CMPT 732-G200 Practices for Visual Computing

Ali Mahdavi Amiri

Images

- An array of pixels

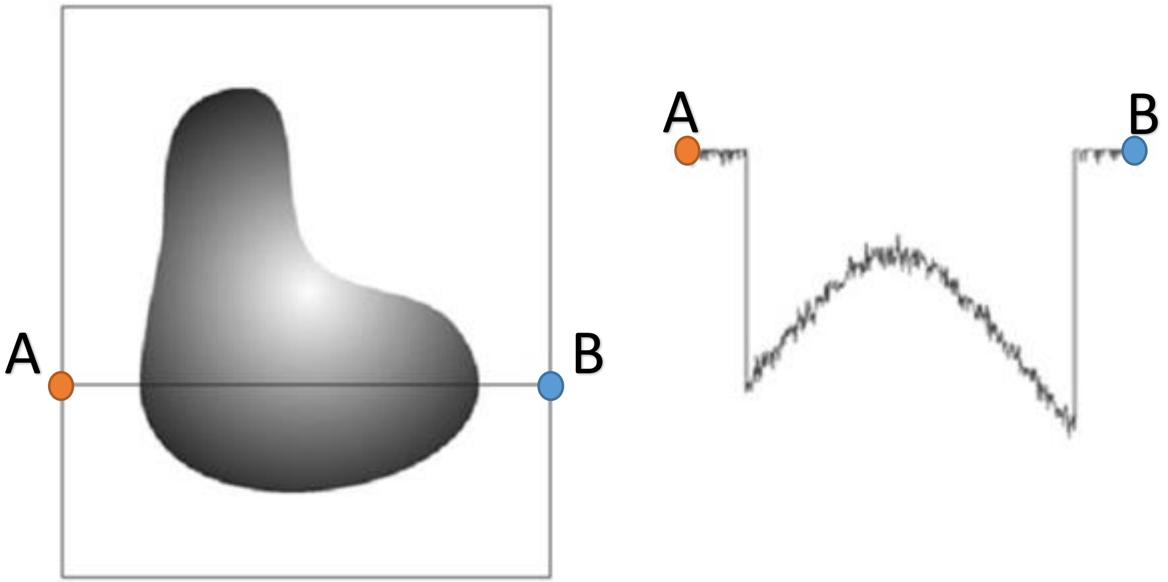
Images

- An array of pixels

- Each pixel has integer values
  - 1 value for grey scale
  - 3 values for RGB images

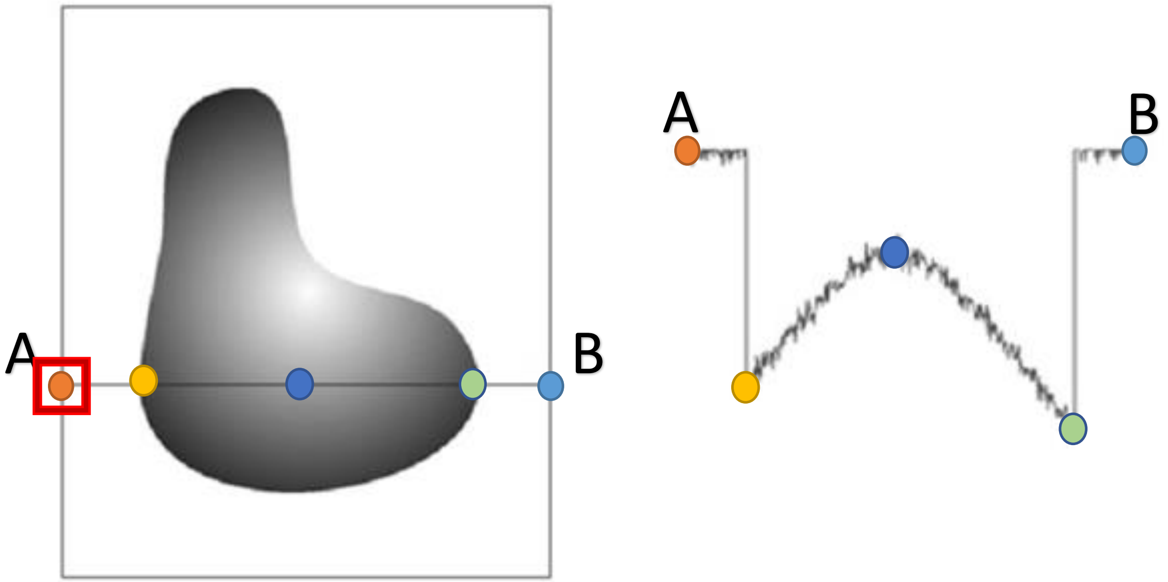
Digitizing Images

- Images are capturing continuous phenomena

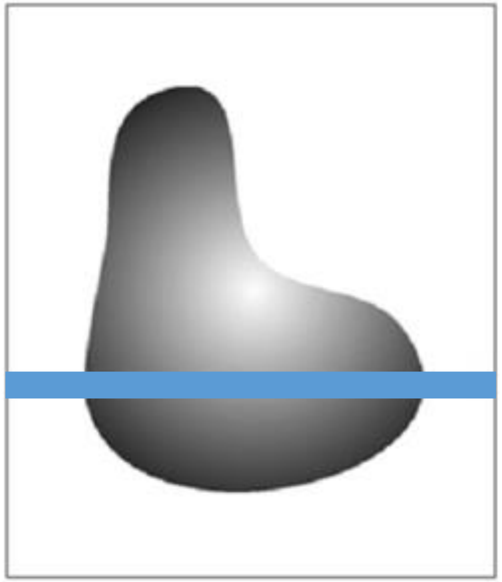


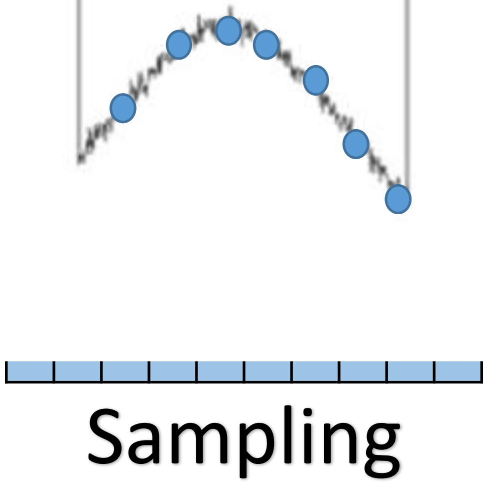
Digitizing Images

- Images are capturing continuous phenomena

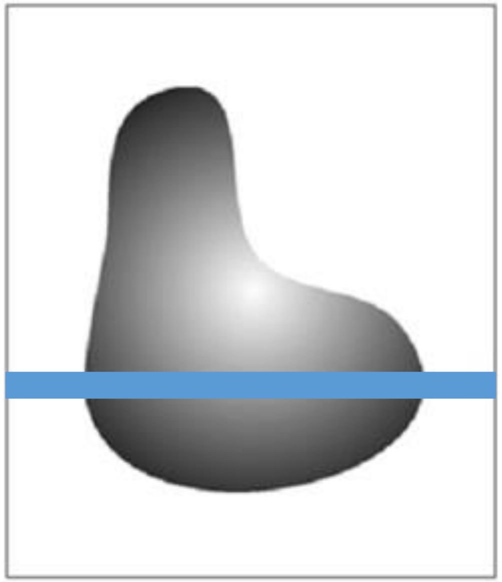


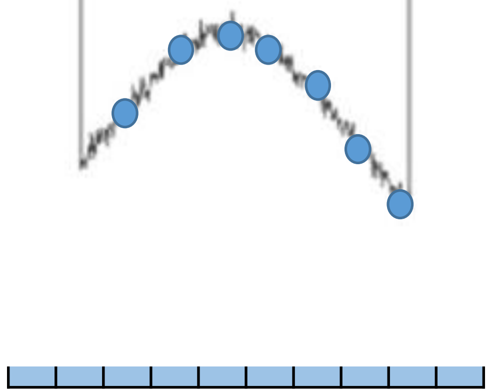
Sampling



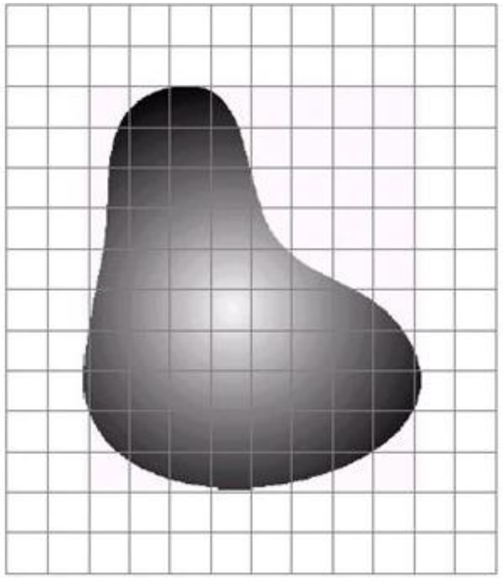


Sampling

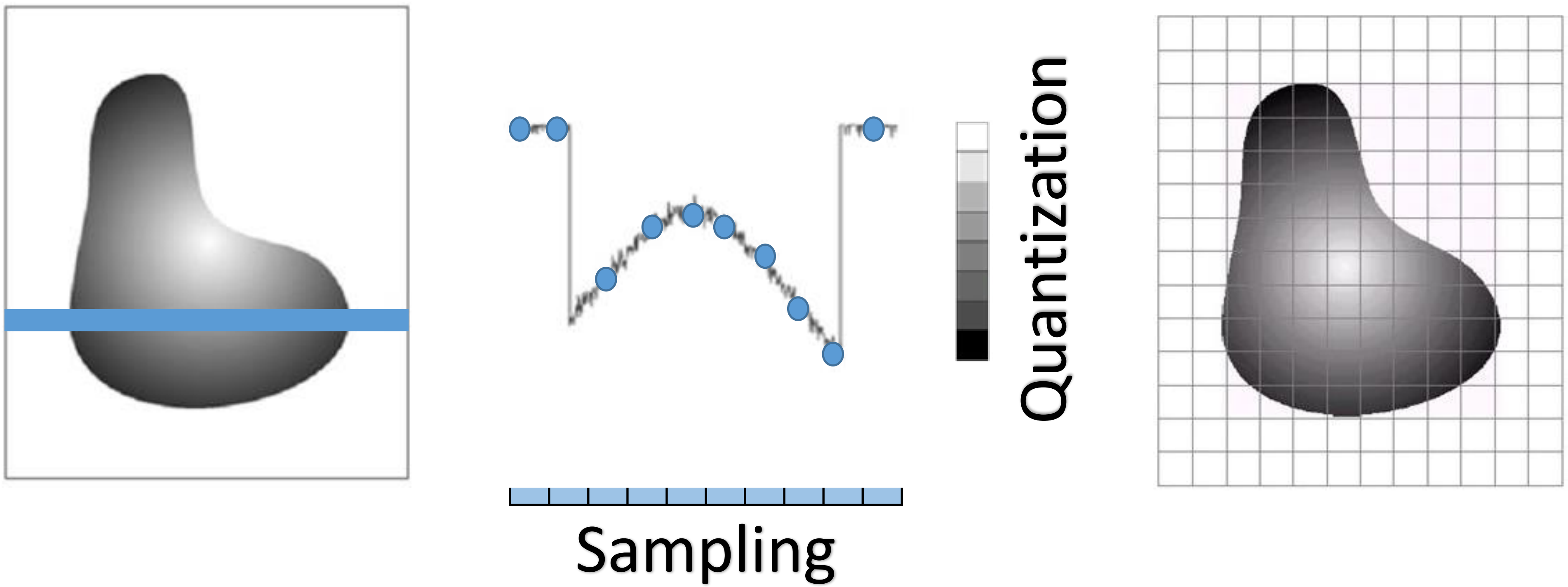




Sampling



Quantization



Quantization

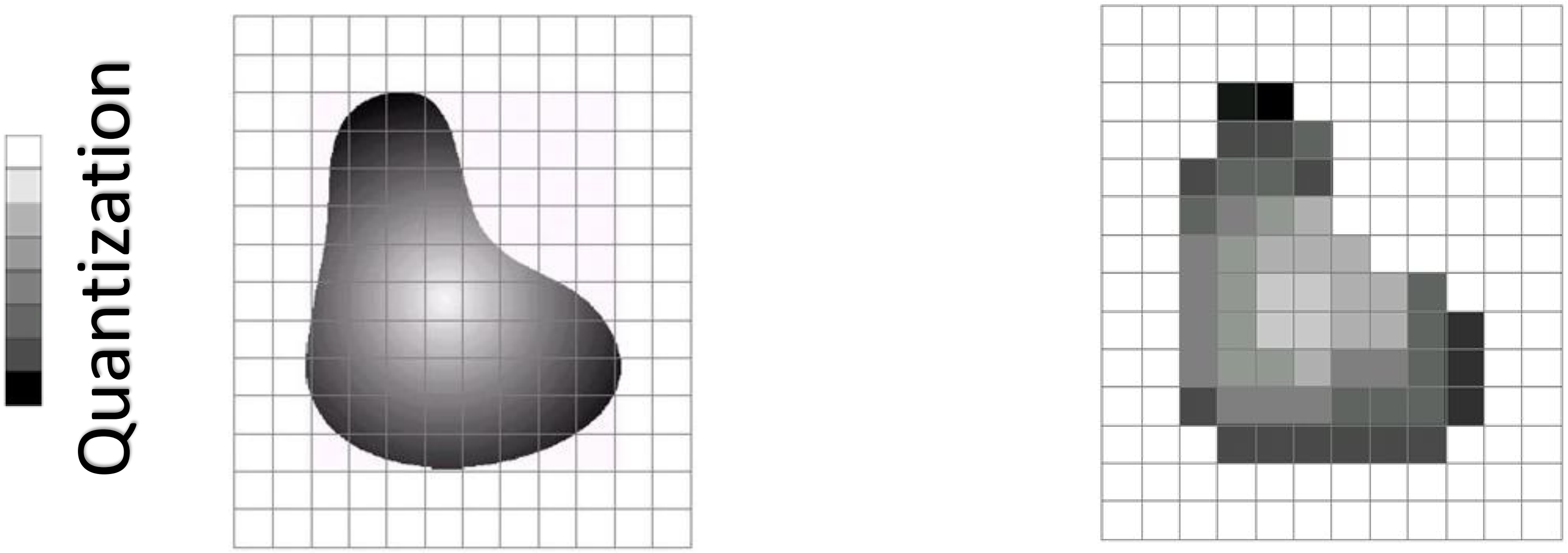


Image Artifacts

- What kind of image artifact is the result of quantization?

Image Artifacts

- Saturation

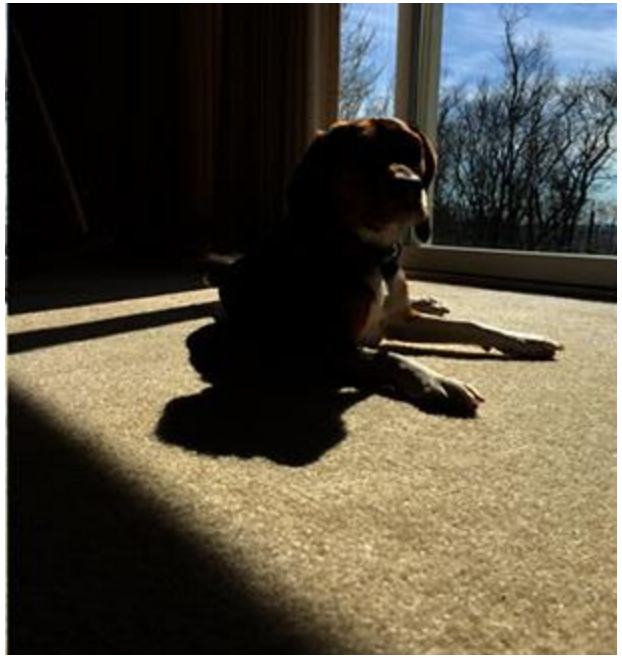
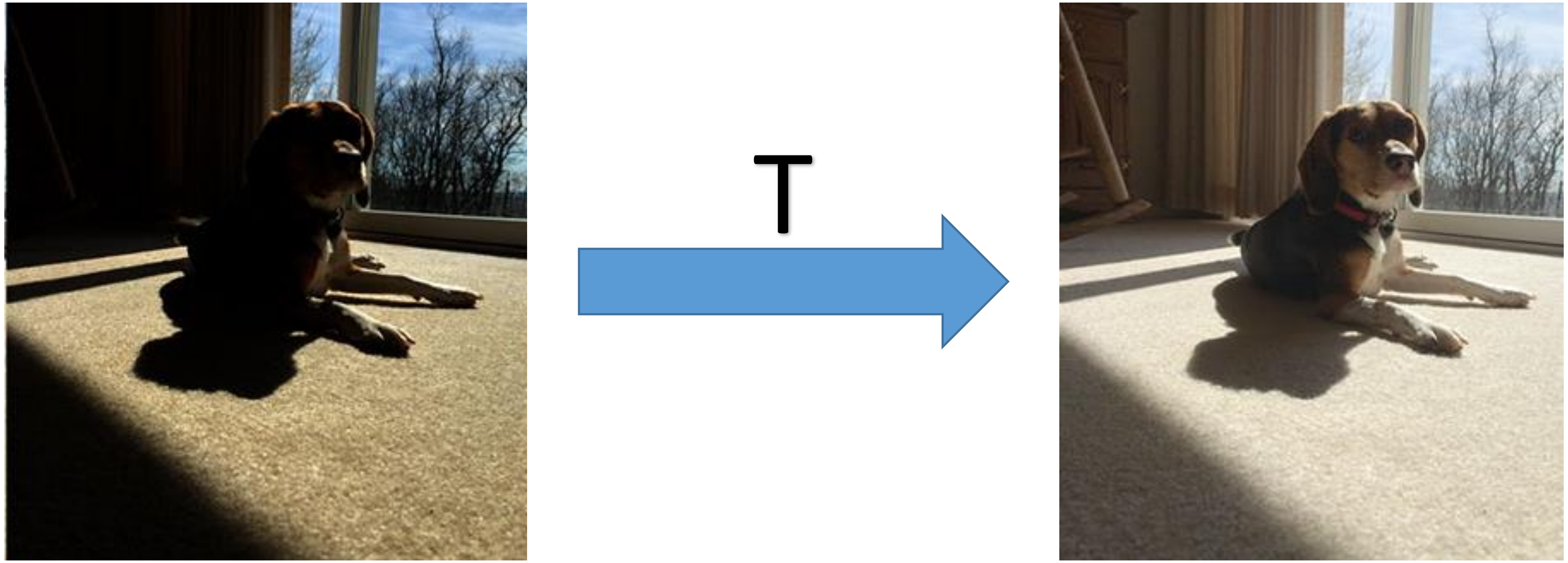


Image Enhancement

- Can we turn on the lights?

