Order-Statistic filters

- Median filter
- Max and min filters
- Midpoint filter
- Alpha-trimmed mean filter

**Order-Statistic filters**

- Median filter
- Max and min filters
- Midpoint filter
- Alpha-trimmed mean filter

**Computer Vision & Digital Image Processing**

Image Restoration and Reconstruction III

**Order-Statistic filters**

- Median filter
- Max and min filters
- Midpoint filter
- Alpha-trimmed mean filter

**Median filter**

- Replaces the value of a pixel by the *median* of the pixel values in the neighborhood of that pixel
  \[ \hat{j}(x,y) = \text{median}(g(x,y)) \]
- The pixel at \((x,y)\) is included in the calculation
- Works well for various noise types, with less blurring than linear filters of similar size
- Odd sized neighborhoods and efficient sorts yield a computationally efficient implementation
- Most commonly used order-statistic filter

**Max and min filters**

- The 100th percentile filter (or max filter) is given by
  \[ \hat{j}(x,y) = \max_{(x,y) \in G} g(x,y) \]
- Useful for finding the brightest points in an image
- Tends to reduce pepper noise (i.e. dark pixel values)
- The 0th percentile filter (or min filter) is given by
  \[ \hat{j}(x,y) = \min_{(x,y) \in G} g(x,y) \]
- Both filters require a data sort

**Max and min filters**

- The 100th percentile filter (or max filter) is given by
  \[ \hat{j}(x,y) = \max_{(x,y) \in G} g(x,y) \]
- Useful for finding the brightest points in an image
- Tends to reduce pepper noise (i.e. dark pixel values)
- The 0th percentile filter (or min filter) is given by
  \[ \hat{j}(x,y) = \min_{(x,y) \in G} g(x,y) \]
- Both filters require a data sort
Midpoint filter

- Replaces the value of a pixel by the midpoint between the maximum and minimum pixels in a neighborhood
  \[ \hat{f}(x, y) = \frac{1}{2} \left( \max_{(s,t) \in S_{x,y}} g(s, t) + \min_{(s,t) \in S_{x,y}} g(s, t) \right) \]
- Combines order statistics and averaging
- Works best for randomly distributed noise (e.g., Gaussian or uniform)

Alpha-trimmed mean filter

- If we delete the \( \frac{d}{2} \) lowest and the \( \frac{d}{2} \) highest intensity values from a neighborhood \( g(s, t) \) of size \( mn \) and let \( g_{r}(s, t) \) represent the remaining \( mn-d \) pixels, the average of the remaining pixels is called an alpha-trimmed mean filter and is given by:
  \[ \hat{f}(x, y) = \frac{1}{mn-d} \sum_{(s,t) \in S_{x,y}} g_{r}(s, t) \]
- \( d \) can vary from 0 to \( mn-1 \)
- If \( d=0 \) the filter becomes the arithmetic mean filter
- If \( d=mn-1 \), the filter reduces to a median filter

Alpha-trimmed mean filter example

Adaptive filters

- All filters considered thus far are applied to an image without regard for how image characteristics may vary from one point to another in the image
- An adaptive filter is one whose behavior can change based on statistical characteristics of an area within the image
  - This is typically the \( mn \) filter region in the \( S_{x,y} \) window
- Generally provides superior performance at the cost of increased filter complexity

Adaptive, local noise reduction filter

- The mean and variance are reasonable parameters upon which to base a simple adaptive filter
  - They are closely related to image properties
    - The mean gives the average intensity over a region
    - The variance gives a measure of the contrast in a region
- A simple filter will operate on a local region \( S_{x,y} \) with the response at any point \((x, y)\) base on four quantities:
  - The value of the noisy image at \((x, y)\): \( g(x, y) \)
  - The variance of the noise corrupting \( f(x, y) \) to form \( g(x, y) \): \( \sigma_{\eta}^2 \)
  - The local mean of the pixels in \( S_{x,y} \): \( \mu_{L} \)
  - The local variance of the pixels in \( S_{x,y} \): \( \sigma_{L}^2 \)

Adaptive, local noise reduction filter algorithm

- If \( \sigma_{\eta}^2 = 0 \), return the value \( g(x, y) \)
  - This is the zero-noise case where \( g(x, y) = f(x, y) \)
- If the local variance \( \sigma_{L}^2 \) is high relative to \( \sigma_{\eta}^2 \), return a value close to \( g(x, y) \)
  - A high local variance is generally associated with image features (i.e., an edge, etc.) and should be preserved
- If \( \sigma_{L}^2 = \sigma_{\eta}^2 \), return the arithmetic mean of the pixels in \( S_{x,y} \)
  - This occurs if the local area has the same properties as the overall image. Local noise is reduced by averaging.
Adaptive, local noise reduction filter equation

- An adaptive expression may be written as:
  \[
  \hat{j}(x, y) = g(x, y) - \frac{\eta^2}{\sigma^2} [g(x, y) - m_i]
  \]
- The only quantity that must be known is \( \sigma^2 \eta \)
- Everything else can be computed from \( S_{x,y} \)
- An assumption here is that \( \sigma^2 \eta \leq \sigma^2_L \)
  - This is generally reasonable given that the noise we are considering is additive and position independent
  - If this is not true then a simple test could set the ratio of the variances to one if \( \sigma^2 \eta > \sigma^2_L \)

Adaptive, local noise reduction filter example

Adaptive median filter

- A median filter works well in the spectral density of the impulse noise is not large
  - A \( P_a \) and \( P_b \) less than 0.2 is a good general rule of thumb
- An adaptive median filter can handle noise with probabilities greater than these
- An additional benefit is that the adaptive median filter attempts to preserve detail while smoothing the impulse noise
- The adaptive median filter works in a rectangular window area \( S_{x,y} \)
  - The size of \( S_{x,y} \) is not fixed
- The output of the filter is a single value that will be used to replace the center value of \( S_{x,y} \)

Adaptive median filter example

Adaptive median filter algorithm

- Consider the following notation
  - \( z_{\text{min}} \) = minimum intensity value in \( S_{x,y} \)
  - \( z_{\text{max}} \) = maximum intensity value in \( S_{x,y} \)
  - \( z_{\text{med}} \) = median intensity of values in \( S_{x,y} \)
  - \( z(x,y) \) = intensity value at \( (x,y) \)
  - \( S_{\max} \) = maximum allowed size of \( S_{x,y} \)
- The algorithm works in two stages (denoted \( A \) and \( B \))
  
  **Stage A:**
  - \( A1 = z_{\text{med}} - z_{\text{min}} \)
  - \( A2 = z_{\text{med}} - z_{\text{max}} \)
  - If \( A1 > 0 \) AND \( A2 < 0 \), goto Stage B
  - Else increase window size
    - If window size \( \leq S_{\max} \) repeat Stage A
    - Else output \( z_{\text{med}} \)
  
  **Stage B:**
  - \( B1 = z(x,y) - z_{\text{min}} \)
  - \( B2 = z(x,y) - z_{\text{max}} \)
  - If \( B1 > 0 \) AND \( B2 < 0 \), output \( z(x,y) \)
  - Else output \( z_{\text{med}} \)

Adaptive median filter example

**FIGURE 5.14** (a) Image corrupted by salt-and-pepper noise with probabilities \( P_a = P_b = 0.25 \). (b) Result of filtering with a \( 3 \times 3 \) median filter. (c) Result of adaptive median filtering with \( S_{\max} = 7 \). (d) Result of adaptive median filtering with \( S_{\max} = 7 \).