S2 6201011631188 โสภณ สุขสมบูรณ์

2 กุมภาพันธ์ 2565

ปฏิบัติการที่ 3 ปริภูมิความถี่

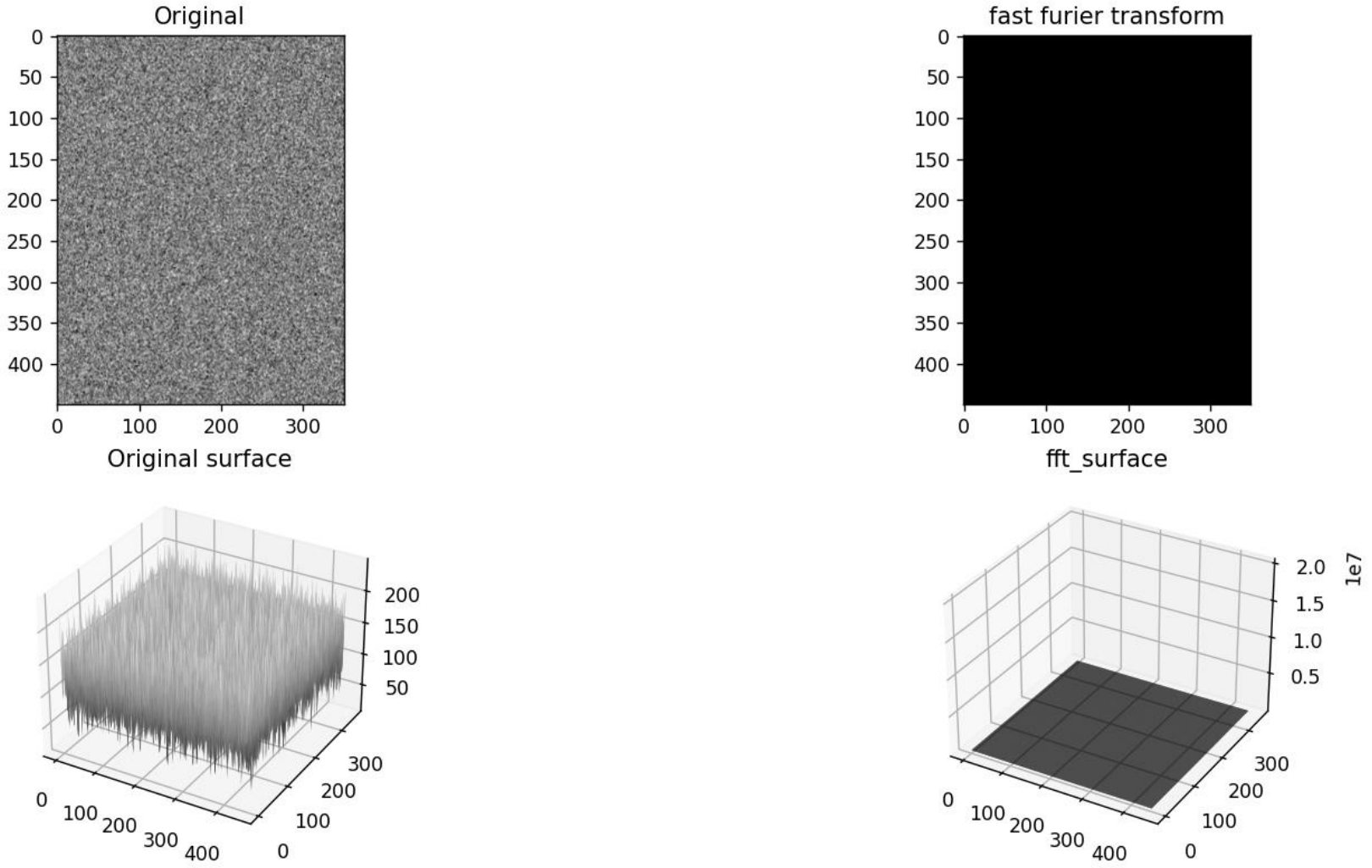
การทดลองที่ 3.1

1. ให้นักศึกษาทำการอ่านรูปภาพ figure1.jpg ดัง Python code ต่อไปนี้แล้วพิจารณาผลที่ได้โดย แสดงผลภาพเดิมและผลที่ได้จากการคำนวณ ทั้งแบบปกติและข้อมูลภาพเชิงพื้นผิว

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

#original
img=cv2.imread('D:\Telecom_Lab\Lab3\Figure1.jpg',0)
fig=plt.figure()
plt.suptitle('Figure1')
plt.subplot(221)
plt.imshow(img,cmap='gray')
plt.title('Original')

#fft
fft=np.fft.fft2(img)
fft_vis=np.abs(fft)
plt.subplot(222)
plt.imshow(fft_vis,cmap='gray')
plt.title('fast furier transform')
```

```
#3D
xx,yy=np.mgrid[0:img.shape[0],0:img.shape[1]]
ax=fig.add_subplot(223,projection='3d')
ax1=fig.add_subplot(224,projection='3d')

ax.plot_surface(xx,yy,img,rstride=1,cstride=1,linewidth=0,cmap='gray')
ax.set_title('Original surface')

ax1.plot_surface(xx,yy,fft_vis,rstride=1,cstride=1,linewidth=0,cmap='gray')
ax1.set_title('fft_surface')


plt.show()
```

Figure1


Original
0
50
100
150
200
250
300
350
400
0
100
200
300
Original surface
200
150
100
50
300
200
100
0
0
100
200
300
400
fast furier transform
0
50
100
150
200
250
300
350
400
0
100
200
300
fft_surface
2.0
1.5
1.0
0.5
1e7
300
200
100
0
0
100
200
300
400

2. ทำคำสั่งขั้นต้นอีกครั้งโดยเพิ่มคำสั่ง fftshift() แล้วแสดงผลภาพที่ได้จากการคำนวณ ทั้งแบบปกติ และข้อมูลภาพเชิงพื้นผิวเช่นกัน

fftshift_vis = np.fft.fftshift(fft_vis)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

#Original
fig=plt.figure()
img=cv2.imread('D:\Telecom_Lab\Lab3\Figure1.jpg',0)
plt.subplot()
plt.subplot(231)
plt.title('Original')
plt.imshow(img,cmap='gray')

#fft
fft=np.fft.fft2(img)
fft_vis=np.abs(fft)
plt.subplot(232)
plt.title('Fast Fourier Transforms')
plt.imshow(fft_vis,cmap='gray')

#fft shift
fftshift_vis=np.fft.fftshift(fft_vis)
plt.subplot(233)
plt.title('FFTshift')
plt.imshow(fftshift_vis,cmap='gray')

#3D plot
```

```
xx,yy=np.mgrid[0:img.shape[0],0:img.shape[1]]

ax=fig.add_subplot(234,projection='3d')
ax1=fig.add_subplot(235,projection='3d')
ax2=fig.add_subplot(236,projection='3d')


ax.plot_surface(xx,yy,img,rstride=1,cstride=1,linewidth=0,cmap='gray')
ax.set_title('Original surface')

ax1.plot_surface(xx,yy,fft_vis,rstride=1,cstride=1,linewidth=0,cmap='gray')
ax1.set_title('fft_surface')

ax2.plot_surface(xx,yy,fftshift_vis,rstride=1,cstride=1,linewidth=0,cmap='gray')
ax2.set_title('fft_shift_surface')

plt.show()
```

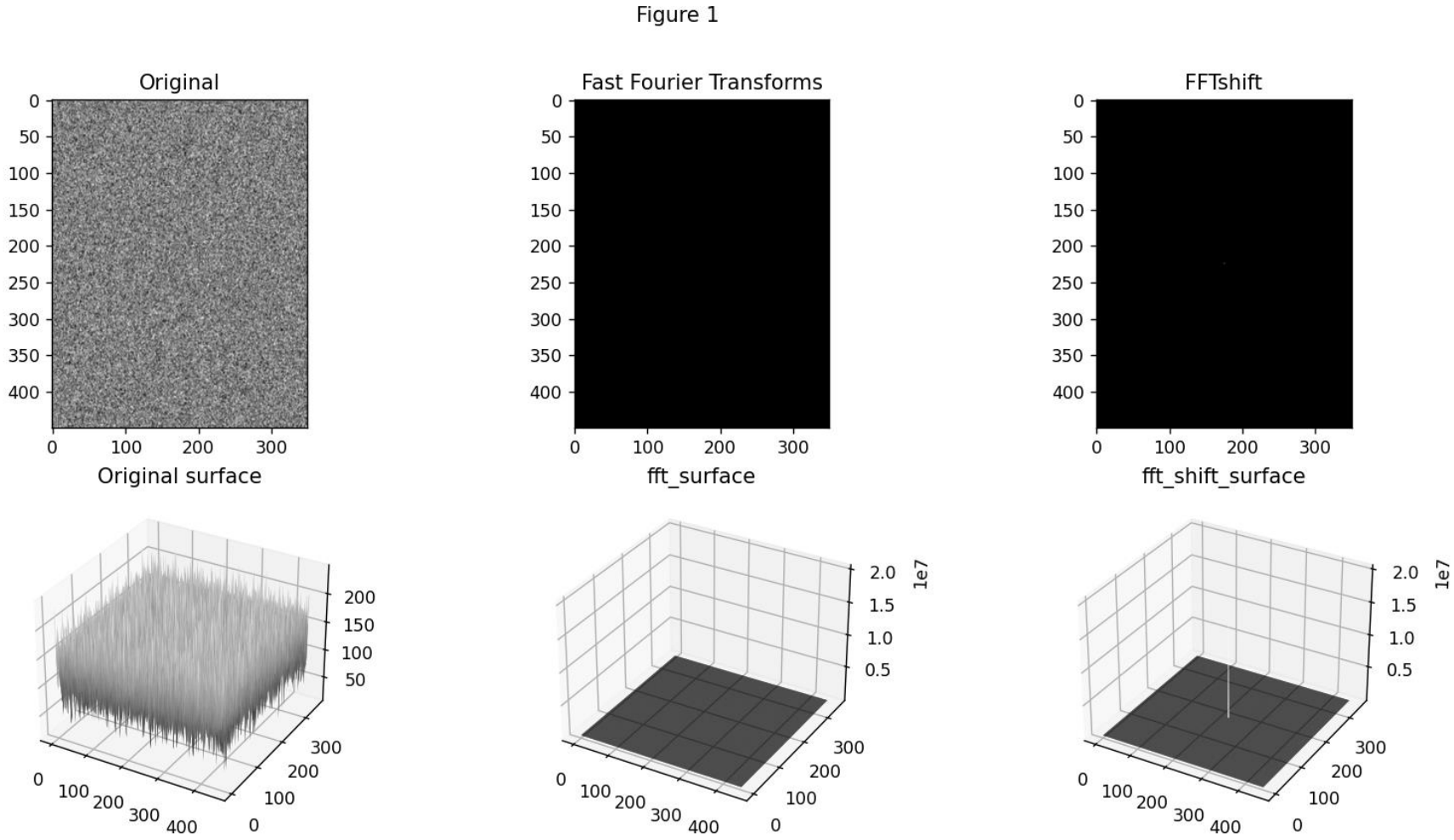


Figure 1
Original
Fast Fourier Transforms
FFTshift
Original surface
fft_surface
fft_shift_surface
1e7

3. ให้นักศึกษาอธิบายคำสั่ง abs() และ fftshift()

-***คำสั่ง abs()*** เป็นคำสั่งที่หาค่าสัมบูรณ์ของข้อมูล นั่นคือเมื่อข้อมูลที่เป็นลบ เช่น

x=np.array[-3,-2,1,4]

#เมื่อป้อนคำสั่ง np.abs() จะได้

np.abs(x)

#จะได้

[3,2,1,4]

ดังนั้นคำสั่งนี้จะเป็นการขจัดค่าที่เป็นลบออกทั้งหมดนั่นเอง

-***คำสั่ง fftshift()*** เป็นคำสั่งที่จะทำการเลื่อนค่าจากIndex0ของความถี่ไปยังตรงกลางศูนย์กลางของสเปกตรัม เช่น

```
import numpy as np
freq=np.fft.fftfreq(6,d=1./6).reshape(2,3)
print(freq)
freq1=np.fft.fftshift(freq,axes=(1,))
print(freq1)
```

