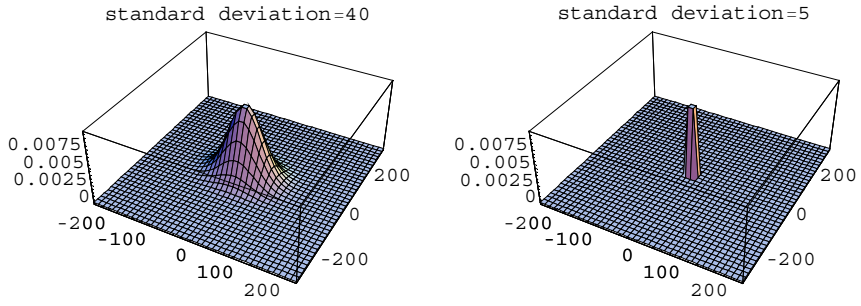
HW6, computer problem
EECS 203A, UCI, Fall 2004

by Nasser Abbasi

FIrst start by reading my *Mathematica* pakcage. This contains *Mathematica* functions I have written for basic image processing and filtering

In[1]:=
```
Clear["Global`*"];
<< nma.m
<< ImageProcessing`
```

In[4]:=
```
g2D[x_, y_, σ_] := 1/(Sqrt[2 π] σ) e^(-(x^2+y^2)/(2 σ^2))
Block[{$DisplayFunction = Identity},

 p1 = Plot3D[g2D[x, y, 40], {x, -240, 240}, {y, -320, 320},
   PlotPoints → 40, PlotRange -> All, PlotLabel -> "standard deviation=40"];

 p2 = Plot3D[g2D[x, y, 5], {x, -240, 240}, {y, -320, 320},
   PlotPoints → 40, PlotRange -> All, PlotLabel -> "standard deviation=5"];
]
Show[GraphicsArray[{p1, p2}]]
```



Out[6]= - GraphicsArray -

Now, sample a 2D continuos gaussian to obtain a discrte version of Gaussian 2D, to use as a filtering window to convolve the image with

```
In[7]:= nma`cd
        Directory[]

Out[7]= F:\nabbasi\data\nabbasi_web_Page\academic\
          my_courses\EECS_203A_UCI_FALL_2004\HW6\computer_prob

Out[8]= F:\nabbasi\data\nabbasi_web_Page\academic\
          my_courses\EECS_203A_UCI_FALL_2004\HW6\computer_prob

In[54]:= nRow = 480; nCol = 640;
         f = BinaryReadList["triangle.raw"];
         f = Partition[f, nCol];
         Print["Image has dimensions ", Dimensions[f]];
         fsaved = Chop[N[f]];
         Print["Max[f]=", Max[f]];
         Print["Min[f]=", Min[f]];

        Image has dimensions {480, 640}

        Max[f]=255

        Min[f]=148
```

Now create the 2D gaussian filter, have it centers at the center of the image

```
In[14]:= σ = 5;
         nRowG = 480; nColG = 640;
         h = Table[g2D[x, y, σ], {x, -nRowG/2, nRowG/2 - 1}, {y, -nColG/2, nColG/2 - 1}];
         hsaved = Chop[N[h]];
         Print["Min value of gaussian, std=5, is ", Min[h]];
         Print["Max value of gaussian, std=5, is ", Max[h]];

        General::spell1 : Possible spelling error: new symbol name "nRowG" is similar to existing symbol "nRow". More…

        General::spell1 : Possible spelling error: new symbol name "nColG" is similar to existing symbol "nCol". More…

        General::spell1 :
         Possible spelling error: new symbol name "hsaved" is similar to existing symbol "fsaved". More…

        Min value of gaussian, std=5, is 1/(5 e^3200 Sqrt[2 π])

        Max value of gaussian, std=5, is 1/(5 Sqrt[2 π])
```

Now generate 'g', the degraded image

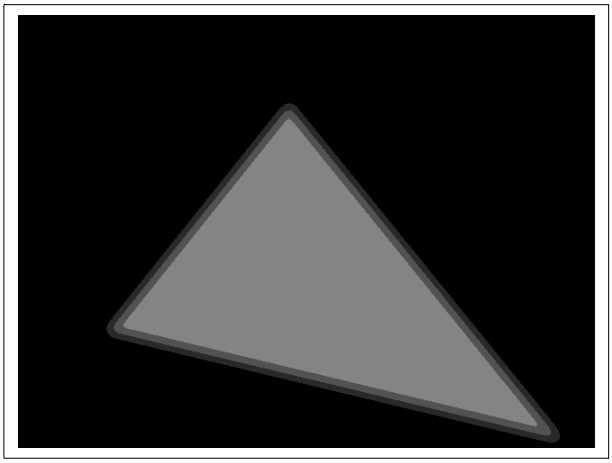
```
h = hsaved;
f = fsaved;

G = DFT[h] DFT[f];
G = nma`centerImage[G];
g = InverseDiscreteFourierTransform[G];
(*g=nma`centerImage[Re[g]];*)

(*scale the data so I can see it in raw format*)
Max[Re[g]]
Min[Re[g]]
t = Re[g];
t = (255 / Max[t]) * t;

Export["triangleDegraded.raw", Round[N[t]], "Byte"];

ArrayPlot[Round[Re[g]]]
```

