

Kittipong Tapyou 65070501003

Homework 05 : Spatial Filtering

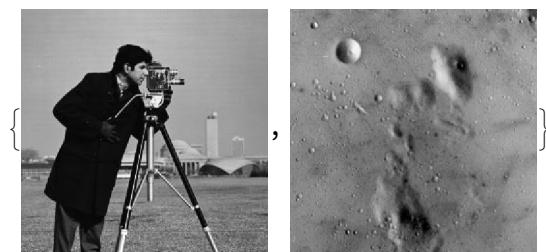
Load Image

In[1158]:=

```
imgs = {, };
```

```
grayImg = ColorConvert[#, "Grayscale"] & /@ imgs
```

Out[1159]=



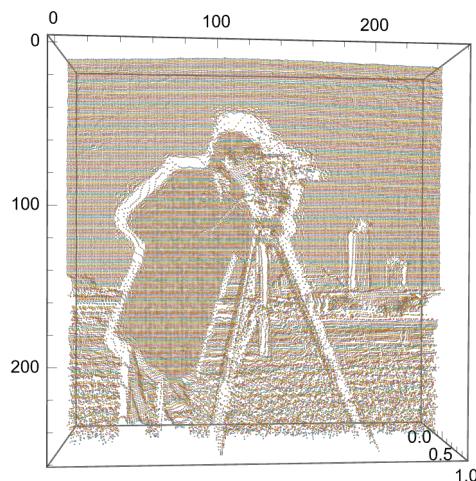
In[1022]:=

```
coorGrayImg = Table[{i, j, ImageData[grayImg[[1]]][[i, j]]}, {i, dim[[1]]}, {j, dim[[2]]}];
```

In[1023]:=

```
ListPointPlot3D[coorGrayImg]
```

Out[1023]=

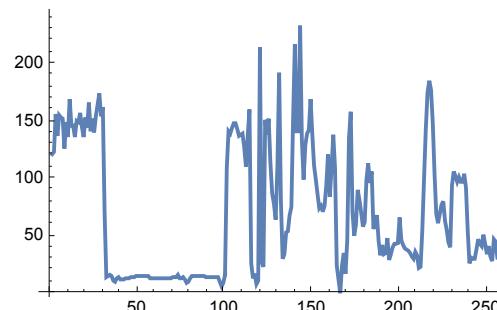


Extract image frequency

In[1038]:=

```
row = 170;
GraphicsGrid[
{{{
Show[
grayImg[1],
Graphics[{Red, Thick, Line[{{1, 256-row}, {256, 259-row}}]}]
],
ListPlot[{ImageData[grayImg[1], "Byte"][[row]], Joined→True}
]}
}]}
]
```

Out[1039]=



ทำการสกัดความถี่จากรูปโดยการอ่านข้อมูล จากแຄวตามเส้นสีแดง
จากในตัวอย่างจะเป็นข้อมูลความถี่จากแຄวที่ 170

Low pass filter : Mean filter

In[1084]:=

```
GetMeanFilter[l_] := Module[{filter},
  Return[
    ConstantArray[1/l^2, {l, l}]
  ]
]
GetMeanFilter[5] // MatrixForm
```

Out[1085]//MatrixForm=

$$\begin{pmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{pmatrix}$$

สร้าง function สำหรับการสร้าง mean filter จากในตัวอย่างจะเป็น mean filter สำหรับขนาด 5*5

In[1139]:=

```

DrawFreq[img_, idx_] := Module[{w, h},
  {w, h} = ImageDimensions[img];
  Return[{
    {
      Show[
        img,
        Graphics[{Red, Thick, Line[{{1, w-idx}, {h, h-idx}}]}]
      ],
      ListPlot[{ImageData[img, "Byte"][[idx]]}, Joined→True, PlotRange→All]
    },
    {
      Show[
        img,
        Graphics[{Red, Thick, Line[{{idx, 1}, {idx, h}}]}]
      ],
      ListPlot[{ImageData[img, "Byte"][[All, idx]], Joined→True, PlotRange→All}
    }
  ]
}
]

```

function สำหรับการแสดงความถี่ของภาพ โดยจะแสดงทั้ง row และ column โดยให้เลือกตำแหน่งไหน

In[1054]:=

```

ConvPadded[imgData_, filter_] := Module[
  {fx, fy, imgDim, ker, padTop, padBot, padLeft, padRight, paddedImg, out, rows, cols},
  {fx, fy} = Dimensions[filter];
  imgDim = Dimensions[imgData];
  ker = Reverse[filter, {1, 2}];

  padTop = Floor[(fx - 1)/2]; padBot = (fx - 1) - padTop;
  padLeft = Floor[(fy - 1)/2]; padRight = (fy - 1) - padLeft;

  paddedImg = ArrayPad[imgData, {{padTop, padBot}, {padLeft, padRight}}, "Fixed"];

  {rows, cols} = imgDim;
  out = ConstantArray[0., {rows, cols}];
  For[i = 1, i ≤ rows, i++,
    For[j = 1, j ≤ cols, j++,
      sub = paddedImg[[i ;; i + fx - 1, j ;; j + fy - 1]];
      out[[i, j]] = Total[Total[sub*ker]];
    ]
  ];
  Return[out]
]

