Image enhancement (GW-Ch. 3)

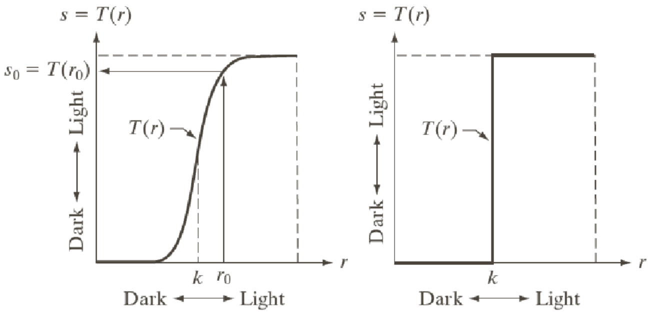
Process of improving image quality so that the result is more suitable for a specific application.

- contrast stretching
- histogram processing
- smoothing
- sharpening: spatial filtering for edge enhancement

Classification of image operations

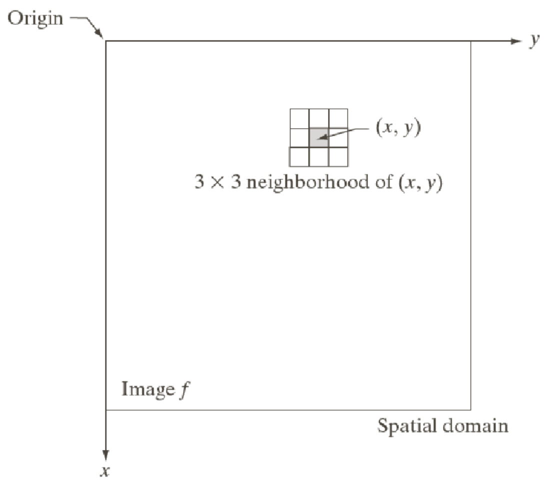
- Point operation: output pixel value depends only on corresponding input pixel (e.g. contrast stretch): $f_{out}(x,y) = \mathcal{O}\,(f_{in}(x,y))$ where $\mathcal{O}$ denotes the grey scale transformation (GST) function.
- Local operation: output pixel value depends on neighbourhood $\mathcal{B}(x,y)$ of input pixel: $f_{out}(x,y) = \mathcal{O}\,(\{f_{in}(x',y') : (x',y') \in \mathcal{B}(x,y)\})$.
- Global operation: output pixel value depends on all input pixels. Example: (discrete) Fourier transform.
- Geometric operation: spatial transformation (scaling, translation)

Point operations



(a) Contrast stretching. (b) Thresholding.

Neighbourhood operation



Transforming an image by moving a neighbourhood over the image.

Linear contrast stretching

- Used if features of interest occupy only a small range of the available grey levels
- Let the input image f_{in} have minimum and maximum grey level m and M ($M > m \geq 0$), respectively. Then the following operation stretches the image to full grey level range $[0..255]$:

$$f_{out}(x, y) = \frac{255}{M - m}\,(f_{in}(x, y) - m)\,.$$

Contrast stretch (normalize)


input

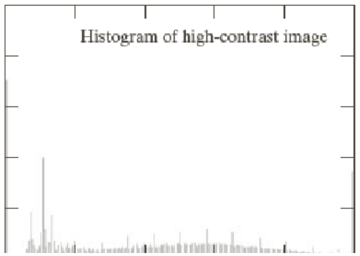

contrast stretched

Histogram processing

The histogram of the image I is defined as

$$h(m) = \#\{(r, c) \in D : I(r, c) = m\}$$

In words, $h(m)$ is the number of times the grey value m occurs in the image I.



Histogram equalisation

- Goal: flat histogram in the output, i.e., on average an equal number of pixels at each grey level.
- If the image has N pixels and grey level range $[0, L - 1]$, the output image will have $N/(L - 1)$ pixels at each grey level.
- The grey scale transformation achieving histogram equalisation is $\mathcal{O}(f(x, y)) = (L - 1)\,\mathsf{P}(f(x, y))$, where

$$\mathsf{P}(\ell) = \frac{1}{N}\sum_{m=0}^{\ell} h(m)$$

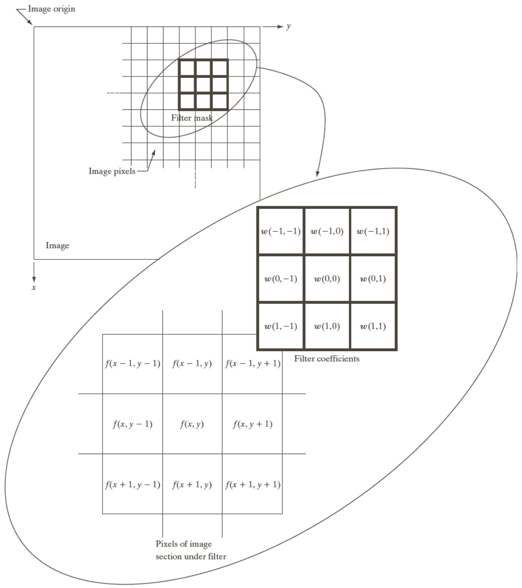
is the normalised cumulative histogram function.