Out[239]=

```
5.7662
```

Out[240]=

```
3.34666
```



Out[244]=

```
- Graphics -
```

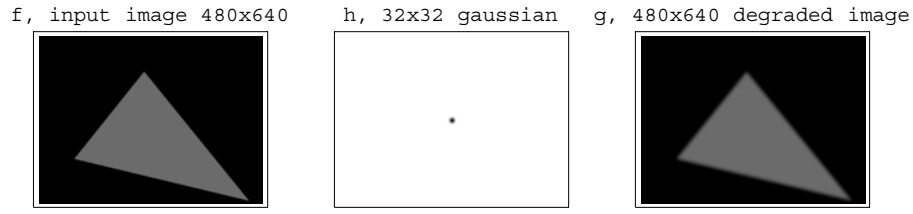
In[135]:=

```
Export["triangleDegraded.raw", t, "Byte"];

(*file=OpenWrite["triangleDegraded.raw",BinaryFormat→True];
 BinaryWrite[file,Flatten[t],"Character8"];
 Close["triangleDegraded.raw"];*)

g = nma`centerImage[Re[g]];
G = DFT[g];

Print["degraded image dimensions=", Dimensions[g]];
```

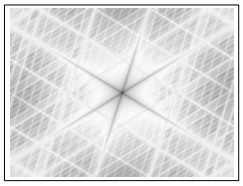
```
degraded image dimensions={480, 640}
```

Now center the images, and generate their DFT

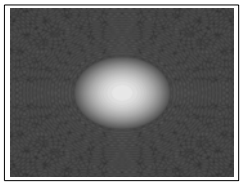
```
f = nma`centerImage[f];
h = nma`centerImage[h];

F = DFT[f];
H = DFT[h];

Block[{$DisplayFunction = Identity},
 p0 = ArrayPlot[f, PlotLabel → "f, input image 480x640"];
 p1 = ArrayPlot[h, PlotLabel → "h, 32x32 gaussian"];
 p2 = ArrayPlot[g, PlotLabel → "g, 480x640 degraded image"];

 p01 = ArrayPlot[2 Log[Abs[F]], PlotLabel → "2 Log[Abs[F]"];
 p11 = ArrayPlot[2 Log[Abs[H]], PlotLabel → "2 Log[Abs[H]"];
 p21 = ArrayPlot[2 Log[Abs[G]], PlotLabel → "2 Log[Abs[G]"];

 p02 = ArrayPlot[Arg[F], PlotLabel → "Phase F"];
 p12 = ArrayPlot[Arg[H], PlotLabel → "Phase H"];
 p22 = ArrayPlot[Arg[G], PlotLabel → "Phase G"];

]
Show[GraphicsArray[{p0, p1, p2}]];
Show[GraphicsArray[{p01, p11, p21}]];
Show[GraphicsArray[{p02, p12, p22}]];
```



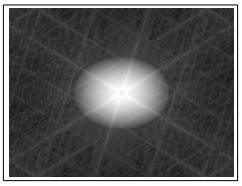
2 Log[Abs[F]



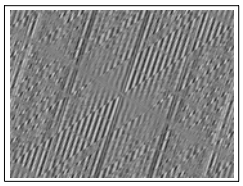
2 Log[Abs[H]



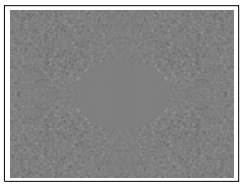
2 Log[Abs[G]



