HW3, Computer problem, part (a)
EECS 203A. UCI. FALL 2004

by Nasser Abbasi

This below shows the solution to this problem.
First initliaize session and load package needed to read/write binary files.

```
In[14]:= Remove["Global`*"];
         Install["binary"];
         << Graphics`Graphics`;
         << DiscreteMath`Combinatorica`;
```

```
In[18]:=
         SetDirectory[ToFileName[Extract[
             "FileName" /. NotebookInformation[EvaluationNotebook[]], {1}, FrontEnd`FileName]]];
```

Define the filter response function. This takes the filter itself and the piece of data below the filter and returns the new pixel value

```
In[24]:=
         filterResponse[filter_, data_] := Module[{n, m, r, c, i, j, v},

            {n, m} = Dimensions[filter];
            {r, c} = Dimensions[data];

            v = 0;
            For[i = 1, i <= r, i = i + 1,
             {
              For[j = 1, j <= c, j = j + 1,
               {
                v = v + filter[[i, j]] data[[i, j]];
               }
              ]
             }
            ];
            (*Print["returning from filter response, v=",v];*)
            Return[v]
           ];
```

Read the image file and display it.

```
In[27]:= process[fileName_, outFileName_] := Module[{},
           nRow = 480;
           nCol = 640;
           maskWidth = 5;  (*mask width/height*)
           data = FastBinaryFiles`ReadListBinary[fileName, Byte];
           nPixels = Length[data];
           Print["Dimensions of data read is =", Dimensions[data]];
           Print["Filter used is =", filter = Table[1, {maskWidth}, {maskWidth}]];

           data = Reverse[Partition[data, nCol]];
```

```
hOriginalData = Histogram[Flatten[data],
  PlotLabel → {"Histogram of ", fileName, " BEFORE. mean=" , N[Mean[Flatten[data]]],
    "\nmedian=", N[Median[Flatten[data]]]}, DisplayFunction → Identity];

Print["Dimensions of data after partition is =", Dimensions[data]];

(*Display the image before processing *)
ListDensityPlot[data, Mesh → False, Frame → False, ImageSize → {nRow, nCol},
 PlotRange → All, AspectRatio → Automatic, PlotLabel → {fileName,
   " Before. Number of pixels =" , nPixels, "\nResolution=", nRow, " by ", nCol}];

ϵ = Floor[maskWidth / 2];
firstCol = ϵ + 1;
firstRow = ϵ + 1;
lastRow = nRow - ϵ ;
lastCol = nCol - ϵ ;

newImage = Table[0, {nRow - 2 ϵ}, {nCol - 2 ϵ}];

ii = 0;
For[i = firstRow, i <= lastRow, i = i + 1,
 {
  ii = ii + 1;
  jj = 0;
  For[j = firstCol, j <= lastCol, j = j + 1,
   {
    jj = jj + 1;
    d = Take[data, {i - ϵ, i + ϵ}, {j - ϵ, j + ϵ}];
    (*Print[d];*)
    (*newImage[[ii,jj]] =filterResponse[filter ,d];*)
    newImage[[ii, jj]] = Mean[Flatten[d]];
    (*Print[newImage[[ii,jj]];*)
   }
  ]
 }
];

(*Display the image after processing *)
(* normalize the new image gray level
  by dividing by number of coefficients in the filter *)
(*newImage=Round[Chop[N[newImage/(maskWidth^2 )]]];*)

Print["New image dimensions=", {r, c} = Dimensions[newImage]];

ListDensityPlot[newImage, Mesh → False, Frame → False,
 ImageSize → {r, c}, PlotRange → All, AspectRatio → Automatic,
 PlotLabel → {fileName , " After. Number of pixels =",
   Length[Flatten[newImage]], "\nResolution=", r, "by", c}];

(* Now write the new image to file *)
Print["Writing new image to file ", outFileName];
```

```
  strm = FastBinaryFiles`OpenWriteBinary[outFileName];
  FastBinaryFiles`WriteBinary[strm, Round[Chop[N[newImage]]], Byte];
  Close[strm];

  hNewImage = Histogram[Flatten[newImage], PlotLabel →
     {"Histogram of", fileName, " After. mean=", N[Mean[Flatten[newImage]]],
      "\nmedian=", N[Median[Flatten[newImage]]]}, DisplayFunction → Identity];

  Print["display histogram of original image"];
  Show[hOriginalData, PlotRange → {{0, 255}, All},
   Frame → True, DisplayFunction → $DisplayFunction];

  Show[hNewImage, PlotRange → {{0, 255}, All},
   Frame → True, DisplayFunction → $DisplayFunction];


 ]
```

In[28]:=

```
process["cat.raw", "cat_linear_averaged.raw"];
process["triangle.raw", "triangle_linear_averaged.raw"];
```