```

function สำหรับการทำ convolution โดยที่ปกติแล้วการทำ convolution นั้นจำเป็นจะต้อง flip kernel ก่อน แต่เนื่องจาก filter ของเรานั้นสมมาตร กระบวนการนี้จึงไม่จำเป็น โดยส่วนที่ทำเพิ่มเติมคือการ padding image ด้วยค่าที่เป็นขอบของรูป เนื่องจากหากทำ conv และนั้น dimension ของรูปจะถูกลดลง

จึงคิดว่าการทำ padding น่าจะเป็นวิธีการที่ดี

In[1089]:=

```
Image[ConvPadded[ImageData[grayImg[1]], "Byte"], GetMeanFilter[7]], "Byte"]
```

Out[1089]=

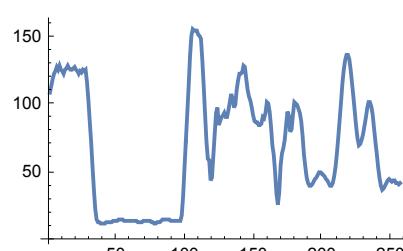
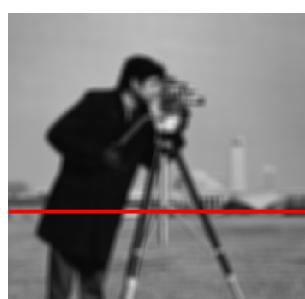
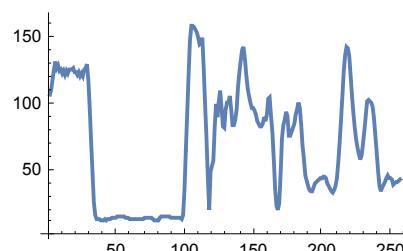
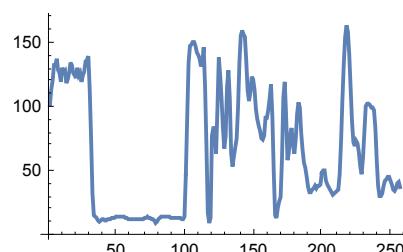
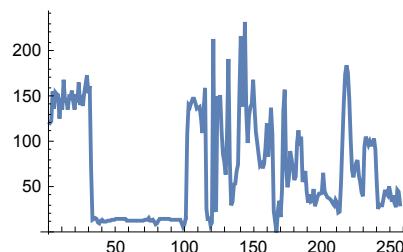


จากตัวอย่างด้านบนจะเป็นการนำ mean filter ขนาด 7×7 มา convolute กับภาพ camera man

In[1096]:=

```
GraphicsGrid[
 {
   DrawFreq[grayImg[1], 170][1],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], GetMeanFilter[3]], "Byte"],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], GetMeanFilter[5]], "Byte"],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], GetMeanFilter[7]], "Byte"]
 }
]
```

