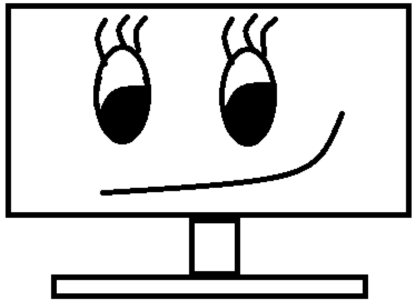
Seneca



CVI620/ DPS920
Introduction to Computer Vision

Noise & Filtering

Seneca College

Vida Movahedi

Overview

- Noise
  - Gaussian
  - Impulsive (Salt & Pepper)
- Filtering
  - Linear Filtering
  - Nonlinear Filtering

Noise [3]

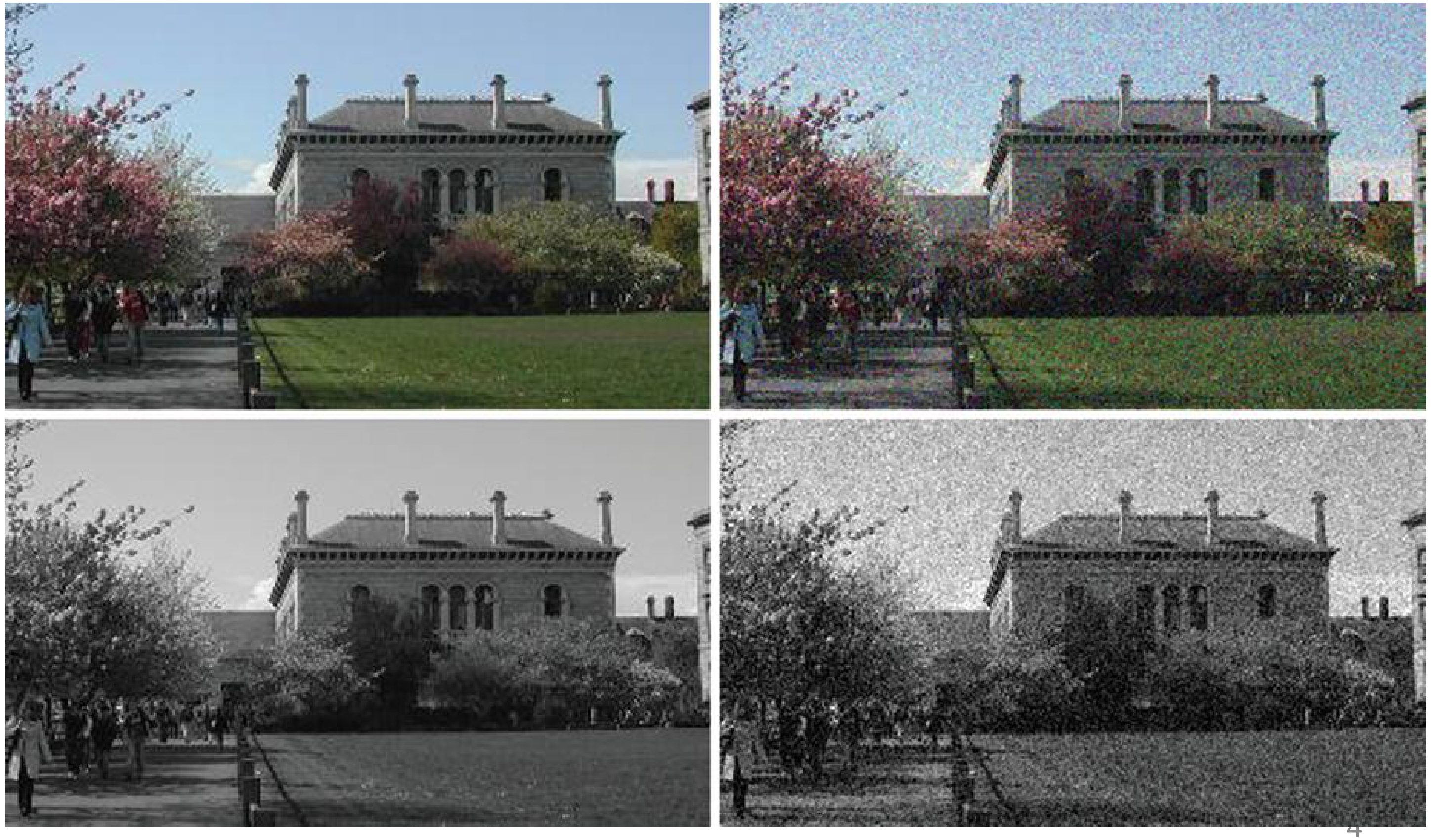
- Noise: anything that degrades the ideal image
- Sources of noise:
  - The environment,
  - The imaging device,
  - Electrical interference,
  - The digitization process, and so on.