จะได้ ค่า freq คือ [0,1,2],[-3, 2,-1] เมื่อทำการshift จะเป็นการเลื่อนตำแหน่ง 0 และ -3 จากทั้ง2 Array ไปตรงกลางของข้อมูล จะได้ค่าเป็น [2,0,1] 0 ถูกเลื่อนไปยังตำแหน่งไปตรงกลางของชุดข้อมูล และ [-1,-3,2] ค่า-3ถูกเลื่อนไปยังตรงกลางของข้อมูลเช่นกัน

4. พิจารณารูป figure2.jpg ถึง figure5.jpg และดำเนินการดังข้อ 1. และ 2. และอภิปราย

สำหรับชุดคำสั่งสำหรับ Figure2 จนถึง Figure5 นั้น ใช้คำสั่งเดียวกันกันคำสั่งในข้อ2. แตกต่างกันเพียงแค่คำสั่ง plt.suptitle('***ชื่อของรูปภาพ***') และคำสั่ง img=cv2.imread('***Addressของรูปภาพ***')

สำหรับการแสดงผล

Figure2

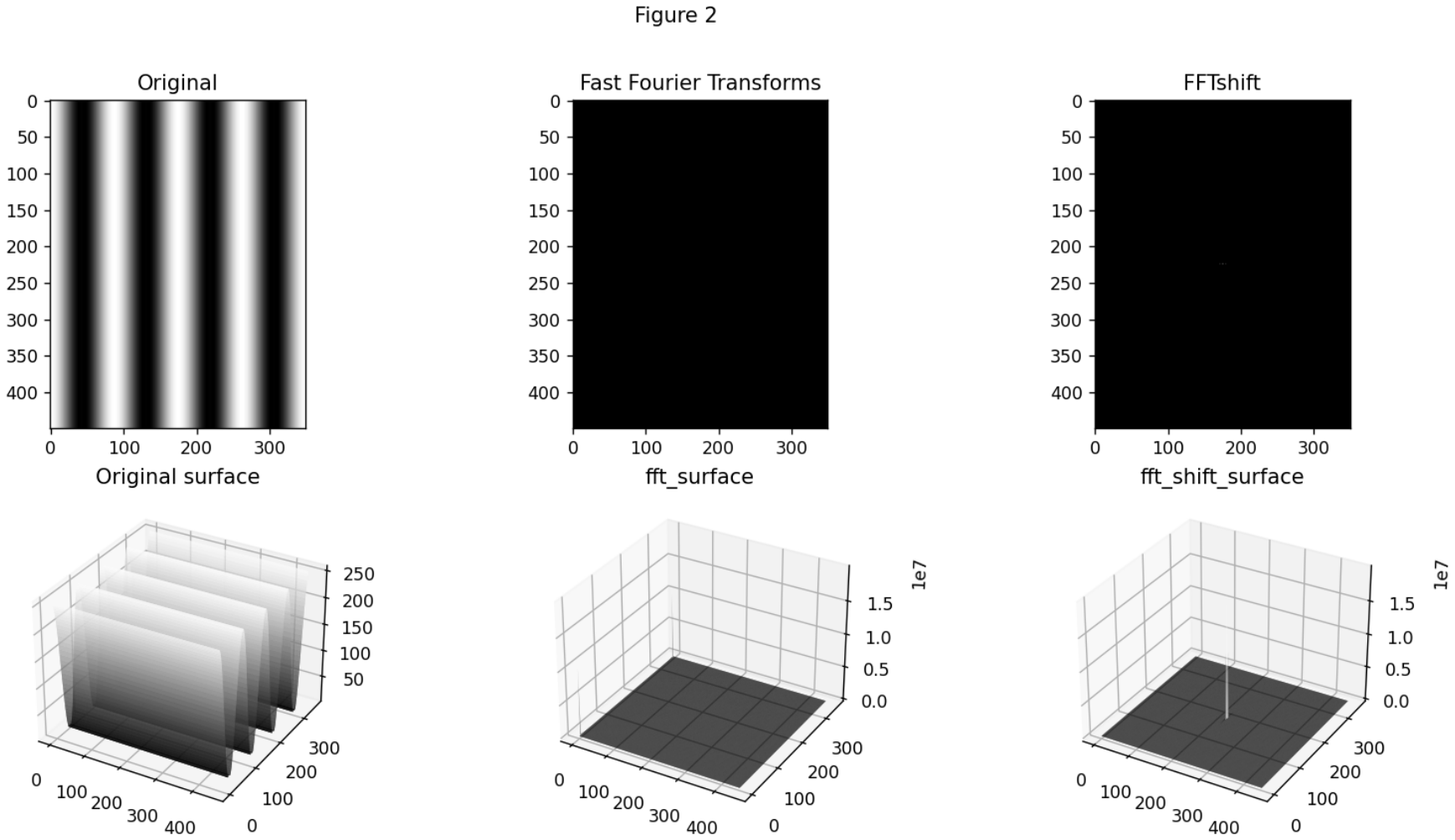


Figure3

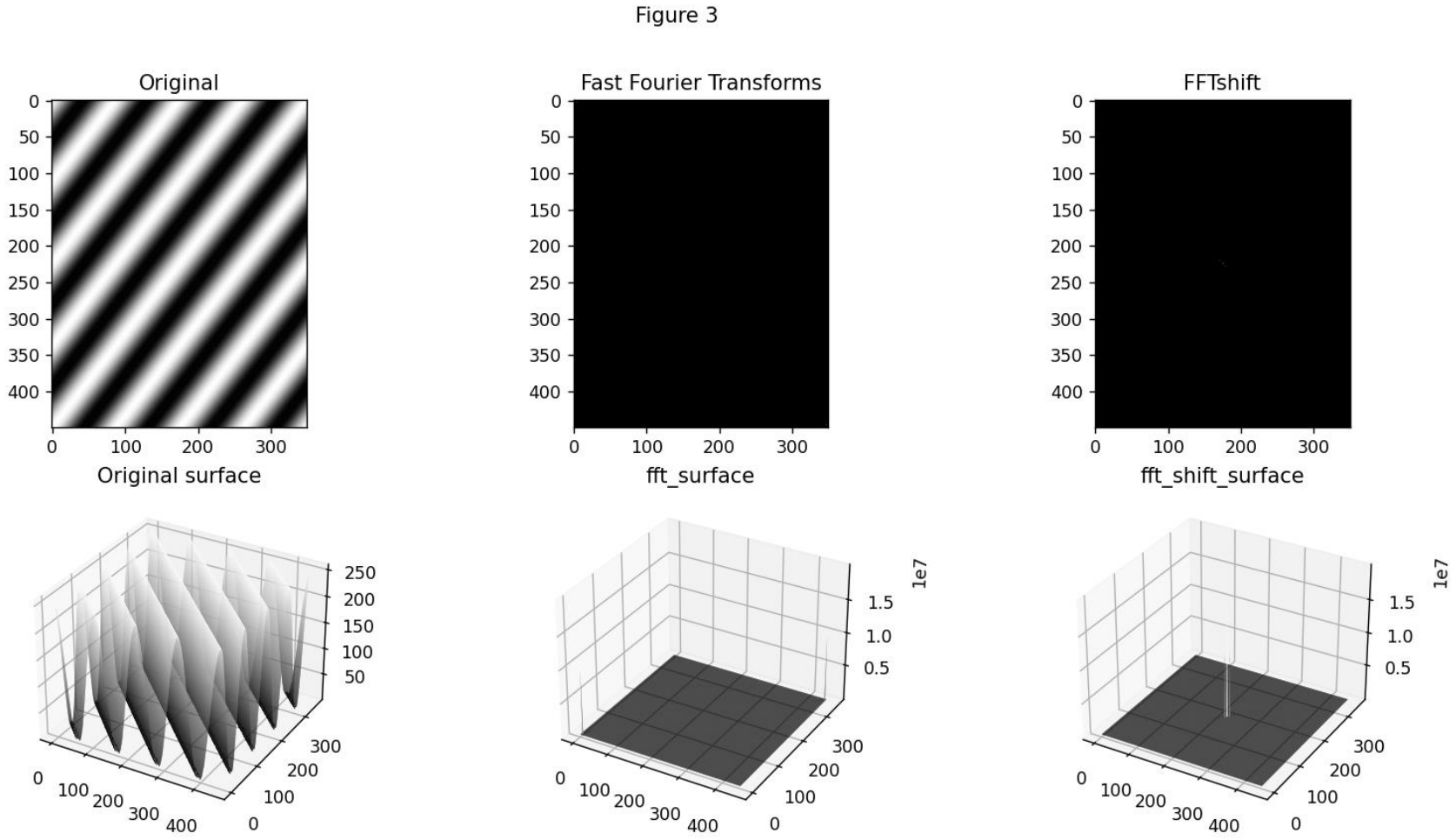


Figure 3
Original
Fast Fourier Transforms
FFTshift
Original surface
fft_surface
fft_shift_surface
1e7
1e7

Figure4

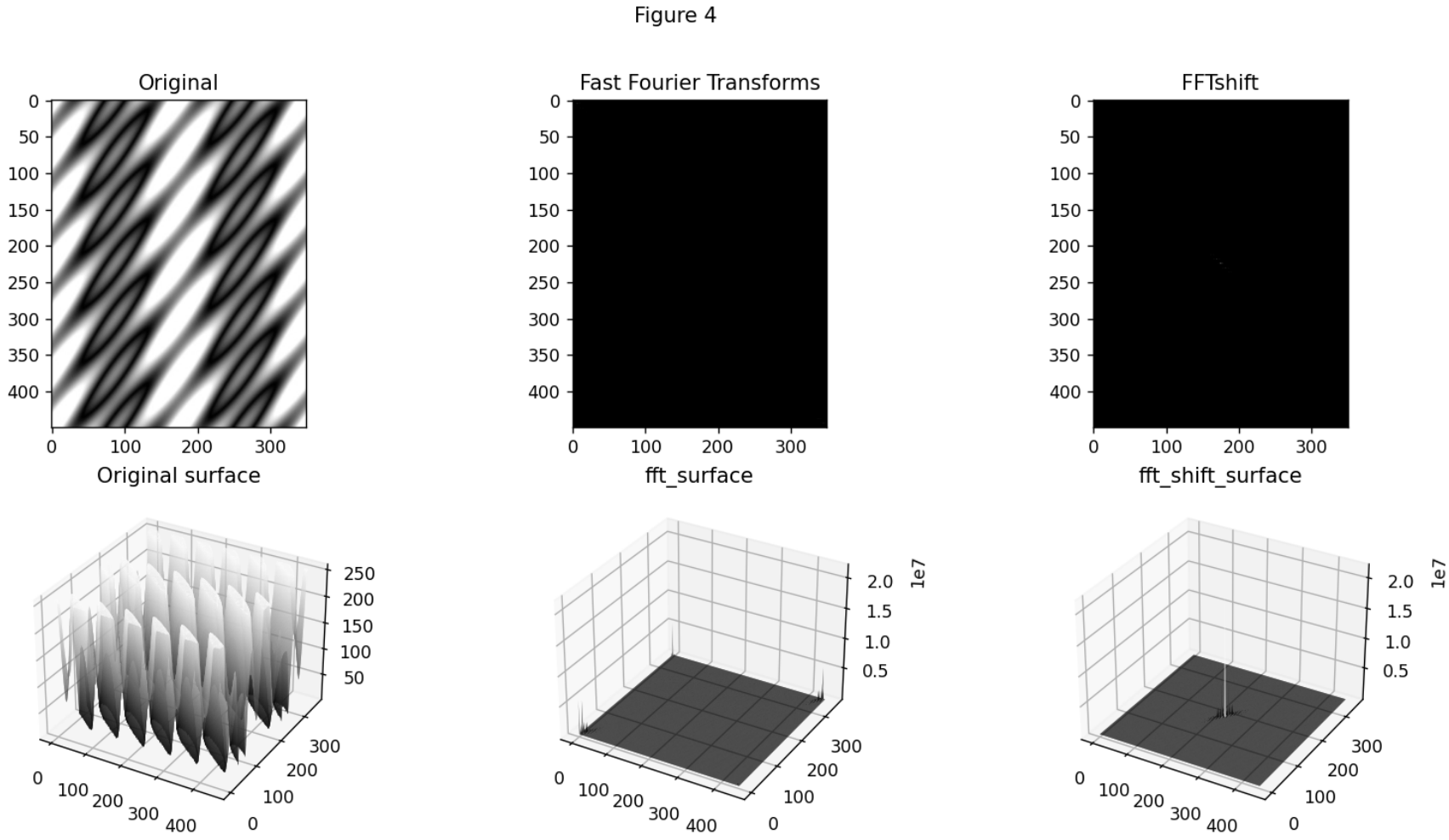


Figure 4
Original
Original surface
Fast Fourier Transforms
fft_surface
FFTshift
fft_shift_surface
1e7

