Area/Mask Processing Methods

(Trucco, Chapt 3)

- A pixel's value is computed from its old value and the values of pixels in its vicinity.

- More costly operations than simple point processes, but more powerful.

• What is a Mask?

- A mask is a small matrix whose values are called *weights*.

- Each mask has an *origin*, which is usually one of its positions.

- The origins of symmetric masks are usually their center pixel position.

- For nonsymmetric masks, any pixel location may be chosen as the origin (depending on the intended use).

1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

1
1
1
1

• Applying Masks to Images (filtering)

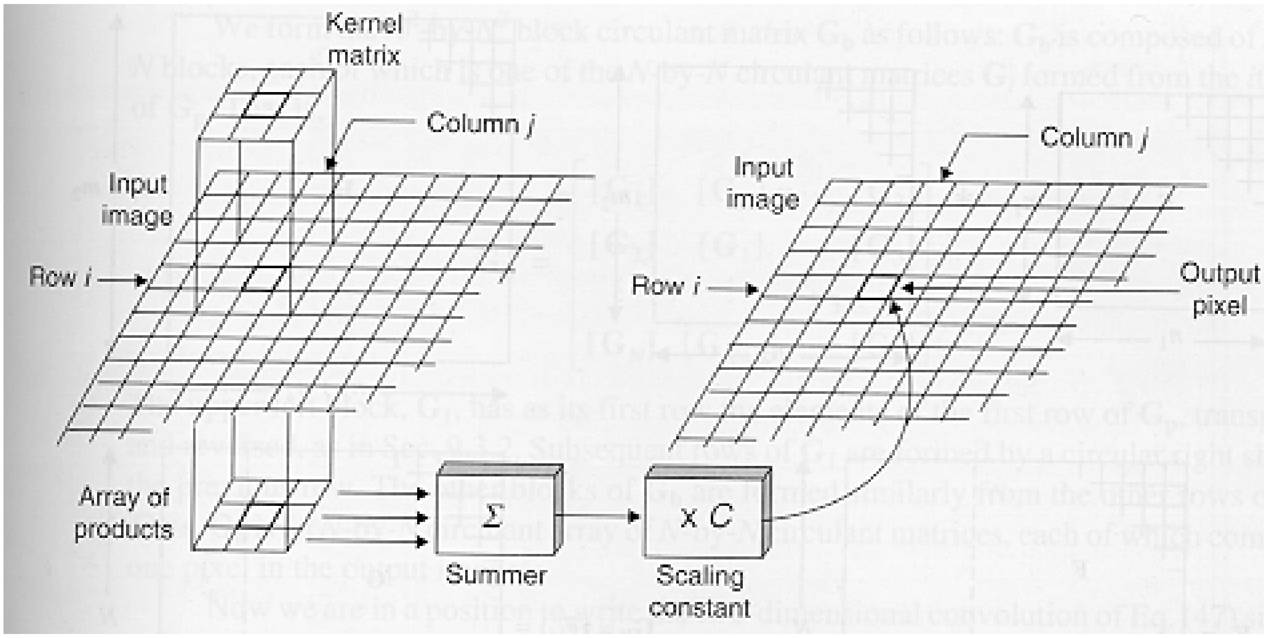
- The application of a mask to an input image produces an output image of the same size as the input.

