

Lecture 1:

Image transformation

Let \mathcal{I} = Collection of images of size N and range of intensity $[0, M]$.

$$= \{ f \in M_{N \times N}(\mathbb{R}) : 0 \leq f(i, j) \leq M ; 1 \leq i, j \leq N \}$$

(for simplicity, assume f is a square image; can be general $N_1 \times N_2$ image)

Image transformation = $\mathcal{O} : \mathcal{I} \rightarrow \mathcal{I}$ (transform one image to another)

Imaging problems

- ① Find a suitable transformation $T \Rightarrow g := T(f)$ becomes good!

- ② Given a ^{noisy image} g and transformation T , find original clean image f .

$$g = T(f) + n$$

known known unknown unknown (Inverse problem)

Definition: (Linear image transformation)

$\mathcal{O}: \mathcal{I} \rightarrow \mathcal{I}$ is linear $\Leftrightarrow \mathcal{O}(af + g) = a\mathcal{O}(f) + \mathcal{O}(g)$ for $\forall f, g \in \mathcal{I}; \forall a \in \mathbb{R}$

Take $f \in \mathcal{I}$. Let

$$f = \begin{pmatrix} f(1,1) & \dots & f(1,N) \\ f(2,1) & & f(2,N) \\ \vdots & f(i,j) & \vdots \\ f(N,1) & \dots & f(N,N) \end{pmatrix} = \sum_{i=1}^N \sum_{j=1}^N \begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & f(i,j) & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix}$$

Let $g = \mathcal{O}(f)$. Assume \mathcal{O} is linear, then:

$$g(\alpha, \beta) = \left[\sum_{x=1}^N \sum_{y=1}^N \mathcal{O} \left(\begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & f(x,y) & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} \right) \right]_{\alpha, \beta}$$
$$= \sum_{x=1}^N \sum_{y=1}^N f(x,y) h(x, \alpha, y, \beta) \quad \text{where}$$

$$h(x, \alpha, y, \beta) = [\mathcal{O}(P_{xy})]_{\alpha, \beta}; \quad P = \begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} \begin{matrix} \leftarrow x^{\text{th}} \\ \downarrow y^{\text{th}} \end{matrix}$$

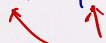
Remark: $h(x, \alpha, y, \beta)$ = how much input value at (x, y) influence the output value at (α, β) .



Pixel (x, y) affecting pixel (α, β) by a weight $h(x, \alpha, y, \beta)$.

Definition: (Point spread function)

$h(\cdot, \alpha, \cdot, \beta)$ is called the PSF at (α, β) .



Fix α, β . Let x, y as variables!

Definition: (Shift-invariant)

Shift invariant $\Leftrightarrow h(x, \alpha, y, \beta) \stackrel{\sim}{=} h(\alpha - x, \beta - y)$ for $\forall 1 \leq x, y, \alpha, \beta \leq N$

Definition: (Convolution) Let $f, g \in \mathcal{I}$.

Convolution of f and $g \Leftrightarrow f * g(\alpha, \beta) \stackrel{\text{(def)}}{=} \sum_{x=1}^N \sum_{y=1}^N f(x, y) g(\alpha-x, \beta-y)$

(Assume f and g are periodically extended: $\begin{cases} f(x+iN, y+jN) = f(x, y) \\ g(x+iN, y+jN) = g(x, y) \end{cases} \forall i, j \in \mathbb{Z}$)

Theorem: PSF is shift-invariant \Rightarrow the operator \mathcal{O} is a convolution with the input image.

Proof: Let $g := \mathcal{O}(f)$. $g(\alpha, \beta) = \sum_{x=1}^N \sum_{y=1}^N f(x, y) \underbrace{h(x, \alpha, y, \beta)}_{\sim h(\alpha-x, \beta-y)}$
 $= f * h(\alpha, \beta)$

Remark:

- $f * h = h * f$ (exercise)
- Convolution is important for understanding image blur.

Definition: (Separable)

Separable $\Leftrightarrow h(x, \alpha, y, \beta) = h_c(x, \alpha) h_r(y, \beta)$ for $\forall 1 \leq x, y, \alpha, \beta \leq N$.