Image Enhancement
- Image Enhancement - Spatial Domain
  - Smoothing filter
  - Median filter
- Convolution
  - 1D Discrete
  - 1D Continuous
  - 2D Discrete
  - 2D Continuous
- Sharpening filter

Salt & Pepper Noise

Neighborhood Averaging

\[
g(x,y) = \frac{1}{M} \sum_{(n,m) \in S} f(n,m)
\]

\( S = \) neighborhood of pixel \((x,y)\)
\( M = \) number of pixels in neighborhood \(S\)

Neighborhoods:
- 3 x 3
- 5 x 5
**Neighborhood Averaging - Example**

3 x 3 Average 5 x 5 Average 7 x 7 Average Median

**Convolution**

\[ (A \ast B)(x) = \sum A(i)B(x-i) \]

Example:

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
\end{array}
\]

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
5 & 6 & 7 & 8 \\
\end{array}
\]

\[
\begin{array}{cccc}
3 & 2 & 1 & 1 \\
2 & 3 & 4 & 5 \\
\end{array}
\]

**What happens near the edges?**

Option 1: Zero padding

Option 2: Wrap around

Option 3: Reflection
**Why one image is reflected in the convolution:**

With reflection:

\[ \begin{array}{cccccc} 0 & 0 & 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array} \]

Without reflection:

\[ \begin{array}{cccccc} 1 & 2 & 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array} \]

Reflection is needed so that convolution is commutative:  
\[ A \ast B = B \ast A \]

**Convolution - Continuous Case**

\[ (f \ast g)(x) = \int f(a)g(x-a) \, da \]

**Convolution - 2 Dimensions**

\[ (A \ast B)(x,y) = \sum_{i} \sum_{j} A(i,j) B(x-i, y-j) \]

Convolution - 2D Continuous Case:  
\[ (f \ast g)(x,y) = \iint f(a,b) g(x-a, y-b) \, da \, db \]
Grayscale Convolution - Example

A \ast B

Convolution Properties

- Complexity:
  - Assume \( A \) is \( n \times n \) and \( B \) is \( k \times k \) then
    \( A \ast B \) takes \( O(nk^2) \) operations.
  - \( A \ast B = B \ast A \)
  - \( (A \ast B) \ast C = A \ast (B \ast C) \)
    - If \( B \) and \( C \) are \( k \times k \) then
      \( (A \ast B) \ast C \) takes \( O(2nk^2) \) operations.
    - However \( A \ast (B \ast C) \) takes \( O(k^4 + nk^2) \) operations, which is faster if \( k \ll n \).
- Separability
  - In some cases it is possible to decompose \( B \) \( (k \times k) \) into \( B = C \ast D \) where \( C \) is \( 1 \times k \) and \( D \) is \( k \times 1 \).
  - In such a case \( A \ast B \) takes \( O(nk^2) \) while \( (A \ast C) \ast D \) takes \( O(2nk) \).

\[
\begin{bmatrix}
1 & -1 \\
-1 & 1
\end{bmatrix} \ast
\begin{bmatrix}
1 & -1 \\
-1 & 1
\end{bmatrix} =
\begin{bmatrix}
0 & 0 \\
0 & 0
\end{bmatrix}
\]

Mask Constraints

- Image Average
  - In order to preserve the overall average of \( A \), the sum of \( B \)'s elements should equal 1

\[
\begin{bmatrix}
A & B
\end{bmatrix}
\]

If \( W_1 + W_2 + W_3 = 1 \) then \( Av(A) = Av(A \ast B) \)

\[
\delta(x,y) = \begin{cases} 
1 & \text{if } x = x_0 \text{ and } y = y_0 \\
0 & \text{otherwise}
\end{cases}
\]

\[
A(x,y) \ast \delta(x-x_0, y-y_0) = A(x-x_0, y-y_0)
\]
Convolution Masks - Example: The Delta Kernel

\[ A(x,y) \ast \delta(x-x_0, y-y_0) = A(x-x_0, y-y_0) \]

\[ \delta(x-x_0, y-y_0) = \begin{cases} 1 & \text{if } x=x_0 \text{ and } y=y_0 \\ 0 & \text{otherwise} \end{cases} \]

\[ A(x,y) = A(x,y) \]

