Enhancement, Restoration, Conversion

Local vs. Global

Local: Only use neighborhood information

$$\boxed{\begin{array}{ccc} x & x & x \\ x & \bullet(x,y) & x \\ x & x & x \end{array} \quad \begin{array}{c} nxn \quad window \\ h(i,j) \end{array}}$$

Spatial Smoothing: Noise removed and reduction of effects due to under sampling.

Simplest: $$g(i,j) = \frac{1}{M} \sum_{s} f(i,j)$$

S = M – pixel neighborhood of points surrounding (and perhaps including (x,y))

S is usually rectangular, e.g. ***nxn square***

nxn window

$$\boldsymbol{g}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\boldsymbol{i}=-\boldsymbol{n}/2}^{\boldsymbol{n}/2} \sum_{\boldsymbol{j}=-\boldsymbol{n}/2}^{\boldsymbol{n}/2} \boldsymbol{h}_{\boldsymbol{sm}}(\boldsymbol{i},\boldsymbol{j}) \quad \boldsymbol{f}(\boldsymbol{x}+\boldsymbol{i},\boldsymbol{y}+\boldsymbol{j})$$

$\boldsymbol{h}_{\boldsymbol{sm}}(\boldsymbol{i},\boldsymbol{j}) =$ smoothing function

e.g. $= \dfrac{1}{n^2}$

This is convolution

Take $\boldsymbol{n} = 3$

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

exactly averaging!!

Remark: This has continuous version.
Nonlinear
$\boldsymbol{I}_{\boldsymbol{t}} = \Delta \boldsymbol{I} /$linear let equation

$$I\left(x,y,t\right)=G_t * I\left(x,y,0\right)$$

$$G_t=\frac{1}{\sqrt{2\pi t}}\,e^{-\left(\frac{x^2+y^2}{t^2}\right)}$$

Can see blurring very easily:
I will choose an odd in order for window center to be on sampling grid. Shift origin:

$$\boldsymbol{k}=\boldsymbol{i}+\boldsymbol{n}/2$$
$$\boldsymbol{l}=\boldsymbol{i}+\boldsymbol{n}/2$$

Fourier transform is:

$$\boldsymbol{H}_{\boldsymbol{sm}}(\boldsymbol{u},\boldsymbol{x})=\frac{1}{\boldsymbol{N}}\sum_{\boldsymbol{k}=0}^{\boldsymbol{n}-1}\sum_{\boldsymbol{l}=0}^{\boldsymbol{n}-1}\boldsymbol{h}_{\boldsymbol{sm}}\left(\boldsymbol{k},\boldsymbol{l}\right|\exp[-\boldsymbol{j}2\Pi(\boldsymbol{uk}+\boldsymbol{vl})/\boldsymbol{N}])$$

Algebra:

$$\boldsymbol{Hsm}(\boldsymbol{u},\boldsymbol{x})=\frac{1}{\boldsymbol{N}}\sum_{\boldsymbol{k}=0}^{\boldsymbol{n}-1}\sum_{\boldsymbol{l}=0}^{\boldsymbol{n}-1}\frac{1}{\boldsymbol{n}^2}\exp[-\boldsymbol{j}2\Pi\boldsymbol{uk}/\boldsymbol{N}]\cdot\exp[-\boldsymbol{j}2\Pi\boldsymbol{vl}/\boldsymbol{n}]$$

Separable transform

Define:

$$\begin{aligned}
\boldsymbol{H}_1(\boldsymbol{u}) &= \frac{1}{\boldsymbol{n}}\sum_{\boldsymbol{k}=0}^{\boldsymbol{n}-1}\exp\left[-\boldsymbol{j}2\Pi\boldsymbol{u}\,{}^{\boldsymbol{k}}\!/\!_{\boldsymbol{N}}\right]\\
\textbf{\textit{Set Z}} &= \boldsymbol{rxp}\left[\boldsymbol{j}2\Pi\,{}^{\boldsymbol{u}}\!/\!_{\boldsymbol{N}}\right]\\
\boldsymbol{H}_1(\boldsymbol{u}) &= \sum_{\boldsymbol{k}=0}^{\boldsymbol{n}-1}\boldsymbol{eZ}^{-\boldsymbol{k}}\ \text{(ignoring scales)}\\
&= \frac{1}{\text{1 - Z}^{-1}}-\frac{\boldsymbol{z}^{-\boldsymbol{n}}}{(1-\boldsymbol{Z}^{-1})}
\end{aligned}$$