Figure5

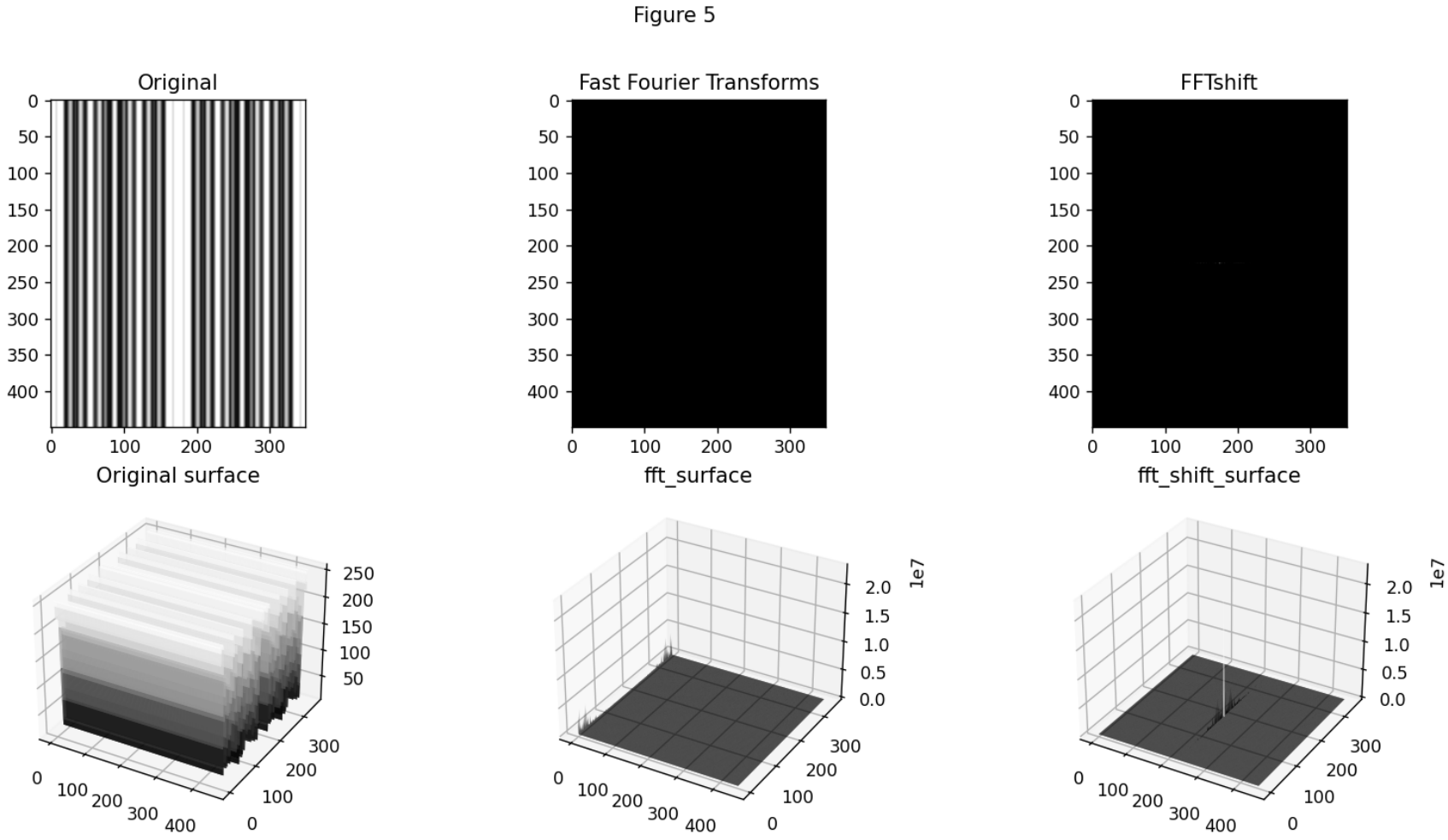


Figure 5
Original
Fast Fourier Transforms
FFTshift
Original surface
fft_surface
fft_shift_surface
1e7

อภิปราย

-จากข้อ1.เมื่อทำการRunโปรแกรมในรูปที่1 เราจะมองไม่ออกว่า กราฟนั้นคืออะไร มีลักษณะอย่างไร แต่เมื่อทำ3D plot ของรูปที่2 (Original image) จะพบว่าSurface plot ของภาพที่2 นั้น มีลักษณะคล้ายฟังก์ชันไซน์ จึงอนุมานโดยรวมได้ว่า ภาพนั้นเกิดจากฟังก์ชันไซน์ที่มีหลายๆฟังก์ชันซ้อนกันในรูปภาพ แต่ภาพที่2จะเห็นชัดที่สุดว่า ภาพที่เราเห็นนั้นเป็นฟังก์ชันไซน์ ดังภาพด้านล่างนี้

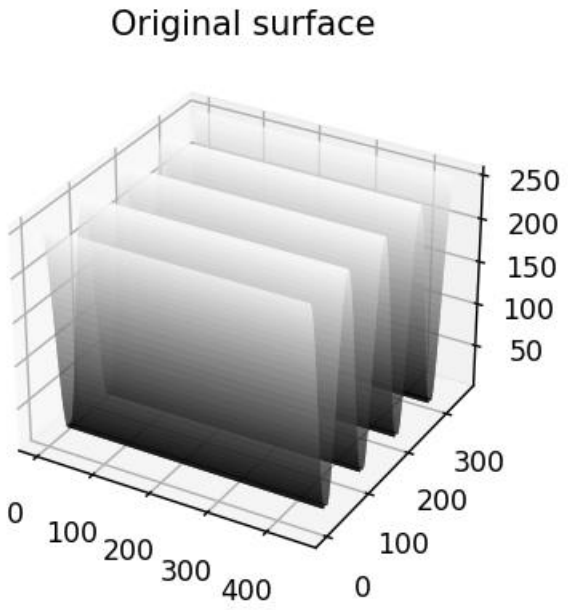
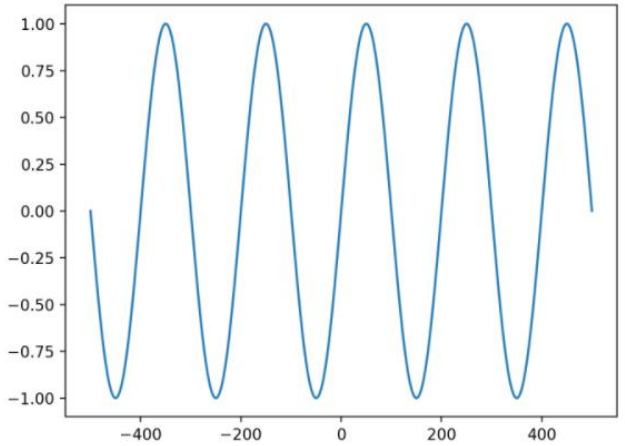
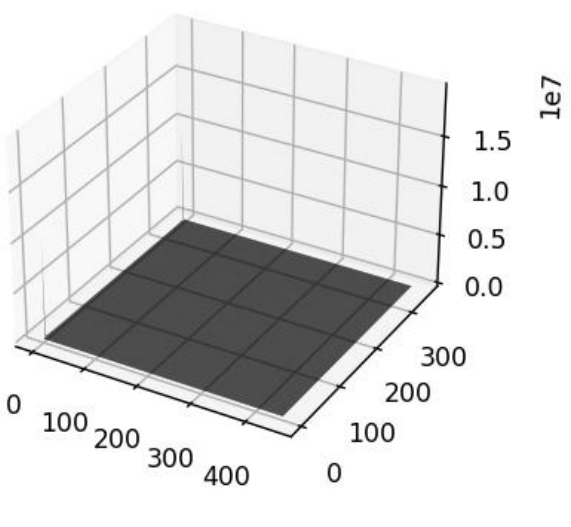


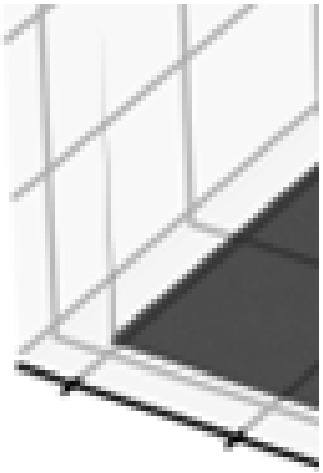
Figure2 - 3D plot

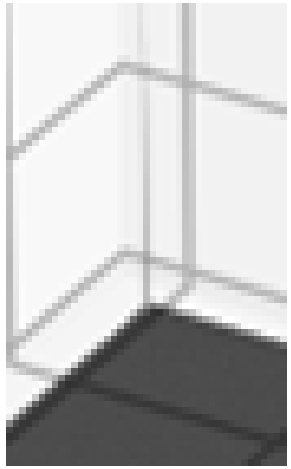


2D-Sinusoidal Function

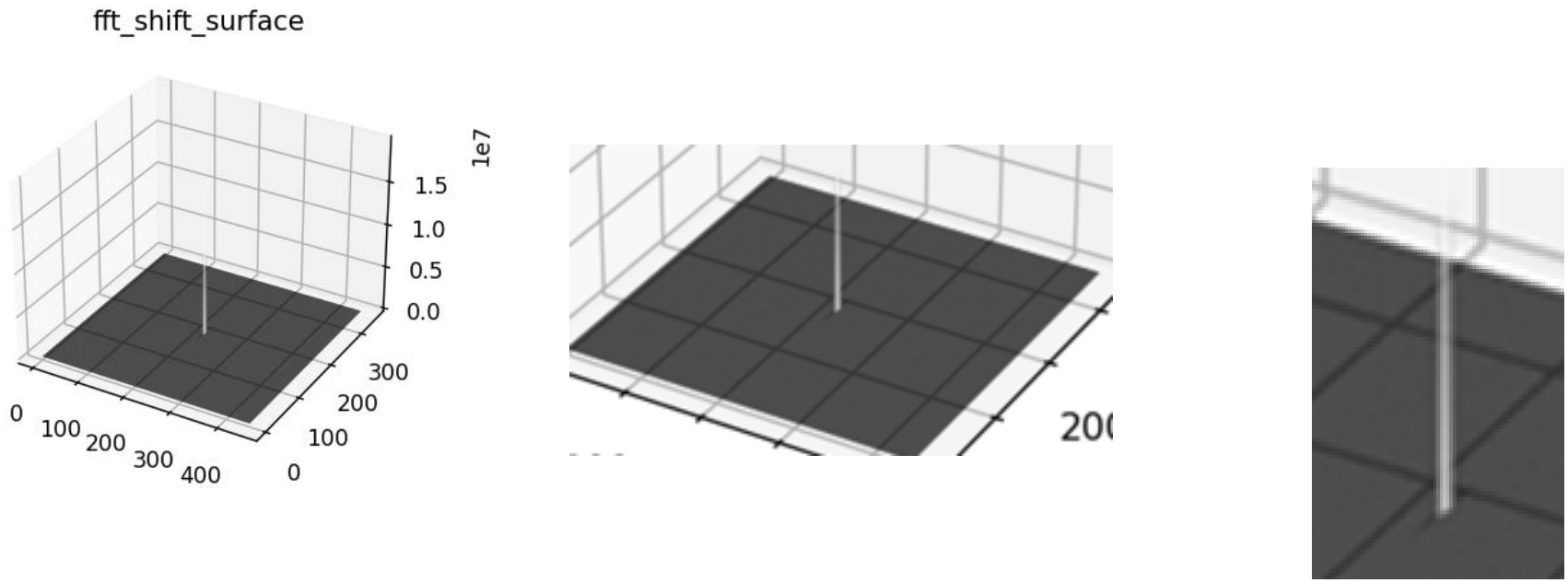
-เมื่อทำFFTของภาพดังกล่าวด้วยคำสั่ง np.fft.fft2 เมื่อทำการZoomภาพดู จะพบว่ากราฟที่ได้จะพบว่ามี DC Component หรือ องค์ประกอบกระแสตรง ที่ปรากฏในกราฟนั่นเอง แต่ด้วยวิธีนี้ทำให้ยากต่อความเข้าใจและมองออกได้ยาก จึงเพิ่มคำสั่ง np.fft.fftshift จะทำการเลื่อนค่า DC Component ไปยังตำแหน่งศูนย์กลางของกราฟนั่นเอง เราจะเห็นได้จากรูปด้านล่างดังนี้







