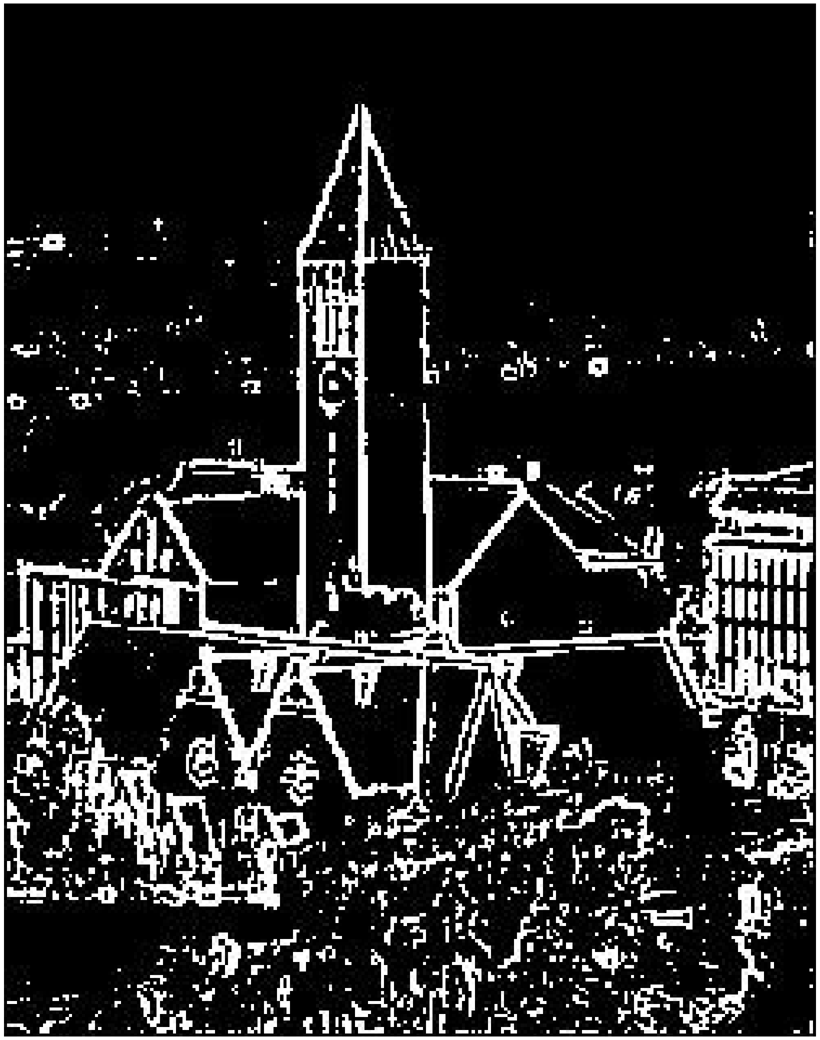


Filtering Images

Lecture 16 (Mar 25)
CS100M - Spring 2008

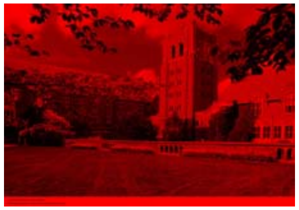
Announcements

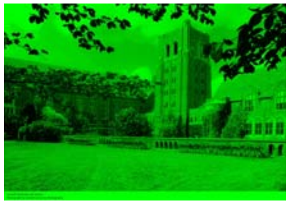
- Section is in the classroom this week
- Questions on Project 4?
  - Use simple arithmetic instead of Matlab functions to get the base-4 digits that you need

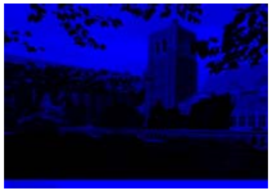
Recall

- An image in Matlab is just an array
  - A 2D array of uint8 values for a gray-scale image
  - A 3D array consisting of 3 layers (red, green, blue) for a color image
    - Each layer is a 2D array of uint8 values

Cornell University Law School
Photograph by Cornell University Photography





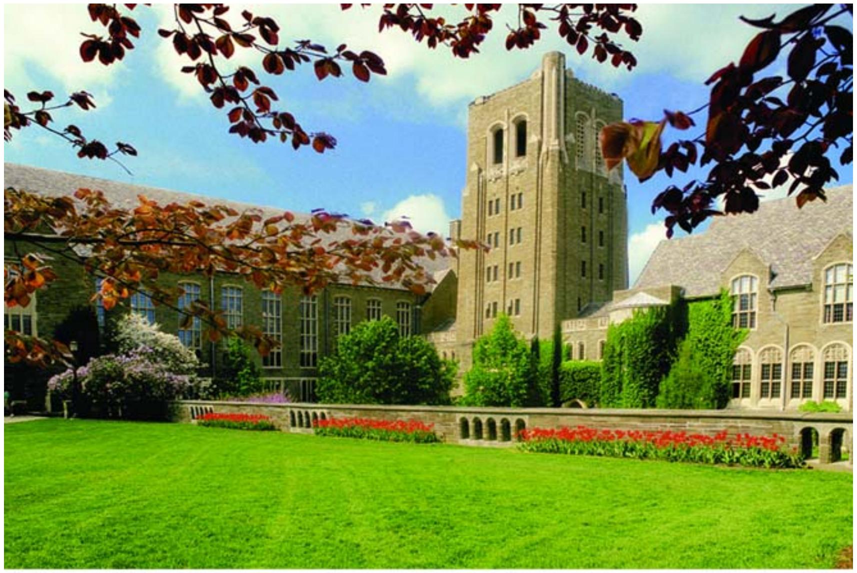


- Images in a file are usually compressed
  - Matlab uses imread and imwrite

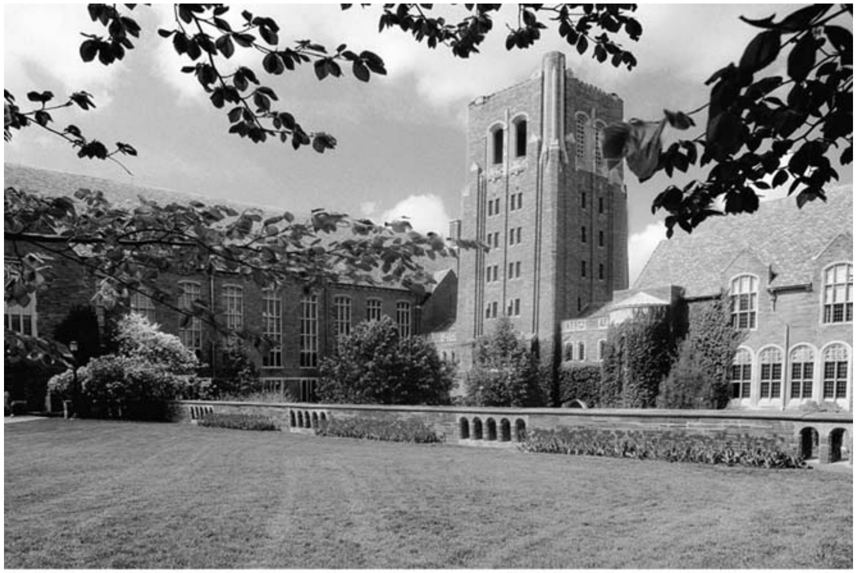
- Matlab uses imshow or image to display an image

rgb2gray

```
A = imread('LawSchool.jpg');
bwA = rgb2gray(A);
imwrite(bwA,'LawSchoolBW.jpg')
```



Cornell University Law School
Photograph by Cornell University Photography



Cornell University Law School
Photograph by Cornell University Photography

Why not take Average?

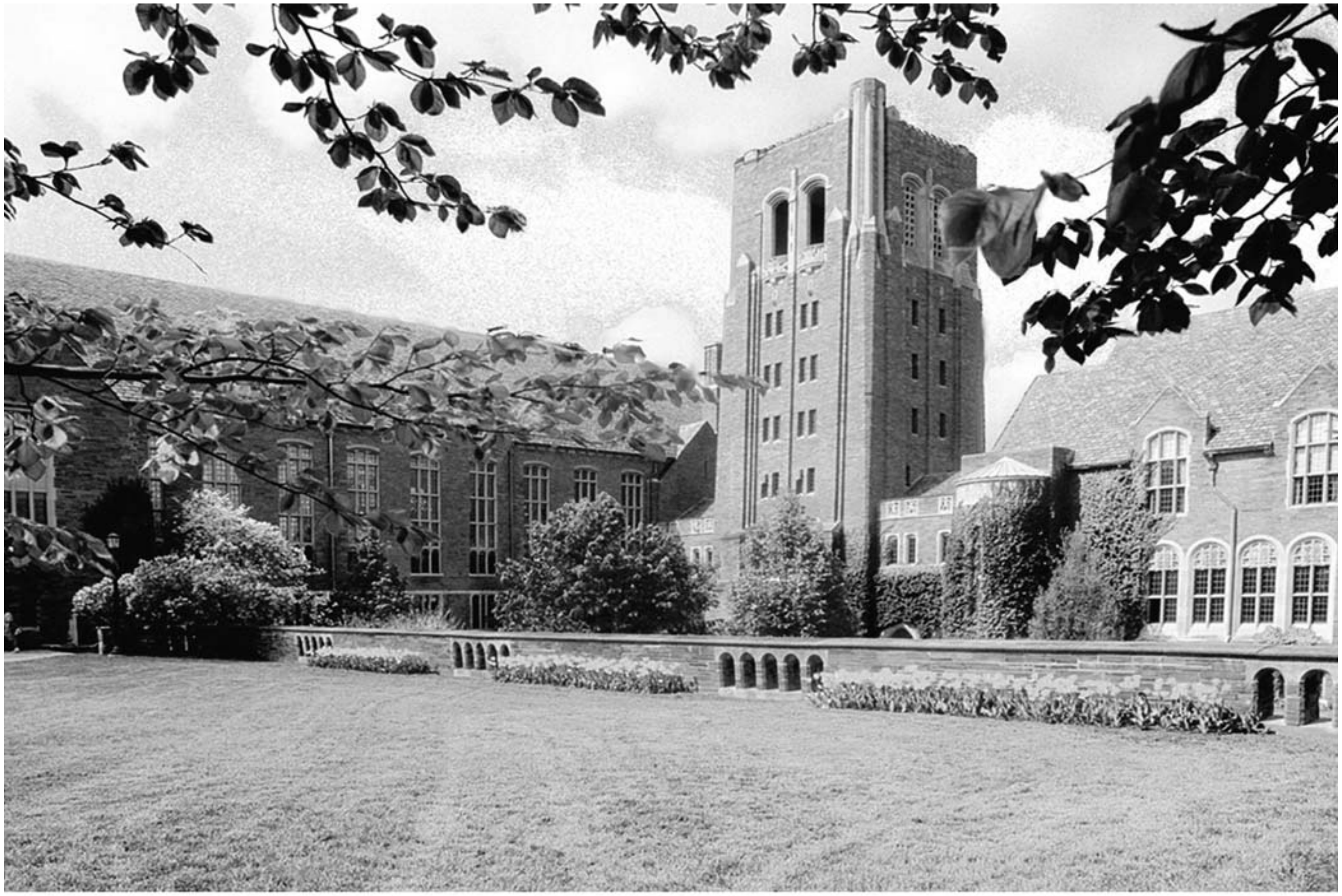
```
bwA = uint8(zeros(m,n));
for i=1:m
  for j = 1:n
     bwA(i,j) = ( A(i,j,1) + A(i,j,2) + A(i,j,3) )/3;
  end
end
imwrite(bwA,‘LawSchoolBW.jpg')
```