- Noise is additive and random:

$$\hat{I}(i,j) = I(i,j) + n(i,j)$$

Gaussian Noise [3]

- A good approximation of real noise
- Modelled as a Gaussian (normal distribution with mean of 0)
- $n \sim N(\mu = 0, \sigma)$



Impulsive noise- Salt and Pepper Noise

- Impulsive: noise peaks or spikes
- Salt & Pepper is a model often used for impulsive noise
- Random values of brightness (darker or lighter) in the image
- Let p be the probability of noise $(0 \leq p \leq 1)$
- And x and y two random numbers between 0 and 1

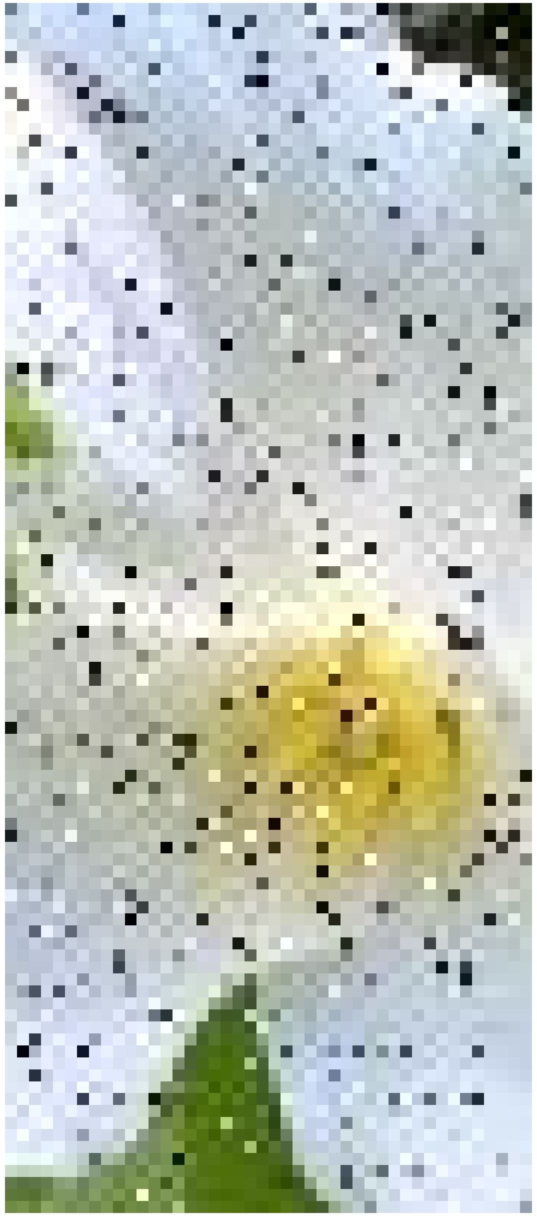
$$I_{noisy}(i,j) = \begin{cases} I_{min} + y(I_{max} - I_{min}) & \text{if } x < p \\ I_{in}(i,j) & \text{otherwise} \end{cases}$$

- In 8-bit images, I_{min}=0 and I_{max}=255

Examples

p = 0.1

p = 0.5



Correcting noise

- Observation: The image does not change sharply most of the time (low frequency), while noise is a sharp peak (high frequency)
- Therefore using the values of the neighbors, we can often lower the noise
- Take the average of the neighboring pixels
  - This is equivalent to low-pass filtering, which keeps the low frequency content and filters high frequency signals
- Disadvantage: This will reduce the sharpness of edges in the image

Averaging

- The value at pixel (i, j) is calculated as the average of the pixels in its neighborhood
- Suitable for removing random noise, or smoothing

$j \rightarrow$, $i \downarrow$

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

I_{in}

				?			