จากรูปข้างต้น เมื่อทำการซูมภาพดู ณ ตำแหน่งขอบของกราฟจะพบว่ามีเส้นอะไรสักอย่างพุ่งขึ้นมา นั่นคือ DC Component นั่นเอง แต่เราไม่ได้พิกัดตรงนั้นคือเท่าไหร่กันเมื่อมองด้วยตาเปล่า จึงใช้คำสั่ง np.fft.fftshift()



เมื่อใช้คำสั่ง np.fft.fftshift() จะสังเกตว่า DC component ทั้งหมดได้มารวม ณ ตำแหน่งตรงกลางแล้วนั่นเอง จากนั้นเมื่อทำการพิจาณาต่อ จะพบว่าตัวกราฟนั้นมีลักษณาคล้ายกับสเปกตรัมของสัญญาณใดๆ ทั้งนี้เราสามารถบอกได้ว่า เมื่อเราทำ FFT และ shift FFT ของ Image ใดๆ เราจะได้ Spectrum ของภาพนั้นๆ เฉกเช่นเดียวกับการทำ FFT ของสัญญาณใดๆนั่นเอง

-ซึ่งจำนวน DC component หรือเส้นสเปกตรัม จะมากหรือน้อยขึ้นอยู่กับจำนวนฟังก์ชันไซน์ที่รวมกันภายในภาพ ดังเช่น Figure4 , Figure5 ซึ่งมีเส้นสเปกตรัมมากกว่า Figure2 , Figure3 อย่างชัดเจน ดังภาพที่แสดง

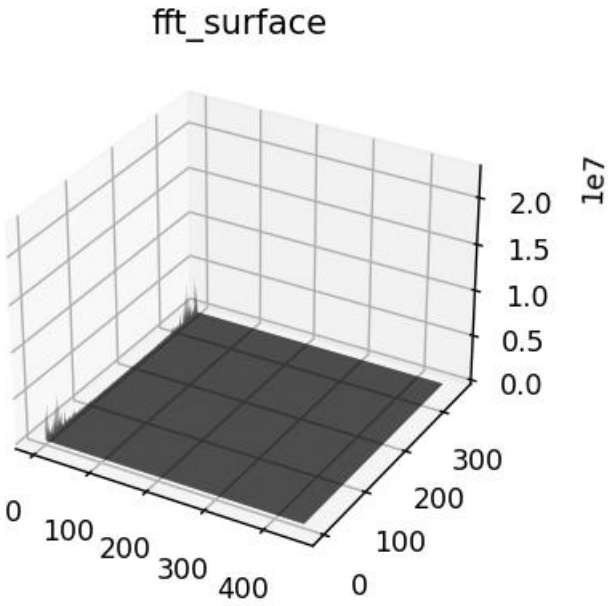


Figure4

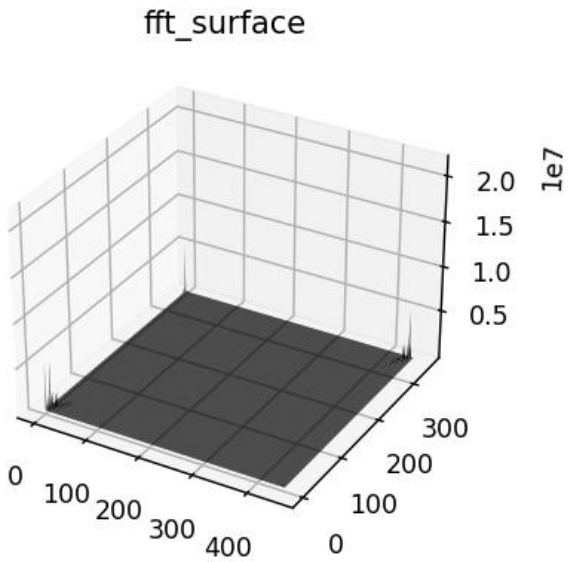


Figure5

-ซึ่งสเปกตรัมจะเป็นตัวบอกการเปลี่ยนแปลงของแถบสี เช่น สีทั้ง7ที่ผ่านปริซึมก็จะมีสเปกตรัมที่แตกต่างกันออกไป แต่เนื่องจากเราเรียนเกี่ยวกับ Grayscale สเปกตรัมนี้จะบอกการเปลี่ยนแปลงของขอบของภาพ ณ ตำแหน่งต่างๆ ภาพที่มีรายละเอียดยิบย่อย หรือมีการเปลี่ยนแปลงของภาพเยอะก็จะมีจำนวนเส้นสเปกตรัมมาก-น้อย ขึ้นกับแต่ละสถานการณ์ เช่น ในภาพที่2 และ 3 จะเห็นเส้นสเปกตรัมน้อยกว่า ภาพที่ 4 และ 5 อันเนื่องมาจากภาพ2และ3นั้นมีการเปลี่ยนแปลงของขอบของรูปภาพน้อยกว่าภาพที่3 และ 4 นั่นเอง

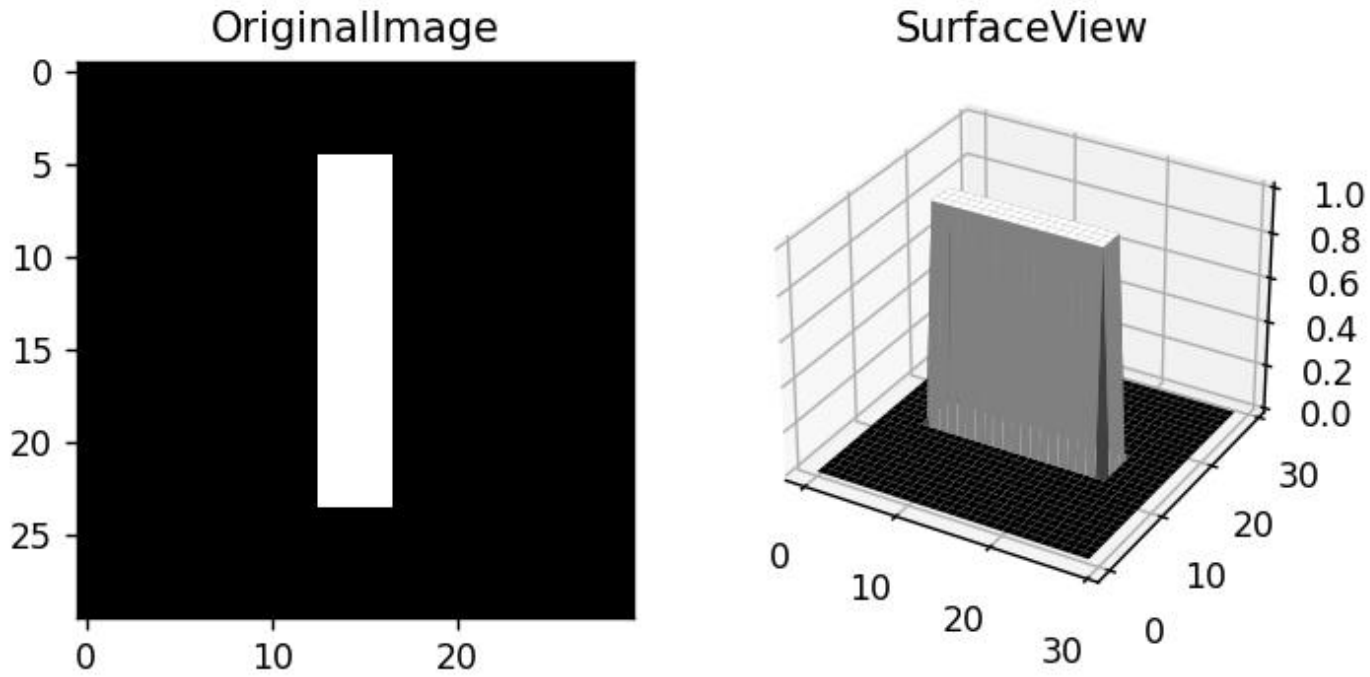
การทดลองที่3.2

1. ให้นักศึกษาพิมพ์คำสั่งดังต่อไปนี้ เพื่อวิเคราะห์ Fourier spectrum ของรูปภาพดิจิทัล และอธิบายการทำงานของ Python code พอสังเขป

<u>คำสั่งที่1</u>

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
img = np.zeros((30,30))
img[5:24,13:17] = 1

fig=plt.figure()
plt.subplot(121)
plt.imshow(img,cmap='gray')
plt.title('OriginalImage')

xx,yy=np.mgrid[0:img.shape[0],0:img.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,img,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('SurfaceView')

plt.show()
```

การแสดงผล



คำอธิบาย

-จากคำสั่ง np.zeros() เป็นคำสั่งที่จะทำการสร้างArrayที่มีขนาดตามที่เรากำหนด ซึ่งค่าทั้งหมดในArrayนั้นจะมีค่าเป็น 0

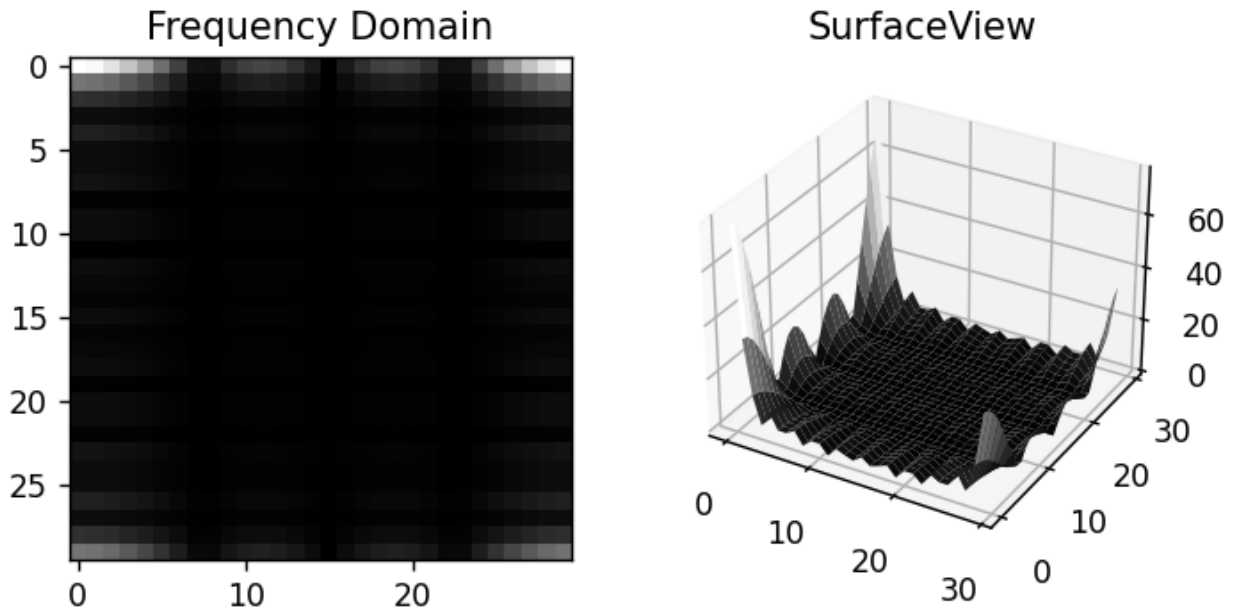
-และคำสั่ง img[5:24,13:17]=1 เป็นคำสั่งที่เขียนขึ้นเพื่อกำหนดให้ค่า ณ ตำแหน่งดังกล่าวเป็น 1

-เพื่อต้องการแสดงผลให้รูปของกราฟจึงใช้คำสั่ง plt.imshow ตามปกติ แต่เนื่องจากเราต้องการที่เป็น3Dด้วยนั้นจึงต้องเพิ่มคำสั่ง np.mgrid,fig.add_subplot สำหรับการแสดงผลเป็น3Dและ ax.plot_surface เป็นคำสั่งสำหรับการกำหนดความหนากว้างของ3D

คำสั่งที่2

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

img = np.zeros((30,30))
img[5:24,13:17] = 1
fft = np.fft.fft2(img)
fft_abs = np.abs(fft)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fft_abs,cmap='gray')
plt.title('Frequency Domain')

xx,yy=np.mgrid[0:fft_abs.shape[0],0:fft_abs.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fft_abs,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('SurfaceView')

plt.show()
```

