Digital Image Processing

Importance of digital image

1.Nowadays, multimedia is everywhere.

2.Digital image(or video stream) is the main form to present information.

3.Different forms of imaging.

What is digital image processing

1.**Image acquisition**

Automatic aperture correction and color balance:Human eyes can adapt the surrounding luminance dynamically, with very high dynamic range and speed;

Image reconstruction:Panaroma, Image warping, Image morphing.

2.**Display and print**

Automatic size adjustment; Color correction(CMYK vs. RGB).

3.**Storage and Transmission**

High efficient storing: less temporal-spatial cost; Convenient transmission: Lossless, efficient, and secure.

4.**Enhancement and restoration**

Image restoration: Inpainting; Improve visual effect: dehazing, retinex, deblur, super-resolution.

5.**Recognition and understanding**

Character recognition: OCR, object detection: face detection; Scence understanding: image retrieval, scene classification.

Course Schedule:

Chapter 1: Basic concept and programming introduction

Grayscale image and color image representation

Color space transfer

Image format

Data structure of image: some typical ones

Programming for image processing: An image in computer: array(s); Image file: its format and read/write; Access DIB: some Windows APIs; An image processing example for VC++; Image processing in Matlab: some typical in instructions.

Popular image processing softwares

Digital image representation

1.Digital image can be represented as a 2D array or matrix.
2.Each pixel in a grayscale image is represented by a byte(8 bits), which covers 256 degrees by [0, 255].
3.Each pixel in a color image is represented by 3 bytes(24 bits), which are for R(red), G(green), and B(blue), respectively.

Image format

1.Purpose:store image information.
2.Characteristic of image: pixel based, rectangle region, with lots of information.
3.Some image formats are highly related to operating systems: windows, unix, mac OS, etc. .
4.Encoding methods: without compression, lossless compression, Lossy compression.
5.Representative formats: BMP,JPEG, TIFF, GIF, PNG..

BMP format

1.One of the standard file formats in Windows system.
2.Sometimes .bmp is also saved as .dib file.
3.Mostly, BMP is organized in a non-compression way.
4.BUT, it supports image compression, e.g., rle format(run length encoding).

BMP file structure

structure
Image file header
Image information header
Palette
Image data

1.Image file header:

```
typedef struct tagBITMAPFILEHEADER{
   WORD bfType;
   DWORD bfSize;
   WORD bfReserved1;
   WORD bfReserved2;
   DWORD bfOffBits;
} BITMAPFILEHEADER, *PBITMAPFILEHEADER;
```

start	size	name	purpose
1	2	bfType	must always be set to 'BM' to declare that this is a .bmp-file.
3	4	bfSize	specifies the size of the file in bytes.
7	2	bfReserved1	must always be set to zero.
9	2	bfReserved2	must always be ser to zero.
11	4	bfOffBits	specifies the offset from the beginning of the file to the bitmap data.

bfType	Describe the file type. It must be OX4D42, namely, 'BM'
bfSize	Describe the bitmap file size with Bytes.
bfReserved1	Reserved, must be zero.
bfReserved2	Reserved, must be zero.
bfOffBits	Describe the offset from the beginning of the fileheader to the real image data with bytes. This parameter is necessary because the length of "BITMAPINFOHEADER" and "Palette" will change according to different situations. Such offset enables you to access the bitmap data quickly.

2.Image information header

```
typedef struct tagBITMAPINFOHEADER{
   DWORD biSize;
   LONG biWidth;
   LONG biHeight;
   WORD biPlanes;
   WORD biBitCount;
   DWORD biCompression;
   DWORD biSizeImage;
   LONG biXPelsPerMeter;
   LONG biYPelsPerMeter;
   DWORD biClrUsed;
   DWORD biClrImportant;
} BITMAPINFOHEADER, *PBITMAPINFOHEADER;
```

Type	Description
biSize	Number of bytes to define BITMAPINFOHEADER structure
biWidth	Image width(number of pixels)
biHeight	Image height(number of pixels).Note: Besides describing height, "biHeight" can be also denote whether the image is upright or not.(Positive->inverted, Negative->upright).Most of the BMP files are inverted bitmap, namely, biHeight > 0.
biPlanes	Number of planes.Always be 1.
biBitCount	Bits per pixel(Bits/pixel),which is 1, 4, 8, 16, 24 or 32.
biCompression	Compression type. Only non-compression is discussed here:BI_RGB.
biSizeImage	Image size with bytes. When biCompression=BI_RGB, biSizeImage=0.
biXPelsPerMeter	Horizontal resolution, pixels/meter.
biYPelsPerMeter	Vertical resolution, pixels/meter.
biClrUsed	Number of color indices used in the bitmap (0->all the palette items are used).
biCrImportant	Number of important color indices for image display. 0->all items are important.

3.Palette and bitmap data

Palette	N*4 bytes	For each item in the Palette, these FOUR bytes are for RGB values:1.byte for blue; 2.byte for green; 3.byte for red; 4.byte always ZERO
Bitmap data		Its size depends on image size and color depth. It stores the index number of palette, or RGB value, which depends on the color depth.

The RGBQUAD array

start	size	name	stdvalue	purpose
1	1	rgbBlue	-	specifies the blue part of the color.
2	1	rgbGreen	-	specifies the green part of the color.
3	1	rgbRed	-	specifies the red part of the color.
4	1	rgbReserved	-	must always be set to zero.

Here is an example in code C:

```
BITMAPFILEHEADER bmfh;
BITMAPINFOHEADER bmih;
RGBQUAD aColors[];
BYTE aBitmapBits[];
```

Basic principle of imaging

1.Imaging tools: Pocket Digital Cameras; Digital single lens reflection(DSLR) Cameras; Webcam; Mobile Phone.
2.Why not make the aperture as small as possible?
The quantity of light is too small.
Lead to diffraction if the aperture is too small.
Difficult to control.
3.A lens focuses light onto the film:There is a specific distance at which objects are "in focus" other points project to a "circle of confusion" in the image; Changing the shape of the lens changes this distance.
4.Changing the aperture size affects depth of field:A smaller aperture increases the range in which the object is approximating in focus.
5.The principle imaging process of DC:
When taking photograph, the light from the scene goes through the lens and reaches CCD.
When CCD being exposed, the photodiode is stimulated to release charges, and produces electrical

signals.
CCD controlling chip controls the electric flow through the controlling signal circuit in photoperceptive component. CCD will collect such electrical signals and output them to an amplifer.
After amplifying and filtering, the electrical signals reach ADC. And ADC transfer such electrical signals(continous) to digital ones(discrete), The value of digital is proportional to the intensity of electrical signal and voltage. These values are corresponding to those of an image.
But the above data cannot be treated as image directly. They will be further processed by DSP(digital signal processing). In DSP, color correction and white balance will be performed to obtain a qualified image, and the image will be encoded into the supported format and resolution, which can be stored as a image file.
After the above steps, the image file appears on your memory card and can be previewed.
6.Physical meaning of color:Colorful view is generated by the interactions between light and objects, e.g., reflection, scattering, transmission, absorption, and diffraction.

Physical meaning of color

Colorful view is generated by the interactions between light and objects, e.g., reflection, refraction, scatteing, transmission, absorption and diffraction.
Color can be divided into chromatic color and achromatic color.
Chromatic color means the monochrome color(red, yellow, blue, etc.) and their combination. The colorful objects reflect the light selectively to the light spectrum of different wavelengths. Hence, they present different color under white light.
Achromatic color, also called non-chromatic OR grayscale, means white, black and the gray intensities between them. The achromatic objects do not select the wavelength, so they are in neutral color.

Light and color

Color vision is the feeling of human brain when lights of different wavelengths reach retina. The visible wavelength lies between 390nm-780nm. Normally, human can identifies 120-180 colors including the 7 principle colors purple, blue, cyan, green, yellow, orange and red.
Retina is the most important part of human eyes, like the film in an camera, to capture the light and produce image. There are two kinds of vision cell on the retina:roads and cones.
Rods: ~100,000,000,sensitive to light but cannot identify different colors.
Cones: ~6,000,000-7,000,000, work under strong light, but can identify different colors.

Three-primary colors

Three-primary colors is a typical explanation of color vision mechanism.
It is believed that there are three kinds of cones or photopigments corresponding to RED, GREEN and BLUE's wavelengths.

When light of specific wavelength reaches human eyes, the corresponding cones are stimulated and cause the color vision(color perception) in human brain.
If three kinds of cones are stimulated equivalently, achromatism will happen.

Color vision basic

Commission internationale de l'eclairage(CIE), based on the existing experimental results, select 700nm(R), 541.6nm(G), and 435.8nm(B) as the three-primary colors, and the corresponding spectrum distribution of stimulation is computed and demonstrated.

Properties of color vision

Absolute color vision is not precise: Human identifies color by visual context.
Perceptive ability: Apart from the above, different people may have different perceptive abilities.
Weber's Law: The sensible difference(visible contrast) is proportional to the intensity of background.
Perception priority and sensitivity
Priority: Under the same setting, human notices first the hue(H) changes, then the saturation(S), then the lightness value(V).
Sensitivity: Human eyes are the most sensitive to the lightness changes and have the best resolution, which is responsible for the eyes' HDR capacity.

Definition of color space

Device dependent color space

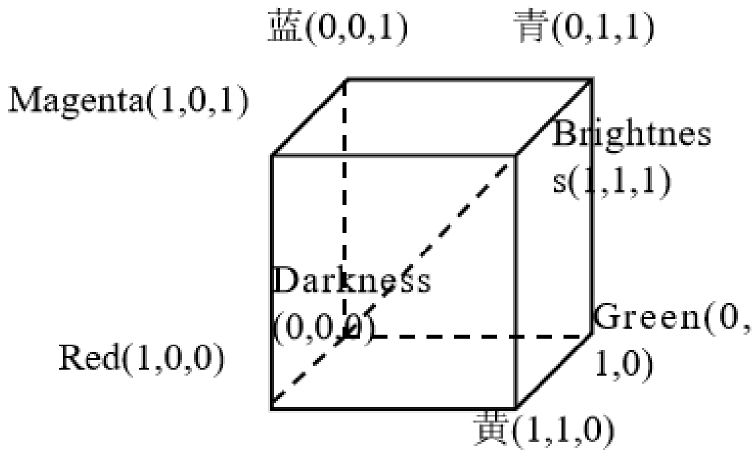
RGB, CMY, HSV

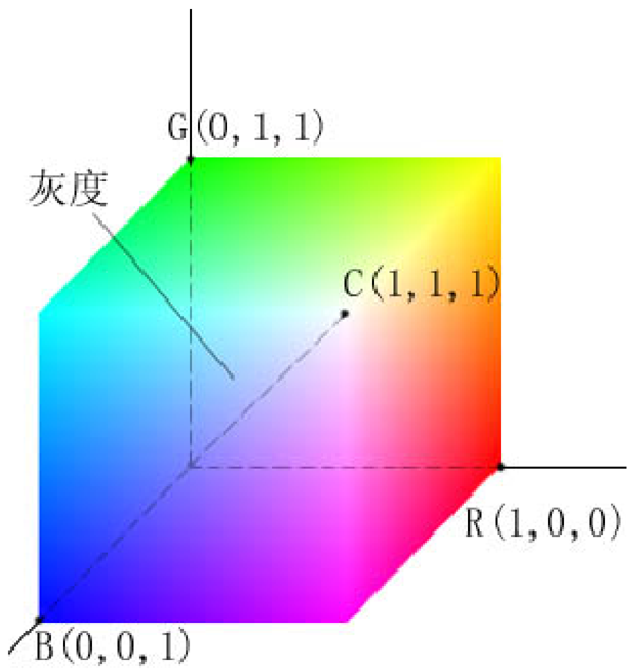
Device independent color space model

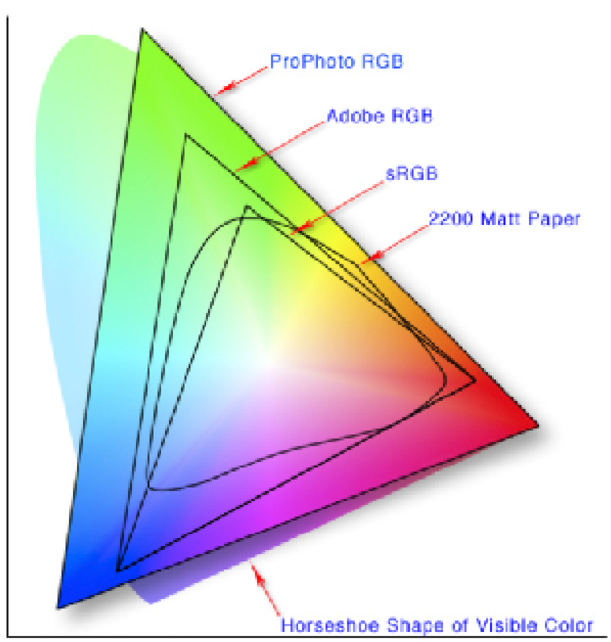
Most of them are defined by CIE(Commission Internationale de L'Eclairage/International Commission on Illumination)
CIE XYZ, CIE L*a *b, CIE YUV

RGB color model

RGB color model is a unit cube in a Cartesian coordinates system.


蓝(0,0,1)
青(0,1,1)
Magenta(1,0,1)
Brightnes
s(1,1,1)
Darkness
(0,0,0)
Green(0,
1,0)
Red(1,0,0)
黄(1,1,0)

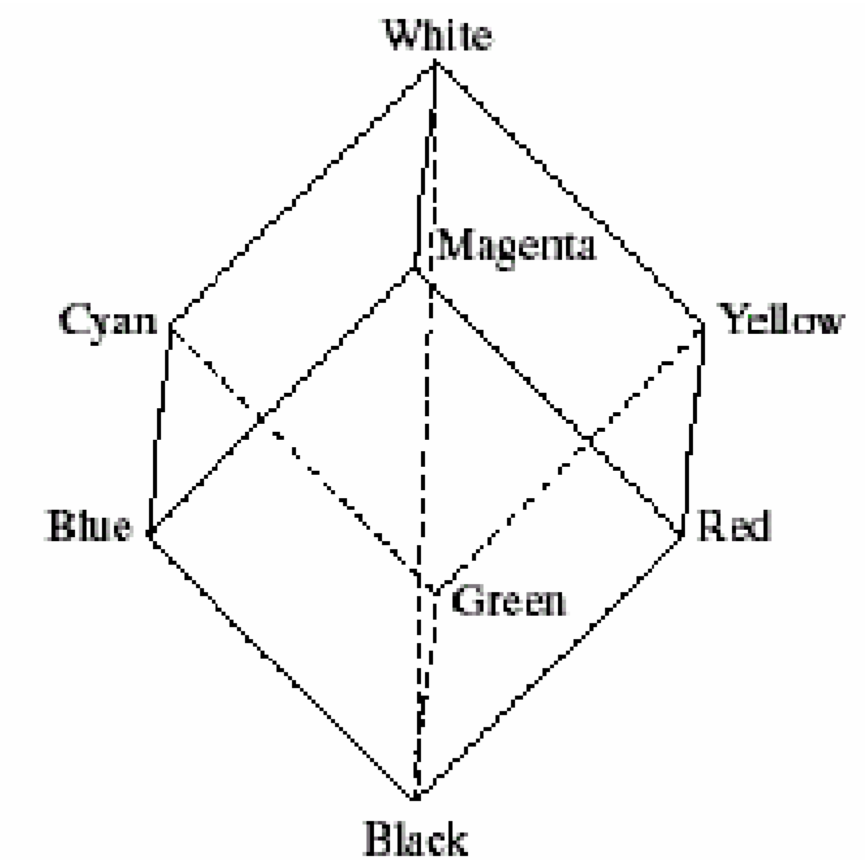

G(0, 1, 1)
灰度
C(1, 1, 1)
R(1, 0, 0)
B(0, 0, 1)


ProPhoto RGB
Adobe RGB
sRGB
2200 Matt Paper
Horseshoe Shape of Visible Color

The magnitudes of each primary color are equivalent on the main diagonal line, which lead to the white color from darkness to brightness, i.e., grayscale.
(0, 0, 0)-dark, (1, 1, 1)-bright. The other 6 corners are respectively red, yellow, cyan, blue and magenta.
RGB is a subset of CIE primary color space.
RGB is usually used in Color cathode ray tube and Color raster graphics display(computer, TV).

CMY color model

Color printing paper don't emit light:
REFLECTION, ABSORTION
So printer users ink or powder to colorize the paper, where the ink can absorb some colors but reflect the other colors.
Ink or pigment:
Complementary color of RGB: Cyan, Magenta, Yellow
The color from CMY model is called subtractive color because it subtracts(absorbs) the some reflects light.