$\boldsymbol{Z}$ – transfer of difference two steps ; one occurs at $\boldsymbol{k}=0$
other at $\boldsymbol{k}=\boldsymbol{n}$

$$=\frac{1-\boldsymbol{Z}^{-\boldsymbol{n}}}{1-\boldsymbol{Z}^{-1}}=\boldsymbol{Z}^{-(\boldsymbol{n}-1)/2}\,\frac{\boldsymbol{Z}^{\boldsymbol{n}/2}-\boldsymbol{Z}^{-\boldsymbol{n}/2}}{\boldsymbol{Z}^{1/2}-\boldsymbol{Z}^{-1/2}}$$

or

$$\boldsymbol{H}1(\boldsymbol{u}) = \exp\left[-\boldsymbol{j}\Pi\boldsymbol{u}\,{}^{(\boldsymbol{n}-1)}\!/\!_{\boldsymbol{N}}\right] \sin\left(\Pi\boldsymbol{un}/\boldsymbol{N}\right) / \sin(\Pi\boldsymbol{uN})$$

Similarly $\varsigma\boldsymbol{r}\ \boldsymbol{H}2(\upsilon), \int_1$

Plot Magnitudes

Edge Information Removal

Blurs

Exercise:
Take smoothing filter from Matlab or $x\upsilon \quad (blur)$.
Apply to image at various window sizes.
Hand in ext Wednesday.

Temporal Smoothing

$\hat{f}_i(x, y) = f_i(x, y) + m_i\ (x, y)$ zero mean time-uncorrelated noise poor.

$i = 1, \quad 2, \quad \ldots, \quad \mathrm{M}$ time changes due only + noise process

Sequence of images corrupted by noise.

$$g\left(\overset{x}{\theta}, y\right) = \frac{1}{\mathrm{M}} \sum_{ensenll} \hat{f}_i\ (x, y)$$

Reduce noise effects by reducing variance of (averaged) noise process.

Spatial Sharpening
Simplest approach
Smoothing = low-pass
Sharpening = high-pass

$g(x, y) = f(x, y) - f_{sm}(x, y)$ ←Smoothed version

Typically: f_{sm} is taken as local average of eight neighboring pixels surrounding $\varsigma, t\ \vartheta\tau$ including (x, y):

$$f_{sm}(x, y) = \frac{1}{8} \sum_{i=-1}^{i=1} \quad \sum_{j=-1}^{j=1} f(x + i,\ y + j)$$

$i \neq 0,\ j \neq 0$

$\therefore 3 \times 3$ window function for (a) is $\begin{bmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 1 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{bmatrix}$

Fourier Analysis:

$$\begin{aligned} \boldsymbol{G}(\boldsymbol{u},\boldsymbol{x}) &= \boldsymbol{F}(\boldsymbol{u},\boldsymbol{x}) - \boldsymbol{F}_{\boldsymbol{sm}}(\boldsymbol{u},\cup) \\ &= \boldsymbol{F}(\boldsymbol{u},\boldsymbol{x}) \underbrace{\left[1 - \underbrace{\frac{\boldsymbol{F}_{\boldsymbol{sm}}(\boldsymbol{u},\boldsymbol{x})}{\boldsymbol{F}(\boldsymbol{u},\boldsymbol{x})}}_{\boldsymbol{H}_2(\boldsymbol{u},\boldsymbol{x})} \right]}_{\boldsymbol{H}_{\boldsymbol{eq}}(\boldsymbol{u},\boldsymbol{v})} \end{aligned}$$

Nonlinear Operations
Not every operator is linear. Class of enhoncent called "rank" or "median".

Idea: Output image intensity of spatial location (x, y) is chosen on the best of the relative rank or intensity of pixels in neighborhood of (x, y).

Given N pixel intensities obtail over a local region, S, denoted or $f_1, \dots f_N$ order in increasing value:

$$\begin{aligned} R(x, y) &= \{f_1, \dots, f_N\} \\ & f_i \le f_{i\tau}1 \end{aligned}$$

Output intensity:
$$g(x, y) \underline{\underline{\Delta}}\ Rank_j \quad R(x, y)$$

Where $Rank_j$ is the intensity of the $0\_x_0\_x$ intensity it position or $rank \quad j$ in $R(x, y)$.

Ex.

i.
$$\begin{aligned} & j = 1 \qquad \text{yields min filter} \\ \mathrm{g}(\mathrm{x}, \mathrm{y}) &= \min R(x, y) \\ &= \{f(x, y) | (x, y) \in \delta\} \end{aligned}$$

ii. max filters is $j = N$:
$$\begin{aligned} g(x, y) &= \max R(x, y) \\ &= \max \{f(x, y) | x, y \in \delta\} \end{aligned}$$

iii. N is odd:
$$m = \frac{n+1}{2}$$

$$med\ R(x, y) \underline{\underline{\Delta}}\ Rank_{n+1}(R(x, y))$$

$V_e, nice$: Try in Matlab or
Suppose have image corrupted by spike-like noise.
Low-pass distributes equally.
Median removes noise without degraditive.