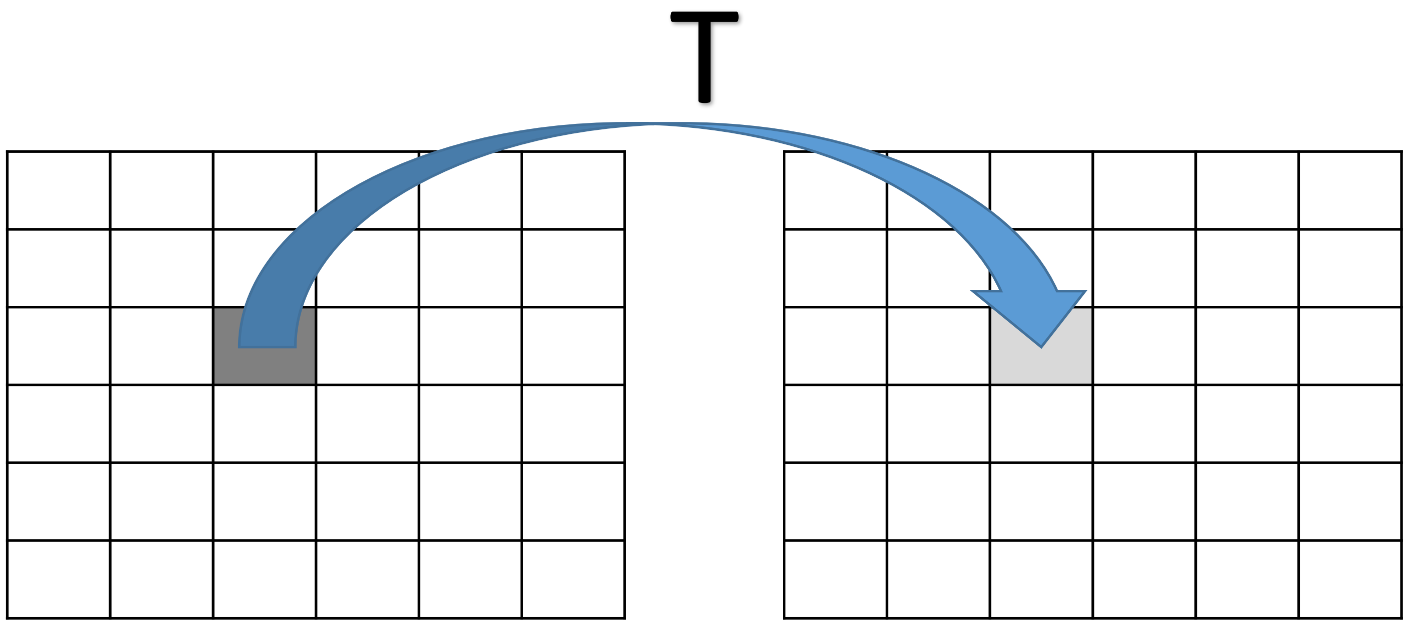
Image Enhancement

- Finding a transformation to re-arrange the pixel value distribution.



Transformations

- Pixel to pixel
  - No information about neighborhood



Transformations

- A transformation function to manipulate the intensity values of an image

Transformations

- A transformation function to manipulate the intensity values of an image

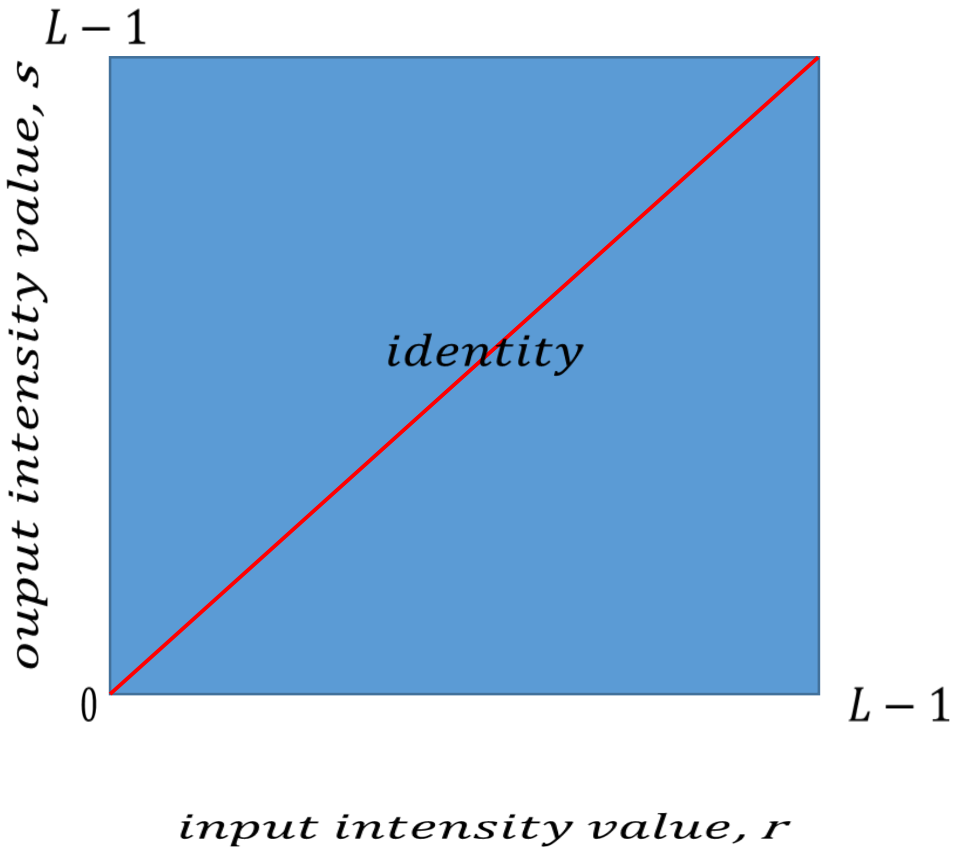
Original Intensity value

$$s = T(r)$$

New Intensity value

Transformation Options

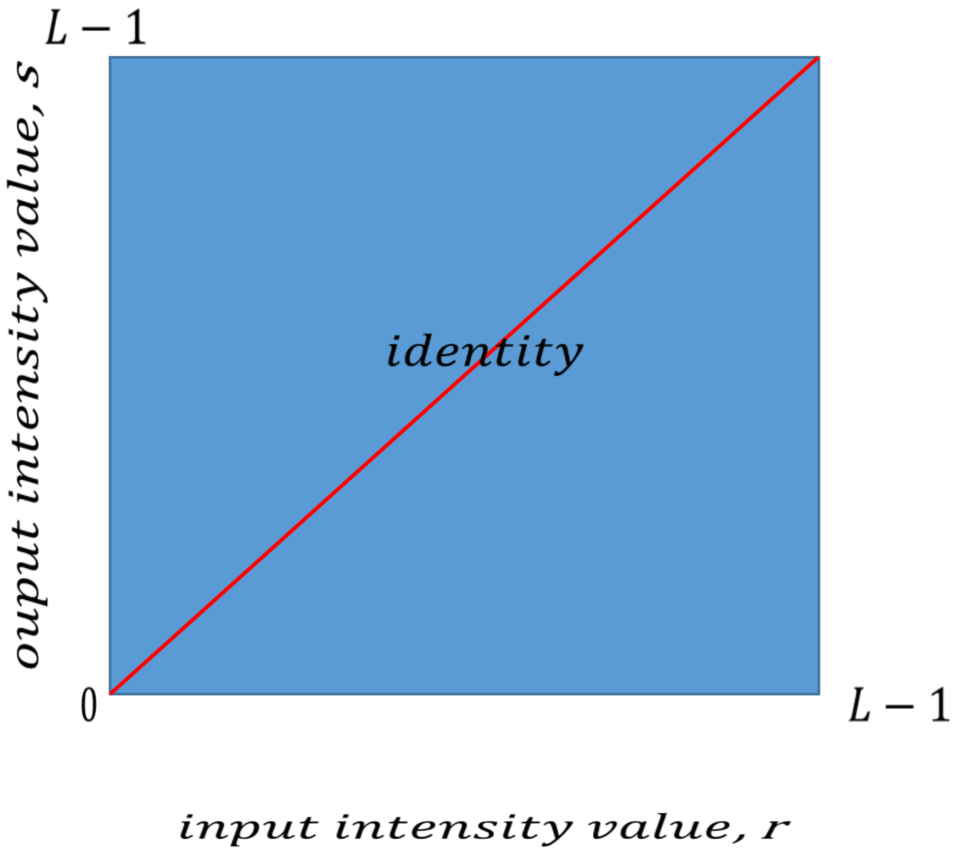
- What type of functions are candidates for T?