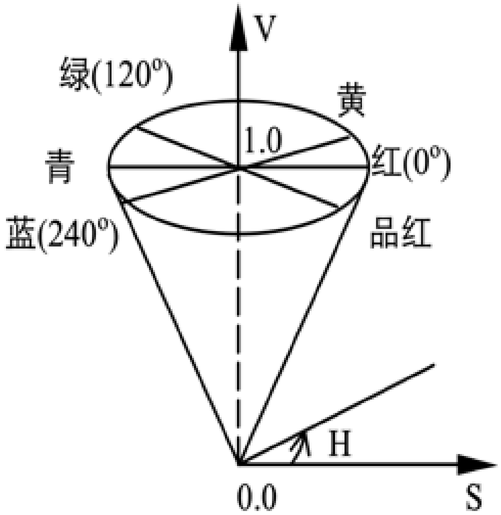


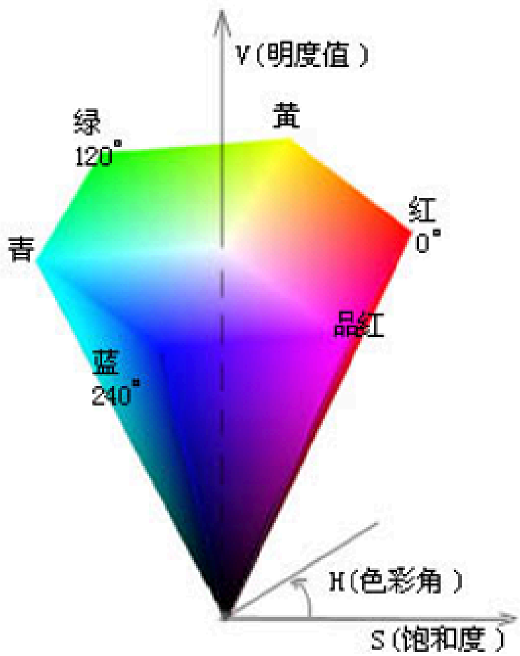
Difference between RGB and CMY

	RGB	CMY
Primary color	R, G, B	C, M, Y
Rule	R+G= Y; R+B=M; G+B=C;R+G+B=W	Y+M=R; C+Y=G; C+M=B; C+M+Y=K
Mechanism	Color+light, energy increases	Light-Pigment mixture, energy decreases
Effect	Lightness increases	Lightness decreases
Complementary relation	Additive color	Subtractive decreases
Application	Color movies, TV, etc.	Painting, Photography, printing, etc.

HSV color model

HIS/HSV color space is defined based on human visual system.(Hue, Saturation, Intensity/Value)
HIS can be depicted as a cone: Top surface of cone: V=1, which includes R=1, G=1, B=1; Hue:around the axis:RED->0^o, GREEN->120^o, BLUE->240^o; Saturation: along the radial direction.
Bottom point of cone: V=O, H=null, S=null, for darkness.
Center of the top surface: S=O, V=1, H=null, for brightness.





Advantage of HSV color model

Close to human vision.
When using RGB or CMY, if you want to change hue, you must adjust R, G, and B at the same time. But using HSV, you DON'T have to do that. Only ONE channel is adjusted.
HSV color model is a uniform color space, linearly distributed. The color difference perception is proportional to Euclidean distance on the HSV cone volume.

CIE color model

CIE color model:A series of color models according to the response of human eyes to RGB, which are carefully measured.
Device independent:Such color models are used to define device-independent color; Help people to reproduce color consistently on different devices such as scanner, monitor, and printer.
CIE color models: CIE XYZ, CIE L*a*b, CIE YUV, etc.

CIE XYZ

XYZ concept is based on the three-primary color theory, which describes human eyes have RGB receptors and all other color are mixture of RGB;
Defined in 1931; Computed by using function $\overline{x(\lambda)}, \overline{y(\lambda)}, \overline{z(\lambda)}$;
Yxy color space given in 1931: Y-> luminance, x, y are color coordinates computed based XYZ;
Describes the color range human can perceives.

CIE L*a*b

Defined in 1976;
Modification of CIE XYZ;
Overcomes the un-proportional problem in Yxy color space: x, y distance does not reflect the color difference you observe;
L: lightness;
a: green to red
b: blue to yellow;
L, a, b can be adjust separately.

CIE YUV

Used in TV;
Like Lab at some level;
Lightness is computed as:

$$Y = 0.3R + 0.59G + 0.11B$$

Color differences U, V are computed as:

$$U = 0.493(B - Y)$$

$$V = 0.877(R - Y)$$

Transformation between color spaces

1.RGB<-> CMY:

$$C = 255 - R$$

$$M = 255 - G$$

$$Y = 255 - B$$

2.RGN<-> HSV:

Search in table:CIE has defined transformation tables:RGB<-> XYZ, HSV<->XYZ; Hence, you can: RGB<-> XYZ, XYZ<-> HSV; HSV<->XYZ, XYZ<-> RGB.

Compute directly:

From RGB to HSV:Assume $m = \max(r, g, b), n = \min(r, g, b)$, where r, g, b are normalized.

$$v = m$$

$$s = \begin{cases} \frac{m-n}{m}, m \neq 0 \\ 0, m = 0 \end{cases}$$

$$h = \begin{cases} 0, s = 0 \\ \frac{60(g-b)}{m-n}, r = m \& g \geq b \\ 360 + \frac{60(g-b)}{m-n}, r = m \& g < b \\ 120 + \frac{60(b-r)}{m-n}, g = m \\ 240 + \frac{60(r-g)}{m-n}, b = m \end{cases}$$

From HSV to RGB:

if ($h == 360$), $h = 0$

$i = [\frac{h}{60}]$, $f = h\%60$

$$p = v * (1 - s)$$

$$q = v * (1 - s * f)$$

$$t = v * (1 - s * (1 - f))$$

$$\begin{cases} r = v, g = t, b = p, i = 0 \\ r = q, g = v, b = p, i = 1 \\ r = p, g = v, b = t, i = 2 \\ r = p, g = q, b = v, i = 3 \\ r = t, g = p, b = v, i = 4 \\ r = v, g = p, b = q, i = 5 \end{cases}$$

3.RGB<-> CIE XYZ

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.608 & 0.714 & 0.200 \\ 0.299 & 0.587 & 0.144 \\ 0.000 & 0.066 & 1.112 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

4.CIE XYZ<-> CIE L*a*b

$L : (0, 100)$

$a, b : (-300, 300)$

$-a$ to $+a$: green to red

$-b$ to $+b$: blue to yellow

$$u' = \frac{4X}{X + 15Y + 3Z}, v' = \frac{9Y}{X + 15Y + 3Z}$$

$$L = 116f(\frac{Y}{Y_n}) - 16, u = 13L(u' - u'_n), v = 13L(v' - v'_n)$$

$$a = 500[f(\frac{X}{X_n}) - f(\frac{Y}{Y_n})], b = 500[f(\frac{Y}{Y_n}) - f(\frac{Z}{Z_n})]$$

5.RGB<-> CIE YUV

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.435 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

JPEG format

Joint Photographic Experts Group, JPEG: Founded in 1986, presented JPEG IN 1992, accepted by ISO in 1994;
File extension: .JPEG, .JFIF, .JPG, .JPE;
Compression format for static image: As a counterpart of Moving Picture Expert Group, Lossy encoding method, Allow user to make balance between image quality and image size;
Encoding based on transformation, e.g. Discrete Cosine Transformation(DCT): JPEG2000 is based on wavelet.
Different manipulations for high-frequency signal and low-frequency signal.
Compression strategy: According to the requirement of compression ratio, remove information from high frequency to low frequency.
Advantages:High frequency information occupies much more memory. Hence, high frequency removal leads to high compression ratio, Low frequency information preserves the principle structure and color distribution of object, which is the key factors of an image, Suitable for internet based visual media.

JPEG syntax

JPEG file is a composed by a series of segments, where each segment begins with a Marker;
Purpose of Marker: identify the image data and relative information;
Marker specification: Each Marker is composed by 2 bytes, where the first byte is constantly 0xFF.

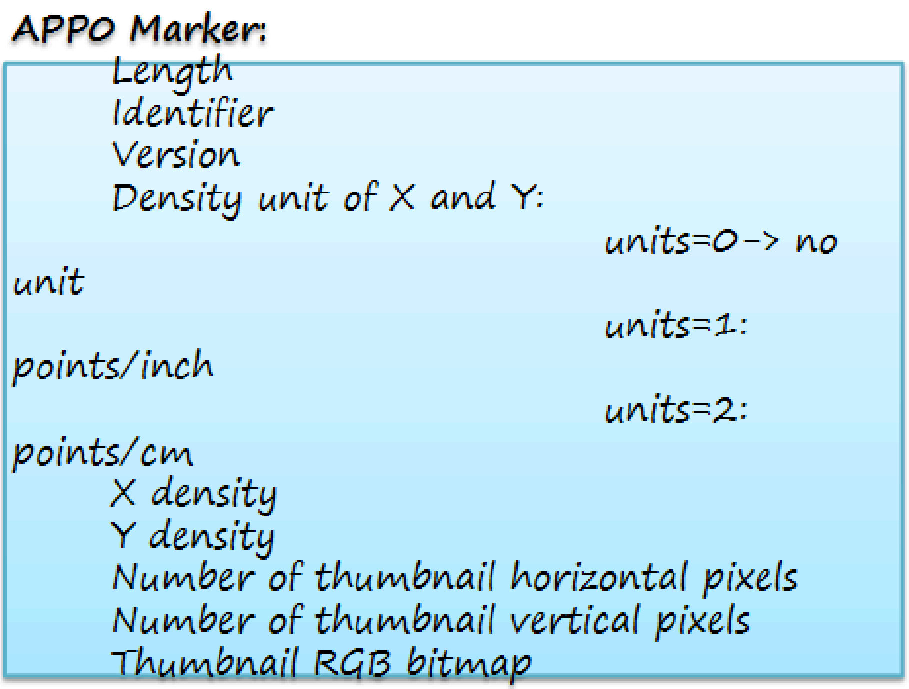
SOI	0xD8	Beginning of image
APP0	0xE0	JFIF application data block
APPn	0xE1-0xEF	Other application data block
DQT	0xDB	Quantization table
SOF0	0xC0	Beginning of frame
DHT	0xC4	Huffman table
SOS	0xDA	Beginning of scan

SOI	0xD8	Beginning of image
EOI	0xD9	End of image

JPEG file organization

(1) SOI(Start of Image) Marker

(2) APPO Marker:



(3)APPn(Markers), n=1~15(optional)

APPn Length

Detail information: e.g. EXIF use APP1 to store meta data

EXIF=Exchanged image file format

(4) One or more DQT(Define Quantization Table)

Three parts;

Quantization table length;

Quantization table number;

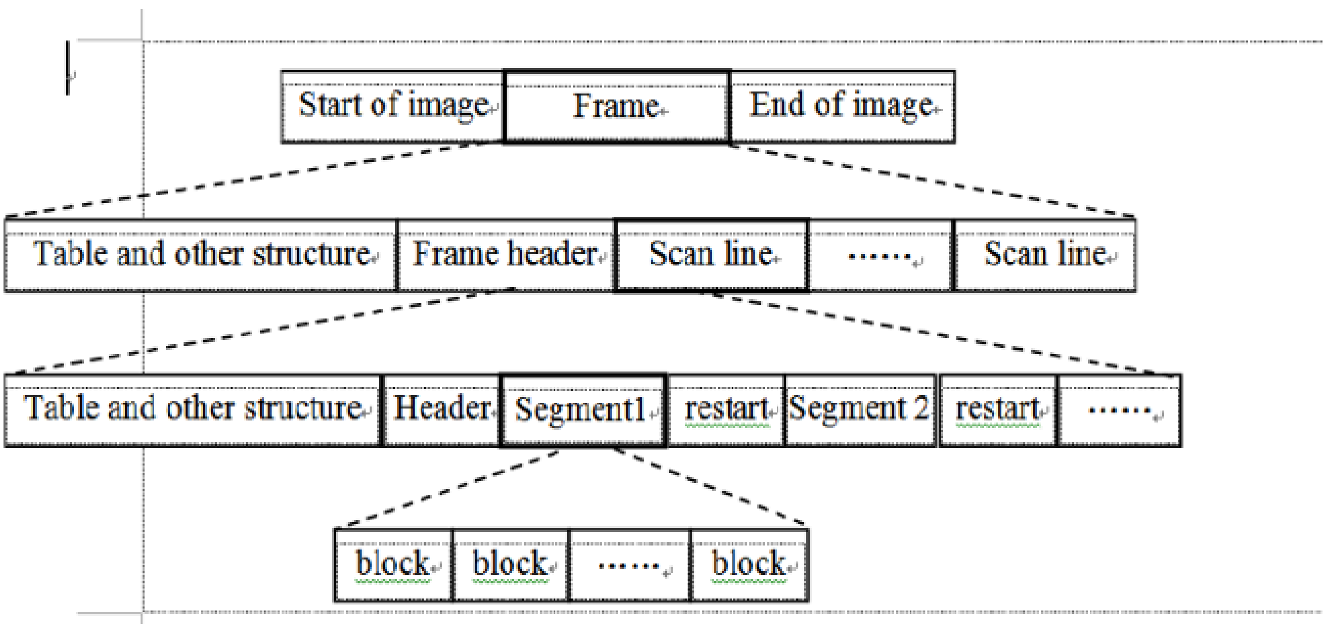
Quantization table;

For a large region, human eyes can identify the subtle changes of color (low frequency region), but are very fair for the high frequency region. Inspired by this concept, we can quantize the high frequency part to make quantization. In order words, each component on the spectrum can be divided by a constant and rounded to be zero. And different constants are required by different components. So finally we get the Quantization Table.

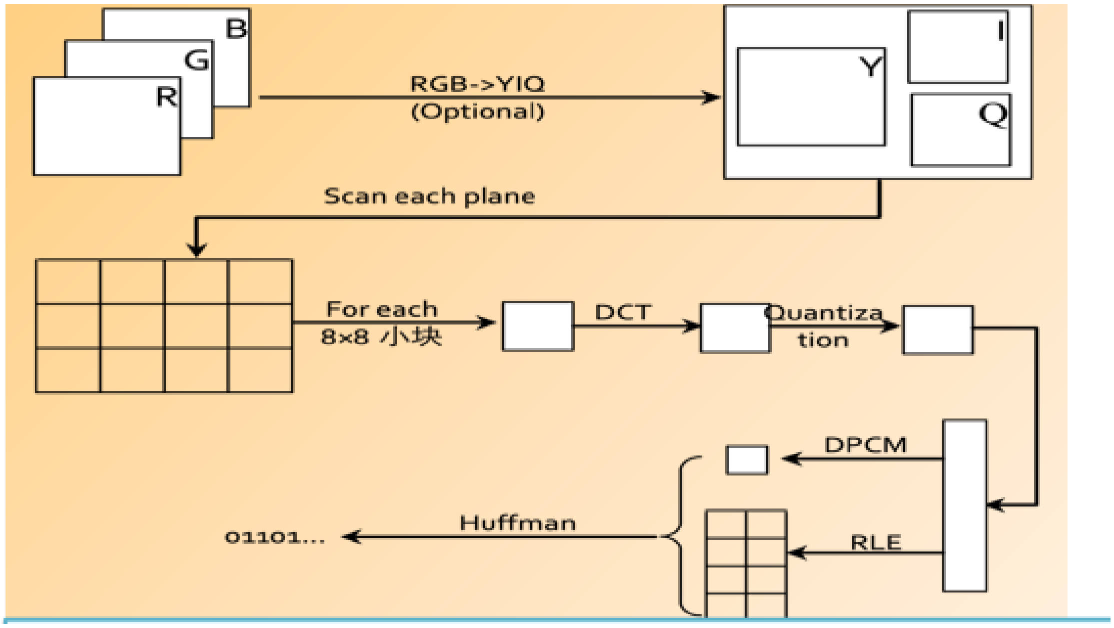
(5) SOF0 (Start of Frame, DCT based)

Start of frame length;
Precision: color depth (bit-width) of each color channel;
Image height;
Image width;
Number of color components;
For each color component:
ID, vertical sample factor, horizontal sample factor, Quantization table #
(6) DHT(Define Huffman Table)
Huffman table length, Type, AC or DC, Index, bits table, value table.
(7) Start of Scan(SoS)
SoS length; Number of color components; For each color component; ID; AC table #; DC table #;Compression image data
(8) End of Image(EOI)

JPEG file structure



JPEG encoding principle



DCPM encoding, (differential pulse-code modulation, DPCM) is a modulation technique invented by the British Alec Reeves in 1937. It is a digital representation of an analog signal where the magnitude of the signal is sampled regularly at uniform intervals.
The DCT transforms an 8×8 block of input values to a linear combination of these 64 patterns. The patterns are referred to as the two-dimensional DCT basis functions, and the output values are referred to as transform coefficients.

JPEG's advantages

1.Not for line drawing, text, symbol, icon, etc..
2.Lts lossy compression leads to unavoidable artifacts.

TIFF format

TIFF(Tagged Image File Format): an extensively used raster image format for:scanner, CAD system, GIS(Geographic Information System)

Filename extension	.tiff,.tif
Type code	TIFF
Uniform Type Identifier	public.tiff
Developed by	Aldus, now Adobe Systems
Initial release	1986

Filename extension	.tiff,.tif
Type of format	Image file format
Extended to	Exif,DCF,TIFF/EP,TIFF/IT,TIFF-FX,GeoTIFF

TIFF characteristics

Universal: Support devices from high-end to low end;
Highly extendable: support public tag and private tag;
Flexible: Support several compression formats;
Accessible: public software library available;
Powerful: Binary image, Grayscale image, Palette, True color, Other extension.

TIFF file structure

```
struct TIFF_img{
   unsigned char **mono;
   unsigned char **cmap;
   unsigned char ***color;
   char TIFF_type;
   char compress_type;
   int height;
   int width;
}
```

TIFF file format

Graphics Interchange Format;
LZW encoding method is employed;
One GIF stores several image frames;
Has color table (global (all frames) or local color table);
Support display order custom or lay over;
Allow interlacing store;
Support text embedding.