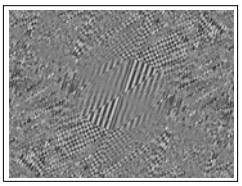
Phase F



Phase H



Phase G



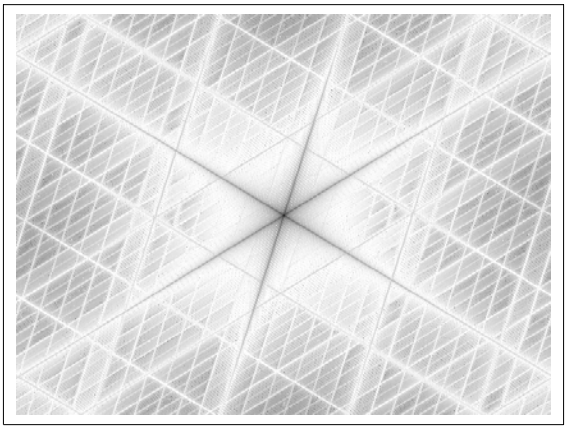
```
(*Show[GraphicsArray[
   ListPlot3D[#1,PlotRange→All,DisplayFunction→Identity]&/@{Abs[F1],Arg[F1]}]];
 Show[GraphicsArray[ListPlot3D[#1,PlotRange→All,DisplayFunction→Identity]&/@
    {Abs[H1],Arg[H1]}]];
 Show[GraphicsArray[ListPlot3D[#1,PlotRange→All,DisplayFunction→Identity]&/@
    {Abs[G1],Arg[G1]}]];*)

Print["Size of F=", Dimensions[F]];
Print["Max element in Re[F]=", Max[Re[F]], "  Min Element in Re[F]=", Min[Re[F]]];
Print["Max element in Im[F]=", Max[Im[F]], "  Min Element in Im[F]=", Min[Im[F]]];
Print["Max element in Abs[F]=",
  Max[Abs[F]], "  Min Element in Abs[F]=", Min[Abs[F]]];

Print["Size of H=", Dimensions[H]];
Print["Max element in Re[H]=", Max[Re[H]], "  Min Element in Re[H]=", Min[Re[H]]];
Print["Max element in Im[H]=", Max[Im[H]], "  Min Element in Im[H]=", Min[Im[H]]];
Print["Max element in Abs[H]=",
  Max[Abs[H]], "  Min Element in Abs[H]=", Min[Abs[H]]];

Print["Size of G=", Dimensions[G]];
Print["Max element in Re[G]=", Max[Re[G]], "  Min Element in Re[G]=", Min[Re[G]]];
Print["Max element in Im[G]=", Max[Im[G]], "  Min Element in Im[G]=", Min[Im[G]]];
Print["Max element in Abs[G]=",
  Max[Abs[G]], "  Min Element in Abs[G]=", Min[Abs[G]]];
```

```
Size of F={480, 640}

Max element in Re[F]=126959.  Min Element in Re[F]=-6050.16

Max element in Im[F]=6806.46  Min Element in Im[F]=-6806.46

Max element in Abs[F]=126959.  Min Element in Abs[F]=0.000544455

Size of H={480, 640}

Max element in Re[H]=0.0226125  Min Element in Re[H]=-0.0225853

Max element in Im[H]=2.73996×10^-18  Min Element in Im[H]=-2.80906×10^-18

Max element in Abs[H]=0.0226125  Min Element in Abs[H]=1.41499×10^-17

Size of G={480, 640}

Max element in Re[G]=2870.87  Min Element in Re[G]=-136.352

Max element in Im[G]=153.582  Min Element in Im[G]=-153.582

Max element in Abs[G]=2870.87  Min Element in Abs[G]=3.69973×10^-17
```

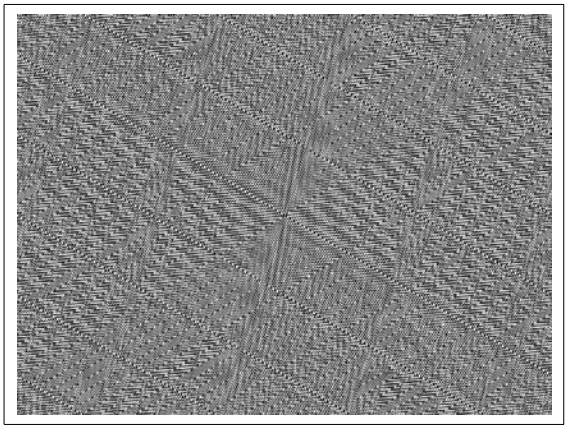
Now generate restored image

```
{nRow, nCol} = Dimensions[G];
{hRow, hCol} = Dimensions[H];

(*divide center of G/H *)
GG = G / H;
Print["Result of G/H has dimensions = ", Dimensions[GG]]


ArrayPlot[10 Log[Abs[GG]], PlotLabel -> "Log[Abs[RestoredImage]]"];
ArrayPlot[Arg[GG], PlotLabel -> "Phase RestoredImage"];
m = nma`centerImage[GG];
```

Log[Abs[RestoredImage]]



Phase RestoredImage



```
ff = InverseDiscreteFourierTransform[m];

ff = nma`centerImage[Re[ff]];

Close["triangleRestore.raw"];
file = OpenWrite["triangleRestore.raw", BinaryFormat → True];
BinaryWrite[file, Floor[ff], "Character8"];
Close["triangleRestore.raw"];

ArrayPlot[ff, PlotLabel -> "restored Image"];

Block[{$DisplayFunction = Identity},
 p0 = ArrayPlot[f, PlotLabel → "f, input image 480x640"];
 p1 = ArrayPlot[g, PlotLabel → "g, degraded image"];
 p2 = ArrayPlot[ff, PlotLabel → "ff, restored  image"];

]
Show[GraphicsArray[{p0, p1, p2}]];
```