Cornell University Law School
Photograph by Cornell University Photography

Why not take Max?

```
bwA = uint8(zeros(m,n));
for i=1:m
  for j = 1:n
     bwA(i,j) = max([A(i,j,1)  A(i,j,2)  A(i,j,3)]);
  end
end
imwrite(bwA,‘LawSchoolBW.jpg')
```

Max:

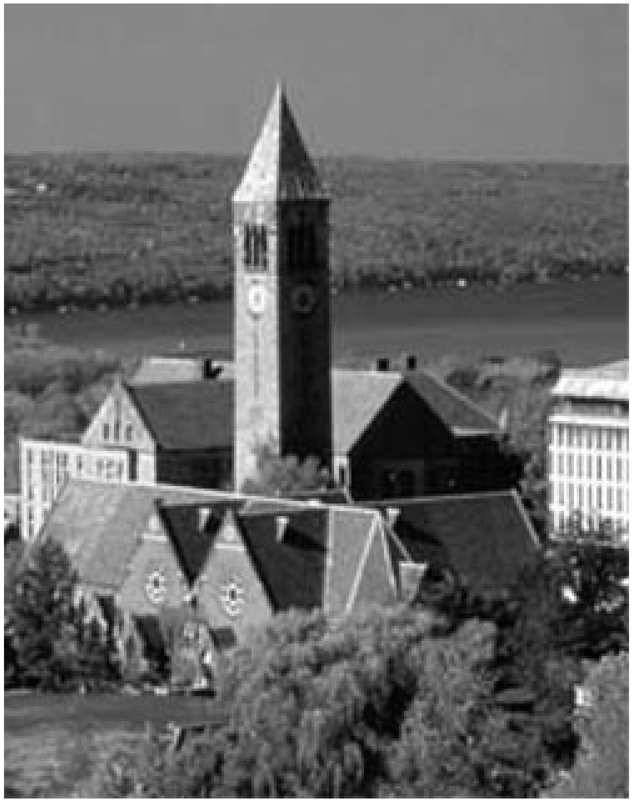


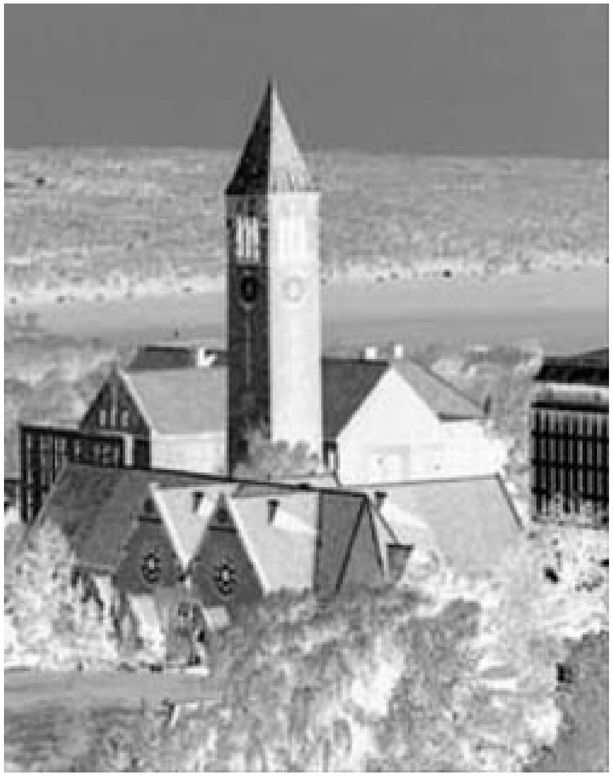
Cornell University Law School
Photograph by Cornell University Photography

Matlab:

Cornell University Law School
Photograph by Cornell University Photography

Problem: Produce a Negative





Idea

If matrix A represents the image and

$$B(i,j) = 255 - A(i,j)$$

for all i and j, then B will represent the negative

uint8 values

- uint8

  = unsigned 8-bit integer

  - $2^8 = 256$
    - Values are between 0 and 255 (inclusive)

- Arithmetic with uint8 produces uint8 results
  - Results that are too big are replaced with 255
  - Results that are negative are replaced with 0

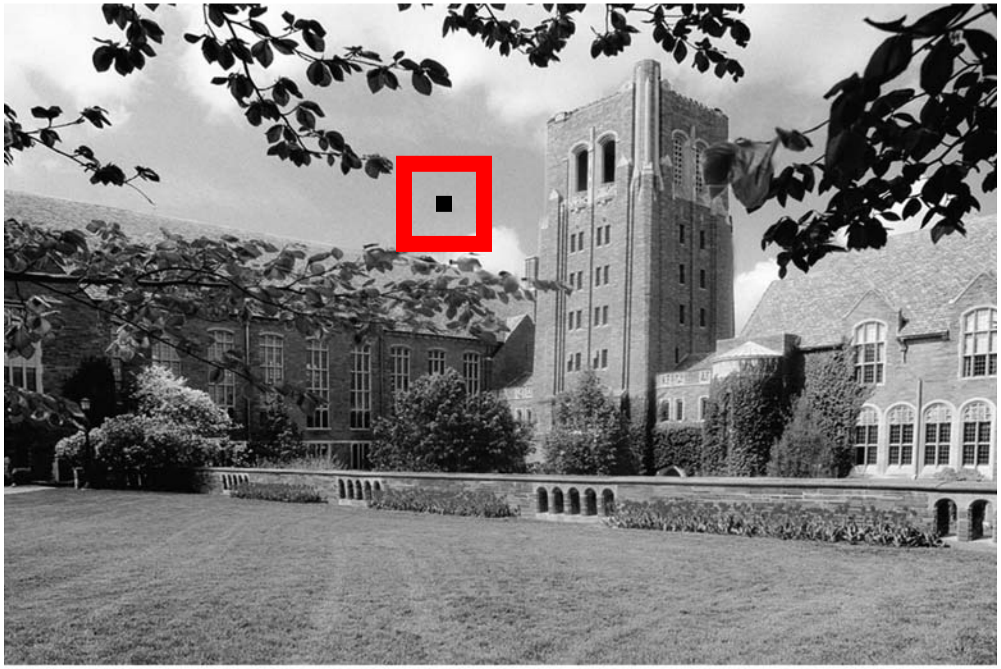
- The Matlab *Workspace* shows the type for each of your variables

  - imread creates an array of type uint8

  - imwrite converts numbers to uint8 before writing

Dirt!

1458-by-2084

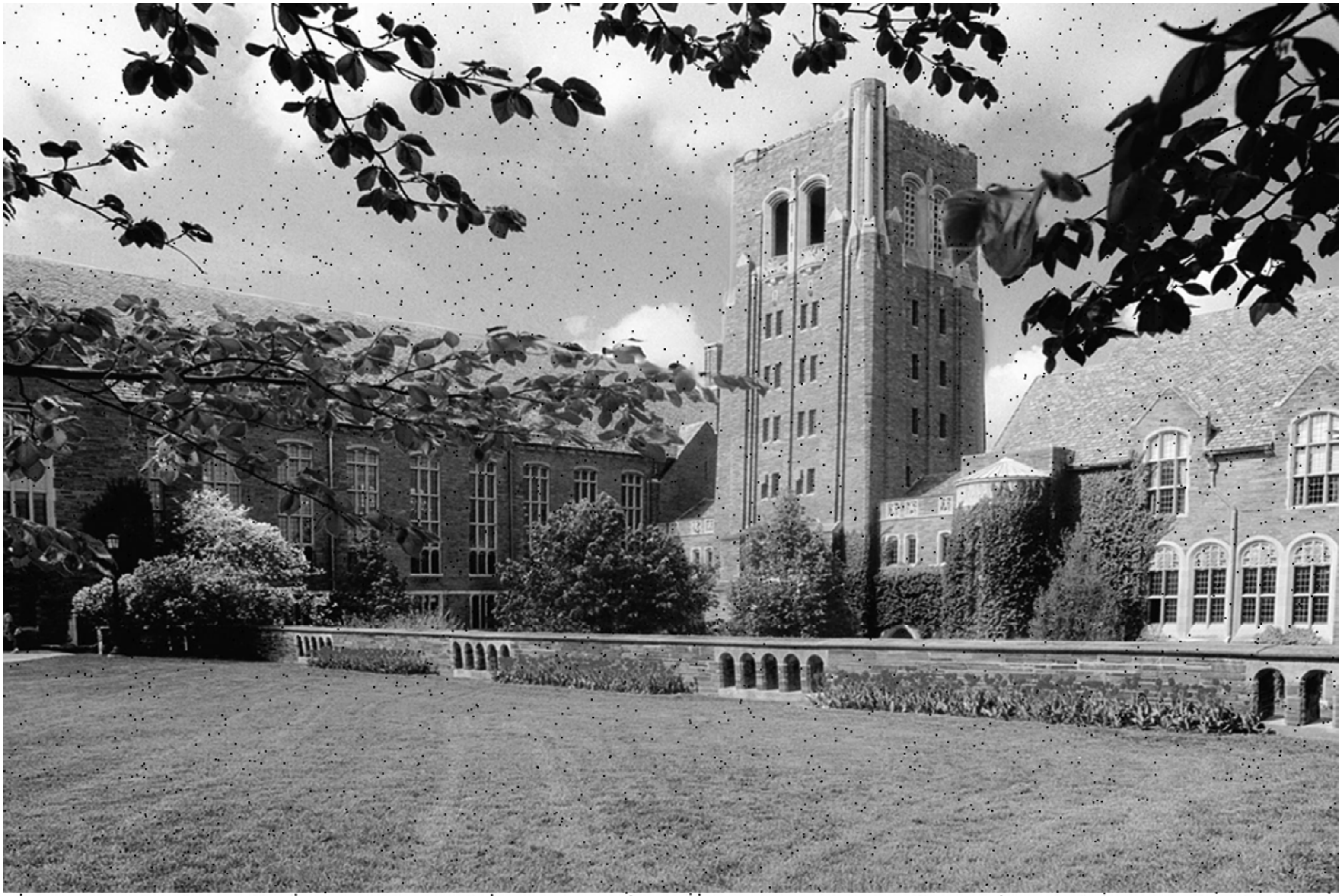


Cornell University Law School
Photograph by Cornell University Photography

150	149	152	153	152	155
151	150	153	154	153	156
153	2	3	1	155	158
154	2	1	2	156	159
156	1	1	3	158	161
157	156	159	160	159	162