Image Processing data structure

1.Matrices
2.Chains
3.Topological Data Structures

4.Rational Structures.
Note:
Matrices to rational structures is a kind of relationship from low level to high level, i.e., from semantic independent to semantic dependent.

Matrices

Matrix is the most common data structure for low-level representation of an image: Elements of the matrix:gray level, brightness, or other properties; Access via coordinates of pixel location.
Binary images: represented by a matrix containing only zeros and ones.
Matrices of different resolution
Multispectral image represented by several matrices: each matrix containing an image corresponding to one spectral band.

Chains

Chain codes: Chains are used for description of object borders
Chains can be represented using static data structures (e.g., 1D arrays); their size is the longest length of the chain expected.

Chains: Run length encoding (RLE)

In binary images, run length coding records only areas that belong to the object in the image; the area is then represented as a list of lists.
Each row of the image is described by a sublist, the first element of which is the row number.
Subsequent terms are coordinate pairs; the first element of a pair is the beginning of a run and the second is the end.
There can be several such sequences in the row.

Image processing programming

1.Windows APIs for accessing DIB

```
# include <windows.h>
HANDLE LoadImage(
   HINSTANCE hinst,
   LPCTSTR IpszName,
   UINT uType,
   int cxDesired,
   int cyDesired,
   UINT fuLoad
);
```

Display DIB: BitBlt, SetDIBitsToDevice, StretchDIBs
2.VC++ programming and example
3.Matlab image processing toolbox
MATLAB and Image Processing Toolbox provide lots of image processing functions.
Users can use these functions to analyze the image data, acquire image details and design suitable filters.
With the designed filters, you can perform de-noise, enhancement, etc..
OR you can design a comprehensive filter by using Simulink:just drag-and-drop is needed.

Image Processing Toolbox

Image display and exploration
GUI Tools
Spatial Transformation and Image Registration
Image Analysis and Statistics
Image Arithmetic
Image Enhancement and Restoration
Linear Filtering and Transform
Morphological Operations
ROI-bases, Neighborhood, and Block Processing
Colormaps and Color Space
Utilities: array operation, demos, preferences and others.

Chapter 2:Binary image and morphology

Binary image
How to obtain a binary image: binarization
Image morphological operation: Definition of set; Erosion; Dilation; 'Open' operation; 'Close' operation; Hit-or-miss transform.

1.Binary Image

Binary Image: Pixel value is limited to 0 or 1. Thus, only 1 bit is required for each pixel. For convenience in programming, we usually use 0 OR 255 instead of 0 OR 1 to represent the binary image.

Advantages vs. Disadvantages

Advantages:Less memory; More efficient; CAN sometimes be applied on grayscale image; More cheap.
Disadvantages:Application field is limited; CANNOT be applied on 3-D data; Less expensive force, FALL to convey the visual details; CANNOT control the contrast.

2.Image binarization

2.1.Acquisition of binary image

Construct a binary image: thresholding the grayscale image by reset the pixel value, i.e.

$$\begin{cases} I(x,y)=0, I(x,y)<Threshold \\ I(x,y)=255, I(x,y)\geq Threshold \end{cases}$$

2.2.How to find a good threshold?

The binary image can be treated as two parts: one is Foreground, the other is Background.
Try to find a suitable threshold to minimize the variances within the Foreground and the Background, while maximize the variance between them.

$$\sigma_{within}(T)=\frac{N_{Fgrd}(T)}{N}\sigma^2_{Fgrd}(T)+\frac{N_{Bgrd}(T)}{N}\sigma^2_{Bgrd}(T)$$

$$\sigma^2_{between}(T)=\sigma^2-\sigma^2_{within}(T)$$

$$=(\frac{1}{N}\sum_{x,y}(f^2[x,y]-\mu^2))-\frac{N_{Fgrd}}{N}(\frac{1}{N_{Fgrd}}\sum_{x,y\in Fgrd}(f^2[x,y]-\mu^2_{Fgrd}))$$

$$-\frac{N_{Bgrd}}{N}(\frac{1}{N_{Bgrd}}\sum_{x,y\in Bgrd}(f^2[x,y]-\mu^2_{Bgrd}))$$

$$= -\mu^2 + \frac{N_{Fgrd}}{N}\mu_{Fgrd}^2 + \frac{N_{Bgrd}}{N}\mu_{Bgrd}^2$$

$$= \frac{N_{Fgrd}}{N}(\mu_{Fgrd} - \mu)^2 + \frac{N_{Bgrd}}{N}(\mu_{Bgrd} - \mu)^2$$

$$\rightarrow \frac{N_{Fgrd}(T)N_{Bgrd}(T)}{N^2}(\mu_{Fgrd}(T) - \mu_{Bgrd}(T))^2$$

Easier deduction:

$$\omega_f = \frac{N_{Fgrd}}{N}, \omega_b = \frac{N_{Bgrd}}{N}, \omega_f + \omega_b = 1$$

$$\mu = \omega_f * \mu_{Fgrd} + \omega_b * \mu_{Bgrd}$$

$$\sigma_{between} = \omega_f(\mu_{Fgrd} - \mu)^2 + \omega_b(\mu_{Bgrd} - \mu)^2$$

$$= \omega_f(\mu_{Fgrd} - \omega_f * \mu_{Fgrd} - \omega_b * \mu_{Bgrd})^2 + \omega_b(\mu_{Bgrd} - \omega_f * \mu_{Fgrd} - \omega_b * \mu_{Bgrd})^2$$

$$\rightarrow \omega_b\omega_f(\mu_f - \mu_b)^2$$

2.3.How to select a good threshold

Step1: Find the maximal and minimal pixel in the image;
Step2: Thresholding the image by "minimal value + 1" to obtain the binary image;
Step3: Determine the foreground and background in terms of the binary image;
Step4: Compute the within-variance and the between-variance respectively;
Step5: Go to Step2 until reaching the maximal pixel value;
Step6: Obtain the threshold corresponding to the maximal between-variance and minimal within-variance.

2.4.Generalization: 1D->3D

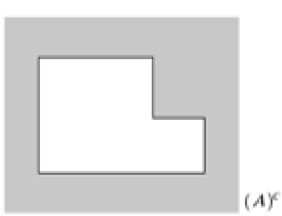
Apply the aforementioned thresholding scheme onto the color image by taking into account r, g and b.

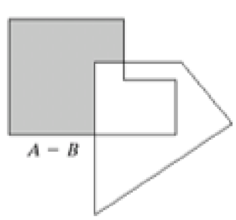
3.Morphological operation

Morphology: Presented in 1960s for studying the structure and form organisms; A branch of bioscience.
Mathematical morphology:
Theory basis: Set theory;
A kind of simple nonlinear algebra operator;
Mainly for binary image, BUT can be extended to grayscale image;
Noise filtering, shape simplification, segmentation, object description, etc.
Mathematical morphology is a tool to analyze the image form and structure;
Basic idea: measure and extract the shape with structure element so as to analyze and recognize the image;
Image morphological operation can simplify the image, remove the unwanted structures while preserving the principle shape;
FOUR morphological operations here: dilation, erosion, opening and closing.

3.1.Definition of set

A is a set in **Z²**, **Z**: the set of integers;
$a = (a_1, a_2)$ is an element of A, $a \in A$
a is not an element of A, $a \notin A$
Empty set: ϕ
A is a subset of B, $A \subset B$
Union of two sets $C = A \cup B$
Intersection of two sets $C = A \cap B$
Disjoint, no intersection $A \cap B = \phi$
$A^c = \{\omega | \omega \notin A\}$, $A - B = \{\omega | \omega \in A, \omega \notin B\}$

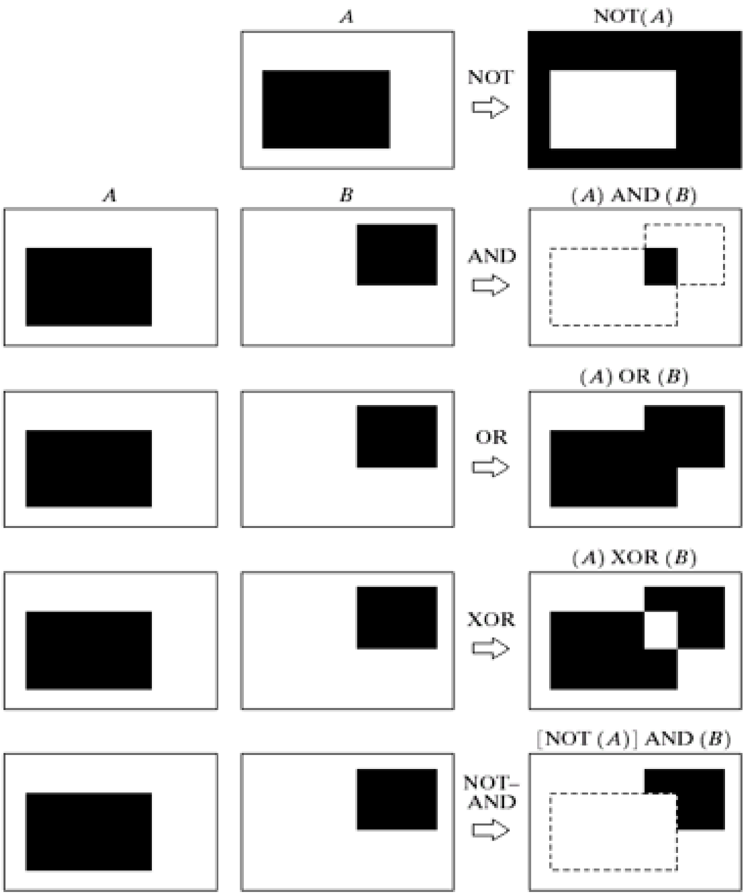




AND, OR, NOT: p, q is two pixels in binary image:

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Foreground: 1-Black; Background: 0-White



$A \cap B$

$A \cup B$

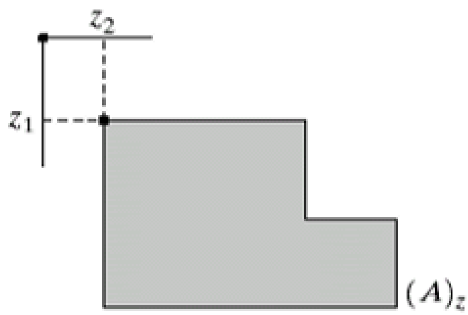
$A\ XOR\ B$

$B - A$

Set operation in Morphology:

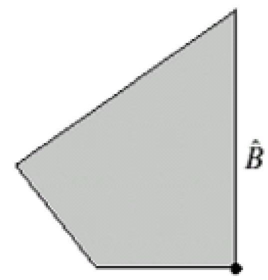
Translation Set

$$(A)_z = \{c \mid c = a + z, \text{ for } a \in A\}$$



Symmetry Set

$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$$



3.2.Erosion

A: Binary image

B: binary template, structure element

$$A \ominus B = \{(x, y) | (B)_{xy} \subset A\}$$

Physical meaning: remove boundary, remove unwanted small objects.

3.3.Dilation

A: Binary image

B: binary template, called structure element

Dilation: enlarging the foreground

A is dilated by B

$$A \oplus B = \{z | (B)_z \cap A \neq \phi\}$$

The intersection set between A and the translated B is not empty.

Physical meaning :

Dilation adopts the connected background pixels into the foreground, which extends its boundary and fill the holes in the foreground. And whether "connected" is decided by the structure element.

3.4.Dilation and Erosion

Dilation: The binary image D obtained from A dilated by B meets the following condition: when B's original point moves to (x,y), its intersection with A is not empty.
Erosion: The binary image E obtained from A eroded by B meets the following condition: when B's original point moves to (x,y), B will be contained by A.

$$(A \ominus B)^c = \{z|(B)_z \subset A\}^c$$

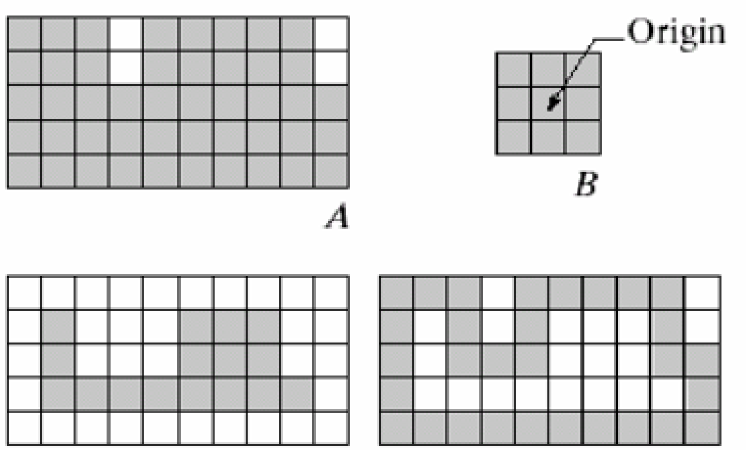
$$= \{z|(B)_z \cap A^c = \phi\}^c = \{z|(B)_z \cap A^c \neq \phi\}$$

$$\rightarrow= A^c \oplus B$$

Similarly

$$(A \oplus B)^c = A^c \ominus B$$

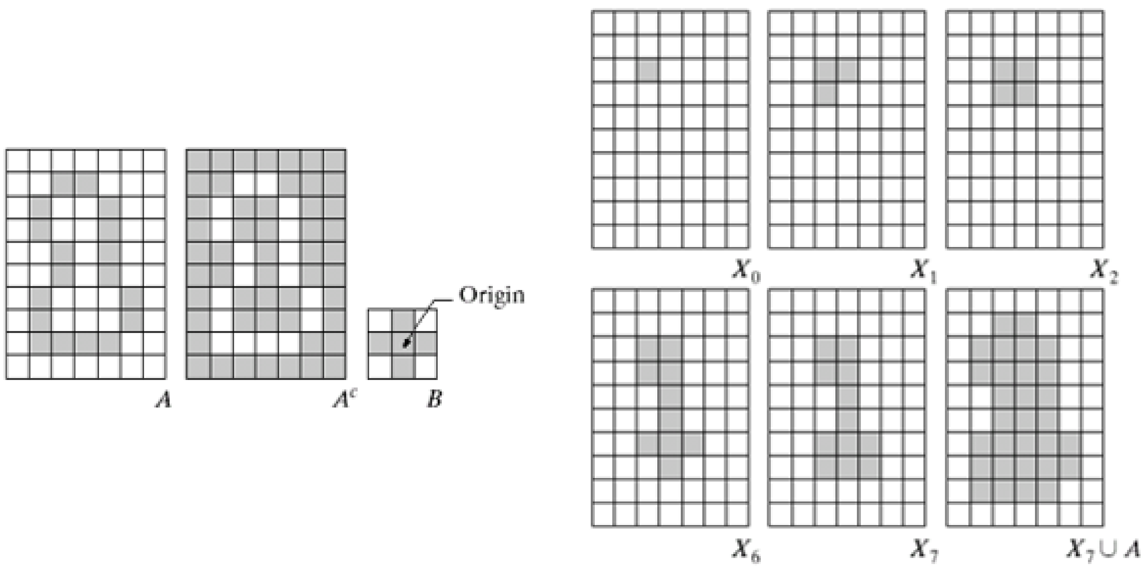
Application:
Extract the boundary of a binary image: (1) erode A(with a structure element), (2) A - erode(A)



Filling holes:

$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1, 2, 3, ...$$

$X_0 = p$, if $X_k = X_{k-1}$, iteration will terminate at the k-th step. The union set of X_k and A contains the foreground and its boundary.



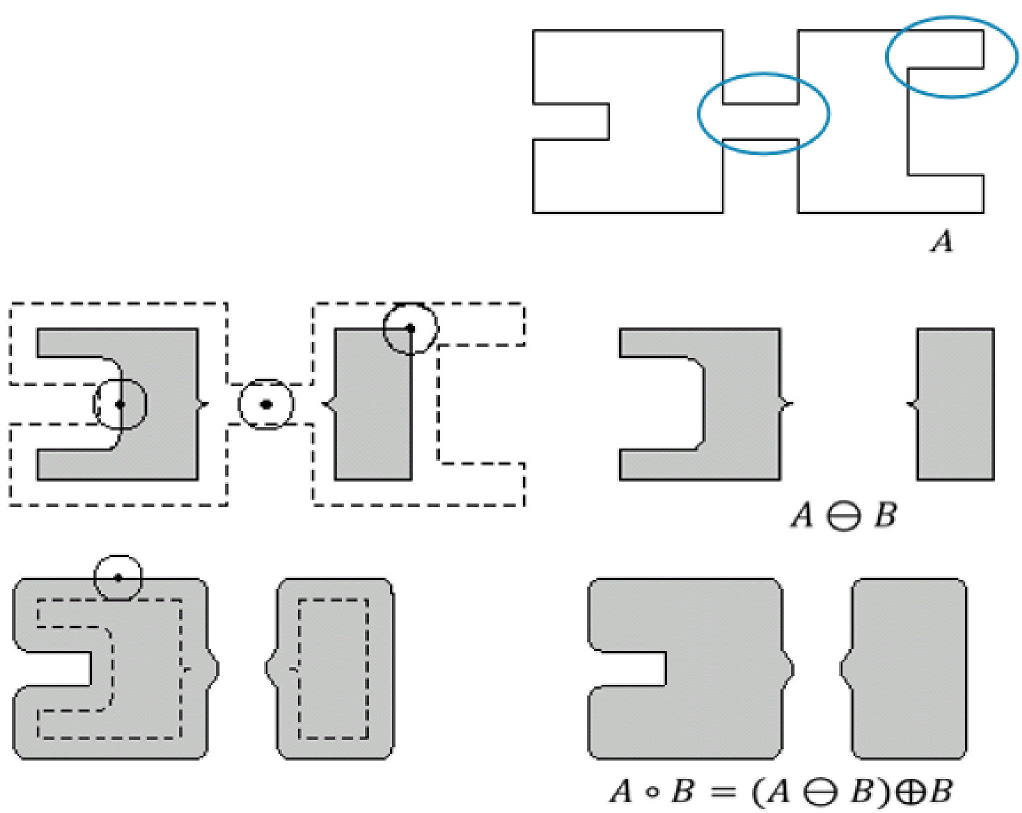
3.5.Opening

Opening: erosion, then dilation

$$A \circ B = (A \ominus B) \oplus B$$

Erosion + Dilation = original binary image?

