EEE-6512: Image Processing and Computer Vision

October 16, 2017

Lecture #7: Spatial Domain Filtering

Damon L. Woodard, Ph.D.
Dept. of Electrical and Computer Engineering
dwoodard@ece.ufl.edu

Chapter Outline

- Convolution
- Smoothing by Convolving with a Gaussian
- Computing the First Derivative
- Computing the Second Derivative
- Nonlinear Filters
- Grayscale Morphological Operators

Non-Linear Filters

Nonlinear Filters - Median Filter

- **Salt-and-pepper noise**: each pixel is set to either the minimum (“pepper”) or maximum (“salt”) possible gray level, or it remains unchanged:

$$I'(x,y) = \begin{cases} 0 & \text{if } 0 \leq \xi < p \\ 255 & \text{if } p \leq \xi < p+q \\ I(x,y) & \text{otherwise} \end{cases} \qquad \xi \sim U(0,1)$$

- **Median filter**: replaces each pixel with the median of all the gray levels in a local neighborhood, generally defined by a square window.

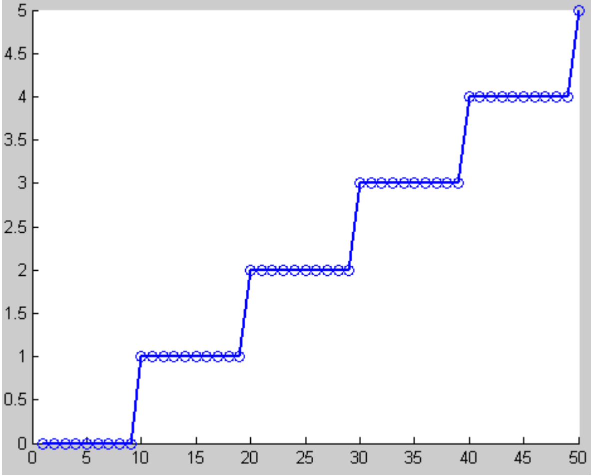
Nonlinear filters

Median filter:
Replace pixel with median of surrounding *nxn* region

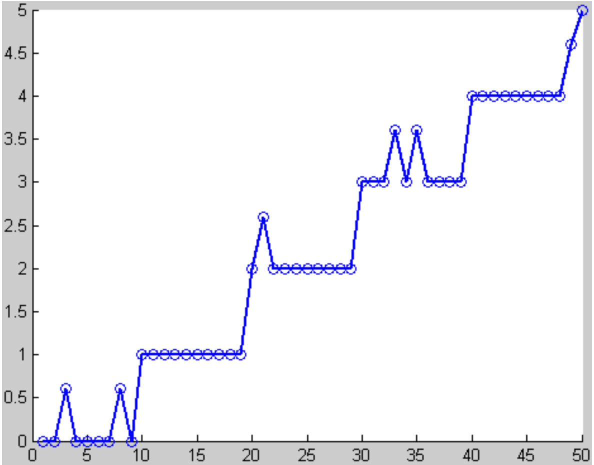
Good for impulse noise (salt-and-pepper noise)

What is mean filter good for?

