Winter 2019

DMET 901 – Computer Vision

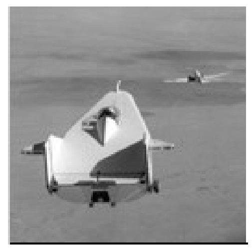
Image Transformation and Filtering

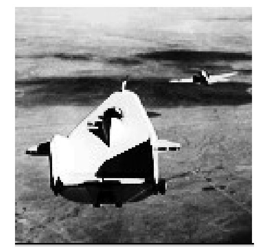
Seif Eldawlatly

Introduction

- Image preprocessing has two goals
  - Suppress information that is not relevant (Noise,…)
  - Enhance information that is relevant (Edges, ….)

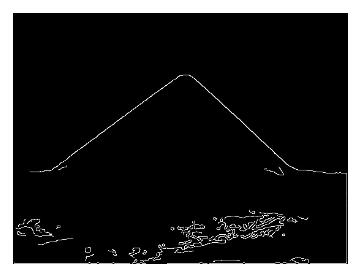
- Three topics
  - Transformations





  - Noise Filtering

  - Edge Detection



Transformations

- Gray-scale Transformations
  - A transform T of the original brightness p from the scale $[p_0, p_k]$ into the brightness q from a new scale $[q_0, q_k]$