Remove small objects, segement object at thin part, smooth boundary of large object but preserve its original area.



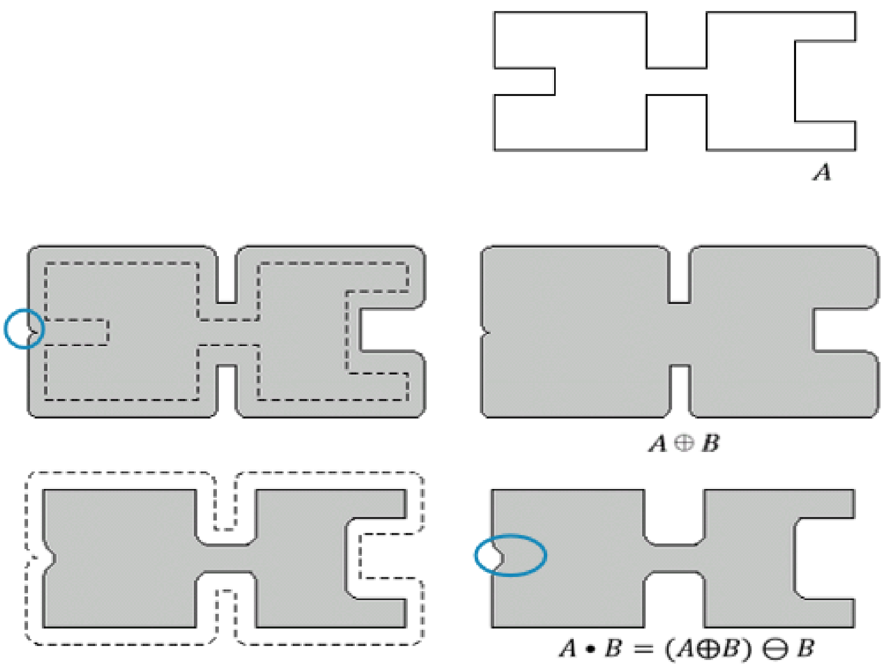
3.6.Closing

Closing: Dilation, then erosion

$$A \bullet B = (A \oplus B) \ominus B$$

Dilation + Erosion = Erosion + Dilation ?

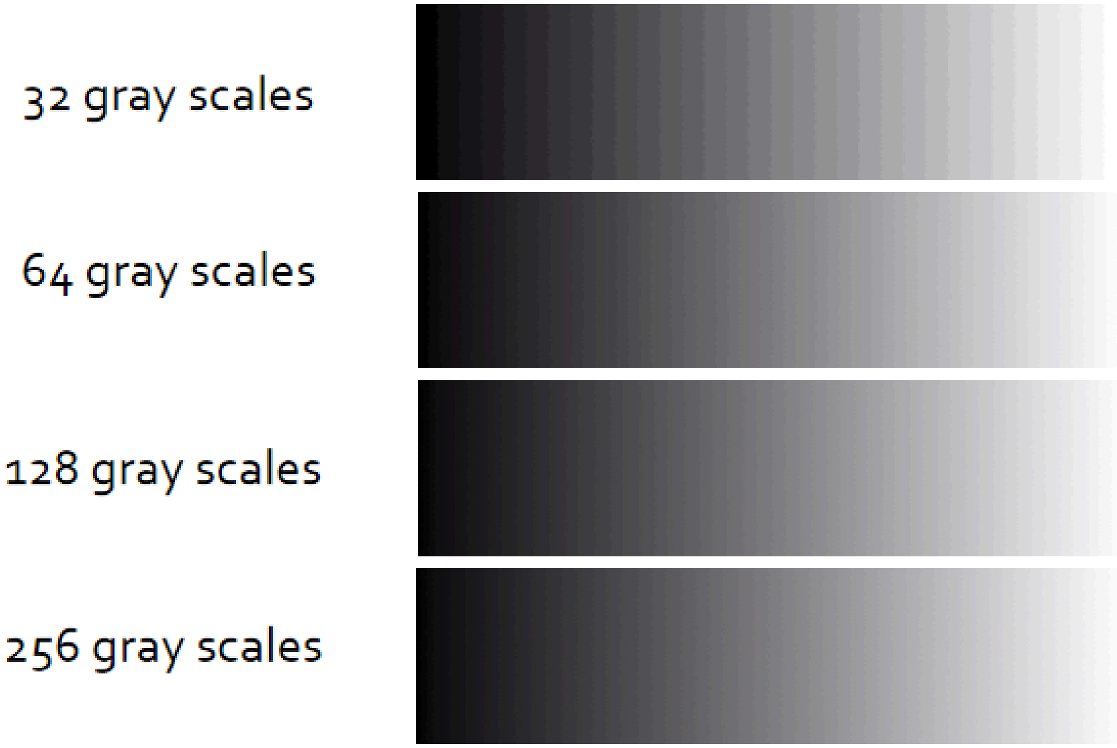
Fill small holes, connect the neighboring objects, smooth boundary while preserving the area at most.



Chapter 3:Basic operations on image(I)

Grayscale image transform: Visibility of image; Visibility enhancement: Logarithm; Histogram of image; Histogram equalization; Etc.

3.1.Grayscale perception



Weber's Law on grayscale image:

Assume the lightness difference between two consequent grayscale is the visible threshold in Weber's Law, then

$$\frac{I_{max}}{I_{min}} = (1 + K_{weber})^{255}$$

Considering $K_{weber} = 0.01...0.02$, $\frac{I_{max}}{I_{min}} = 13...156$

The traditional display contrast: cathode-ray tube (CRT): 100:1 ; Paper Print: 10: 1

In addition, the perceptive ability is ruled by $\log(I)$, which is called Fechner's Law.

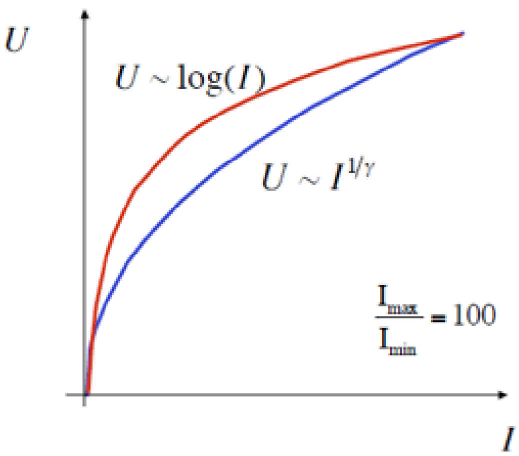
For a CRT display:

Voltage adjustment can influence the luminance and its changing curvature. Thus the visibility and visual detail expressive force can be adjusted.

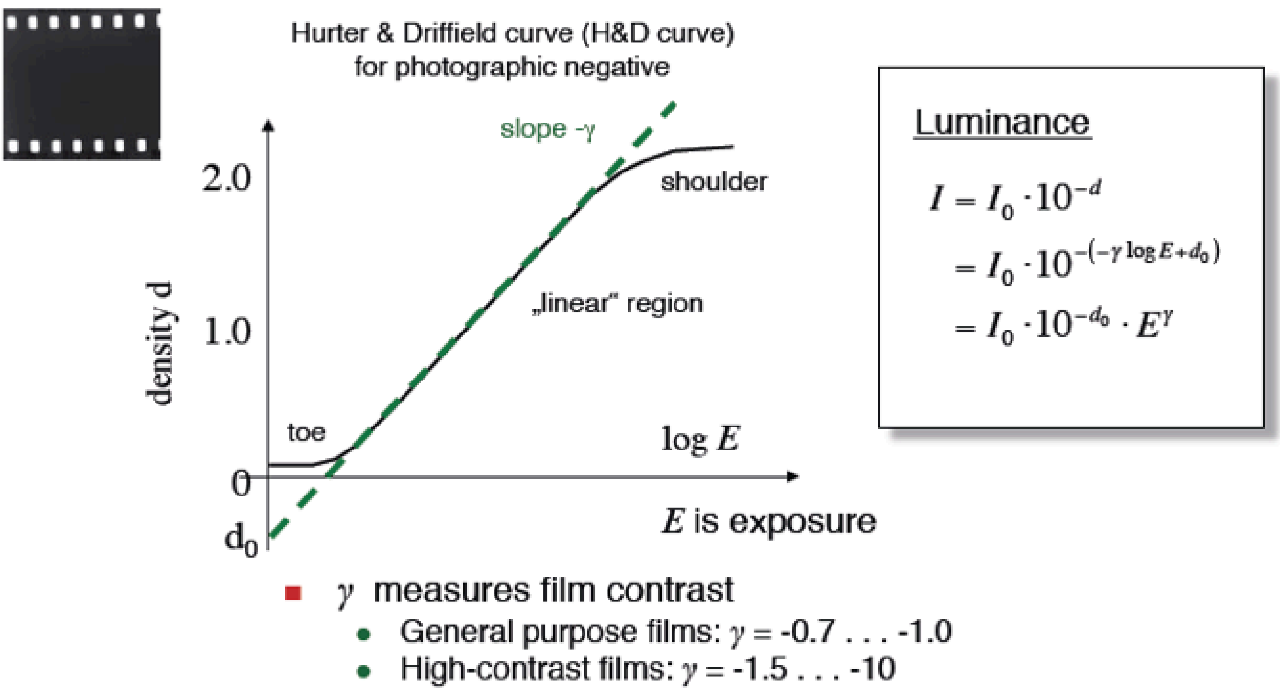
γ correction:

$$U \sim I^{\frac{1}{\gamma}}$$

Ideal logarithmic curve and a γ curve:



Gamma correction in photography:



3.2.Visibility enhancement: logarithmic operation

In order to enhance the image's visibility, adjust the pixel value by a logarithm operator.

$$L_d = \frac{\log(L_w + 1)}{\log(L_{max} + 1)}$$

L_d is display luminance, L_w is the real luminance, L_{max} is the maximal luminance value in the image. This mapping function make sure that: no matter the dynamic range of the scene, the maximal luminance value will be 1(white), and other values changes smoothly.

3.3.Grayscale image and histogram

Grayscale image---Concept

2D array composed by pixels (M rows $\times$ N columns)

Each pixel is represented by 8 bits. The grayscale is divided into $2^8 = 256$ levels, grayscale intensity $p = 0, 1, 2, ..., 255$

Smaller the grayscale intensity, darker the pixel, and vice versa.

Histogram---Concept

Grayscale histogram is a kind of statistical graph, which indicates the proportions of the number of pixels of different gray levels in the total number of pixels of the given image.

The proportion of the number of pixels of different gray levels in the total number of pixels of the given image.

Histogram---Mathematical form

Given gray levels ranging from $[0, L-1]$, the image's grayscale histogram can be represented by the following discrete function:

$$h(r_k) = n_k$$

where, r_k is the k-th gray level, n_k is the number of pixels of gray level r_k, $0 \leq k \leq L-1, 0 \leq n_k \leq n-1$, n is the total number of pixels in the image.

Usually probability density function is employed to normalized the histogram:

$$P(r_k) = \frac{n_k}{n}$$

$P(r_k)$ is the probability of gray level r_k. And

$$\sum_{k=0}^{L-1} P(r_k) = 1$$

Color histogram---Concept

Color histogram is a kind of statistical graph, which indicates the proportions of the number of pixels of different **r, g, b** intensity levels in the total number of pixels of the given image.

Histogram---Characteristics

Basis of spatial domain based processing technique;

Reflects the grayscale distribution of image, but fails to convey the structure changes of visual information in the image;

Unique for a given image;

Different images may share the same histogram.

Different images may share the same histogram.

Histogram---summary

Histogram can be used for image enhancement, compression, and segmentation. They are practical means of image processing.

Histogram equalization

Non-uniform distribution histogram of the original image is trandformed by function T to be a uniform distribution histogram, then adjust the original image according to the equalized histogram.

After image histogram equalization, the image's histogram becomes uniform, i.e., different gray levels have the same probability to appear in the image.

Find the transform function T and build the following mapping

$$s = T(r)$$

to make sure that each gray level r in the original image has a corresponding gray level in the new image s.

find T(continuous)

Assumptions:

(1) Let r and s represent the gray levels before and after the transform, and $0 \leq r, s \leq 1$

(2) $P(r)$ and $P(s)$ are the probability corresponding to r and s.

Rules:

(1)$T(r)$ is Monotonically increasing function, $0 \leq r \leq 1$ and $0 \leq T(r) \leq 1$.

(2)Inverse transform $r = T^{-1}(s)$ is also a Monotonically increasing function.

Considering grayscale transform will not change the locations and the number of pixels, we have

$$\int_0^r P(r)dr = \int_0^s P(s)ds = \int_0^s 1ds = s = T(r)$$

i.e.

$$s = T(r) = \int_0^r P(r)dr$$

Summary:

The value of transform function T at r, namely s, is the area of the histogram curve covers between $[0, r]$ in the original image.

find T(discrete)

For a gray level r_k in the original histogram, its gray level s_k after transform can be obtained by just summing up all the number of pixels lie between $[0, r_k]$

Histogram equalization

- problem:
  In terms of the standard histogram equalization, the probabilities of different gray levels after equalization should be same (top). However, it's not true in practice (bottom).Why ?
- Answer
  In step 2, s_k is not exactly equivalent to one of the eight gray levels. Hence, it is included by a nearby gray level. So several neighboring s_k may falls into the same gray level. After discrete histogram equalization, different gray level's probability may be different.
- In Matlab

```
I = imread('pout.tif');
imshow(I);
I2 = histeq(I);
imshow(I2);
```

- Conclusion
  Histogram equalization is essentially to reduce the gray levels of image contrast for increasing the contrast. Gray levels of low probabilities are classified into few or one gray-scale. Such gray levels will not be enhanced in the equalization process. If these gray levels are important to the original image, you need to use the local histogram equalization.

3.4.Histogram Fitting

Concept

Change the histogram of given image to fit another image's histogram or a predefined distribution.
Aim to enhance the grayscale range we have interest, and improve the image quality.
Realize histogram fitting by using histogram equalization.

Continuous

Step1: Base on the equa

$$s = T(r) = \int_0^r P(r)dr$$

map r in the original histogram to be s.

Step2: based on equat

$$v = T(z) = \int_0^z P(z)dz$$

map the gray level z in the resulted histogram to be v.

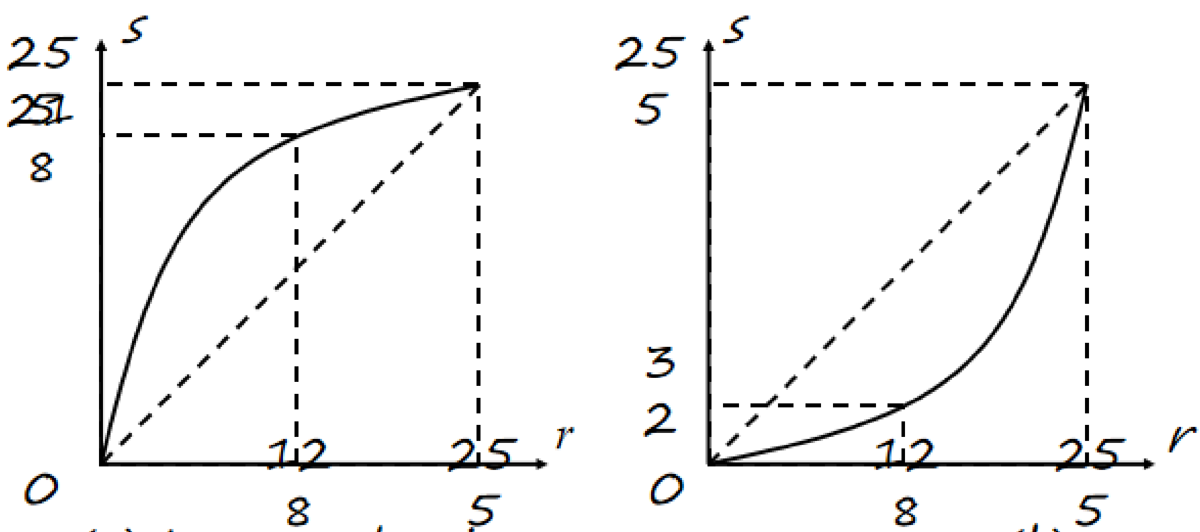
Step3: obtain $z = G^{-1}(v)$ from $v = G(z)$. $v = s$ because s and v have the same distribution. Obtain $z = G^{-1}(s)$ corresponding r.

Discrete
Compute two tables in advance (refer to the example in histogram equalization);
Select a pair of v_k and s_k where $v_k = s_k$;
Look for the corresponding z_k and r_j in the tables;
Finally, the gray scale level r_j in the original image is mapped to z_k to obtain the expected image.

