Image sharpening in the frequency domain

Goal: Enhance image so that it shows more obvious edges.

Method 1: Laplacian masking

Recall that : $\Delta f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$.

In the discrete case, $\Delta f(x,y) \approx f(x+1,y) + f(x,y+1) + f(x,y-1) + f(x-1,y) - 4f(x,y)$

or $\Delta f \approx p * f$ where $p = \begin{pmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{pmatrix}$

We can observe that $-\Delta f$ captures the edges of the image (leaving other region zero)

$$\therefore \text{Shapen image} = f + \overbrace{(-\Delta f)}^{\text{add more edges}}$$

In the frequency domain: $\mathrm{DFT}(g) = \mathrm{DFT}(f) - \mathrm{DFT}(\overset{p*f}{\overset{\shortparallel}{\Delta f}})$

$$= \mathrm{DFT}(f) - c\,\mathrm{DFT}(p).\mathrm{DFT}(f)$$

$$\therefore \mathrm{DFT}(g) = [1 - \underset{c\,\mathrm{DFT}(p)}{\underset{\shortparallel}{H_{laplacian}(u,v)}}]\,\mathrm{DFT}(f)(u,v)$$

Method 2: Unsharp masking

Idea: Add high-frequency component

Definition: Let f = input image (blurry)

Let f_{smooth} = smoother image (using mean filter / Gaussian filter etc)

Define a sharper image as:

$$g(x,y) = f(x,y) + k\left(f(x,y) - f_{smooth}(x,y)\right)$$

When $k=1$, the method is called unsharp masking.

When $k>1$, the method is called highboost filtering.

In the frequency domain, let $DFT(f_{smooth})(u,v) = \underbrace{H_{LP}}_{\text{Low-pass filter}}(u,v)\, DFT(f)(u,v)$

Then: $DFT(g) = \left[1 + k(1 - H_{LP}(u,v)\right] DFT(f)(u,v)$

Image denoising in the spatial domain

Definition: Linear filter = modify pixel value by a linear combination of pixel values of local neighbourhood.

Example 1: Let f be an $N \times N$ image. Extend the image periodically. Modify f to $\tilde{f}$ by:

$$\tilde{f}(x,y) = f(x,y) + 3f(x-1,y) + 2f(x+1,y).$$

This is a linear filter.

Example 2: Define

$$\tilde{f}(x,y) = \frac{1}{4}\left(f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)\right)$$

This is also a linear filter.

Recall: The discrete convolution is defined as:

$$I * H(u,v) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} I(u-m, v-n)H(m,n)$$

(Linear combination of pixel values around (u,v))

Therefore, **Linear filter is equivalent to a discrete convolution**.

Geometric illustration

$$\begin{tikzcd} I \arrow[d, "* H"] \arrow[drr, phantom, "\cong"] & & \hat{I} \arrow[d, "\times \hat{H}"] \\ \tilde{I} \arrow[rr, shift left, "DFT"] & & \hat{\tilde{I}} \arrow[ll, shift left, "iDFT"] \end{tikzcd}$$

(spatial) (frequency)

Example 3: In Example 1, if f is defined on $[-M, M] \times [-N, N]$, then:

$$\tilde{f} = f * H$$

where

$$H = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 0 \end{pmatrix}$$

In Example 2, $\tilde{f} = f * H$ where

$$H = \frac{1}{4}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

H is called the <u>filter</u>

Commonly used filter (linear)

- **Mean filter:**

$$H = \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

(Here, we only write down the entries of the matrix for indices $-1 \leq k, l \leq 1$ for simplicity. All other matrix entries are equal to 0.)

This is called the *mean filtering with window size* 3×3.

- **Gaussian filter:** The entries of H are given by the Gaussian function $g(r) = exp\left(-\frac{r^2}{2\sigma^2}\right)$, where $r = \sqrt{x^2 + y^2}$.

Properties of linear filtering

- **Associativity:** $A * (B * C) = (A * B) * C$
- **Commutativity:** $I * H = H * I$
- **Linearity:**

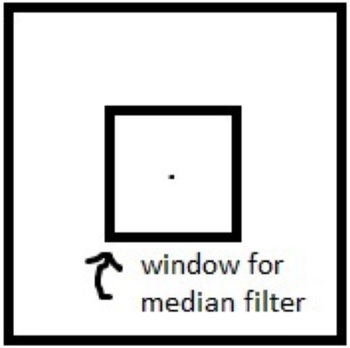
$$(s \cdot I) * H = I * (s \cdot H) = s \cdot (I * H)$$
$$(I_1 + I_2) * H = (I_1 * H) + (I_2 * H)$$

Remark: Convolution of Gaussian with a Gaussian is also a Gaussian

$\therefore$ Successive Gaussian filter = Gaussian filter with larger σ.

Non-linear spatial filter

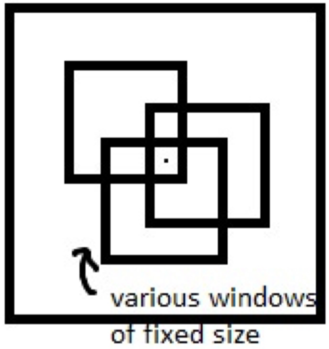
- **Median filter**



Take a window with center at pixel (x_0, y_0). Update the pixel value at (x_0, y_0) from $I(x_0, y_0)$ to $\tilde{I}(x_0, y_0) = \text{median}(I \text{ within the window})$

Example 4: If pixel values within a window is $0, 0, 1, 2, 3, 7, 8, 9, 9$, then the pixel value is updated as 3 (median).