Out[1096]=



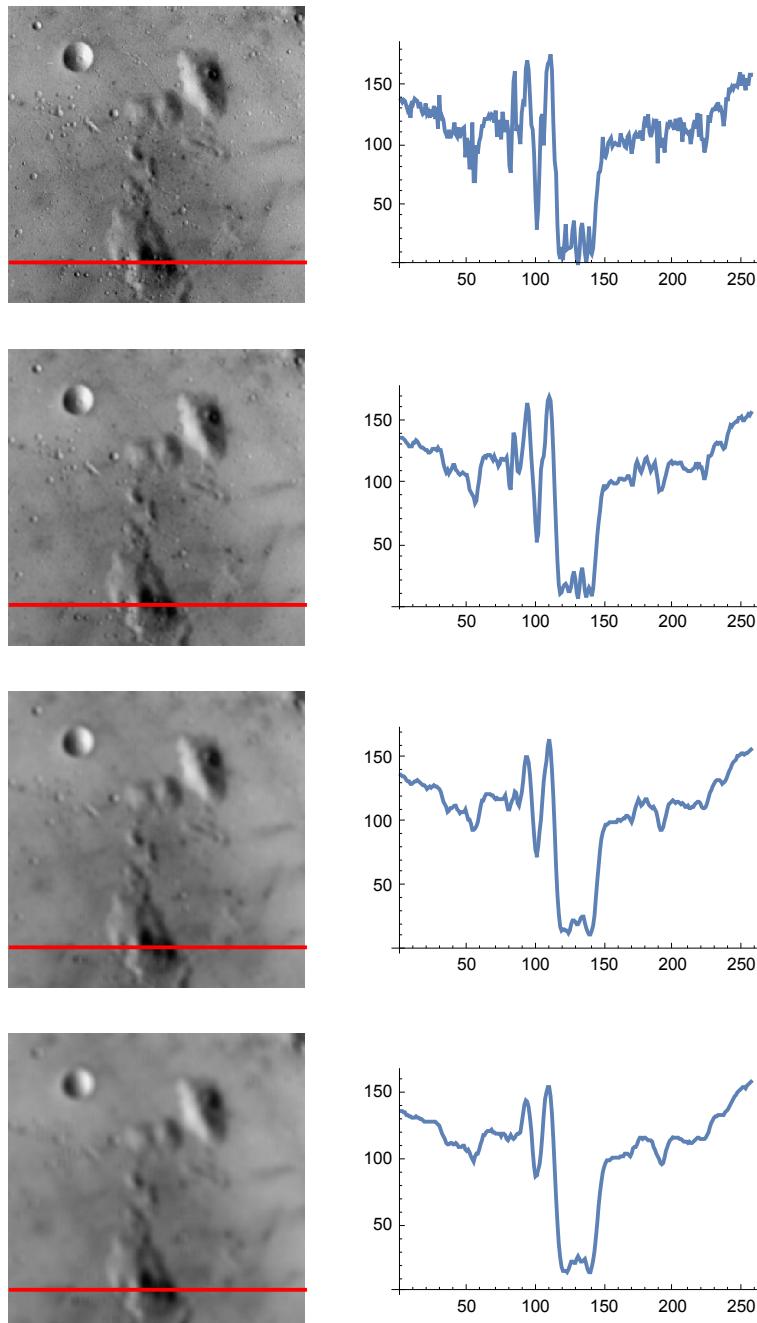
จากการทดลองด้านบน โดยที่แควรจะเป็นรูปต้นฉบับจะเห็นว่ามีความถี่ที่สูง แต่เมื่อลองทำ mean filter ที่ขนาด 3, 5, 7 ตามลำดับ รูปจะเบลอขึ้นและสัญญาณจะเรียบขึ้น หากขนาด kernel

เยื่ออะซีนรูป ก็จะเบลอมากรึเปล่า

In[1098]:=

```
GraphicsGrid[
{
  DrawFreq[grayImg[2], 220][1],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], GetMeanFilter[3]], "Byte"],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], GetMeanFilter[5]], "Byte"],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], GetMeanFilter[7]], "Byte"]
}]
```

Out[1098]=



จากรูปด้านบนจะเป็นการนำผิวดวงจันทร์มาทำการ convolute กับ mean filter จะเป็นได้ว่า

ภาพจะเบลอมากขึ้นสอดคล้องกับตัวอย่างด้านบนที่กล่าวถึงขนาดของ kernel
ว่าหากใหญ่ขึ้นก็จะเบลอมากขึ้น

Low pass filter : Gaussian filter

In[1106]:=

```
gaussianFilter = 1/273 * {{1, 4, 7, 4, 1}, {4, 16, 26, 16, 4}, {7, 26, 41, 26, 7}, {4,
```

```
gaussianFilter // MatrixForm
```

Out[1107]//MatrixForm=

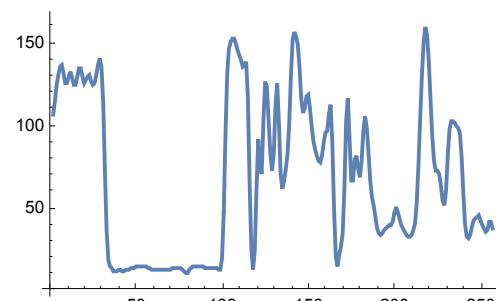
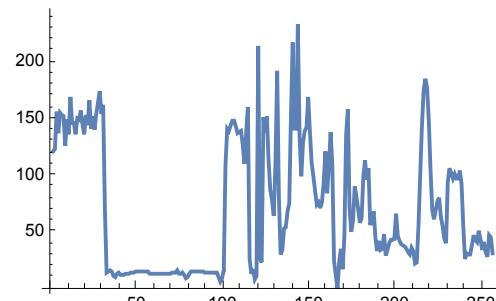
$$\begin{pmatrix} \frac{1}{273} & \frac{4}{273} & \frac{1}{273} & \frac{4}{273} & \frac{1}{273} \\ \frac{4}{273} & \frac{16}{273} & \frac{2}{273} & \frac{16}{273} & \frac{4}{273} \\ \frac{1}{273} & \frac{2}{273} & \frac{41}{273} & \frac{2}{273} & \frac{1}{273} \\ \frac{1}{273} & \frac{21}{273} & \frac{273}{273} & \frac{21}{273} & \frac{39}{273} \\ \frac{4}{273} & \frac{16}{273} & \frac{2}{273} & \frac{16}{273} & \frac{4}{273} \\ \frac{1}{273} & \frac{273}{273} & \frac{21}{273} & \frac{273}{273} & \frac{1}{273} \\ \frac{1}{273} & \frac{4}{273} & \frac{1}{273} & \frac{4}{273} & \frac{1}{273} \end{pmatrix}$$

