Introduction to Digital Image Processing
数字图象处理

主讲老师：　　张　荣

Purpose and effects of this course

- learn basic theories and methods of image processing
- easy to use image processing software and equipments
- can develop software and hard ware of image processing
- establish foundation for further research

References: Books

- 章毓晋 《图象处理和分析》清华出版社
- 刘政凯 《微型计算机图象处理技术》安徽科技出版社
- 《计算机图象处理技术》 浙江大学出版社
- K．R．Castleman 《Digital Image Processing》清华出版社,中文版电子工业出版社发行
- R. C. Gonzales ，《 Digital Image Processing Second Edition》电子工业出版社

教材: R. C. Gonzales

- Chapter1: Introduction
- Chapter2: Fundaments of Image and Vision
- Chapter3: Image Transforms
- Chapter4: Image Enhancement
- Chapter5: Image Restoration and Reconstruction
- Chapter6: Image Coding
- Chapter7: Image Segmentation
- Chapter8: Object Representation and description

References: Journals

- 《电子学报》
- 《通信学报》
- 《自动化学报》
- 《遥感学报》
- 《中国图象图形学报》
- 《人工智能与模式识别》

References: Journal

- IEEE Transactions on
  - Image Processing
  - Signal Processing
  - Medical Image
  - Geoscience and Remote Sensing
  - Pattern analysis and machine intelligence
- Image and Vision Computing
- Computer Vision and Image Understanding

Chapter1 Introduction

- 1.1 What Is Digital Image Processing
- 1.2 The Origins of Digital Image Processing
- 1.3 Example of Fields that Use Digital Image Processing
- 1.4 Fundamental Steps in Digital Image Processing
- 1.5 Components of an Image Processing System

1.1 What Is Digital Image Processing

- 1.1.1 Terms
  - **Picture(图片):** A visual representation or image painted, drawn, photographed, or otherwise rendered on a flat surface
  - **Image(图象):** a two-dimensional function, $f(x,y)$
  - **Digital image(数字图象):** when x, y, and f are all discrete quantities.
  - **Video(视频):** sequence of image
  - **Pixel(象素):** the elements of digital image
  - **Digital image processing(数字图象处理):** processing $f(x,y)$ by a digital computer.

162	161	159	161	162	160	158	156	156	161
162	161	159	161	162	160	158	156	156	161
163	158	159	159	160	158	155	155	156	158
159	157	159	156	159	159	154	152	155	153
155	157	156	156	158	157	156	155	154	156
156	157	155	151	157	156	155	156	156	154
157	156	156	156	156	156	154	156	155	155
158	157	155	155	156	155	155	155	155	155
156	155	156	153	156	155	156	155	154	156
155	155	157	154	157	155	157	158	158	158

gray image

The pixels in the top left corner (10*10)

64	63	75	95	157	99
120	135	55	75	116	67
99	132	60	54	100	75
64	150	113	50	81	138
110	130	162	60	76	109
97	82	179	81	74	113

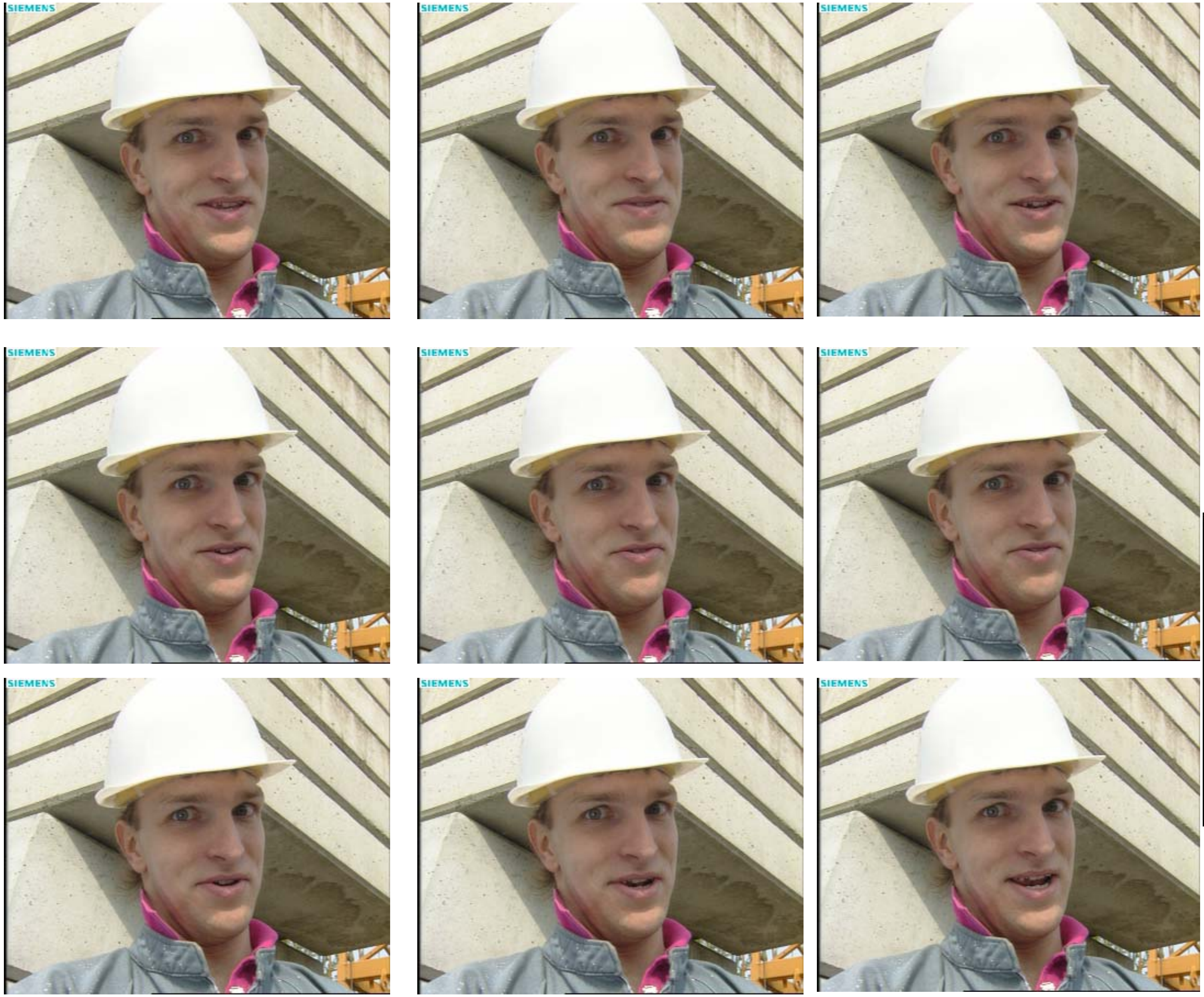
Color image

150	57	43	94	140	97
125	97	35	72	86	52
74	118	41	46	106	58
79	144	98	51	89	127
84	132	132	46	83	84
84	80	166	53	77	97

71	31	10	46	73	33
62	33	23	24	46	34
31	46	36	29	51	30
48	57	35	28	54	36
39	46	52	42	40	48
29	39	69	46	42	43

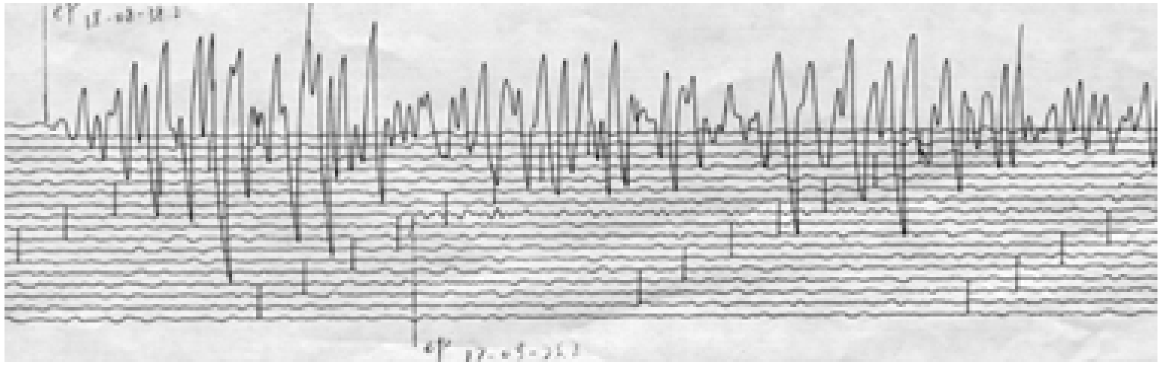
Pixels in RGB components

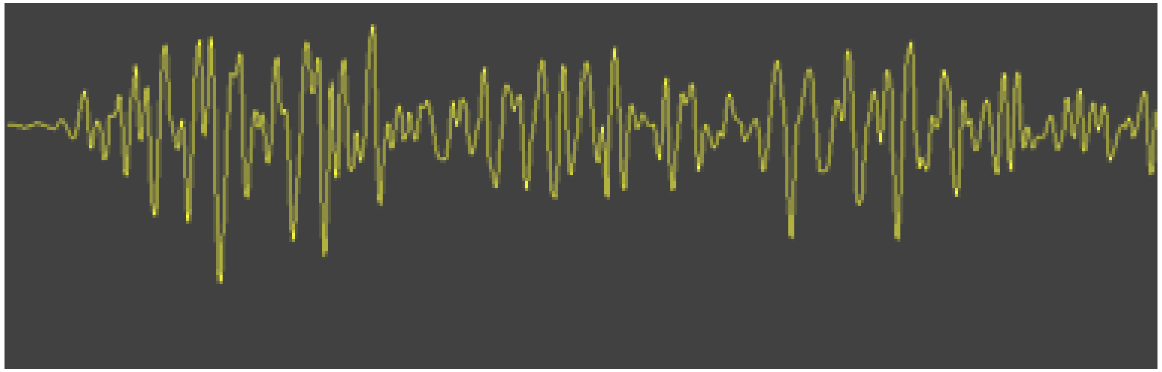
Left top (6*6)



video

1.1 What Is Digital Image Processing





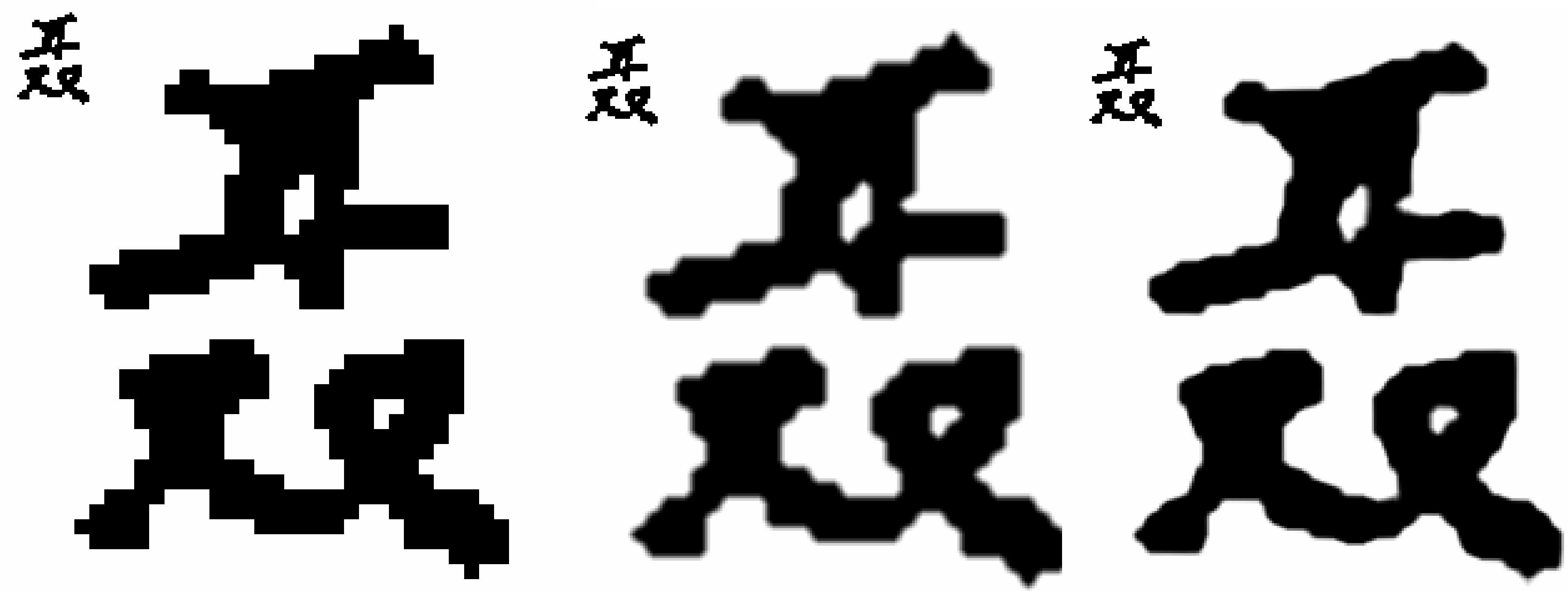
Eg1: Seismogram digitalization and database management system