$$q = T(p)$$

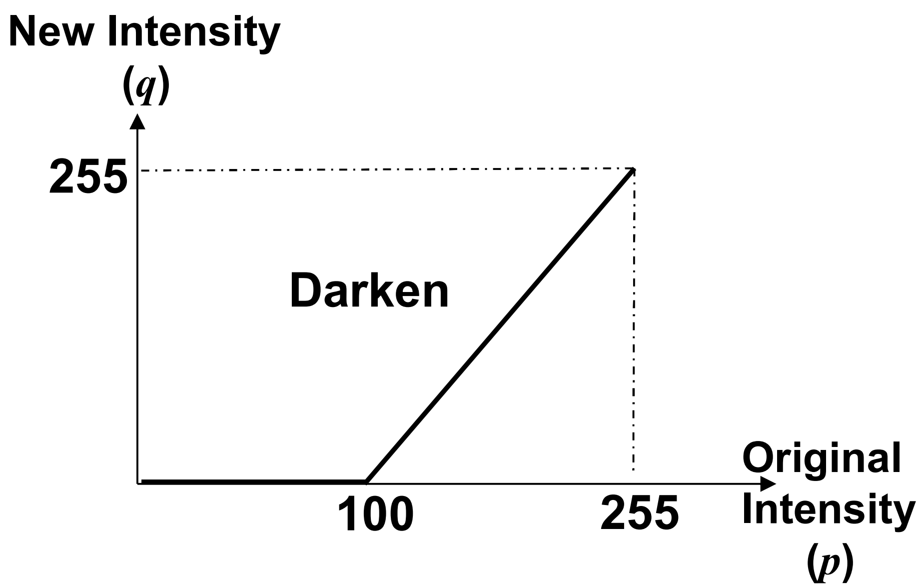
- Example: Increase the contrast

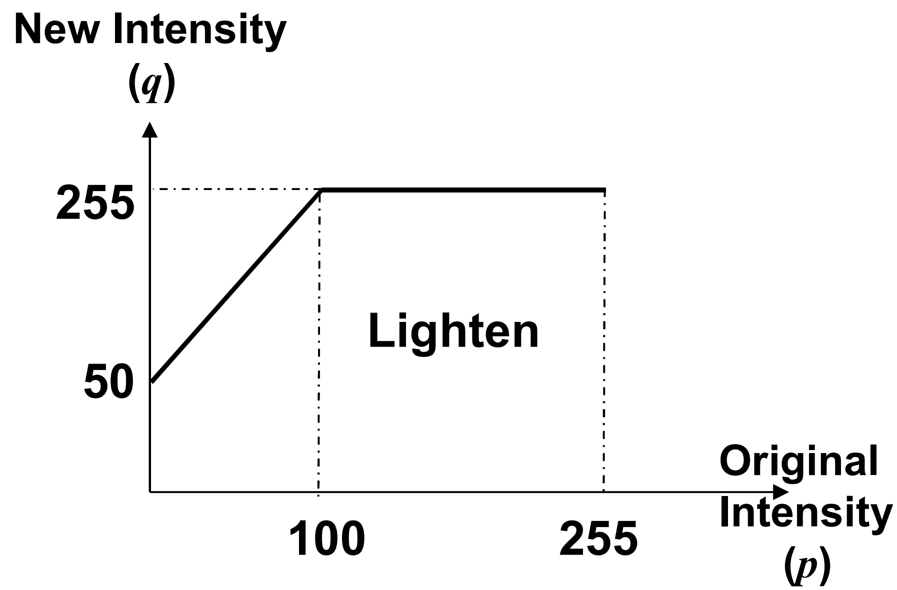
Original Image

Increase Contrast

Gray-scale Transformation

- Examples of simple operations on gray-scale images:
  Change Brightness



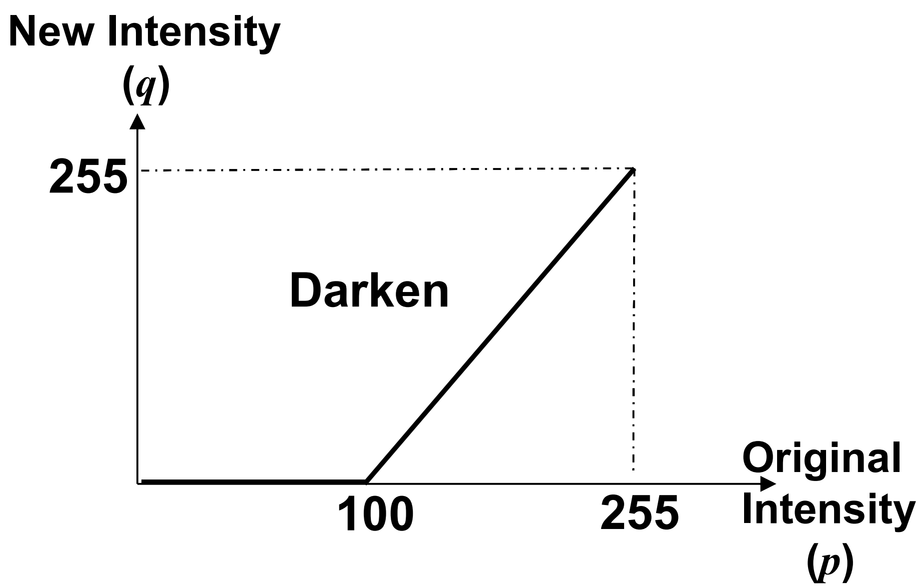


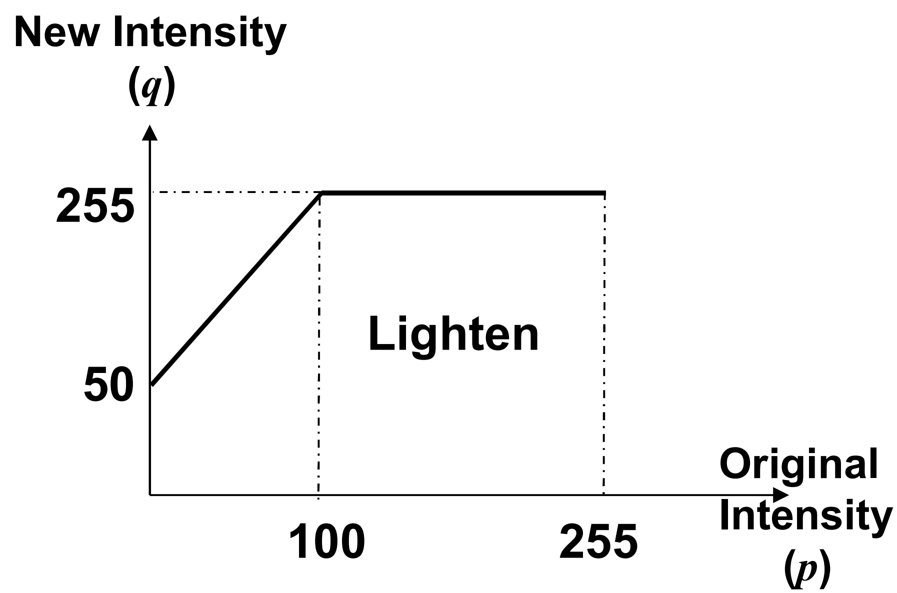
$$q = T(p) = \begin{cases} 0, & if \; p \le 100 \\ \dfrac{255}{155}(p - 100), & if \; p > 100 \end{cases}$$