The "dirty pixels" look out of place

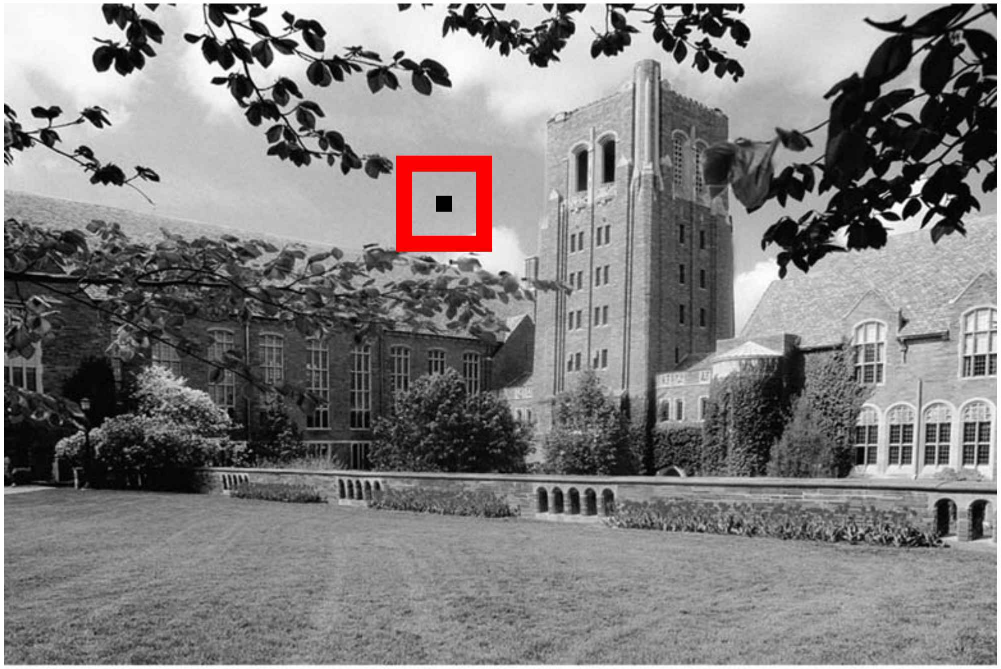
Can We Filter Out the "Noise"?



Cornell University Law School
Photograph by Cornell University Photography

Idea

1458-by-2084



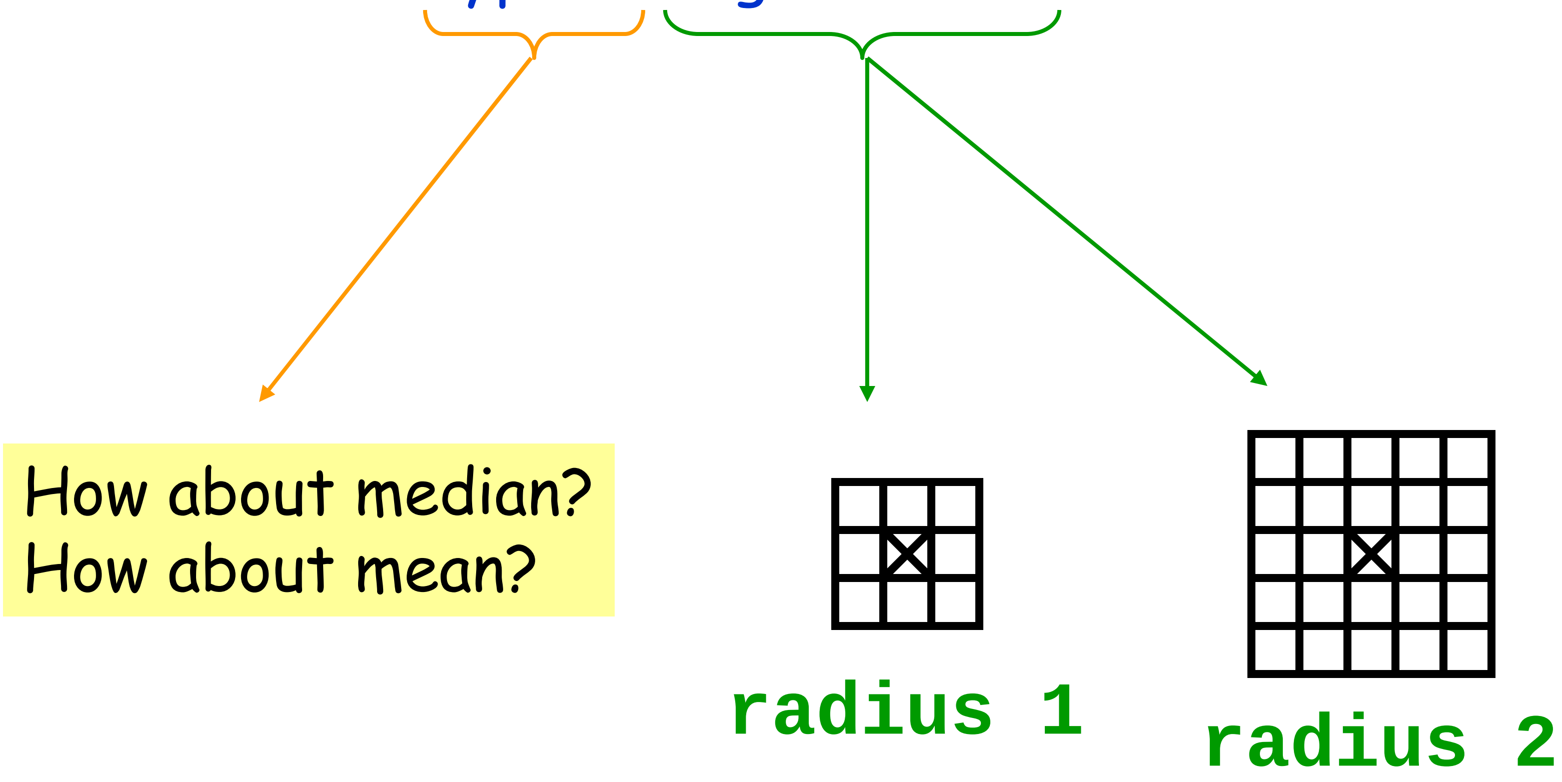
Cornell University Law School
Photograph by Cornell University Photography

150	149	152	153	152	155
151	150	153	154	153	156
153	?	?	?	155	158
154	?	?	?	156	159
156	?	?	?	158	161
157	156	159	160	159	162

Assign "typical" neighborhood value to each dirty pixels

Getting Precise

Typical neighborhood value



How about median?
How about mean?

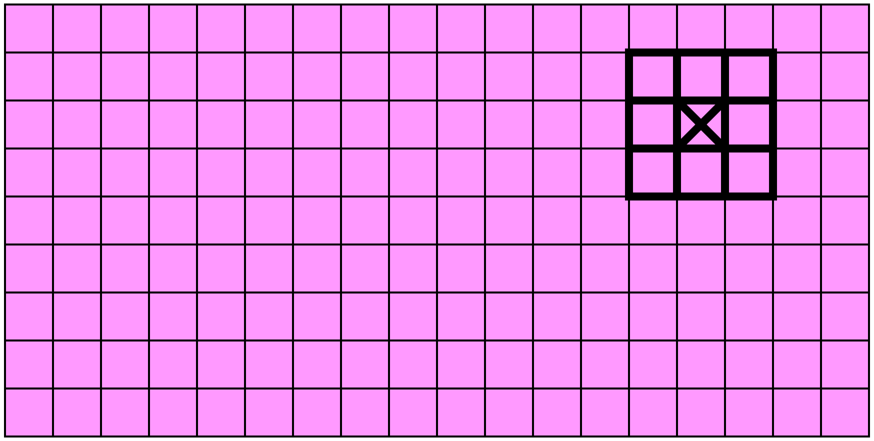
Median Filtering

Visit each pixel

Replace its value by the median of the values in its neighborhood

Using a radius 1 "Neighborhood"

0
6
6
6
6
7
7
7
7



7	7	6
7	0	6
7	6	6

Before

7	7	6
7	6	6
7	6	6

After

What We Need…

1. A function that computes the median value in a 2-dimensional array C:

   m = medVal(C)

2. A function that builds the filtered image using median values of radius r neighborhoods:

   B = medFilter(A,r)