3.5.Histogram transform

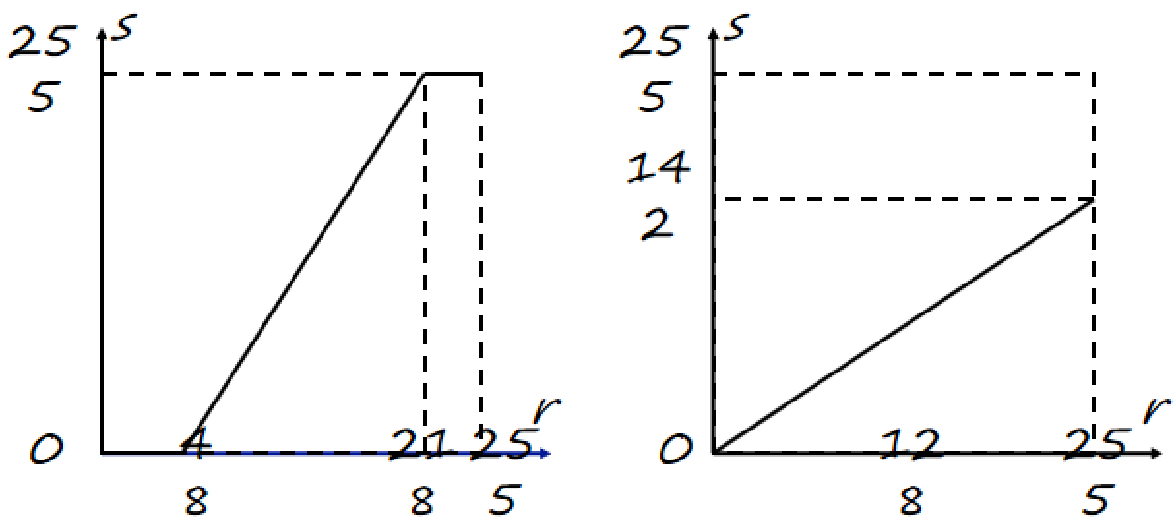
Concept
Histogram trandform is used to find the transform function to connect the before and the after transformed histograms.
After histogram transform, any gray level in the original image can find its corresponding new gray level, which results in a new image.
Both histogram equalization and belong to histogram transform.
image enhancement
Improve the visual effect of given image by series of techniques, or make the image more suitable for human or computer to carry out analysis.
Image enhancement is not for image fidelity preservation, but for strengthen some important information for human or computer and weaken the unwanted information.
Eliminate the noise and improve the visual effct to strenfthen the information of interest.

1. Lumiancce adjustment



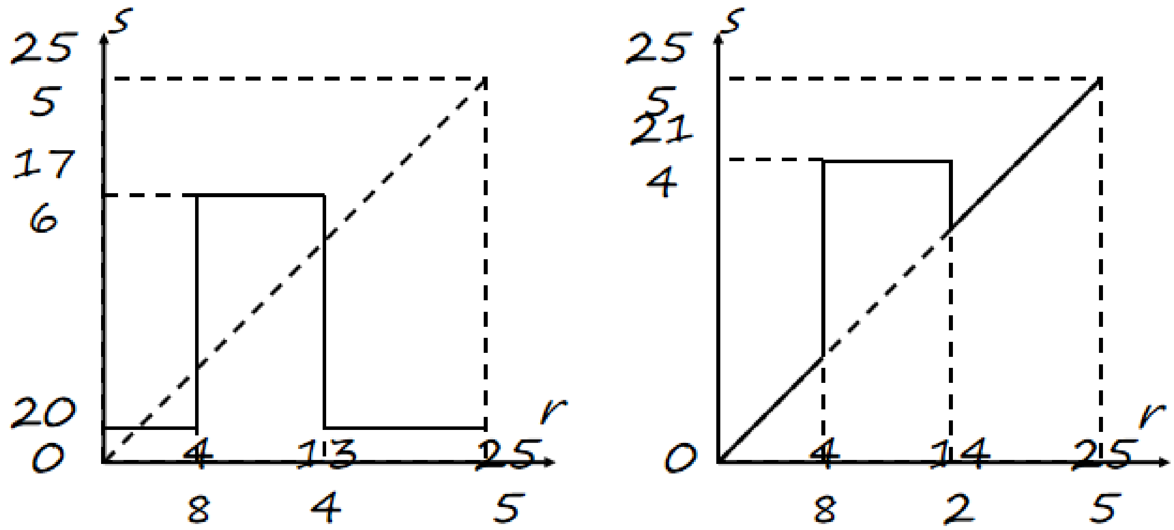
(a) Increase luminance (b) decrease luminance

2. Contrast adjustment

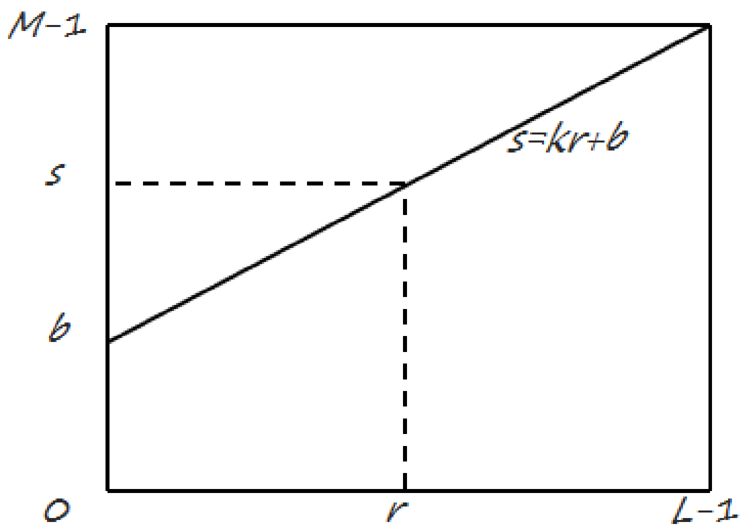


(a) Increase contrast (b) Decrease contrast

3. Color quantization



Linear histogram transform



$$s = T(r) = kr + b$$

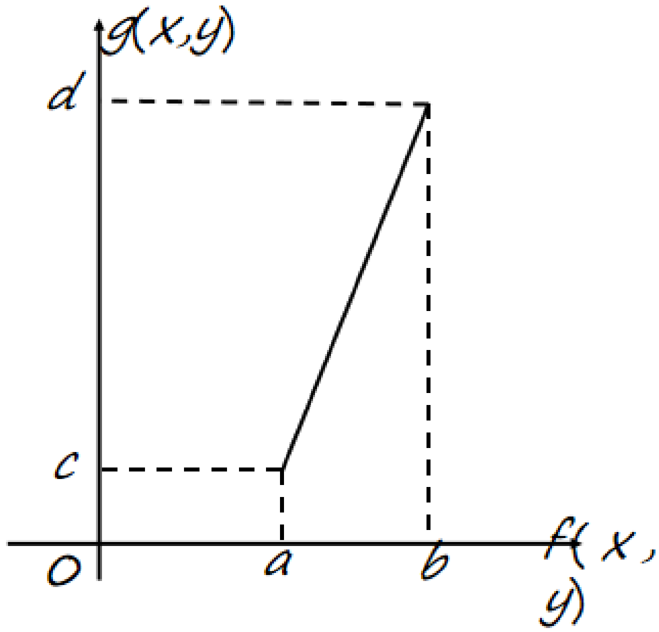
r : grayscale intensity of the input pixel

s : grayscale intensity of the output intensity

k : constant

b : constant

Contrast stretching:

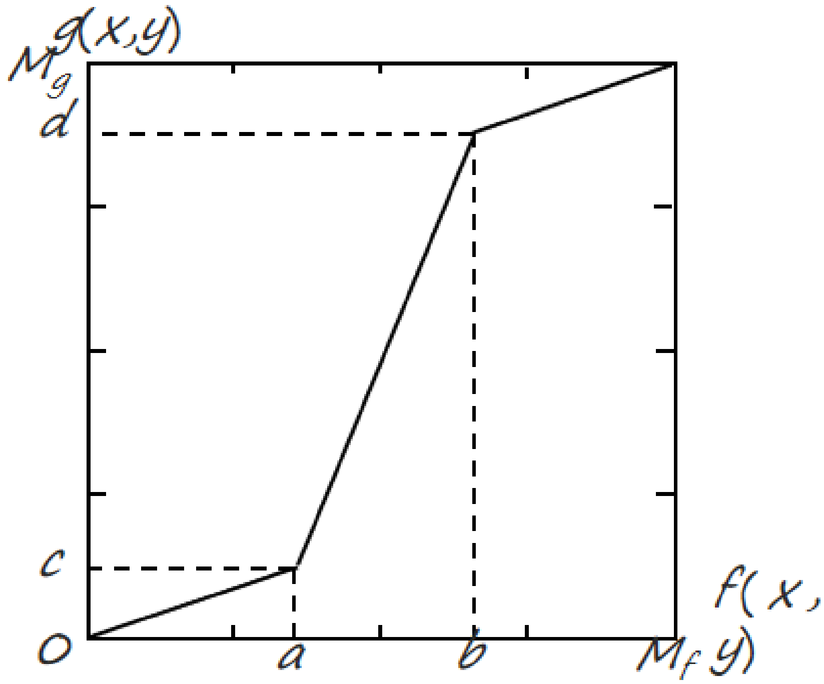


$$g(x,y)=\begin{cases} d, f(x,y)>b \\ \frac{d-c}{b-a}[f(x,y)-a]+c, a\le f(x,y)\le b \\ c, f(x,y)<a \end{cases}$$

Input image $f(x,y)\in[a,b]$
Output image $g(x,y)\in[c,d]$

Piecewise linear grayscale trandform:

Linear grayscale transform by using a piecewise function, which strengthen the region of interest, weaken the region of no-interest.



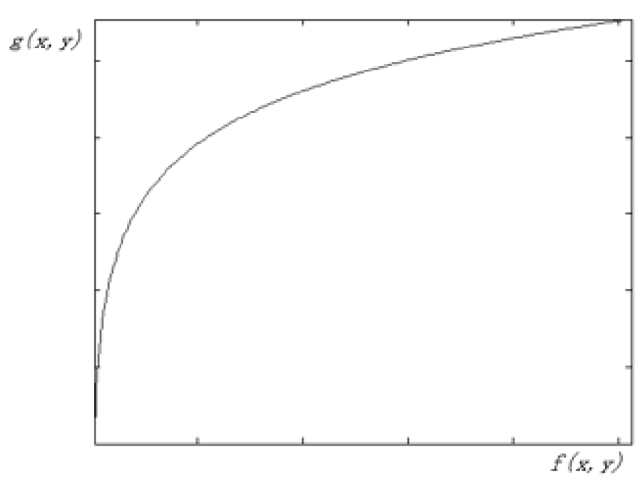
$$
\begin{cases}
\frac{M_g-d}{M_f-b}[f(x,y)-b]+d, b \le f(x,y) \le M_f \\
\frac{d-c}{b-a}[f(x,y)-a]+c, a \le f(x,y) < b \\
\frac{c}{a}f(x,y), 0 \le f(x,y) < a
\end{cases}
$$

$f(x,y) \in [0, M_f], g(x,y) \in [0, M_g]$

Nonlinear histogram trandform

Logarithmic function and Exponential function are two popular nonlinear transform functions.

Logarithmic function:

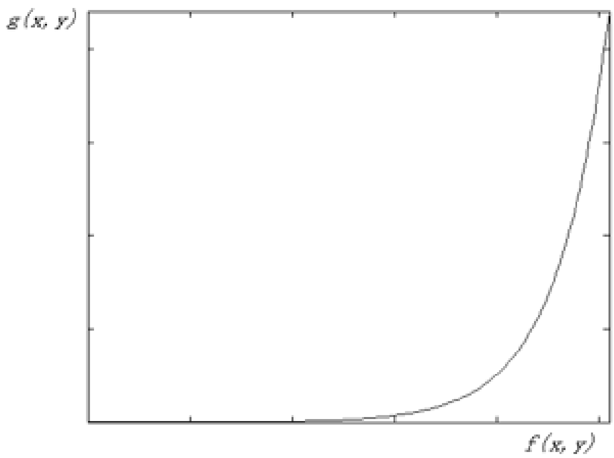


Stretch low grayscale region
Compress high grayscale region

$$g(x,y) = a + \frac{\ln[f(x,y)+1]}{b\ln c}$$

a, b, c are all adjustable

Exponential transform:



Stretch low grayscale region
Compress high grayscale region

$$g(x,y) = b^{c[f(x,y)-a]} - 1$$

a, b, c are all adjustable

Chapter 4:Basic operations on image(II)

Concept
Produce a new image by changing the location of pixel through trandformation.
Doe NOT change pixel value, but change the pixel location.
Equation

$$g(x,y) = f(x',y') = f[a(x,y), b(x,y)]$$

OR

$$\begin{cases} x' = a(x,y) \\ y' = b(x,y) \end{cases}$$

$f(x,y)$: input image;
$g(x,y)$: output image;
$a(x,y), b(x,y)$: describe the spatial transformation
Two types

1. Simple transformation —— the procedure of transformation (the pixel location before and after transformation) is known, e.g., translation, mirror, transpose, rotation, scale, shear, etc.
2. General transformation —— the procedure of transformation is complex. It's difficult to measure the transformation parameters. Usually, general transformtion will be employed when people fix the image distortion.

4.1Simple geometric transform

4.1.1.Translation

Concept
Translate all the pixels in an image vertically and horizontally to produce a new image.
Equation

$$\begin{cases} x' = x + x_0 \\ y' = y + y_0 \end{cases}$$

OR

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Each pixel in the original image is translated x_0 and y_0 respectively.
Explanation
The image after translation is same as the one before translation, but the "canvas" should be enlarged, or the visual information will be lost.

4.1.2.Mirror

Concept

Flip around x axis or y axis, and produce a new image symmetric to the original one.

Equation

Flip around x axis:

$$\begin{cases} x' = x \\ y' = -y \end{cases}$$

Flio around y axis:

$$\begin{cases} x' = -x \\ y' = y \end{cases}$$

OR

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

When $s_x = 1$, and $s_y = -1$, flip around the x axis
When $s_x = -1$, and $s_y = 1$, flip around the y axis

4.1.3.Rotation

Concept

Rotate the image around the origin θ, and produce a new image.

Equation

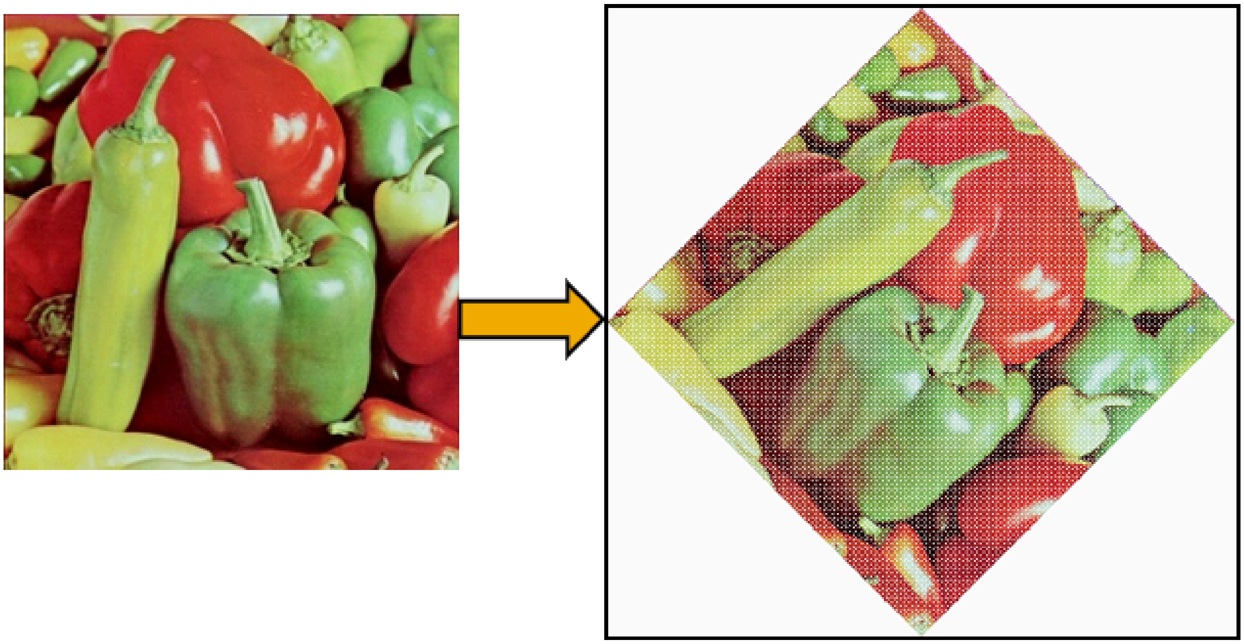
$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \end{cases}$$

OR

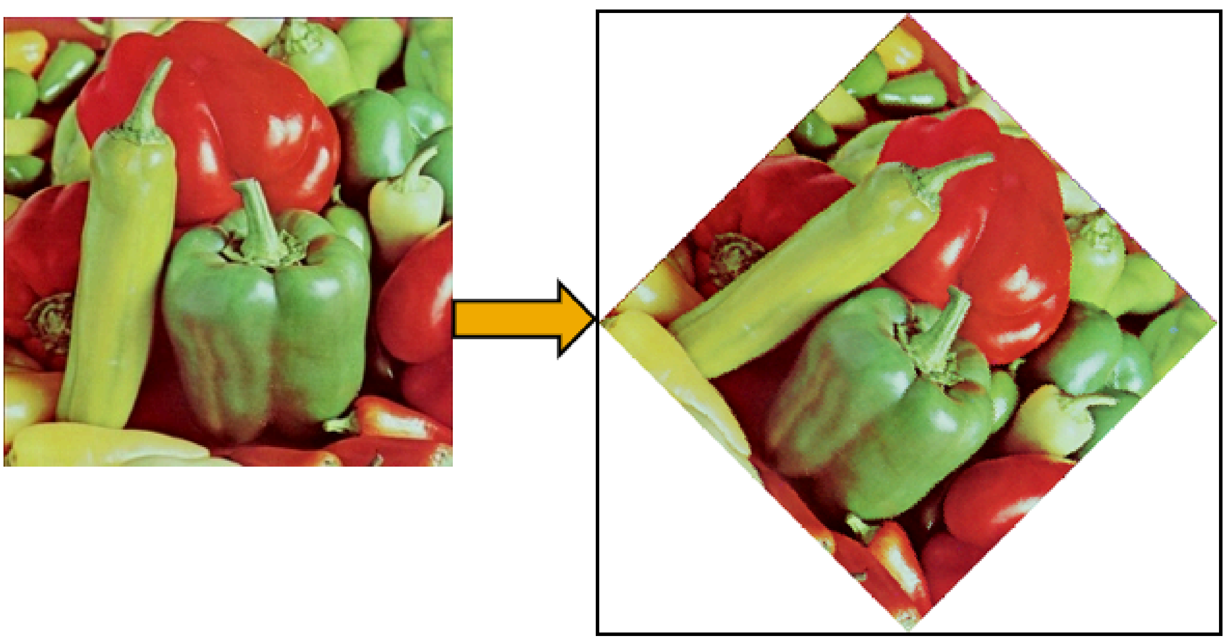
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

hole problem

After rotation, there will be a lot of "holes" in the new image.