$$q = T(p) = \begin{cases} \dfrac{1}{100}(205p - 5000), & if \; p \le 100 \\ 255, & if \; p > 100 \end{cases}$$

Gray-scale Transformation

- Examples of simple operations on gray-scale images:
  Change Brightness





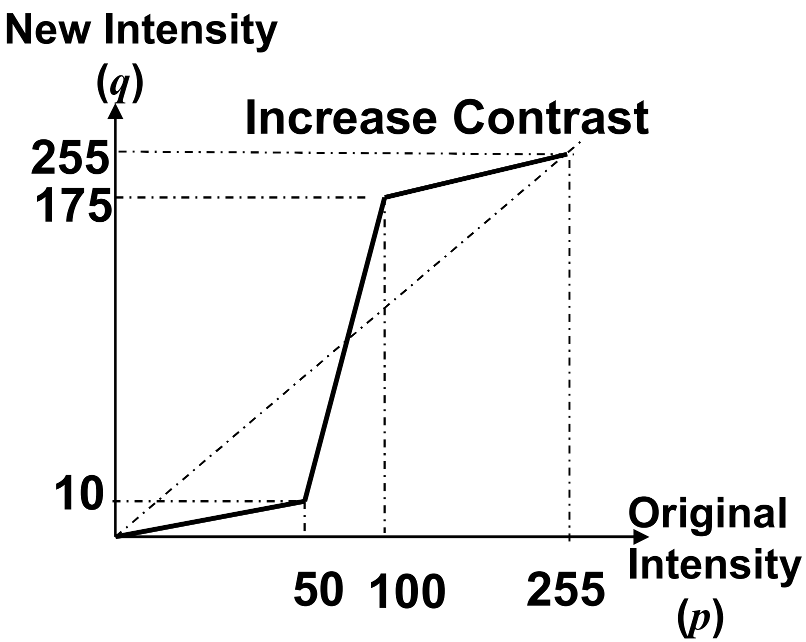
Original

Darkened

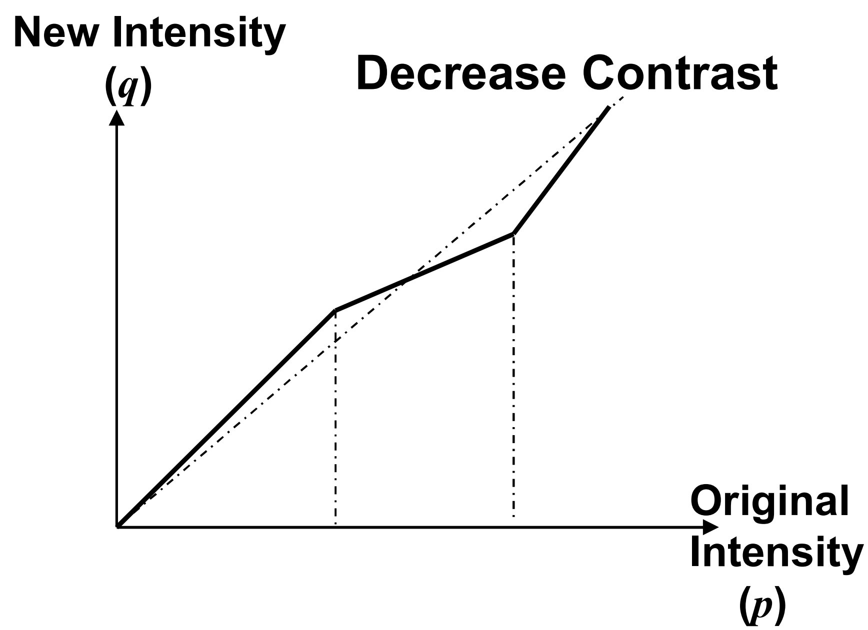
Lightened

Gray-scale Transformation

- Examples of simple operations on gray-scale images:
  Change Contrast



Bright becomes brighter,
Dark becomes darker

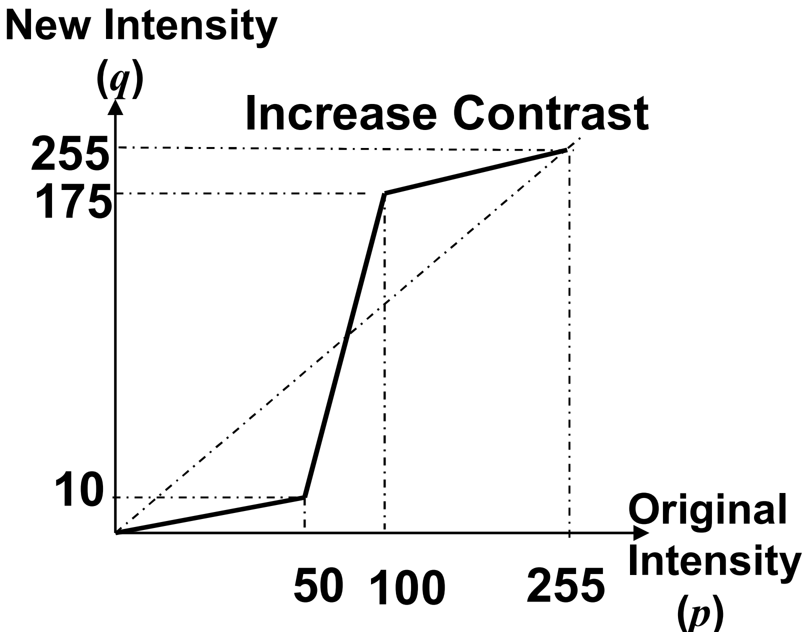


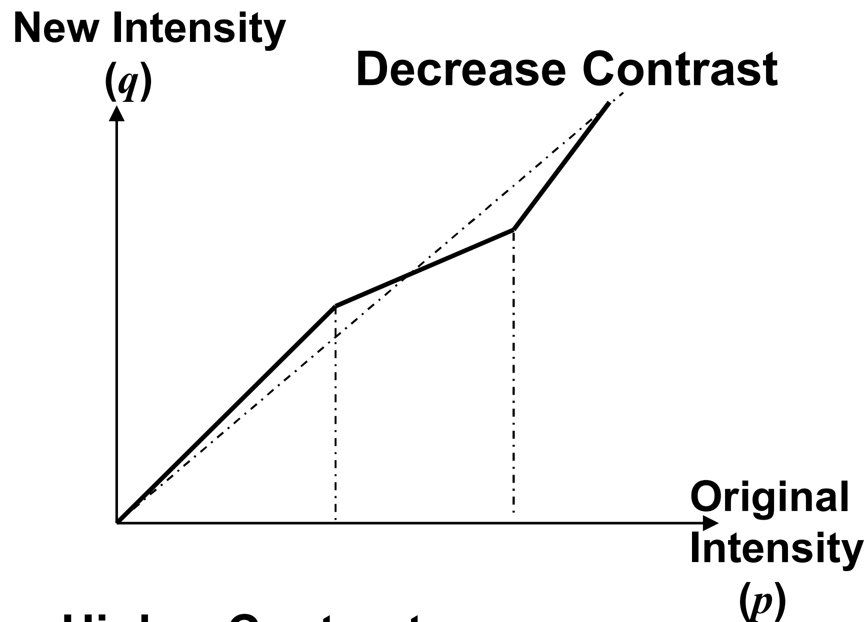
Bright becomes darker,
Dark becomes brighter

