No :

06. Part 2

* Generating Spatial Filter Masks

In image processing, generating a spatial filter mask involves defining a set of coefficients based on the desired operation.

- Different types of filters exist.

1. linear filters
2. continuous function-based filters.
3. nonlinear filters.

linear filters

* Implemented as a sum of products.
* Ex: Mean (Averaging) Filter
  Gausian Filter
  Laplacian Filter
  Sobel Filter
  Prewitt Filter

No: Date: / /

2. Continuous Function-Based Filters

* Based on mathematical functions
e.g. Gaussian smoothing

$$g(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

- σ (sigma) controls the spread of the Gaussian function.
- The 2D Gaussian filter has a bell-shaped curve.

effect:

- A small σ result in slight blurring
- A large σ ~~re~~ causes more blurring.

CamScanner

No: Date: / /

2

3. Nonlinear Filters / order static filters

* Do not use a sum of products.

Effect:

The Max Filter helps remove dark noise spots from an image.

* In this use custom operations

Eg: Median Filter

Max Filter

Min Filter

Alpha-trimmed Mean Filter:

Smoothing Spatial Filters

Smoothing filters are used for blurring and noise reduction.

* Blurring:

* Helps remove small details before object extraction.

* Can connect small gaps in lines and curves.

* often applied in preprocessing steps to simplify images.

* Noice Reduction:
  * Reduces unwanted random dots or grany texture in an image.
  * Smoothing is done by changing each pixel's value based on the near by pixels. This helps reduce sharp changes in the image.
* can be doing using:
  - Linear filters - (Like Mean, Gaussian filters)
  - Non linear fillters (Median filter)

Smoothing Linear filters

01) Average / Mean filter

- It takes the average of nearby pixel values and replace the current pixel with that average.
- This makes image smoother by reducing sudden changes in brightness.
- Helps remove noise, as noise usually appears as sudden bright or dark spots.

* Effect of mean / average filter:

* Blur edges -

Edges in an image are places where there is a sudden change in brightness, like going from light to dark. This filter makes these edges softer and less sharp.

* Smoothing : false contours -
sometimes image have unnatural lines or bands (called false contours) because there aren't enough brightness level. This filter smooth these lines, making the image look more natural

* Removing Irrelevant Details
Removing small details in the image that are not important. This small details can be seen as noise. By removing them image looks cleaner.

- **Arithmetic mean filter**

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

* Imagine you have an image, and you pick a pixel at position (x, y)
* Around this pixel, you define a small rectangular area ($m \times n$ pixel), called a neighborhood.
* Sum $\sum g(s, t)$ means add up all pixel values in the neighborhood.
* The division by $m \times n$ gives the average value.
* The result $\hat{f}(x, y)$ is the new value of the pixel at (x, y) in the filtered image.

Ex:- We have 3×3 neighborhood of pixels around a point (x, y) in a grayscale image.

$$\begin{bmatrix} 50 & 20 & 30 \\ 40 & 10 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

$50 + 20 + 30 + 40 + 10 + 60 + 70 + 80 + 90 =$

450

$f(x,y) = 450/90 = 50$

filtered image

$$\begin{bmatrix} x & x & x \\ x & 50 & x \\ x & x & x \end{bmatrix}$$

Q2) Geomatric mean filter.

* The geomatric mean filter smooths an image while keeping edges and details cleaner than the arithmetic mean filter.
* It reduce the impact of very ~~height~~ bright or dark pixels by multiplying the surrounding pixel values taking their root instead of averaging them.

$$\hat{f}(x,y) = \left[\prod_{(s,t) \in S_{xy}} g(s,t) \right]^{\frac{1}{m \times n}}$$

m×n are the dimensions of the neighborhood (in this case 3×3 = 9 pixels)

ex.

$$\begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \end{bmatrix}$$

product of pixel value

$= 4 \times 5 \times 6 \times 1 \times 2 \times 3 \times 7 \times 8 \times 9$

$= 181440$

$= (181440)^{\frac{1}{9}}$

≈ 4.56

output →

$$\begin{bmatrix} x & x & x \\ x & 4.56 & x \\ x & x & x \end{bmatrix}$$

No.
Date.

03) Harmonic mean filter. bright spots

- Works well for salt noise but fails for pepper noise.
- Also does well for other kinds of noise such as Gaussian noise.

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

ex:- $\begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \end{bmatrix}$

$= \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9}$

≈ 3.92

$\hat{f}(x,y) = \frac{9}{3.92} \approx 2.30$

04) Contra Harmonic mean filter.

* Used for noise reduction
* Q controls the filter's behaviour:
  - positive Q removes pepper noise (black spot)
  - Negative Q removes salt noise (white spots)
  - Q = 0 make it the same as the arithmetic mean filter.

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{x,y}} g(s,t)^{Q}}$$

$g(s,t)$ = pixel values in the filter window.

Q = order of the filter

$f(x,y)$ = new filtered pixel value.