$$s = T(r) = ?$$

Transformation Options

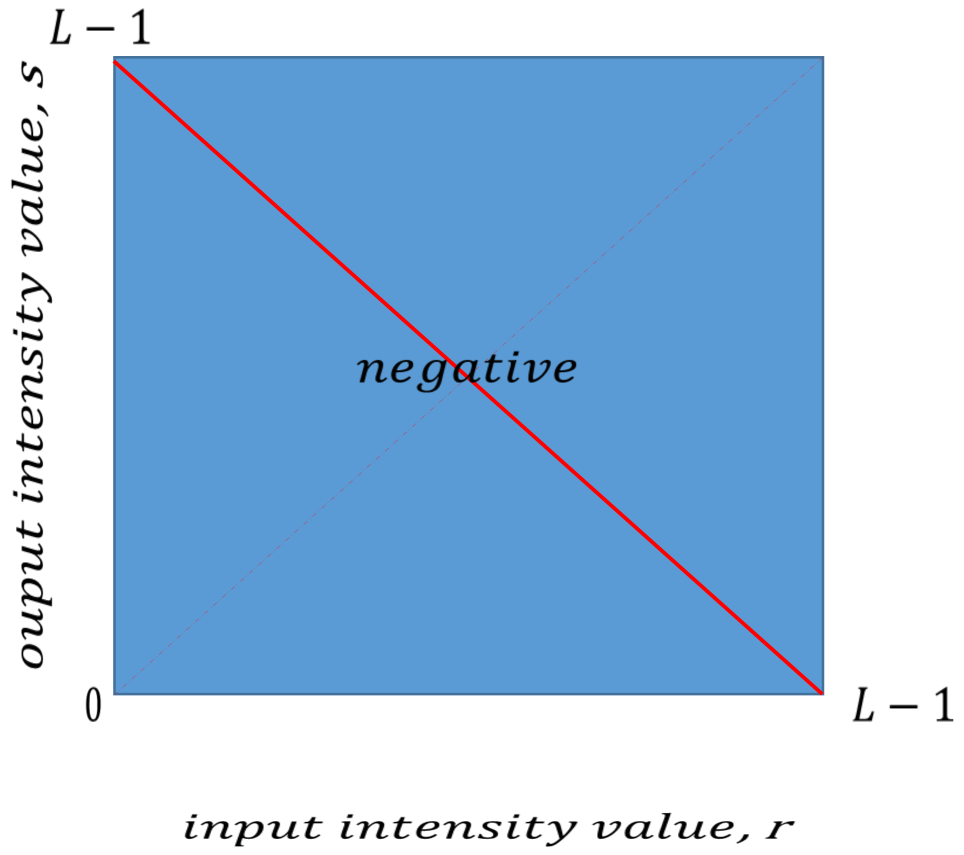
- Identity function



$$s = r$$

Transformation Options

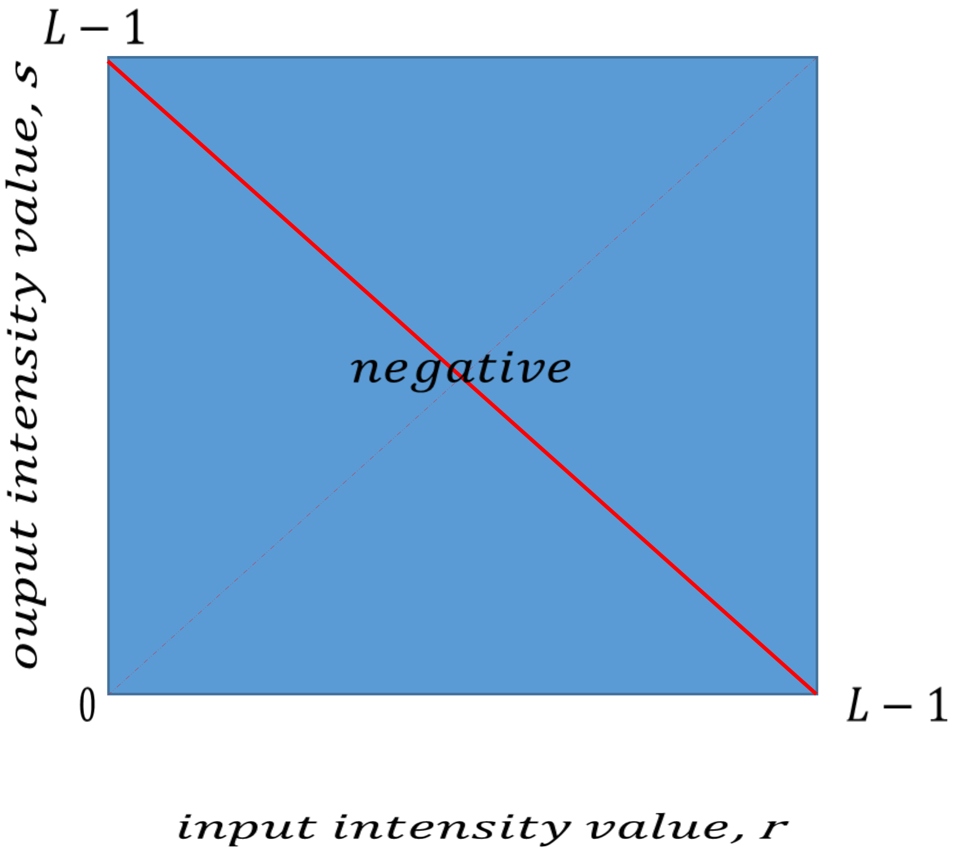
- Negative function



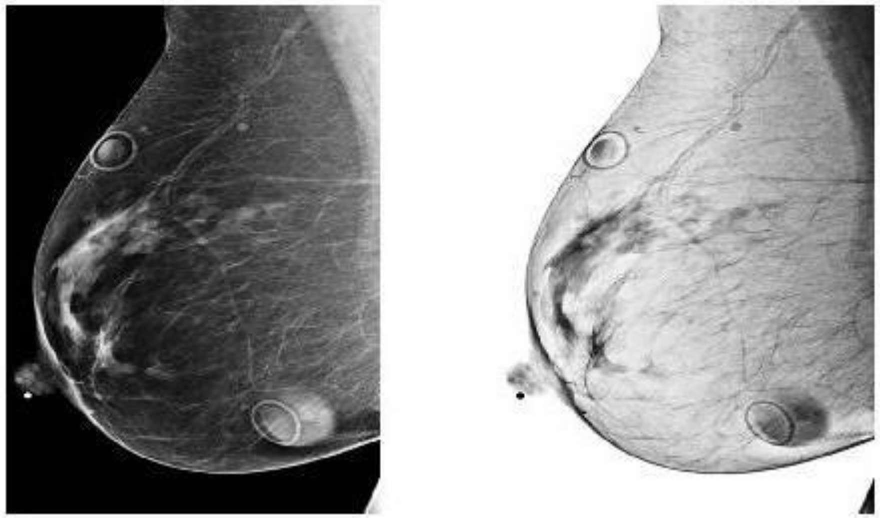
$$s = T(r) = ?$$

Transformation Options

- Negative function

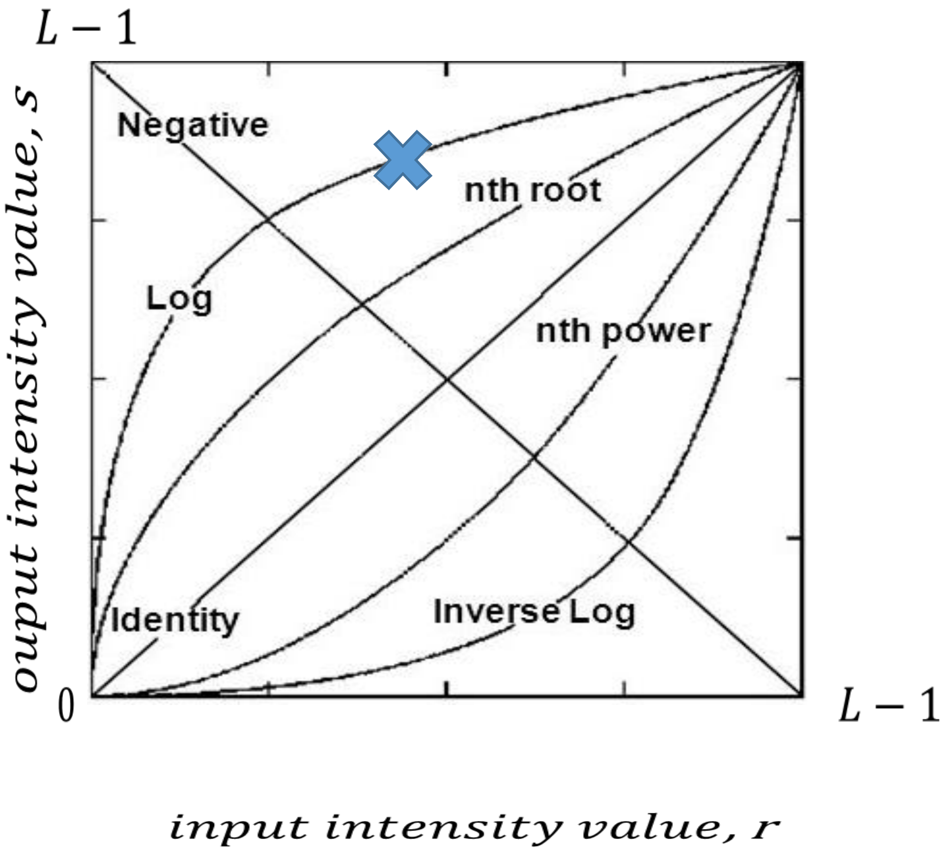


$$s = L - 1 - r$$

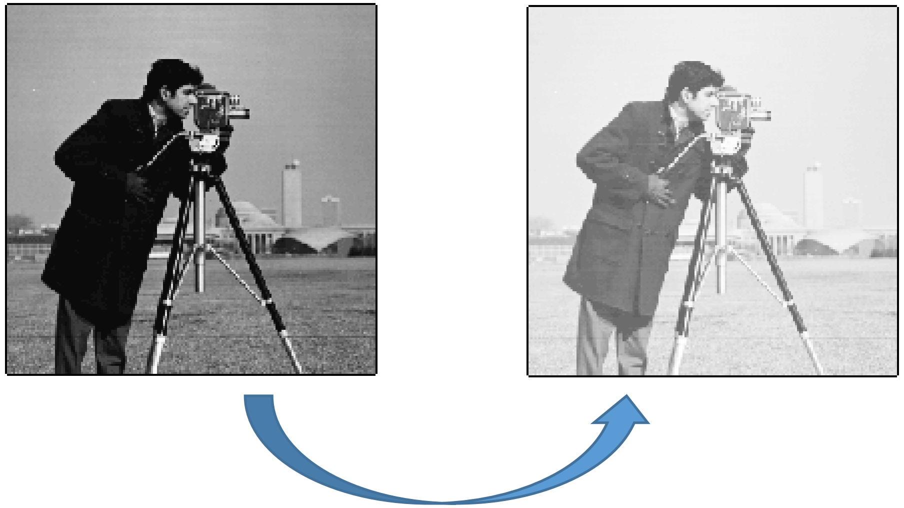


Transformation Options

- Log function

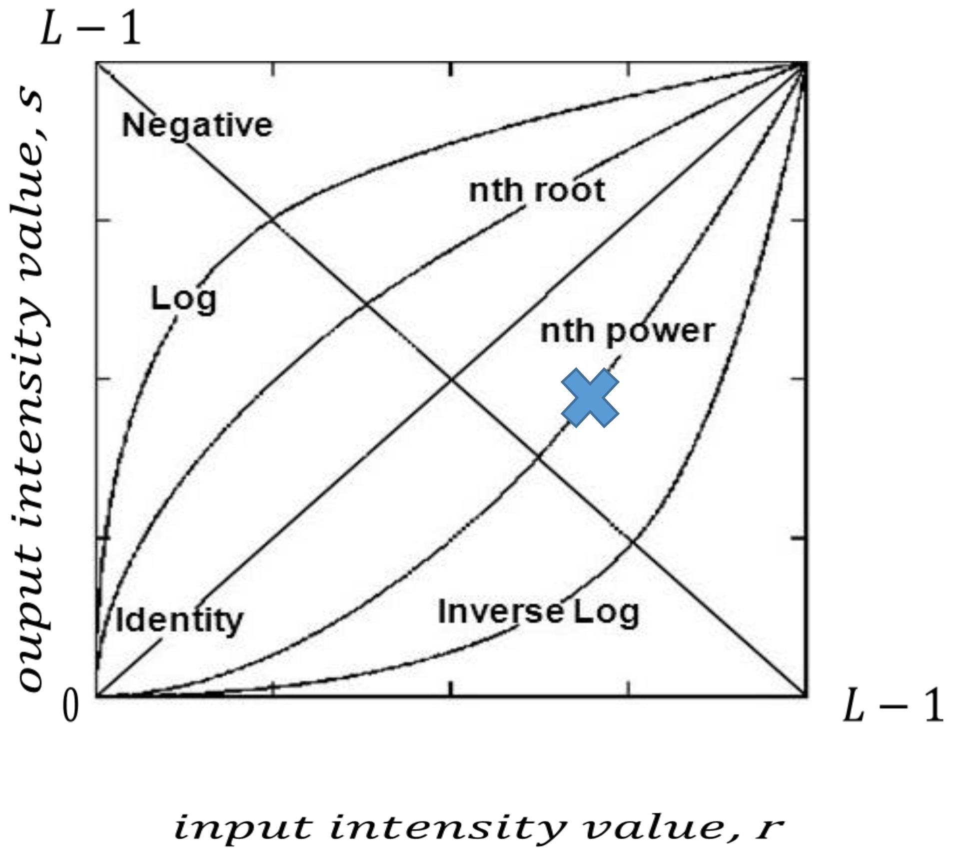


$$s = c\log(r+1)$$



Transformation Options

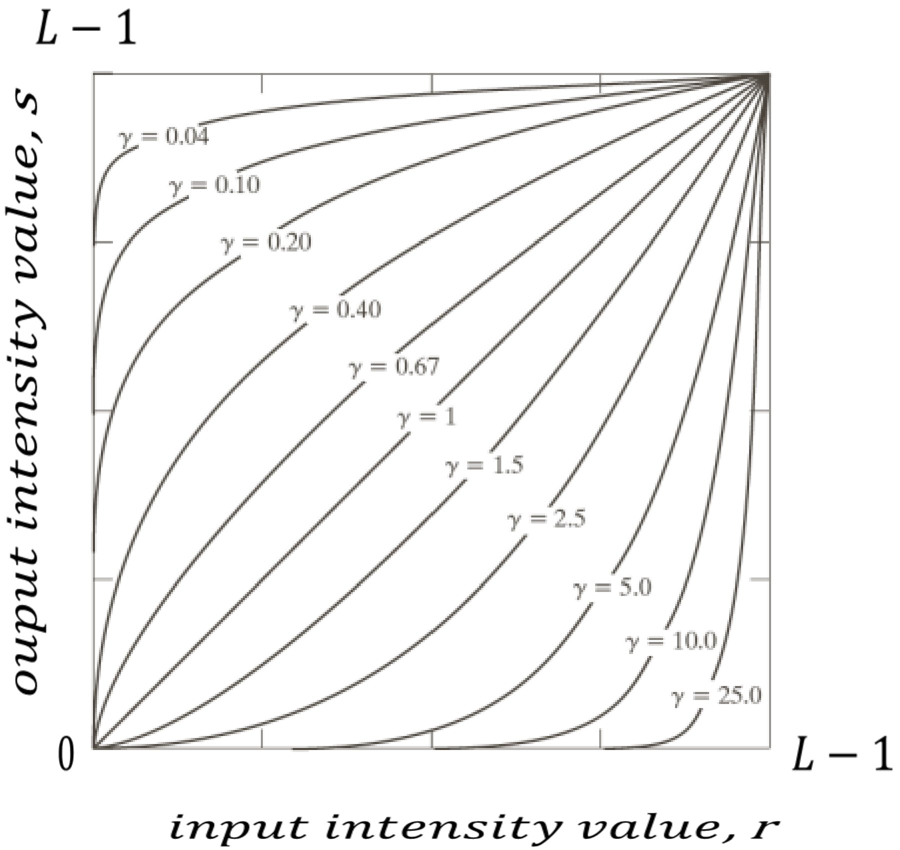
- Power-Law (Gamma) transformation



$$s = cr^{\gamma}$$

Transformation Options

- Changing gamma makes a family of functions



$$s = cr^{\gamma}$$

Transformation Options

- Power-Law (Gamma) transformation

Original image
MRI of a fractured human spine

$c = 1$

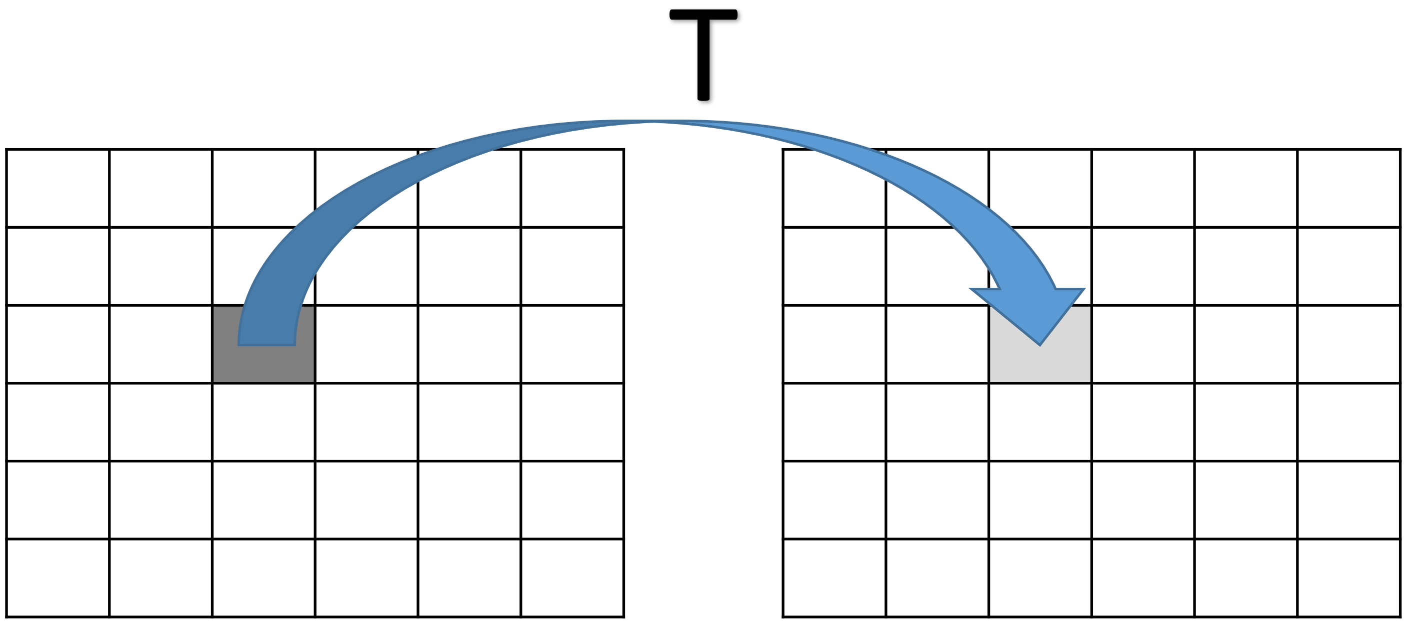
$\gamma = 0.6$

$\gamma = 0.4$

$\gamma = 0.3$

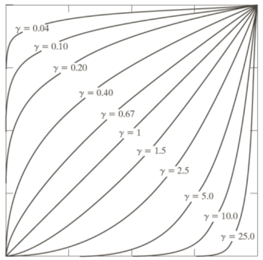
Recap (so far)

- We defined a set of simple pixel to pixel transformations that enhance the quality of images to reveal more details/information.



Cons

- Hard to find the best transformation function.

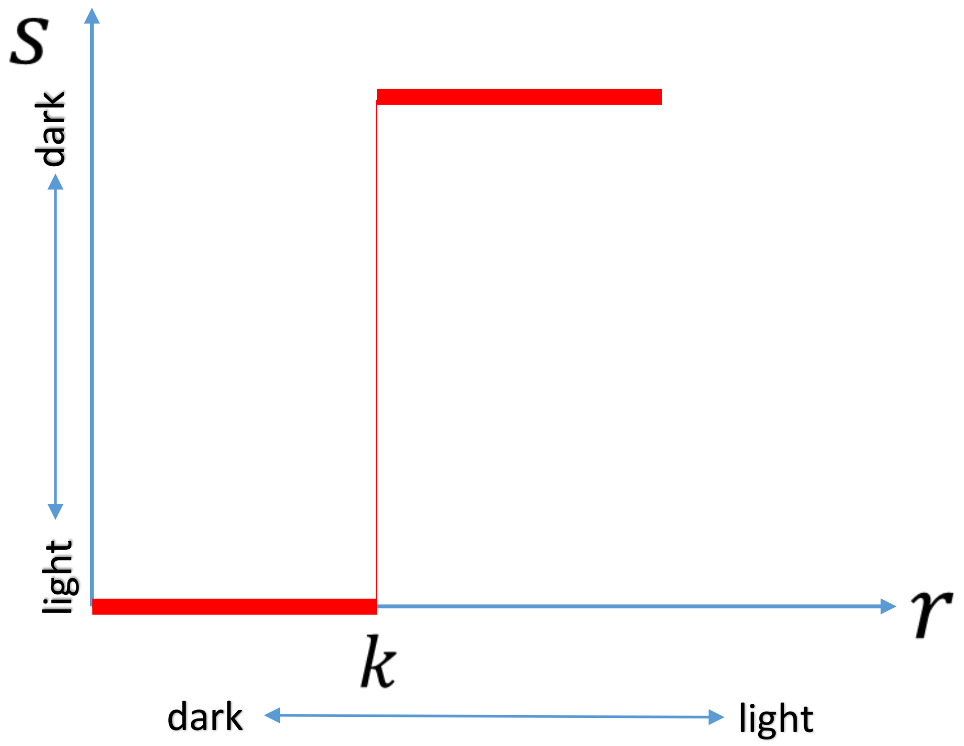


- It is blind to the distribution of pixels.
  - What if all the pixels are between 0-32?

- What is an alternative?

Thresholding Transformations

- 1D function that maps all the pixels smaller than a value to zero, larger than a value to (e.g., 1 or k).

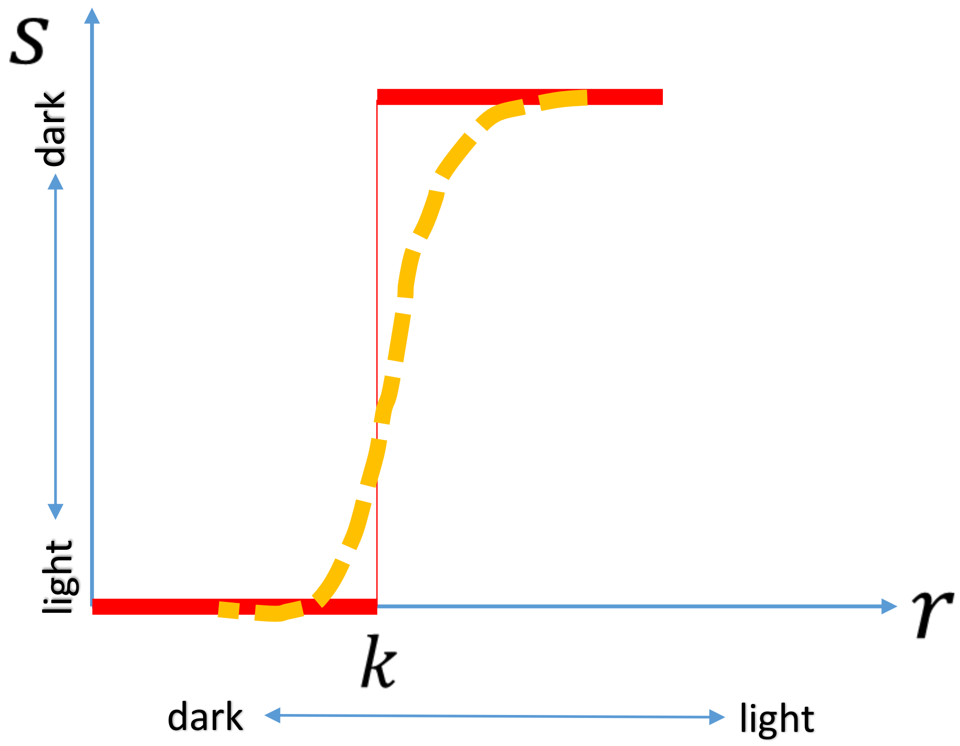


$$s = T(r)$$

Thresholding function

Stretching Transformations

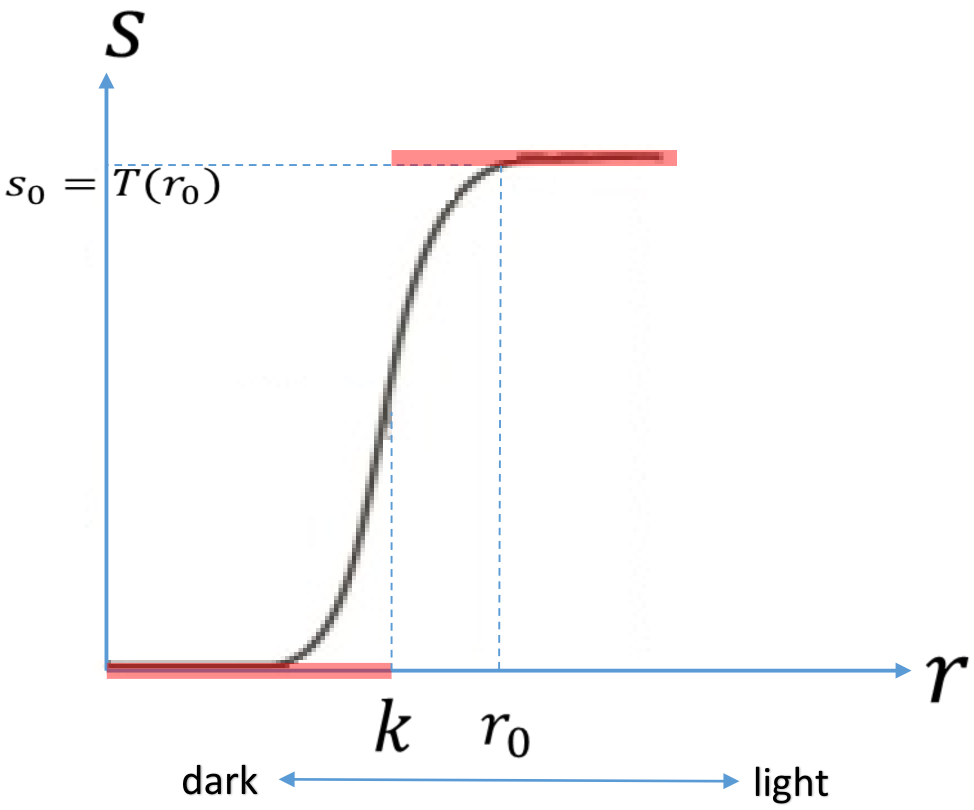
- We want to somehow smooth the middle to stretch the contrast.



$$s = T(r)$$

Stretching Transformations

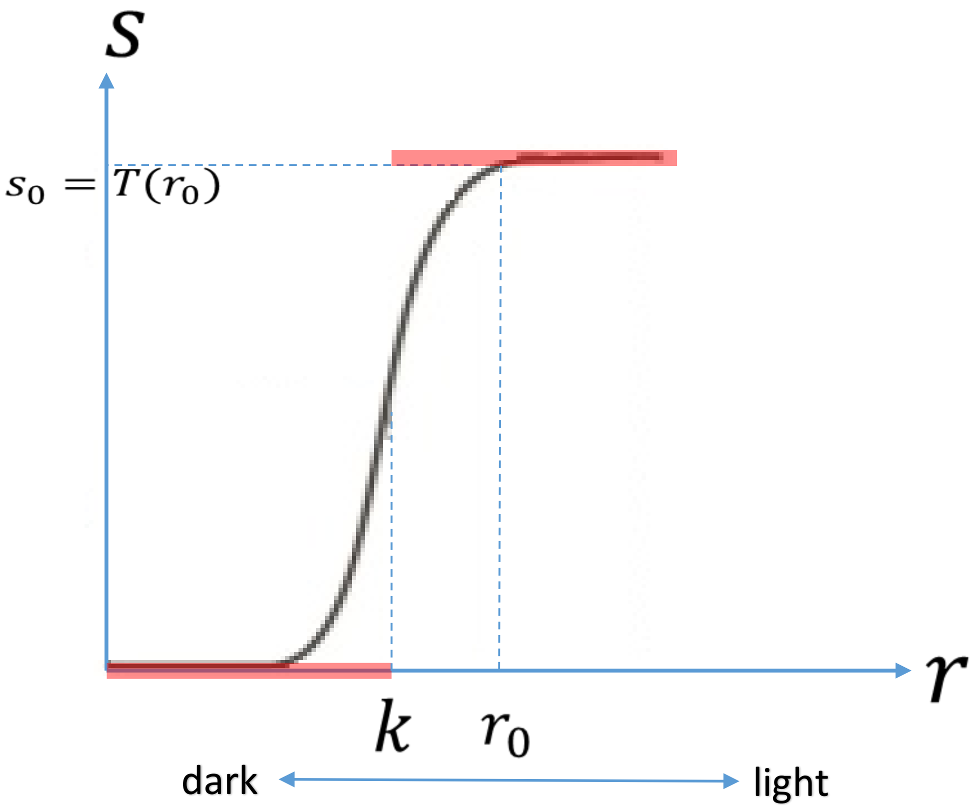
- We want to somehow smooth the middle to stretch the contrast.