ทำการสร้าง gaussian filter จากตัวอย่างที่ให้มา

In[1108]:=

```
GraphicsGrid[
{
  DrawFreq[grayImg[1], 170][1],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], gaussianFilter], "By"]
}]
```

Out[1108]=

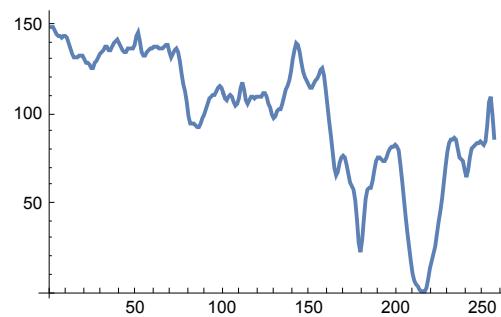
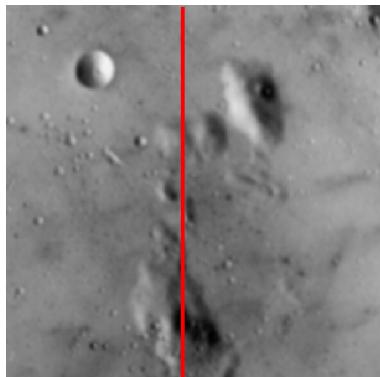
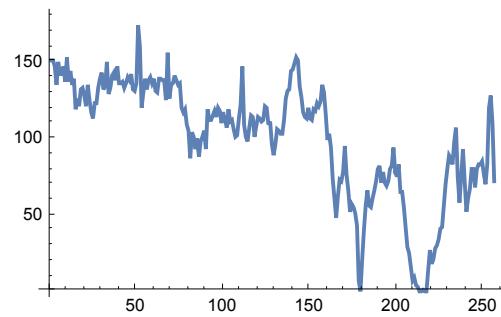
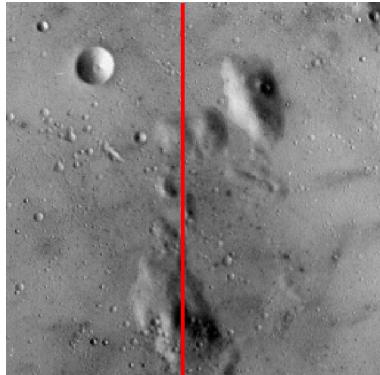


จากตัวอย่างจะเห็นได้ว่าภาพที่ได้นั้นถึงแม้จะเบลอแต่ยังคงเก็บรายละเอียดได้อยู่

In[1112]:=

```
GraphicsGrid[
 {
   DrawFreq[grayImg[2], 120][2],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], gaussianFilter], "By"]
 }
]
```

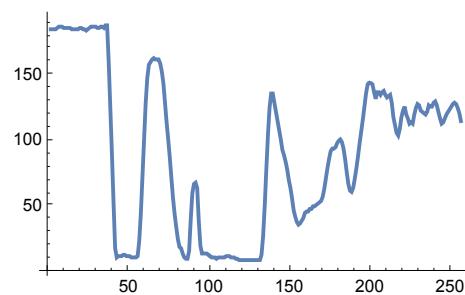
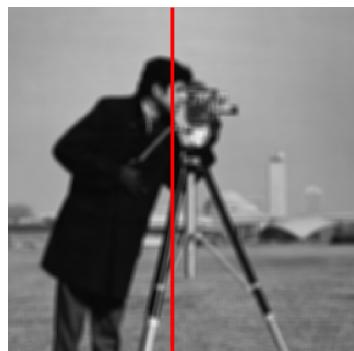
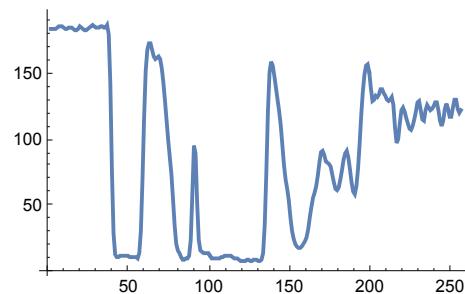
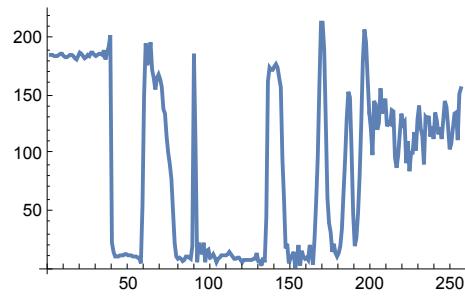
Out[1112]=