1.1 What Is Digital Image Processing

后处理前后的图象比较

Eg2:Post-Processing of visual telephone

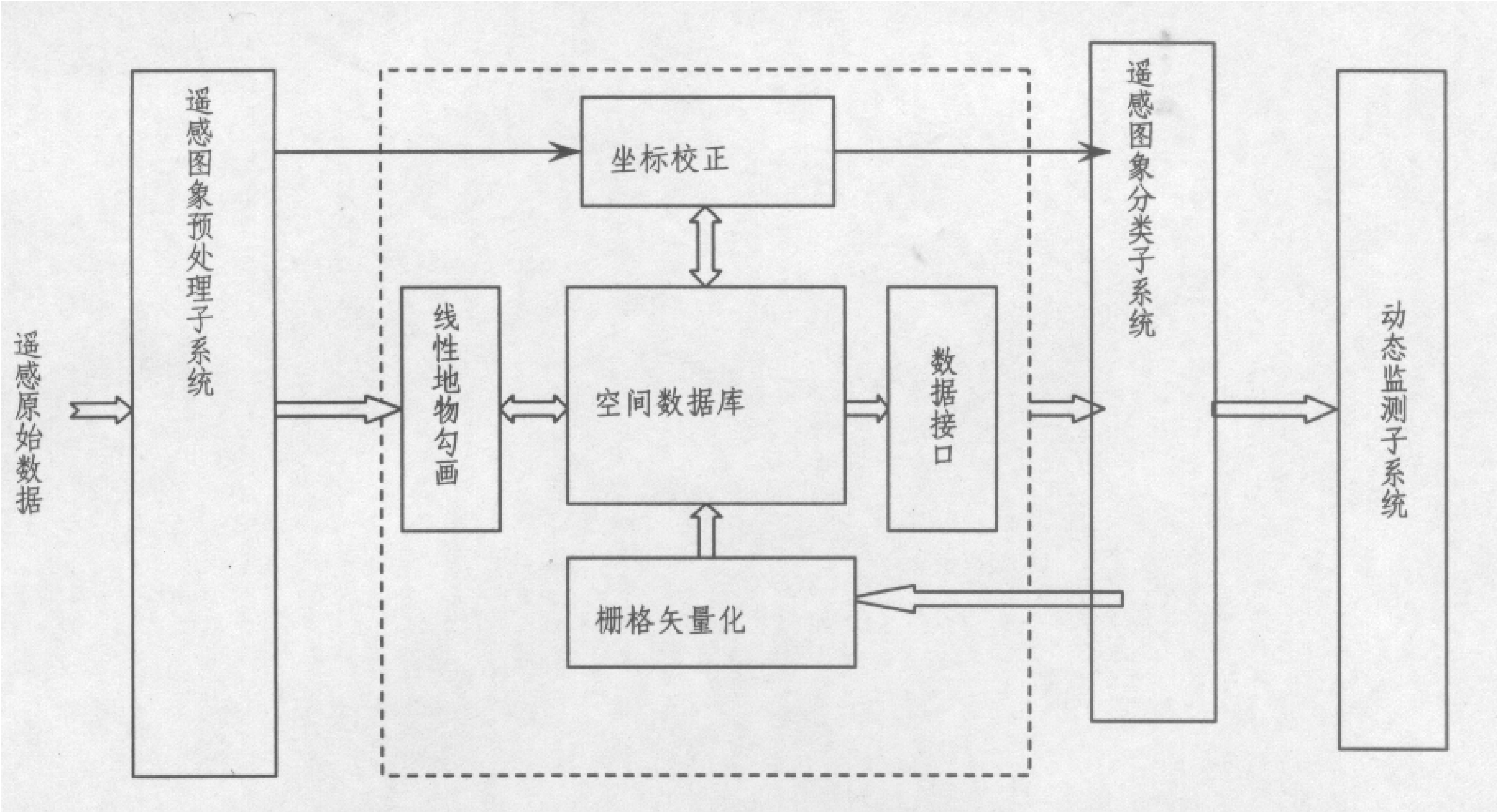
1.1 What Is Digital Image Processing



Scale enlargement Bi-linear interpolation enlargement Free scale enlargement

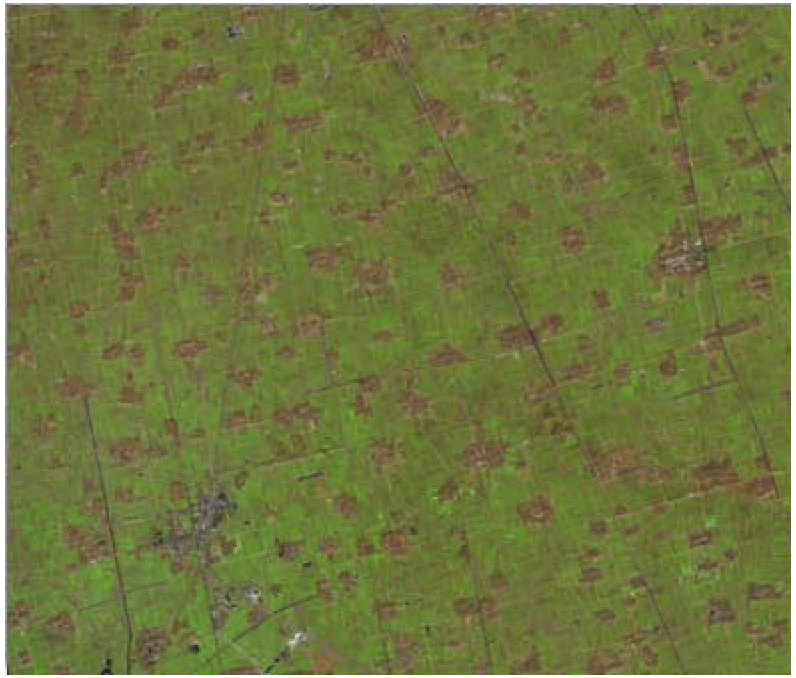
Eg3:Free-scale Enlargement of Handwriting Chinese Character

1.1 What Is Digital Image Processing



Eg4: Resource and Environment Dynamic Monitoring System of Anhui Province

1.1 What Is Digital Image Processing



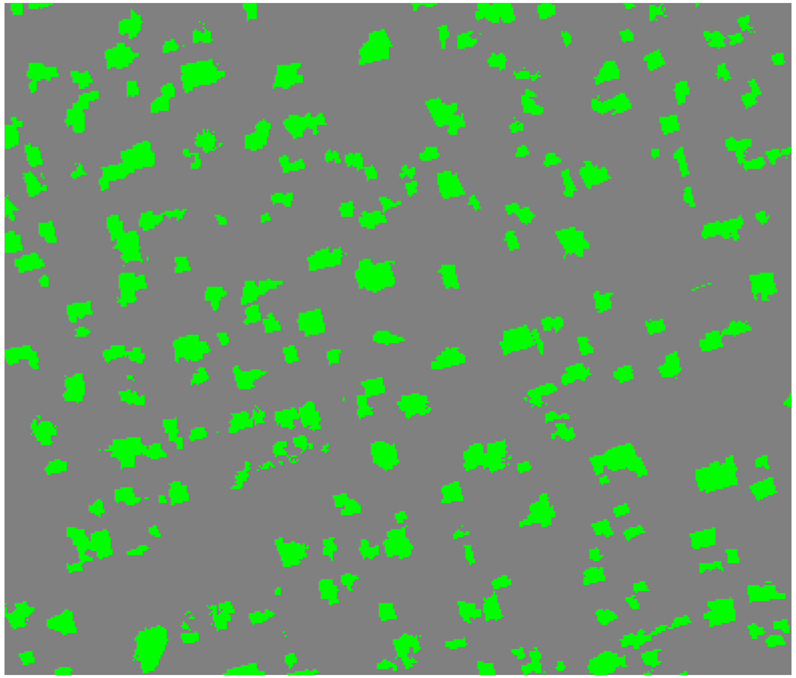


Fig.1 543 band false color image

fig2 result map

Eg4:Results of Adaptive cluster based on background database

1.1 What Is Digital Image Processing

43658 colors

55 colors

35120 colors

94 colors

Eg5: Color Quantization

1.1 What Is Digital Image Processing

1.1.2 contents in DIP

Digitizing
Store
Transmission
Display
Enhancement
Transform
Coding

Restoration
Reconstruction
Segmentation
Object detecting
Feature extracting
Object measurement

Image database
Classification
Recognition
Image model
Image marching
Image understanding

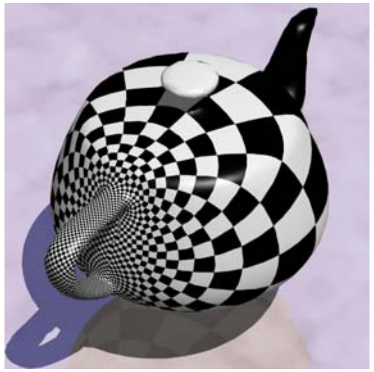
1.1 What Is Digital Image Processing

1.1.3 Three layers of DIP

- **Low-level processing:** inputs and outputs are images;
- **Mid -level processing:** inputs are images and outputs are attributes
- **High -level processing:** “making sense” , performing cognitive functions

1.1 What Is Digital Image Processing

1.1.4 relationship between with other areas



Computer Graphics

Pattern recognition

Computer vision

They are based most same theories such as:

AI(Artificial Intelligent)

NN(Neural Network)

GA(Genetic Algorithm)

FL(Fuzzy Lagical)

1.2 The origins and development of DIP

- newspaper industry:
  - **Time:** 1920s
  - **System :** Bartlane cable picture transmission system
  - **Function:** sent picture between Landon and New York
  - **Image quality:** 5~15 levels, halftone technique

1.2 The origins and development of DIP

- Space application:
  - **Time:** 1960s
  - **System :** *Ranger 7*
  - **Function:** enhance and restore images from moon, Mars, and others

- Medical application:
  - Time: 1970s
  - System: CT, X-ray

1.2 The origins and development of DIP

- Development history :
  - 50 years: born
  - 1964 good results for images from space
  - 70 years developing
  - 80 years established theory
  - 90 years applications on wide areas

- The history in our university :
  - 80-83 go abroad to study image processing.
  - 83 established image processing lab.
  - 85 bought IIS image system
  - 88 Information Processing center