Grayscale Smoothing

Grayscale averaging = convolution with:

\[
\begin{array}{cccccc}
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\
\end{array}
\]

\[
\begin{array}{cccc}
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
1/9 & 1/9 & 1/9 \\
\end{array}
\]

"Soft" Averaging: Convolution with a Gaussian

Discrete case:

\[
\begin{array}{cccc}
1/8 & 2/8 & 1/8 \\
2/8 & 4/8 & 2/8 \\
1/8 & 2/8 & 1/8 \\
\end{array}
\]

A separable kernel

Normal vs Gaussian Grayscale Smoothing

Original

Noisy image

3 X 3 Average

3 X 3 Gaussian Average

5 X 5 Average

5 X 5 Gaussian Average

7 X 7 Average

7 X 7 Gaussian Average
**Median Filtering**

\[ S = \text{neighborhood of pixel (x,y)} \]

New value at (x,y) = \text{median} \{ f(k(x,y)) \} 
\( (x, y) \in S \)

\[ 10, 10, 20, 20, 25, 25, 30, 30, 250 \]

\[ \text{median} \]

Median + Average: average the k central values.

\[ 10, 10, 20, 20, 25, 25, 30, 30, 250 \]

\[ 24 \]

\[ \text{median} \]

\[ (x, y) \in S \]

**Median vs Average Filtering**

- 3 X 3 Average
- 5 X 5 Average
- 7 X 7 Average
- Median

**Multiple Median Filtering**

- Median x 2
- Median x 4
- Median x 6
- Median x 8
- Median x 7

- Salt & Pepper Noise
- Large Noise
<table>
<thead>
<tr>
<th>Original</th>
<th>Salt &amp; Pepper Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median Filter</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>Salt &amp; Pepper Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median Filter</td>
<td></td>
</tr>
<tr>
<td>Oriented Median Filter</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Salt &amp; Pepper Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4 Average</td>
</tr>
</tbody>
</table>
Oriented Filtering - Example

Directional Smoothing

Define oriented masks:

\[ V_\theta \]

Choose neighborhood with smallest variance and replace pixel value with the average of that neighborhood.

Directional Smoothing - Example
Directional Smoothing - Example

Original + Noise  3x3 Average

Directional Smoothing (2x5, 5x2, diagonalx2)

---

Sharpening

- A sharpening filter is applied in order to enhance edges and fine details (high frequency) in an image:
  - Assume $A^*G$ is a smoothing filtering.
  - $A^*(\delta - G)$ contains the fine details of the image.
  - $A^*(1+\lambda(\delta - G)) = A^*S(\lambda)$ amplifies fine details in the image.
  - The parameter $\lambda$ controls the amount of amplification.

$$G = \begin{pmatrix}
0 & \frac{1}{\pi} & 0 \\
\frac{1}{\pi} & \frac{1}{\pi} & \frac{1}{\pi} \\
0 & \frac{1}{\pi} & 0
\end{pmatrix}$$

$$S(\lambda) = \begin{pmatrix}
0 & -\frac{1}{\pi} & 0 \\
-\frac{1}{\pi} & \frac{1}{\pi} & -\frac{1}{\pi} \\
0 & -\frac{1}{\pi} & 0
\end{pmatrix}$$

---

Ringing effect in edge enhancement

---

---

---

---

---

---

---

---

---
How can we enhance such an image?

**Solution: Image Representation**

\[
\begin{array}{cccc}
1 & 1 & 0 & 0 \\
3 & 5 & 7 & 2 \\
0 & 3 & 5 & 0 \\
0 & 0 & 0 & 0
\end{array}
\]

\[
\begin{array}{cccc}
2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{array}
\]

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

**Frequency Domain**

Map of "Sizes and Orientations"

Evaluating an Image in terms of "Sizes":

- Large
- Small
- Thin & Horizontal
- Vertical