### OTHER WINDOW OPERATORS - NOISE FILTERS / SMOOTHING

#### Smoothing (Averaging)

- Reduces noise but blurs edges.
- \[ p^* = \frac{\sum w_i p_i}{\sum w_i} \]
- **Scale Factor**: \[ \frac{9}{9} = \frac{3}{3} \]

#### Balanced Binary Smoothing

- Image must be 1's and 0's.
- If window result \( \geq 3 \), then \( p^* = 1 \) in new image.
- Else \( p^* = 0 \) in new image.
- Equivalent to: \( p_1 + p_2 + p_3 + p_4 + p_5 \geq 3 \)

#### Median Filter (Noise Filter)

- Does not blur edges.
- Select window, sort pixel intensities from lowest to highest.
- \( p^* = \text{median value of intensities of pixels under window} \)

**Example:**

<table>
<thead>
<tr>
<th>Window</th>
<th>10 0 1</th>
<th>-10 0 1</th>
<th>10 10 15 15 20 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixels</td>
<td>10 0 10</td>
<td>-10 0 15</td>
<td>Median: 10</td>
</tr>
<tr>
<td></td>
<td>50 15 15</td>
<td>20 50 50</td>
<td>Average: 30</td>
</tr>
</tbody>
</table>
|         |         |         | Adjacent pixels: 10

**Noise Filter Example:**

- \( p^* = 10 \) to new image.
Figure 4.23 (a) Original image; (b) image corrupted by impulse noise; (c) result of $5 \times 5$ neighborhood averaging; (d) result of $5 \times 5$ median filtering. (Courtesy of Martin Connor, Texas Instruments, Inc., Lewisville, Tex.)
4. 15 pt's. Perform "Balanced Binary Smoothing on the following image

\[
\begin{array}{cccc}
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
\end{array}
\]

\[
P_1 \quad P_2 \quad P_3 \\
P_4 \quad P_5 \\
P_6 \quad P_7 \quad P_8
\]

If \( P_2 + P_4 + P_5 + P_6 + P_8 \geq 3 \)

Then \( P_X = 1 \)

Else \( P_X = 0 \)
Section 5.3, p.230 Text

- **THE ARITHMETIC MEAN FILTER** in the text is the averaging or smoothing warp discussed earlier.

- **GEOMETRIC MEAN FILTER**:

\[
\phi^* = \left[ \prod_{i=1}^{m} \phi_i \right]^{1/m}, \quad \text{MAX M WOPR}
\]

All \( w_i = 1 \)

(Coupled with different weighting?)

Does not blur image as much as the averaging filter.

- **HARMONIC MEAN FILTER**:

\[
\phi^* = \frac{\sum_{i=1}^{m} \left( \frac{1}{\phi_i} \right)}{m}, \quad \text{MAX M WOPR}
\]

All \( w_i = 1 \)

(Coupled with different weighting?)

Works well for Gaussian and salt noise but not pepper. Need to set a non-zero lower limit for \( P_i \) noise.

- **CONTRAHARMONIC MEAN FILTER**:

\[
\phi^* = \frac{\sum_{i=1}^{m} \left( \frac{1}{\phi_i} \right)^{\phi+1}}{\sum_{i=1}^{m} \left( \frac{1}{\phi_i} \right)^\phi}, \quad \text{MAX M WOPR}
\]

\( \phi > 0 \) reduces pepper noise

\( \phi < 0 \) reduces salt noise

\( \phi = 0 \) averaging filter

\( \phi = -1 \) harmonic mean filter

All \( w_i = 1 \)

Set non-zero lower limit for sum.
ORDER-STATISTIC NOISE FILTERS

MEDIAN FILTER DISCUSSED EARLIER

MAX FILTER:
\[ r^* = \max \{ r_i \text{ in MAX M window} \} \]
REDUCES PEPPER NOISE

MIN FILTER:
\[ r^* = \min \{ r_i \text{ in MAX M window} \} \]
REDUCES SALT NOISE

MIDPOINT FILTER
\[ r^* = \frac{1}{2} \left[ \max \{ r_i \} + \min \{ r_i \} \right] \text{ MAX M window} \]
REDUCES GAUSSIAN AND UNIFORM NOISE

ALPHA-TRIMMED MEAN FILTER:
SELECT AN EVEN INTEGER \( \delta \) WHERE \( 0 \leq \delta \leq (2M - 1) \).
DELETE \( \frac{\delta}{2} \) \( r_i \) THAT HAVE THE LOWEST INTENSITIES FROM UNDER THE MAX M WINDOW.
DELETE \( \frac{\delta}{2} \) \( r_i \) THAT HAVE THE HIGHEST INTENSITIES FROM UNDER THE MAX M WINDOW.
AVERAGE THE REMAINING \( r_i \) UNDER THE MAX M WINDOW.

\[ r^* = \frac{\sum_{i=0}^{2M-2} r_i \text{ (\( \delta \) pixels left)}}{2M - \delta} \]
ADAPTIVE NOISE FILTERS

LOCAL NOISE REDUCTION FILTER:

\[ P^*_q = \frac{p}{q=\frac{(m*n)-1}{2}} - \frac{1}{q} \sum_{q=\frac{(m*n)-1}{2}}^{\frac{(m*n)+1}{2}} \left( P_q - \frac{(m*n)-1}{2} \right) - M_L \]

"Center" pixel under max m window
(If not an integer, decide which pixel to use)

\[ M_L = \frac{\sum P_q}{q=1} \] (Mean under window)

\[ \sigma_n^2 = \text{Variance of noise over image} \]

\[ \sigma_L^2 = \text{Variance of noise under max m window} \]

Limit \[ \frac{\sigma_n^2}{\sigma_L^2} \leq 1 \]

ADAPTIVE MEDIAN FILTER

- Changes size of window
- Attempts to find impulse noise and
  increases size, if an impulse is
  found, until a median value is found,
  max window size is reached,
- See text for details (p241).
- Performs well overall and generally
  produces sharper images than the
  non-adaptive median filter.
PERIODIC NOISE

PERIODIC NOISE CAN BE REDUCED USING FREQUENCY DOMAIN FILTERS THAT WE WILL DISCUSS LATER.

HYPERLINK TO FIG'S 5.7 - 5.14, NOISE FILTER EXAMPLES