Convolution

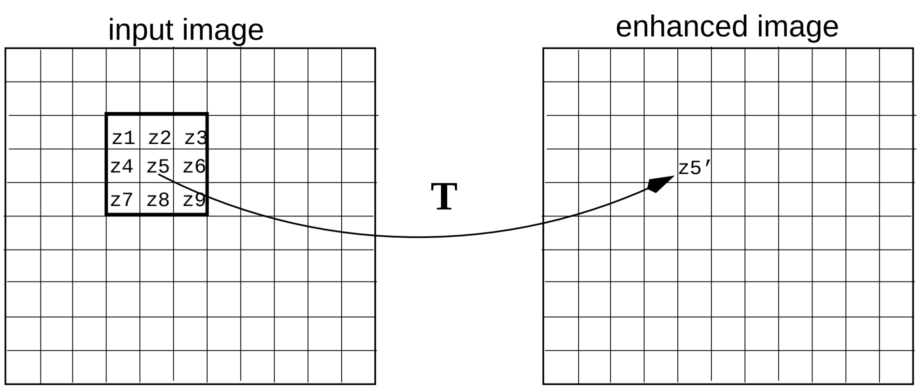
(1) For each pixel in the input image, the mask is conceptually placed on top of the image with its origin lying on that pixel.

(2) The values of each input image pixel under the mask are multiplied by the values of the corresponding mask weights.

(3) The results are summed together to yield a single output value that is placed in the output image at the location of the pixel being processed on the input.



Area or Mask Processing Methods



$g(x,y) = T[f(x,y)]$

T operates on a neighborhood of pixels

w1	w2	w3
w4	w5	w6
w7	w8	w9

`z5′ = R = w1z1 + w2z2 + ... + z9w9`

- Mathematical definition of -discrete- convolution:

$$g(i,j) = \sum_{k=-n/2}^{n/2} \sum_{l=-n/2}^{n/2} h(k,l)\, f(i-k, j-l)$$

(for a mask with odd dimensions)

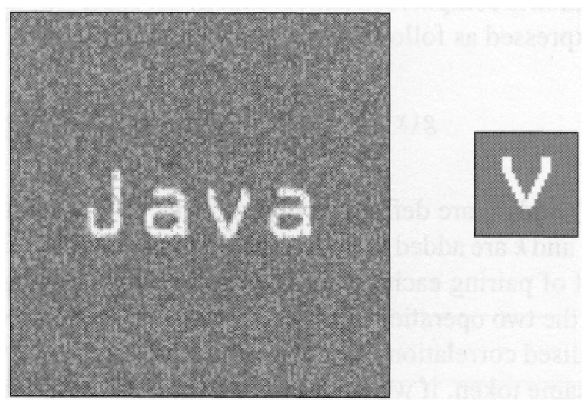
Cross Correlation

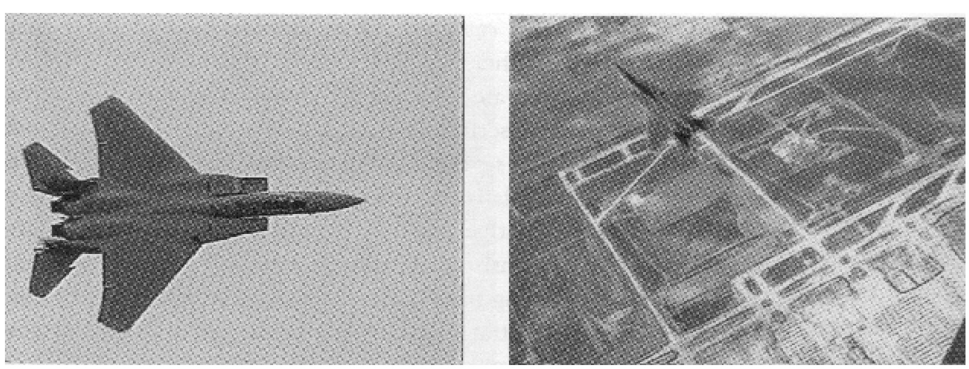
- Correlation translates the mask directly to the image without flipping it.

- It is often used in applications where it is necessary to measure the similarity between images or parts of images.

- If the mask is symmetric (i.e., the flipped mask is the same as the original one) then the results of convolution and correlation are the same.

$$g(i, j) = \sum_{k=-n/2}^{n/2} \sum_{l=-n/2}^{n/2} h(k, l) f(i+k, j+l)$$





Non-linear filtering

- Linear filters have the property that the output is a linear combination of the inputs.