Gray-scale Transformation

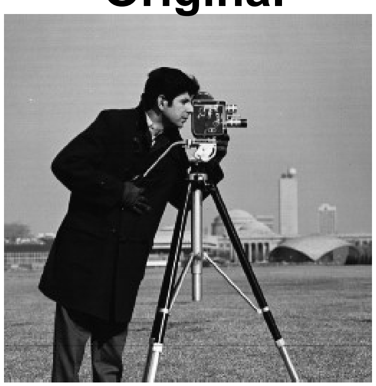
- Examples of simple operations on gray-scale images:

  Change Contrast





Original



Higher Contrast

Image Filtering

- Key idea: Images are redundant, a bad pixel can be replaced by a local average

- Examples:
  - Averaging
  - Averaging with limited data validity
  - Averaging using a rotating mask
  - Median filter

Image Filtering

- Averaging Filter

$$h_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4	8	6	11	8
10	5 *h*	9	7	10
6	3	6	4	3
9	5	7	9	8
12	2	5	7	4

5x5 Noisy Image

	6	7	7	
	7	6	7	
	6	5	6	

Filtered Image

Convolution

- The integral of the product of two functions after one is reversed and shifted

- In mathematical terms for discrete functions

$$f(i,j) = \sum_{(m,n)\in O}\sum g(i-m, j-n)h(m,n)$$

  where h is the convolution mask, g is the original image and O defines the size of the mask

- Both noise filtering and edge detection depend on the idea of convolution

- The dimensions of the filter are always odd so that there is always a central pixel

Convolution

- Example

$$f(i,j)=\sum_{m=1}^{-1}\sum_{n=1}^{-1} g(i-m,j-n)h(m,n)$$

$$h=\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

g

	1	2	3	4	5
1	4	8	6	11	8
2	10	5 h	9	7	10
3	6	3	6	4	3
4	9	5	7	9	8
5	12	2	5	7	4

f

	1	2	3	4	5
1					
2		6.375	6.875	7.4375	
3		5.6875	6	6.25	
4		5.6875	5.875	6.625	
5					

$$f(2,2)=\frac{1}{16}\left[4\times1+10\times2+6\times1+8\times2+5\times4+3\times2+6\times1+9\times2+6\times1\right]=6.375$$

$$f(2,3)=\frac{1}{16}\left[8\times1+5\times2+3\times1+6\times2+9\times4+6\times2+11\times1+7\times2+4\times1\right]=6.875$$

Image Filtering

- Weighted Averaging Filter

$$h_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

More weight for central pixel

$$h_3 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

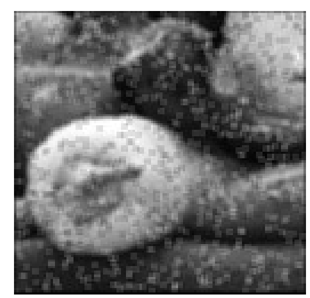
More weight for central pixel and the 4 neighbors

- Main Disadvantage of Averaging Filters: Blurring

  Because of considering pixels in the average that may have different properties than the processed pixel

Original Image

Noisy Image



Filtered Image

Image Filtering

- Averaging with limited data validity
  - Previous filters were linear
  - To avoid blurring, this method is based on the idea that the calculated average should be computed only from points in the neighborhood that satisfy certain condition
  - This makes such filter non-linear

$$h(i,j)=\begin{cases} 1 & for\, g(m+i,n+j)\in[\min,\max] \\ 0 & otherwise \end{cases}$$