In[1114]:=

```
GraphicsGrid[
{
  DrawFreq[grayImg[1], 120][2],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], gaussianFilter], "By"],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], GetMeanFilter[5]], "By"]
}]
```

Out[1114]=



จากรูปตัวอย่างด้านบนเมื่อใช้ kernel ที่ขนาดเท่ากับ ระหว่าง mean filter และ gaussian filter
พบว่าภาพเบลอเหมือนกัน แต่รายละเอียดของ gaussian
จะเยือกกว่าเนื่องจากเป็นการเฉลี่ยแบบถ่วงน้ำหนักทำให้ยังคงสามารถคงข้อมูลของจุดนั้นๆได้ดีกว่า
mean filter จะเห็นได้ว่ารูปล่างที่ใช้ mean filter รายนละเอียดของกล้องจะหายไปค่อนข้างเยอะ

High pass filter : Laplacian filter

In[1119]:=

```
laplacianFilter = {{0, 1, 0}, {1, -4, 1}, {0, 1, 0}};
laplacianFilter // MatrixForm
```

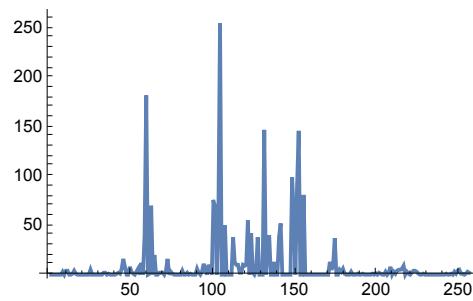
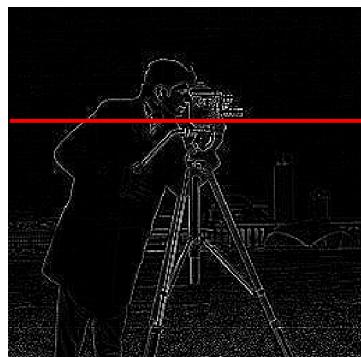
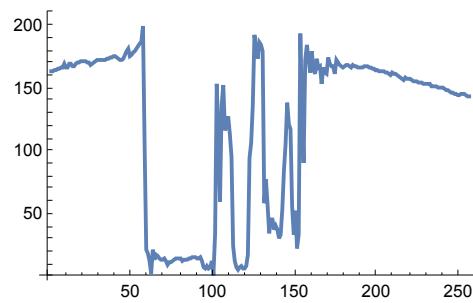
Out[1120]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

In[1149]:=

```
GraphicsGrid[
{
  DrawFreq[grayImg[1], 80][1],
  DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], laplacianFilter], "B"]
}
]
```

Out[1149]=

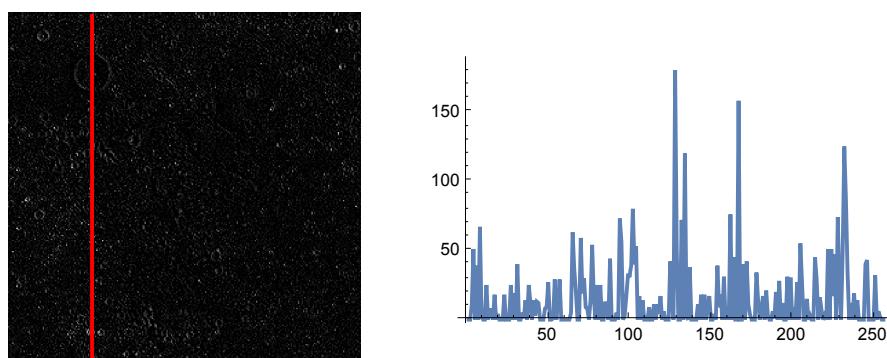
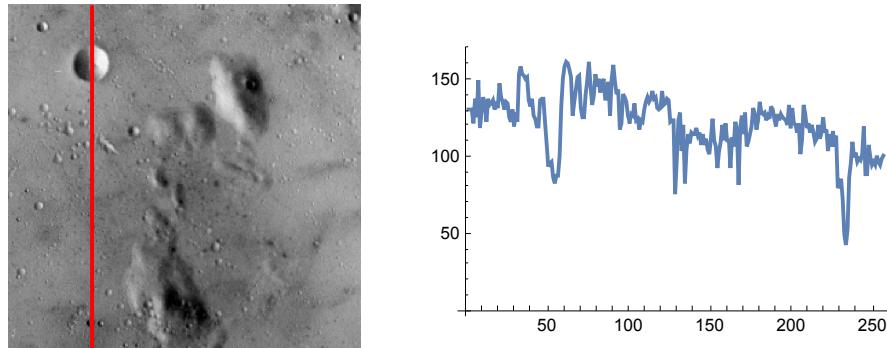