$$s = T(r)$$

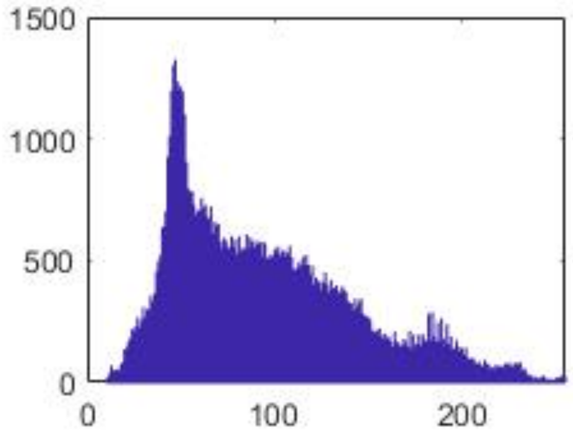
Stretching Transformations

- One way to define such a function is to use histograms equalization

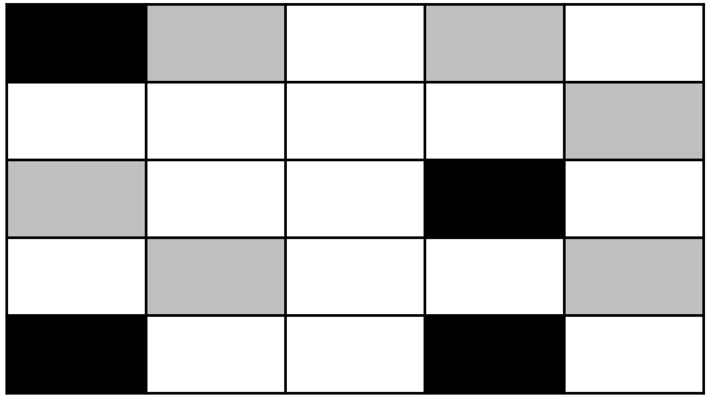


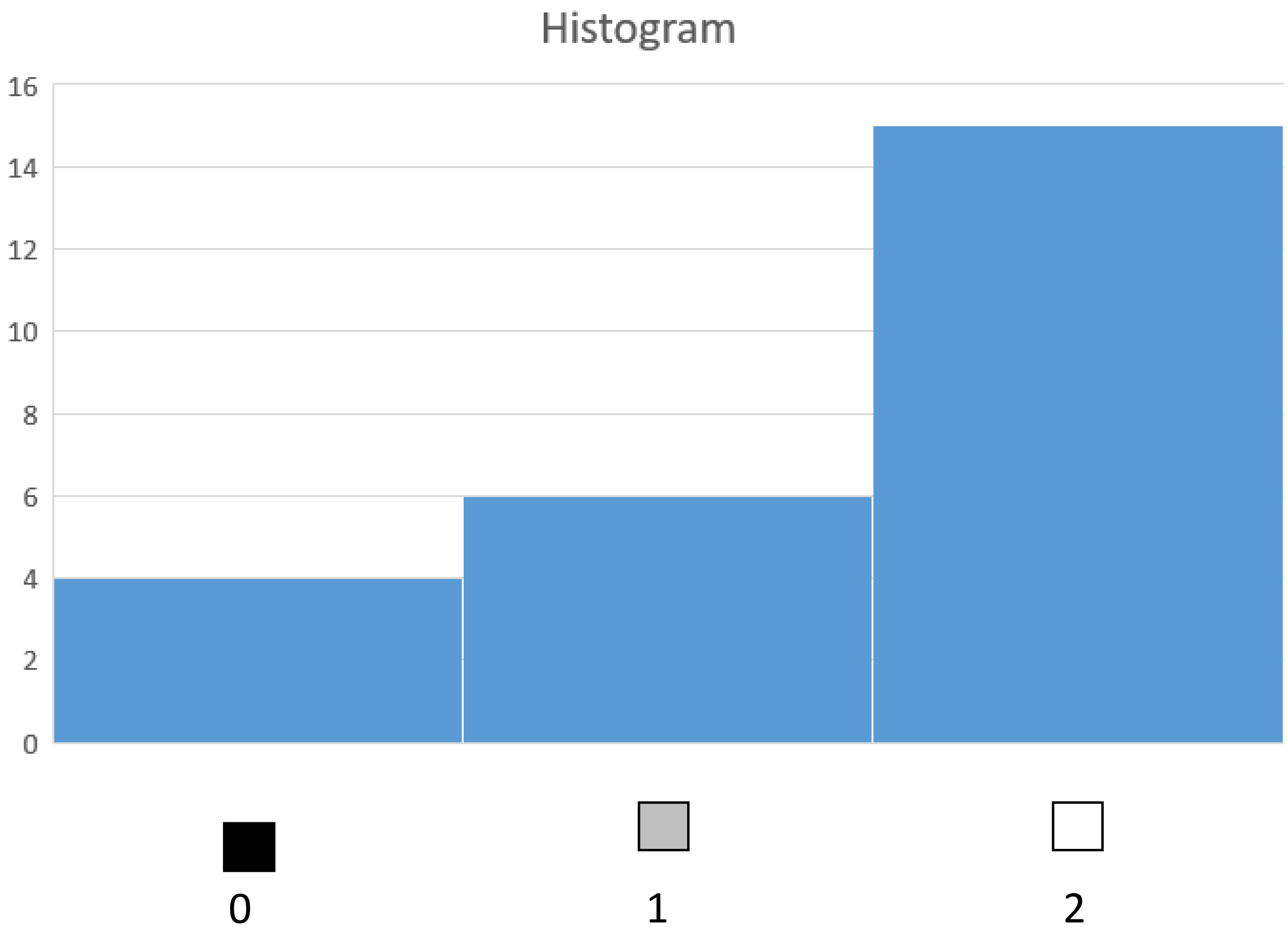
$$s = T(r)$$

Histogram



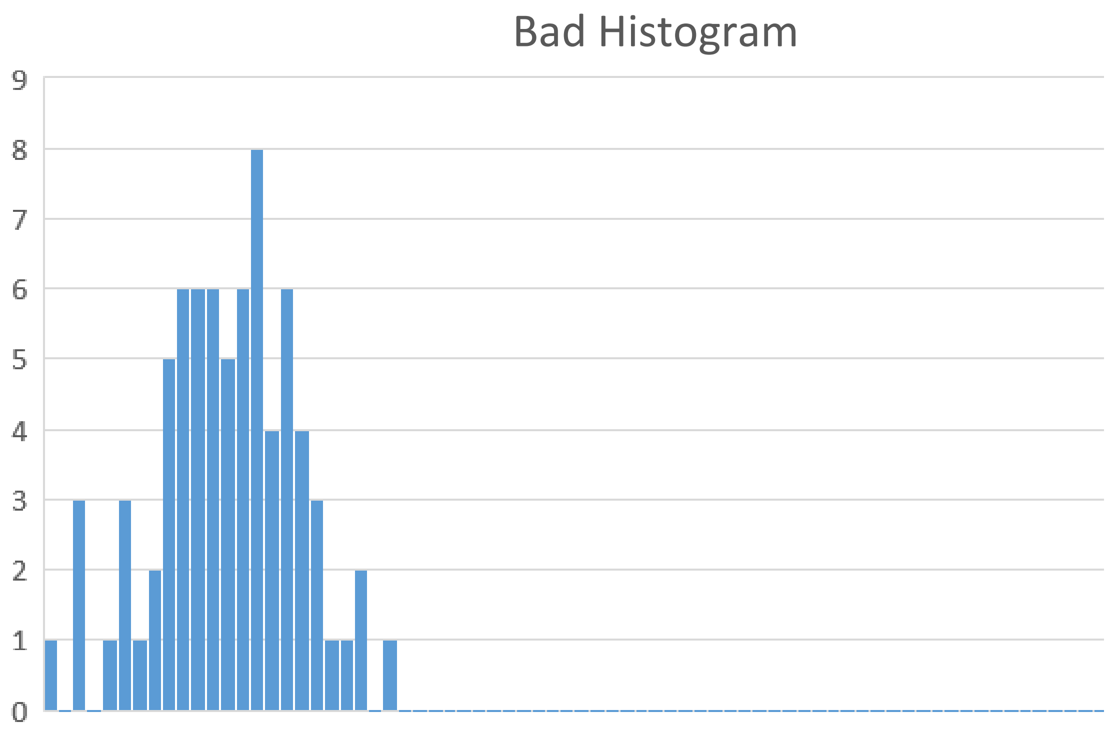
Histogram

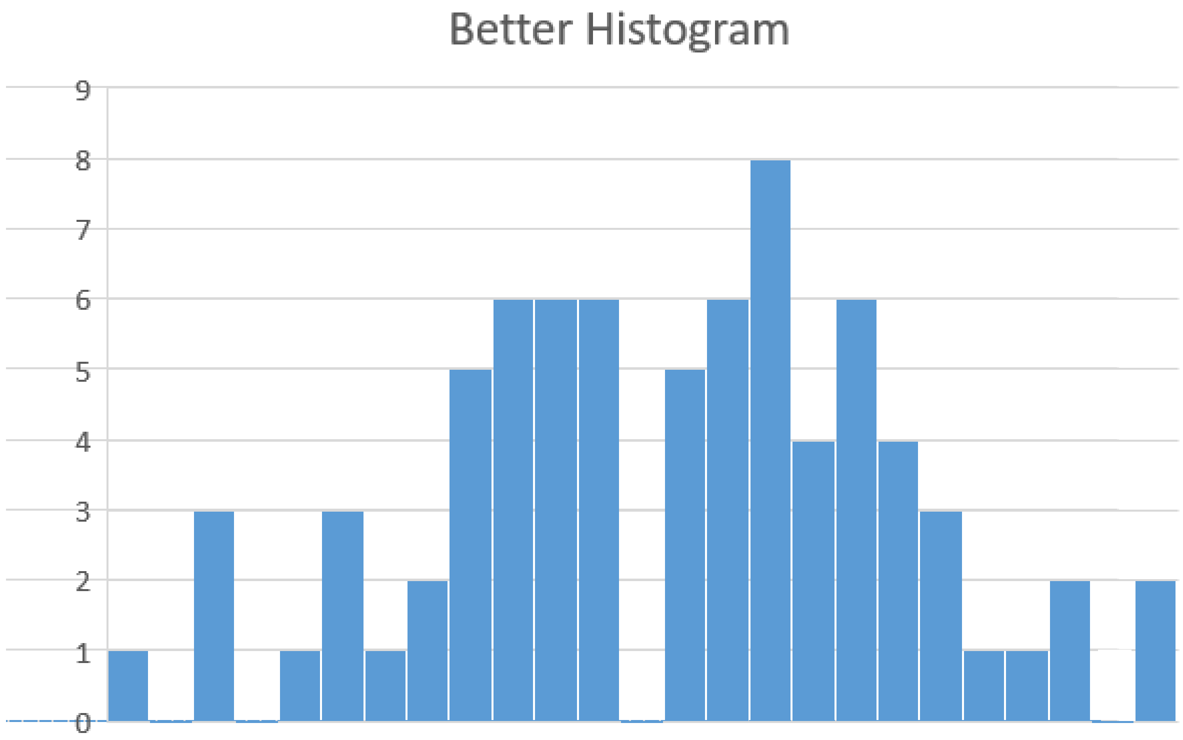




Histogram Equalization

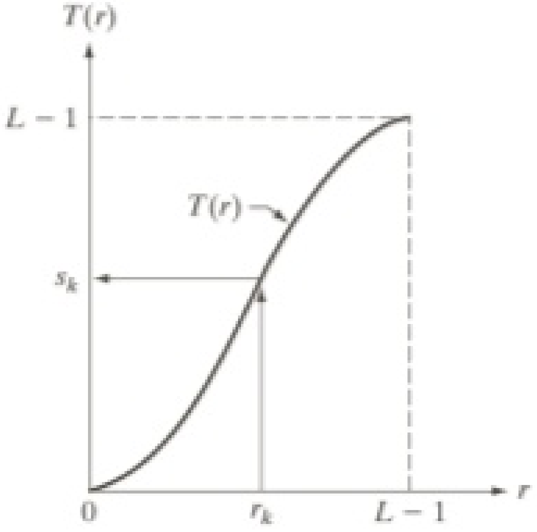
- We are looking for a transformation to map the histogram of an image to a well distributed histogram.





Histogram Equalization

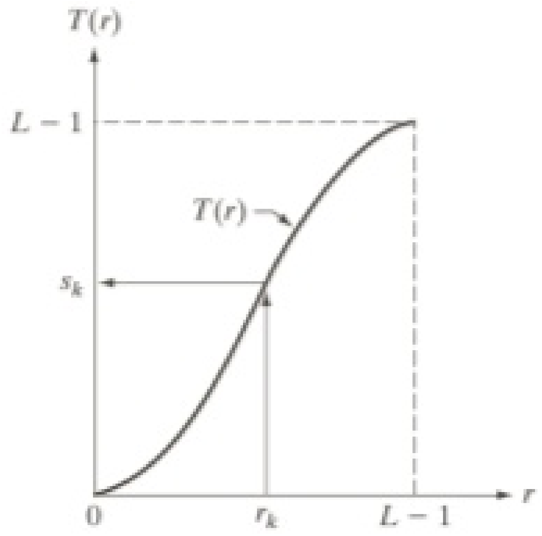
- Function should be monotonically increasing (why?)



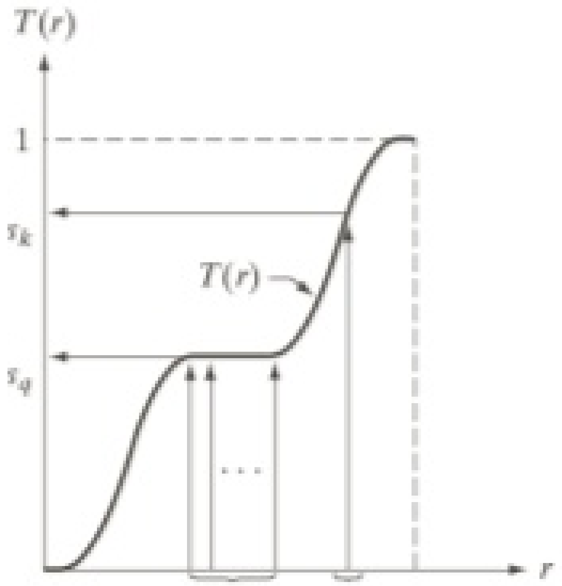
Strictly monotonically increasing

Histogram Equalization

- Function should be monotonically increasing (why?)

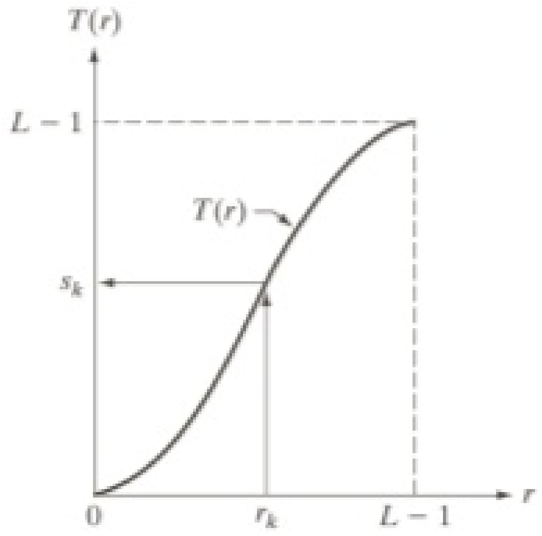


Strictly monotonically increasing

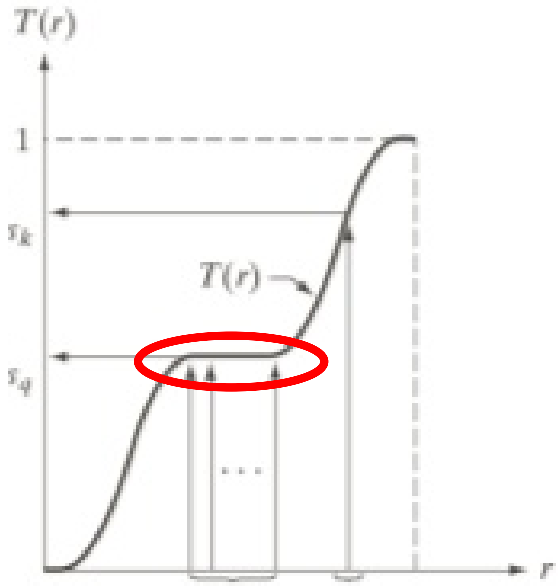


Histogram Equalization

- Function should be monotonically increasing (why?)

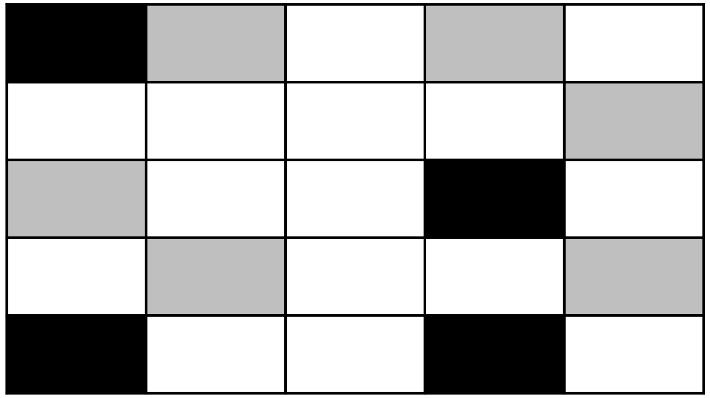


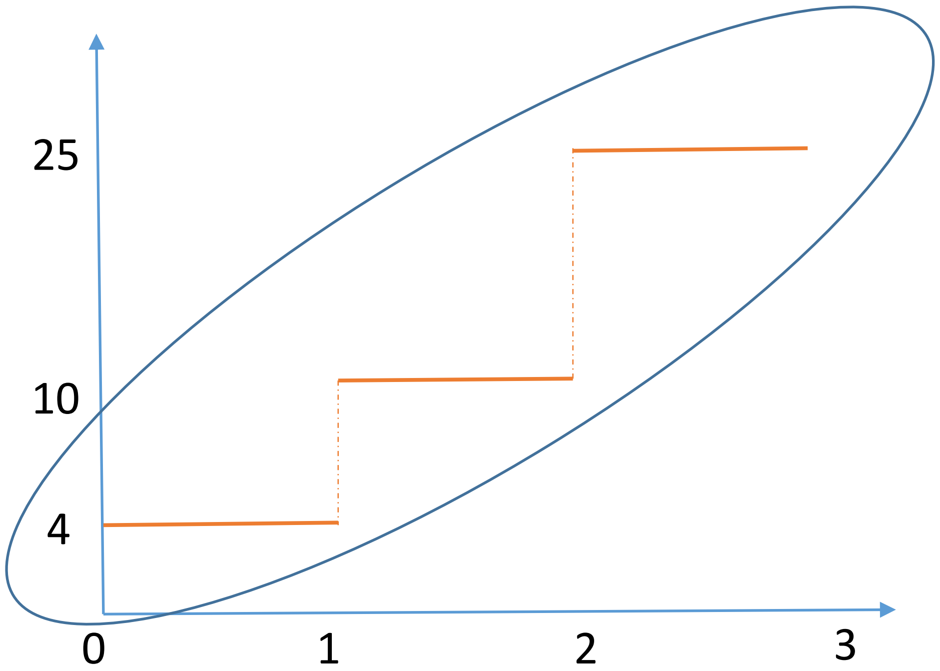
Strictly monotonically increasing



Histogram Equalization

- One way to define such a function is to use **cumulative distribution function** or cdf.

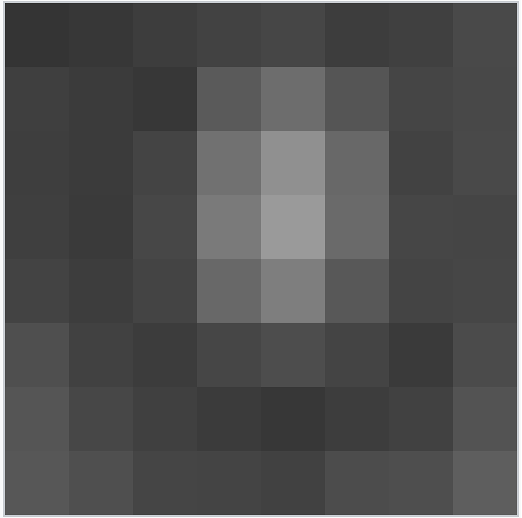




Monotonically increasing

Histogram Equalization

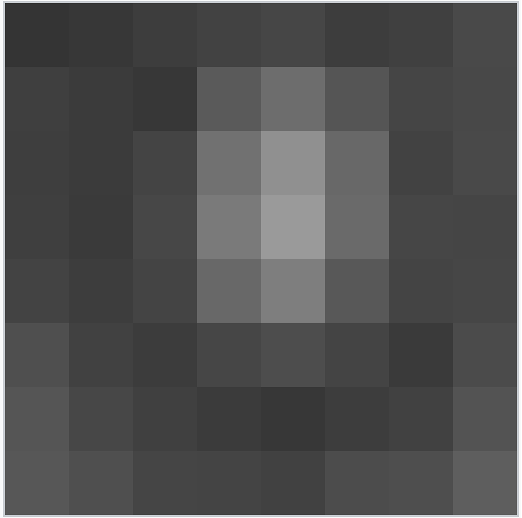
- Example



$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

Histogram Equalization

- Example

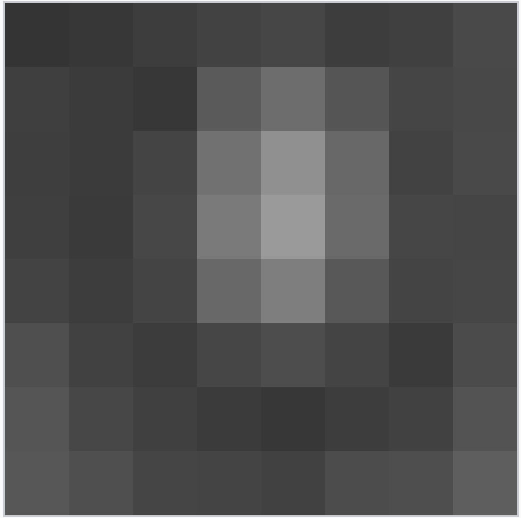


Smallest value

$$\begin{bmatrix} \boxed{52} & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