* for the salt noise (bright pixels, near 255):
  Use negative Q (e.g. Q = -1)
* for pepper noise (dark pixel, near 0):
  Use positive Q (e.g Q = +1)

* example to remove salt noise

$$\begin{bmatrix} 5 & 6 & 7 \\ 4 & 255 & 6 \\ 3 & 5 & 4 \end{bmatrix}$$ using formula Q = -1

$$f(x,y) = \frac{5+6+7+4+6+3+5+4}{\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{4}+\frac{1}{6}+\frac{1}{3}+\frac{1}{5}+\frac{1}{4}}$$

$$= \frac{40}{2.42} \approx 16.53$$

filtered Image:
(remove salt noise)

$$\begin{bmatrix} 5 & 6 & 7 \\ 4 & 16 & 6 \\ 3 & 5 & 4 \end{bmatrix}$$

* removing pepper noise :

$$\begin{bmatrix} 5 & 6 & 7 \\ 4 & 0 & 6 \\ 3 & 5 & 4 \end{bmatrix}$$

using the formula with $Q = +1$

$$f(x,y) = \frac{5^2+6^2+7^2+4^2+6^2+3^2+5^2+4^2}{5+6+7+4+6+3+5+4}$$

$$= \frac{212}{40} \approx 5.3$$

filtered image :
(remove pepper noise)

$$\begin{bmatrix} 5 & 6 & 7 \\ 4 & 5 & 6 \\ 3 & 5 & 4 \end{bmatrix}$$

* choosing the wrong value for Q when using the contra harmonic filter can have drastic resut.

big problems that make things worse instead of better

→ (33 page)

No : Date: / /

* A spatial averaging filter in which all coefficients are equal is called a "Box filter"

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

To maintain the brightness level.

* In spatial averaging filter, the goal is to smoothing or blurring an image by averaging the pixel values around each point. It gives more attention to the middle pixel increase than to the surrounding pixels, to reduce unwanted blurring while still smoothing the image.

e.g;
$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 2 \end{bmatrix}$$

5. Gaussian smoothing filter

- Gaussian filter is non-uniform law pass filter.
  It smooth an image by reducing high freqaency details but it doesn't blar everything equally.
- The number in the filter (kernal) are largest in the middle and get smaller as you move outward.
- Central pixel have a higher weighting than these on the surrounding pixels.

σ value එක ↑ වෙද්දී

- Larger values of σ (sigma) makes the blur spread out more, covering a bigger area.

ලොකු area එකක් blur වෙනවා

- kernal size must increase with increasing σ (sigma) to maintain the gaussian nature of the filter.

- σ අගය [illegible] [illegible]
[illegible] bell shape
එක [illegible] [illegible] [illegible],

* if you increase σ you need a bigger filter to fully capture the smooth bell-shaped distribution.

- Gaussion kernal coefficients depend on the value of σ (sigma)
- At the edge of the mask, coefficient must be close to 0, means they don't affect the image too much.
- The filter applies the same smoothing effect in all directions.
- Gaussian filter might not preserve image brightness (total brightness of the image might change slightly)

[illegible]

* Smoothing order static (Non linear) Filters

* arranging the pixels values in the image area are ~~convered~~ covered by the filter and selecting a specific ranked value (position value) to replace the central pixel.

* Useful nonlinear filters
  - Median filter
  - Max and Min filter
  - Midpoint filter

01. Median filter

* reduce certain typed of random noise with less blurring than the linear smoothing filters of similer size.
* provides excellent results when appling to reduce salt and pepper noise.

$$\hat{f}(x, y) = \underset{(s,t) \in S_{x,y}}{\text{median}} \{ g(s, t) \}$$

* If the details in the image ([illegible] image detail) are small, the filter can blur or remove them completely.
* This leads to the image becoming "fuzzy", meaning it loses some clarity or sharpness.
* The median filter is not very good at removing additive Gaussian Noise (A type of noise where random values are added to the image). in this case linear filters (like average filter) works better.

02.) Max filter

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{ g(s,t) \}$$

* Useful for finding brightest points in an image.
* Good for pepper noise.

03) Min filter.

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{ g(s,t) \}$$

* Useful for finding darkest points in an image.
* Good for salt noise.

04) Midpoint fiter.

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{ g(s,t) \} + \min_{(s,t) \in S_{xy}} \{ g(s,t) \} \right]$$

* Good for randon gaussian and uniform noise.

05) Alpha - trimmed Mean filter.

* Good for combination of salt and pepper and Gaussian noise,

$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t) \in S_{xy}} g_r(s,t)$$

* $f(x,y)$ represents the filtered pixel value at location (x,y)
* mn is the total number of pixels in the filter window.
* $g(s,t)$ represent the remaining mn-d pixels.
* d is the number of pixels to be trimmed (removed) from both the highest and lowest end of the pixel value range within the window.

* $\Sigma g(s,t)$ means you sum the remaining pixel values after trimming.
* when $d=0$: Arithmetic mean filter.
* we can delete the $d/2$ lowest and $d/2$ highest grey levels.

Ex. $\begin{bmatrix} 10 & 15 & 20 \\ 25 & 30 & 35 \\ 100 & 150 & 200 \end{bmatrix}$

* Sorting : 10, 15, 20, 25, 30, 35, 100, 150, 200
* if $d=4$, we removed $d/2 = 2$ lowest value = (10 & 15) and 2 highest value = (150 & 200)
* Remaining values :
  [20, 25, 30, 35, 100]
* Compute mean : $\frac{20+25+30+35+100}{5}$
  $= 42$

* $\begin{bmatrix} x & x & x \\ x & 42 & x \\ x & x & x \end{bmatrix}$