Summary

- Mask Operators
  - masks
  - Correlation and convolution
- Filters
  - Mean
  - Special filters: Sobel.
  - Edges
  - Median
Point Operators versus Mask operators

Original image

Output Image

Original Image: neighbors of the pixel

Output Image

Correlation or Convolution?

Convolution kernel
The filter mask: application

Correlation

\[ g(i, j) = \sum_{k,l} f(i + k, j + l)h(k, l) \]

Convolution

\[ g(i, j) = \sum_{k,l} f(i - k, j - l)h(k, l) = \sum_{k,l} f(k, l)h(i - k, j - l) \]

Example: Correlation

Mask

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-1</td>
</tr>
</tbody>
</table>

2D- Impulse

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Correlation

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**Example: Convolution**

### Mask

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2D-Impulse

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Convolution

Symmetric Mask

$$h(n, m) = h(-n, -m), \quad n, m = -K \ldots K$$

**Correlation vs Convolution?**

---

**Low-Pass Filtering Masks**

### Symmetric Masks

- Goal: reduce the noise, reduce details. **BLURS** the image
Low-Pass Filtering and size of masks

With increasing size of the Mask more smoothing (blurring)

Other Low-Pass: Gaussian Filters

\[ G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \]
Sharpening filters: gradient

- To emphasize intensity transitions
- The gradient is the quantitative measure of local variation
- Given the image \( f(x,y) \), gradient is a vector defined as

\[
\nabla f(x, y) = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]^T = [g_x, g_y]^T
\]

The gradient vector: magnitude and angle

Gradient image is \( \approx |g_x| + |g_y| \)

Possible discrete approximations to horizontal derivative

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
\end{array}
\quad \text{or} \quad
\begin{array}{ccc}
-1 & 0 & 1 \\
\end{array}
\]

Sharpening Masks: High-pass filters

Sobel masks are operators to compute discrete derivatives

\[
\begin{array}{ccc}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{array}
\quad \text{or} \quad
\begin{array}{ccc}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{array}
\]

horizontal \quad \text{vertical}
Example: derivative images

Original image

Sobel: horizontal

Gradient image

Sobel: vertical

Second-order filter and Laplacian

Laplacian is defined as

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

A discrete approximation is

$$\nabla^2 f(n, m) = f(n+1, m) + f(n-1, m) + f(n, m+1) + f(n, m-1) - 4 f(n, m)$$
Second order Derivative Filters:

Influence of Diagonal neighbors

Example: Laplacian

Original

After Laplacian (top left mask)
Non-linear filtering: Median Filter

Replace the value of the pixel by the median of its \( K \times K \) neighbors

Example: Mean versus Median

\[
\begin{array}{ccc}
30 & 10 & 20 \\
10 & 250 & 20 \\
25 & 10 & 30 \\
\end{array}
\]

\[\text{Mean} = 45\]

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

\[\text{median}\]

\textbf{FIGURE 3.37} (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a \( 3 \times 3 \) averaging mask. (c) Noise reduction with a \( 3 \times 3 \) median filter. (Original image courtesy of Mr. Joseph E. Pascante, Lexi, Inc.)
Border Effects

- zero: set all pixels outside the source image to 0 images;
- constant (border color): set all pixels outside the source image to a specified border value;
- repeat edge pixels
- mirror: reflect pixels across the image edge

Separable Filters

- Filters that can be explained as

\[ H = vh^T \]

\( v \) and \( h \) are vectors with 1D impulse responses.

\[
\begin{bmatrix}
-1 \\
0 \\
1
\end{bmatrix},
\begin{bmatrix}
1 \\
2 \\
1
\end{bmatrix} \Rightarrow vh^T = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix}
\]

Sobel: is a separable mask.
Application of separable filters: two step 1D based convolutions.

- The 1-D filter $h$ is applied to all rows of the image.

- The 1-D filter $v$ is to all columns of the image resulting from previous step.

Complexity: 2D versus 1D filtering

- Direct application of 2D masks ($K \times K$)

  $K^2$ \textit{(add – multiply) operations per pixel}

- Application of 1D convolution to rows and then to columns

  $2K$ \textit{(add – multiply) operations per pixel}
Matlab commands: filtering

• Filtering operations:
  • `imfilter`: applying either correlation of convolution (default correlation)
  • `filter2`: applying correlation
  • `conv2`: 2D convolution but also 1D in both directions (rows and columns)

  *Filtering commands can be applied with different alternatives to deal with border effects*

• Designing masks (filters)
  • `fspecial`: design specified 2D masks

Bibliography

• Gonzalez and Woods- Digital Image Processing using MATLAB (part of the figures of slides).
• Richard Szeliski- Computer Vision: Algorithms and Applications (chapter 3).

  *(available online: http://szeliski.org/Book/)*