Histogram Equalization

- Example



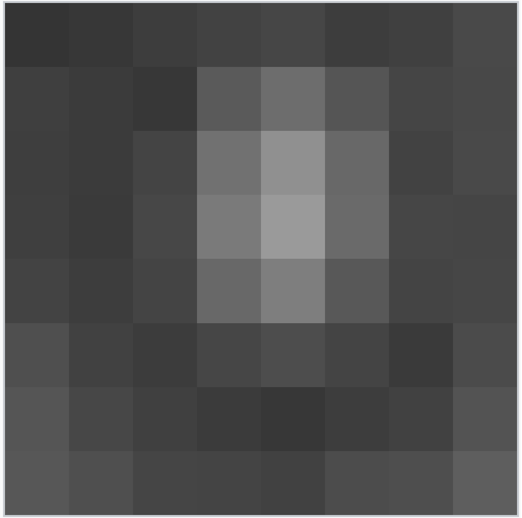
Smallest value

$$\begin{bmatrix} \boxed{52} & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

r	cdf
52	1

Histogram Equalization

- Example



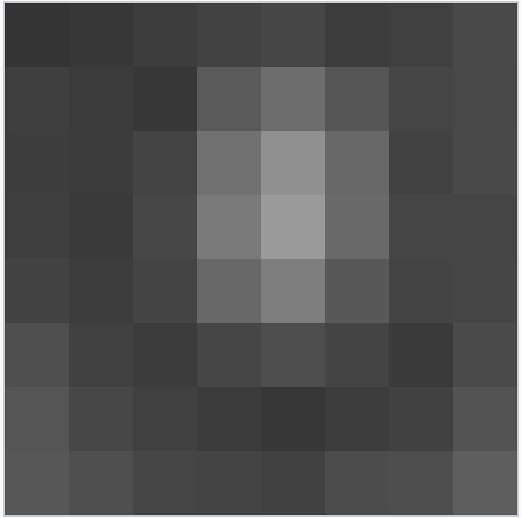
Next Smallest value

$$\begin{bmatrix} 52 & \boxed{55} & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & \boxed{55} & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & \boxed{55} & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

value	cdf
52	1
55	3+1=4

Histogram Equalization

- Example



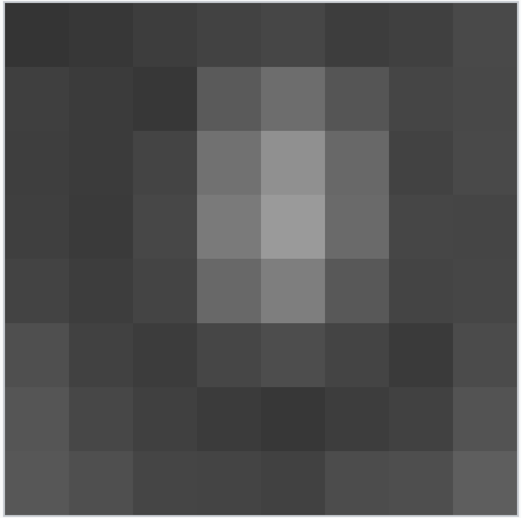
Next Smallest value

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & \boxed{58} & 75 \\ 69 & 85 & 64 & \boxed{58} & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

r	cdf
52	1
55	4
58	4+2=6

Histogram Equalization

- Example



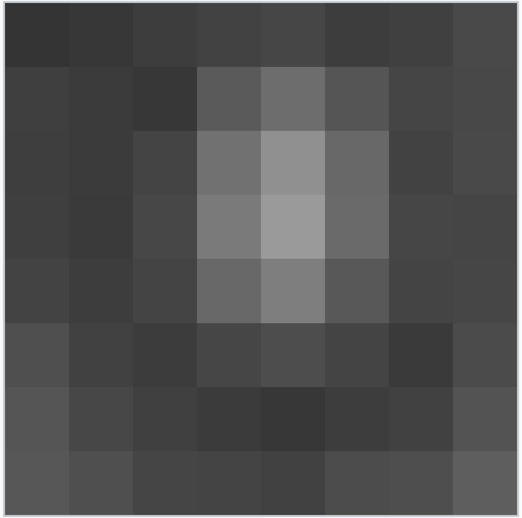
Next Smallest value

$$\begin{bmatrix} 52 & 55 & 61 & \boxed{59} & 70 & 61 & 76 & 61 \\ 62 & \boxed{59} & 55 & 104 & 94 & 85 & \boxed{59} & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

r	cdf
52	1
55	4
58	6
59	6+3=9

Histogram Equalization

- Example



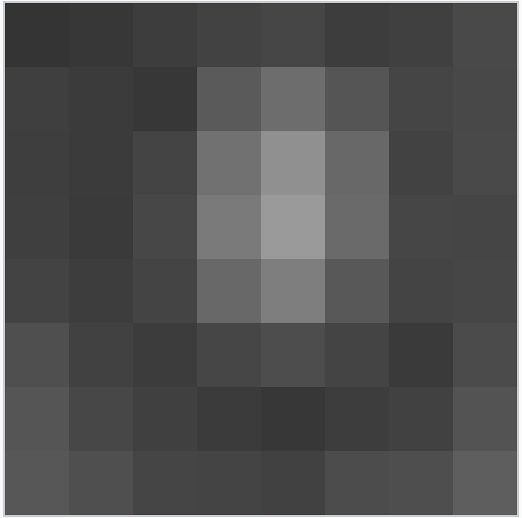
Largest value

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & \boxed{154} & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

Histogram Equalization

- Example



Largest value

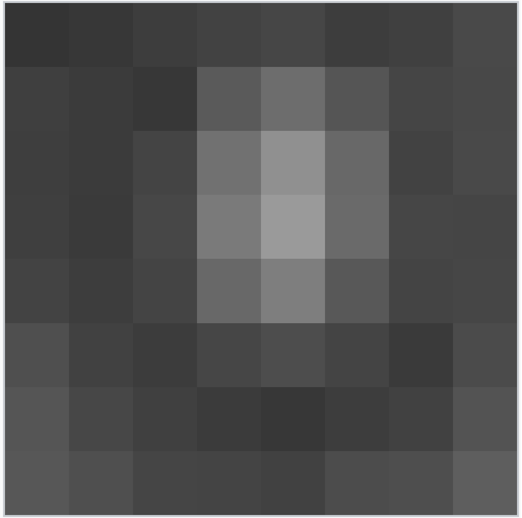
$$
\begin{bmatrix}
52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\
62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\
63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\
64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\
67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\
68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\
69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\
70 & 87 & 69 & 68 & 65 & 73 & 78 & 90
\end{bmatrix}
$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example



Largest value

$$\begin{bmatrix} 52 & 55 & 61 & 59 & 70 & 61 & 76 & 61 \\ 62 & 59 & 55 & 104 & 94 & 85 & 59 & 71 \\ 63 & 65 & 66 & 113 & 144 & 104 & 63 & 72 \\ 64 & 70 & 70 & 126 & 154 & 109 & 71 & 69 \\ 67 & 73 & 68 & 106 & 122 & 88 & 68 & 68 \\ 68 & 79 & 60 & 79 & 77 & 66 & 58 & 75 \\ 69 & 85 & 64 & 58 & 55 & 61 & 65 & 83 \\ 70 & 87 & 69 & 68 & 65 & 73 & 78 & 90 \end{bmatrix}$$

r	cdf
52	1
55	4
58	6
59	9
⋮	
144	63
154	64

s
0
⋮
255

We get help from cdf

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1) \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

We get help from cdf

Histogram Equalization

- Example

cdf of a pixel with value r

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

We get help from cdf

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(\underbrace{M}_{\text{height}} \times \underbrace{N}_{\text{width}}) - cdf_{min}} \times (L-1) \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

We get help from cdf

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right\rfloor$$

Probability distribution of pixels

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example

To map pixel with minimum intensity to 0

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right\rfloor$$

To map pixel with maximum intensity to L-1
Why?

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1) \right\rfloor$$

$$v: 58 \rightarrow \left\lfloor \frac{cdf(58) - cdf_{min}}{8 \times 8 - cdf_{min}} \times 255 \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1) \right\rfloor$$

$$v: 58 \rightarrow \left\lfloor \frac{cdf(58) - cdf_{min}}{8 \times 8 - cdf_{min}} \times 255 \right\rfloor = \left\lfloor \frac{6-1}{64-1} \times 255 \right\rfloor$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
. . .
255

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1) \right\rceil$$

$$v: 58 \rightarrow \left\lfloor \frac{cdf(58) - cdf_{min}}{8 \times 8 - cdf_{min}} \times 255 \right\rceil = \left\lfloor \frac{6-1}{64-1} \times 255 \right\rceil = \lfloor 20.2380 \rceil = 20$$

r	cdf
52	1
55	4
58	6
59	9
. . .	
144	63
154	64

s
0
20
. . .
255

Histogram Equalization

- Example

$$s = \left\lfloor \frac{cdf(r) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1) \right\rfloor$$

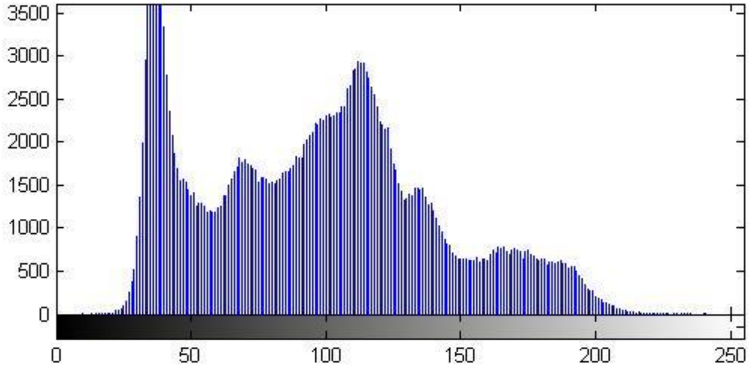
r	cdf
52	1
55	4
58	6
59	9
⋮	
144	63
154	64

s
0
12
20
32
⋮
251
255

Following the same formula produces a better distribution for pixel values

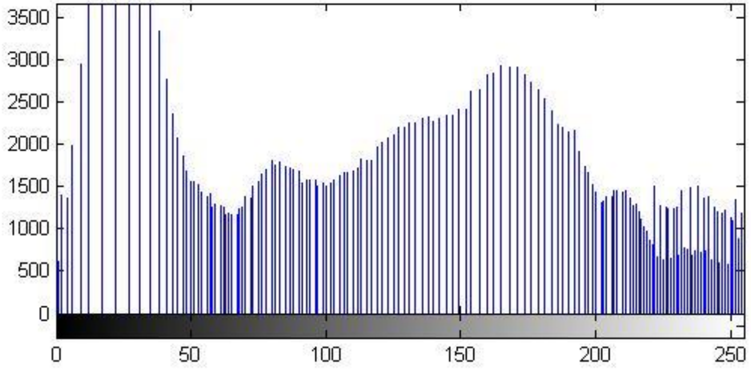
Histogram Equalization

Original Image



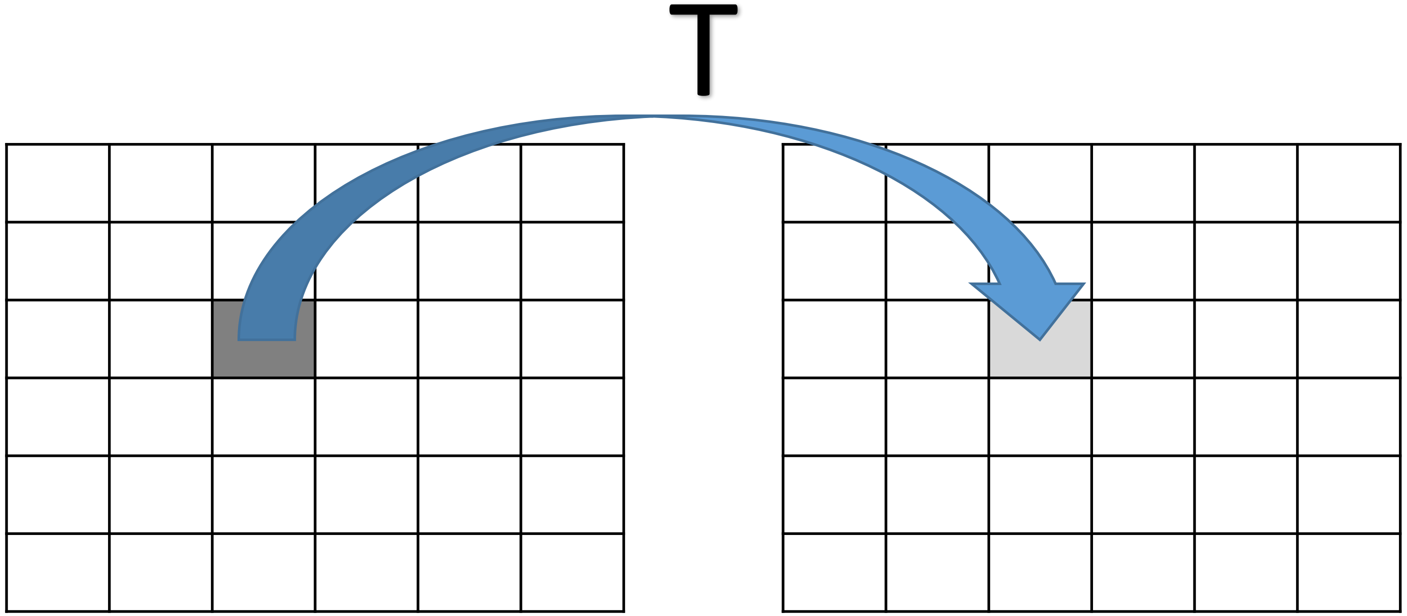
Enhanced Image

Histogram equalization



Local Neighbourhood Operations

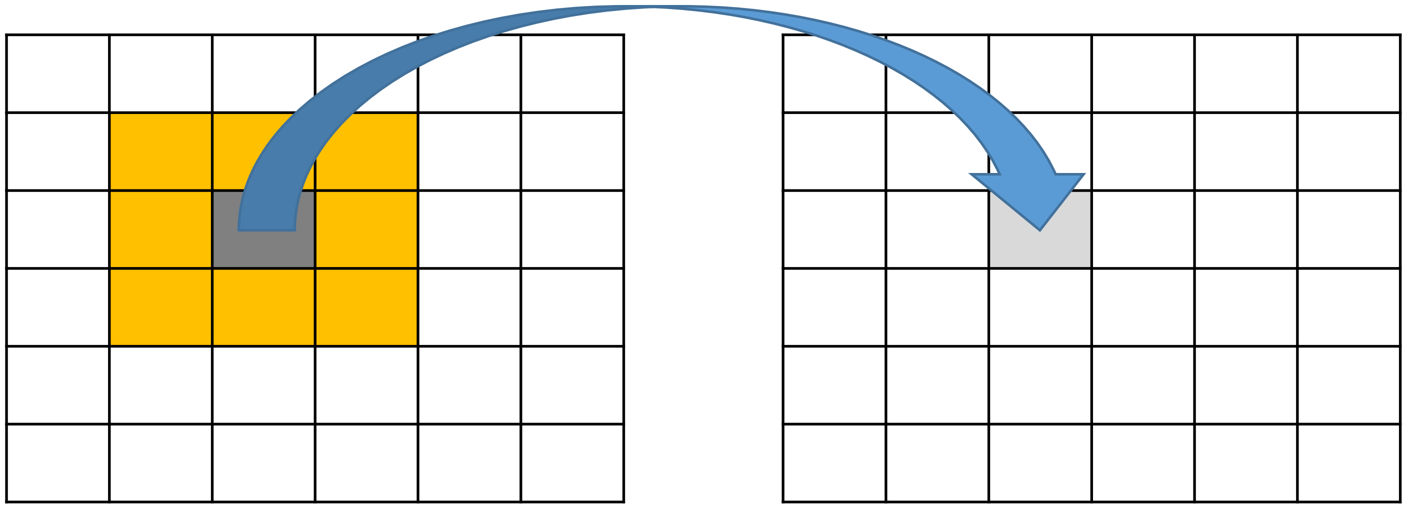
- So far, the modification was only based on the value of pixels
- We did not consider neighbouring pixels



Local Neighbourhood Operations

- Define a transformation via a local mask

w_0	w_1	w_2
w_3	w_4	w_5
w_6	w_7	w_8

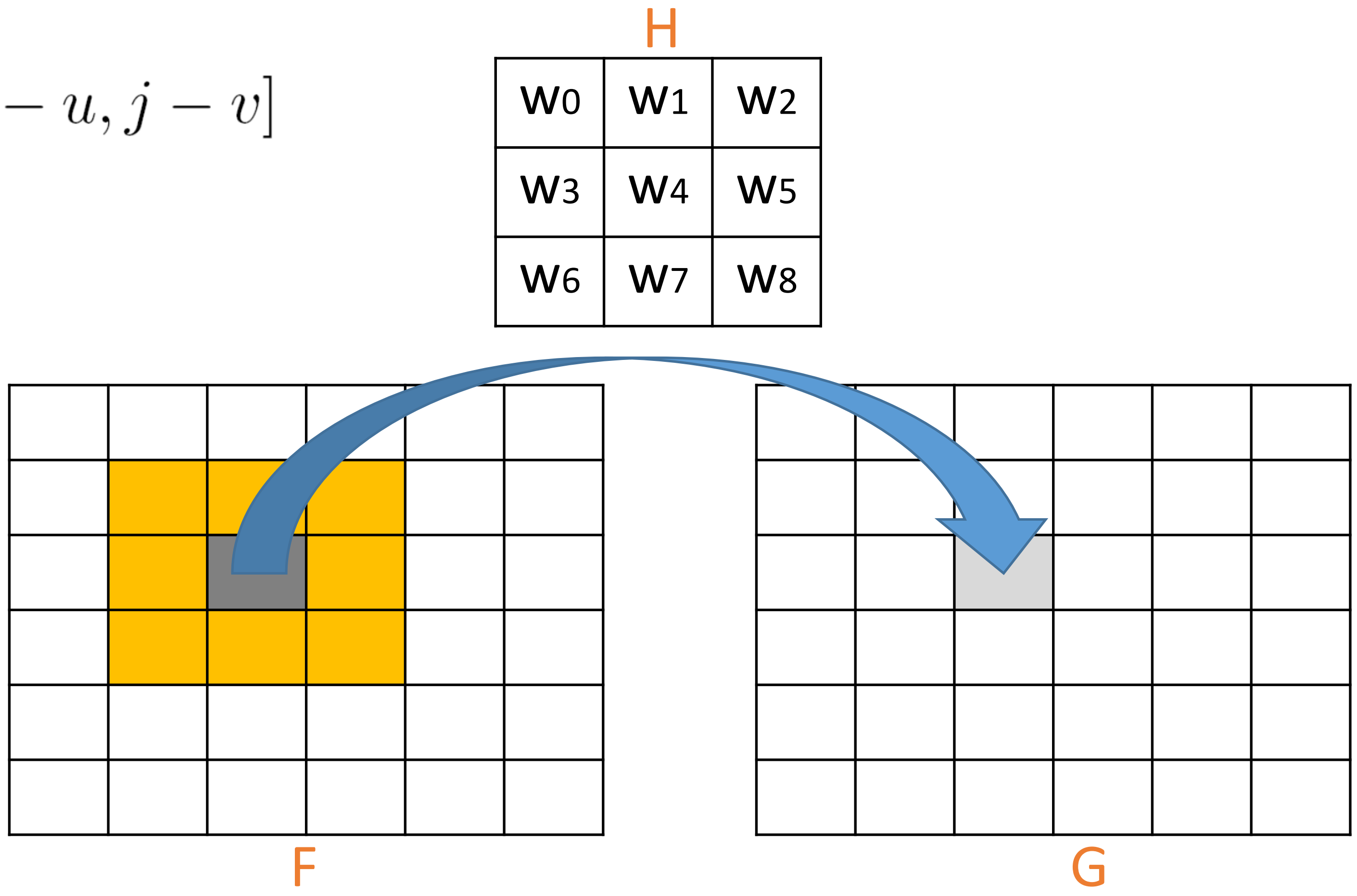


Local Neighbourhood Operations

- Convolution

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i-u,j-v]$$

$$G = H * F$$



Local Neighbourhood Operations

- What does this filter do?

Original

$$*$$

0	0	0
0	1	0
0	0	0

Local Neighbourhood Operations

- Identity

Original

*

0	0	0
0	1	0
0	0	0

=

Identical image

Local Neighbourhood Operations

- What does this filter do?

Original

$*$

0	0	0
1	0	0
0	0	0

Local Neighbourhood Operations

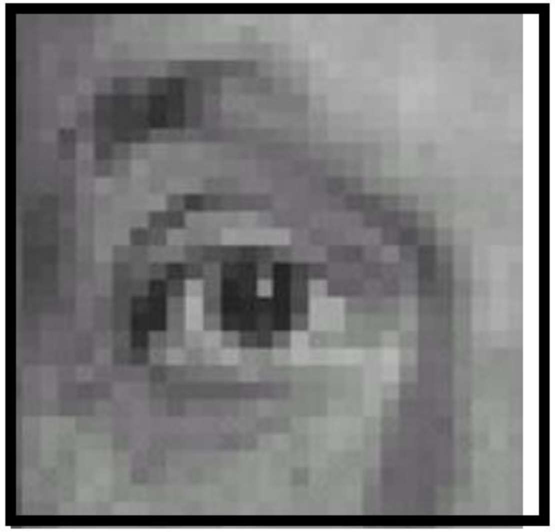
- Shift to left by one pixel.