จากตัวอย่างที่เห็น เมื่อนำภาพไปทำ convolute กับ laplacian จะทำให้ได้ภาพที่เป็นขอบ (edge) ซึ่งเกิดจากส่วนที่มีการเปลี่ยนแปลงความถี่ของภาพ

In[1153]:=

```
GraphicsGrid[
 {
   DrawFreq[grayImg[2], 60][2],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], laplacianFilter], "B"]
 }
]
```

Out[1153]=



ในตัวอย่างนี้เป็นการทำ high pass ของ พื้นผิวดวงจันทร์
 ซึ่งจะเห็นได้ว่าภาพที่ได้นั้นจะมีแต่ส่วนที่เป็นความถี่สูง คือการเปลี่ยนแปลงของความเข้มสูง
 แต่จากตัวอย่างภาพของขอบนั้นไม่ค่อยชัดเท่าไร แต่จะชัดในส่วนที่เป็นหลุมเล็กๆบนดวงจันทร์
 หากมีการขยายความถี่อาจจะทำให้เห็นขอบชัดขึ้น เช่น การคูณด้วยค่าคงที่

High pass filter : Laplacian of Guassian filter

In[1154]:=

```
LOGFilter = {{0, 0, 1, 0, 0}, {0, 1, 2, 1, 0}, {1, 2, -16, 2, 1}, {0, 1, 2, 1, 0}, {0,
LOGFilter // MatrixForm
```

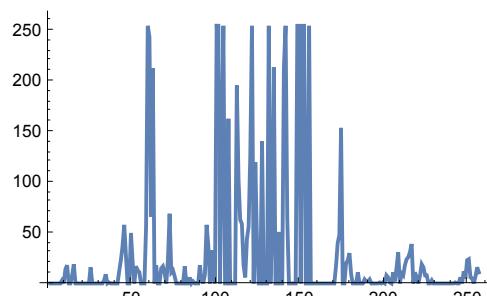
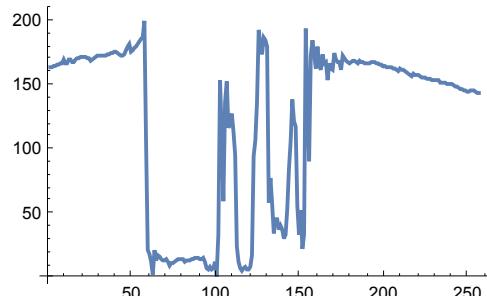
Out[1155]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

In[1156]:=

```
GraphicsGrid[
 {
   DrawFreq[grayImg[1], 80][1],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[1]], "Byte"], LOGFilter], "Byte"],
 }
]
```

Out[1156]=

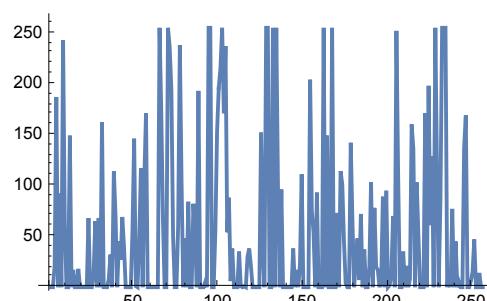
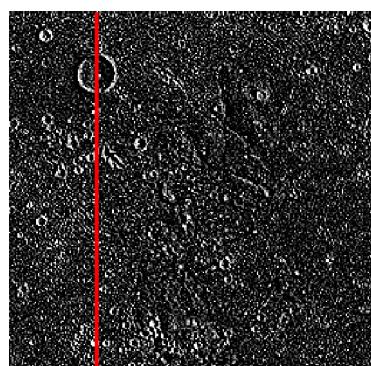
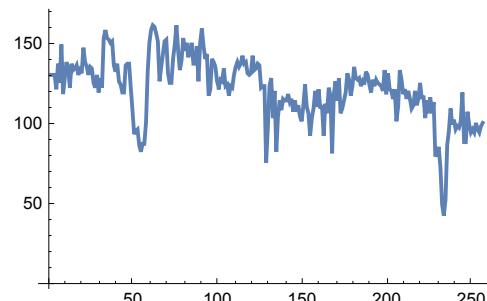
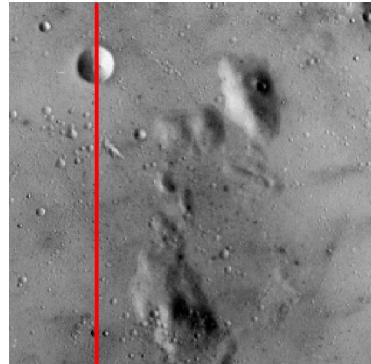