การแสดงผล



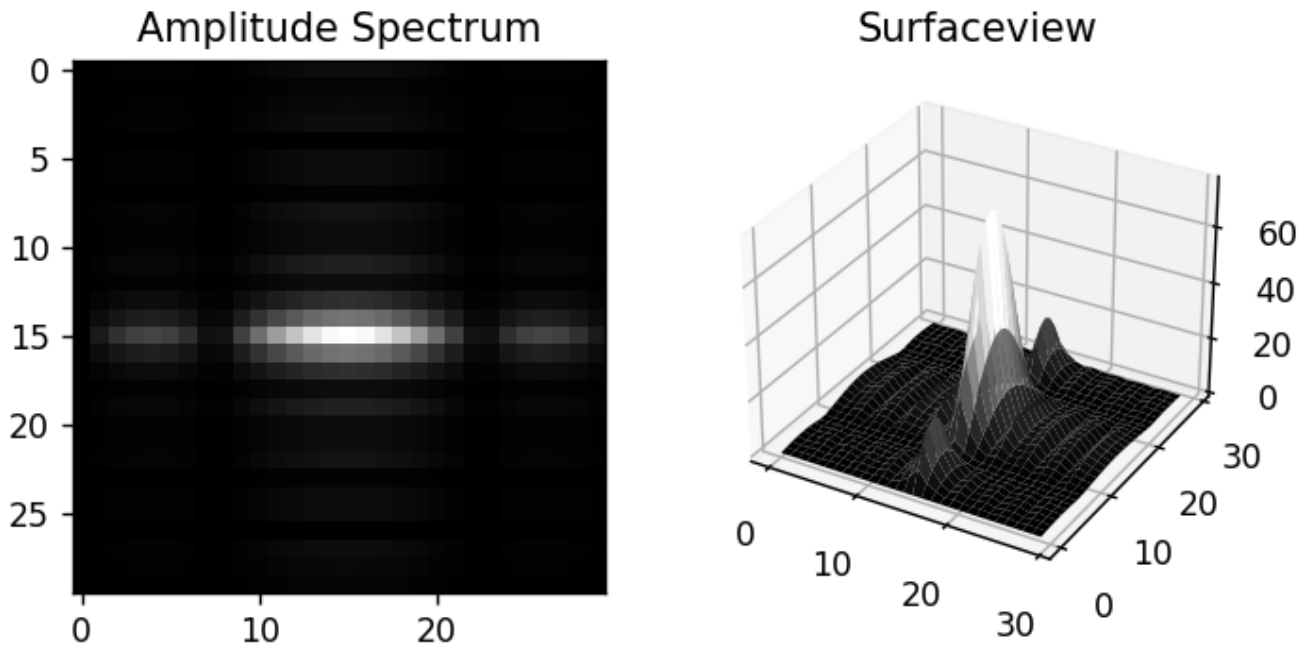
คำอธิบาย

จากคำสั่งในข้อก่อนหน้าเราได้ทำการเพิ่มคำสั่งใหม่เข้าไปคือ fft = np.fft.fft2() และ fft_abs = np.abs() ซึ่งเป็นคำสั่งที่ทำการแปลงข้อมูลให้อยู่ในรูป Frequency Domain โดยวิธี Fast Fourier Transfrom และเนื่องจากการแปลงFFTจะมีComplex Numberอยู่ในข้อมูลจึงครอบอีกชั้นด้วยคำสั่ง Absolute จะได้ดังภาพที่เห็นข้างต้น

คำสั่งที่3

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

img = np.zeros((30,30))
img[5:24,13:17] = 1
fft=np.fft.fft2(img)
fft_abs=np.abs(fft)
fftshift=np.fft.fftshift(fft)
fftshift_abs=np.abs(fftshift)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fftshift_abs,cmap='gray')
plt.title('Amplitude Spectrum')

xx,yy=np.mgrid[0:fftshift_abs.shape[0],0:fftshift_abs.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fftshift_abs,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('Surfaceview')

plt.show()
```

การแสดงผล



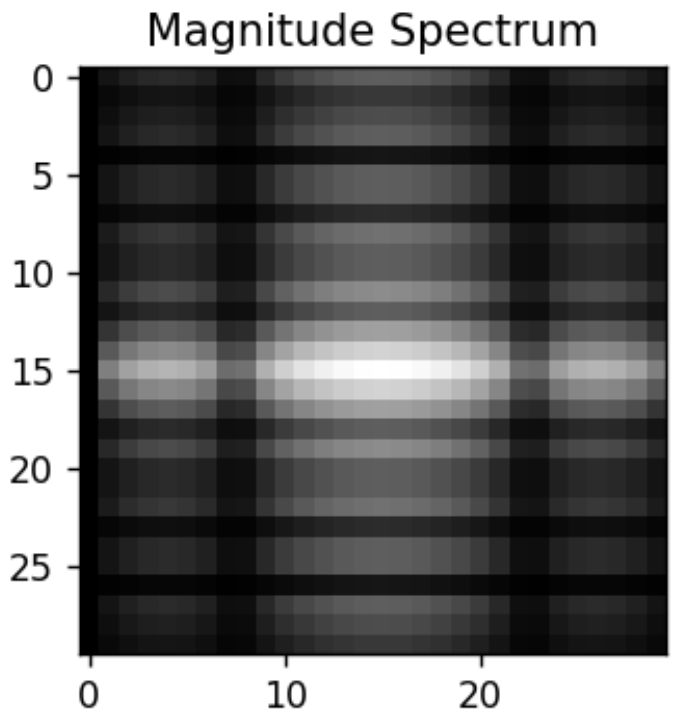
คำอธิบาย

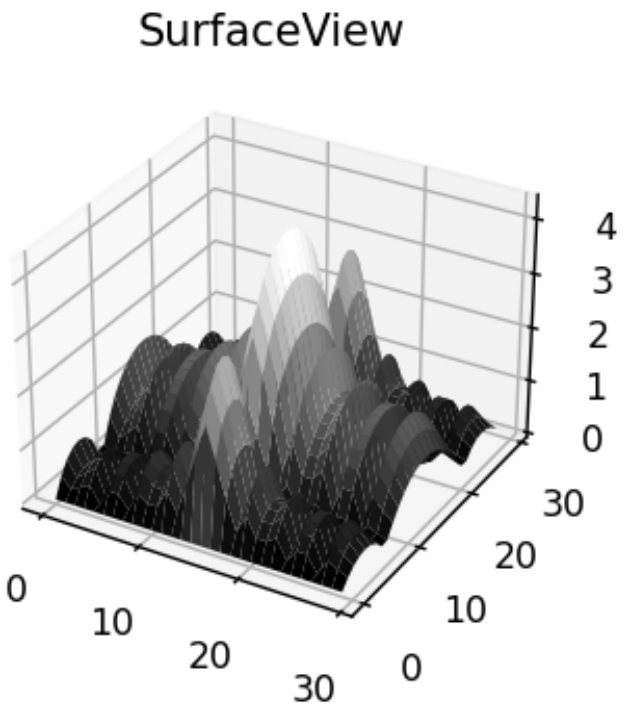
จากข้อก่อนหน้าเราจะเห็น DC Component ของภาพเช่นเดียวกับการทดลองที่3.1 จึงได้เพิ่มคำสั่ง np.fft.fftshift เพื่อทำการshiftค่าทั้งหมดไปไว้ยังกึ่งกลางของข้อมูล ซึ่งถ้าเราสังเกตจะพบว่ามันคือ Amplitude ของสัญญาณสัญญาณหนึ่ง ซึ่งจะกล่าวต่อไป

คำสั่งที่4

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

img = np.zeros((30,30))
img[5:24,13:17] = 1
fft = np.fft.fft2(img)
fft_abs = np.abs(fft)
fftshift=np.fft.fftshift(fft)
fftshift_abs=np.abs(fftshift)
fftshift_log=np.log(1+fftshift_abs)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fftshift_log,cmap='gray')
plt.title('Magnitude Spectrum')

xx,yy=np.mgrid[0:fftshift_abs.shape[0],0:fftshift_abs.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fftshift_abs,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('SurfaceView')

plt.show()
```