1.3 Examples of Fields that Use DIP

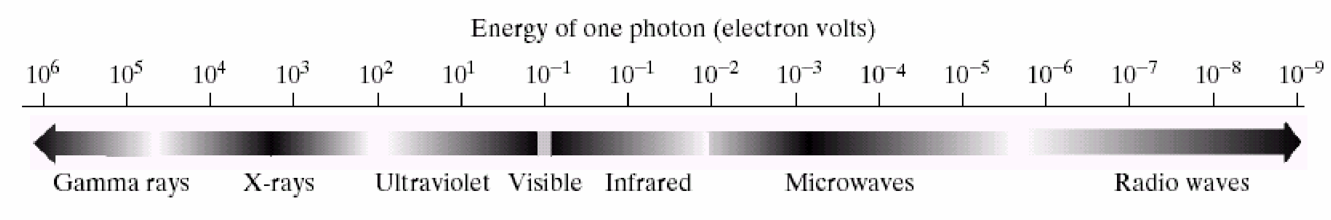


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

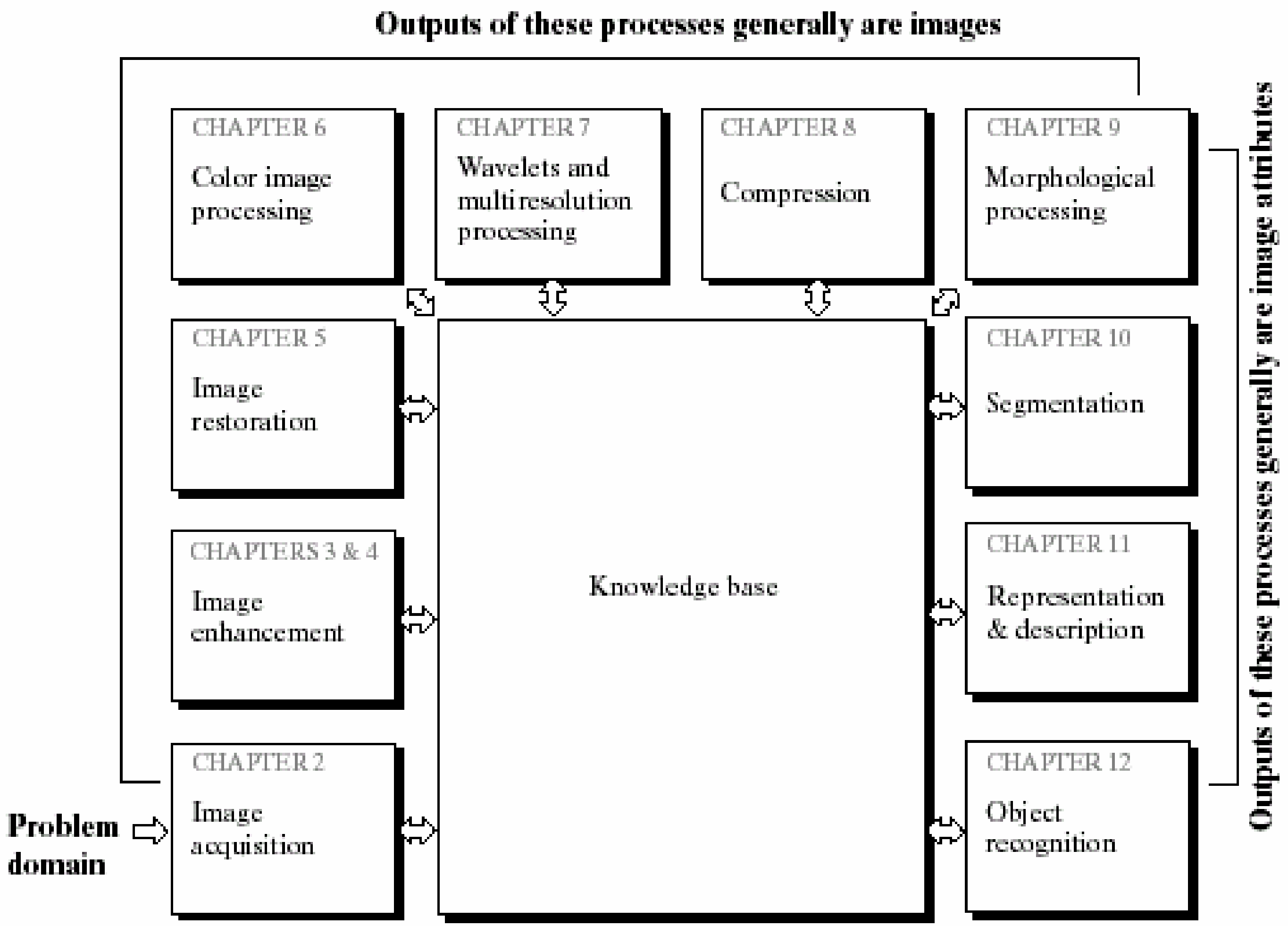
- Gamma-Ray Imaging
- X-Ray Imaging
- Imaging in the Ultraviolet Band
- Imaging in the Visible and Infrared Bands
- Imaging in the Microwave Band
- Imaging in the Radio Band
- Example in Which Other Modalities Are Used

Areas where DIP is used

- Multimedia technology and communication
  image compression
  VCD,DVD,HDTV,Video telephone
- Medicine
- Industrial
- Military
- Commercial
- Natural resource

1.4 Fundamental Step in DIP

FIGURE 1.23 Fundamental steps in digital image processing.



1.4 Fundamental Step in DIP

Research directions:

image compression

wavelet transform

digital watermarking

image reconstruction

indexing by image content

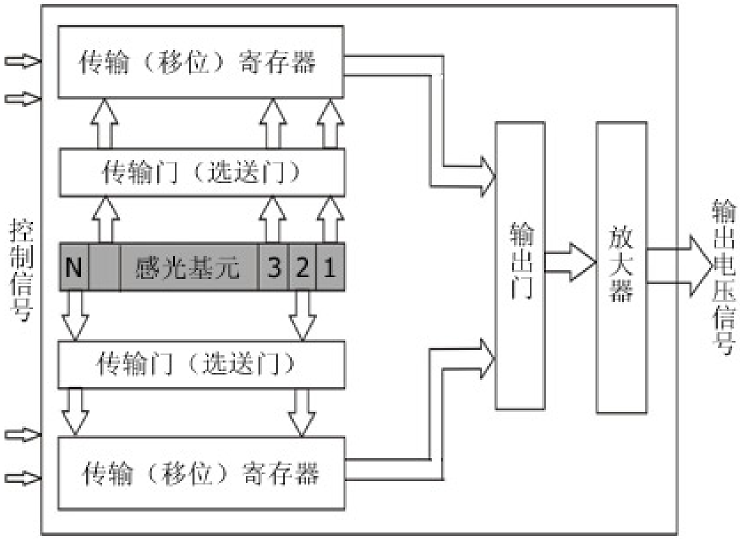
image recognition

image understanding

1.5 Components of an Image Processing System

- Sensing and digitizer
- Specialized image processing hardware
- Computer
- Mass storage
- Image displays
- Hardcopy
- Networking
- Software

1.5.1 Sensing and digitizer



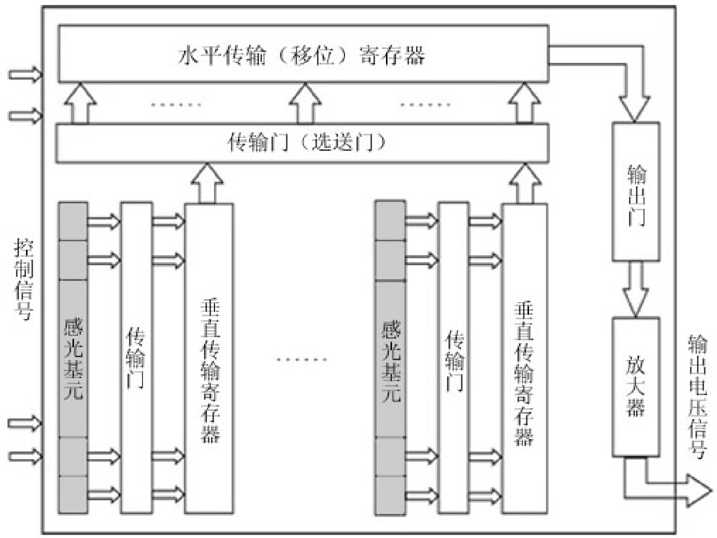
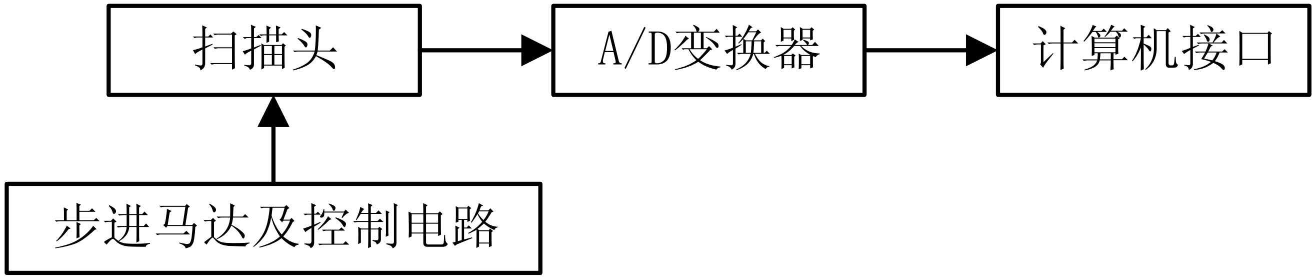