I_{out}

5x5 neighborhood

$$\text{new value} = \frac{9+62+\cdots+102+62}{25}$$

Linear Filtering

Linear filtering

- **Filtering**: an algorithm that starts with some image $I_{in}(i, j)$ and computes a new image $I_{out}(i, j)$ using a neighborhood operator
- **Kernel**: A template defining the neighborhood and the operator
- **Linear filter / linear kernel**: Values are calculated as a weighted sum of values in the neighborhood

$$I_{\text{out}}(i,j) = \sum_{x,y \in \text{Kernel}} k(x,y) \cdot I_{\text{in}}(i+x, j+y)$$

Averaging- box kernel

- Averaging is equivalent to convolution with a box kernel

$j \rightarrow$

$i \downarrow$								
	62	79	23	119	120	105	4	0
	10	10	9	62	12	78	34	0
	10	58	197	46	46	0	0	48
	176	135	5	188	191	68	0	49
	2	1	1	29	26	37	0	77
	0	89	144	147	187	102	62	208
	255	252	0	166	123	62	0	31
	166	63	127	17	1	0	99	30

$$k = 1/25 \ \text{x} \ \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & \underline{\mathbf{1}} & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

5x5 (normalized) box kernel

- $$I_{out} = I_{in} * k$$

Convolution (*) is a mathematical operation

$$k = 1/9 \ \text{x} \ \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & \underline{\mathbf{1}} & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

3x3 (normalized) box kernel

5x5 (normalized) box kernel

```
// Using this function
blur(img, dst, cv::Size(5, 5));

// Or use this function
boxFilter(img, dst, CV_8U, cv::Size(5,5));

// Or build a box kernel yourself and then filter
Matx<float,5,5> myK= cv::Matx<float,5,5>::all(1 / 25.0f);
filter2D(img, dst, CV_8U, myK);
```

Examples

Salt & pepper noise with p = 0.1

After 5x5 box filter

Separable filtering

Some filters are separable into smaller filters. Applying smaller filters is faster (faster implementation).

For example:
Convolving with

$^1/_{25}$ x

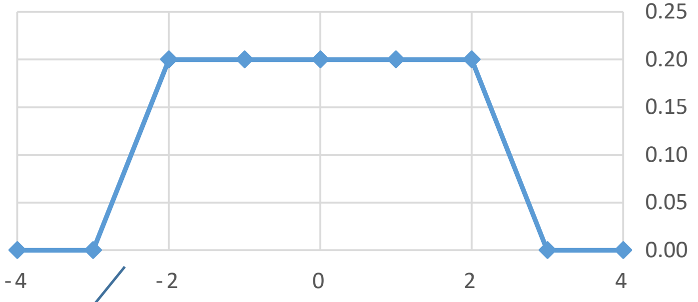
1	1	1	1	1
1	1	1	1	1
1	1	$\underline{1}$	1	1
1	1	1	1	1
1	1	1	1	1

5x5 (normalized) box kernel

Is equivalent to
convolving with

1/5

1	1	$\underline{1}$	1	1



A low-pass filter!

And then
convolving with

1/5

1
1
$\underline{1}$
1
1

Gaussian Filter (smoothing)

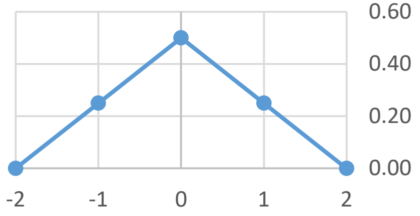
- The Gaussian Filter (2-D bell curve) is separable
- It can be applied by first convolving with a 1D Gaussian Filter horizontally and then vertically
-
```
cv::Mat cv::getGaussianKernel(
  int          ksize,              // Kernel size
  double       sigma,              // Gaussian half-width
  int          ktype = CV_32F      // Type for filter coefficients
);
```
- The coefficients are computed from $$k_i = \alpha \cdot e^{-\frac{(i-(ksize-1)2)^2}{(2\sigma)^2}}$$
- It can be applied using sepfilter2D (instead of filter2D)

Bilinear kernel

- Also smoothing (removing noise)
- Equivalent to convolving with two separable ‘tent’ functions
- Example: 3x3 bilinear kernel:

1-D Tent kernel:

$$\frac{1}{4}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline\end{array}$$



2-D Bilinear kernel:

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Examples

p = 0.1

After 3x3 bilinear filter

Nonlinear Filtering

Nonlinear filter

- The output pixel value is not a linear function of pixel values in the input
- Example: Median Filter
- The output value is the median of the pixels in the neighborhood

```
void cv::medianBlur(
  cv::InputArray  src,                 // Input image
  cv::OutputArray dst,                 // Result image
  int             ksize                // Kernel size
);
```

Examples

p = 0.1

After 3x3 median filter

References

[1] Computer Vision: Algorithms and Applications, R. Szeliski (http://szeliski.org/Book)

[2] Learning OpenCV 3, A. Kaehler & G. Bradski

- Available online through Safari Books, Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5153244920003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US

[3] Practical introduction to Computer Vision with OpenCV, Kenneth Dawson-Howe

- Available through Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5142810950003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US