Contrast stretching & histogram equalisation

(a) (b) (c)

(a): input image. (b): contrast stretch of (a). (c): histogram equalisation of (a).

Spatial filtering



Spatial filtering

- A filter kernel or filter mask is a set of coefficients $w(s,t)$ where (s,t) runs over small neighbourhood $\mathcal{N}$ of the origin: $\mathcal{N} = \{(s,t) : -a \leq s \leq a, -b \leq t \leq b\}$. Usually $\sum_{(s,t)\in\mathcal{N}} w(s,t) = 1$.
- Spatial filtering transforms an input image f to an output image g by moving the mask to each pixel (x,y) and summing the pixel values in the neighbourhood of multiplied by the corresponding filter coefficient:

$$\boxed{g(x,y) = \sum_{-a}^{a} \sum_{-b}^{b} w(s,t)\, f(x+s, y+t)}$$

Spatial filtering

- The expression

$$g(x,y) = (w \star f)(x,y) = \sum_{-a}^{a} \sum_{-b}^{b} w(s,t)\, f(x+s, y+t)$$

is called the correlation of w and f.

- The expression

$$g(x,y) = (w \ast f)(x,y) = \sum_{-a}^{a} \sum_{-b}^{b} w(s,t)\, f(x-s, y-t)$$

is called the convolution of w and f.

Spatial filtering

- Note that these operations can be easily mapped to one another:

$$g(x, y) = (w \,☆\, f)(x, y) = (\tilde{w} \star f)(x, y)$$

where $\tilde{w}(s, t) = w(-s, -t)$ is the mirrored version of w.

Smoothing filters

- Local averaging: lowpass filtering within a small neighbourhood or mask surrounding each pixel (discrete convolution).
- Order statistic (percentile) filters: rank the pixel values within the mask surrounding the center pixel (e.g., median filtering) for noise reduction without blurring of the edges.
- Morphological filters: general class of nonlinear filters.

Local averaging

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Left: uniform. Right: nonuniform.

Uniform filter in Matlab

```
function [g] = uniform (f)

A=im2double(f);       % convert image to double
nr=size(A,1);         % number of rows
nc=size(A,2);         % number of columns

% Shift A cyclically in four directions
A_up=[A(2:nr,:); A(1,:)];          % one row up
A_down=[A(nr,:); A(1:nr-1,:)];     % one row down
A_left = [A(:,2:nc) A(:,1)];       % one column to left
A_right = [A(:,nc) A(:,1:nc-1)]; % one column to right

B=0.2*(A+A_up+A_down+A_left+A_right); % uniform filter
g=im2uint8(B);                        % convert to 8-bit
```

Smoothing filters

(a) (b) (c)

(a): Original. (b): uniform filter. (c): percentile filter.

Sharpening filters

- Goal is to enhance fine details such as edges
- Can be performed by using highpass filters based upon spatial differentiation
- Can be implemented by discrete convolution with appropriate masks.

Prewitt operator

-1	-1	-1
0	0	0
+1	+1	+1

$\frac{\partial f}{\partial y}$ (discrete)

-1	0	+1
-1	0	+1
-1	0	+1

$\frac{\partial f}{\partial x}$ (discrete)

Kernels for the Prewitt operator.

Sobel operator

- Each point is convolved with two kernels, one for detecting horizontal edges and the other for vertical edges.
- Output of the operator is the maximum or square root of the two convolutions.

-1	-2	-1
0	0	0
+1	+2	+1

-1	0	+1
-2	0	+2
-1	0	+1

Kernels for the Sobel operator.

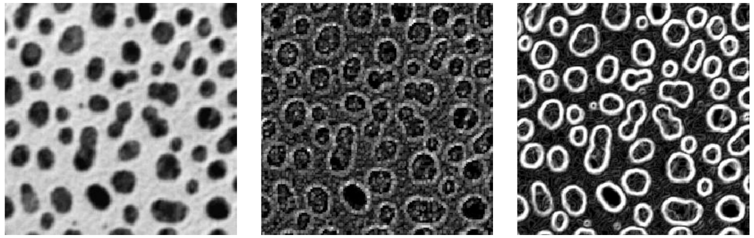
Laplace operator

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Laplacian convolution kernels (discrete 2^{nd} derivative)

Sharpening filters



(a) (b) (c)

Effect of sharpening filters.
(a): Original. (b): Laplace filter. (c): Sobel filter.