- **Edge-preserving filter**



- **Step 1:** Consider all windows with certain size around pixel (x_0, y_0) (not necessarily be centered at (x_0, y_0));
- **Step 2:** Select a window with minimal variance;
- **Step 3:** Do a linear filter (mean filter, Gaussian filter and so on).

- <u>Non-local mean filter</u>

Let g be a $N \times N$ image.

$\left.\begin{array}{l} X = (x, y) \\ X' = (x', y') \end{array}\right\}$ Two pixels.

Define: $S_X = \{(x+s, y+t) : -a \le s, t \le a\}$; $S_{X'} = \{(x'+s, y'+t) : -a \le s, t \le a\}$

Define: $g_X = g|_{S_X}$ and $g_{X'} = g|_{S_{X'}}$.

$\underbrace{(2a+1)}_{m} \times \underbrace{(2a+1)}_{m}$ image

Let $\tilde{g}_X$ = smoothed image of g_X by Gaussian smoothing

$\tilde{g}_{X'}$ = smoothed image of $g_{X'}$ by Gaussian smoothing.

Define the weight: $w(X, X') = e^{\frac{-\|\tilde{g}_X - \tilde{g}_{X'}\|_F^2}{h^2}}$

noise level parameter

(small when X and X' are far away)

far away in term of small images

$$\boxed{\begin{array}{l}\text{Non-local mean filter of } g: \\ \hat{g} = \dfrac{\sum_{X' \in \text{image domain}} w(X, X')\, g(X')}{\sum_{X' \in \text{image domain}} w(X, X')}\end{array}}$$

Image denoising by solving Anisotropic heat diffussion

Consider the PDE:

$$(*) \quad \frac{\partial I(x,y,\sigma)}{\partial \sigma} = \sigma \left[\frac{\partial^2 I(x,y,\sigma)}{\partial x^2} + \frac{\partial^2 I(x,y,\sigma)}{\partial y^2} \right] = \sigma \, \nabla \cdot (\nabla I)$$

($\nabla \cdot$ = divergence ; $\nabla \cdot (v_1, v_2) = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}$) ($\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$)

Then: $g(x,y,\sigma) = \frac{1}{2\pi \sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ satisfies $(*)$.

<u>Observation</u>: We'll see that Gaussian filter is approximately solving $(*)$.

Given an image $I(x,y)$ (Assume I is continuously defined on the whole 2D domain)

Gaussian filter = convolution of I with the Gaussian function:

$$\tilde{I}(x,y,\sigma) = I * g(x,y,\sigma) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x-u, y-v)\, I(u,v)\, du\, dv$$

(Analogous to discrete convolution) $= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v;\sigma)\, I(x-u, y-v)\, du\, dv$

$$\therefore \frac{\partial \tilde{I}}{\partial \sigma} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial g(u,v,\sigma)}{\partial \sigma} I(x-u, y-v)\, du\, dv$$

$$= \sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial^2 g(u,v;\sigma)}{\partial u^2} I(x-u, y-v)\, du\, dv + \sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\partial^2 g}{\partial v^2}(u,v;\sigma)\, I(x-u,y-v)\, du\, dv$$

$$= \sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v;\sigma) \frac{\partial I}{\partial x^2}(x-u, y-v)\, du\, dv + \sigma \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v;\sigma) \frac{\partial I}{\partial y^2}(x-u, y-v)\, du\, dv$$

$$= \sigma\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v;\sigma)\, I(x-u, y-v)\, du\, dv.$$

$$\left(\text{Using the fact: } \frac{\partial}{\partial x}(g * f) = \frac{\partial g}{\partial x} * f\right)$$

$$= \sigma\, \nabla \cdot (\nabla \tilde{I})(x,y,\sigma)$$

$\therefore$ Gaussian filter = solving "diffussion" eqt !!

Anisotropic diffusion for edge-preserving Image denoising

General "diffusion" eqt :

$$\frac{\partial I(x,y;\sigma)}{\partial \sigma} = \nabla \cdot \big(\underbrace{K(x,y)} \nabla I(x,y;\sigma) \big)$$

- controls the rate of diffusion
- Smaller K = smaller diffusion at (x,y)

Edge detector : Edge of an image can be detected by: $|\nabla I(x,y)|$.

If (x,y) is on the edge, $|\nabla I(x,y)|$ is big.

If (x,y) is in the interior region, $|\nabla I(x,y)| \approx 0$

$\therefore$ Suitable $K(x,y)$ to preserve edge:

1. $K(x,y) = \dfrac{1}{|\nabla I(x,y)| + \varepsilon^2}$ Avoid singularity

2. $K(x,y) = e^{-|\nabla I(x,y;\sigma)|/b}$

$\therefore$ The denoising problem can be written as:

$$\frac{\partial I(x,y;\sigma)}{\partial \sigma} = \nabla \cdot \left(e^{-\frac{|\nabla I(x,y;\sigma)|}{b}} \nabla I(x,y;\sigma) \right)$$

In the discrete case, we solve:

$$I^{n+1}(x,y) - I^{n}(x,y) = \mathcal{D}_1 \left(e^{-\frac{|\nabla I^{n}(x,y)|}{b}} \mathcal{D}_2 \, I^{n}(x,y) \right)$$

$\mathcal{D}_1$ = matrix approximating $\nabla \cdot$

$\mathcal{D}_2$ = matrix approximating ∇

Recall:

$$\frac{\partial I}{\partial x}(x,y) \approx \underbrace{I(x+1,y) - I(x,y) \text{ etc}}_{\text{linear operator}}$$

$\therefore$ Starting with $I^{0}(x,y) = I(x,y)$,

we iteratively modify $I^{n}(x;y)$.

Such a process is called <u>Anisotropic diffusion image denoising.</u>