Fill by "interpolation": Row interpolation —— find the holes in each row, and fill them with the intensity of the pixel before it in the row.



4.1.4.Scale

Concept

Multiple the image with a coefficient to produce a new image.

Equation

$$\begin{cases} x' = cx \\ y' = dy \end{cases}$$

OR

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} c & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Scale the image horizontally with coefficient c (enlarge when $c > 1$, shrink when $0 < c < 1$); scale the image vertically with coefficient d (enlarge when $d > 1$, shrink when $0 < d < 1$)
If $c = d$, the image is scaled with the same ratio. If NOT, the image is scaled horizontally and verticaly with different ratios.
Shrink(down-sampling) —— construct a new image by select pixels by predefined intervals from the original image.
Enlarge(a simple up-sampling) —— there will be blank rows and columns. If you fill them by using interpolation, you will find some "mosaics" in the result.

4.1.5.Shear

Concept
It is a non-vertical projectin effect of the scene on the plane.
Equation
Shear on x aixs:

$$\begin{cases} a(x,y) = x + d_x y \\ b(x,y) = y \end{cases}$$

Shear on y axis:

$$\begin{cases} a(x,y) = x \\ b(x,y) = y + d_y x \end{cases}$$

4.1.6.Combination

Concept
Combine the simple geometric transformation together.
Equation

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Transformation matrix is obtained by taking into account all the simple geometric transformations.

4.2.Interpolation

Interpolation is a typical tool for geometric transformation.
The interpolated pixel value can be obtained by using different interpolating methods.

4.2.1.Nearest neighbor

Concept

The output pixel value equals to its nearest neighbor's grayscale intensity.

Calculation process

To calculate the value at a certain point P' in the new image after geometric transformation, the inverse transformation of the geometric transformation can be calculated first to determine the position P in the original image corresponding to P'. Normally, the position of P cannot be exactly at a pixel position in the original image (i.e. the coordinates of point P are usually not exactly integers). Find the pixel Q that is closest to point P, and use the pixel value of point Q as the pixel value of point P' in the new image.

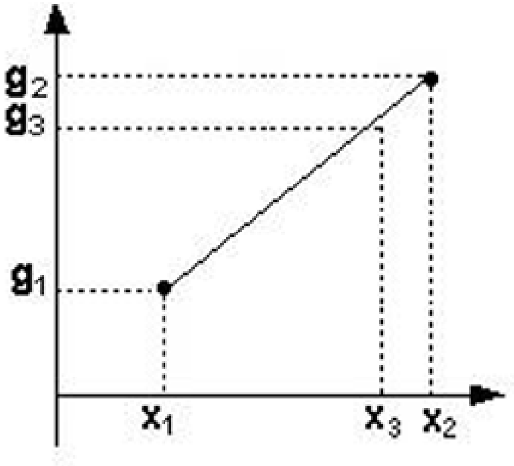
$$(x', y') \overset{\text{inverse transformation}}{\Longrightarrow} (x, y) \overset{\text{rounding operation}}{\Longrightarrow} (x_{int}, y_{int}) \overset{\text{assign value}}{\Longrightarrow} I_{new}(x', y') = I_{old}(x_{int}, y_{int})$$

limitations

The result cannot keep consistent and smooth when the image consists obvious geometric structures.

4.2.2.Linear interpolation

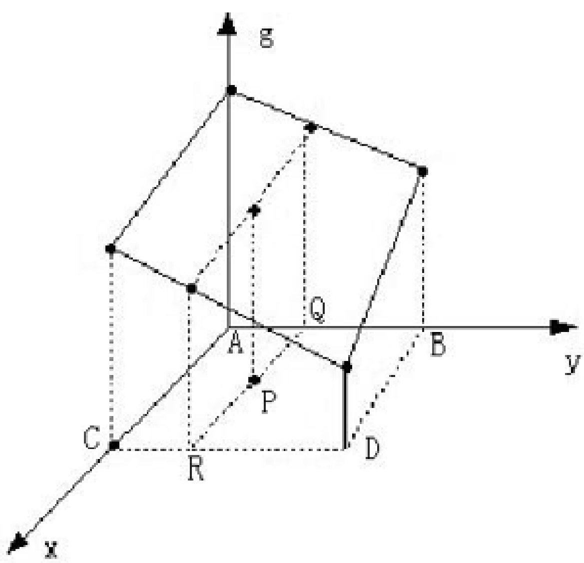
Equation(1D)



The grayscle value at x_1 and x_2 are g_1 and g_2 respectively. Then the grayscle value at x_3 should be:

$$g_3 = \frac{g_2 - g_1}{x_2 - x_1}(x_3 - x_1) + g_1$$

Equation(2D)



Linear interpolation in 2D domain called bilinear interpolation.
The gratscale values at A, B, C and D are known. How to get the grayscale value at P?
1.Define the bilinear equation $g(x, y) = ax + by + cxy + d$;
2.Substitude the coordinates and grayscale intensity of A, B, C and D into the equation, and equation system can be obtained.
3.Solve the equation system to obtain a, b, c, and d.
4.Substitude the coordinate of P into the equation and obtain its grayscle intensity.

4.2.3.RBF interpolation

Radial Basis Function (RBF) based interpolation

$$G(x) = \sum_{i=1}^{n} \omega_i G(c_i)$$

where $\omega_i = \frac{\phi(|x-c_i|)}{\sum\limits_{i=1}^{n} \phi(|x-c_i|)}$

where x is a scalar or a vector. Hence, RBF can do 1D interpolation, 2D interpolation or high dimensional interpolation. It depends on the dimensionally of x.

A redial basis function(RBF) is a function based on a scalar radius.

$$\phi(r) = \phi(|x - x_i|)$$

Gaussian and Multiquadrics functions have adjustable parameter σ.

Kernel functions available in RBF

Gaussian:

$$\phi(r) = \exp(-\frac{r^2}{2\sigma^2})$$

Multiquadrics:

$$\phi(r) = \sqrt{1 + \frac{r^2}{\sigma^2}}$$

Linear:

$$\phi(r) = r$$

Cubic:

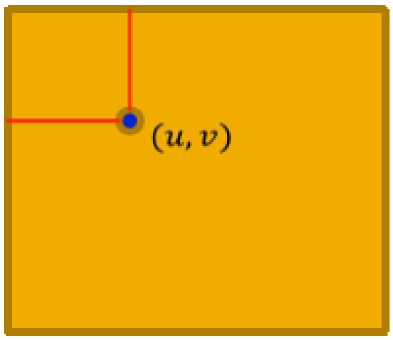
$$\phi(r) = r^3$$

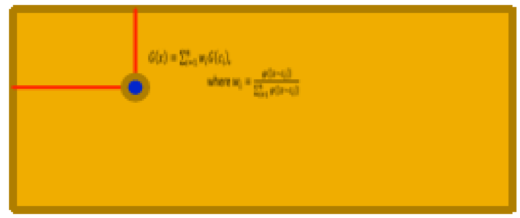
Thinplate:

$$\phi(r) = r^2 \ln(r + 1)$$

4.3.Warp and morph

Warp: only change the pixel's location





Concept of Image Morph

Morph is not warp(geometric transformation)

Morph is a kind of morphological changing, which makes an image change to another image gradually.

Morph handles both the location and the intensity of a pixel.

The beginning image and end image are two key frames. Other frames between the two key frames are generated automatically.

Morph

Change image A to image B

Concept: change each pixel's color in image A to be that of the corresponding pixel in image B gradually.

Method: According to the number of frames between image A and B, set the step length between such interpolation frames. You can use the interpolation method here.

For example, if you want to produce n intermediate frames, you just use $I_i = I_{begin} + i * \frac{I_{end} - I_{begin}}{n}$ to obtain the intermediate pixel value.

Given a color image, RGB should be taken into account together.

$$r_{i,j} = r_{i,j}^a + \frac{r_{i,j}^b - r_{i,j}^a}{N} * n, n = 0, 1, ..., N-1$$

$$g_{i,j} = g_{i,j}^a + \frac{g_{i,j}^b - g_{i,j}^a}{N} * n, n = 0, 1, ..., N-1$$

$$b_{i,j} = b_{i,j}^a + \frac{b_{i,j}^b - b_{i,j}^a}{N} * n, n = 0, 1, ..., N-1$$

How to choose the key frames

The two key frames should be similar in structure and size, so as to obtain a natural and consistent morphing result.

How to obtain the structure

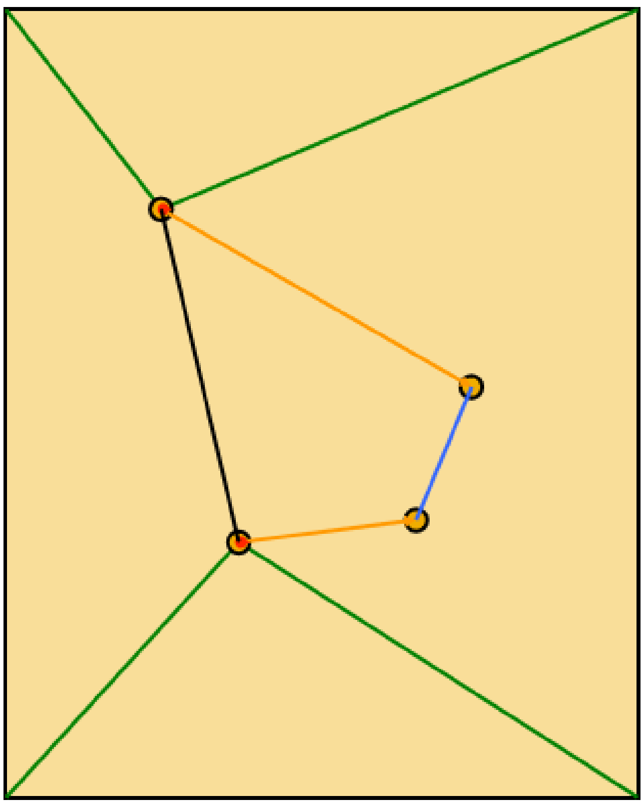
Label a series of feature points to define the structure of the beginning and end frame.

The feature points should keep correspondence.

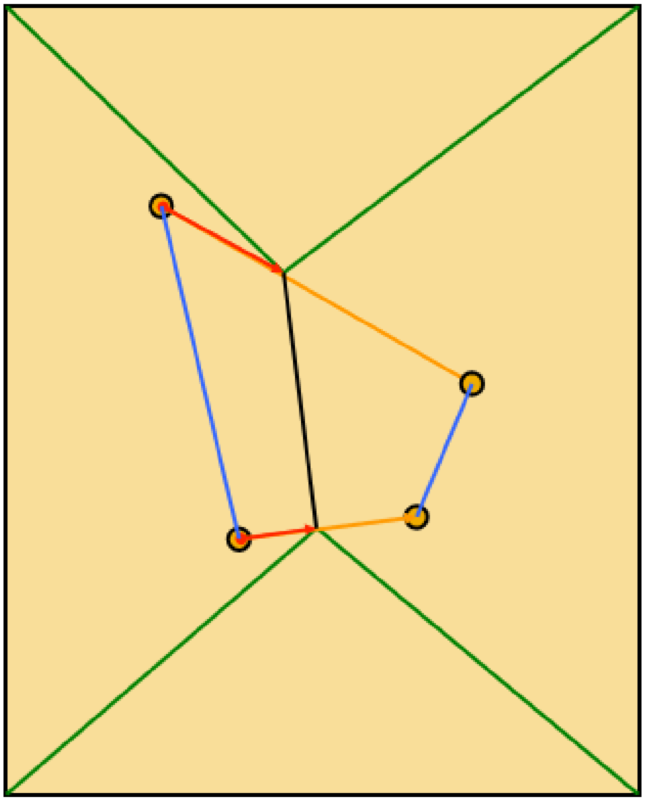
Morph based on segment(two points)

After the feature points are determined, a series of segments are obtained.

If we want to morph segment A to be B, both the location and the color of each pixel in the image will chabge accordingly.



Different interpolation methods can be employed.



4.4.Application

Lambertian model

Assume there are m point light sources:

$$I = \rho \sum_{1 \le i \le m} S_i I_i n l_i \equiv \rho E(n)$$

where $S_i = \begin{cases} 0, if - the - point - cannot - be - seen - from - light - i \\ 1, otherwise \end{cases}$

Expression ratio image

Before deformation:$I = \rho E(n)$

After deformation:$I' = \rho E(n')$

e xpression ratio image:$\frac{I'}{I} = \frac{E(n')}{E(n)}$

ERI :

captures illumination changes caused by surface deformation

is material independent

Surface deformation mapping

Two surfaces:

same normals at the corresponding points
different materials
same deformations

	Before deform.	After deform.
Surface 1:	$I_1 = \rho_1 E(n)$	$I_1' = \rho_1 E(n')$
Surface 2:	$I_2 = \rho_2 E(n)$	$I_2' = \rho_2 E(n')$

$$\frac{I_1'}{I_1} = \frac{I_2'}{I_2} \Rightarrow I_2' = \frac{I_1'}{I_1} I_2$$

Expression mapping

Assumption:
Human faces have approximately the same normals

	Neutral	Expression
Person A:	I_a	I_a'
Person B:	I_b	$I_b' = \frac{I_a'}{I_a} I_b$

Chapter 5:Image convolution

5.1.Continuous 1-D

Definition

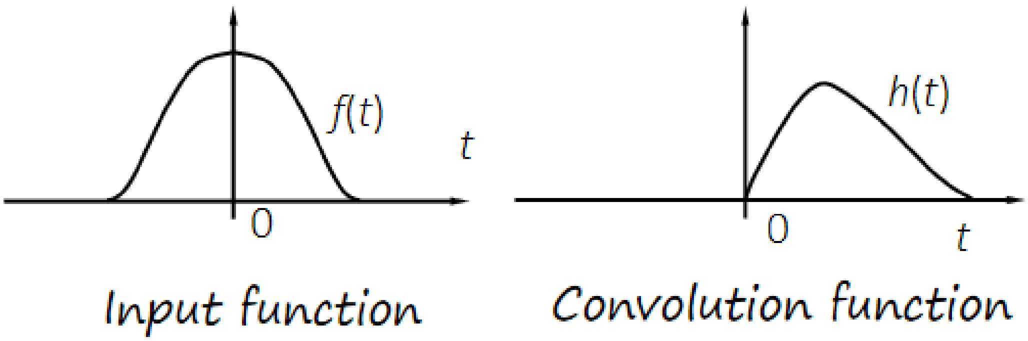
The convolution $g(x)$ of two 1D functions $f(x)$ and $h(x)$

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(t)h(x-t)dt$$

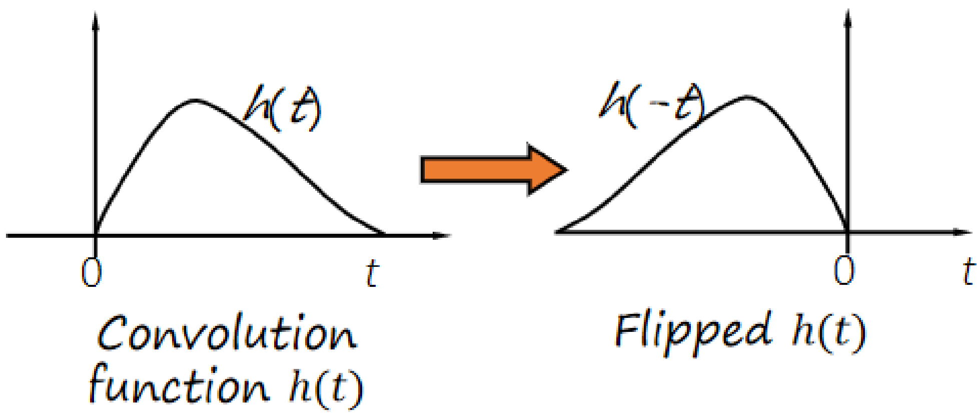
which means the convolution of two 1D functions can formulated as an integral of multiplication.
Usually we call $f(x)$ as input function, $h(x)$ as convolution function.