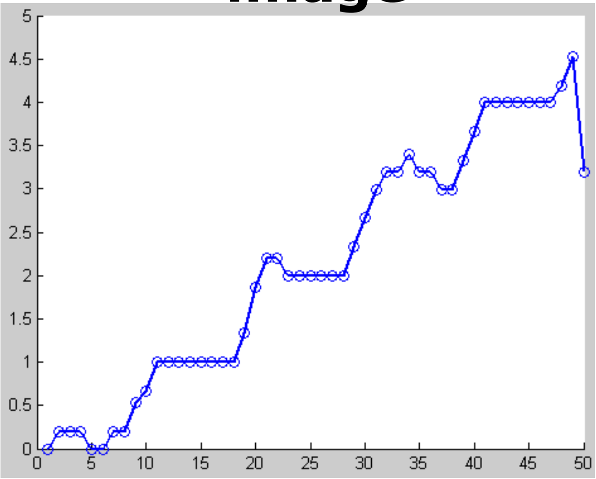
additive Gaussian noise



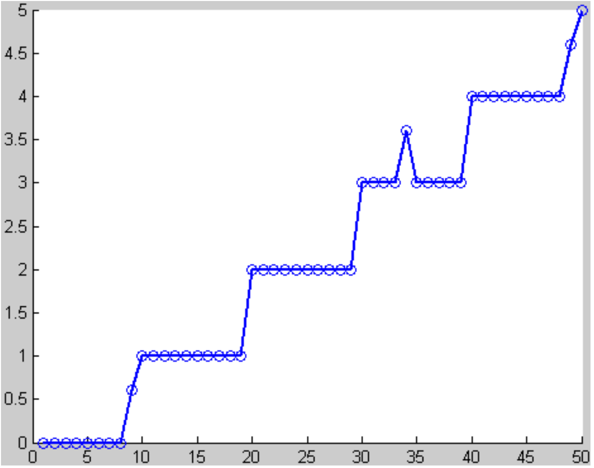
image



impulse noise



mean filtering



median filtering

Non-Local Means

- **Non-local means (NLM)**: a particularly effective way to reduce the effects of noise in an image by computing a weighted average over *all* pixels in the image.

$$I'(x, y) = \frac{1}{\eta} \sum_{(x',y') \in I} w_{x',y'} I(x', y')$$

- Similar looking regions have more influence on the outcome than regions whose appearance is far from the window around the target pixel.

Bilateral Filtering

- **Bilateral filter**: contains two kernels, a *spatial kernel* and a *range kernel*.
- The spatial kernel g_s weights neighboring samples according to their proximity to the central sample, while the range kernel g_r weights neighboring samples according to their similarity in value to the central sample.

$$f'(x) = f(x) \circledcirc \langle g_s(x), g_r(z) \rangle = \frac{1}{\eta(x)} \sum_i f(i) g_s(x-i) g_r(f(x) - f(i))$$

Bilateral Filtering (cont'd)