- Filters which do not satify this property are called non-linear.

- *Erosion* and *Dilation* are examples of non-linear filters.

• Normalization of mask weights

- The sum of weights in the convolution mask affect the overall intensity of the resulting image.

- Many convolution masks have coefficients that sum to 1 (the convolved image will have the same average intensity as the original one).

- Some masks have negative weights and sum to 0.

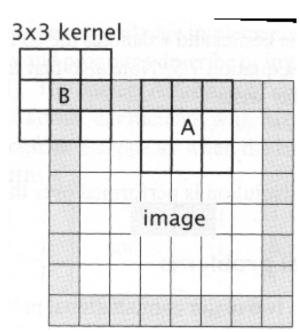
- Pixels with negative values may be generated using masks with negative weights.

- Negative values are mapped to the positive range through appropriate normalization.

• Practical problems

- How to treat the image borders?

- Time increases exponentially with mask size.



Smoothing (or Low-pass) filters

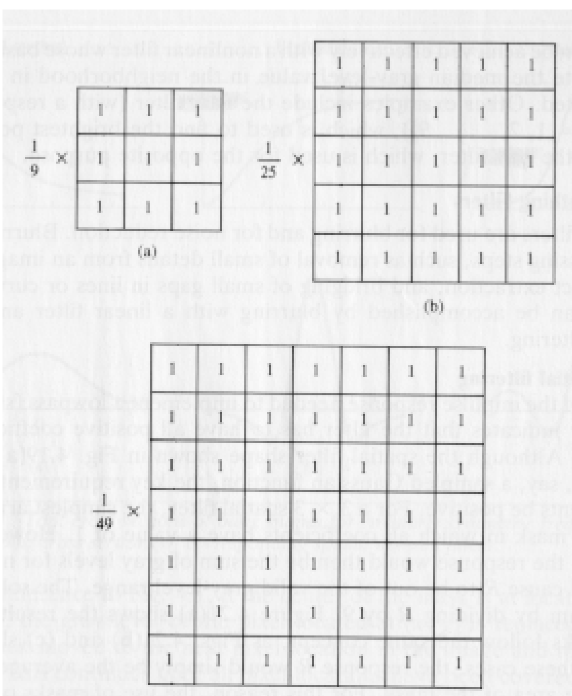
- Useful for noise reduction and image blurring.

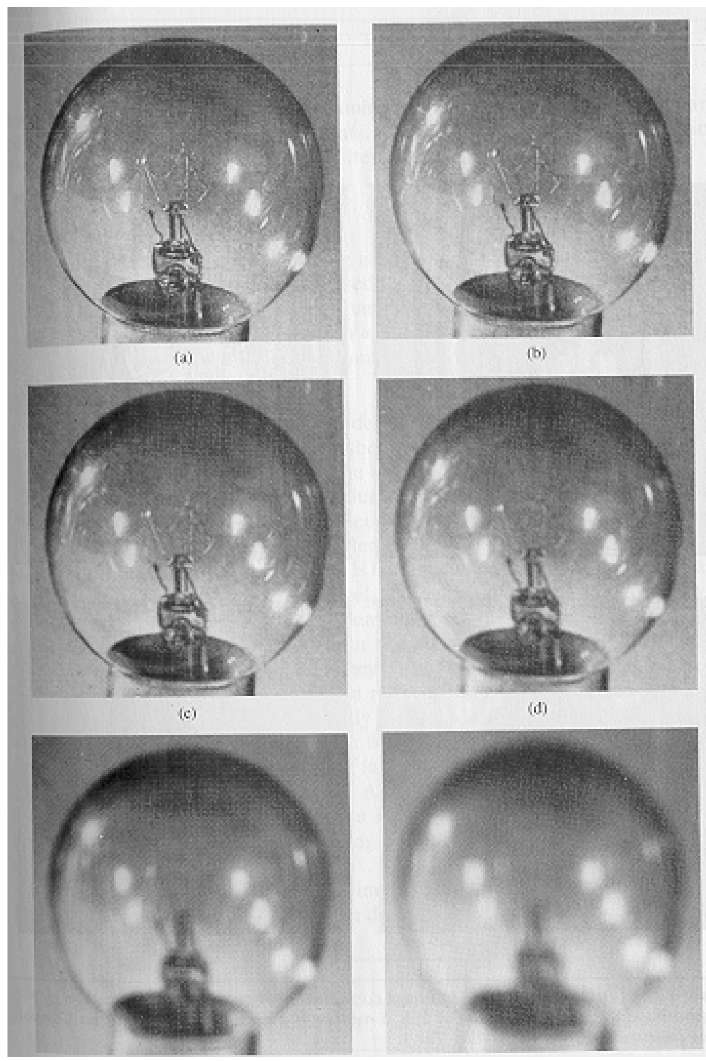
- It removes the finer details of an image.