Dimensions of data read is ={307211}

Filter used is $=\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

Dimensions of data after partition is ={480, 640}

```
{cat.raw,  Before. Number of pixels =, 307211,
Resolution=, 480,  by , 640}
```

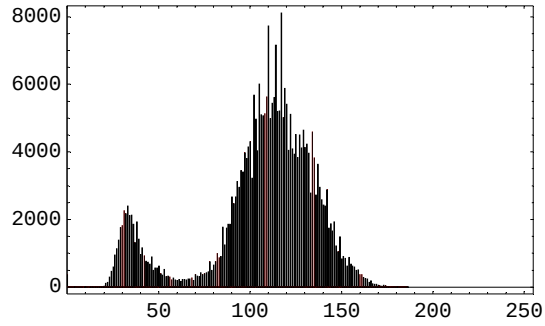
New image dimensions={476, 636}

```
{cat.raw,  After. Number of pixels =, 302736,
Resolution=, 476, by, 636}
```

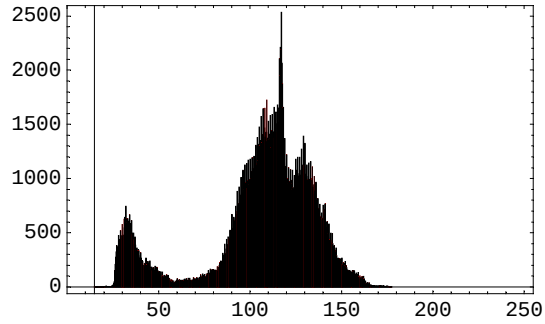
Writing new image to file cat_linear_averaged.raw

display histogram of original image

{Histogram of , cat.raw, BEFORE. mean=, 105.799, median=, 112.}



{Histogram of, cat.raw, After. mean=, 106.071, median=, 112.08}

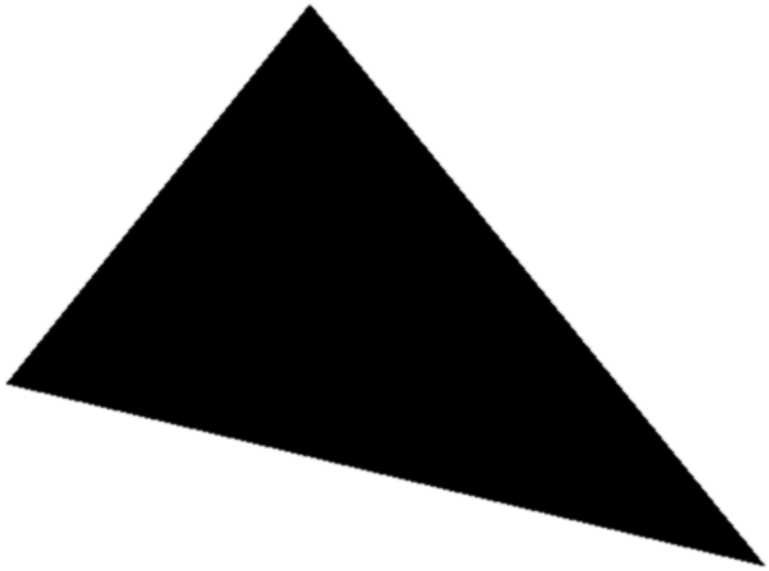


Dimensions of data read is ={307200}

Filter used is $= \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$

Dimensions of data after partition is ={480, 640}

```
{triangle.raw,  Before. Number of pixels =, 307200,
Resolution=, 480,  by , 640}
```



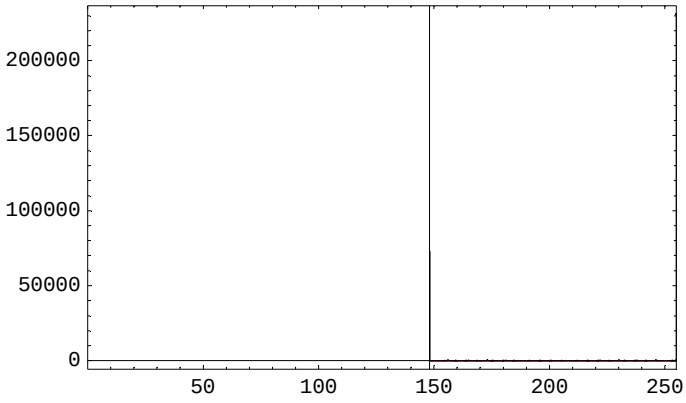
New image dimensions={476, 636}

```
{triangle.raw,  After. Number of pixels =, 302736,
Resolution=, 476, by, 636}
```

Writing new image to file triangle_linear_averaged.raw

display histogram of original image

{Histogram of , triangle.raw, BEFORE. mean=, 229.063, median=, 255.}



{Histogram of, triangle.raw, After. mean=, 228.68, median=, 255.}