Computing Medians

x :	21	89	36	28	19	88	43

```
x = sort(x);
```

x :	19	21	28	36	43	88	89

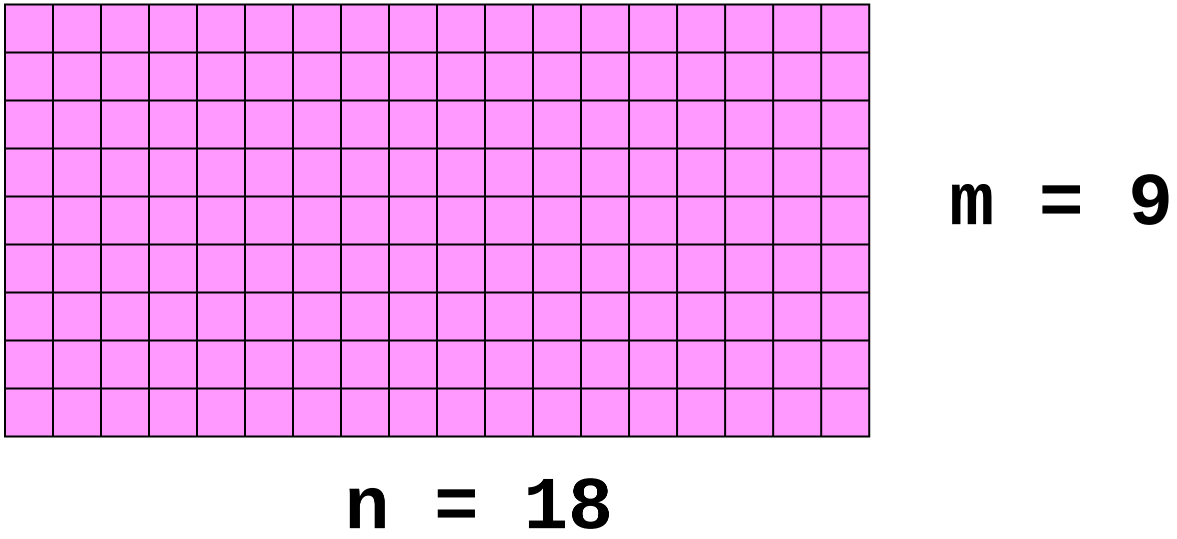
```
n = length(x);   % n = 7
m = ceil(n/2);   % m = 4
med = x(m);      % med = 36
```

If n is even, then use : `med = ( x(m) + x(m+1) )/2`

Median of a 2D Array

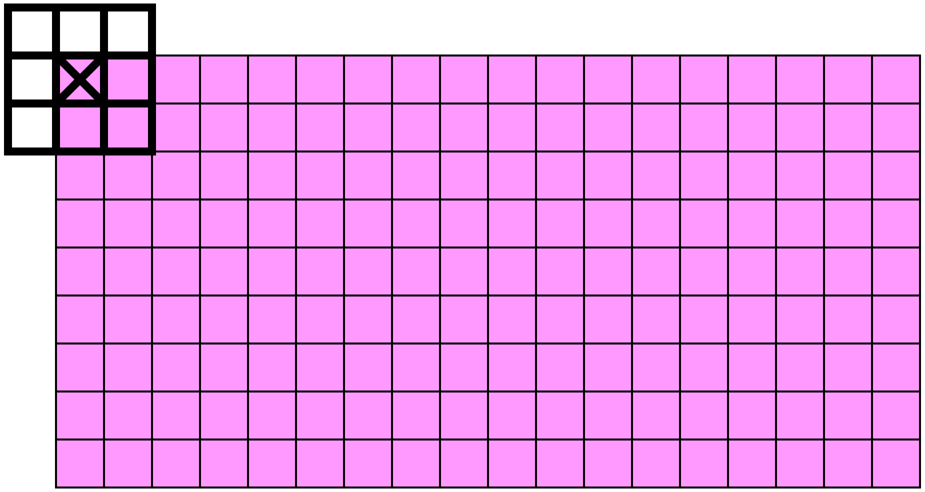
```
function med = medVal(C)
% Return the median value in the 2D array C.

% Assemble C's entries into a 1-dim array and sort
[p,q] = size(C);
n = p*q;
v = C(1:n);     % Can access 2D-array with 1D subscripts
v = sort(v);

% Compute median of v and assign to med
```

How to Visit Every Pixel



```
for i=1:m
   for j=1:n
        Compute new gray value for pixel (i,j).
   end
end
```

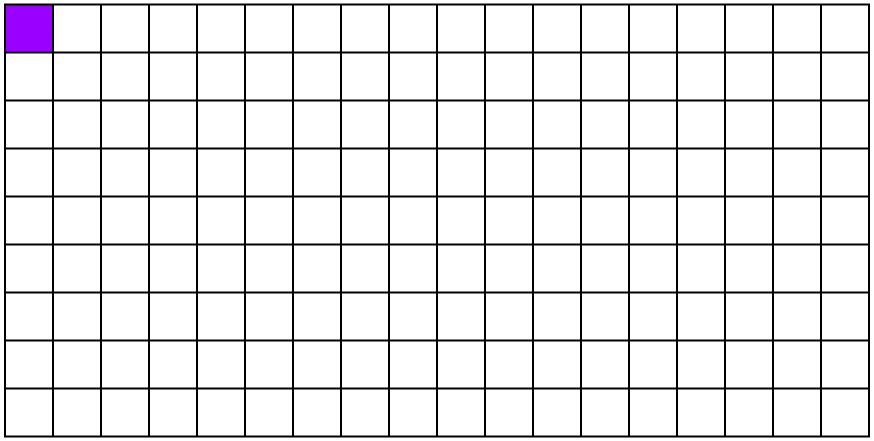
Original:



```
i = 1
```

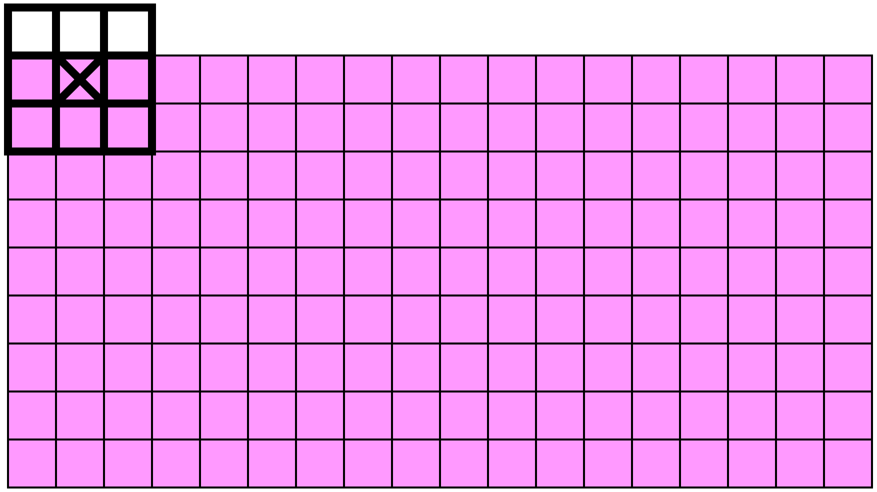
```
j = 1
```

Filtered:



Replace ☒ with the median of the values under the window.

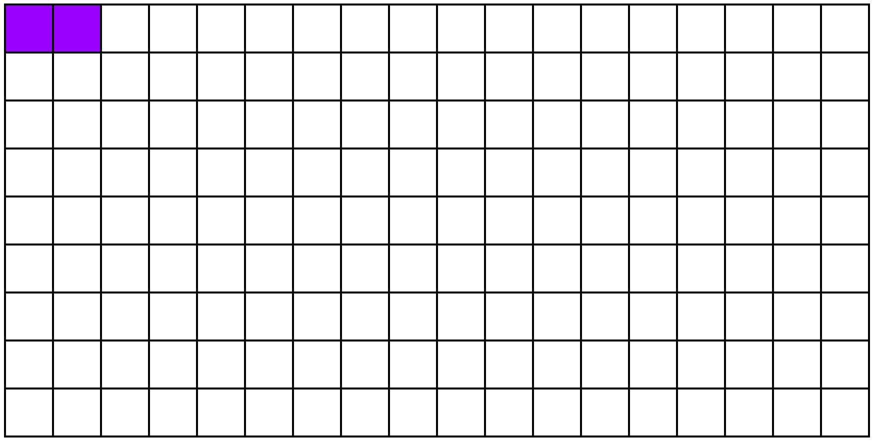
Original:



i = 1

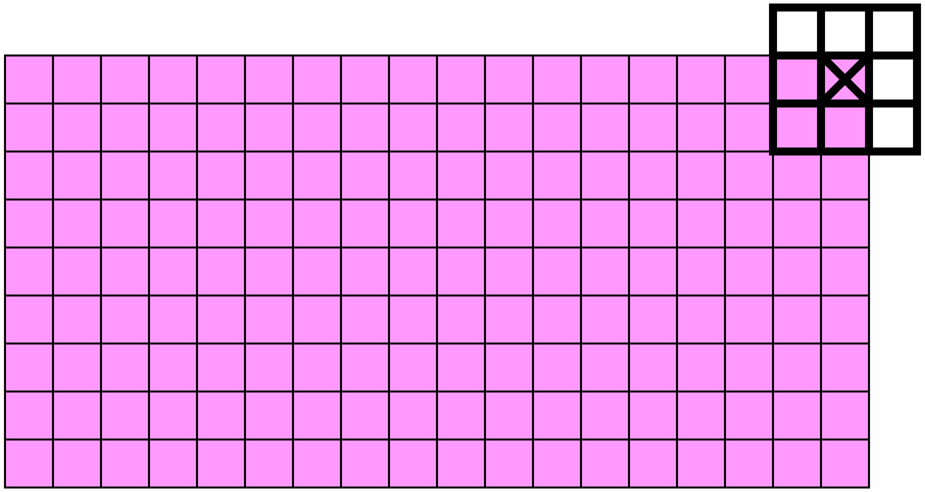
j = 2

Filtered:



Replace ☒ with the median of the values under the window.