• Averaging or Mean filter

  - The elements of the mask must be positive.

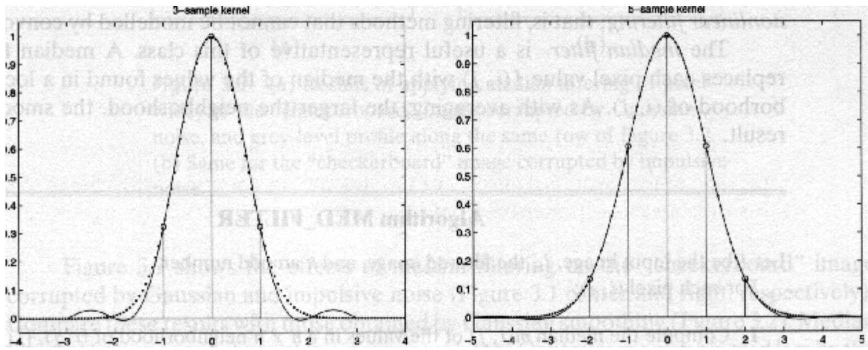
  - The size of the mask determines the degree of smoothing.





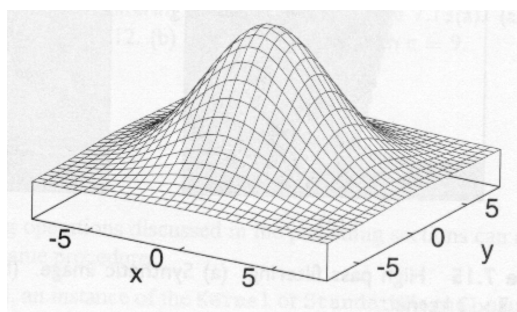
- **Gaussian (linear filter)** (from "Machine Vision" by Jain et al., Chapt 4 and Trucco Chapt 3)

  - The weights are samples from a Gaussian function.



(example using 1D Gaussian)

$$h(x, y) = exp[\frac{-(x^2 + y^2)}{2\sigma^2}]$$



  - The Gaussian's mask weights fall off to (almost) zero at the mask's edges.

  - Gaussian smoothing can be implemented efficiently thanks to the fact that the kernel is <u>separable:</u>

$$g(i, j) = \sum_{k=-n/2}^{n/2} \sum_{l=-n/2}^{n/2} h(k, l) f(i-k, j-l) =$$

$$\sum_{k=-n/2}^{n/2} \sum_{l=-n/2}^{n/2} exp[\frac{-(k^2 + l^2)}{2\sigma^2}] f(i-k, j-l) =$$

$$\sum_{k=-n/2}^{n/2} exp[\frac{-k^2}{2\sigma^2}] \sum_{l=-n/2}^{n/2} exp[\frac{-l^2}{2\sigma^2}] f(i-k, j-l)$$