Diagram of Linear CCD

Diagram of Area CCD

1.5.1 Sensing and digitizer

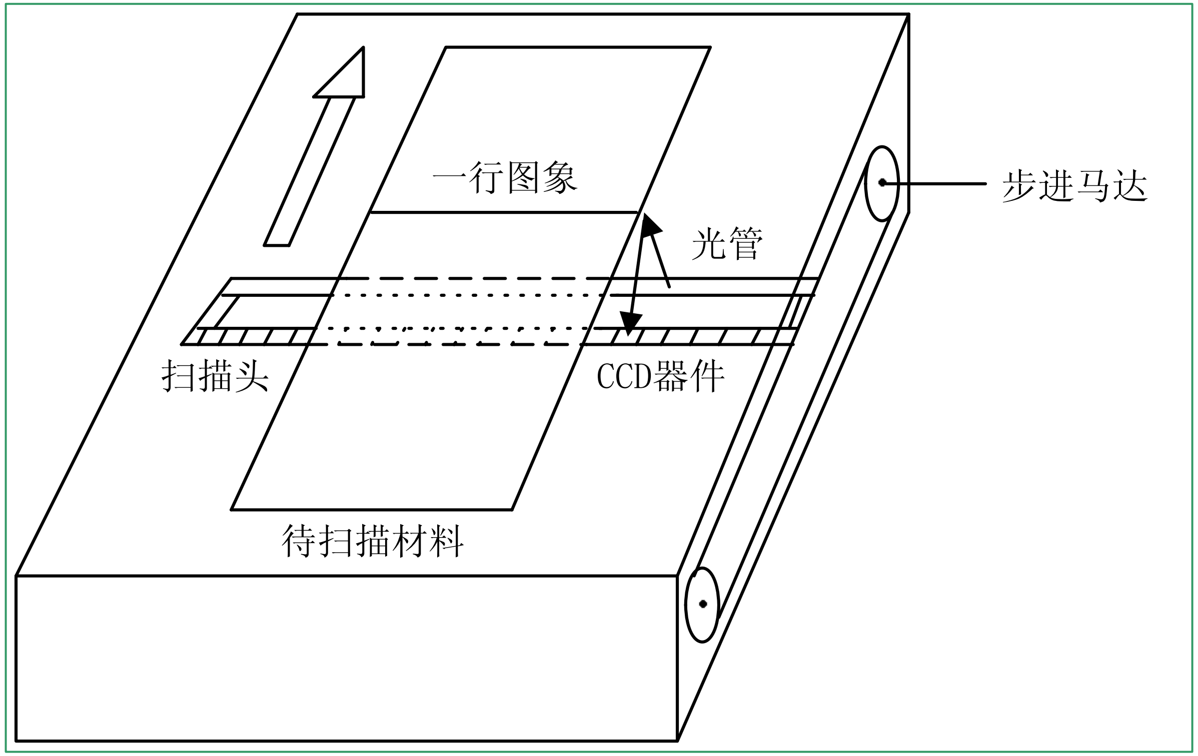
Scanner



扫描仪基本组成框图

1.5.1 Sensing and digitizer

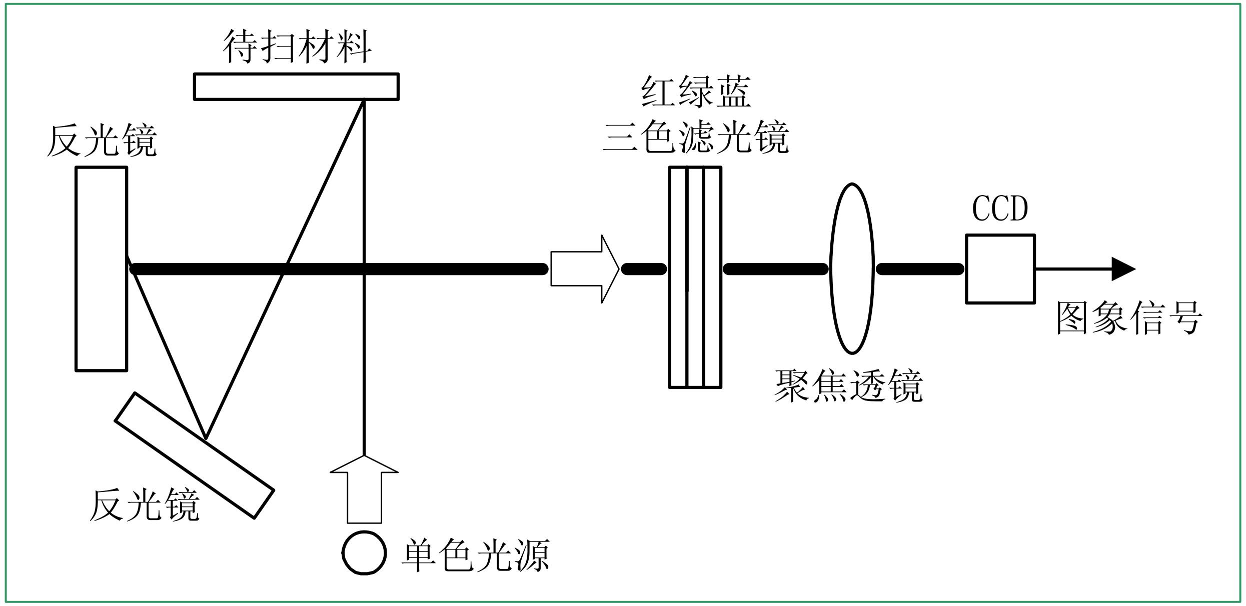
Scanner



扫描仪工作原理示意图

1.5.1 Sensing and digitizer

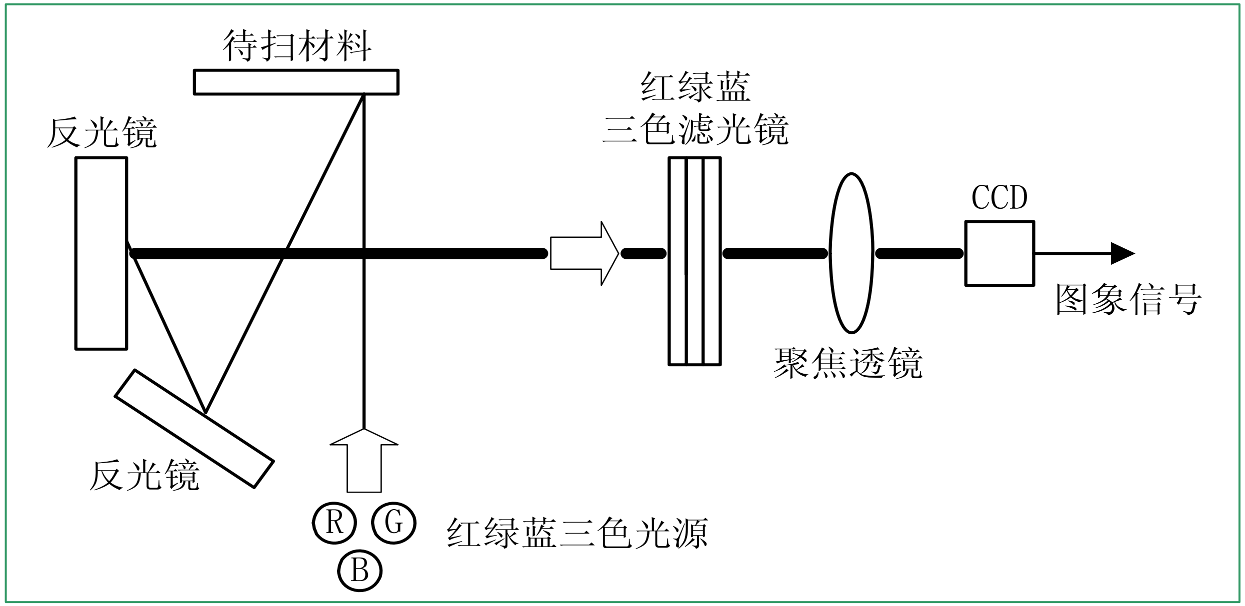
Scanner



彩色图象扫描方式之一

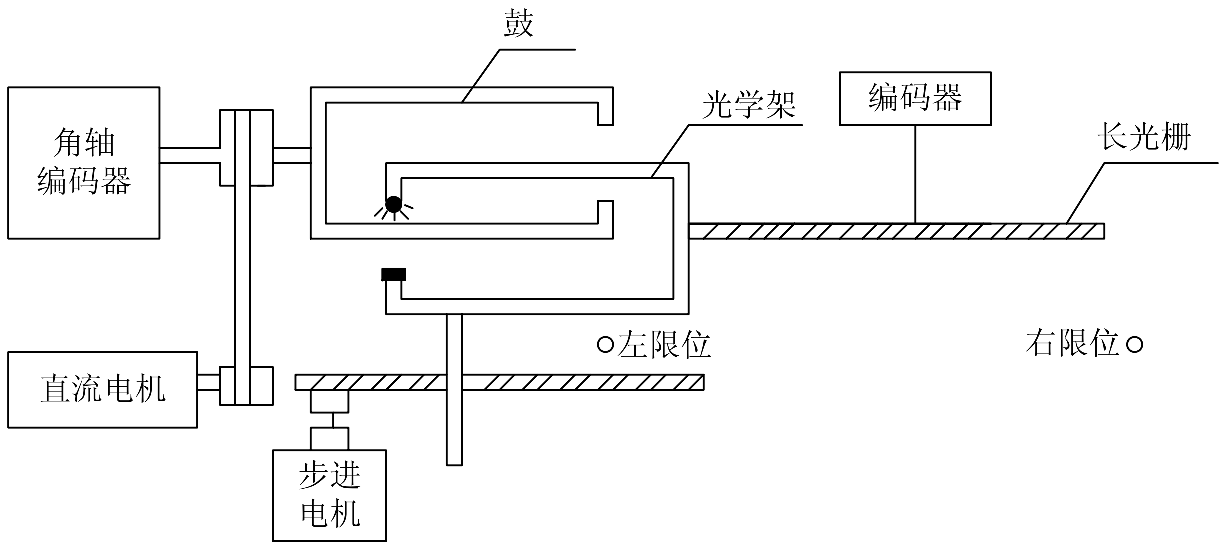
1.5.1 Sensing and digitizer

Scanner



彩色图象扫描方式之二

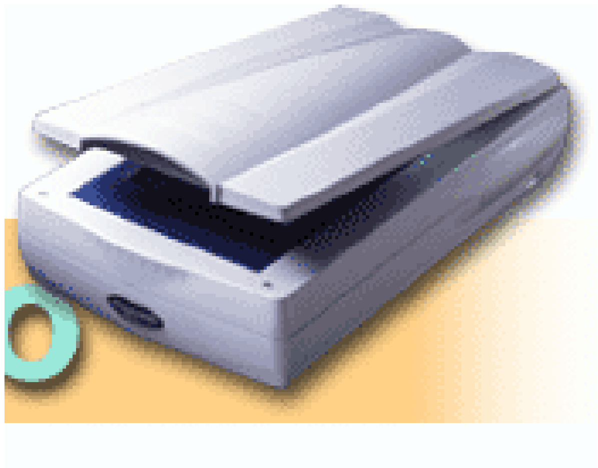
1.5.1 Sensing and digitizer



扫描记录仪的结构示意图

1.5.1 Sensing and digitizer

Scanner



Mustek PowerPro

Scanner Type	flat color scanner
Scan Mode:	
Color Mode	24 bits/pixel
Gray Mode	8 bits/pixel
Text/Line Art	1 bits/pixel
Scan Area	216 x 356 mm 8.5" x 14"
Resolution	600dpi x 1200dpi
Scanning Data Buffer	1 MB
Brightness Control	255 adjustable steps
Contrast Control	255 adjustable steps
Light Source	Cold Cathode lamp

1.5.1 Sensing and digitizer

Digital Camera

Olympus u300

Price (street)	US$400
Max resolution	2048 x 1536
Low resolution	1600 x 1200, 1280 x 960, 1024 x 768, 640 x 480
Image ratio w:h	4:3
Effective pixels	3.14 million
Sensor size	1/2.5"
Sensor type	CCD
Colour filter array	G-R-G-B
Max shutter	1/1000 sec

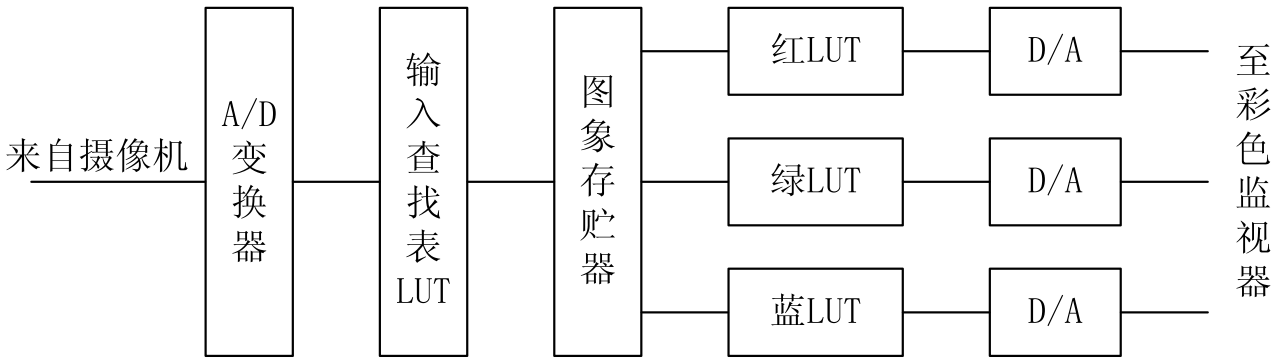
1.5.2 Specialized Image Processing Hardware

