Images

- photometric aspects of image formation
- gray level images
- point-wise operations
- linear filtering

Image model

- Analog intensity function
- Temporal/spatial sampled function
- Quantization of the gray levels
- Point sets
- Random fields
- List of image features, regions

Mathematical tools

- Analysis
- Linear algebra
- Numerical methods
- Set theory, morphology
- Stochastic methods
- Geometry, AI, logic

Basic Photometry

Radiance - amount of energy emitted along certain direction
Irradiance - amount of energy received along certain direction
BRDF - bidirectional reflectance distribution
Lambertian surfaces - the appearance depends only on radiance, not on the viewing direction
Image intensity for a Lambertian surface

\[ I(x) = \gamma R(y); \]
Challenges

Images

• Images contain noise – sources sensor quality, light fluctuations, quantization effects

Image Noise Models

• Additive noise: Most commonly used
  \[ \hat{I}(x, y) = I(x, y) + n(x, y) \]
• Multiplicative noise:
  \[ \hat{I}(x, y) = I(x, y) \cdot n(x, y) \]
• Impulsive noise (salt and pepper):
  \[ I(i, j) = \begin{cases} \hat{I}(i, j) & \text{if } x < l \\ l_{\text{min}} + y(l_{\text{max}} - l_{\text{min}}) & \text{if } x \geq l \end{cases} \]

Additive Noise Models

• Noise Amount: SNR = \( \frac{\sigma_I}{\sigma_n} \)
• Gaussian - Usually, zero-mean, uncorrelated

Uniform
How can we reduce noise?

- Image acquisition noise due to light fluctuations and sensor noise can be reduced by acquiring a sequence of images and averaging them.
- Solution – filtering the image

Discrete time system

- maps 1 discrete time signal to another
  \[ g(x) = T(f(x)) \]
- Special class of systems – linear, time-invariant systems
  Superposition principle
  \[ T(af(x) + bf(x)) = aT(f(x)) + bT(f(x)) \]
  Shift (time) invariant – shift in input causes shift in output
  \[ g[x] = T[f[x]] \rightarrow g[x - x_0] = T[f[x - x_0]] \]
  Examples
The output of a linear system is related to the input and the transfer function via convolution:

\[ f[x] = \sum_{k=-\infty}^{\infty} f[k] \delta[x-k]. \]

\( \delta[x] \) is the unit impulse - if \( x = 0 \) it's 1 and zero everywhere else.

Every discrete time signal can be written as a sum of scaled and shifted impulses.

The output of the linear system can be written as:

\[ g[x] = \sum_{k=-\infty}^{\infty} f[k] h[x-k] \]

**Notation:**

\[ g[x] = f[k] * h[x] \]

**Averaging filter**

\[ g[x] = \sum_{k=-\infty}^{\infty} f[k] h[x-k] \]

**Averaging box filter**

\[ h[x] = \frac{1}{3} [1, 1, 1] \]

**Averaging filter**

Center pixel weighted more:

\[ h[x] = [0.25, 0.5, 0.25] \]

**2D convolution - next**

**Convolution in 2D**

\[ g[x, y] = \sum_{k=-\frac{H}{2}}^{\frac{H}{2}} \sum_{l=-\frac{W}{2}}^{\frac{W}{2}} f[k, l] h[x-k, y-l] \]
How big should the mask be?

- The bigger the mask,
  - more neighbors contribute.
  - smaller noise variance of the output.
  - bigger noise spread.
  - more blurring.
  - more expensive to compute.
Limitations of averaging

- Signal frequencies shared with noise are lost, resulting in blurring.
- Impulsive noise is diffused but not removed.
- The secondary lobes of the sinc let noise into the filtered image.
- It spreads the noise, resulting in blurring.
- Impulsive noise is diffused but not removed.

Frequency Domain Interpretation:

\[
Sinc(u) = \frac{\sin u}{u}
\]

Low-pass filter

Gaussian Filter

- A particular case of averaging
  - The coefficients are samples of a 1D Gaussian.
  - Gives more weight at the central pixel and less weights to the neighbors.
  - The further away the neighbors, the smaller the weight.

\[
g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}}.
\]

Sample from the continuous Gaussian

Frequency Domain Interpretation

Gaussian filter is the only one that has the same shape in the space and frequency domains. There are no secondary lobes - i.e. a truly low-pass filter.
How big should the mask be?

- The std. dev of the Gaussian $\sigma$ determines the amount of smoothing.
- The samples should adequately represent a Gaussian.
- For a 98.76% of the area, we need

$$m = 5\sigma$$

$$5 \cdot (1/\sigma) \leq 2\pi \Rightarrow \sigma \geq 0.796, \ m \geq 5$$

5-tap filter

$$g[x] = \{0.136, 0.6065, 1.00, 0.606, 0.136\}$$

Image Smoothing

- Convolution with a 2D Gaussian filter

$$I(x, y) = I(x, y) * a(x,y) = I(x, y) * a(x) * a(y)$$

- Gaussian filter is separable, convolution can be accomplished as two 1-D convolutions

$$I(x, y) = I(x, y) * a(x) * a(y) = \sum_{k=-\frac{m}{2}}^{\frac{m}{2}} I[k]g[x-k]g[y-j]$$

Non-linear Filtering

- Replace each pixel with the MEDIAN value of all the pixels in the neighborhood.
- Non-linear
- Does not spread the noise
- Can remove spike noise
- Expensive to run

Example:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>10</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>9</td>
<td>99</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

$$I, O = \text{sort}(10,11,10,9,11,10,9,10)$$

9,9,10,10,10,10,11,11