Image Filtering

Outline
- The definition of filtering
  - Linear filters vs. non-linear filters
- Linear filter:
  - Kernel
  - Convolution
- Implementations:
  - for general filters
  - for incremental filters
  - for separable filters

Filtering
- Definition:
  - The process of replacing a pixel’s intensity with a value based on some functions of its nearby pixels’ intensities
- Classify by effect:
  - Smoothing filters
  - Sharpening filters
- Classify by function used:
  - Linear filters
  - Non-linear filters

Example of Filtering
- Characteristic:
  - Each output pixel uses a different set of neighbors
  - Most of the neighbors used for the current pixel are also used for the next pixel
  - For some filters, it is possible to derive a more efficient implementation

Linear vs. Non-linear
- Linear filters:
  - Calculate the linear combination of the intensities of neighboring pixels
  - Can be expressed as convolution with a kernel
  - Can be analyzed in frequency (Fourier) domain
- Non-linear filters:
  - Involve more complex calculation
  - Cannot be expressed as convolution
  - Cannot be analyzed in frequency domain

Linear Convolution
- For each pixel in digital image:
  - Kernel is aligned with the pixel
  - Multiplicative sum is calculated and normalized
  - The result is stored in the corresponding pixel of the result image
  - Process is repeated for all pixels
Kernel

- Kernel keeps the coefficients used in linear combination
- Width & height are odd so that the kernel has a center
- Can be considered as an image of $(2W+1) \times (2W+1)$
- $W$ determines how many nearby pixels are involved
- Normalized so that the sum of all entries equals 1

$$K = \frac{1}{|K|} \sum_{u=-W}^{W} \sum_{v=-W}^{W} K[u][v]$$

Convolution

- Continuous function:
  $$G(x,y) = F(x,y) \ast K(u,v) = \int_{u=-W}^{W} F(x-u,y-v) K(u,v) du dv$$
- Discrete function:
  $$G[x,y] = F[x,y] \ast K[u,v] = \sum_{u=-W}^{W} \sum_{v=-W}^{W} F[x-u,y-v] K[u,v]$$
- Properties:
  - Commutative: $a \ast b = b \ast a$
  - Associative: $(a \ast b) \ast c = a \ast (b \ast c)$
  - Distributive: $a \ast (b + c) = (a \ast b) + (a \ast c)$

Handling the Boundaries

- Different approaches can be used
  - Ignore the boundary — the output image will contain a white/black border
  - Assume that pixels that are outside of the input image have a fixed intensity, i.e., zero
  - Assume that a pixel that is outside of the input image has the same intensity as its closest neighbor that is in the image

$$F[-u,q] = F[0,q], F[width+u,q] = F[width-1,q]$$
$$F[p,-v] = F[p,0], F[p,\text{height}+v] = F[p,\text{height}-1]$$

Implementation for General Filters

- Number of calculations for each pixel:
  - $(2W+1)^2$ multiplications
  - $(2W+1)^2$ additions
- Overall complexity for an image with $M \times N$ pixels:
  - $O(M \times N \times W^2)$

Separable Property

- Some kernels can be represented as the product of two vectors
  - $K(u,v) = K_x(u) \cdot K_y(v)$
- The 2D convolution can be decomposed into two 1D convolutions
  $$G(x,y) = F(x,y) \ast (K_x(u) \cdot K_y(v)) = (F(x,y) \otimes K_x(u)) \otimes K_y(v)$$
  $$G[p,q] = \sum_{u} (K_x[u] \cdot \sum_{v} (F[p-u,q-v]))$$

Implementation for Separable Filters

- Number of calculations for each pixel:
  - $2(2W+1)^2$ multiplications
  - $2(2W+1)^2$ additions
- Overall complexity:
  - $O(M \times N \times W)$
Incremental Property

- For some kernels, the subtraction below produce a sparse matrix
  \[
  \begin{bmatrix}
  0 & k_{w-1,w} & \cdots & k_{w-1,0} \\
  \vdots & \ddots & \ddots & \ddots \\
  0 & k_{w-1,w} & \cdots & k_{w-1,0}
  \end{bmatrix}
  \begin{bmatrix}
  0 & k_{w-1,w} & \cdots & k_{w-1,0} \\
  \vdots & \ddots & \ddots & \ddots \\
  0 & k_{w-1,w} & \cdots & k_{w-1,0}
  \end{bmatrix}
  \]

- The intensity of the next pixel in the scanline can be more efficiently calculated through updating the intensity of the current pixel.

Implementation for Incremental Filters

- Number of calculations for each pixel:
  - \( L \) multiplications & additions
  - \( L \): the number of non-zero entries in the matrix after subtraction
  - Overall complexity:
    - \( O(M \times N \times L) \)

Example Filters I

- Original
- Emboss
- Glass Edge
- Foil Engrave

Example Filters II

- Original
- Emboss
- Glass Edge
- Foil Engrave