- PCVION中的帧存板
- 实时图象处理板MVP-AT
- 实时图象处理卡Cobra/C6

1.5.2 Specialized Image Processing Hardware

PCVION中的帧存板

Look Up Table

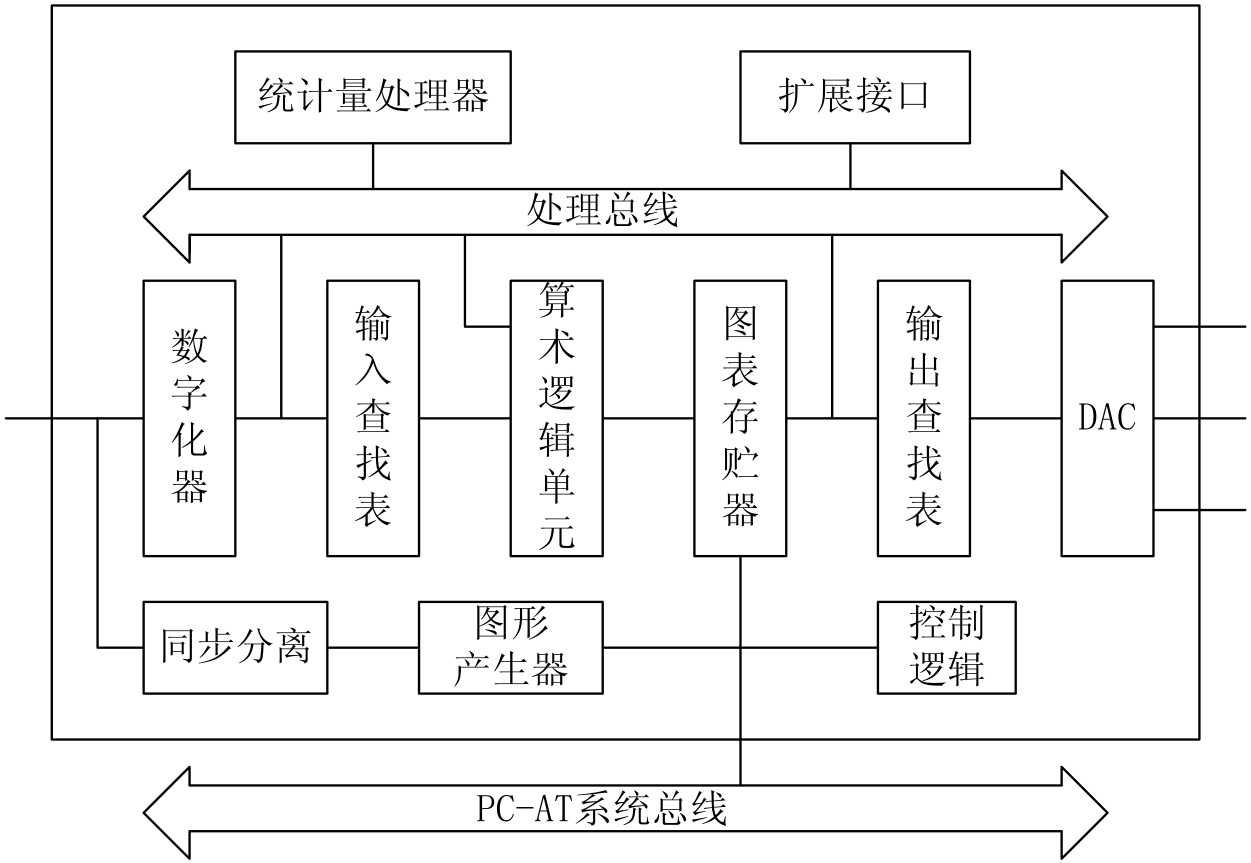


帧存板的组成

1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64

1.5.2 Specialized Image Processing Hardware

实时图象处理板VP-AT



MVP-AT板方框图

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

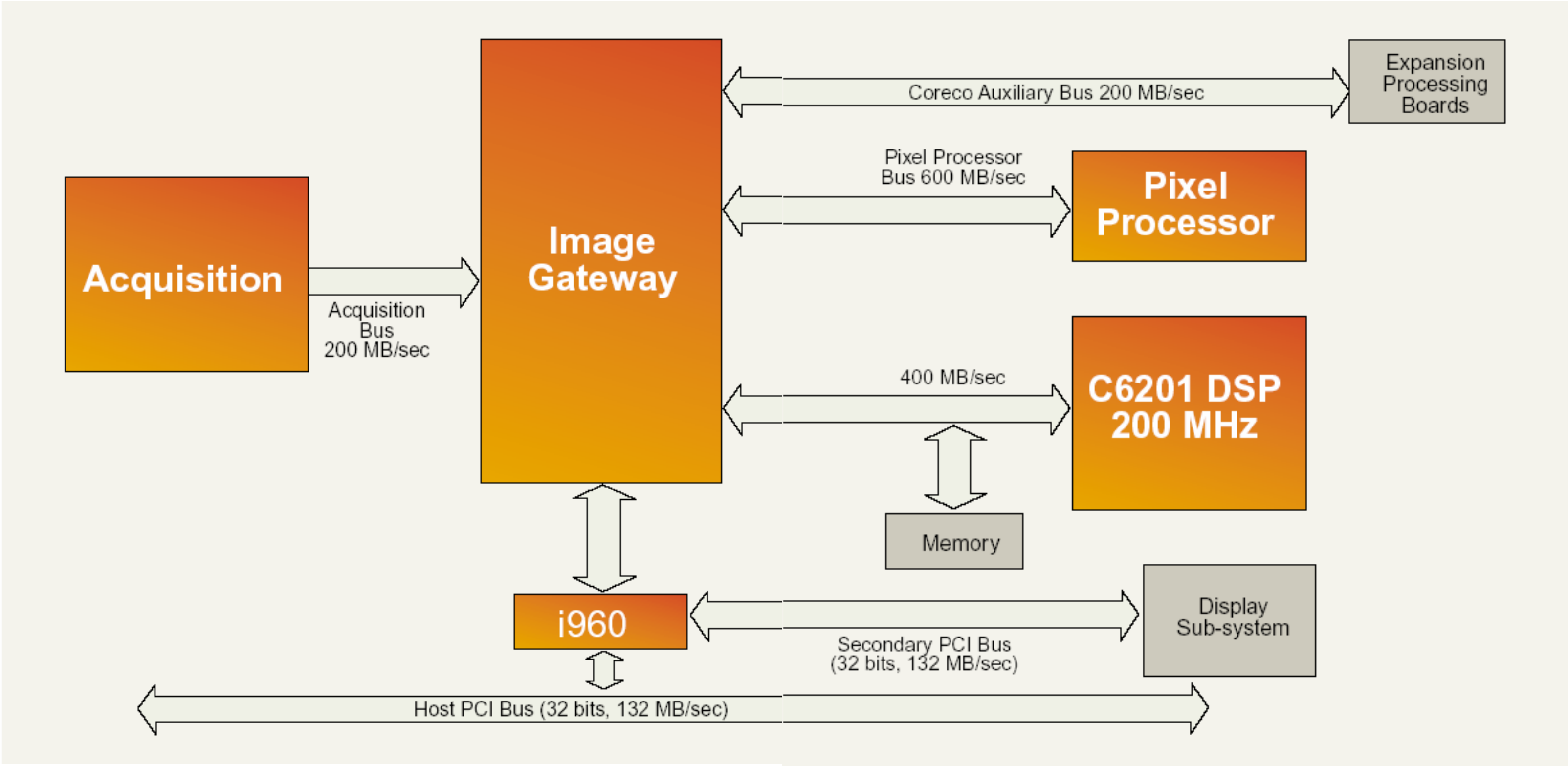


Diagram of the Cobra/C6 architecture

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

Acquisition

- RS-170, CCIR, NTSC, PAL or Y/C acquisition
- Three Analog camera inputs
- Digital filtering improves color reproduction
- Pixel jitter less than 2 nSecs
- Video fidelity > 45dB
- RS-422 interface for digital acquisition to 100 Mbytes/sec, up to 12 bits/pixel.
- Expansion slot for additional acquisition module at rates up to 200 Mbytes/sec.

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

Image Gateway

- an extremely powerful transfer controller designed to deliver high I/O throughput to the Cobra/C6.
- Featuring seven I/O ports with a combined bandwidth of 1520 Mbytes/sec,
- can simultaneously interconnect any five ports for a maximum combined transfer rate of 720 Mbytes/sec.

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

Pixel Processing

- Image processing at rates up to 200 MB/sec.
- Neighborhood or point-to-point acceleration
- Real-time multiple imaging functions, i.e. unsharp, flat-field correction
- Reprogrammable imaging functions
- Image pre-processing or co-processing for the Cobra/C6
- Field-installed

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

Pixel Processing

512 x 512, 8-bit images.

Category	**Function**	**Speed**
Point-to-Point	adding two images subtracting two images averaging unsharp mask flat-field correction	1.3 mSecs 1.3 mSecs 1.3 mSecs 1.3 mSecs 1.3 mSecs 1.3 mSecs
Neighborhood operations	convolution (3x3) convolution (5x5) convolution (7x7) convolution (9x9) convolution (13x13) median Filter laplacian/sobel	1.3 mSecs 2.6 mSecs 2.6 mSecs 5.2 mSecs 10.5 mSecs 2.6 mSecs 2.6 mSecs
Statistical operations	histogram histogram equalization	1.3 mSecs 2.6 mSecs
Morphology	erosion	1.0 mSecs

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

TMS320C6201 DSP

- 200 MHz Operating Frequency
- 64k of high speed on-die cache memory
- 800 Mbytes/sec I/O bandwidth
- 1.6 Gigabyte/sec on-chip transfer rates
- Executes up to eight instructions simultaneously
- Integrated on-chip host port for control
- Six 32/40 bit ALU's
- Two 16 bit single cycle multipliers
- VLIW simplifies parallel programming

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

i960

- as an interface between the Cobra/C6 and the PCI bus.
- manage the flow of data between the host computer and Image Gateway.
- controls the acquisition sub-system, Pixel Processor, Multi-Processing Bus and secondary PCI bus, leaving 100% of the processing power of the TMS320C6201 DSP available for image processing and analysis.

1.5.2 Specialized Image Processing Hardware

实时图象处理卡Cobra/C6

Key Features:

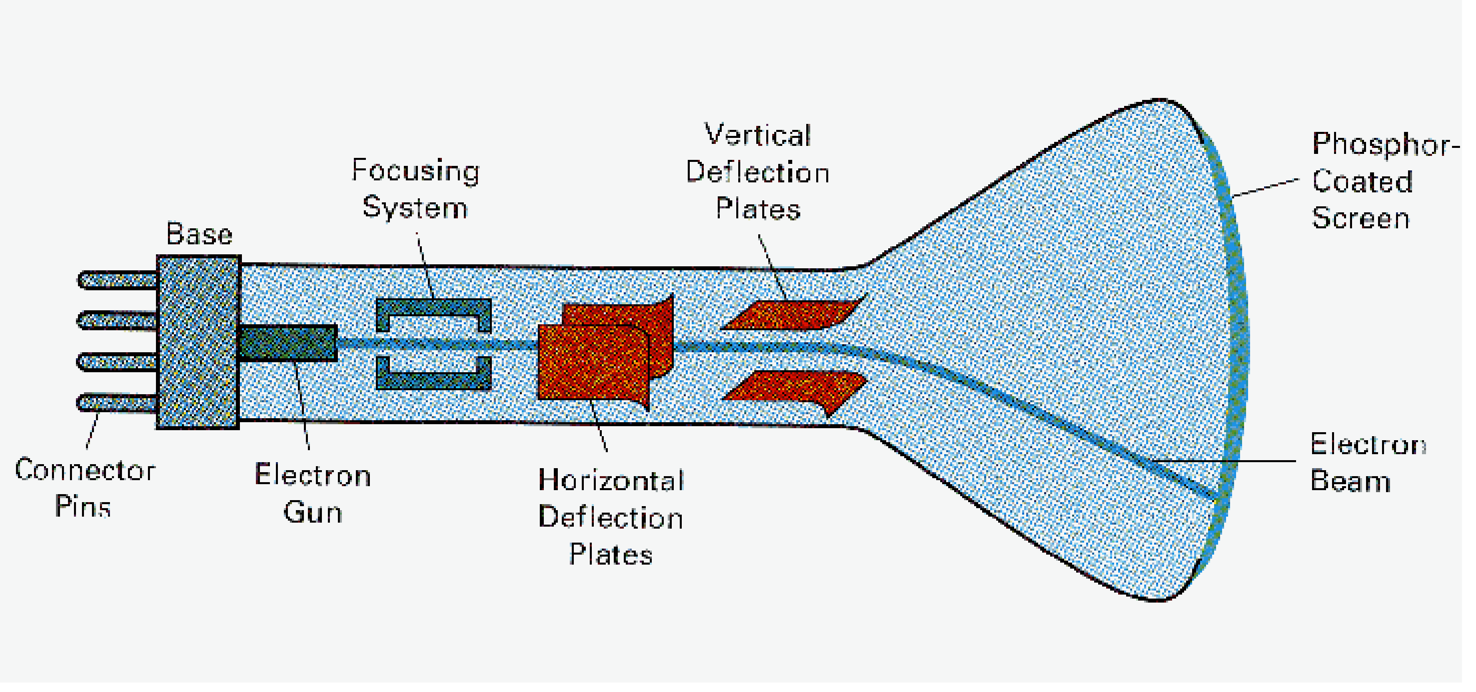
- •200 MHz Operating Frequency
- •64k of high speed on-die cache memory
- •800 Mbytes/sec I/O bandwidth
- •1.6 Gigabyte/sec on-chip transfer rates
- •Executes up to eight instructions simultaneously
- •Integrated on-chip host port for control
- •Six 32/40 bit ALU's
- •Two 16 bit single cycle multipliers
- •VLIW simplifies parallel programming