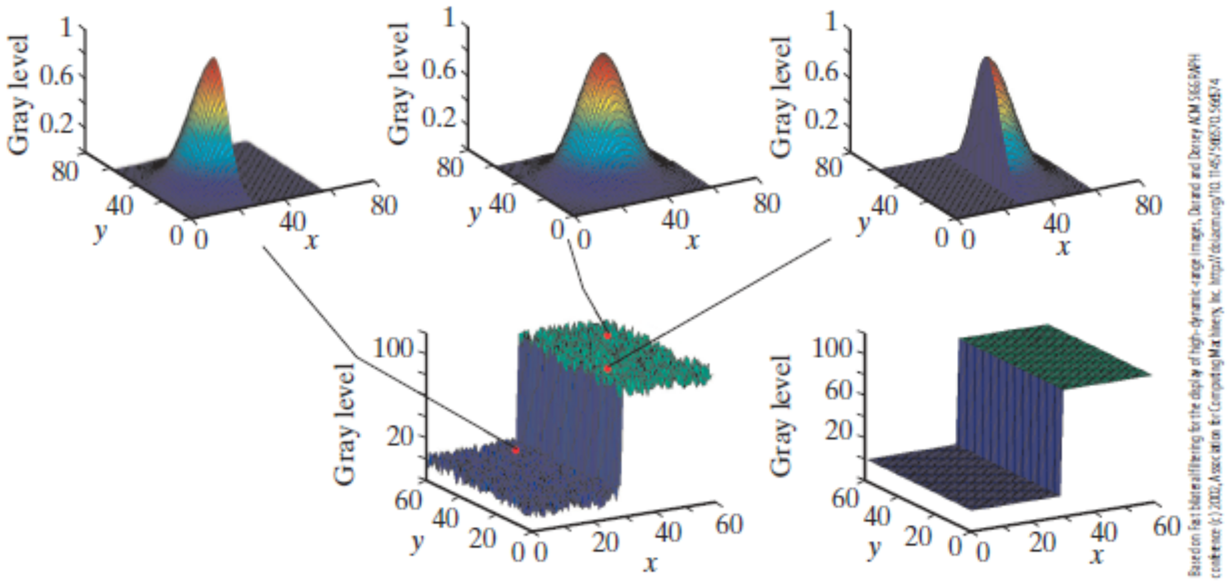
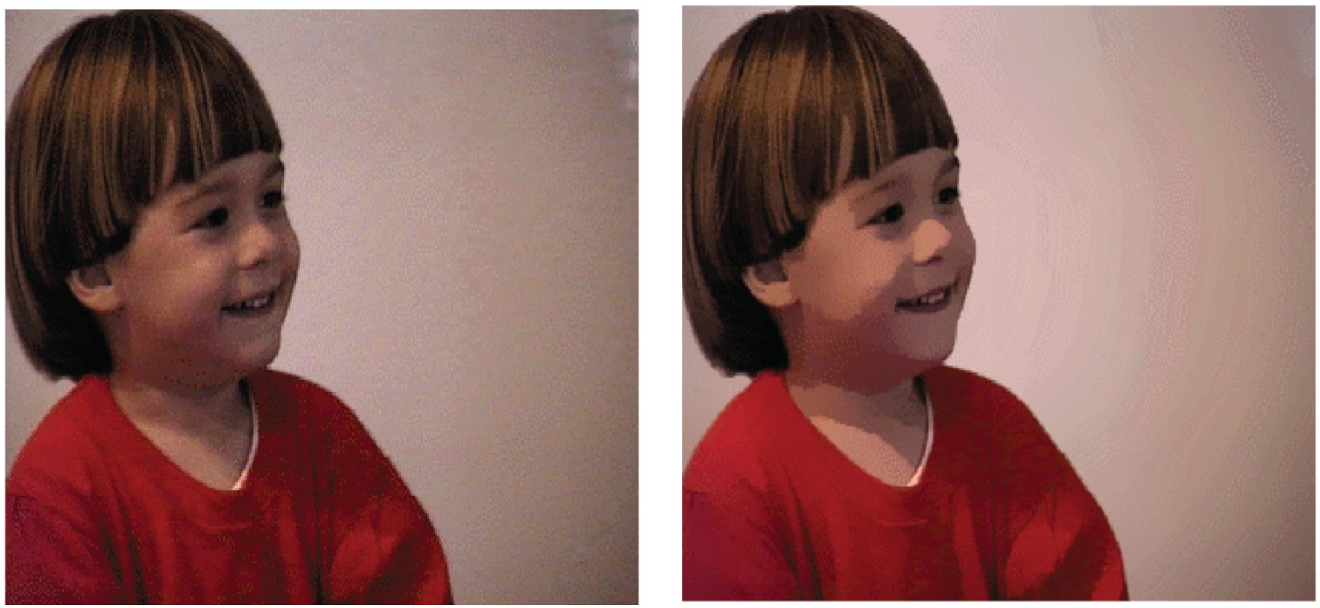


Figure 5.17 Bilateral filtering of a noisy step edge preserves the crisp edge as it smooths out the noise on either side of the edge. The top row shows the kernel at three locations: Far from the edge, the kernel approximates a Gaussian, whereas near the edge it approximates half a Gaussian.

Bilateral Filtering (cont'd)

Figure 5.18 Repeated applications of bilateral filtering yield a cartoon-like image in which the colors are flattened in local regions. The result (right) was obtained by applying $n_{iter} = 5$ iterations of bilateral filtering on the input (left).



Repeated applications of bilateral filtering yield cartoon like images.

Bilateral Filtering for Large Windows

When the window size is large, bilateral filtering is extremely slow, requiring computation that is proportional to the size of the window for every pixel in the image.