การแสดงผล



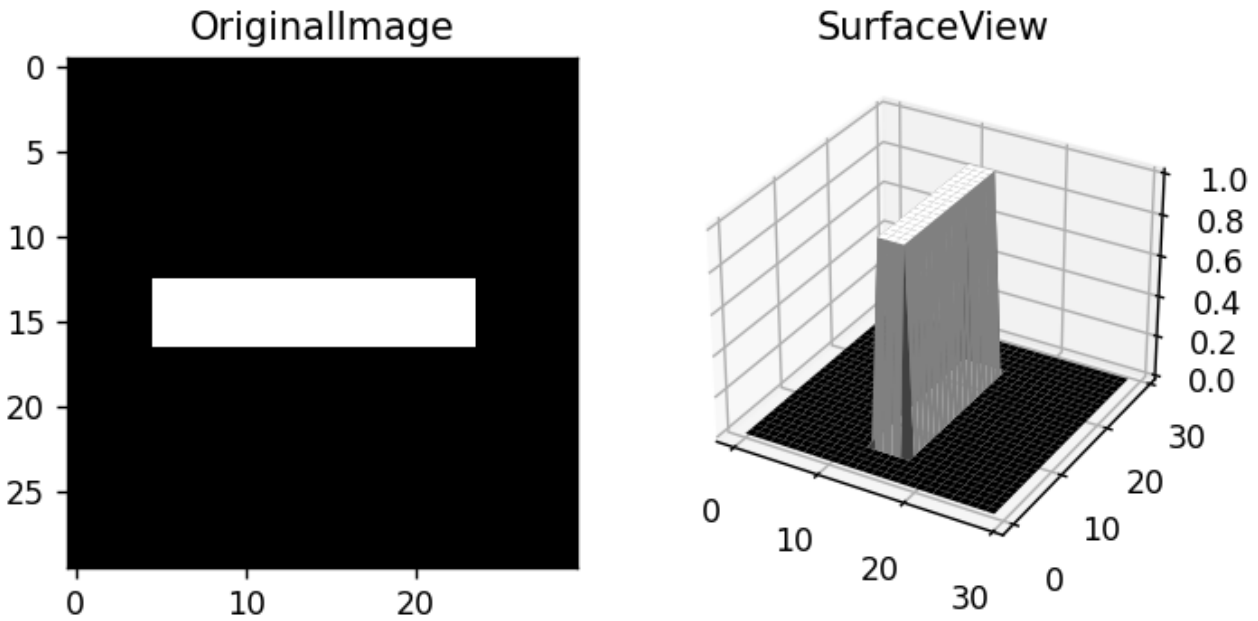


คำอธิบาย

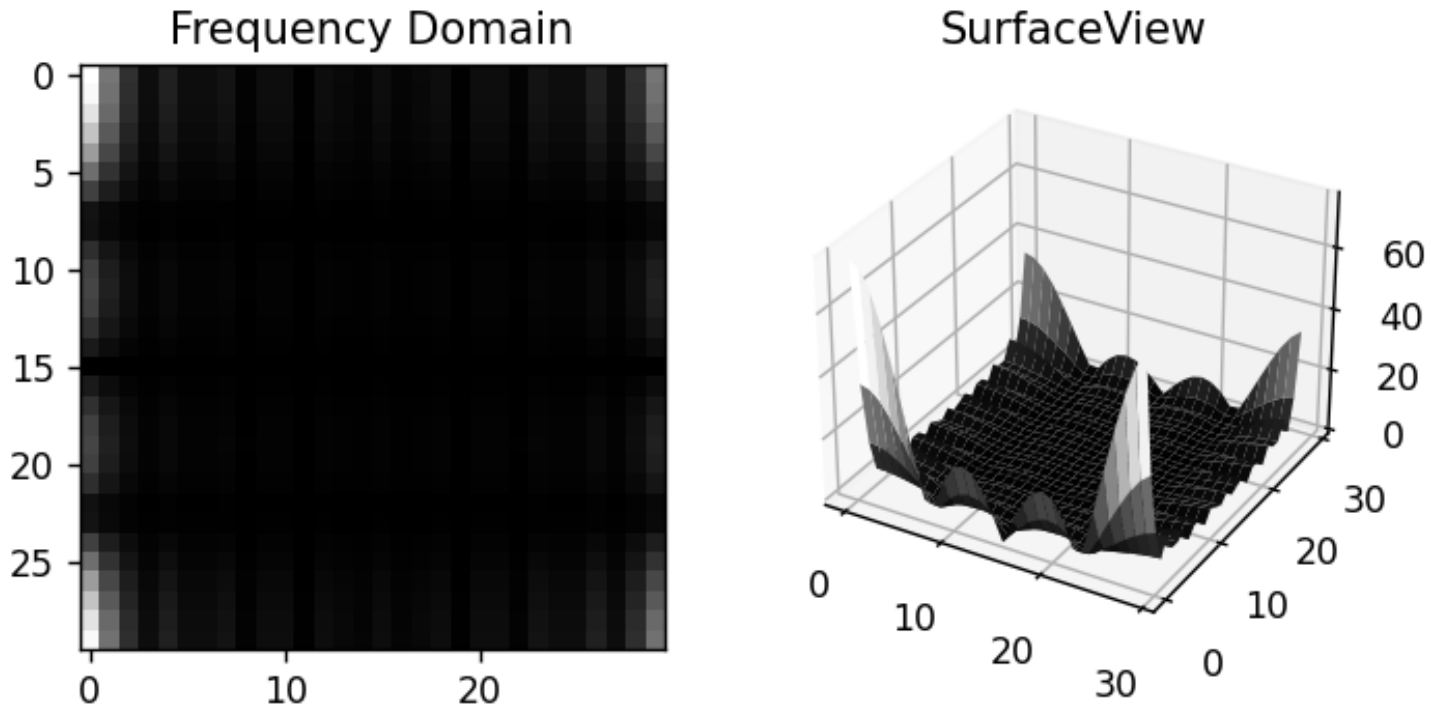
จากคำสั่งก่อนหน้า เราได้ทำการเพิ่มคำสั่ง np.log() ซึ่งเป็นคำสั่งหาค่า Magnitude นั่นเอง จากภาพจะเห็นรายละเอียดเพิ่มมากกว่าตัวของคำสั่งก่อนหน้า

2. ให้นักศึกษาวิเคราะห์ Fourier spectrum ของรูปที่ตั้งฉากกับภาพ img ในข้อ 1. โดยดำเนินการทุก ขั้นตอนเช่นเดียวกัน แล้วอภิปราย

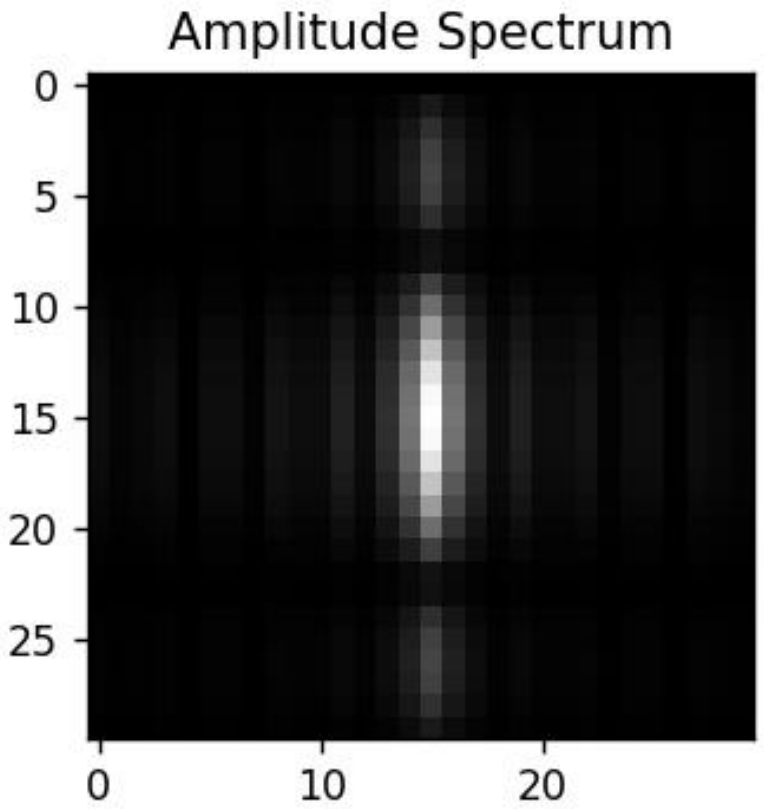
ขั้นตอนที่1

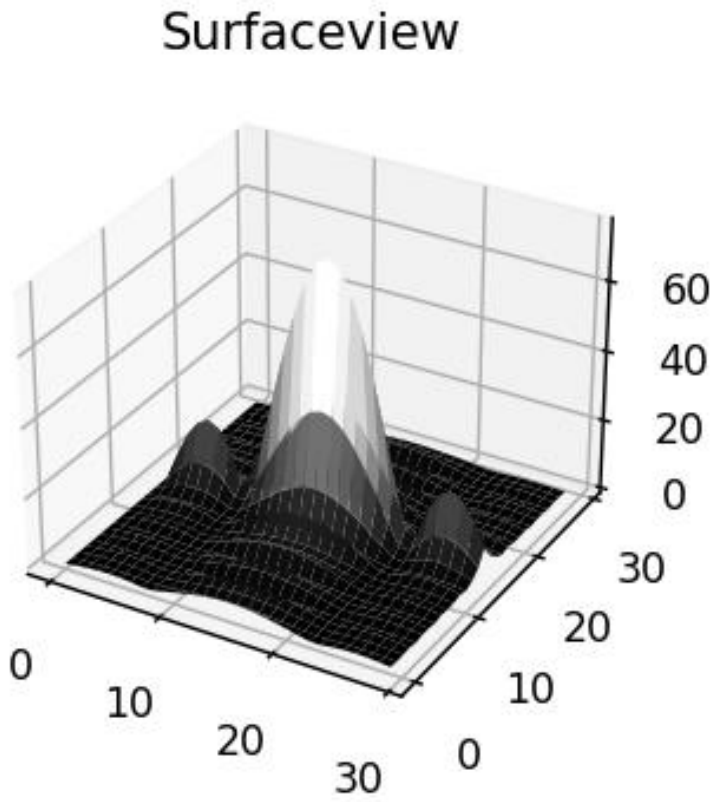


<u>ขั้นตอนที่2</u>

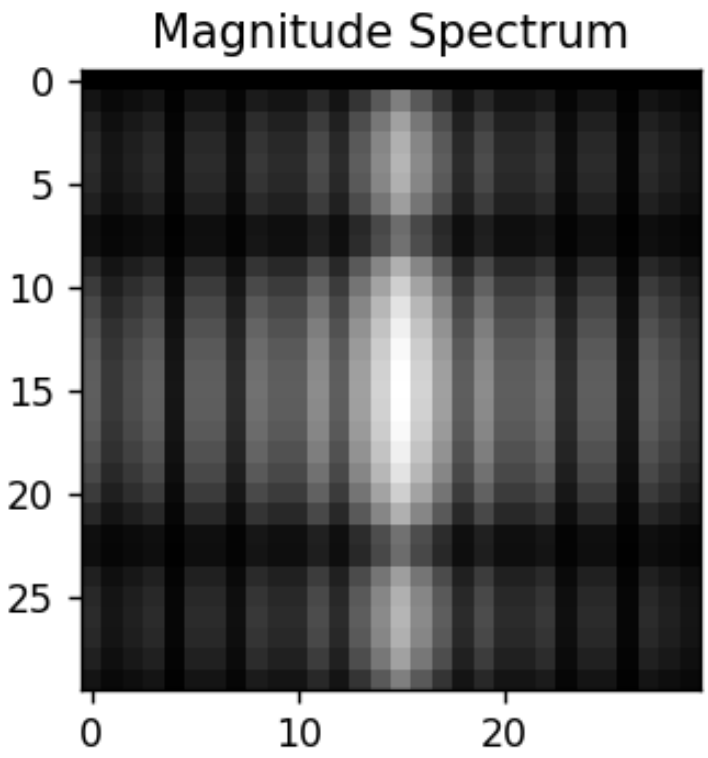

Frequency Domain
0
5
10
15
20
25
0
10
20
SurfaceView
60
40
20
0
30
20
10
0
0
10
20
30

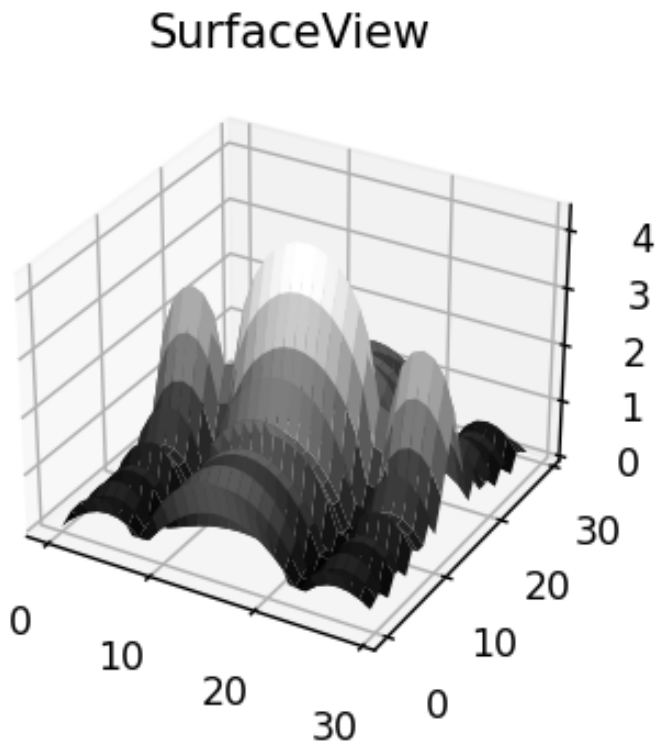
ขั้นตอนที่3


Amplitude Spectrum
0
5
10
15
20
25
0
10
20


Surfaceview
60
40
20
0
0
10
20
30
0
10
20
30

ขั้นตอนที่4


Magnitude Spectrum
0
5
10
15
20
25
0
10
20


SurfaceView
4
3
2
1
0
30
20
10
0
0
10
20
30

การอภิปราย

จากCodeในส่วน ***img[5:24,13:17]=1*** เมื่อทำมาPlotเป็นกราฟจะสังเกตว่าเป็นลักษณะของ Square wave ซึ่งในทางทฤษฎีที่เรียนในวิชา Signal และ Telecomunication รูปแปลงฟูเรียร์ของ ***Square wave*** ก็คือ ***Sinc Function*** นั่นเอง และสาเหตุที่การใช้คำสั่ง ***np.abs()***เพื่อเป็นการกำจัดจำนวนเชิงซ้อนที่เกิดจากการทำ Fast Fourier Transfrom นั่นเอง เพื่อให้แกนสำหรับการแสดงผลเป็น x และ y ไม่ใช่ Re และ Im นอกจากนี้ในส่วนของคำสั่ง ***np.log()*** เพื่อเป็นการหา Magnitude ของภาพข้างต้น