จากรูปที่ทำ LOG กับรูปนั้น โดยจะมีการทำ guassian ก่อนเพื่อเป็นการขยาย noise
จากนั้นทำการจับขอบของภาพ ทำให้ได้ edge ที่คมชัดขึ้น มากกว่าการทำ laplacian ปกติ

In[1157]:=

```
GraphicsGrid[
 {
   DrawFreq[grayImg[2], 60][2],
   DrawFreq[Image[ConvPadded[ImageData[grayImg[2]], "Byte"], LOGFilter], "Byte"],
 }
]
```

Out[1157]=



จากในรูปนี้เมื่อเทียบกับการทำ laplacian ปกตินั้น จะเห็นได้ว่าขอบของหลุมนั้น ดูยากมาก แต่หากทำ gaussian ก่อน จะทำให้ได้ขอบที่ชัดเจนมากกว่าเดิมมากๆ

Image Sharpening

In[1177]:=

```
meanBlur = Image[ConvPadded[ImageData[#, "Byte"], GetMeanFilter[5]], "Byte"] &@ grayI
gaussianBlur = Image[ConvPadded[ImageData[#, "Byte"], gaussianFilter], "Byte"] &@ gra
```

การทำ image sharpening นั้นจำเป็นต้องหา low pass ของภาพก่อน โดยจากในตัวอย่างด้านบน จะนำภาพไปผ่าน lowpass ทั้ง mean filter และ gaussian filter โดยที่ตั้งขนาดของ kernel ให้มีขนาดเท่ากันคือ 5×5 เท่ากัน

In[1190]:=

```

highpassImgMean = Table[
    Image[ImageData[grayImg[i], "Byte"] - ImageData[meanBlur[i], "Byte"],
        {i, Length[grayImg]}]
];
highpassImgGaus = Table[
    Image[ImageData[grayImg[i], "Byte"] - ImageData[gaussianBlur[i], "Byte"],
        {i, Length[grayImg]}]
];

```

นำภาพปกติมาลบกับ low pass image เพื่อให้ได้ส่วนที่เป็นขอบ แต่ต่างกับการทำ high pass เนื่องจาก การทำ high pass นั้นอาจจะทำให้ได้ขอบที่ไม่เด่นชัดมากเกินไป ถ้าขอบชัดเกินไป อาจจะทำให้ภาพกลายเป็นภาพที่มี noise มาตรฐานที่จะลดลง

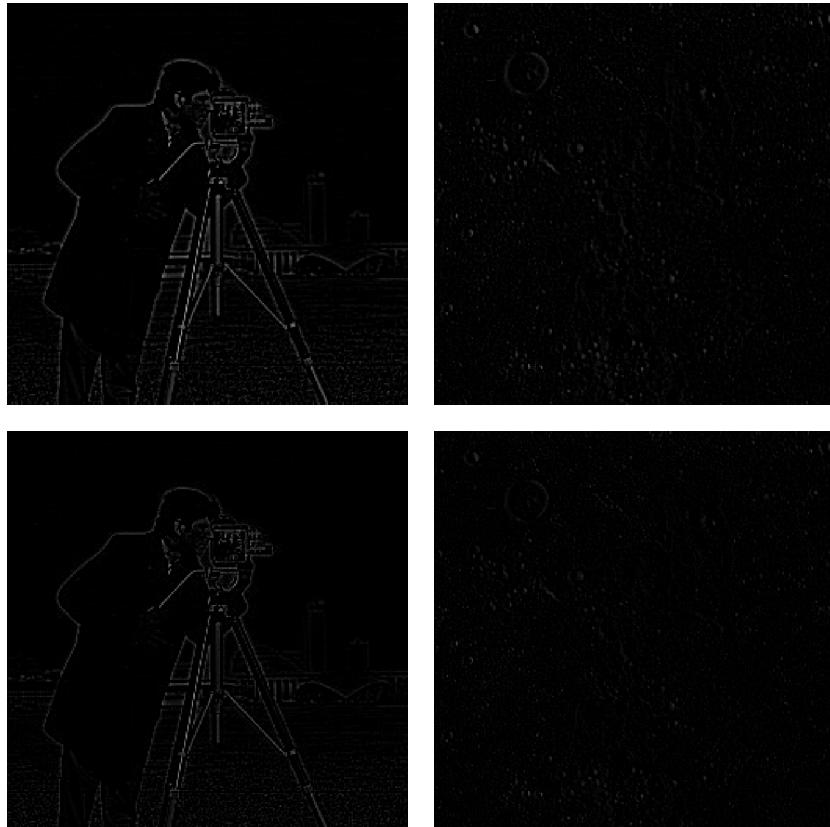
In[1192]:=

```

GraphicsGrid[
{
    highpassImgMean,
    highpassImgGaus
}
]