Computing process

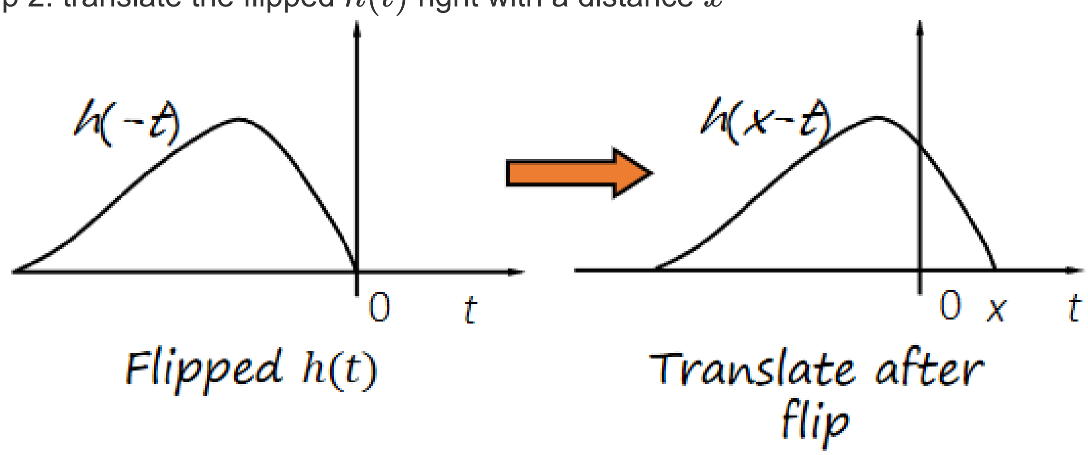
Given two functions, how to compute their convolution result at an arbitrary point x



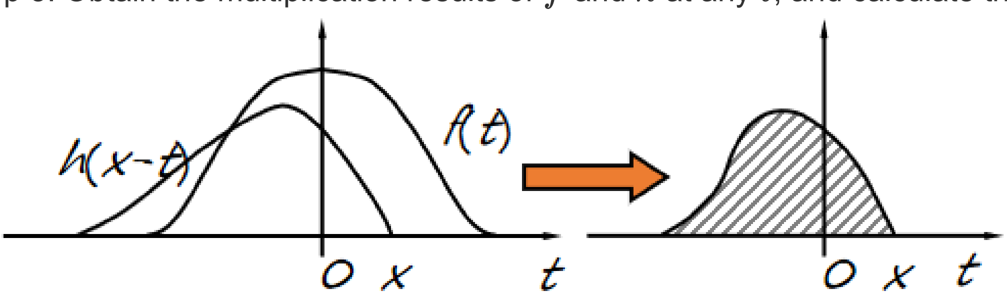
Step 1: flip the convolution function $h(t)$



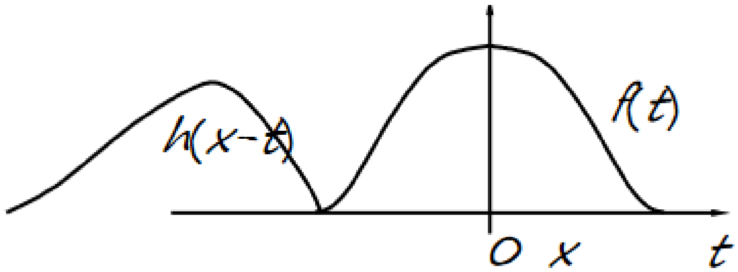
Step 2: translate the flipped $h(t)$ right with a distance x

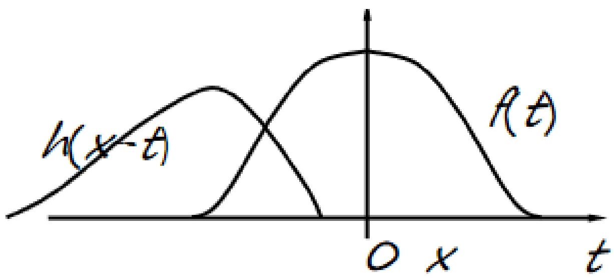


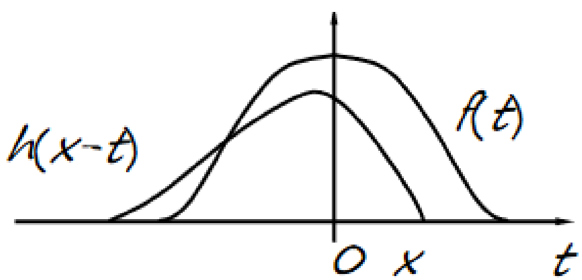
Step 3: Obtain the multiplication results of f and h at any t, and calculate their integral.

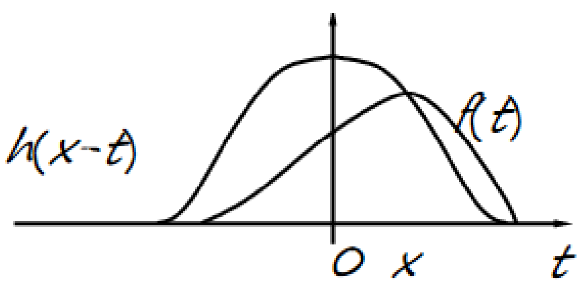


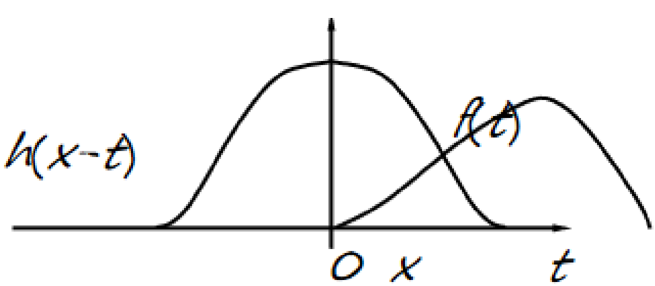
a) Repeat such computing process for each x, you will get an output curve $g(x)$
b) When x changes, the flipped function $h(x-t)$ go through the input function $f(t), g(x)$ depends on the integral of the overlap section of these two functions.

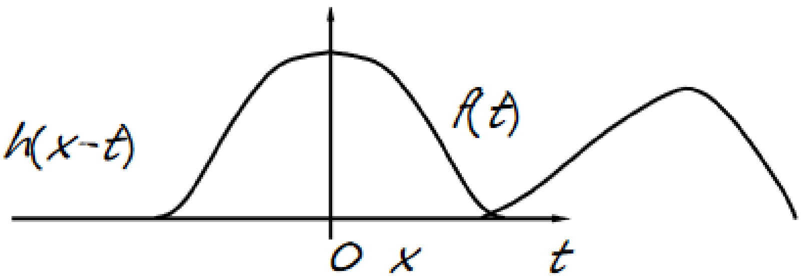












5.2.Property of convolution

Exchangeable

$$g(x)=f(x)*h(x)=\int_{-\infty}^{\infty} f(t)h(x-t)dt$$

$$f(x)*h(x)=h(x)*f(x)$$

Proof:

$$f(x)*h(x)=\int_{-\infty}^{\infty} f(t)h(x-t)dt$$

Let $s=x-t, t=x-s, ds=-dt$
We have

$$f(x)*h(x)=\int_{-\infty}^{\infty} f(x-s)h(s)ds=h(x)*f(x)$$

Distributive

$$f*(g+h)=f*g+f*h$$

Proof:

$$f*(g+h)=\int_{-\infty}^{\infty} f(x-t)[g(t)+h(t)]dt=\int_{-\infty}^{\infty} f(x-t)g(t)dt+\int_{-\infty}^{\infty} f(x-t)h(t)dt$$

$$\Rightarrow f*(g+h)=f*g+f*h$$

Associative

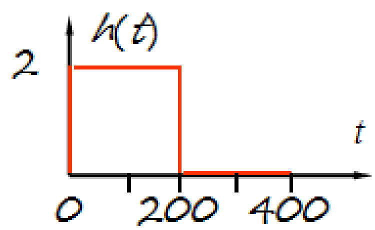
$$f * (g * h) = (f * g) * h$$

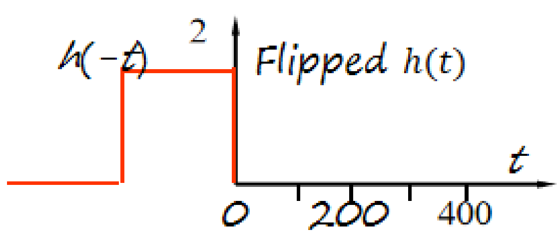
5.3.Discrete convolution

For a discrete sequence, the integral can be treated as summing up problem.
For $f(x)$ and $h(x)$ whose lengths are M, their convolution is:

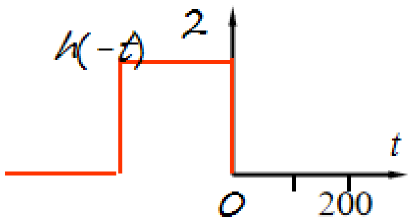
$$g(x) = f(x) * h(x) = \frac{1}{M} \sum_{i=0}^{M-1} f(t)h(x - t)$$

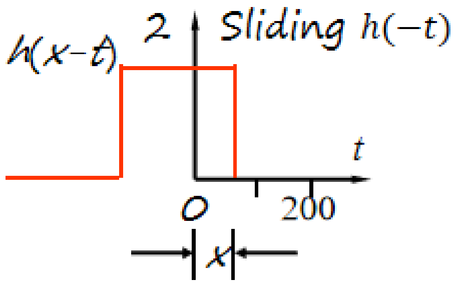
Step 1: Flip $h(t)$ to be $h(-t)$



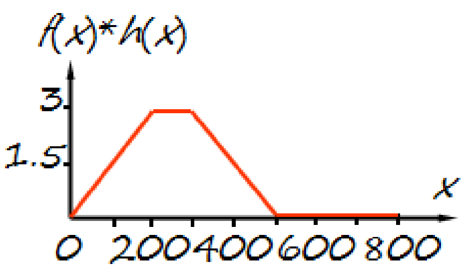


Step 2: Slide $h(-t)$ by adding x, i.e., $h(x - t)$

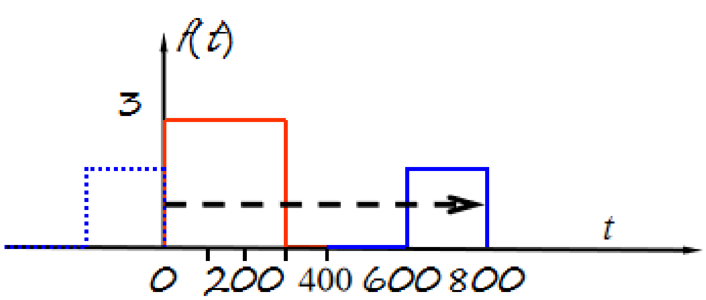




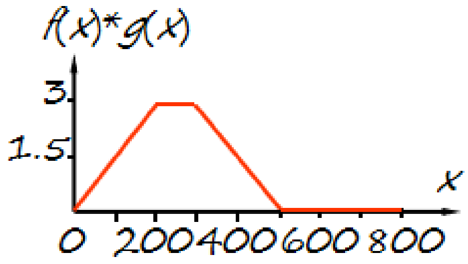
Step 3: For each sample $t(t=0,1,2,...,399)$, compute $f(t)h(x-t)$;Sum up all the $f(t)h(x-t)$ and divide it by $M(M=400)$. Now you get the convolution at x.
Step 4: Compute the convolution value for each x in the "effective range". You get a curve of $g(x)$.



Effective range:
Covers all the points that $h(t)$ go through during the sliding operation.
In this example, the effective range of x is from 0 to 799.



Hence the convolution $g(x)$ is also a function and its variable is x.



In the plain words, convolution is essentially equivalent to computing a weighed sum of image pixels.

5.4.Spatial domain filtering

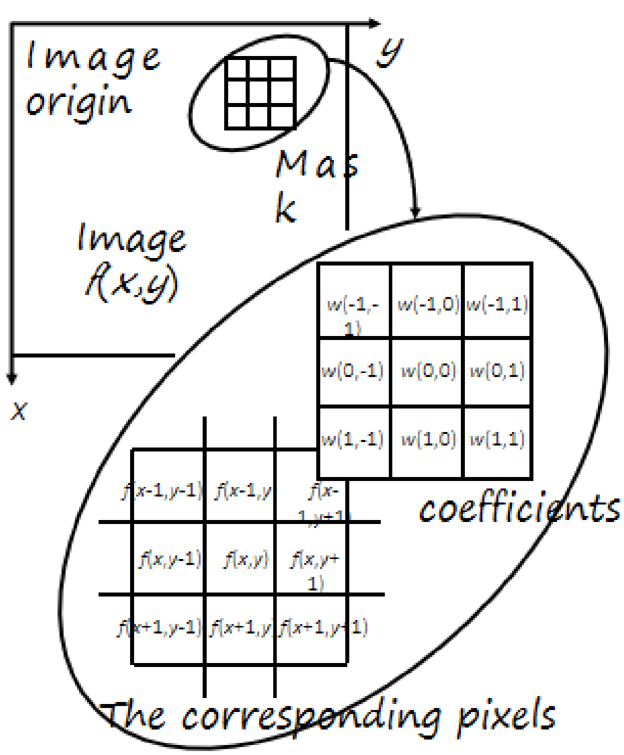
Concept

Filter is a windows sized by $M \times N$, wherein the elements in the windows take operation on the corresponding pixels from the original image in the windows. And the results are saved as the pixels in a new image.

Alias: Filter, mask, kernel, template, window

The elements in the filter are coefficients instead of pixel values, ehich represent the weight applied on the pixels in the original image.

Procedure



For each pixel (x, y) in the image, its response value to the filter is computed in terms of the predefined relationship between the elements in the filter.

For spatial linear filtering, the response value is computed by summing up the multiplication between the coefficient and its corresponding pixel.

Response

Usually, the width and the height of the mask are odd number such as $2a+1$ and $2b+1$. When the mask window is centered at (x,y), the updated value of (x,y) is computed as:

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

After updating all the pixels in the image, you will obtain a new image g.

Simplified form:

$$R=\sum_{i=1}^{mn}\omega_i z_i$$

Spatial filtering for smoothing

When you find there are too many artifacts or noises in the image, you can carry out smoothing operation on the image to suppress the noise and reduce the artifacts. However, the smoothing operation will make the image blurring.

Smoothing can reduce noises and blurring, which can be used in preprocessing. For instance, remove the subtle details when you just want to extract the big target.

5.4.1.Linear smoothing filter

Concept

The output of the linear smoothing filter is the mean value of the pixels in the mask. It's also called mean filter.

application

Mean filter is mainly used for subtle detail removal, namely, eliminating the unwanted region smaller than the mask.

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

where the size of the fulter is $(2a+1)\times(2b+1)$, ω is the filter, f is the input image, g is the output image.

Explanation

The size of the mask is very important for the final result. When the mask is small, the blurring effect is very subtle, and vice versa.

Application

In order to obtain a brief description of the object of interest, linear smoothing filter is used to blur the image to remove the smaller object while keeping the larger object.

Hence, the size of the mask depends on the object to be merged into the background.

5.4.2.Statistical sorting filter

Concept

Statistical filter is a kind of nonlinear spatial filter, whose response is based on the sorting of center pixel depends on the sorting result in the windows.

The most popular statistical filter is median filter.

Median filter

Subtitute the center pixel with the median value in the neighborhood.

Provide excellent de-noide ability, which introduce less blurring than the mean filter.

Be effective to deal with the pulse noise (or pepper noise) because this kind of noise looks like bright or dark point in the image.

The median value ξ —— in a set of numbers, about half of them are smaller than ξ, others larger than ξ.

In order to perform median filtering on a pixel in the image, pixel sorting should be carried out in the mask window to select the median value and change the pixel value to the median value.

Usually we use $n \times n$ median filter to remove the unwanted brighter or darker pixels in the neighborhood, and their area is less than $n^2/2$(half of the mask window).

Purpose

Enhance the detail or sharpen the blurred part in the image.

Tool

Differential operator is a sharpening tool, whose response depends on the variation between neighborhood pixel values.Di

fferential Operator strengthens the edges and other obvious variations (including noise) in the image, and weakens the slight changes.

Sharpening operator

Concept of differential operator

Second order differential based image enhancement —— Laplacian operatorF

irst oeder differential based image enhancement —— gradient based method

Differential

For a function $f(x)$, we use difference to represent the differential operator:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Similarly, the second order differential is:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

gradient based method

For a function $f(x, y)$, we first define a 2-D vector:

$$\nabla f = [\frac{G_x}{G_y}] = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Its magnitude is computed as:

$$\nabla f = [G_x^2 + G_y^2]^{\frac{1}{2}} = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{\frac{1}{2}}$$

It's time consuming to compute the gradients for all the pixels in the image. Hence, absolute value is usually used to replace the original gradient magnitude.

$$\nabla f \approx |G_x| + |G_y|$$

Another approach:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Original image, z_i is the pixel value, z_5 is the center pixel.