Adaptive Smoothing

- **Adaptive smoothing** simply computes a weighted average of the neighbors of a pixel, with weights that discourage smoothing across boundaries:

$$I^{(t+1)}(x, y) = \frac{\sum_i \sum_j I^{(t)}(x+i, y+j) w^{(t)}(x+i, y+j)}{\sum_i \sum_j w^{(t)}(x+i, y+j)}$$

- Weights are defined to be inversely related to the magnitude of the magnitude of the image gradient.

Grayscale Morphological Operators

Grayscale Dilation and Erosion

- A binary image can be viewed as a set of ON pixels.
- The dilation or erosion of a binary image is given by the union or intersection of translated versions of the structuring element placed within the image:

$$I \oplus B = \bigcup_{\boldsymbol{\delta} \in B} I_{\boldsymbol{\delta}} = \bigcup_{\boldsymbol{\delta} \in I} B_{\boldsymbol{\delta}} = \{\boldsymbol{\delta} : \check{B}_{\boldsymbol{\delta}} \cap I \neq \emptyset\} \qquad \text{(binary or grayscale)}$$

$$I \check{\ominus} B = \bigcap_{\boldsymbol{\delta} \in \check{B}} I_{\boldsymbol{\delta}} = \bigcap_{\boldsymbol{\delta} \notin I} \neg \check{B}_{\boldsymbol{\delta}} = \{\boldsymbol{\delta} : B_{\boldsymbol{\delta}} \subseteq I\} \qquad \text{(binary or grayscale)}$$

Grayscale morphology

- **Grayscale dilation**
  - Similar to convolution, but replace sum with max and replace multiplication with addition
  - Reduces dark details

- **Grayscale erosion**
  - Similar to convolution, but replace sum with min and replace multiplication with subtraction
  - Reduces bright details

Grayscale Opening and Closing

- Grayscale opening and closing are defined in the same way as binary opening and closing:

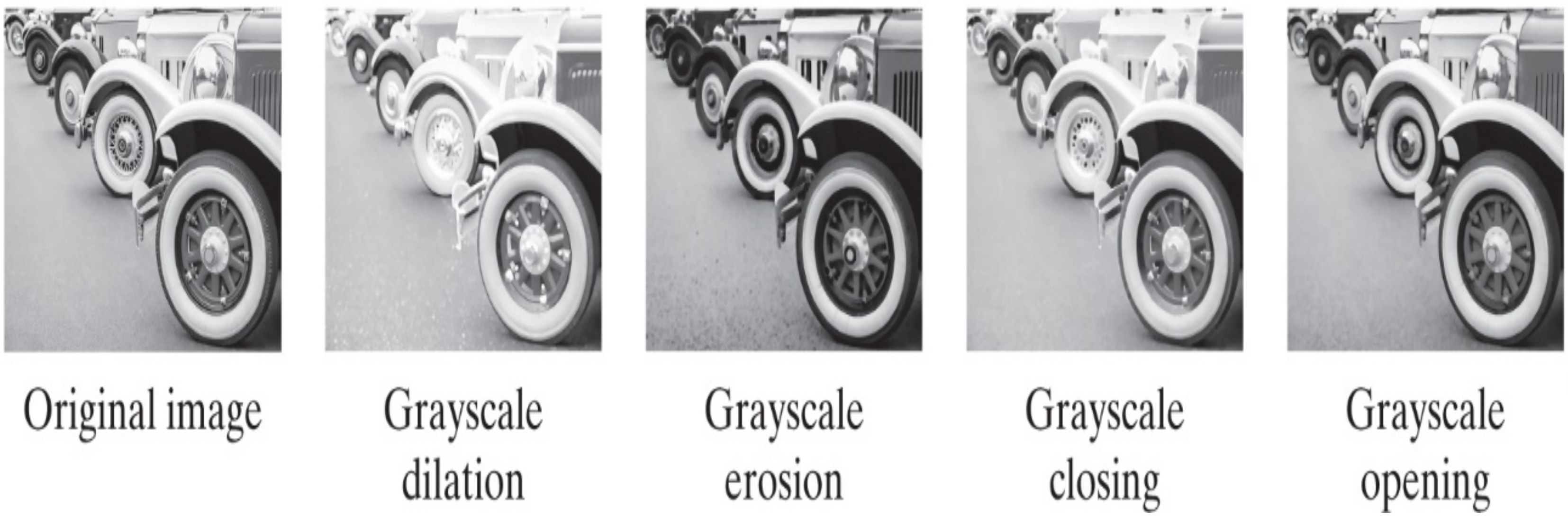
$$I \bullet B = (I \oplus B) \check{\ominus} B \quad (\text{grayscale closing})$$