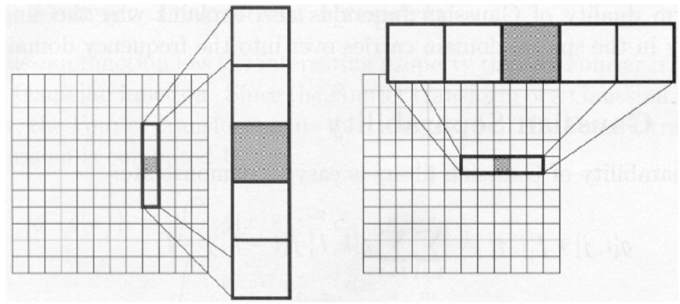
Algorithm

To convolve an image I with a nxn 2D Gaussian mask G with $\sigma = \sigma_g$

1. Build a 1-D Gaussian mask g, of width n, with $\sigma_g = \sigma_G$

2. Convolve each column of I with g, yielding a new image I_c

3. Convolve each row of I_c with g



- The value of σ determines the degree of smoothing.

- As σ increases, the size of the mask must also increase if we are to sample the Gaussian satisfactorily.

$$height = width = 5\sigma \text{ (subtends 98.76\% of the area)}$$

7 × 7 Gaussian mask

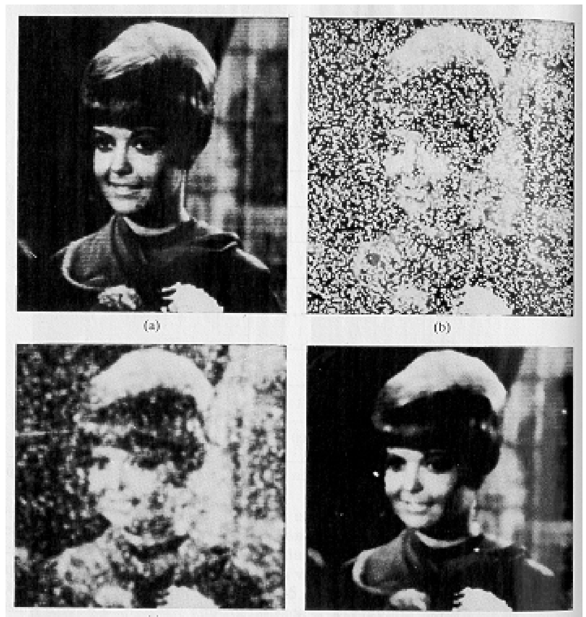
1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

15 × 15 Gaussian mask

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

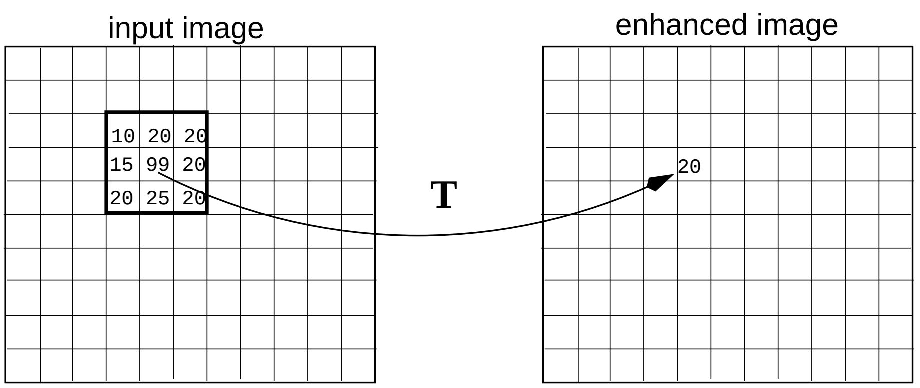
• Median filter (non-linear)

- Effective for removing "salt and pepper" noise (random occurences of black and white pixels).



