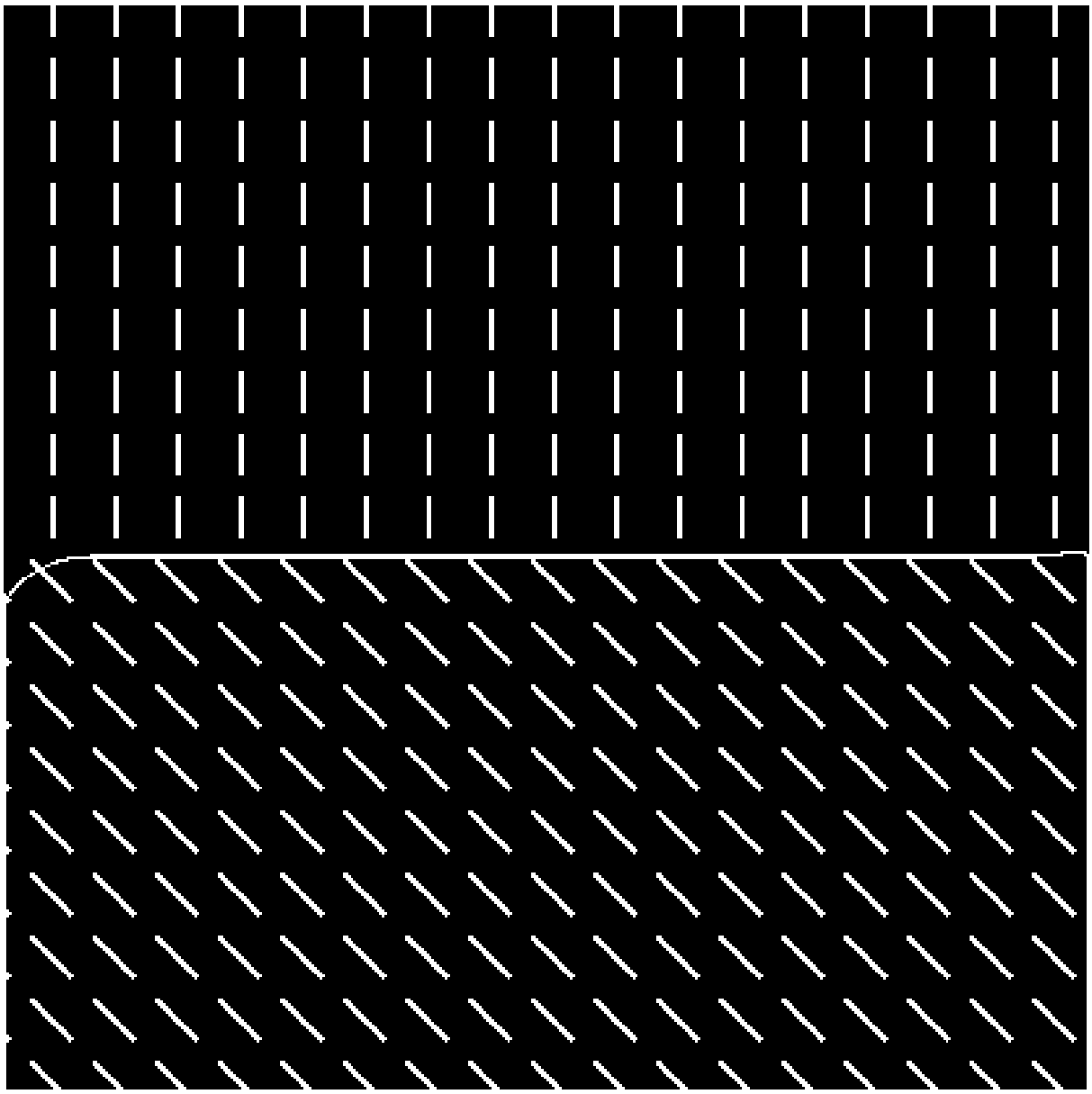

Figure 8: texture2: Segmented Image

The texture segmentation for this input image texture2.gif is really good. A horizontal line with y=211 or the line joining points (2,211) and (414,211) can be considered as the line of segmentation.

3.2 texture1.gif

The original image that is considered for this question is as shown in Figure 9. The gabor filter is applied by using the parametric values, F = 0.042 cycles/pixel; $\theta = 0$ and $\sigma = 24$. Figure10 displays the result obtained after applying gabor filter to the image in the form of 2-D gray scale image and Figure11 shows the gabor filtered output as a 3-D plot. Then, Smoothing filter with $\sigma = 24$ is applied to gabor filter outputs. Figure12 displays the result obtained after applying smooth filter in the form of 2-D gray scale

image and Figure13 shows the smooth filtered output as a 3-D plot. The outputs shown in Figures10 and 12 has been rescaled for the sake of display. Figure14 shows the image after applying the threshold to the smoothened image. A threshold of 12 was used for this image. Figure15 shows the segmentation of different textures.