-1	0
0	1

0	-1
1	0

Robert cross gradient operator, $G_x = (z_9 - z_5), G_y = (z_8 - z_6)$

$$\nabla f = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{\frac{1}{2}}$$

$$\nabla f \approx |z_9 - z_5| + |z_8 - z_6|$$

Laplacian operator

For a function $f(x, y)$, Laplacian operator is defined as:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The discrete Laplacian operator:

Along x aixs:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

Along y axis:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

Hence, the discrete Laplacian operator is:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

Mask of Laplacian operator:

0	1	0
1	-4	1
0	1	0

It is rotation invariant.

Extending the mask:

The elements in the diagonal direction can also be taken into account:

$$\nabla^2 f = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(x+i, y+i) - 9f(x, y)$$

OR

$\nabla^2 f = [f(x-1, y-1) + f(x, y-1) + f(x+1, y-1) + f(x-1, y) + f(x+1, y) + f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)] - 8f(x, y)$

1	1	1
1	-8	1
1	1	1

It's also rotation invariant, i.e., isotropic.

When you fuse the Laplacian result and the original image together, you have to consider the symbol difference between them.

Image enhancement by Laplacian:

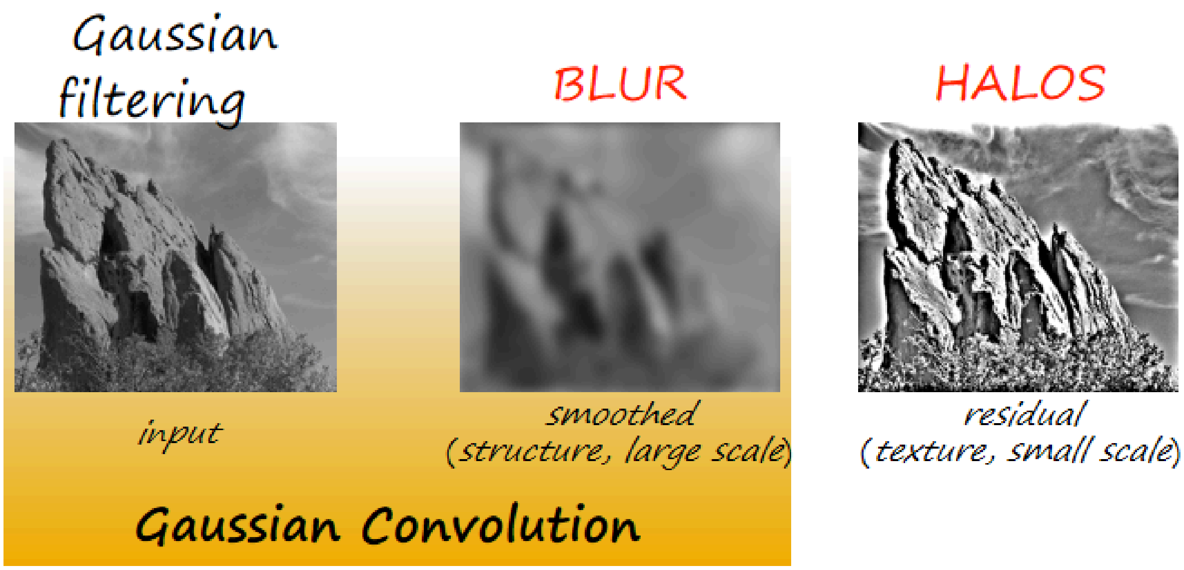
$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{If the center element of the mask is negative} \\ f(x,y)+\nabla^2 f(x,y) & \text{If the center element of the mask is positive} \end{cases}$$

Fuse the original image and the Laplacian result can preserve the sharpening effect and restore the original visual information.

5.4.3.Image Smoothing

Split an image into:

- large-scale features, structure
- small-scale features, texture



Impact of Blur and Halos

If the decomposition introduces blur and halos, the final result is corrupted.

Sample manipulation: increasing texture (residual ´ 3)

Bilateral filter: General Idea

An image has two main characteristic

The space domain S, which is the set of possible positions in an image. This is related to the resolution, i.e., the number of rows and columns in the image.

The intensity domain R, which is the set of possible pixel values. The number of bits used to represent the pixel value may vary. Common pixel representations are unsigned bytes (0 to 255) and floating point.

Every sample is replaced by a weighted average of its neighbors

These weights reflect two forces:

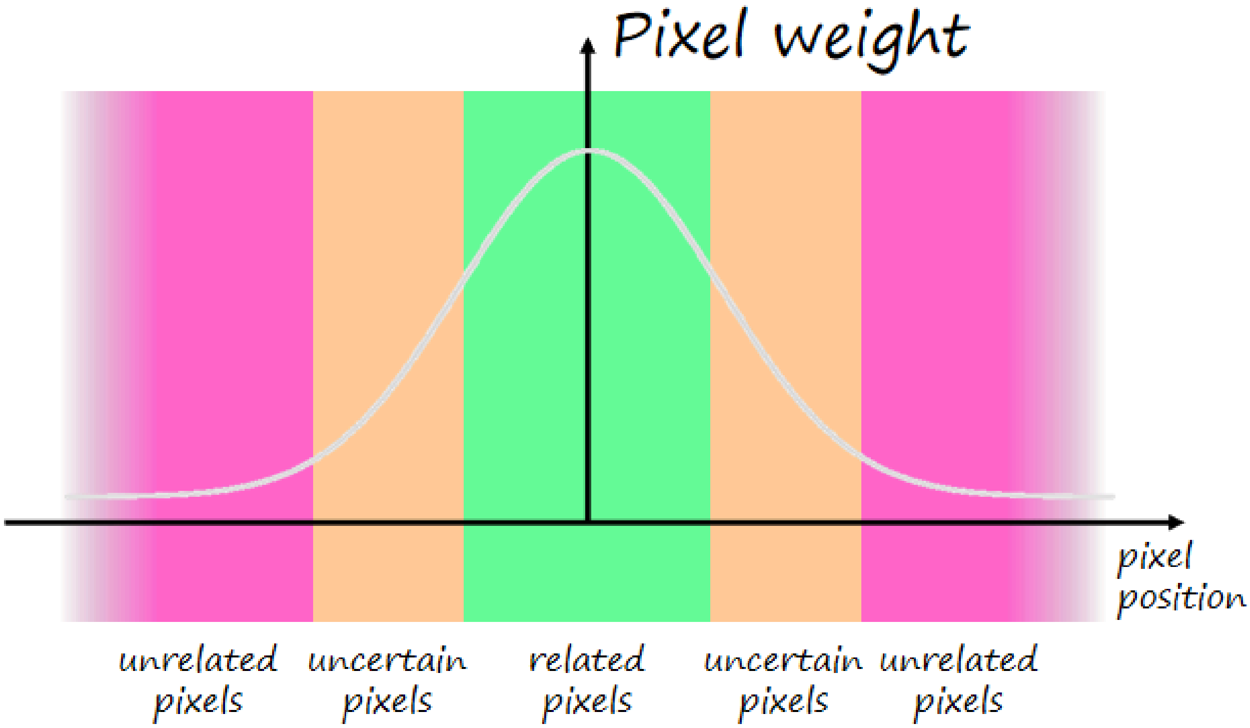
- How close are the neighbor and the center sample, so that larger weight to closer samples,
- How similar are the neighbor and the center sample – larger weight to similar samples.

All the weights should be normalized to preserve the local mean.

Revisit Gaussian Blur:

$$GB[I]_p = \sum_{q \in S} G_\sigma(\|p-q\|) I_q$$

Gaussian Profile:



$$G_\sigma = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$$

How to set σ ?

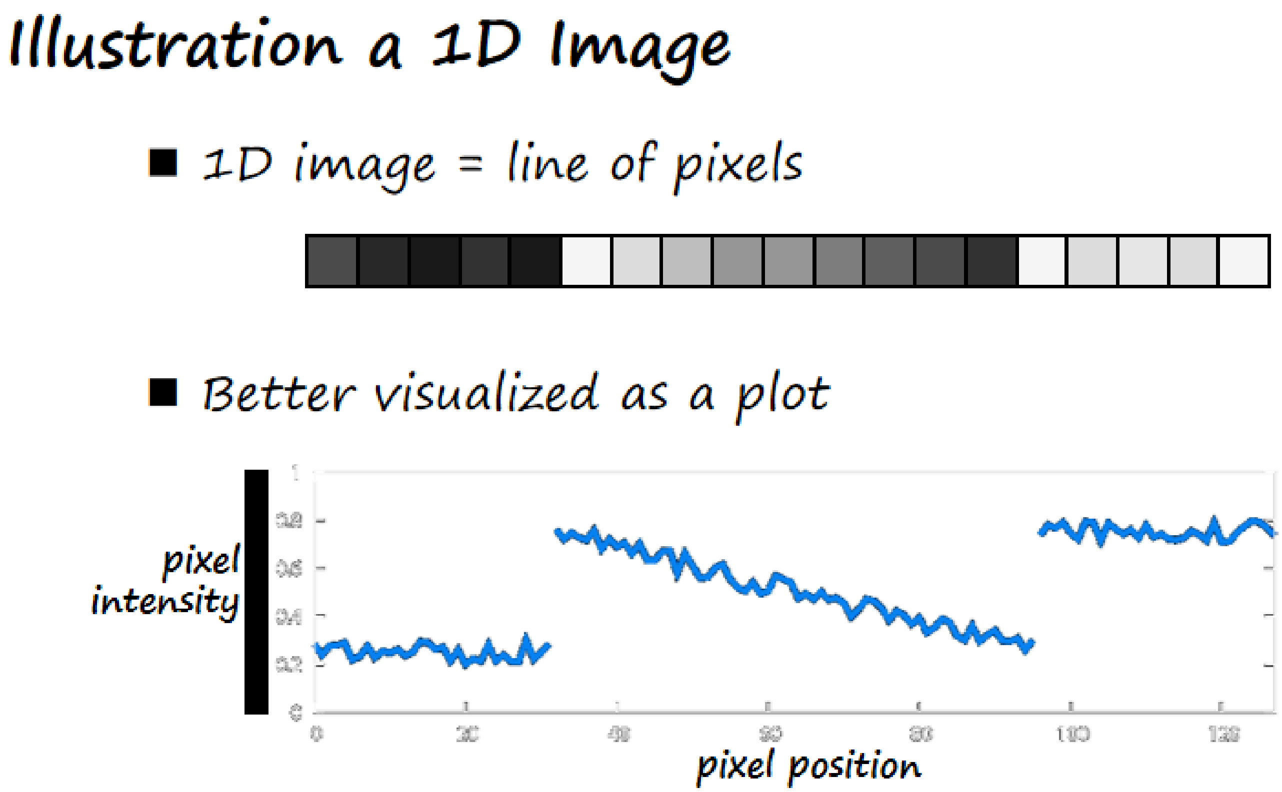
- Depends on the application
- Common strategy: proportional to image size: e.g. 2% of the image diagonal; property: independent of image resolution

Properties of Gaussian Blue:

- Does smooth images
- But smoothes too much: edges are blurred. Only spatial distance matters. No edge term.

Bilateral Filter Definition: an Additional Edge Term

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(\|I_p - I_q\|) I_q$$



space σ_s: spatial extent of the kernel, size of the considered neighborhood.
intensity σ_r: amplitude extent of an edge
Only pixels close in space and in intensity are considered.

How to Set the Parameters?
Depends on the application. For instance:

- space parameter: proportional to image size : e.g. 2% of image diagonal
- intensity parameter: proportional to edge amplitude: e.g. mean or median of image gradients
- independent of resolution and exposure

Iterating the Bilateral Filter

$$I_{(n+1)} = BF[I_{(n)}]$$

Generate more piecewise-flat images
Often not needed in computational photo.

Bilateral Filtering Color Images

For gray-level images:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q$$

For color images:

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|C_p - C_q|) C_q$$

The bilateral filter is extremely easy to adapt to your need.

5.4.4.Overview

Denoising

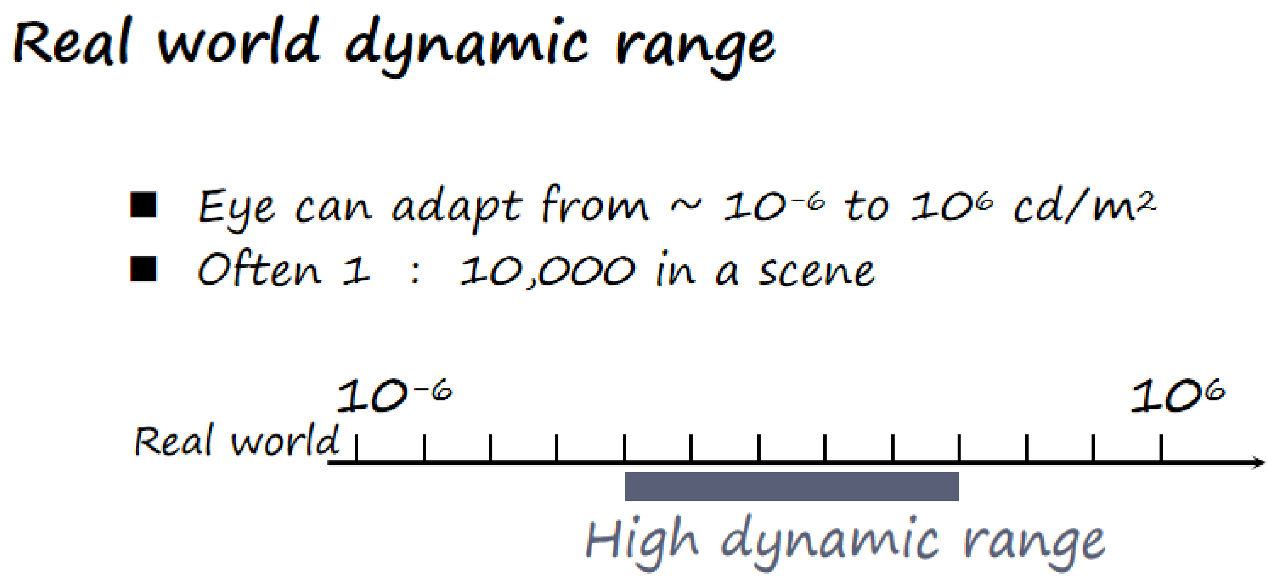
Small spatial sigma (e.g. 7x7 window)

Adapt intensity sigma to noise level

Maybe not best denoising method, but best simplicity/quality tradeoff:

- No need for acceleration (small kernel)
- But the denoising feature in e.g. Photoshop is better

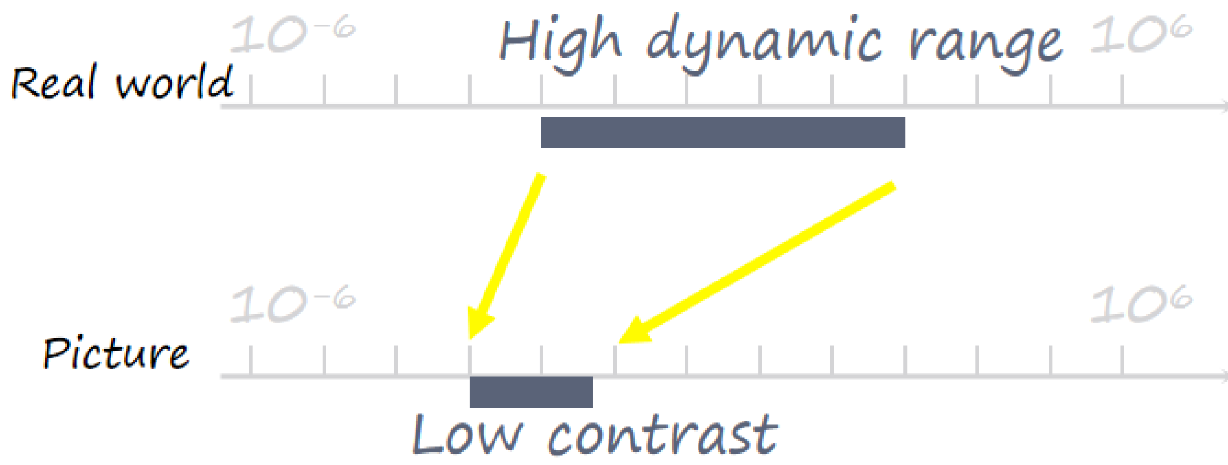
Tone mapping



Problem: Contrast reduction

Match limited contrast of the medium

Preserve details



Input: high-dynamic-range image (floatting point per pixel)

Naive technique: Scene has 1:10,000 contrast, display has 1:100

The halo nightmare: For strong edges; Because they contain high frequency

Bilateral filtering to the rescue
Large scale = bilateral (log intensity)
Detail = residual

Hard to Compute
Nonlinear
Complex, spatially varying kernels: Cannot be pre-computed
Brute-force implementation is slow > 10min

Relight & texture editing

Chapter 6:Image filtering

Smoothing
Sharpening
Bilateral filter
Guided filter(Optional)

Chapter 7:Fourier transform

Fourier and his work
Background of Fourier transform
Fourier transform

Discrete Fourier transform(1D)

FFT

FFT in Matlab

2D Fourier transform

Chapter 8:Image feature

SIFT

SURF

Deep feature