และจากคำสั่งในข้อที่3.2.2 ทำการพิจารณา Fourier Spectrum รูปที่ตั้งฉากกับ img ผลลัพธ์ที่ได้จากการทดลองนั้นพบว่า รูปตั้งฉากของimg นั้น Fourier Spectrum ของภาพนั้นไม่มีความแตกต่างจากภาพต้นแบบ แต่พบว่ามีเพียงสิ่งเดียวที่แตกต่างกับภาพต้นแบบ นั่นคือทิศทางของSinc Function ที่เมื่อนำ Magnitude Spectrum ของภาพทั้งสองกันมาเปรียบเทียบกัน จะพบว่า ตั้งฉากซึ่งกันและกันเท่านั้น แต่ขนาด รูปร่าง นั้นไม่แตกต่างกันเลย ซึ่งสามารถสรุปได้ว่า การปรับตำแหน่งของภาพ , หมุนภาพ หรือ กระทำการใดๆ หากขนาดของภาพไม่มีการเปลี่ยนแปลง จะไม่ส่งผลต่อขนาดและรูปร่างของ Fourier Spectrum ของภาพดังกล่าวนั่นเอง

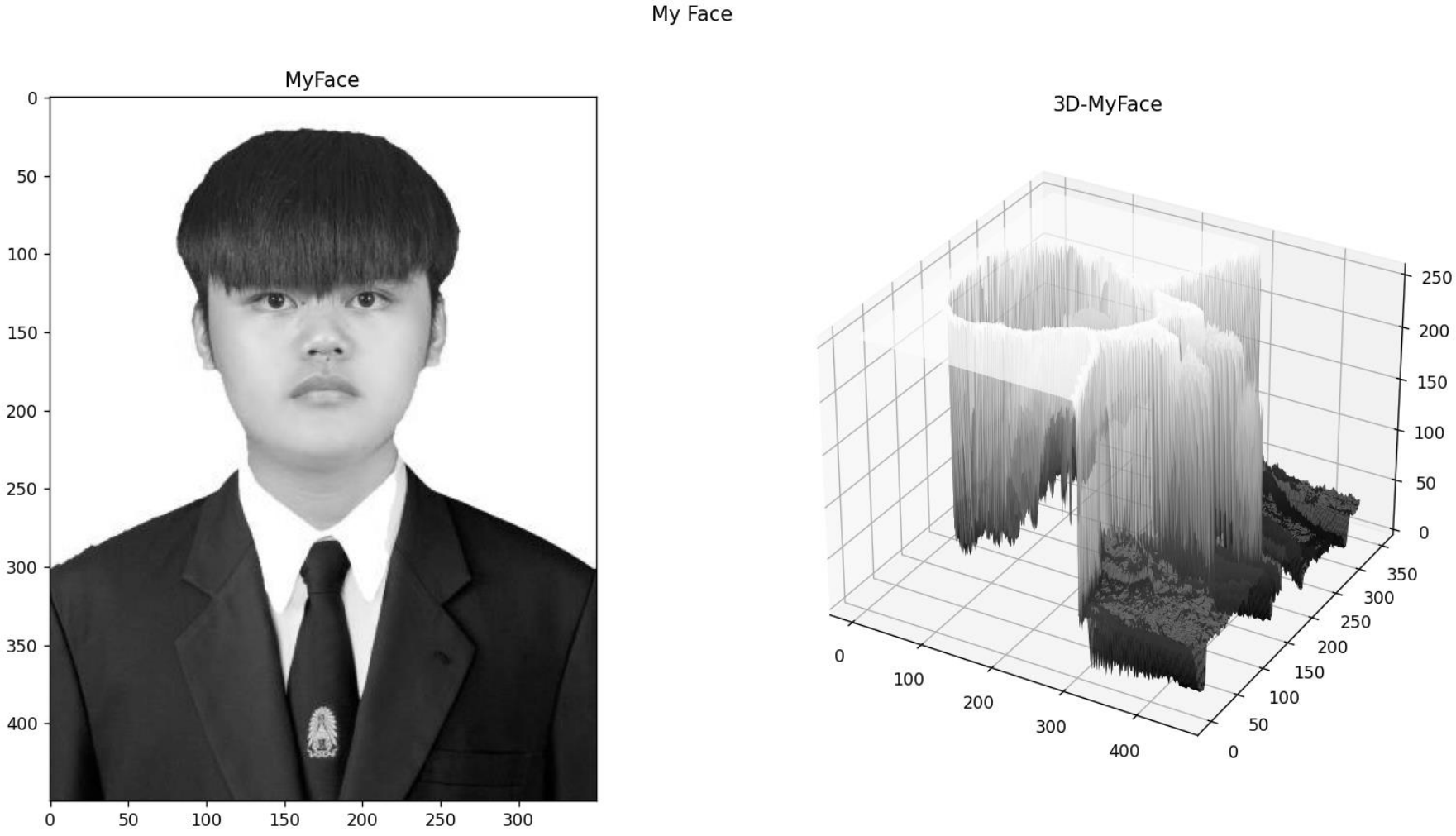
การทดลองที่3.3

1. ให้นักศึกษาวิเคราะห์ Fourier spectrum ของรูปใบหน้าตัวเอง โดยดำเนินการดังการทดลองที่ 3.2 และอภิปราย

-ขั้นตอนที่1 นำเข้ารูปภาพและแสดงผลในรูปแบบ3D

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

MyFace=cv2.imread('D:\Telecom_Lab\Lab3\MyFace.jpeg',cv2.IMREAD_GRAYSCALE)
fig=plt.figure()
plt.suptitle('My Face')
plt.subplot(121)
plt.title('MyFace')
plt.imshow(MyFace,cmap='gray')


xx,yy=np.mgrid[0:MyFace.shape[0],0:MyFace.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,MyFace,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('3D-MyFace')

plt.show()
```

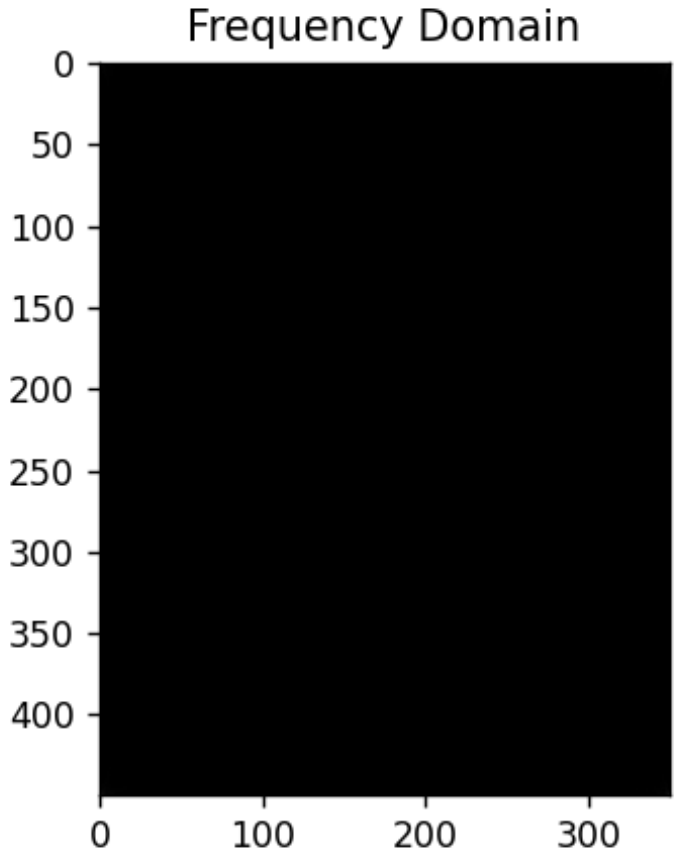
การแสดงผล

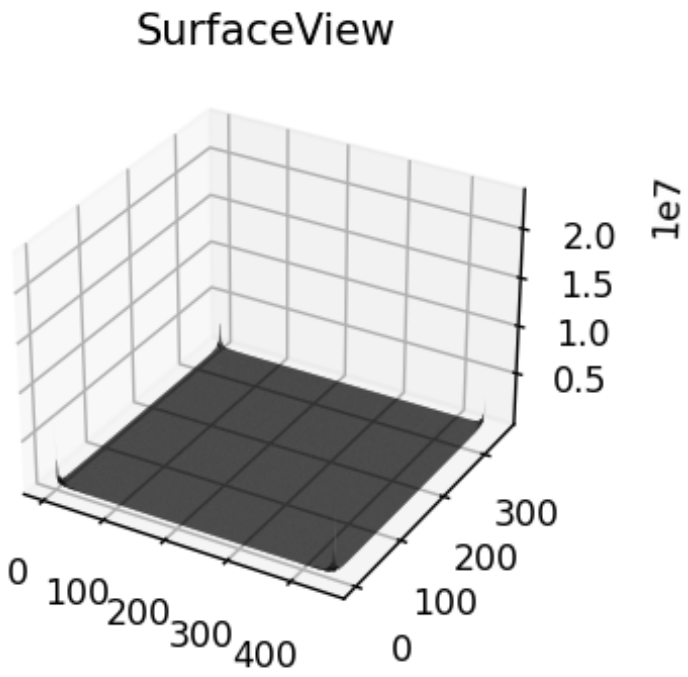


-***ขั้นตอนที่2 แปลงภาพด้วยFFT***

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

Myface=cv2.imread('D:\Telecom_Lab\Lab3\MyFace.jpeg',cv2.IMREAD_GRAYSCALE)
fft = np.fft.fft2(Myface)
fft_abs = np.abs(fft)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fft_abs,cmap='gray')
plt.title('Frequency Domain')

xx,yy=np.mgrid[0:fft_abs.shape[0],0:fft_abs.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fft_abs,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('SurfaceView')

plt.show()
```