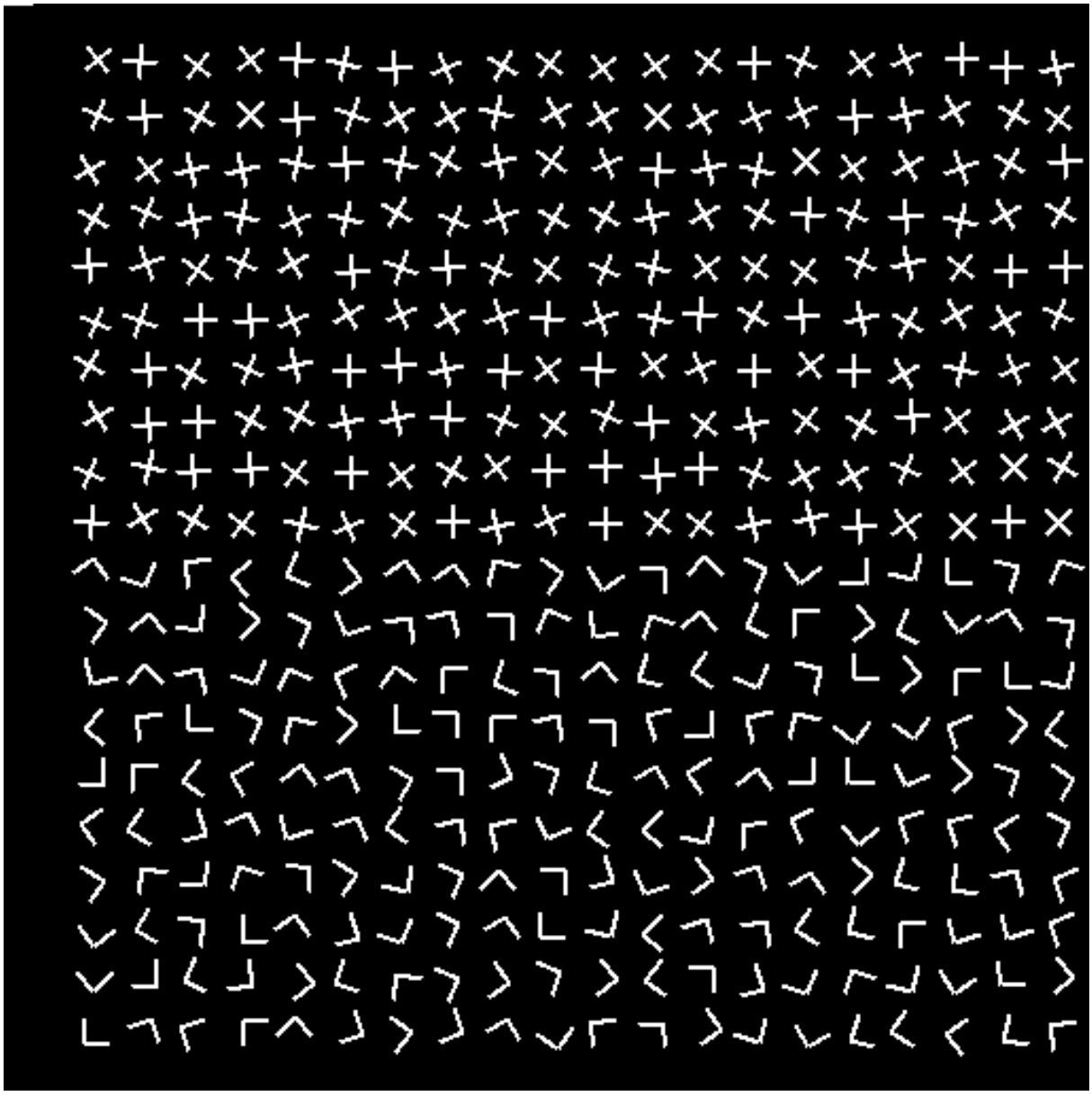

Figure 9: texture1: Original Input Image

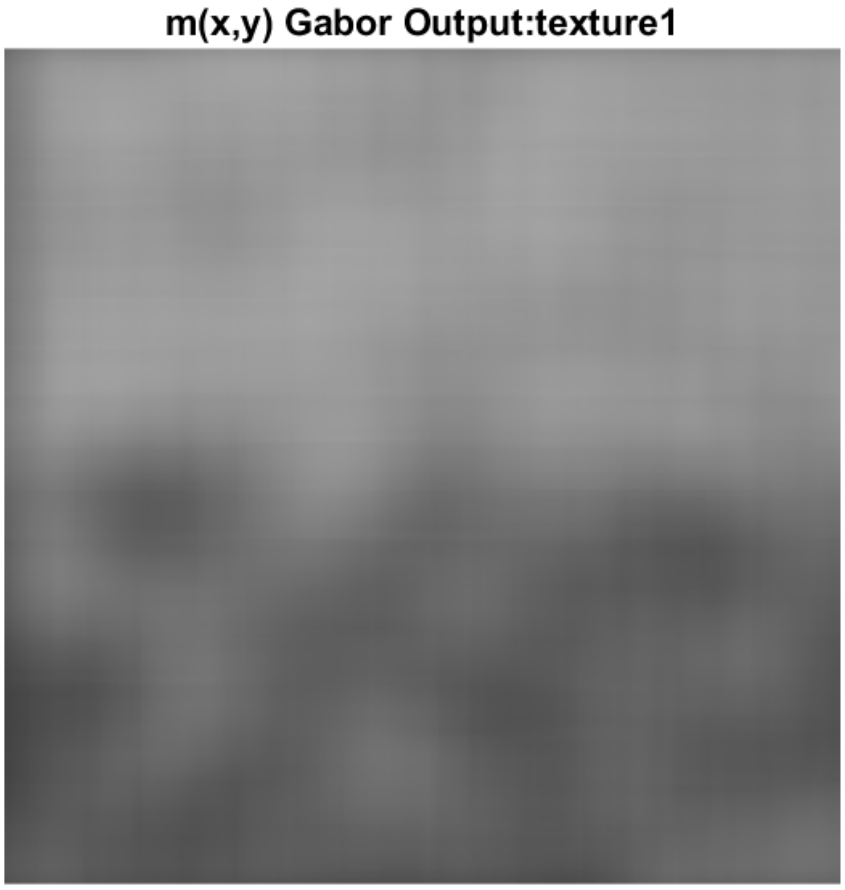

Figure 10: texture1: Gabor Filtered Output Image

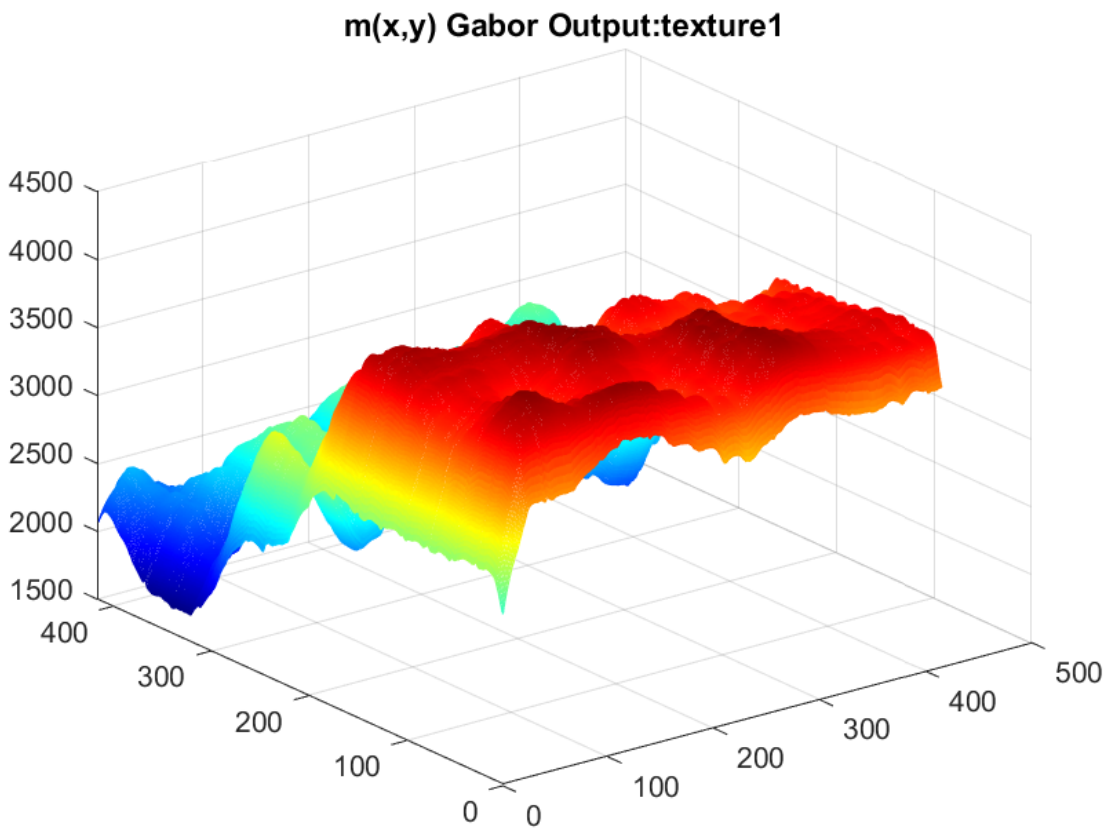

Figure 11: texture1: 3-D Plot of Gabor Filtered Output Image

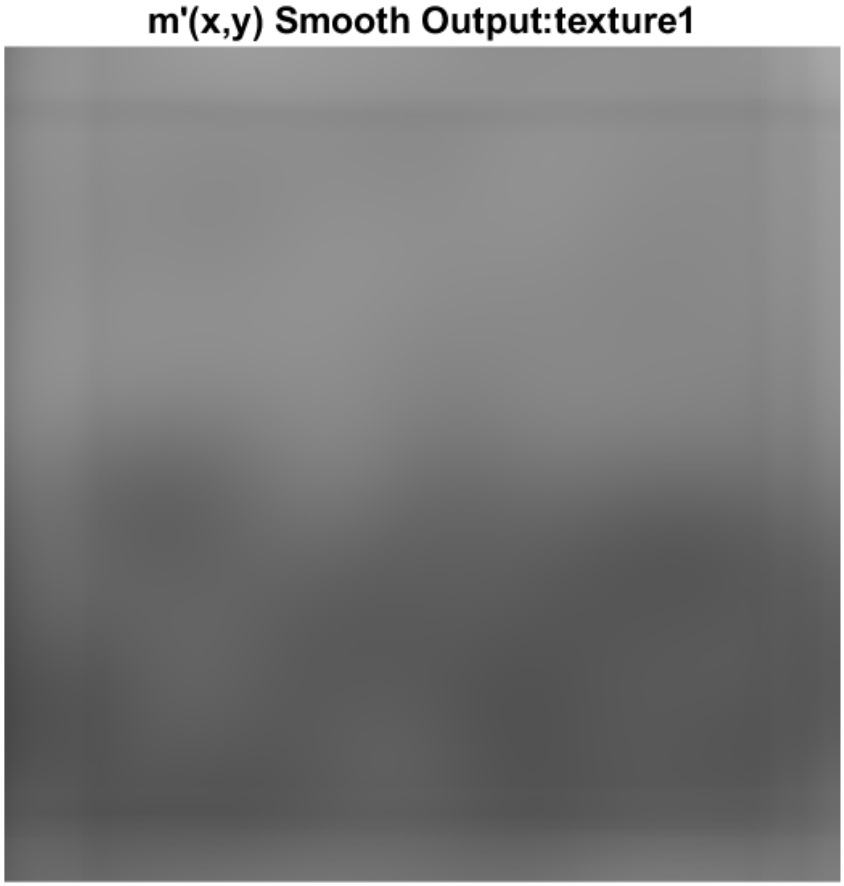

Figure 12: texture1: Smooth Filtered Output Image

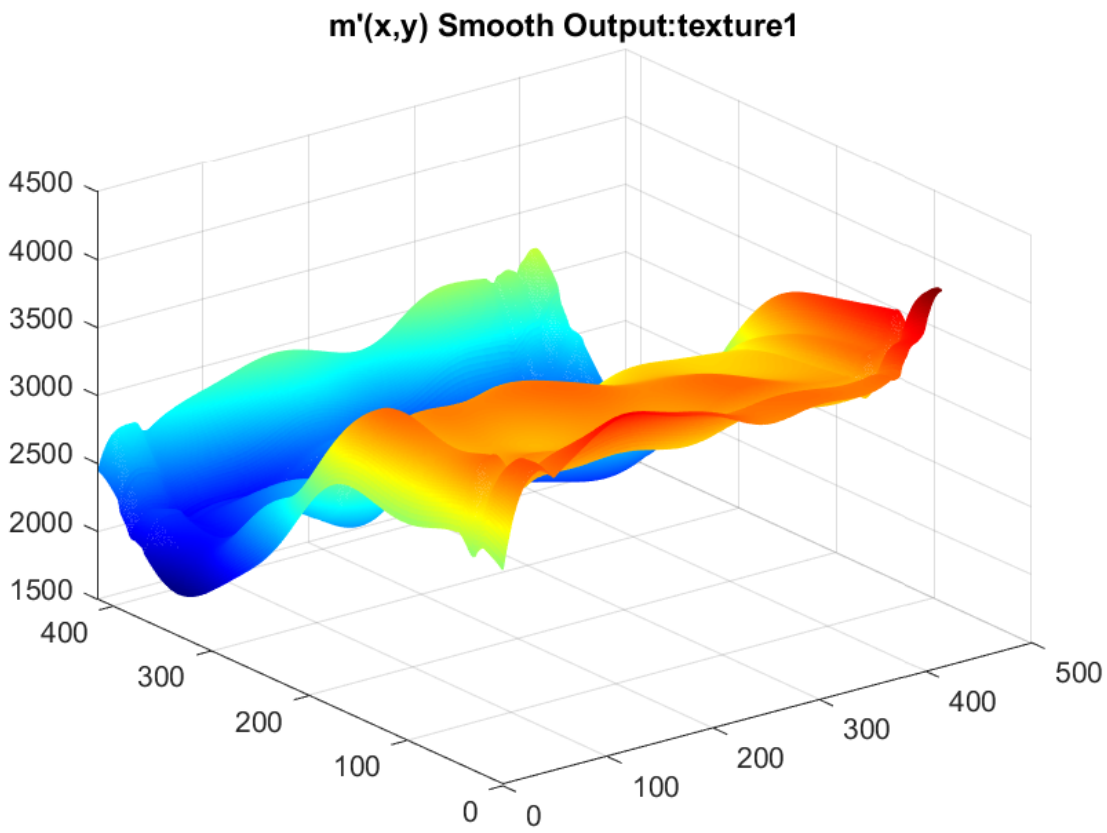

Figure 13: texture1: 3-D Plot of Smooth Filtered Output Image