Image Filtering

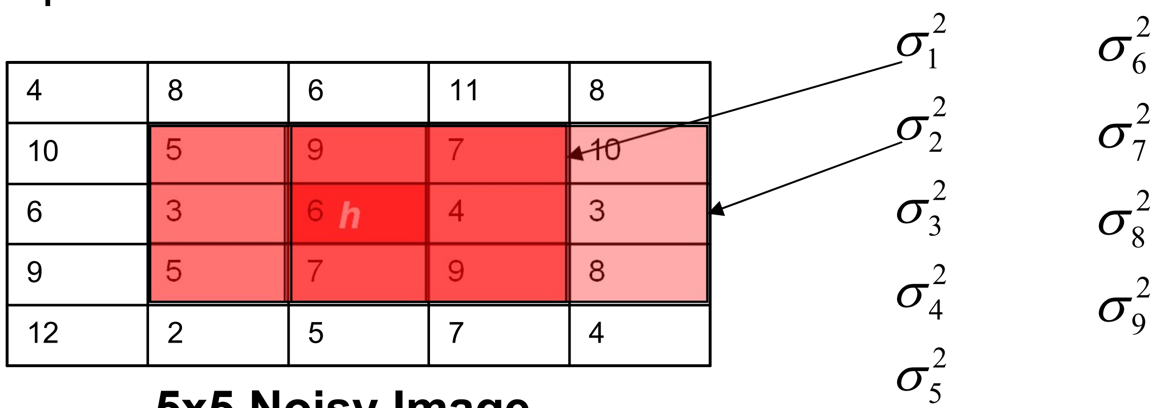
- Averaging using a rotating mask
  - Searches for the homogeneous part of the processed pixel neighborhood to avoid blurring
  - A brightness dispersion σ^2 is used as the region homogeneity measure

$$\sigma^2 = \frac{1}{n}\left(\sum_{(i,j)\in R}\left(g(i,j) - \frac{1}{n}\sum_{(i,j)\in R} g(i,j)\right)^2\right)$$

  - To search for the most homogenous part of the neighborhood, the dispersion for all possible mask rotations is calculated
  - The position of least dispersion is considered as the best filter for the processed pixel

Image Filtering

- Example: Consider a 3x3 filter



5x5 Noisy Image

Calculated Dispersions

- Rotating Mask Algorithm

1. Consider each image pixel $(i,\ j)$
2. Calculate the dispersion for all possible mask rotations around the pixel $(i,\ j)$ according to the given equation
3. Choose the mask with minimum dispersion
4. Assign to pixel $(i,\ j)$ in the output image the average brightness in the chosen mask

Image Filtering

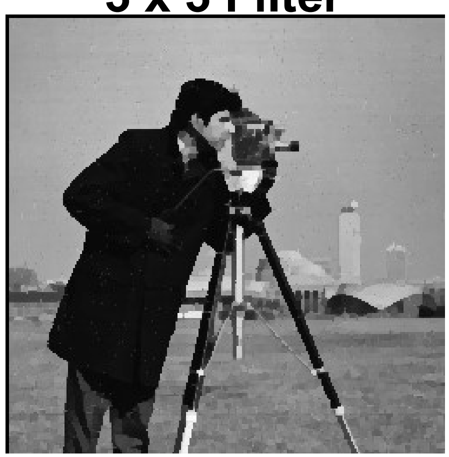
- Example

Noisy Image

Filtered using Average Filter (3 x 3)

Filtered using Rotating Mask

3 x 3 Filter



5 x 5 Filter

Image Filtering

- Median Filter
  - In a set of ordered values, the median is the central pixel

2	0	7
50	10	11
30	20	12

0 2 7 10 11 12 20 30 50

  - Works very well with salt and pepper noise

Image Filtering

- Median Filter

Original Image

Noisy Image

Filtered Image

Noisy Image

Filtered using Rotating Mask

Filtered using Median Filter

- Disadvantage: Damages thin lines

Image Filtering

- Disadvantage of Median Filter: Damages thin lines

0	255	0	0	0
0	255	0	0	0
0	255	0	0	0
0	255	0	0	0
0	255	0	0	0

Image with a 1-pixel wide white line

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Filtered using 3x3 median filter