1.5.3 Image Displays

- CRT(Cathode-Ray Tube)
- Liquid-crystal displays
- Plasma panels

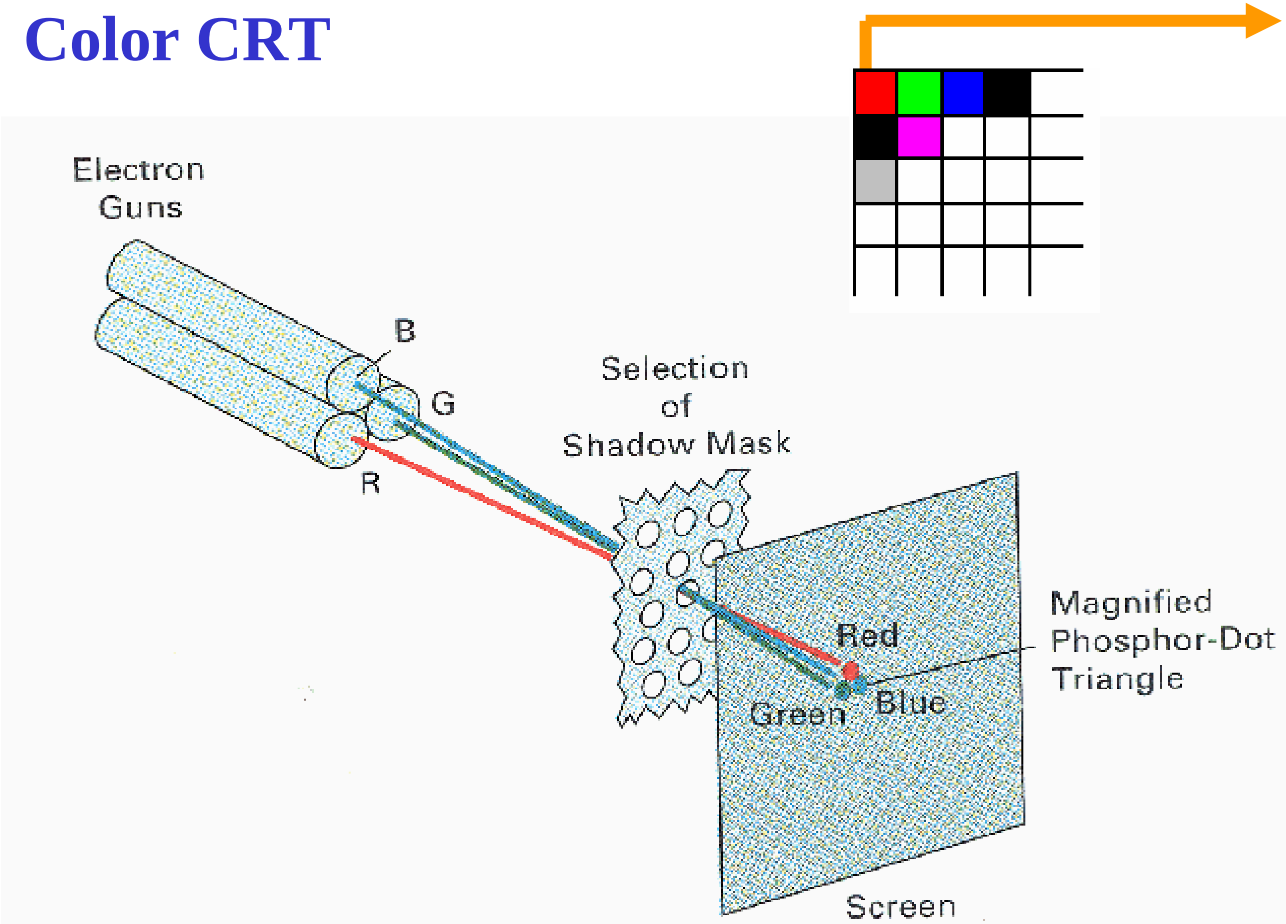
1.5.3 Image Displays

CRT



1.5.3 Image Displays

Color CRT

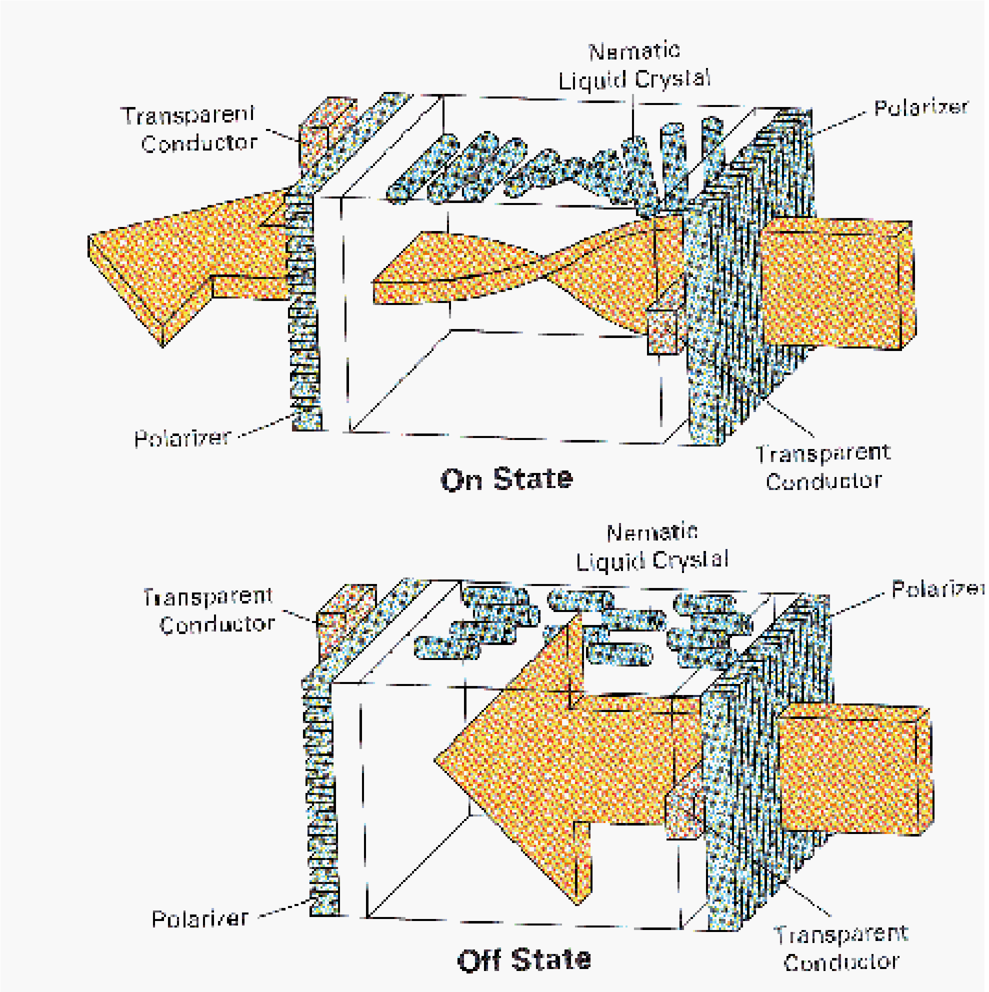


Frame Buffer

11111111
00000000
00000000
00000000
11111111
00000000
00000000
00000000
11111111
00000000
00000000
00000000
. . .

1.5.3 Image Displays

Liquid Crystal Display

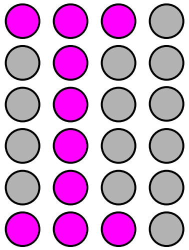


1.5.4 Hard Copy

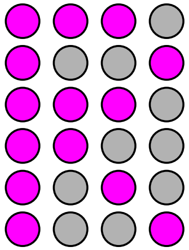
- Dot-matrix Printer
- Laser Printer
- Ink-jet Printer
- Thermal

1.5.4 Hard Copy

Dot-matrix Printer

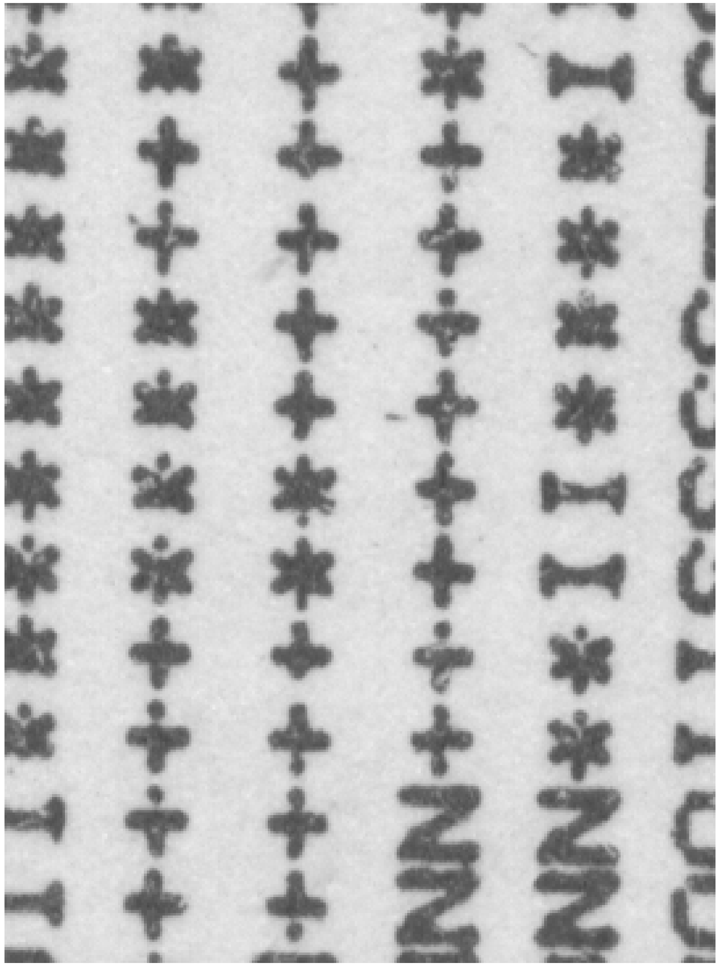


Dot Matrix of “I”



Dot Matrix of “R”

1.5.4 Hard Copy



Dither Technique

1.5.4 Hard Copy

Ink-jet Printer

1.5.5 Software

- Photoshop
- PhotoshImpact
- Corel Custom Photo
- MS PhotoDraw
- Fireworks
- ACDSee

Chapter2 Fundamentals of Image and Vision

- 2.1 General Introduction and Classification
- 2.2 Human Eye and Brightness Vision
- 2.3 Color Vision
- 2.4 Photometry and Imaging Model
- 2.5 Imaging Transformation
- 2.6 Sampling and Quantization
- 2.7 Relationships between Pixels
- 2.8 Arithmetic and logic Operations

2.1 General Introduction and Classification

- **Fundament of Vision** :eye and it's perception, color model
- **Fundament of Imaging**: imaging model, projection transformation, sampling and quantization
- **Fundament of image**: relationship between pixels, arithmetic and logic operations, coordinate transformation.

2.2 Human Eye and Brightness Vision

2.2.1 Structure of the Human Eye

- Shape: nearly a sphere, diameter=20mm
- Components: cornea(角膜), sclera(巩膜), choroid(脉络膜), retina(视网膜), lens, fovea(中央凹), blind spot(盲点), nerve (神经)and sheath(鞘).

2.2 Human Eye and Brightness Vision

2.2.2 Comparison of Cones and Rods

	cones	rods
number	6~7 million	75~175 million
Sensitivity	color	shape
vision	Photopic (Bright-light)	Scotopic(dim-light)

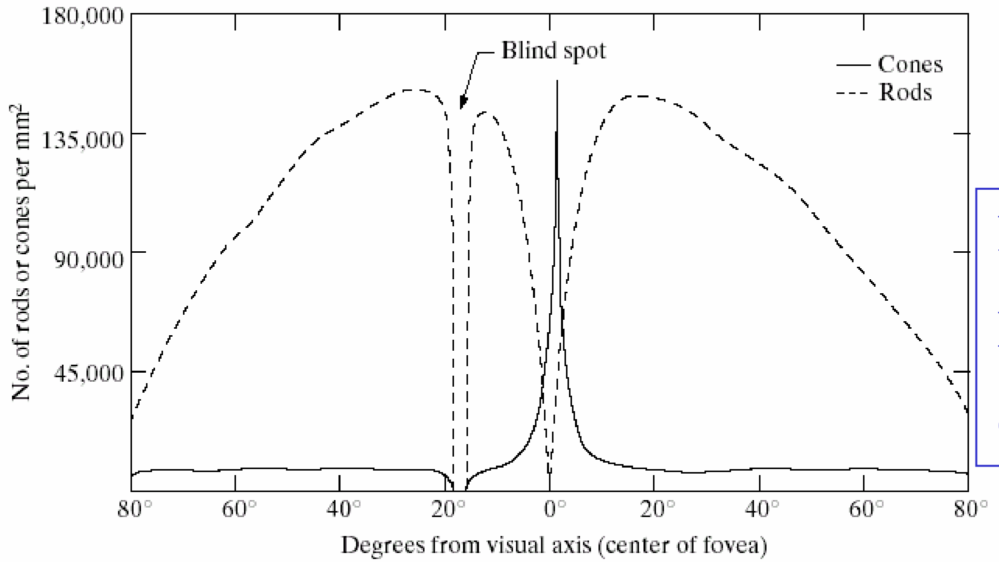


FIGURE 2.2 Distribution of rods and cones in the retina.

Fovea area: 1.5mm*1.5mm

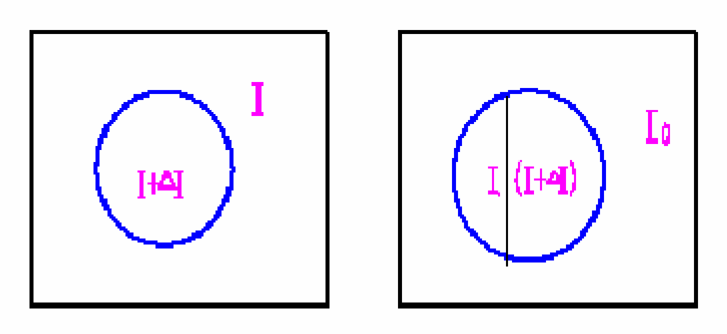
Density of cones: 15000/mm^2

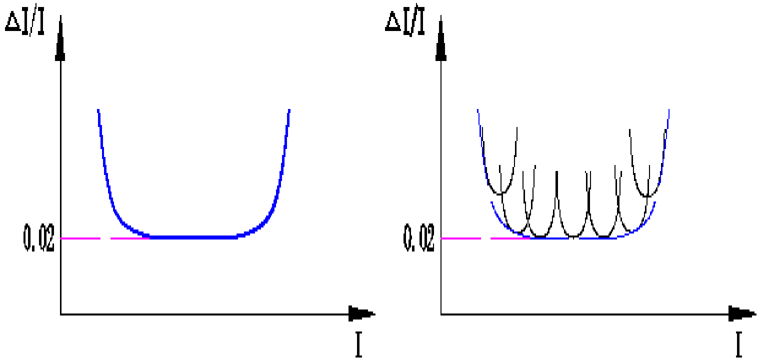
Cones in fovea: 337 000

2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

Phenomena 1: Weber Ratio



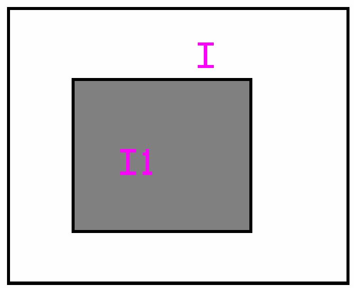


2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

Explanation: $$d(\ln I) = dI / I$$

Definitions of Contrast:



$$C_p = (I_1 - I) / I = \Delta I / I$$

$$C_I = I_{\max} / I_{\min}$$

Weber Ratio=0.02=2%

2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

Phenomena 2:
Simultaneous
contrast

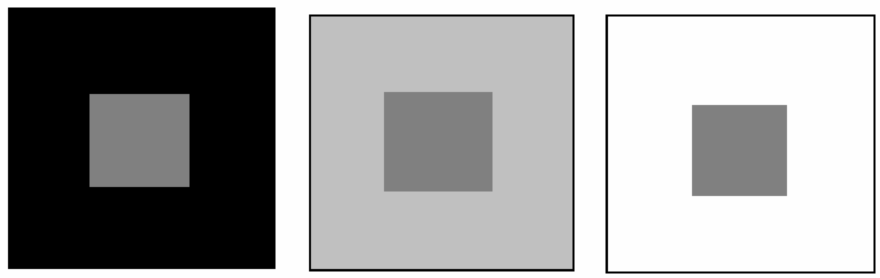


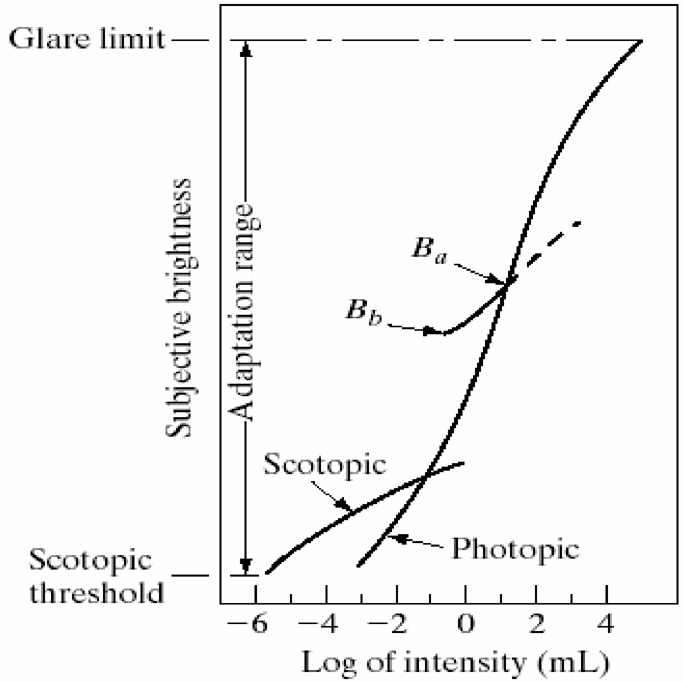
Figure 2.8

- Although all the center squares have the same intensity,they appear to the eye to become darker as the background gets lighter.