Original

$*$

0	0	0
1	0	0
0	0	0

$=$



Local Neighbourhood Operations

- What does this filter do?

Original

$$* \ \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Local Neighbourhood Operations

- Average (blur)

Original

$$* \ \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} =$$

Local Neighbourhood Operations

- Weighted Average (blur)

Original

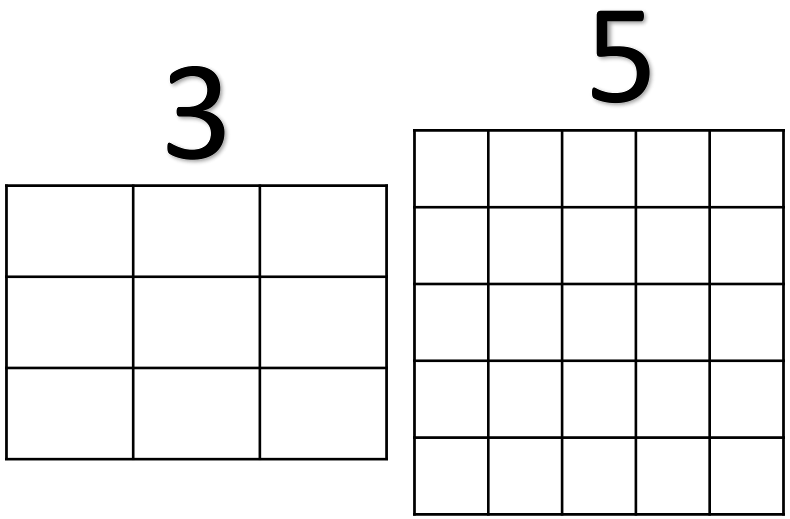
$*$

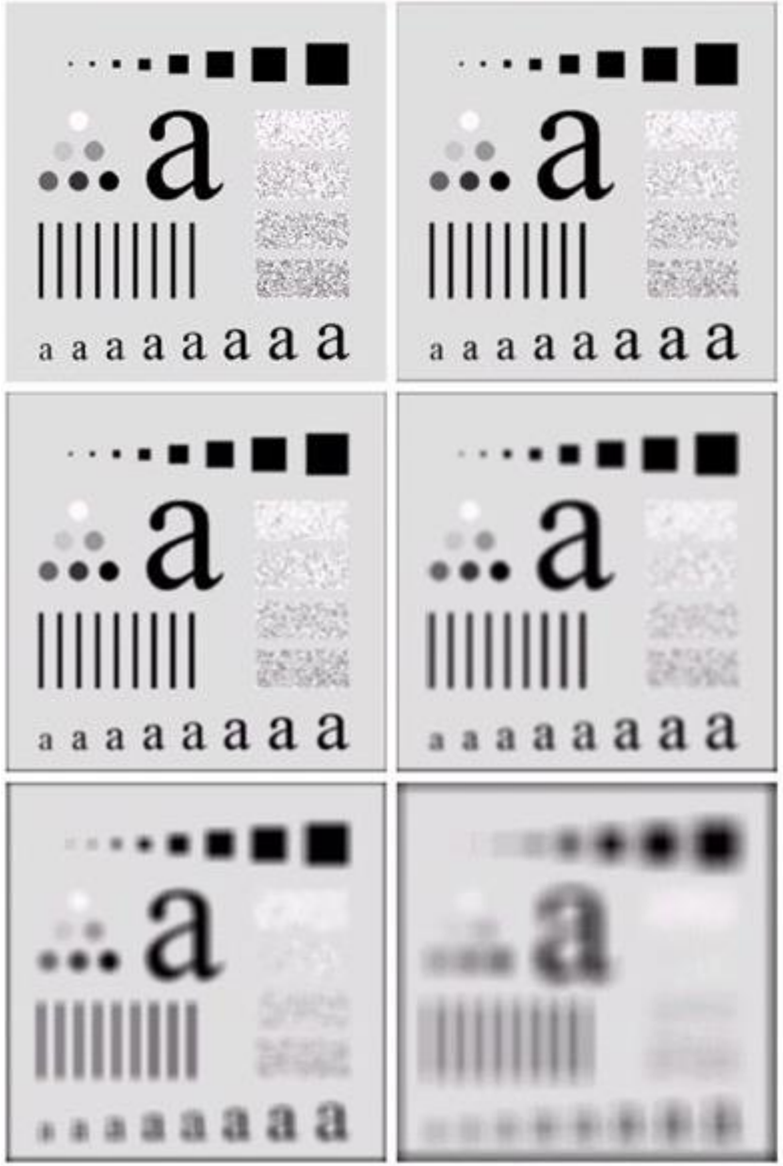
$\frac{1}{16}$	$\frac{8}{16}$	$\frac{1}{16}$
$\frac{1}{16}$	$\frac{8}{16}$	$\frac{1}{16}$
$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$

$=$

Local Neighbourhood Operations

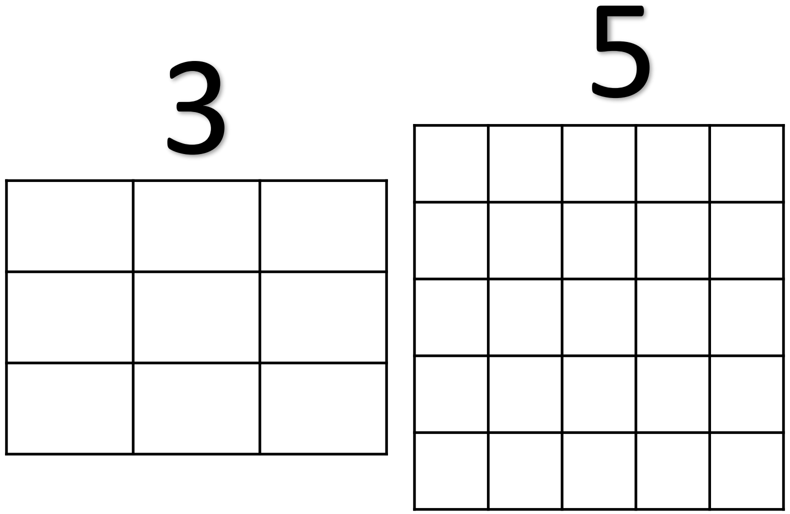
- Filter size



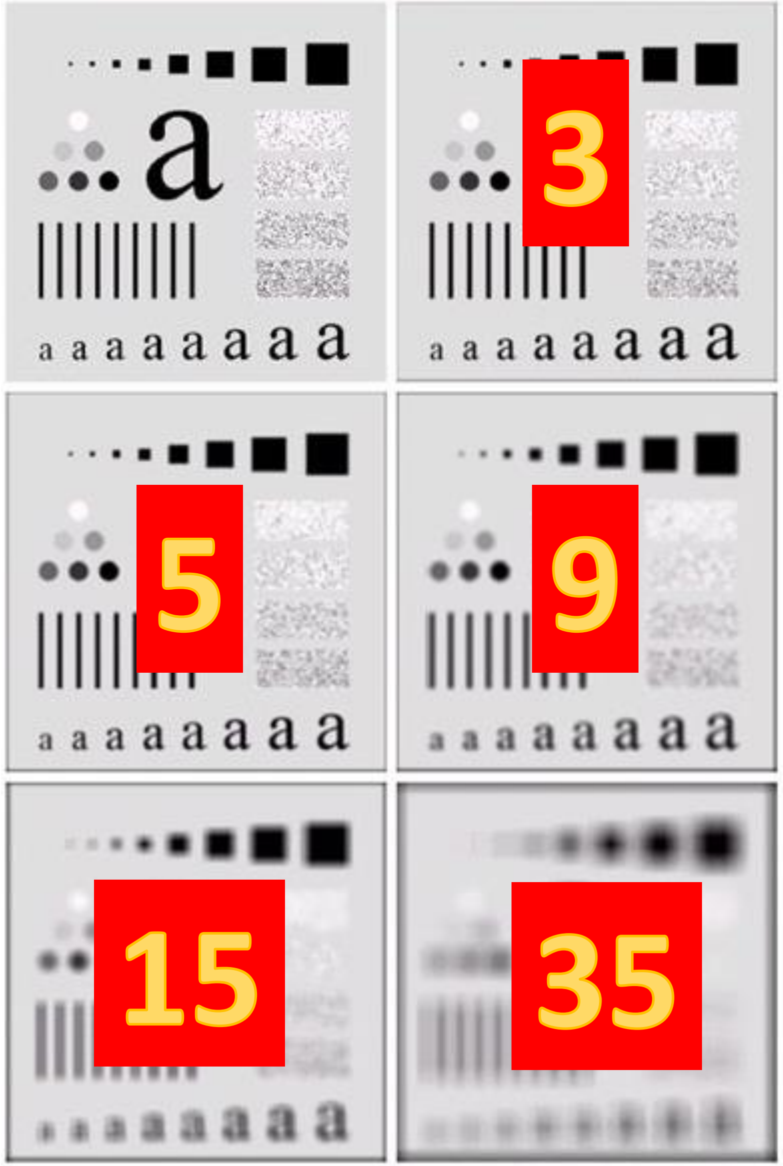


Local Neighbourhood Operations

- Filter size



Why do we get more blurriness with larger filter size?



Local Neighbourhood Operations

- Can we get sharpening artifacts using filters we learned?

Original

*

?

=

Local Neighbourhood Operations

- Sharpening

−

=

Local Neighbourhood Operations

- Adding more details will sharpen the image.
  - You can control the sharpness by alpha

$+ \alpha$

$=$

Local Neighbourhood Operations

- Adding more details will sharpen the image.
  - You can control the sharpness by alpha

$+ \alpha$

$=$

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Local Neighbourhood Operations

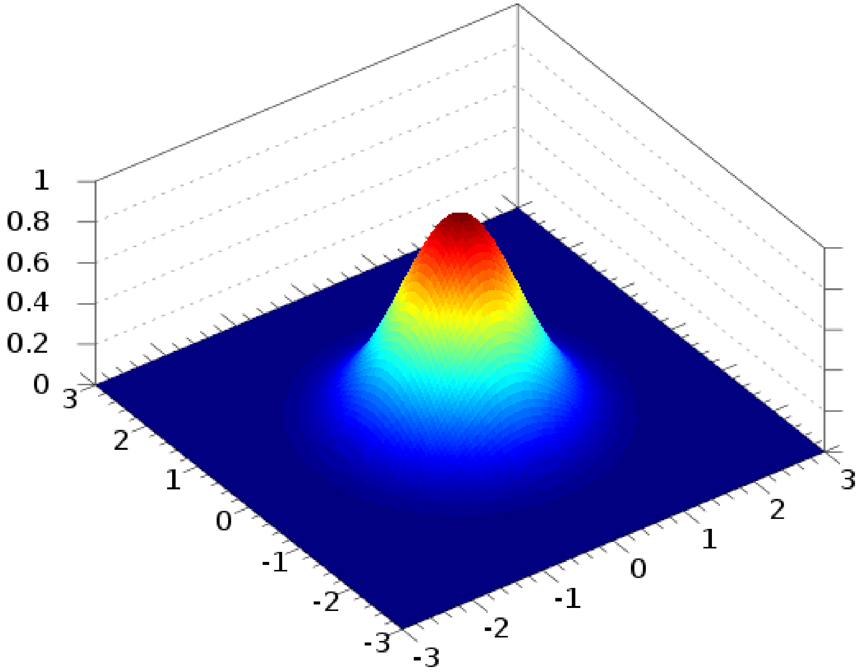
- Can we get sharpening artifacts using filters we learned?

Original

$$* \left(\begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \right) =$$

Local Neighbourhood Operations

- Another smoothing (blur) operation is Gaussian



$$f(x,y) = A\exp\left(-\left(\frac{(x-x_o)^2}{2\sigma_x^2} + \frac{(y-y_o)^2}{2\sigma_y^2}\right)\right)$$

Local Neighbourhood Operations

- If we use Gaussian as the convolution function, we can find a filter (convolution mask) by integration. (filter is not unique)

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Edge Detection

- Similar to finding the derivative
  - Why?

Edge Detection

- Similar to finding the derivative
  - First derivative

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y)$$

0	0	0
0	-1	1
0	0	0

Edge Detection

- Similar to finding the derivative
  - First derivative

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y)$$

0	0	0
0	-1	1
0	0	0

  - Second derivative

$$\frac{\partial^2 I}{\partial x^2} \approx I(x+1, y) - 2I(x, y) + I(x-1, y)$$

0	0	0
1	-2	1
0	0	0

Edge Detection

- Similar to finding the derivative
  - First derivative

$$\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y)$$

0	0	0
0	-1	1
0	0	0

  - Second derivative

$$\frac{\partial^2 I}{\partial x^2} \approx I(x+1, y) - 2I(x, y) + I(x-1, y)$$

0	0	0
1	-2	1
0	0	0

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{\partial I(x,y)}{\partial x} - \frac{\partial I(x+1,y)}{\partial x} = I(x+1,y) - I(x,y) - \big(I(x,y) - I(x-1,y)\big) = I(x+1,y) - 2I(x,y) + I(x-1,y)$$

Edge Detection

- Similar to finding the derivative
  - Derivative on both axes (Laplacian)

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \approx I(x+1,y) - 2I(x,y) + I(x-1,y) + I(x,y+1) - 2I(x,y) + I(x,y-1)$$

0	1	0
1	-4	1
0	1	0

Edge Detection

- Another version

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Edge Detection

Edge Detection

- Sobel operator

$$G_x = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$G = \sqrt{G_x^2 + G_y^2}$$

Or

$$G = |G_x| + |G_y|$$

faster

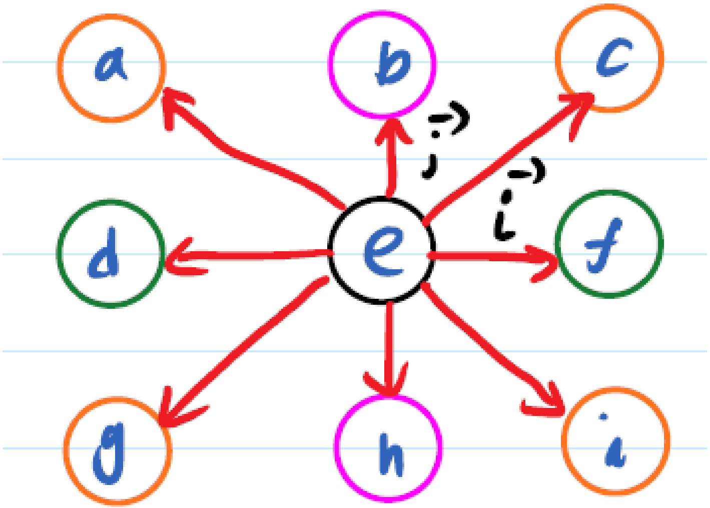
Edge Detection

- Sobel operator

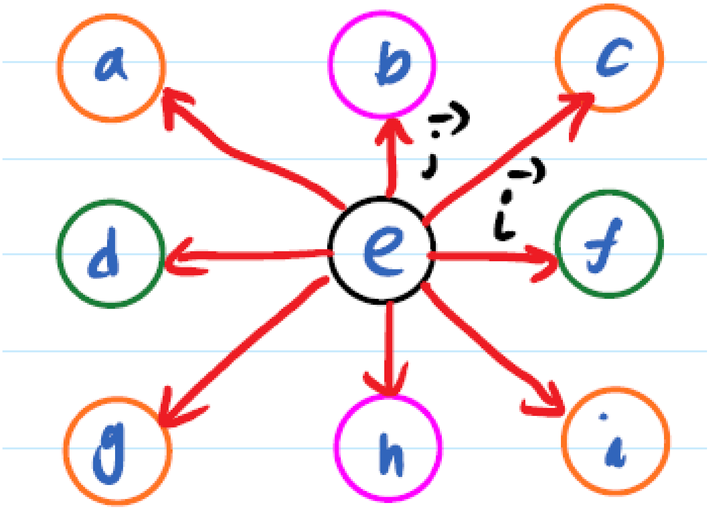
$$G_x = \frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

How did we find these filters?

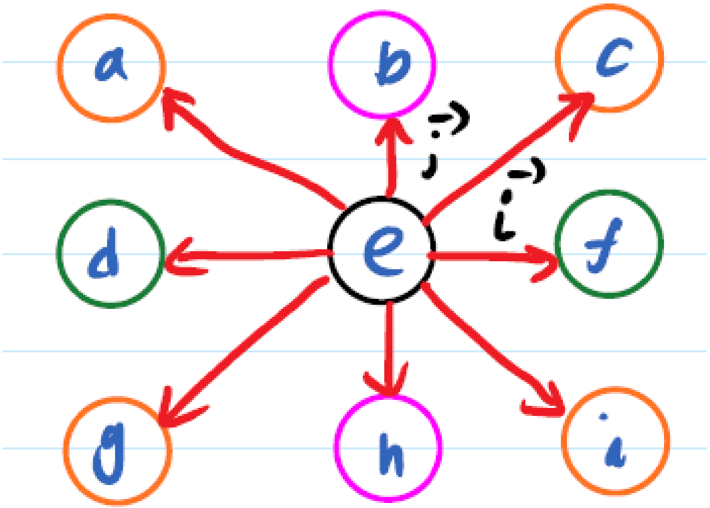


We take the derivative along these eight vectors



We take the derivative along these eight vectors

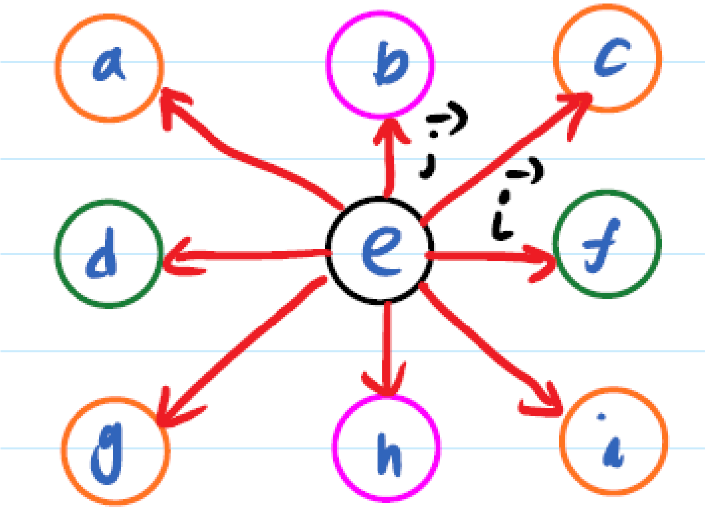
Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:



We take the derivative along these eight vectors

Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})$$

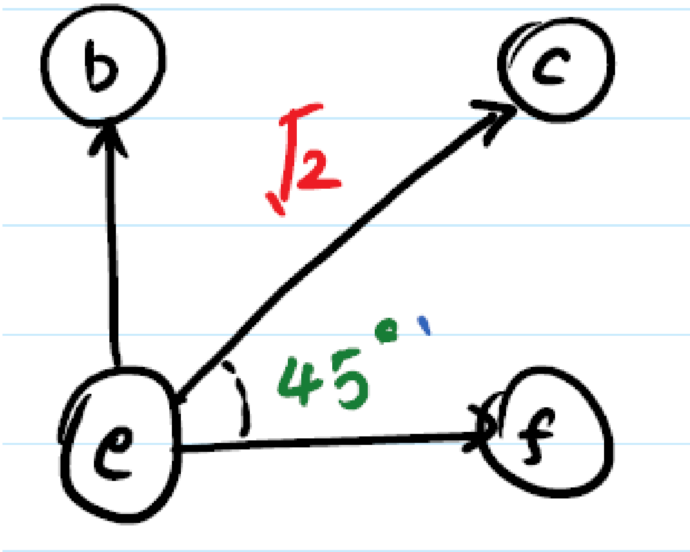


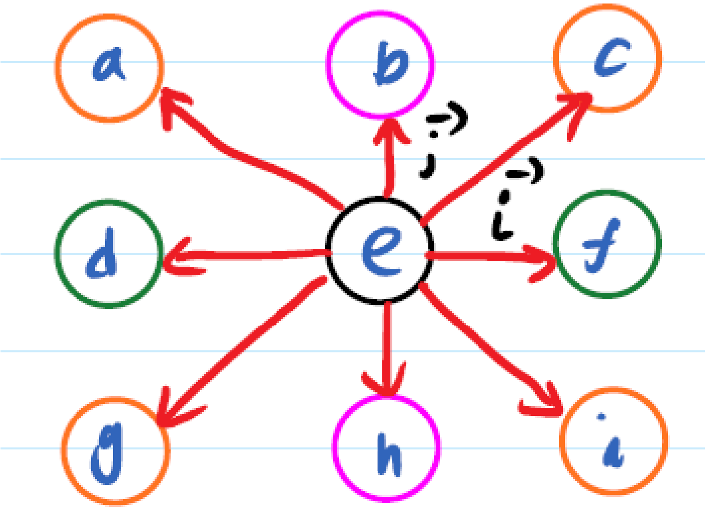
We take the derivative along these eight vectors

Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})$$

Let’s find the derivative along diagonals:





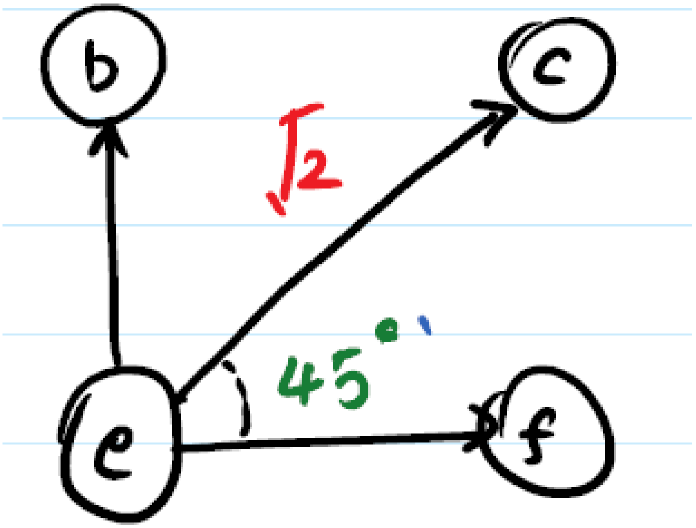
We take the derivative along these eight vectors

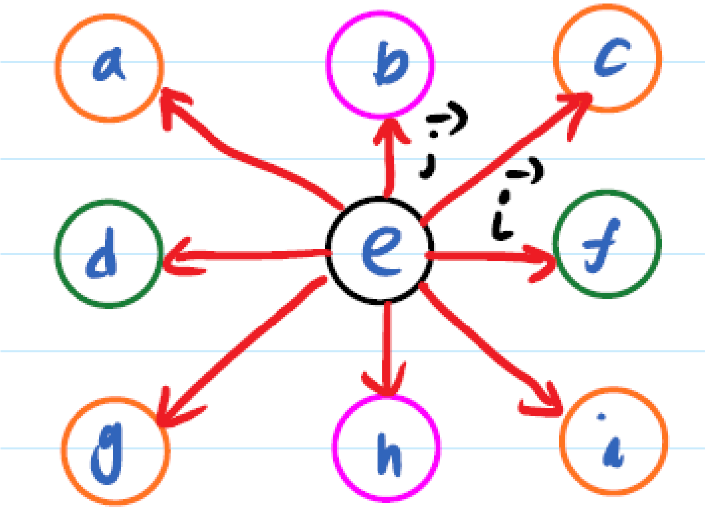
Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:

$$(f - e)\vec{\imath} + (\mathrm{d} - \mathrm{e})(-\vec{\imath}) + (b - e)\vec{\jmath} + (\mathrm{h} - \mathrm{e})(-\vec{\jmath})$$

Let's find the derivative along diagonals:

$$\frac{(c - e)}{\sqrt{2}}$$ Normalize the length





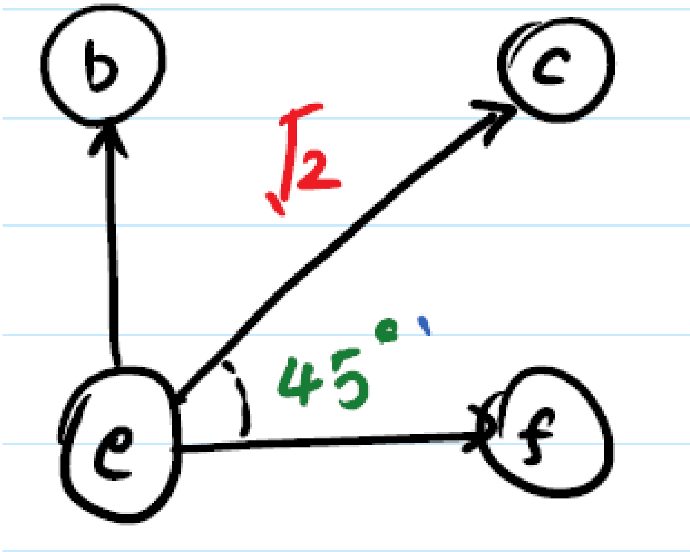
We take the derivative along these eight vectors

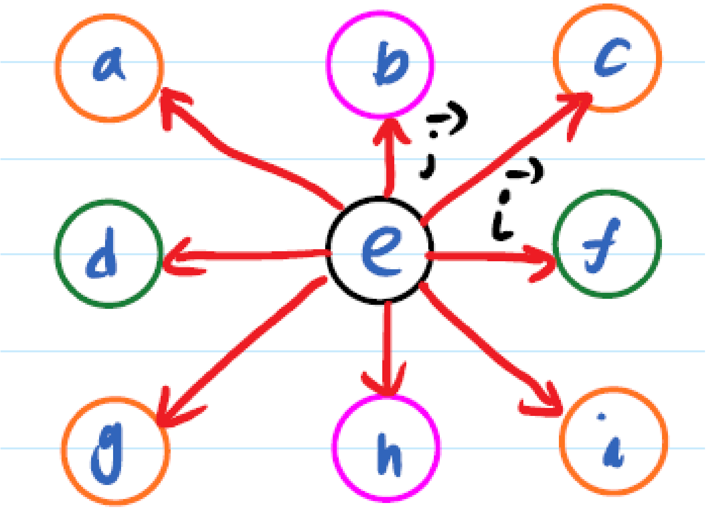
Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})$$

Let's find the derivative along diagonals:

$$\frac{(c-e)}{\sqrt{2}}$$ Normalize the length, project on $\vec{\imath}$ and $\vec{\jmath}$



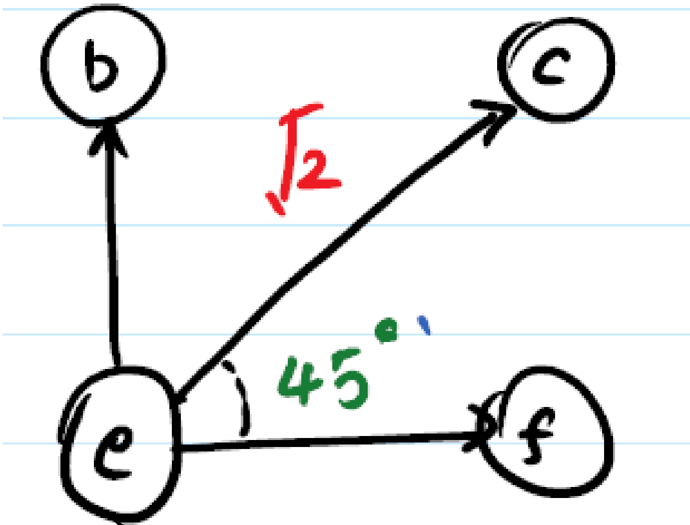


We take the derivative along these eight vectors

Derivatives along $\vec{\imath}$ and $\vec{\jmath}$ are easy:

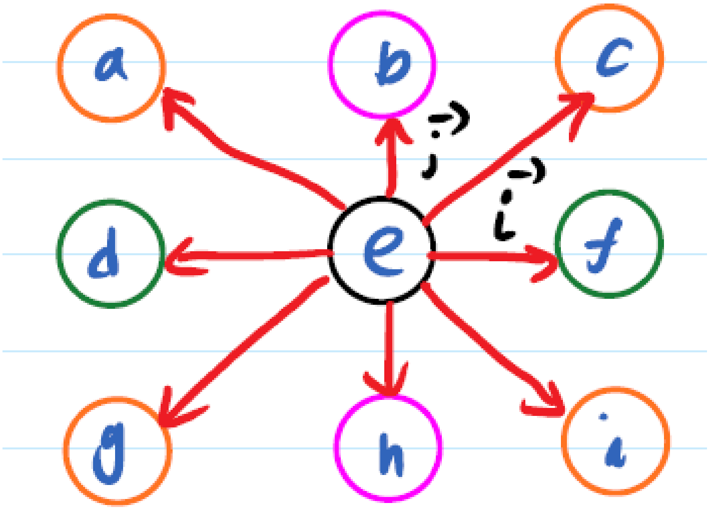
$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})$$

Let's find the derivative along diagonals:



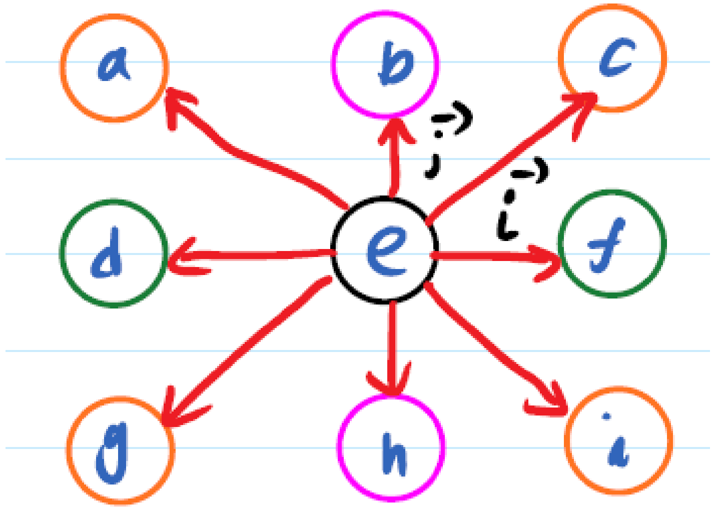
$\dfrac{(c-e)}{\sqrt{2}}$ Normalize the length, project on $\vec{\imath}$ and $\vec{\jmath}$

$$\frac{(c-e)}{\sqrt{2}}\cos(45)\,\vec{\imath}+\frac{(c-e)}{\sqrt{2}}\sin(45)\,\vec{\jmath}=\frac{1}{2}(c-e)(\vec{\imath}+\vec{\jmath})$$



Similarly, we find all the diagonal derivatives

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})+$$
$$\frac{1}{2}(c-e)(\vec{\imath}+\vec{\jmath})+\frac{1}{2}(a-e)(\vec{\jmath}-\vec{\imath})+\frac{1}{2}(g-e)(-\vec{\jmath}-\vec{\imath})+\frac{1}{2}(i-e)(\vec{\imath}-\vec{\jmath})$$



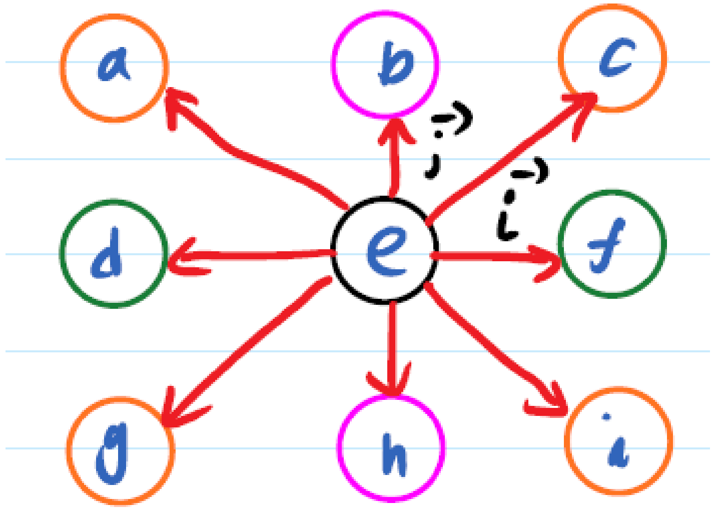
Add all them up

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})+$$

$$\frac{1}{2}(c-e)(\vec{\imath}+\vec{\jmath})+\frac{1}{2}(a-e)(\vec{\jmath}-\vec{\imath})+\frac{1}{2}(g-e)(-\vec{\jmath}-\vec{\imath})+\frac{1}{2}(i-e)(\vec{\imath}-\vec{\jmath})$$

$$=\left(f-d+\frac{1}{2}c-\frac{1}{2}a-\frac{1}{2}g+\frac{1}{2}i\right)\vec{\imath}$$

$$+\left(b-h+\frac{1}{2}a-\frac{1}{2}g-\frac{1}{2}i+\frac{1}{2}c\right)\vec{\jmath}$$



Correspond values to find filters

$$(f-e)\vec{\imath}+(\mathrm{d}-\mathrm{e})(-\vec{\imath})+(b-e)\vec{\jmath}+(\mathrm{h}-\mathrm{e})(-\vec{\jmath})+$$

$$\frac{1}{2}(c-e)(\vec{\imath}+\vec{\jmath})+\frac{1}{2}(a-e)(\vec{\jmath}-\vec{\imath})+\frac{1}{2}(g-e)(-\vec{\jmath}-\vec{\imath})+\frac{1}{2}(i-e)(\vec{\imath}-\vec{\jmath})$$

$$=\left(f-d+\frac{1}{2}c-\frac{1}{2}a-\frac{1}{2}g+\frac{1}{2}i\right)\vec{\imath}$$

$$+\left(b-h+\frac{1}{2}a-\frac{1}{2}g-\frac{1}{2}i+\frac{1}{2}c\right)\vec{\jmath}$$

$\frac{-1}{2}$	0	$\frac{1}{2}$
-1	0	1
$\frac{-1}{2}$	0	$\frac{1}{2}$

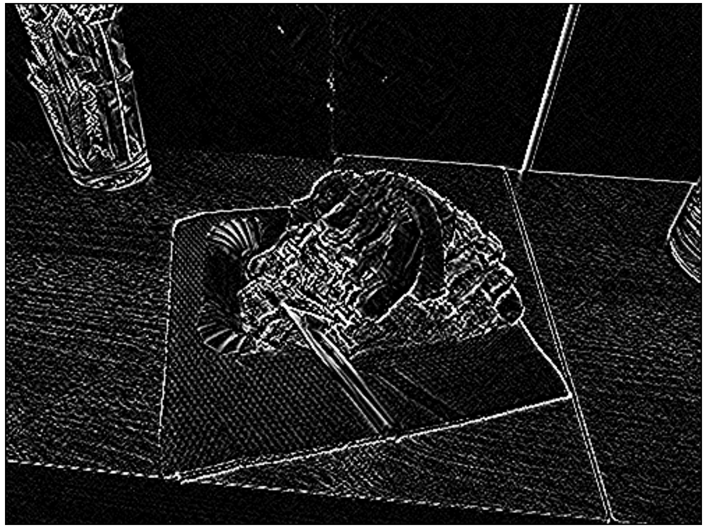
$\vec{\imath}$

$\frac{1}{2}$	1	$\frac{1}{2}$
0	0	0
$\frac{-1}{2}$	-1	$\frac{-1}{2}$

$\vec{\jmath}$

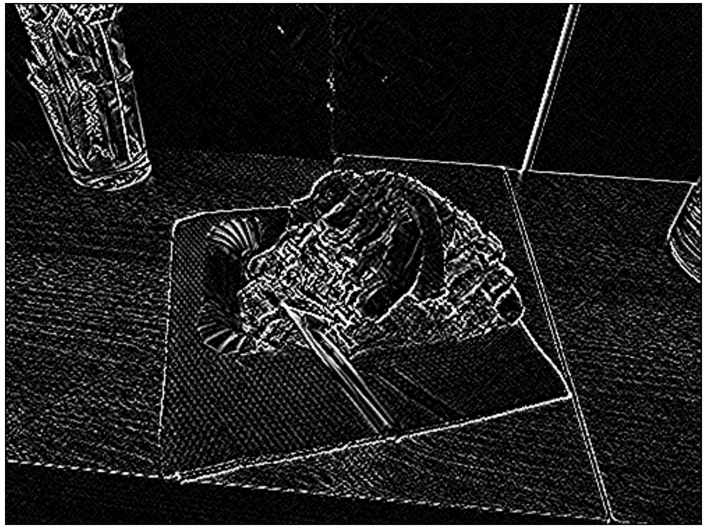
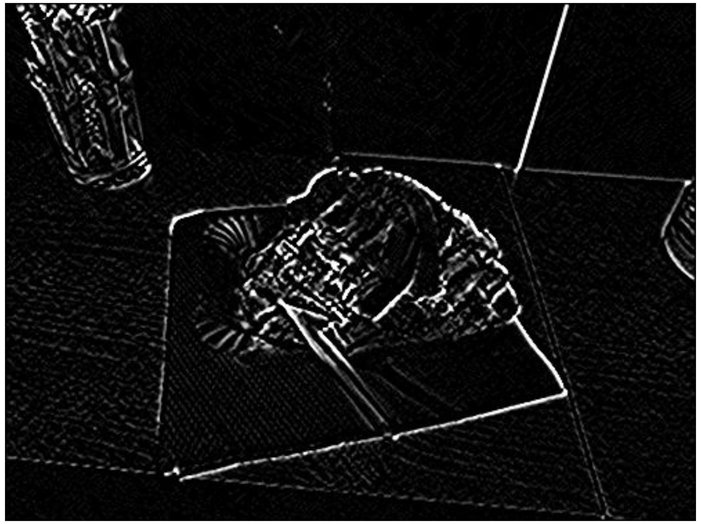
Edge Detection

- Sobel operator
  - One of the most common edge detection filters



Edge Detection

- Notice that there are many artifacts. To avoid these artifacts, we can first smooth the image by Gaussian and then apply the filter.

Edge Detection

- Sobel operator is itself a combination of a smoothing and an edge detection. How?

Edge Detection

- Sobel operator is itself a combination of a smoothing and an edge detection. How?

- hint: vertical smoothing (weighted average), horizontal edge detection (derivative)

-1	0	1
-2	0	2
-1	0	1

Edge Detection

- Sobel operator is itself a combination of a smoothing and an edge detection. How?
- hint: vertical smoothing, horizontal edge detection (derivative)

-1	0	1
-2	0	2
-1	0	1

1
2
1

-1	0	1

Edge Detection

- Sobel operator is itself a combination of a smoothing and an edge detection. How?

- hint: vertical smoothing, horizontal edge detection (derivative)

-1	0	1
-2	0	2
-1	0	1

1
2
1

-1	0	1

f(x+h)-f(x-h)

Image Features

- You can define many different convolution kernels (filters).