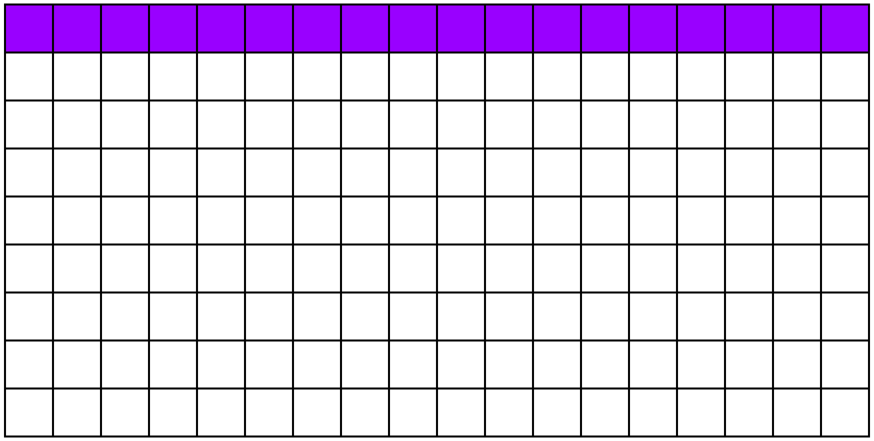
Original:



i = 1

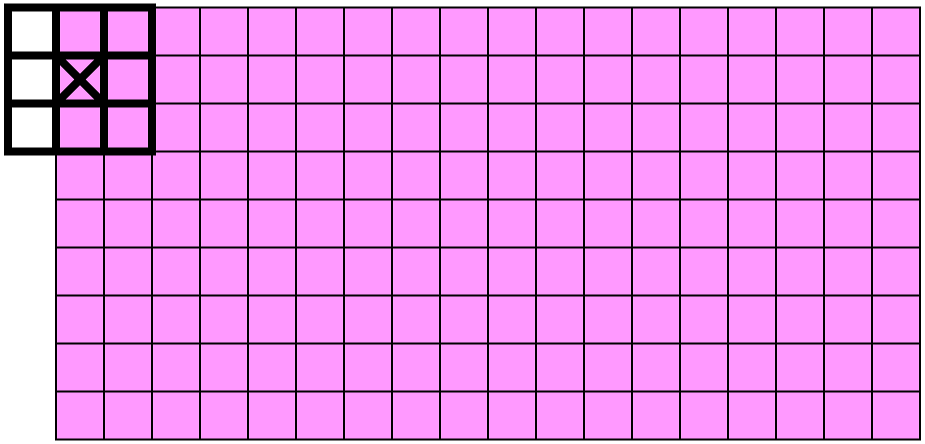
j = n

Filtered:



Replace ☒ with the median of the values under the window.

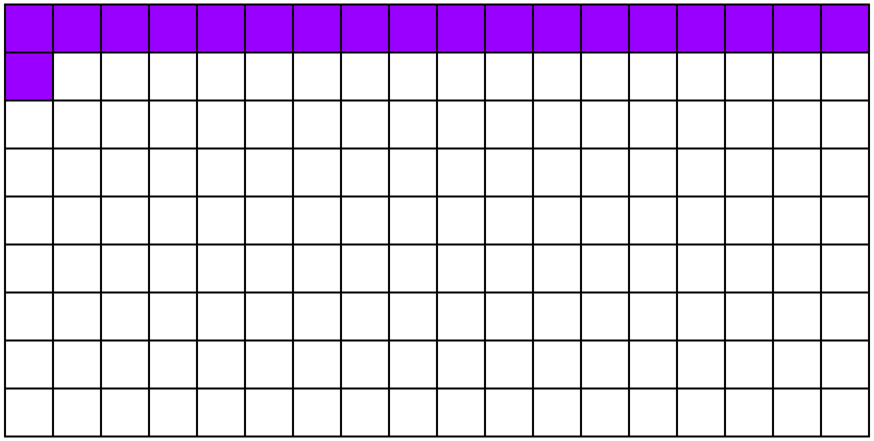
Original:



i = 2

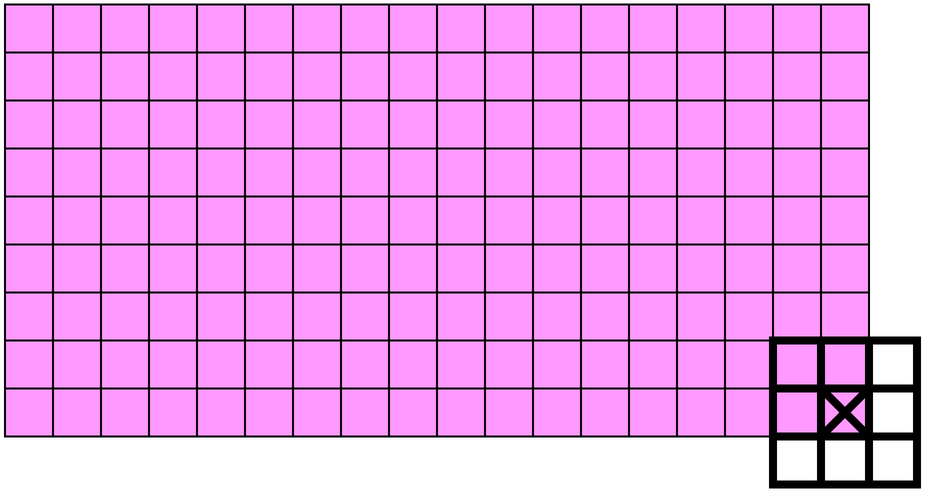
j = 1

Filtered:



Replace ☒ with the median of the values under the window.

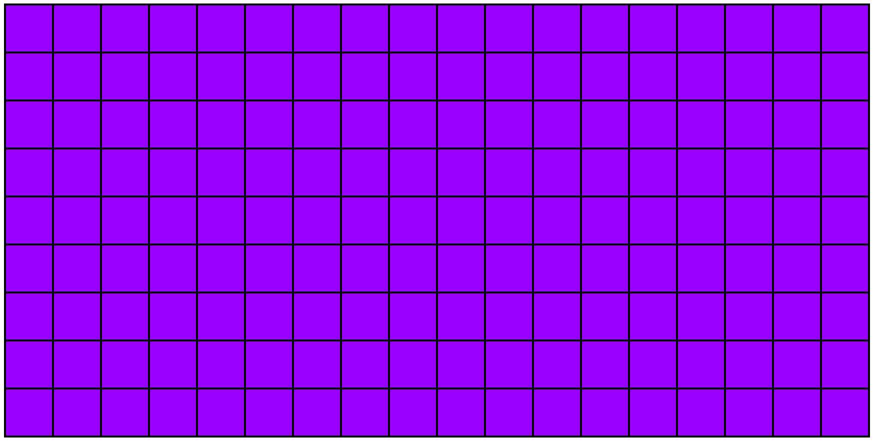
Original:



i = m

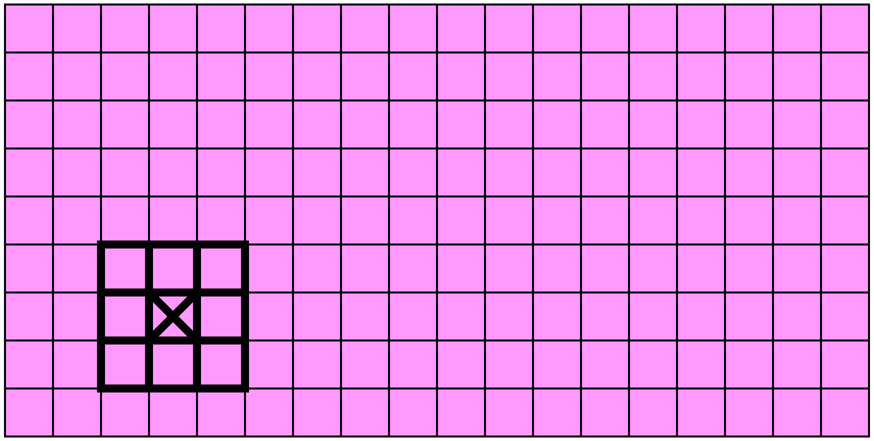
j = n

Filtered:



Replace ☒ with the median of the values under the window.

Window Inside…



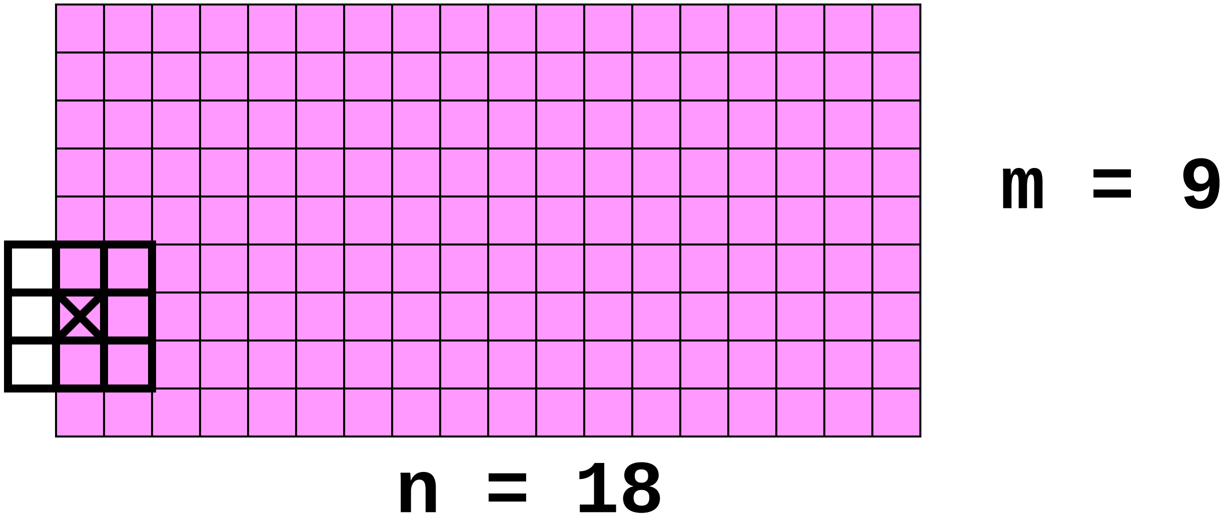
m = 9

n = 18

New gray value for pixel (7,4) =

```
medVal( A(6:8,3:5) )
```

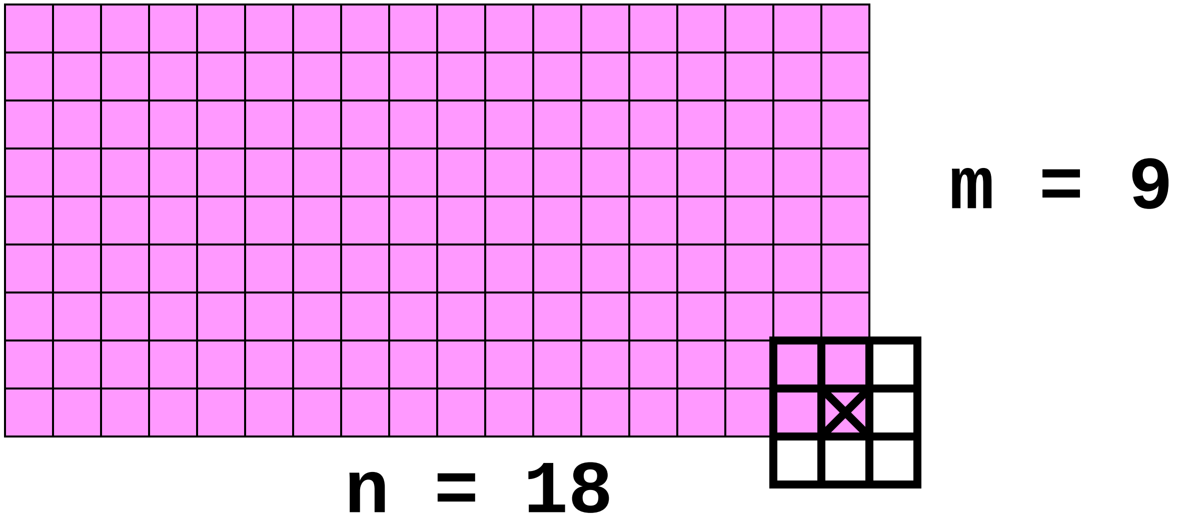
Window Partly Outside…



New gray value for pixel (7,1) =

medVal(A(6:8,1:2))

Window Partly Outside...