การแสดงผล

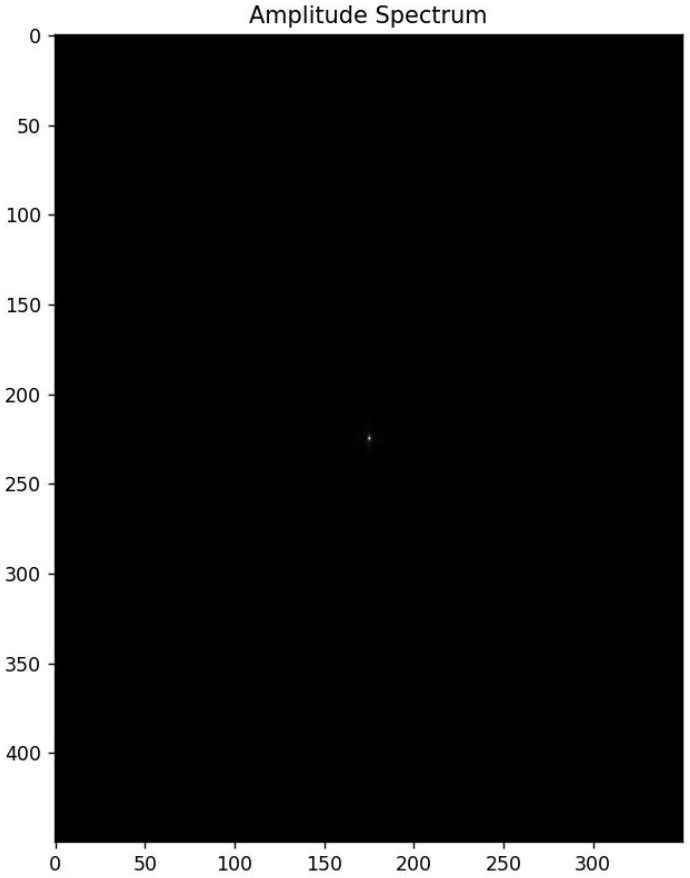

Frequency Domain
0
50
100
150
200
250
300
350
400
0
100
200
300

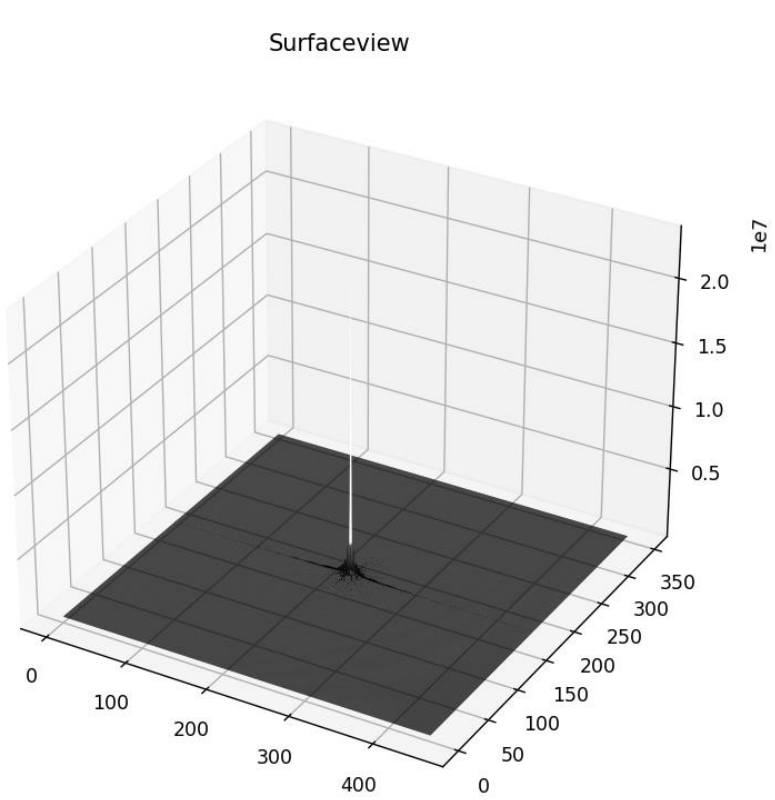

SurfaceView
2.0
1.5
1.0
0.5
1e7
0
100
200
300
400
0
100
200
300

ขั้นตอนที่3 ทำการShifting

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

Myface=cv2.imread('D:\Telecom_Lab\Lab3\MyFace.jpeg',cv2.IMREAD_GRAYSCALE)
fft=np.fft.fft2(Myface)
fft_abs=np.abs(fft)
fftshift=np.fft.fftshift(fft)
fftshift_abs=np.abs(fftshift)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fftshift_abs,cmap='gray')
plt.title('Amplitude Spectrum')

xx,yy=np.mgrid[0:fftshift_abs.shape[0],0:fftshift_abs.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fftshift_abs,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('Surfaceview')

plt.show()
```

การแสดงผล

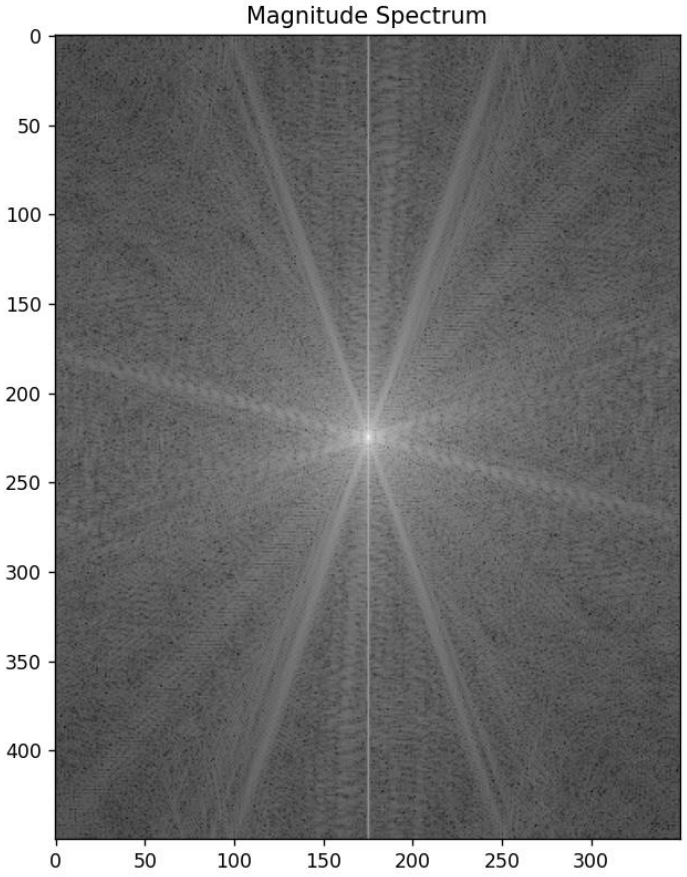

Amplitude Spectrum
0
50
100
150
200
250
300
350
400
0
50
100
150
200
250
300

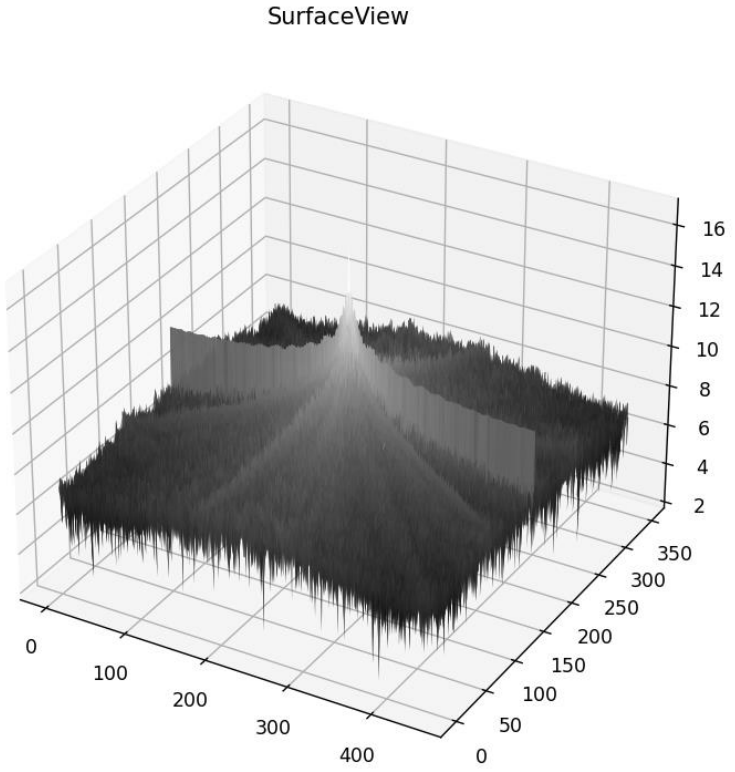

Surfaceview
1e7
2.0
1.5
1.0
0.5
0
100
200
300
400
0
50
100
150
200
250
300
350

ขั้นตอนที่4 ใช้คำสั่งLogarithmเพื่อหา Magnitudeของภาพ

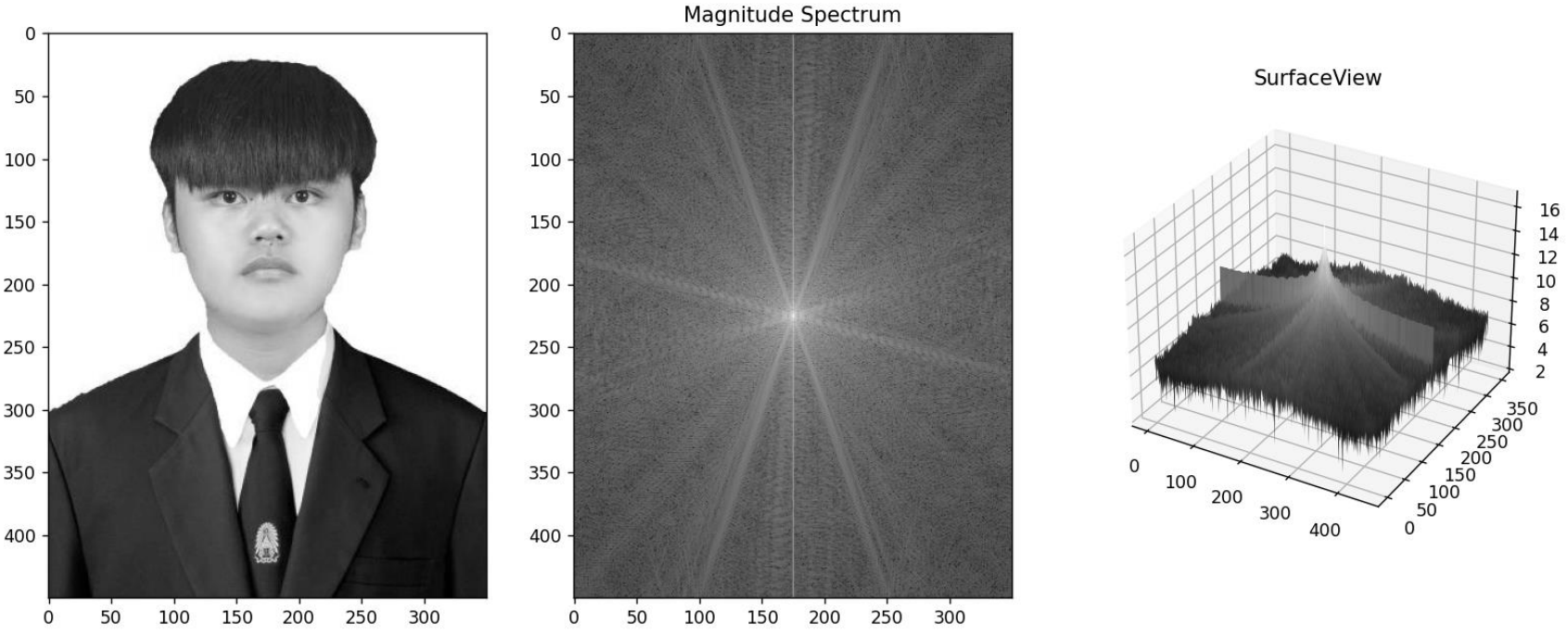
```
import numpy as np
import matplotlib.pyplot as plt
import cv2

Myface=cv2.imread('D:\Telecom_Lab\Lab3\MyFace.jpeg',cv2.IMREAD_GRAYSCALE)
fft = np.fft.fft2(Myface)
fft_abs = np.abs(fft)
fftshift=np.fft.fftshift(fft)
fftshift_abs=np.abs(fftshift)
fftshift_log=np.log(1+fftshift_abs)

fig=plt.figure()
plt.subplot(121)
plt.imshow(fftshift_log,cmap='gray')
plt.title('Magnitude Spectrum')

xx,yy=np.mgrid[0:fftshift_log.shape[0],0:fftshift_log.shape[1]]
ax=fig.add_subplot(122,projection='3d')
ax.plot_surface(xx,yy,fftshift_log,rstride=1,cstride=1,linewidth=0,cmap='gray')
plt.title('SurfaceView')

plt.show()
```

การแสดงผล


Magnitude Spectrum
0
50
100
150
200
250
300
350
400
0
50
100
150
200
250
300


SurfaceView
16
14
12
10
8
6
4
2
350
300
250
200
150
100
50
0
0
100
200
300
400

การอภิปราย



จากภาพข้างต้น พิจารณาที่รูปที่2 Magnitude Spectrum จะพบว่ามีเส้นสเปกตรัม 5เส้นโดยประมาณ ซึ่งเส้นแต่ละเส้นจะบอกถึงการเปลี่ยนแปลงของความเข้มของแสงและเงาภายในภาพ เมื่อย้อนกลับไปที่รูปที่1 MyFace จะพบว่า จริงๆแล้วภาพนี้มีปัญหาอยู่ เนื่องจากใบหน้าของนักศึกษามีเงามาตกกระทบระดับหนึ่งรวมทั้งมีการเปลี่ยนแปลงของความถี่ของแสงมาก ทำให้เมื่อพิจารณา Magnitude Spectrum จะบอกได้ยากว่า ส่วนไหนเป็นส่วนของใบหน้า ส่วนไหนเป็นส่วนของพื้นหลัง เนื่องจากมีจำนวนเส้นสเปกตรัมมากเกินไปนั่นเอง