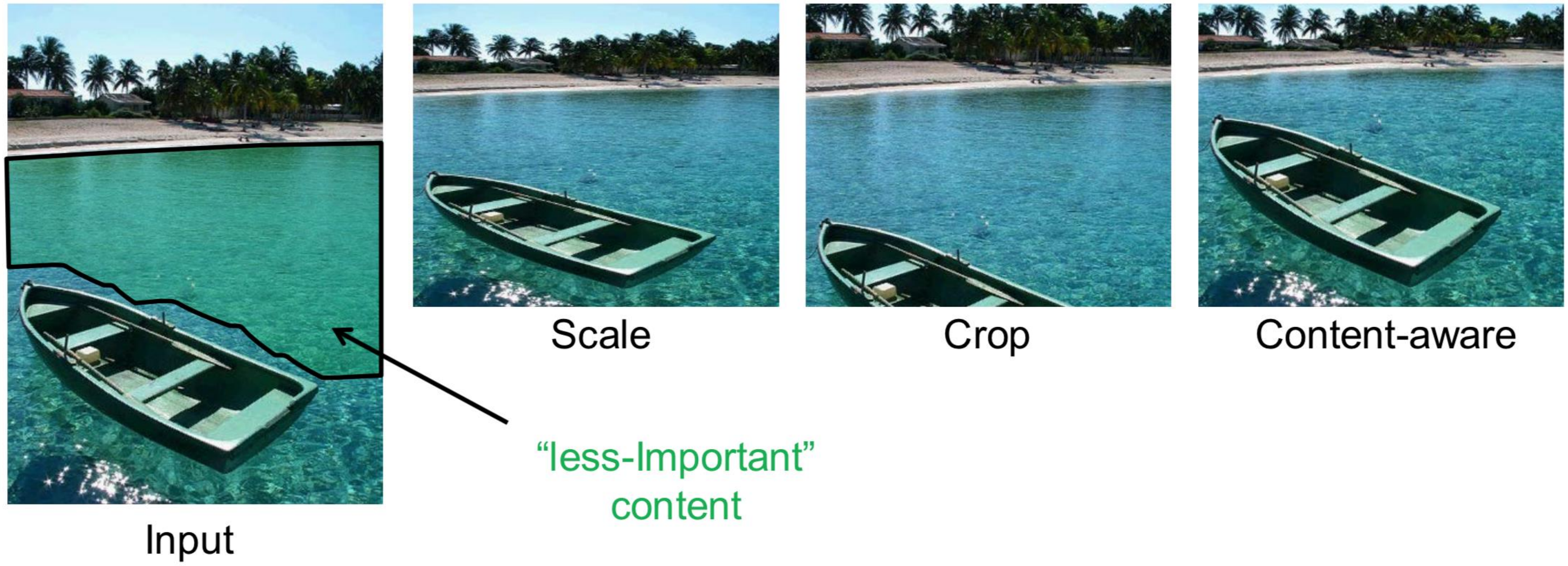
- Applying these filters may result a set of image features (e.g., corners, edges, etc), such filters are called feature detectors.

Image Features

- Find a list of feature detector filters and try to implement them in Matlab.

Application

- Image Resizing
  - An application of edge detection aside from Active Contours



Application

- How can we define “less important”?
  - The parts that have less visible edges or they have a lower energy

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

Application

- How can we define "less important"?
  - The parts that have less visible edges or they have a lower energy

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

reminder: $\frac{\partial I}{\partial x} \approx I(x+1, y) - I(x, y)$

Application

- We cannot remove single pixels

1	2	1	2
1	2	3	4
2	4	1	0
0	1	2	1
1	2	3	1

Application

- We cannot remove single pixels

How can we shift the pixels?

1	2	1	2
1	2	3	4
2	4	1	0
0	1	2	1
1	2	3	1

Application

- We can remove a column or row that has the smaller sum of energy

1	2	1	2
1	2	3	4
2	4	1	0
0	1	2	1
1	2	3	1

Application

- We can remove a column or row that has the smaller sum of energy

1	2	1	2
1	2	3	4
2	4	1	0
0	1	2	1
1	2	3	1
5	11	10	8

Application

- We can remove a column or row that has the smaller sum of energy

remove the column

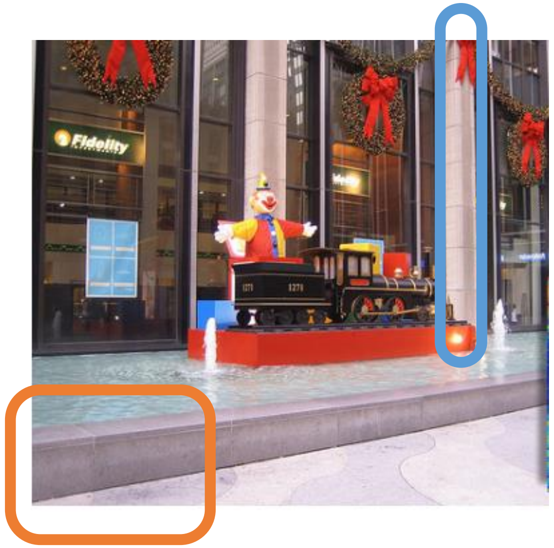
1	2	1	2
1	2	3	4
2	4	1	0
0	1	2	1
1	2	3	1
5	11	10	8

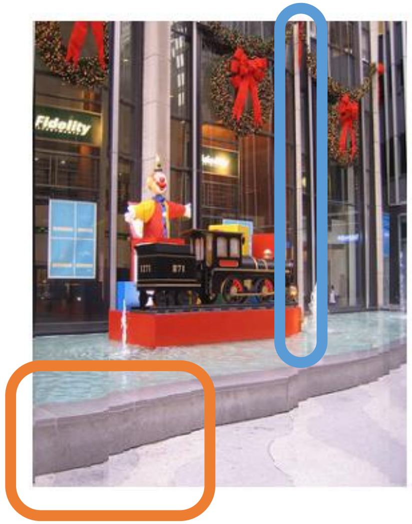
Application

- Problem with column removing
  - The chance of removing important features is high

Application

- Problem with column removing
  - The chance of removing important features is high





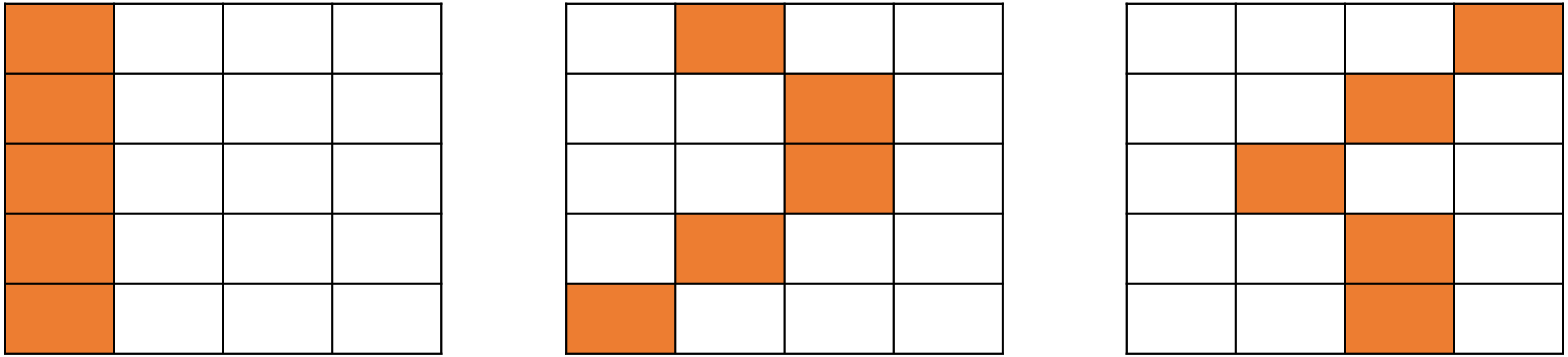
Application

- Instead of a column, we can remove a seam

Application

- How do you define a seam?
  - vertical seam: it has a pixel at each row; two consecutive rows have pixels with a difference with maximum one.

$$\mathbf{s}^{\mathbf{x}} = \{s_i^x\}_{i=1}^{n} = \{(x(i), i)\}_{i=1}^{n}, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$



Application

- How do you select a seam?
  - Dynamic programming

Application

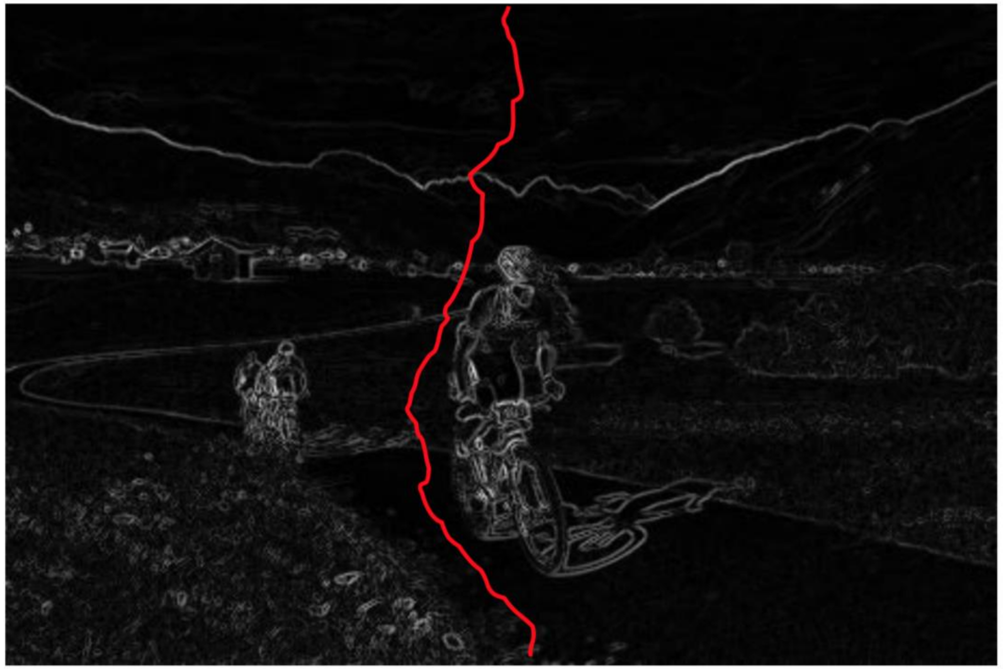
- How do you select a seam?
  - First find the energy of each pixel using discrete derivative

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

Application

- How do you select a seam?
  - First find the energy of each pixel using discrete derivative

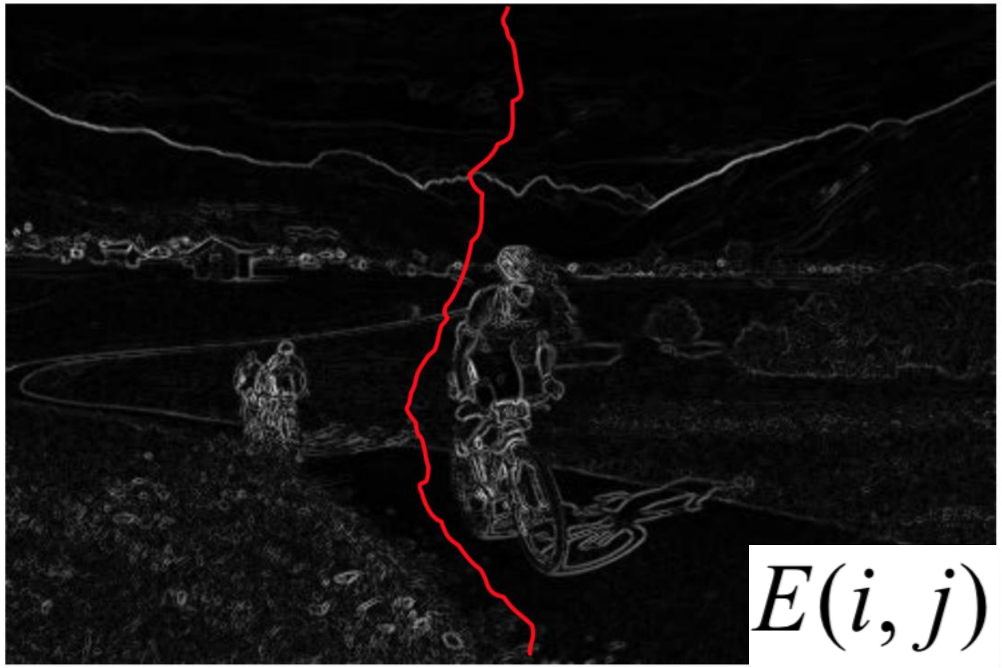
$$E(i, j)$$



Application

- How do you select a seam?
  - First find the energy of each pixel using discrete derivative
  - Then use dynamic programming with this recursive equation

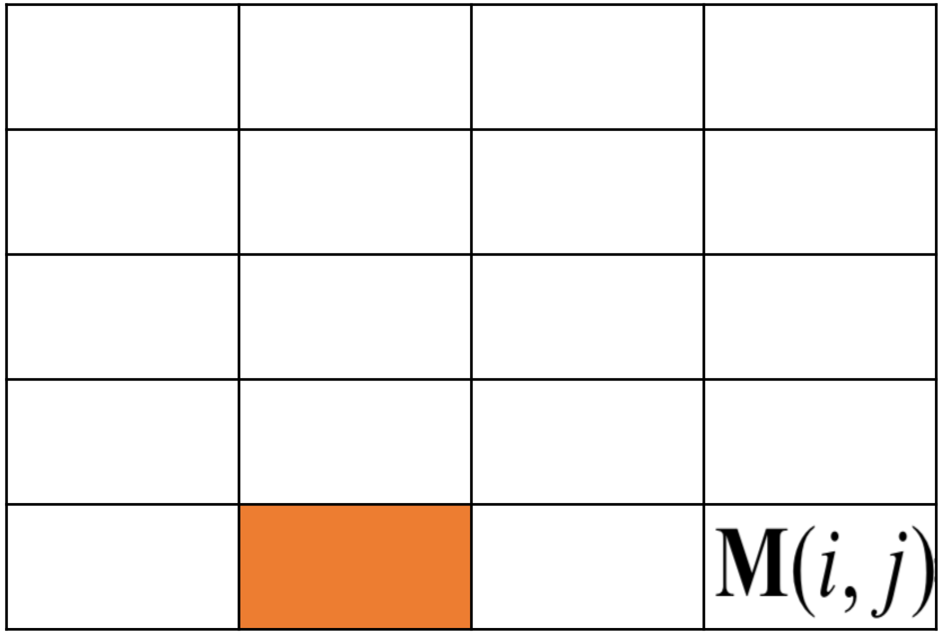
			$\mathbf{M}(i,j)$



$$\mathbf{M}(i,j) = E(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$

Application

- How do you select a seam?
  - First find the energy of each pixel using discrete derivative
  - Then use dynamic programming with this recursive equation

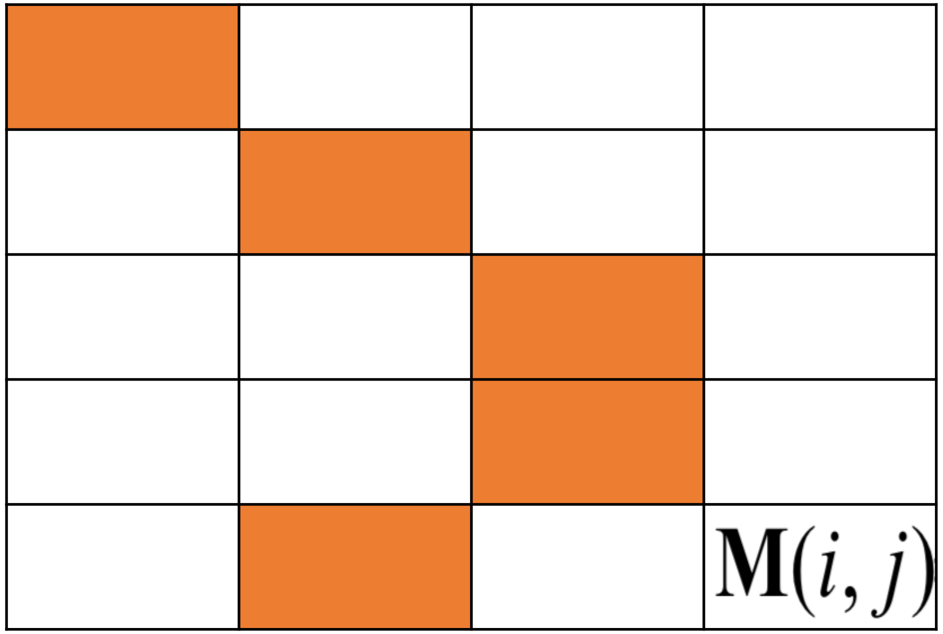


Find the pixel with smaller M(i,j) value
and backtrack to find the seam

$$\mathbf{M}(i, j) = E(i, j) + \min\left(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1)\right)$$

Application

- How do you select a seam?
  - First find the energy of each pixel using discrete derivative
  - Then use dynamic programming with this recursive equation



Find the pixel with smaller M(i,j) value and backtrack to find the seam

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)

E(i,j)

5	8	12	3
9	2	3	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\Big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\Big)$$

M(i,j)

E(i,j)

5	8	12	3
9	2	3	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	2	3	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\big(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\big)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	3	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)

E(i,j)

5	8	12	3
9	7	3	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	9
7	3	4	2
4	5	7	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)

E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	8

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	8

Example

$$\mathbf{M}(i, j) = E(i, j) + \min\left(\mathbf{M}(i-1, j-1), \mathbf{M}(i-1, j), \mathbf{M}(i-1, j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

Example

$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

M(i,j)
E(i,j)

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

Example

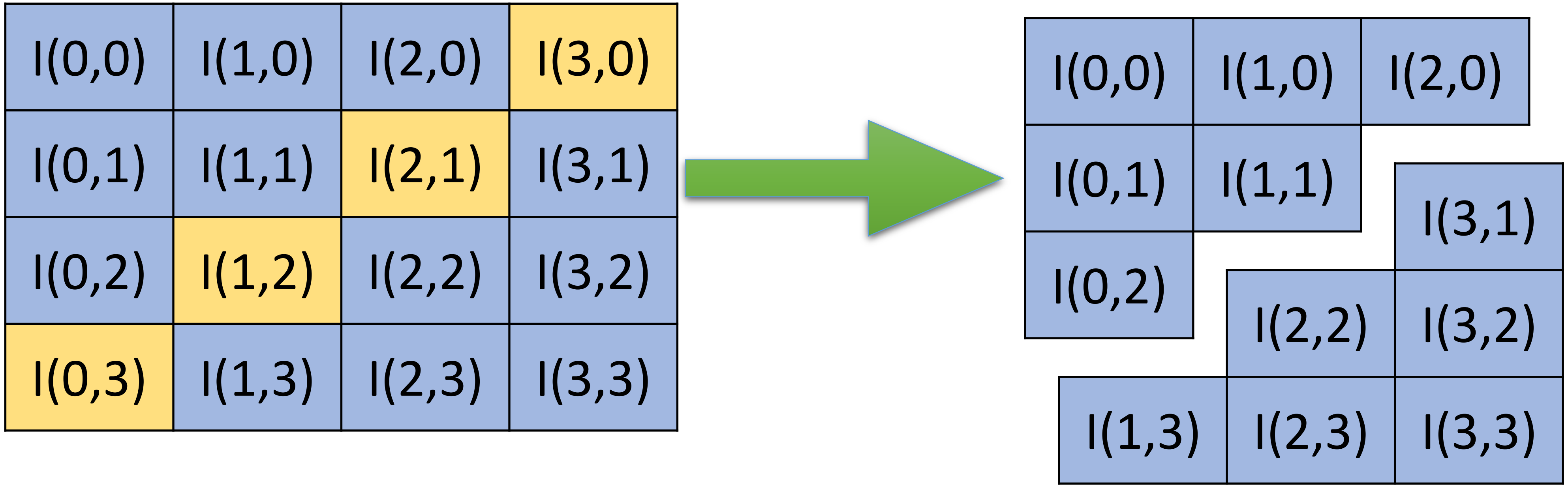
- Remove the seam on the actual image

5	8	12	3
9	7	6	12
14	9	10	8
13	14	15	16

I(0,0)	I(1,0)	I(2,0)	I(3,0)
I(0,1)	I(1,1)	I(2,1)	I(3,1)
I(0,2)	I(1,2)	I(2,2)	I(3,2)
I(0,3)	I(1,3)	I(2,3)	I(3,3)

Example

- Remove the seam on the actual image



Example

- Remove the seam on the actual image

I(0,0)	I(1,0)	I(2,0)	I(3,0)
I(0,1)	I(1,1)	I(2,1)	I(3,1)
I(0,2)	I(1,2)	I(2,2)	I(3,2)
I(0,3)	I(1,3)	I(2,3)	I(3,3)

I(0,0)	I(1,0)	I(2,0)
I(0,1)	I(1,1)	I(3,1)
I(0,2)	I(2,2)	I(3,2)
I(1,3)	I(2,3)	I(3,3)

Results

Results

- Adding seams