New gray value for pixel (9,18) =

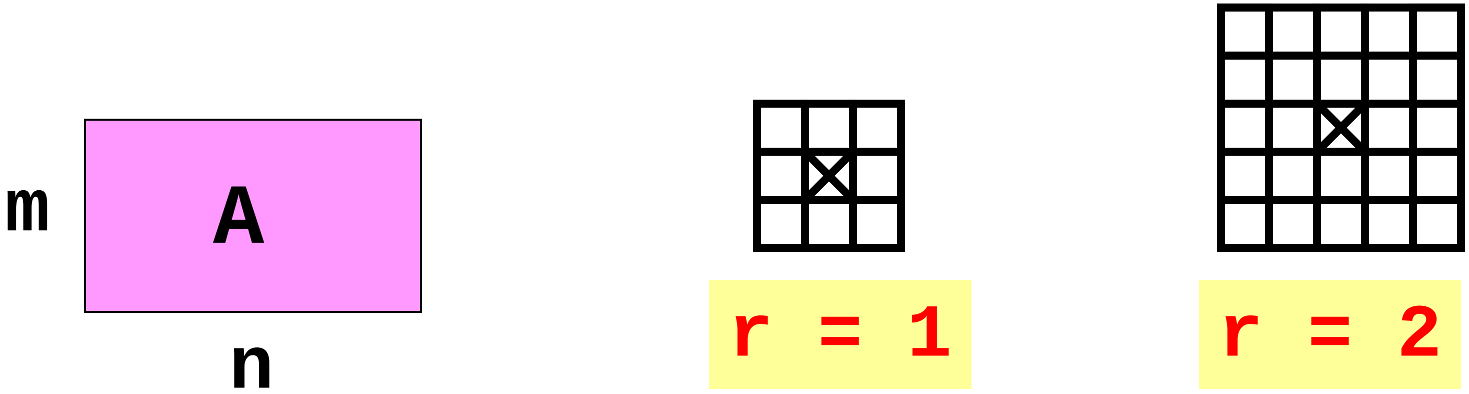
```
medVal( A(8:9,17:18) )
```

Filtering by Median

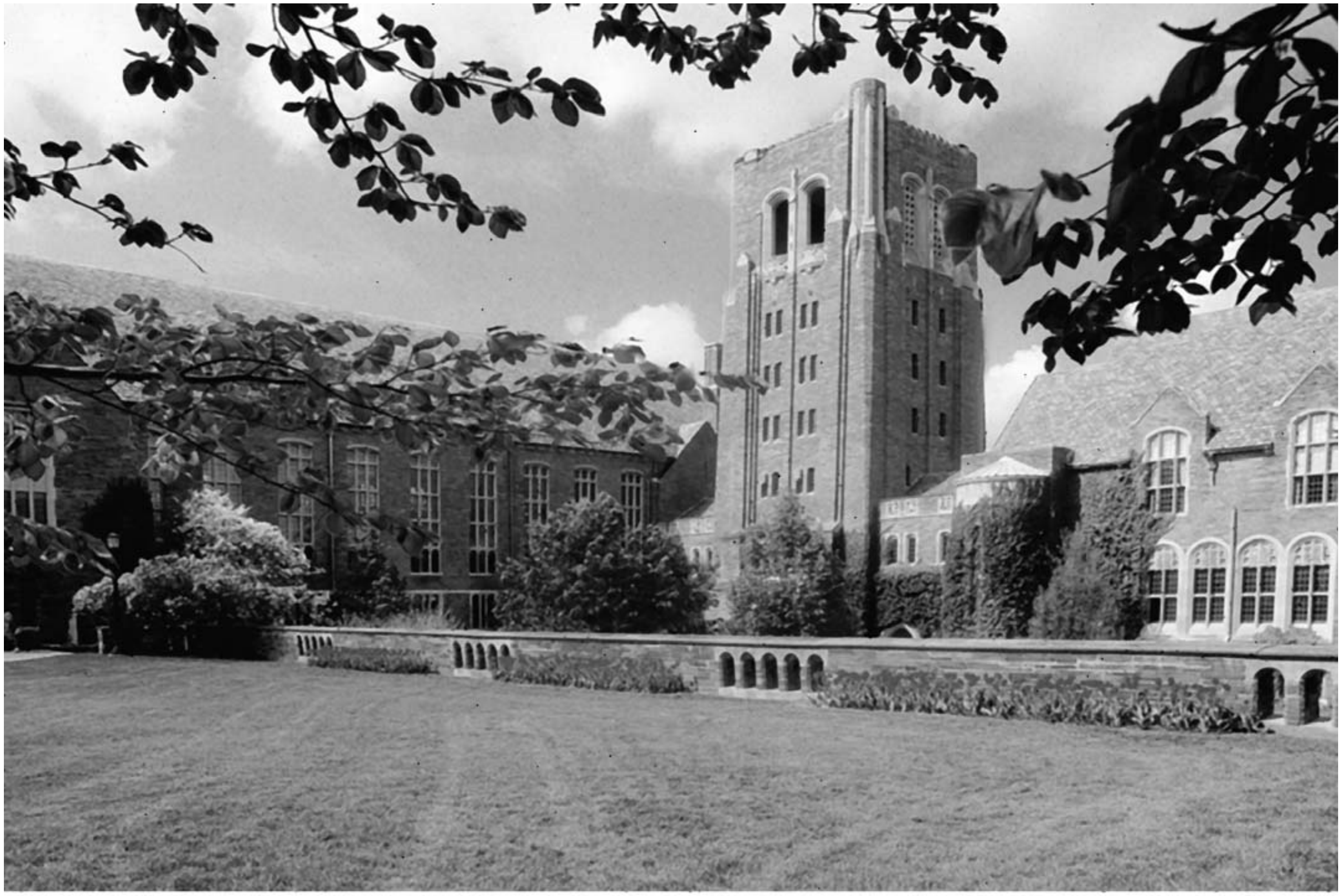
```
function B = MedianFilter(A,r)
% B is a uint8 array obtained from A by median filtering
% with radius r neighborhoods.
[m,n] = size(A);
B = uint8(zeros(m,n));
for i=1:m
    for j=1:n
        C = pixel (i,j) neighborhood
        B(i,j) = MedVal(C);
    end
end
```

The Pixel (i,j) Neighborhood

```
iMin = max(1,i-r)
iMax = min(m,i+r)
jMin = max(1,j-r)
jMax = min(n,j+r)
C = A(iMin:iMax,jMin:jMax)
```

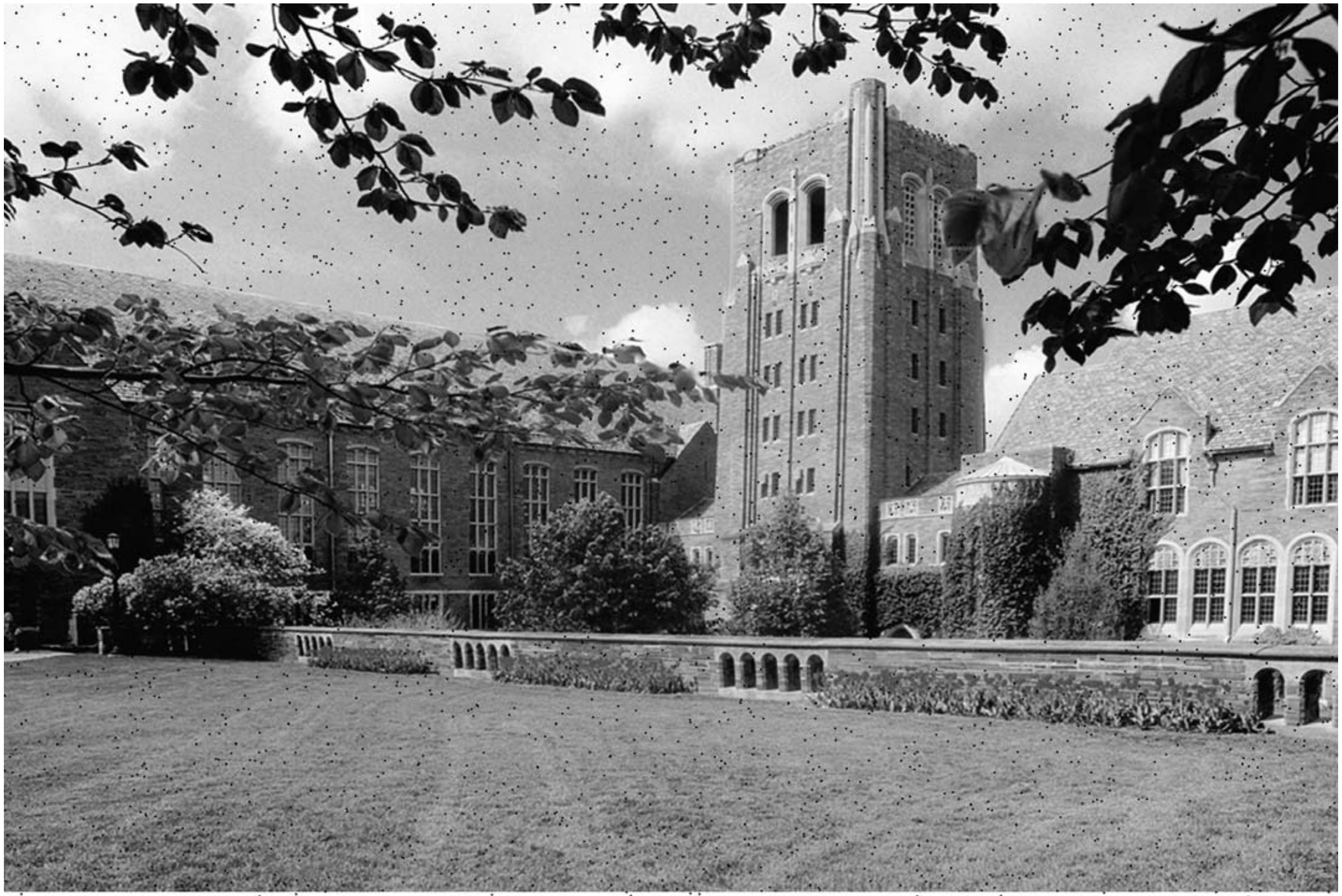


B = MedianFilter(A, 2);