```

Out[1192]=



เมื่อลองดูที่ผลลัพธ์จะเห็นได้ว่าเราจะเห็นขอบของภาพนิดหน่อย แต่ในส่วนของ mean filter นั้น จะทำให้ได้ขอบที่มากกว่า การดำเนินการกับ gaussian ดูได้จากแคลบันเป็น mean filter และล่างเป็น gaussian filter โดยเฉพาะอย่างยิ่งภาพ camera man นั้นจะเห็นได้ชัดเจนว่าขอบของสองภาพนั้นต่างกัน

In[1194]:=

```

shapImgMean = Table[
    Image[ImageData[grayImg], "Byte"] + ImageData[highpassImgMean][i],
    {i, Length[grayImg]}];
shapImgGaus = Table[
    Image[ImageData[grayImg], "Byte"] + ImageData[highpassImgGaus][i],
    {i, Length[grayImg]}];

```

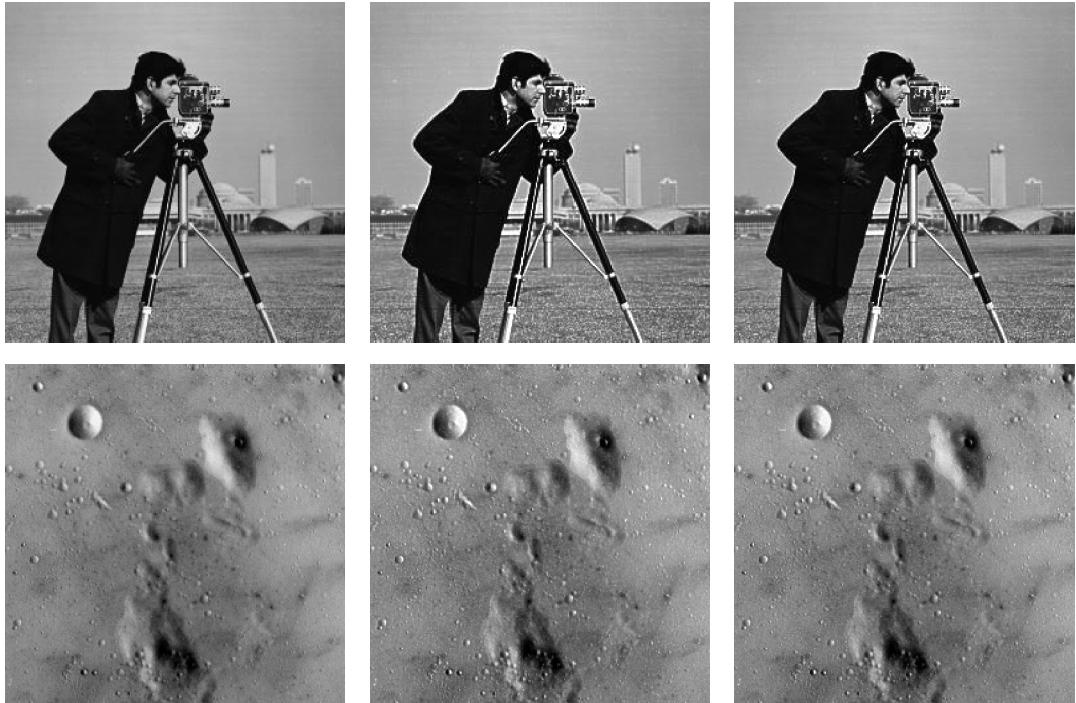
In[1200]:=

```

GraphicsGrid[
Transpose[{{
    grayImg,
    shapImgMean,
    shapImgGaus
}}]
]

```

Out[1200]=



จากการลดลงของภาพไปทำ sharpening พบว่าหากเรา加大ภาพไปทำ low pass โดยใช้ mean filter ทำให้ได้ภาพที่ดูคมชัดกว่า low pass ที่ใช้ gaussian ดูได้จากภาพดวงจันทร์จะเห็นได้ว่ามีรายละเอียดของหลุมและพื้นผิวที่เยือกกว่า (column แรกคือภาพต้นฉบับ ตามด้วย mean filter และ gaussian filter) ซึ่งอาจจะต้องพิจารณาเป็นรายกรณีไปว่าควรใช้วิธีการไหน เนื่องจากในบางงานนั้นอาจจะไม่ต้องการความคมชัดขนาดนั้น หรือหากในบางภาพอาจจะทำให้ภาพของเราเมื่อคุณภาพที่ลดลงเนื่องจากมีการ extract ข้อมูลหรือการเปลี่ยนแปลงมากเกินไป