Not b two image functions yielding sample sequences $R_1\ al\ R_2$.

$$med\ (R_1(x, y) + R_2(x, y)) \neq med\ R_1(x, y)$$

$$+ med\ R_2(x, y)$$

However:

$$\boldsymbol{med}\ (\boldsymbol{KR}(x, y)) \quad = \quad \boldsymbol{K\ med}\ \boldsymbol{R}(x, y)$$

$$\boldsymbol{med}\ (\boldsymbol{K} + \boldsymbol{R}(x.y)) \quad = \quad \boldsymbol{K} \in \boldsymbol{med}\ \boldsymbol{R}(x, y)$$

Properties

1. Median filter reduces verioma of intenatis in the image.
2. Median filter preserve certain edge shapes.
   Preserves triple pulse
3. No new grey values generated

Edge letectise

$$f(x, y)$$

$$\nabla\ f(x, y) \quad = \quad \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial j} \end{bmatrix}$$

$$\frac{\partial f}{\partial x} \approx \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \overset{\text{convolition with}}{\leftrightarrow} [-1, \quad 1]$$

$$\frac{\partial f}{\partial y} \approx \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \overset{\text{convolition with}}{\leftrightarrow} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

How do those work:

$$f(x + \Delta x) = f(x) + \Delta x\ f^{1}(x) + \frac{(\Delta x)^2}{21}\ f^{11}(x) + \ldots$$

$$\therefore\ f(x + \Delta x) - f(x) = \Delta x\ f^{1}(x) + \frac{(\Delta x)^2}{21}\ f^{11}(5)$$

mean value theorm

$x \langle \varepsilon \langle x + \Delta x$

Define

$$D_1\,(x) = \frac{f(x + yx) - f(x)}{\Delta x}$$

Error $(D_1) \approx \vartheta(\Delta x)$

D_1 is a better approximation to the slope of $f(x)$ at the midpoint of the interval $[x,\, x + \Delta x]$.

$\therefore$ Using D_1 operator for $\frac{2f}{2x}$ and $\frac{2f}{2y}$ approximations results in approximations to the gradnet not at (x, y) but at different points in (x, y) plane.

$\therefore$ Use centered difference:

$$D_2(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$$

This has several nice proportions:

1. $D_2(x)$ and $D_2(y)$ due to centered routine both comprise good estimates for the respective directions of $f(x, y)$ of midpoints of interval, i.e. (x, y).

2. Error $(D_2) \cong \theta((\Delta x)^2)$

3. $D_2(x)$ and $D_2(y)$ maybe implanted by convolving image with nests.

$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Mery Vanitor

$$\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

Varits: 3x3

$$\underbrace{\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}}_{D_2\Delta(x)} \text{ and } \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}}_{D_2\Delta(y)}$$

"Smoothed" or "averaged" centered difference operations.

Other smoothed operates with a weighting that emphasizes the central pixel eveth

Sobel Weighting modes, denotes by $D_s(x)$ and $D_s(y)$ and givenly

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Third operator for gradient approximation is **Roberts**:

$$\begin{aligned} D_x\{(x, y)\} &= f(x + yx, y + \Delta y) - f(x, y) \\ D_-\{(x, y)\} &= f(x, y + \Delta y) - f(x + \Delta x, y) \end{aligned}$$

Note that this operation:

1. Is a variate of the $\boldsymbol{D}_1$ operation with derivates approximate along orthoyond orientation 45º at 135º in image plane;
2. Midpoint of both intervals used for approximation is the same point, i.e. it is the point $\left(x+\Delta x/_2,\ y+\Delta y/_2\right)$ located at its center of the rectangle found by its four points used;
3. Correspond to combination with rists

$$\boldsymbol{D}_{x}=\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \boldsymbol{D}_{-}=\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Roberts Cross Operation

$$\boldsymbol{Dx}=\boldsymbol{Dy}$$

$$\begin{aligned} \boldsymbol{f}_{edge}(x,y) &= \max\{|\boldsymbol{D}_t|, |\boldsymbol{D}_{-1}|\} \\ &= \max\{|\boldsymbol{f}(x,y+\boldsymbol{D}y)-\boldsymbol{f}(x+\boldsymbol{D}x,y)|, |\boldsymbol{f}(x+\boldsymbol{D}x,y+\boldsymbol{D}y)-\boldsymbol{f}(x,y)|\} \end{aligned}$$

Directionally oriented edge information

K