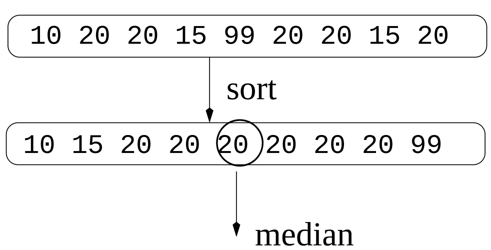
- Replace each pixel value by the median of the gray-levels in the neighborhood of the pixels

Area or Mask Processing Methods



$$g(x,y) = T[f(x,y)]$$

T operates on a neighborhood of pixels



Sharpening (or High-pass)

- It is used to emphasize the fine details of an image (has the opposite effect of smoothing).

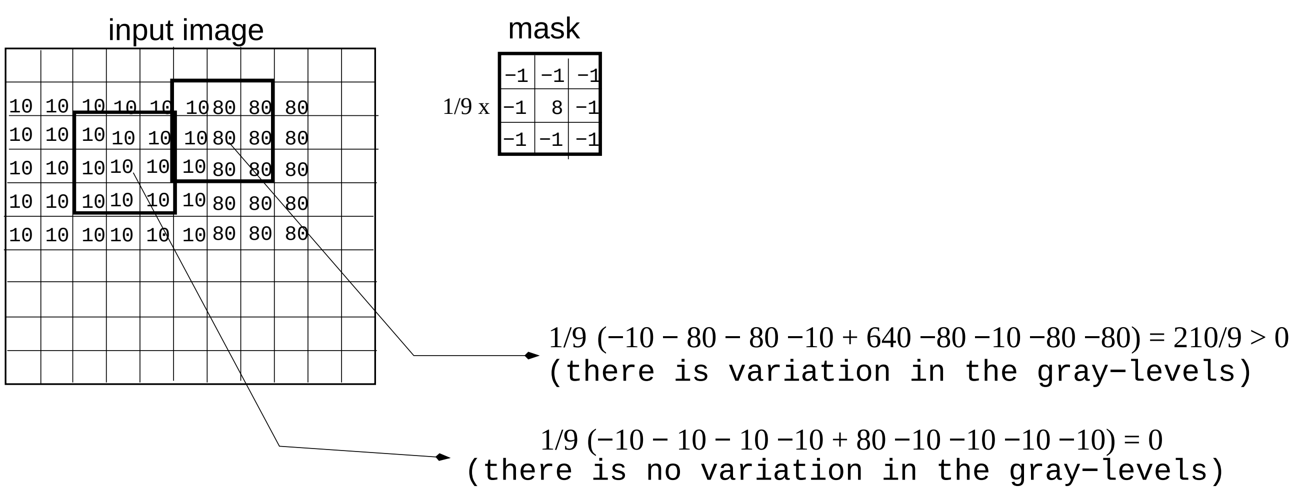
- Points of high contrast can be detected by computing intensity differences in local image regions.

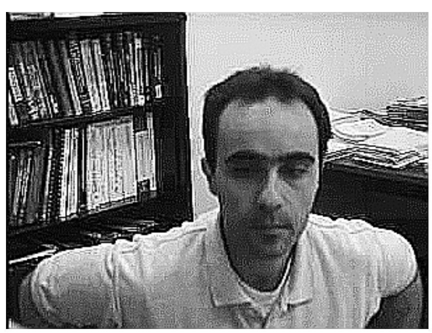
- The weights of the mask are both positive and negative.

- When the mask is over an area of constant or slowly varying gray level, the result of convolution will be close to zero.

- When gray level is varying rapidly within the neighborhood, the result of convolution will be a large number.

- Typically, such points form the border between different objects or scene parts (i.e., sharpening is a precursor step to edge detection).





• Sharpening using derivatives

- Computing the derivative of an image has as a result the sharpening of the image.

- The most common way to differentiate an image is by using the gradient.

$$grad(f) = \begin{pmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{pmatrix}$$

- The gradient can be approximated by *finite differences* which can be implemented efficiently as masks.

- Examples of masks based on gradient approximations with finite differences:

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

Prewitt masks

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

$$M_y \qquad\qquad M_x$$