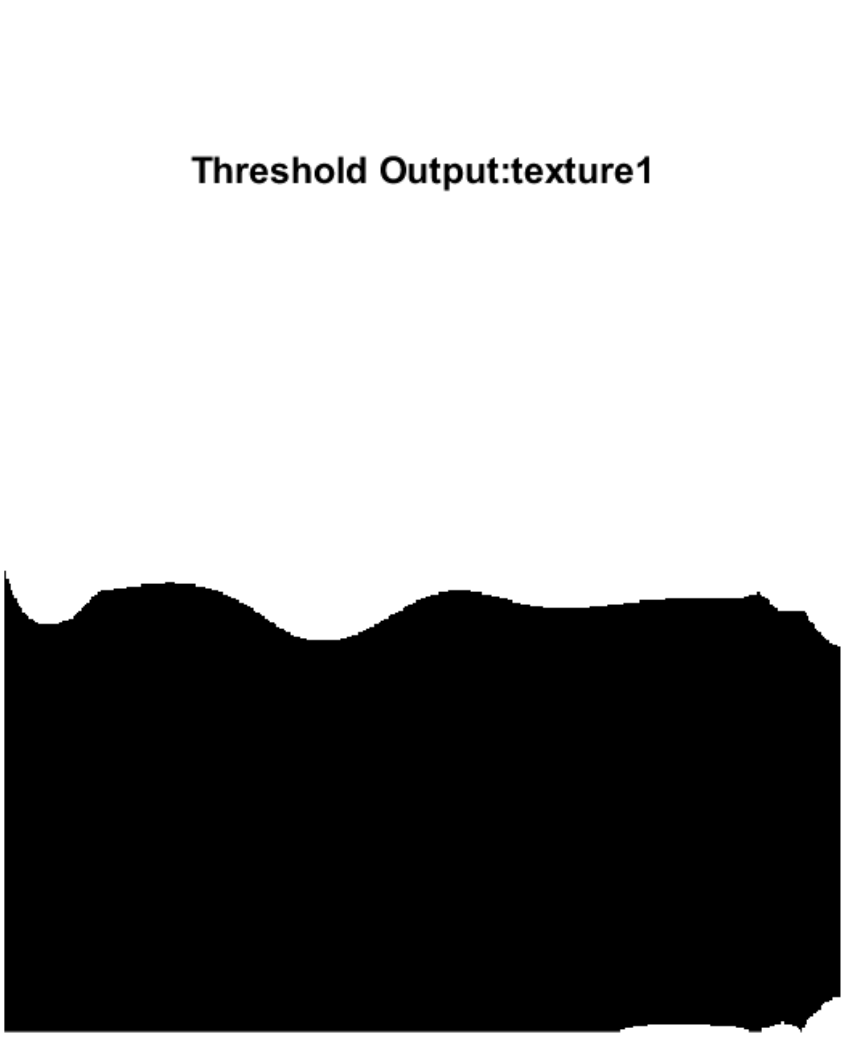

Figure 14: texture1: Thresholded Image

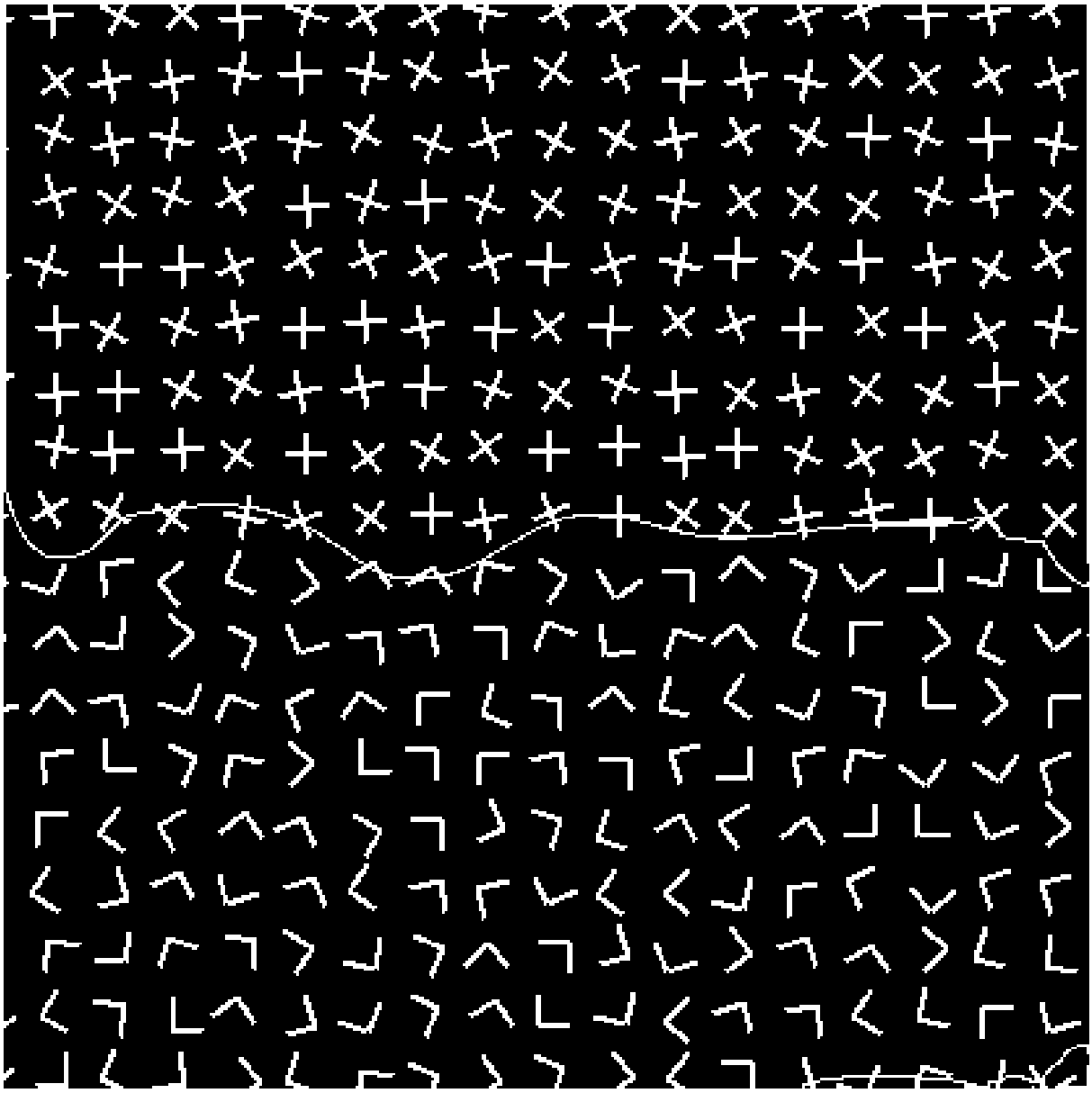

Figure 15: texture1: Segmented Image

The texture segmentation for this input image texture1.gif is really good. A horizontal line with y=200 or the line joining points (2,200) and (414,200) can be considered as the line of segmentation.

3.3 d9d77.gif

The original image that is considered for this question is as shown in Figure 16. The gabor filter is applied by using the parametric values, F = 0.062 cycles/pixel; $\theta = 63$ and $\sigma = 38$. Figure17 displays the result obtained after applying gabor filter to the image in the form of 2-D gray scale image and Figure18 shows the gabor filtered output as a 3-D plot. The output shown in Figures17 has been rescaled for the sake of display. Figure19 shows the image after applying the threshold to the gabor filtered image. A threshold of

0.021 was used for this image. Figure20 shows the segmentation of different textures.