Cornell University Law School
Photograph by Cornell University Photography

Before Filtering

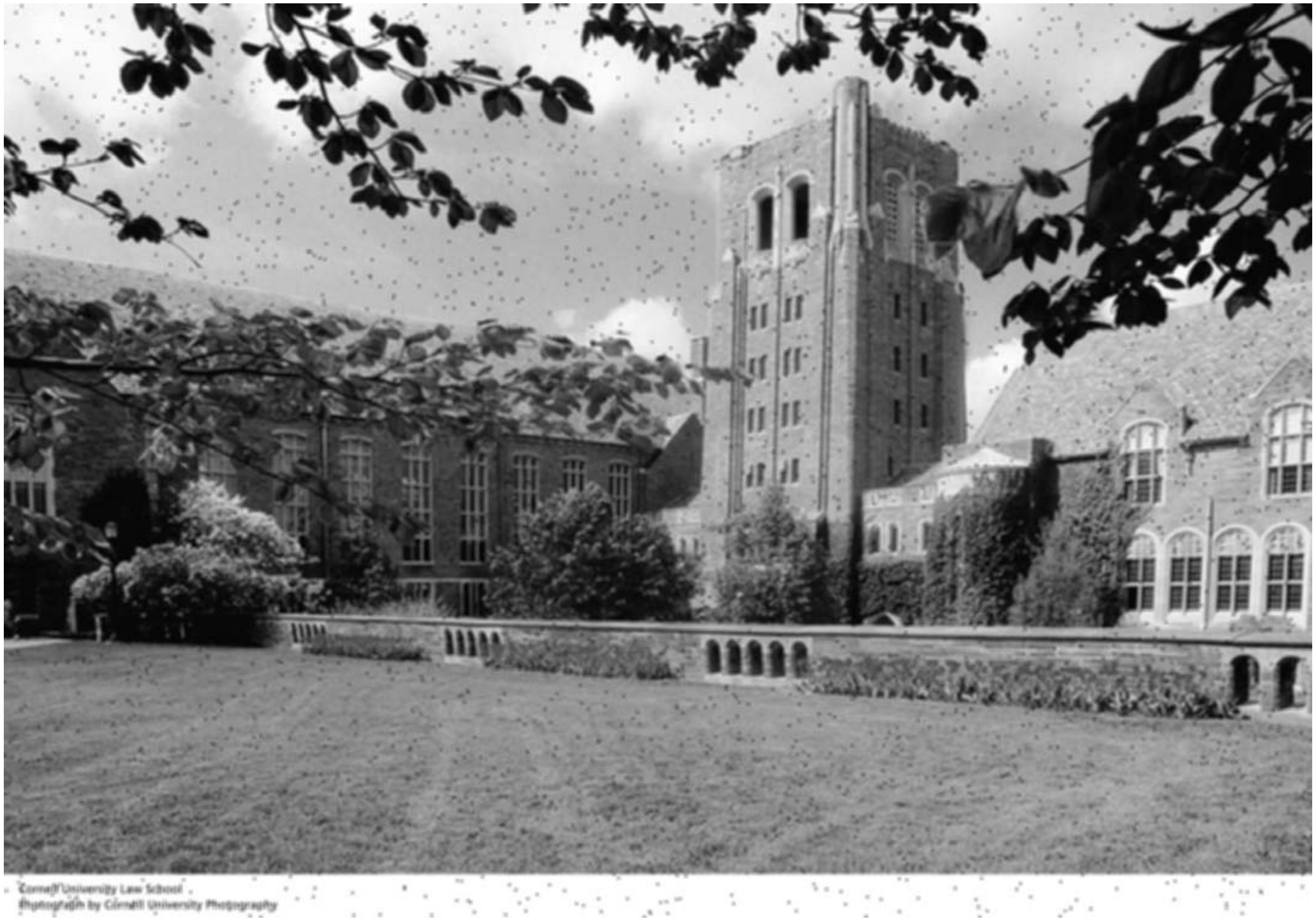


Cornell University Law School
Photograph by Cornell University Photography

What About Using the Mean Instead of the Median?

Replace each gray value with the average gray value in the radius r neighborhood

Mean Filter with r = 3



Cornell University Law School
Photograph by Cornell University Photography

Why it Fails

150	149	152	153	152	155
151	150	153	154	153	156
153	2	3	156	155	158
154	2	1	157	156	159
156	154	158	159	158	161
157	156	159	160	159	162

85	86
87	88

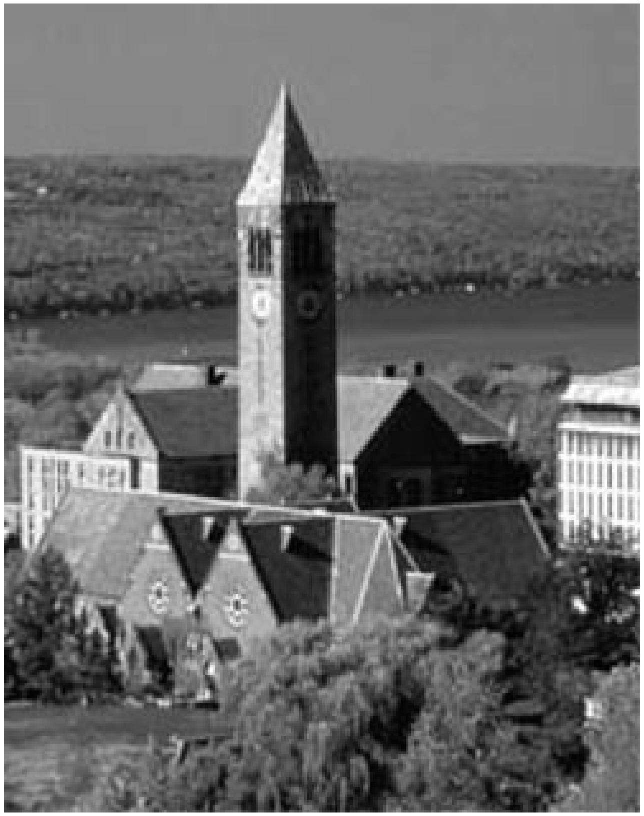
The mean does not capture representative values

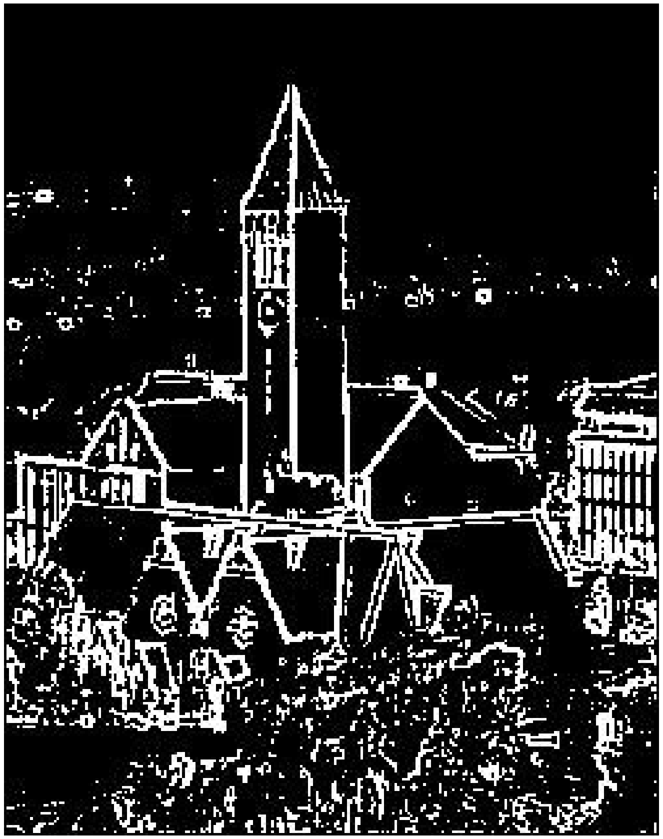
And Median Filters Leave Edges (Pretty Much) Alone

200	200	200	200	200	200
200	200	200	200	200	100
200	200	200	200	100	100
200	200	200	100	100	100
200	200	100	100	100	100
200	100	100	100	100	100

Inside the box, the 200’s stay at 200 and the 100’s stay at 100

Finding Edges

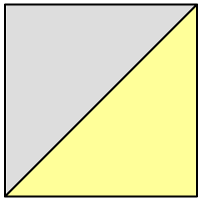




What is an Edge?

Near an edge, grayness values change abruptly.