2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

FIGURE 2.4 Range of subjective brightness sensations showing a particular adaptation level.



Phenomena 3: Subjective Brightness

2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

Explanation: increment of brightness is defined as:

$$\Delta S = k^{'} \Delta B / B$$

after integral, S is:

$$S = k^{'} \ln B + k_0 = k \lg B + k_0$$

Where k’ and k_0 are constant, and k=k’=ln10

2.2 Human Eye and Brightness Vision

2.2.3 Log characteristic

Conclusion:

- Log transformation is usually used in preprocessing step of DIP system , in order to improve the visual results.
- The subjective brightness is relatived
- The discriminability of human eye is limited

2.2 Human Eye and Brightness Vision

2.2.4 Frequency characteristic

Visual model: Modulation Transfer Function

$$\boxed{\begin{array}{c}\text{Input image}\\ f(x,y)\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{Optical System}\\ H(u,v)\end{array}} \longrightarrow \boxed{\begin{array}{c}\text{Output image}\\ g(x,y)\end{array}}$$

$$G(u,v) = H(u,v)F(u,v)$$

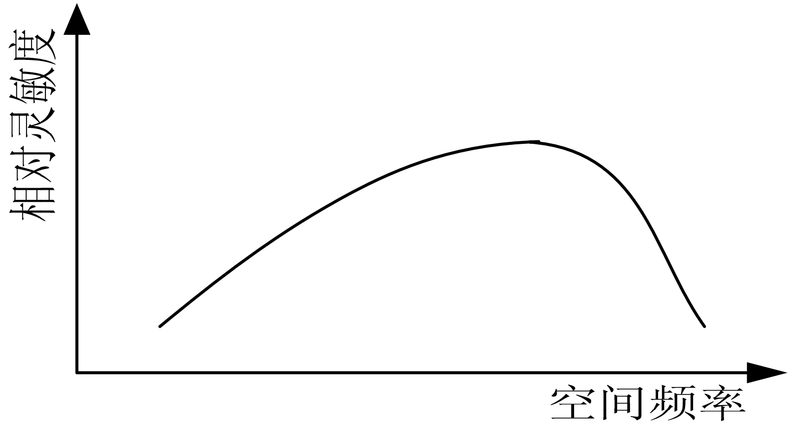
$$H(u,v) = G(u,v)/F(u,v)$$

$$|H(u,v)| = |G(u,v)|/|F(u,v)|$$

2.2 Human Eye and Brightness Vision

2.2.4 Frequency characteristic

MTF of human vision:



（b）

$$|H(u,v)|^2 = (1 + 0.05D^2)\exp[-(D/50)^2]$$

2.2 Human Eye and Brightness Vision

2.2.5 Optical illusions

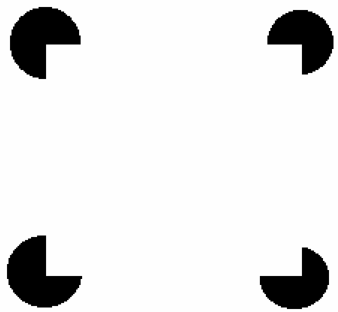


Figure 2.9-1

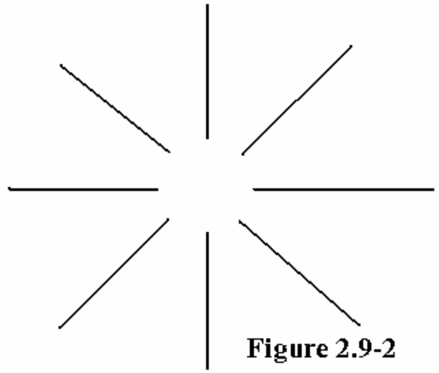
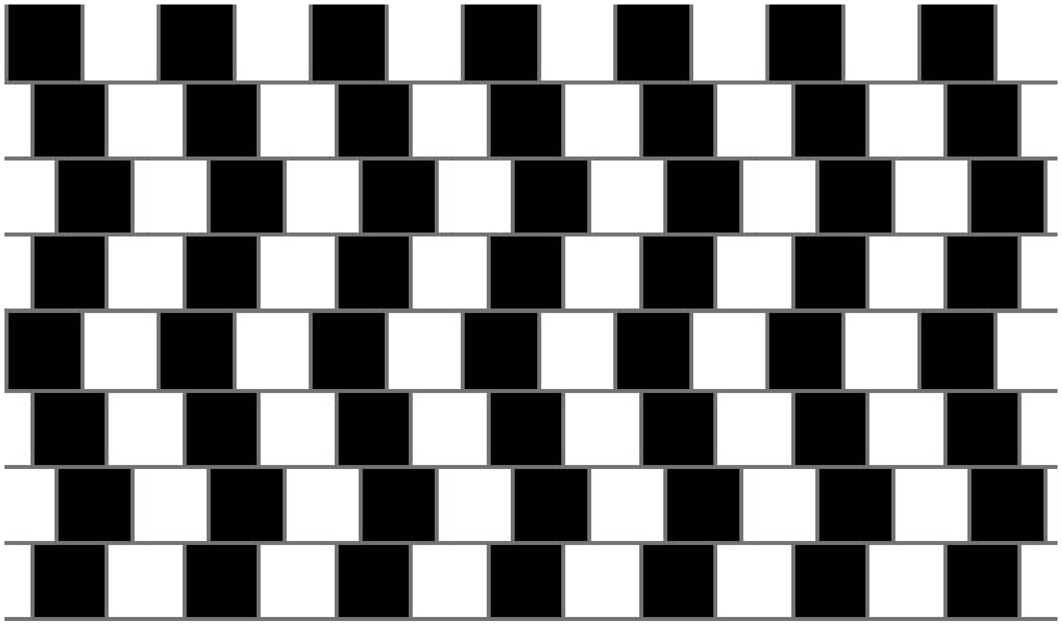


Figure 2.9-2



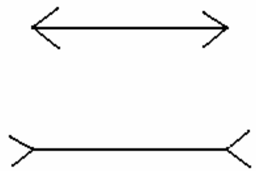


Figure 2.9-3

2.3 Color Vision

- **Color Fundament**: wavelengths of visible band, primary colors, second colors, CIE standards
- **Color Model:** RGB, CMY, HSI,

2.2 Human Eye and Brightness Vision

Experimental Curves of light absorption by the red, green and blue cones in the eye

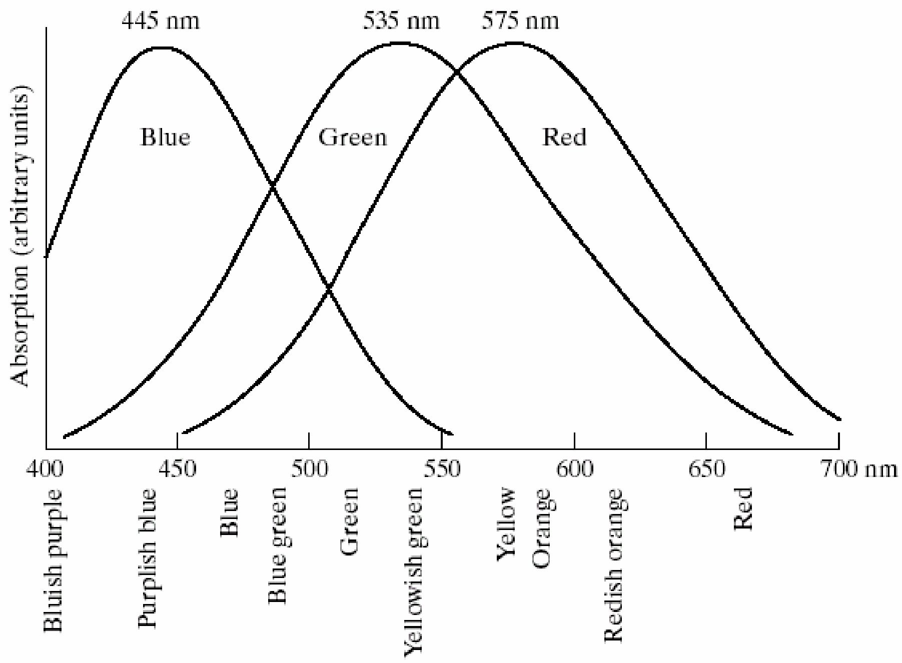


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

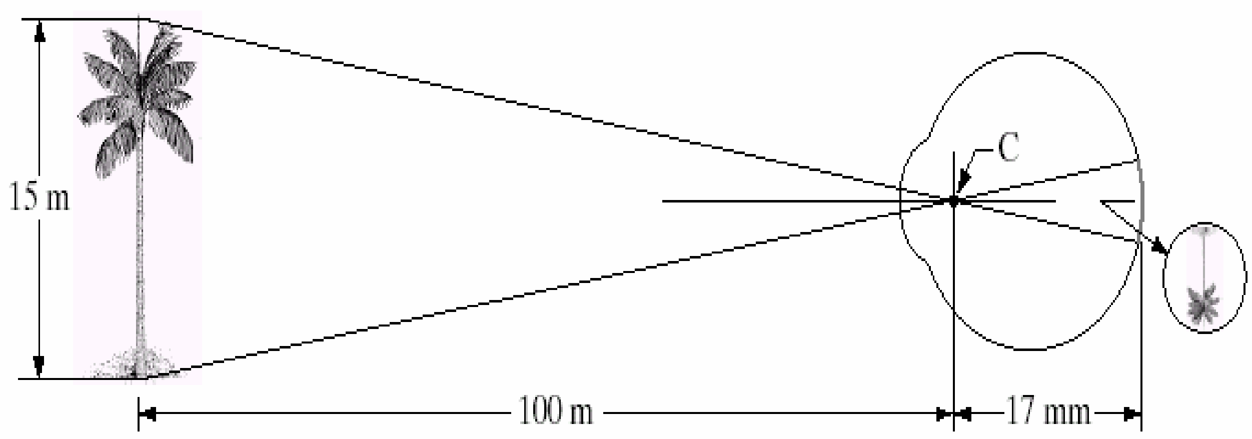
Cones	Percent
Red	65%
Green	33%
Blue	2%

But the blue cones is the most sensitive

2.2 Human Eye and Brightness Vision

2.2.6 Image Formation in the Eye

FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.