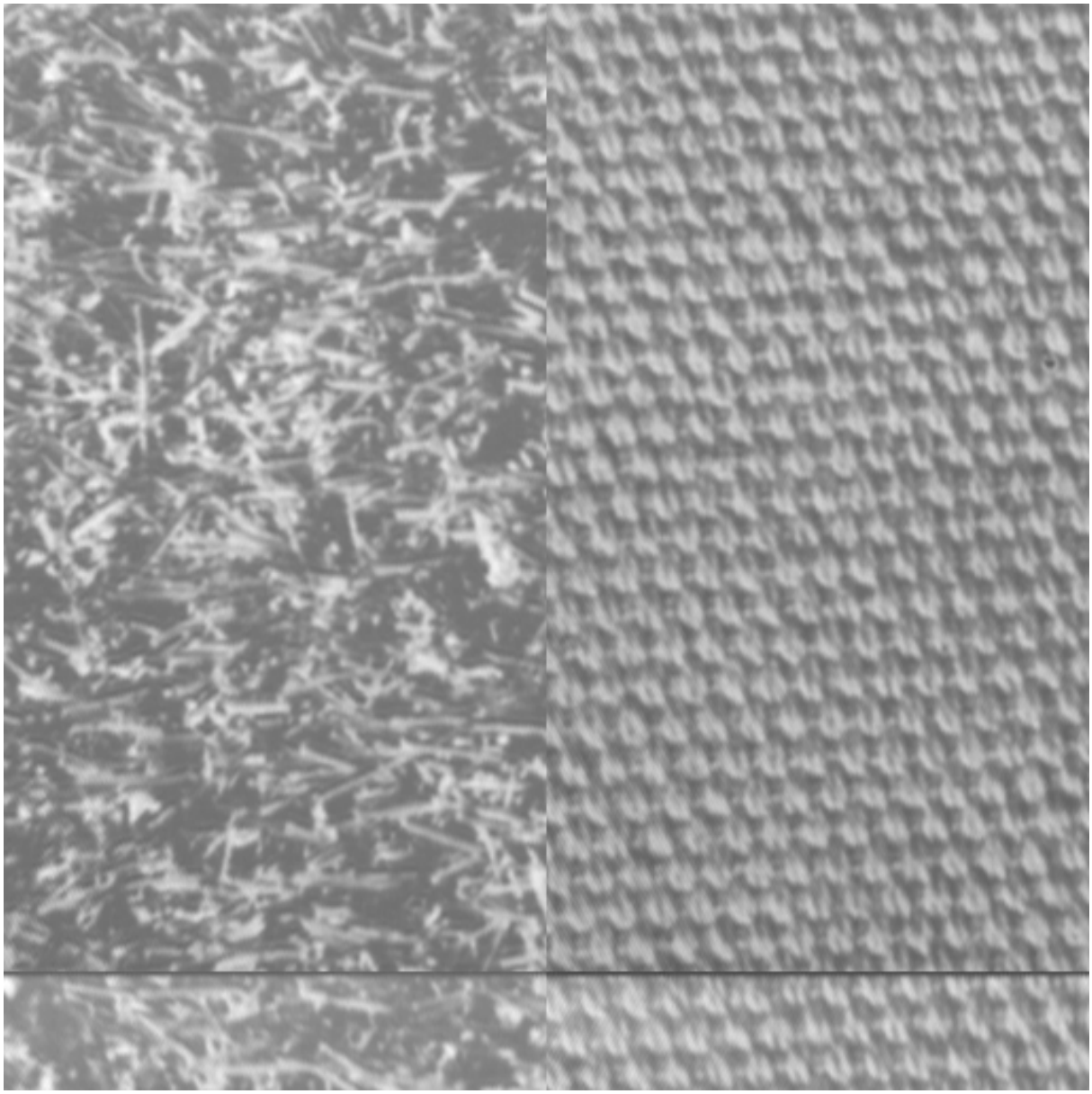

Figure 16: d9d77: Original Input Image

Figure 17: d9d77: Gabor Filtered Output Image

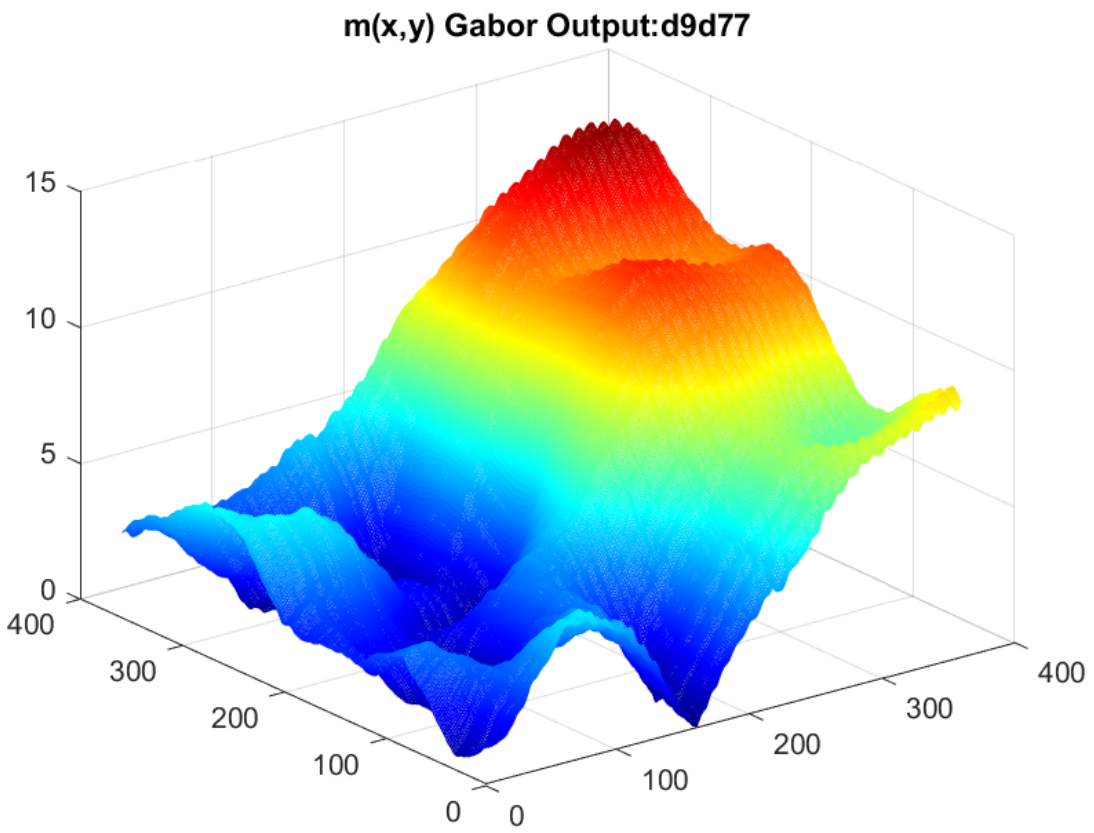

Figure 18: d9d77: 3-D Plot of Gabor Filtered Output Image

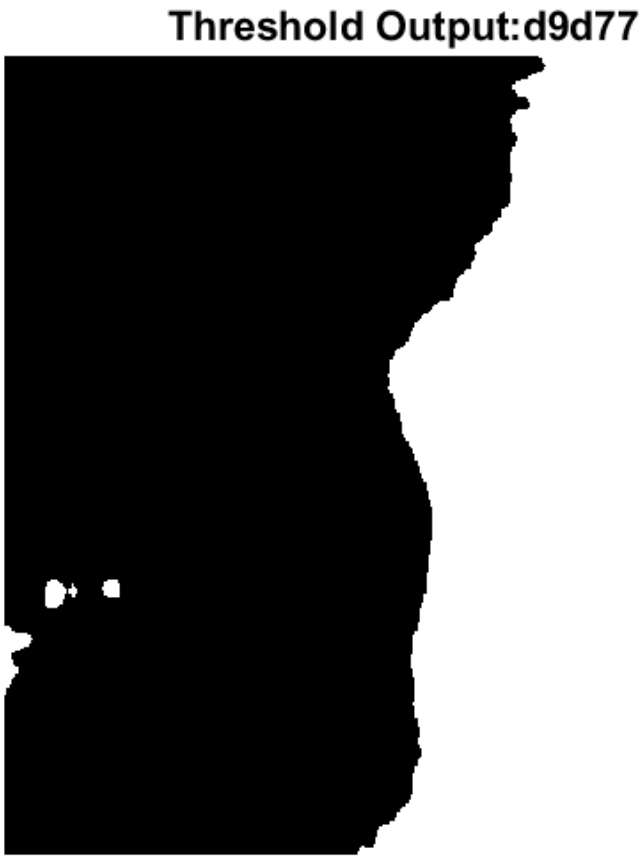

Figure 19: d9d77: Thresholded Image

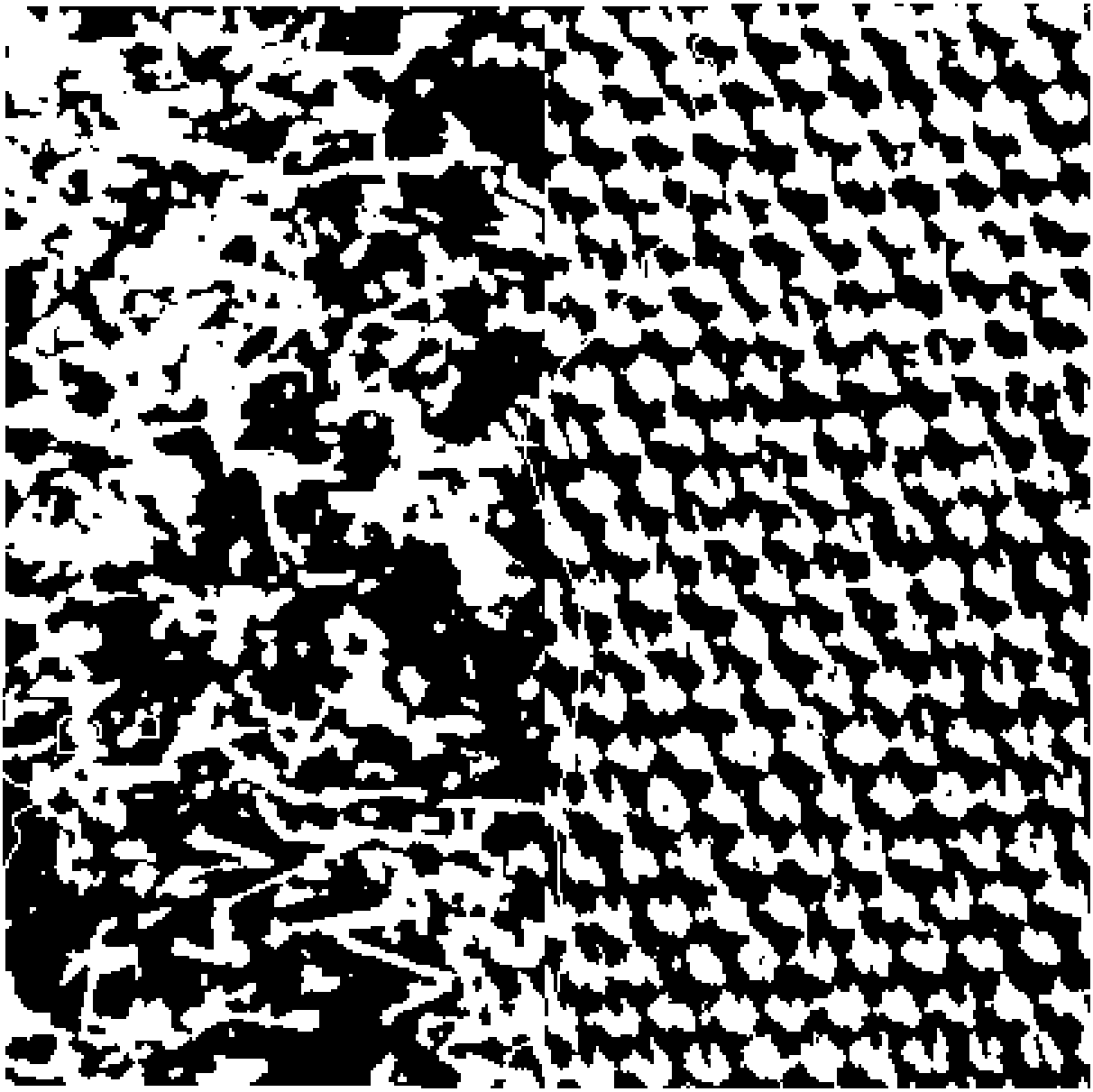

Figure 20: d9d77: Segmented Image

The texture segmentation for this input image d9d77.gif is not a perfect output but it is reasonably good segmentation. A vertical line with x=184 or the line joining points (184,2) and (184,359) can be considered as the line of segmentation.

3.4 d4d29.gif

The original image that is considered for this question is as shown in Figure 21. The gabor filter is applied by using the parametric values, F = 0.06038 cycles/pixel; $\theta = -50.5$ and $\sigma = 8$. Figure22 displays the result obtained after applying gabor filter to the image in the form of 2-D gray scale image and Figure23 shows the gabor filtered output as a 3-D plot. Then, Smoothing filter with $\sigma = 2$ is applied to gabor filter outputs. Figure24 displays the result obtained after applying smooth filter in the form of 2-D gray scale

image and Figure25 shows the smooth filtered output as a 3-D plot. The outputs shown in Figures22 and 24 has been rescaled for the sake of display. Figure26 shows the image after applying the threshold to the smoothened image. A threshold of 0.12 was used for this image. Figure27 shows the segmentation of different textures.