200	200	200	200	200	200
200	200	200	200	200	100
200	200	200	200	100	100
200	200	200	100	100	100
200	200	100	100	100	100
200	100	100	100	100	100



The Rate-of-Change Array

- Suppose *A* is an image array with integer values between 0 and 255
  - Let B(i,j) be the maximum difference between A(i,j) and any of its eight neighbors

Example

90	81	65
62	60	59
56	57	58

Rate-of-change at middle pixel is 30

Computing the Rate-Of-Change Array

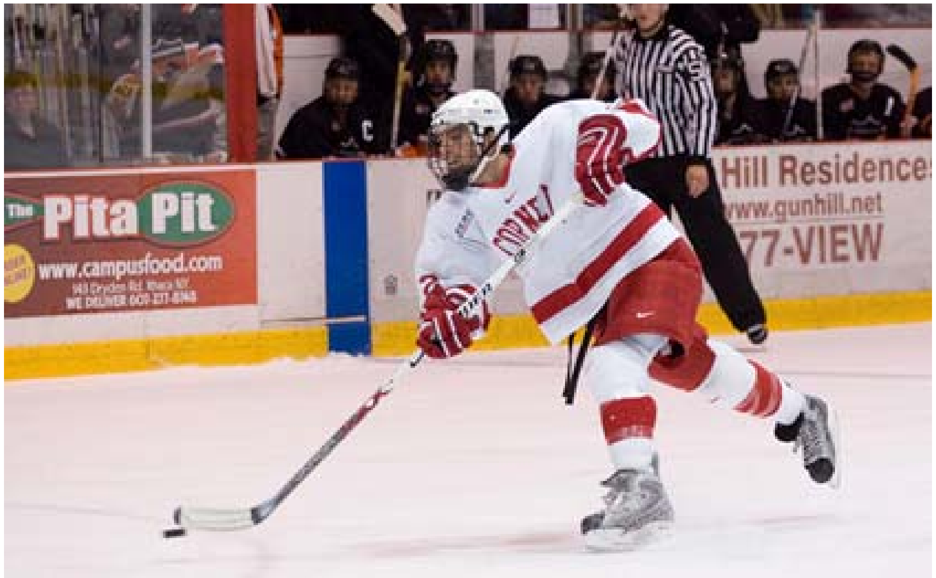
```
function B = Edges(P)
% P is a jpeg file
% B is the corresponding Rate-Of-Change array

A = double(rgb2gray(imread(P)));
[m,n] = size(A);
B = uint8(zeros(m,n));
for i=2:m-1
    for j = 2:n-1
        B(i,j) = ???
    end
end
```

Recipe for B(i,j)

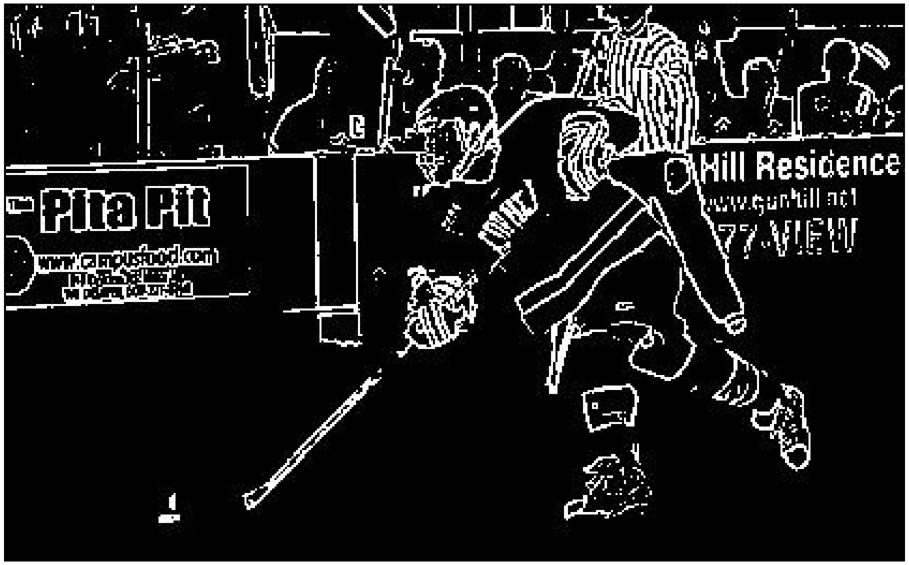
```
% The 3-by-3 subarray: A(i,j) and its 8 neighbors…
  Neighbors = A(i-1:i+1,j-1:j+1);
% Subtract A(i,j) from each entry…
  Diff = Neighbors – A(i,j));
% Take absolute value of each entry..
  posDiff = abs(Diff);
% Compute largest value in each column…
  colMax = max(posDiff);
% Compute the max of the column max’s…
  B(I,j) = max(colMax)
```

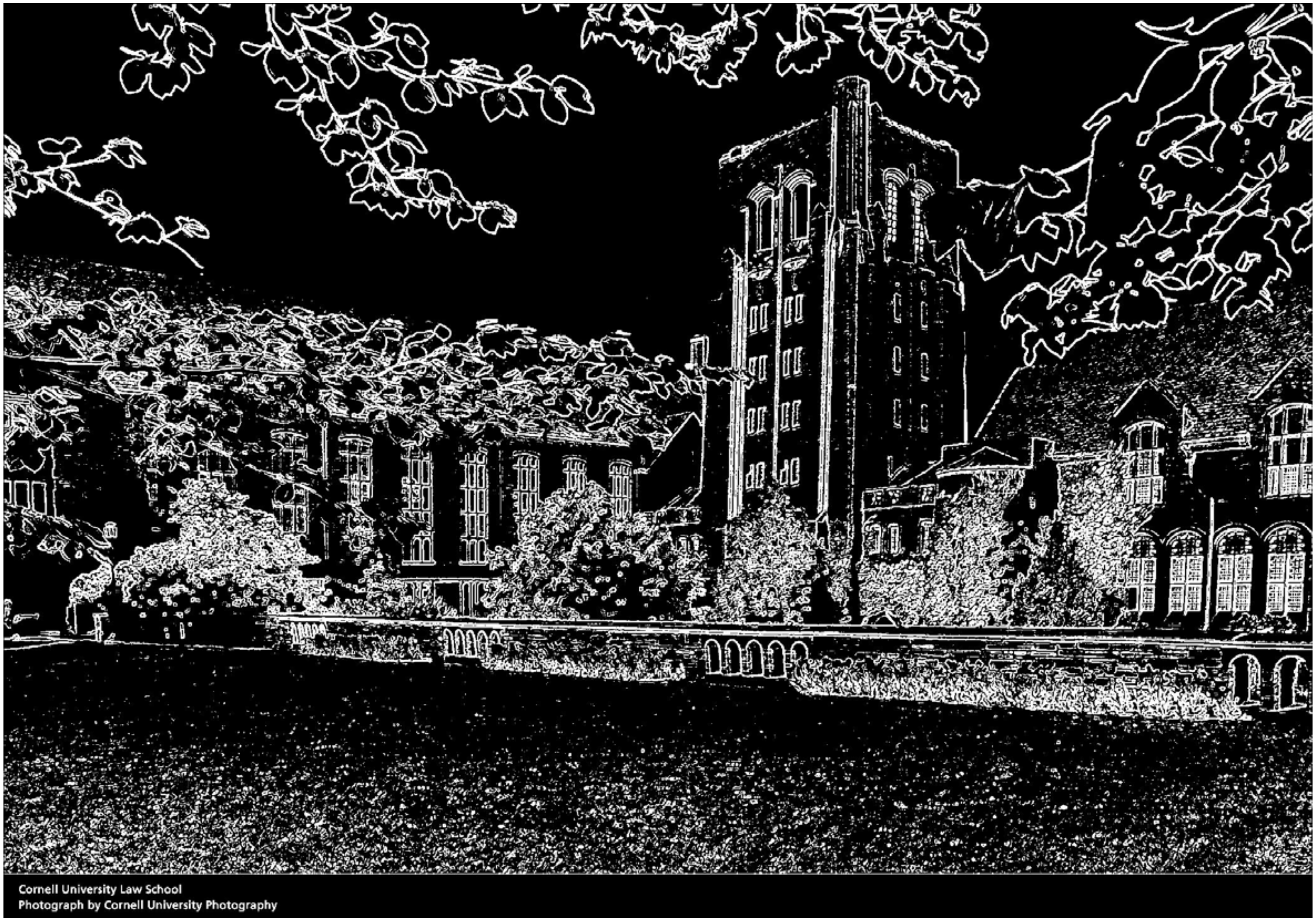
Rate-of-Change Array to Image

```
  B = Edges('Tower.jpg');
% Compute 0-1 array: 1 for B entries > 20
  importantPixels = B > 20;
% Display those pixels with maximum brightness
  C = uint8( 255*importantPixels );
  imshow(C)
```



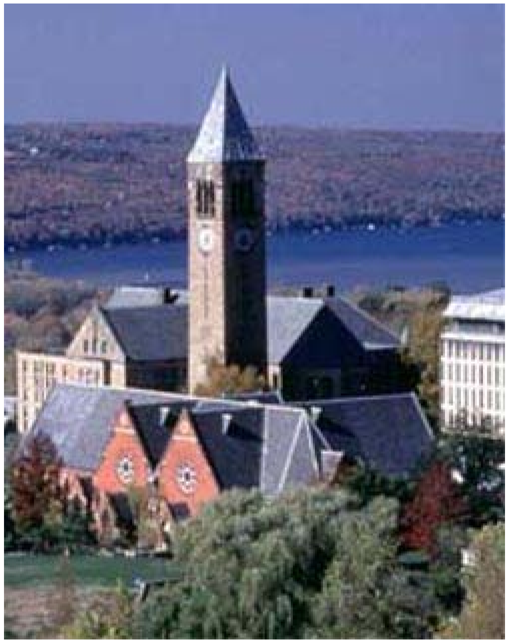
Threshhold
= 40

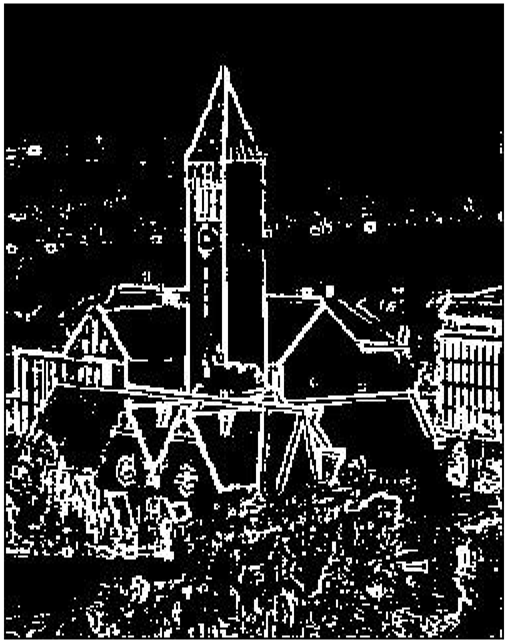




Cornell University Law School
Photograph by Cornell University Photography

Threshhold = 20





Threshhold = 30

Prelim 2

- Statistics

Mean	85.9
Median	88
StDev	10.4

- Difficulties
  - 1b: Shifting data to match Matlab's subscript rules
  - 5a: Splitting a string based on a finding a substring