$$I \circ B = (I \check{\ominus} B) \oplus B \quad (\text{grayscale opening})$$

- Opening grayscale image with SE removes light details smaller than SE.
- Closing removes dark detail smaller than the SE.

Grayscale Dilation, Erosion, Closing, and Opening

Figure 5.24 An image, and the result of applying grayscale dilation, erosion, closing, and opening, respectively, using a flat, circular structuring element.



Top-Hat Transform

- **White top-hat (WTH)** transform:
  - Also known as "top-hat by opening"
  - The difference between the original image and the grayscale opening of the image
  - Preserves objects that are brighter than their surroundings

- **Black top-hat (BTH)** transform:
  - Also known as "top-hat by closing"
  - The difference between the grayscale closing of the original image and the image itself.
  - Preserves objects that are darker than their surroundings

Top-Hat Transform (cont'd)

$$I'_{WTH} = I - (I \circ B) \qquad \text{(white top-hat)}$$

$$I'_{BTH} = (I \bullet B) - I \qquad \text{(black top-hat)}$$

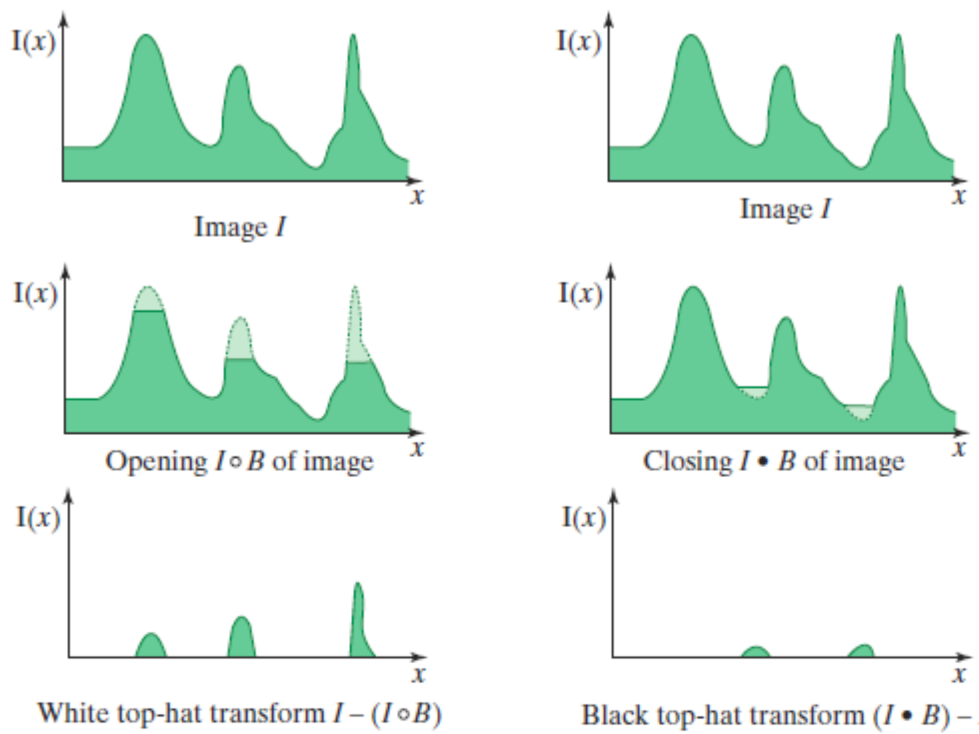


Figure 5.25 White top-hat (WTH) transform (left), and black top-hat (BTH) transform (right).

Questions?