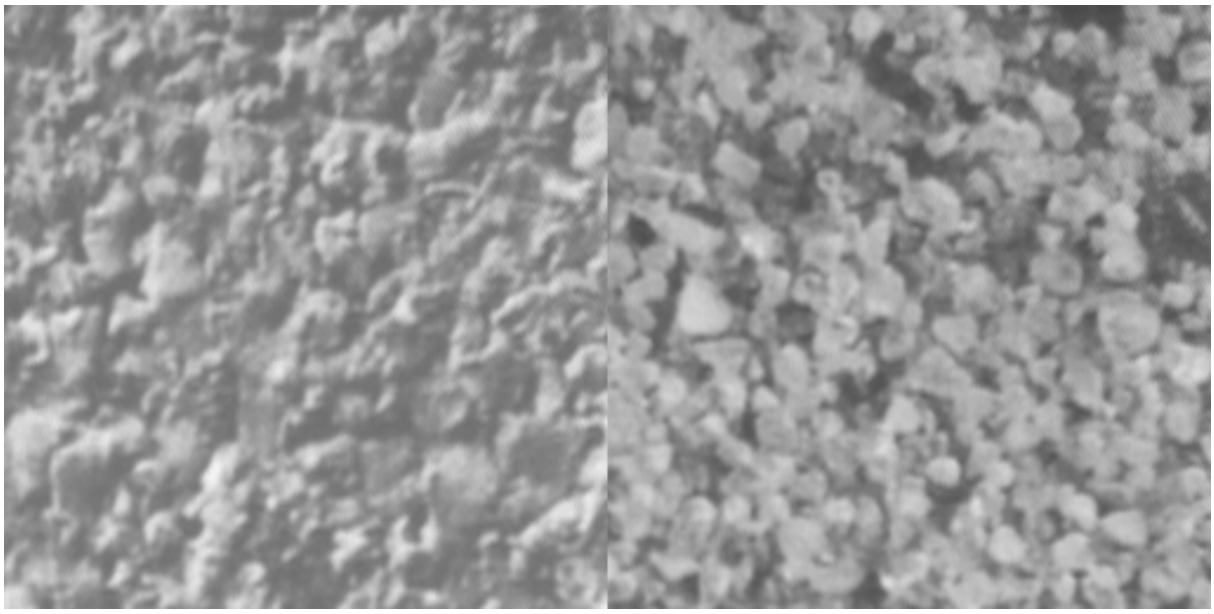

Figure 21: d4d29: Original Input Image

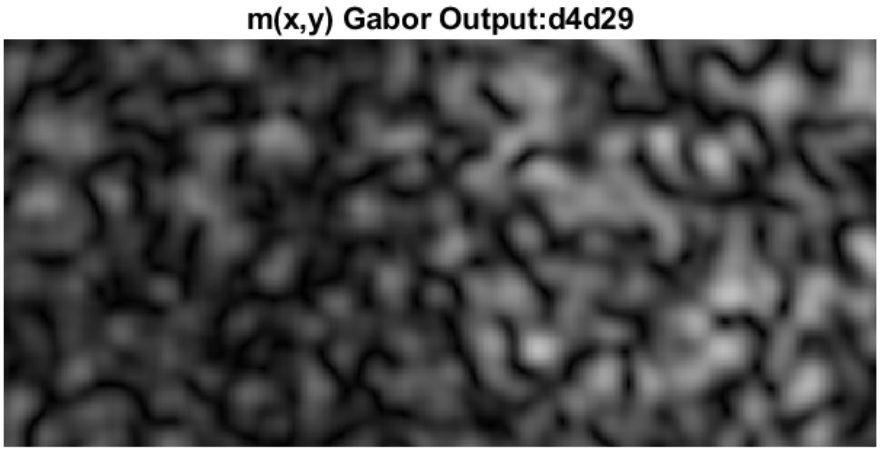

Figure 22: d4d29: Gabor Filtered Output Image

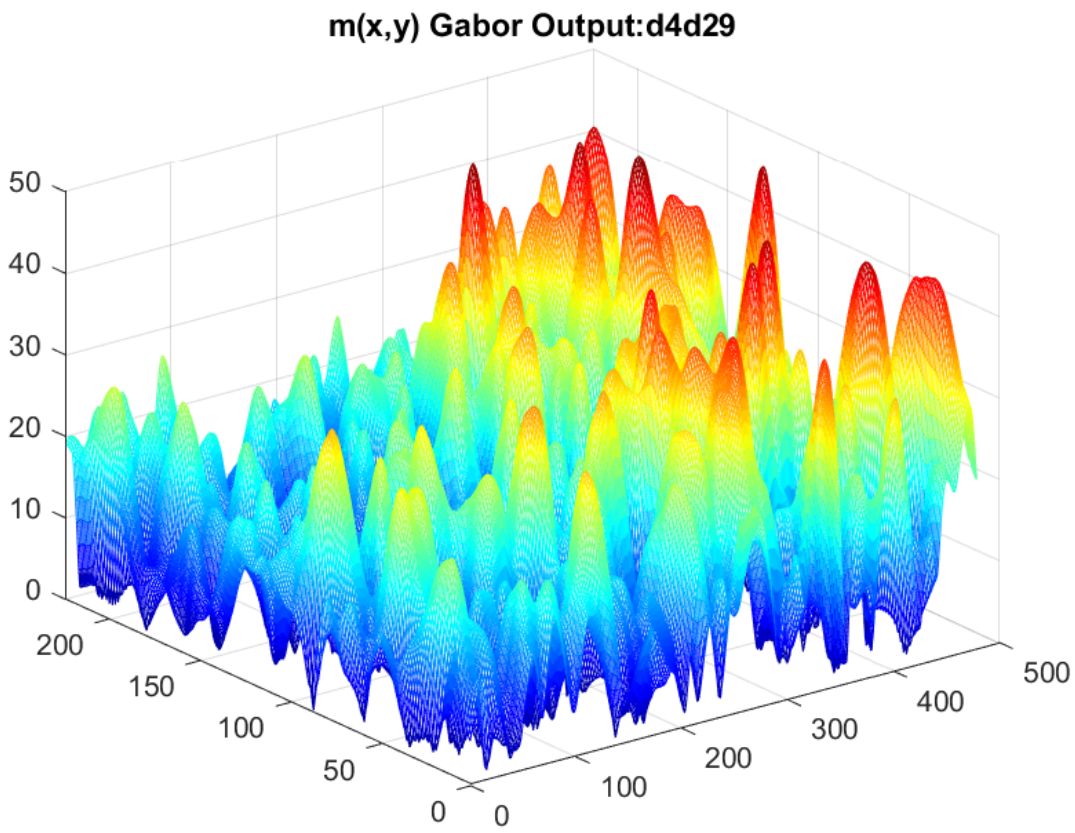

Figure 23: d4d29: 3-D Plot of Gabor Filtered Output Image

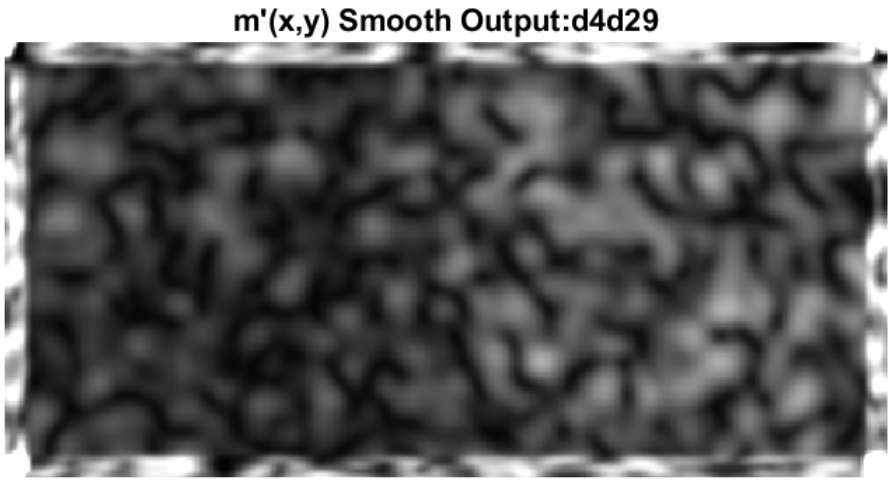

Figure 24: d4d29: Smooth Filtered Output Image

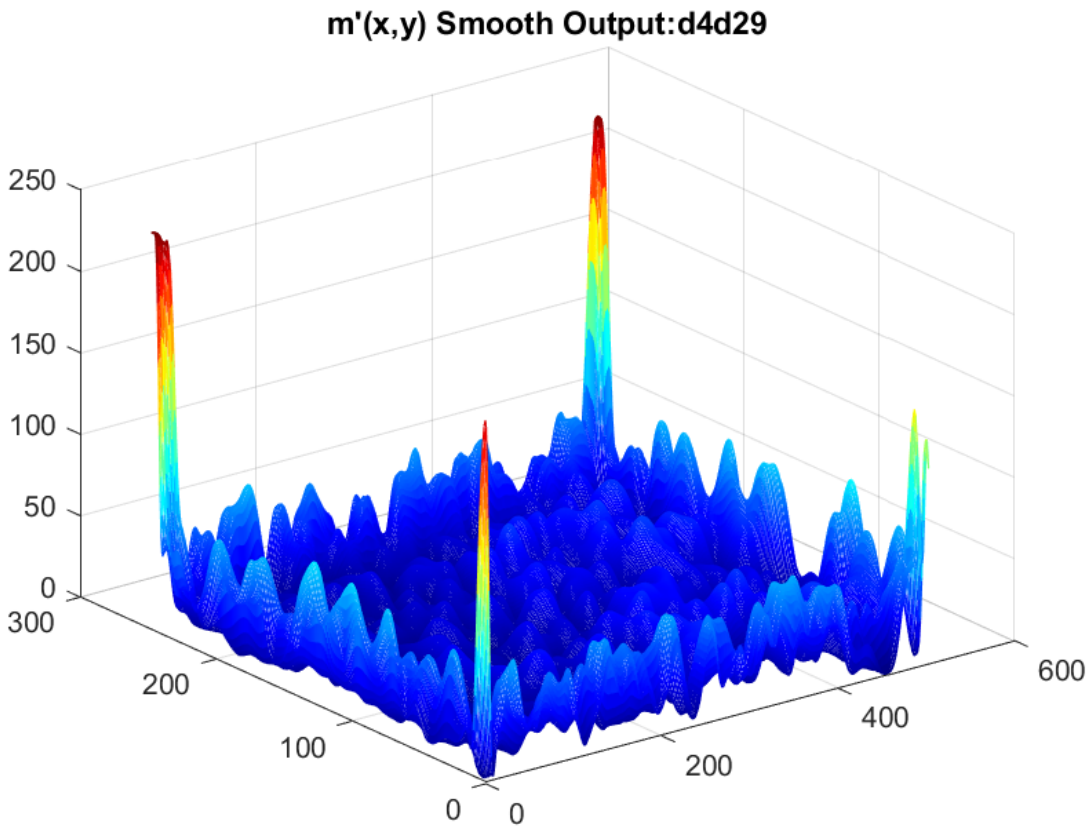

Figure 25: d4d29: 3-D Plot of Smooth Filtered Output Image

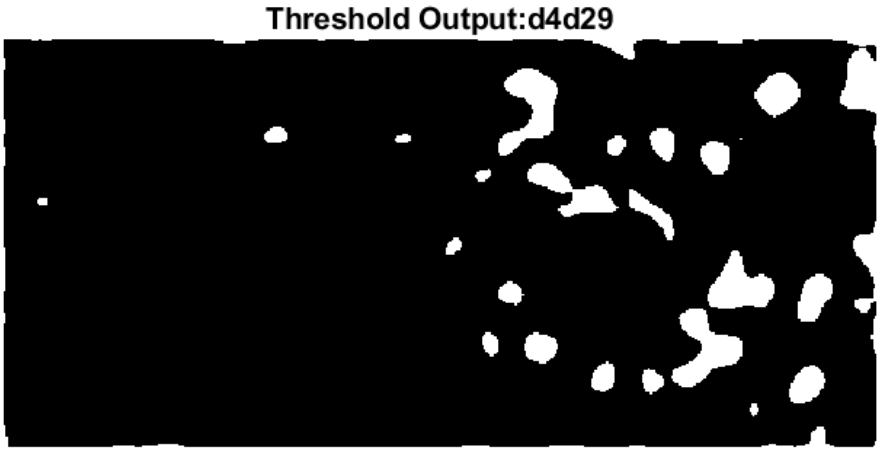

Figure 26: d4d29: Thresholded Image

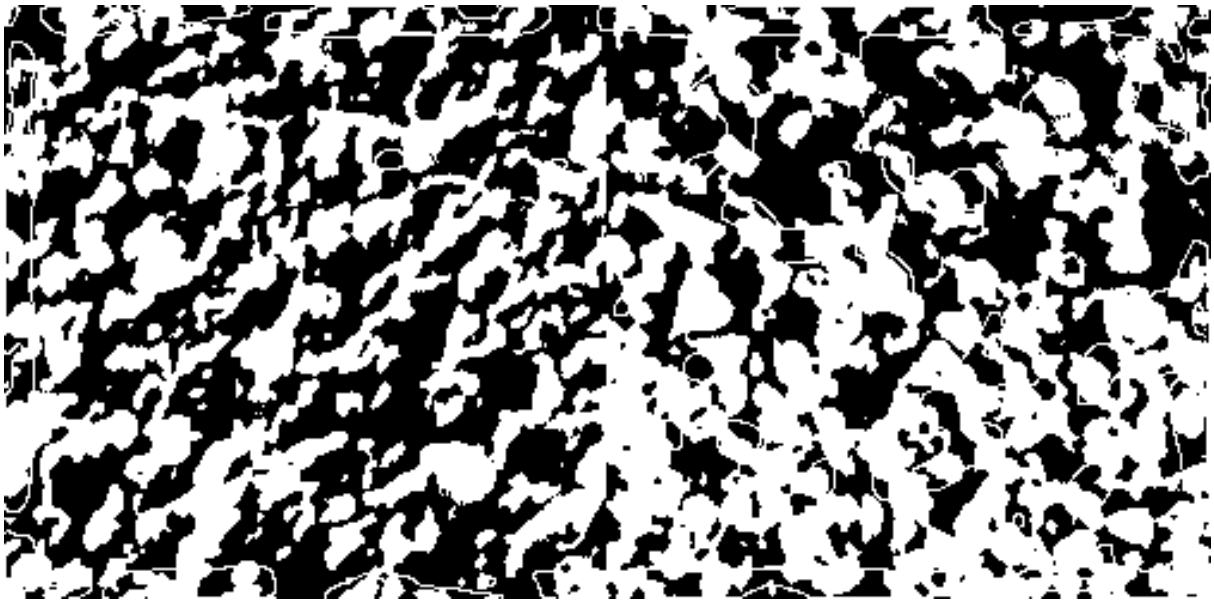

Figure 27: d4d29: Segmented Image

The texture segmentation result for this input image d4d29.gif is not as desired. It was really difficult to choose parameters for this image due to which the segmentation is not perfect. The limits for this segmentation cannot be stated as it is not in the form of a vertical or horizontal line. Ideally the line of segmentation must be the vertical line x=252.

4 Conclusion

After completing the necessary tasks to meet the objectives of this project, the following conclusions can be made:

- Gabor filter can be effectively used to perform texture segmentation. The filter gave good segmentation results with respect to texture1.gif and texture2.gif while finding the right parameter values for d9d77 and d4d29 was really difficult which led to average segmentation results on these images.
- Choosing appropriate parameter values is an important step to achieve good segmentation results.
- Applying a smoothing filter on the gabor filter output is really useful in reducing error.
- Thresholding reduces the image size as they are not zero-padded. So, care must be taken while applying thresholding to small images.