If h is the height of that object in the retinal image,

$$\frac{15}{100} = \frac{h}{17} \Rightarrow h = 2.55mm$$

2.3 Color Vision

2.3.1 Color Fundament: CIE Standard

$$C = X[R] + Y[G] + Z[B]$$

Eg:

Red=1[R]+0[G]+0[B]
Yellow=1[R]+1[G]+0[B]
Cyan=0[R]+1[G]+1[B]
Magenta=1[R]+0[G]+1[B]
Black=0[R]+0[G]+0[B]
White=1[R]+1[G]+1[B]

Hue 色调

Saturation饱和度

Intensity辉度

Chroma 色度

2.3 Color Vision

2.3.1 Color Fundament: CIE Standard

If the tristimulus values are denoted, X,Y and Z, then the trichromatic coefficients are defined as:

$$x = \frac{X}{X+Y+Z} \qquad (2.3.1)$$

$$y = \frac{Y}{X+Y+Z} \qquad (2.3.2)$$

$$z = \frac{Z}{X+Y+Z} \qquad (2.3.3)$$

$$x + y + z = 1 \qquad (2.3.4)$$

2.3 Color Vision

2.3.1 Color Fundament: CIE chromaticity diagram

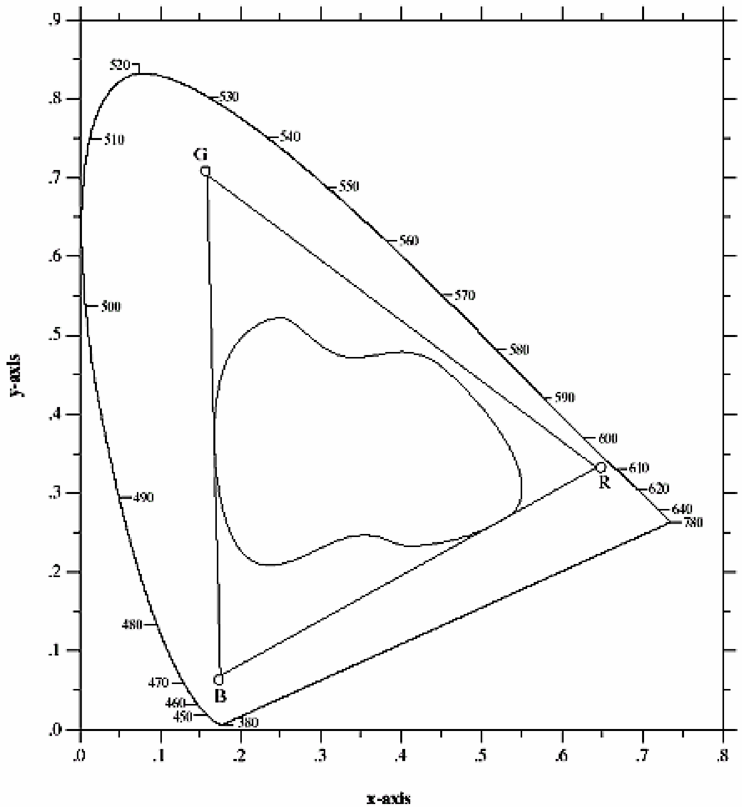
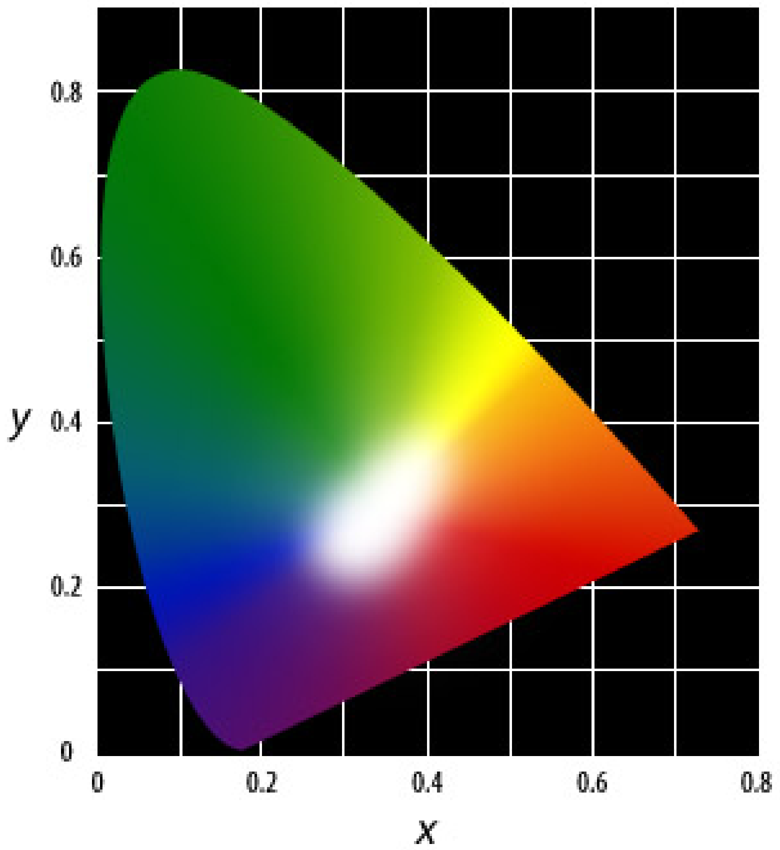
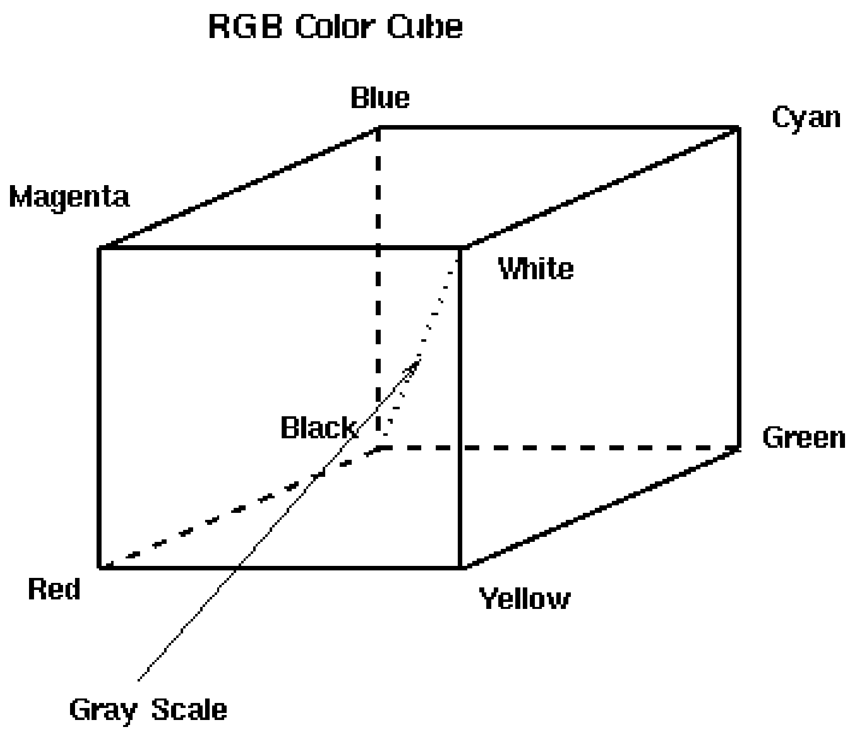


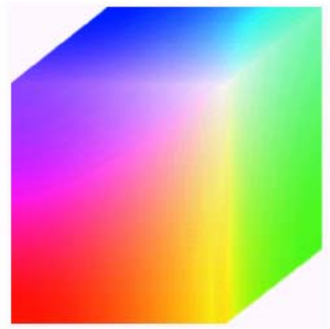
FIGURE 6.6 Typical color gamut of color monitors (triangle) and color printing devices (irregular region).



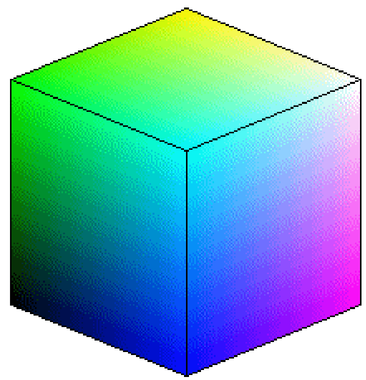
2.3 Color Vision

2.3.2 Color Model: RGB model





6.8 RGB 24-bit color cube.



2.3 Color Vision

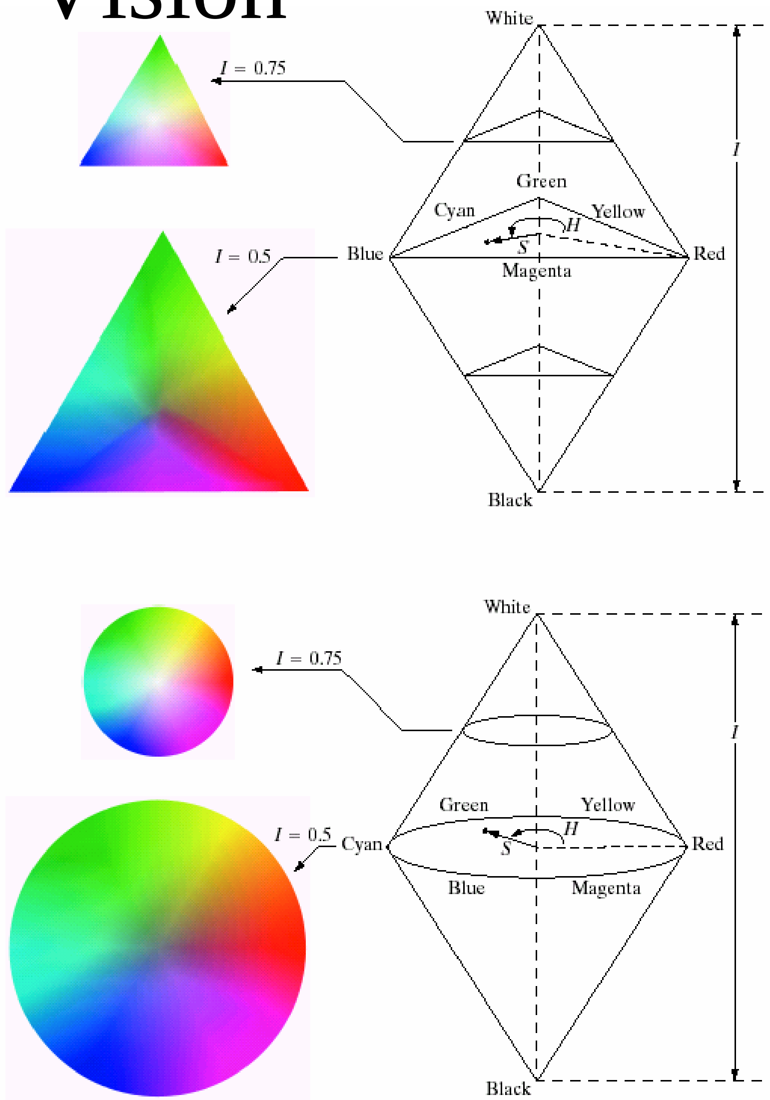
2.3.2 Color Model: HSI model

- **Hue:** associated with the dominant wavelength in a mixture of light waves
- **Saturation:** refer to the relative purity or the amount of white light mixed with hue.
- **Intensity:** brightness

2.3 Color Vision

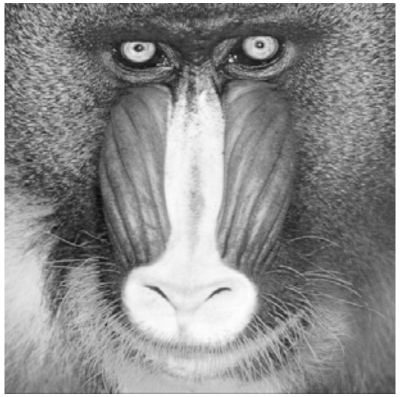
2.3.2 Color Model: HSI model

Color	H	S	I
Red	0	1	0.5
Green	120	1	0.5
Blue	240	1	0.5
White	0	0	1
Black	0	0	0

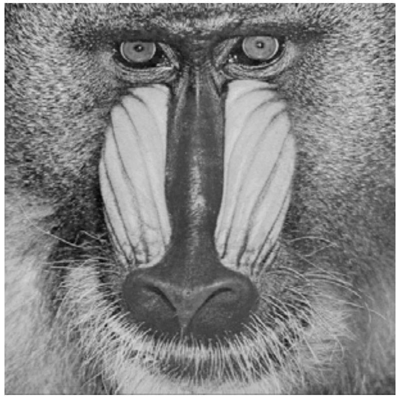


2.3 Color Vision

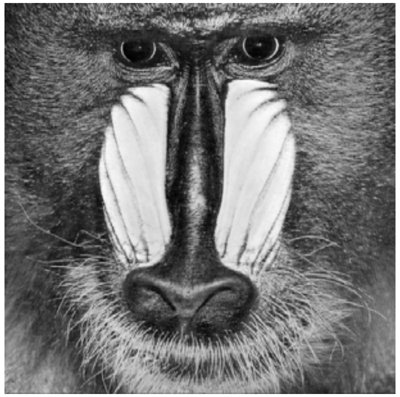
Please guess the color of the monkey's nose



R

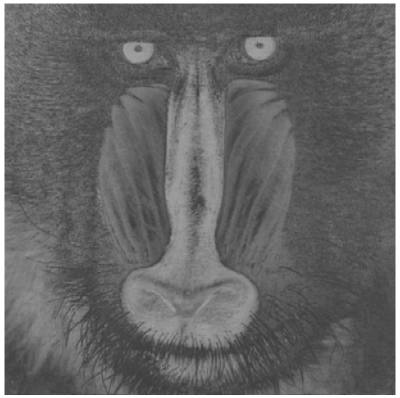


G

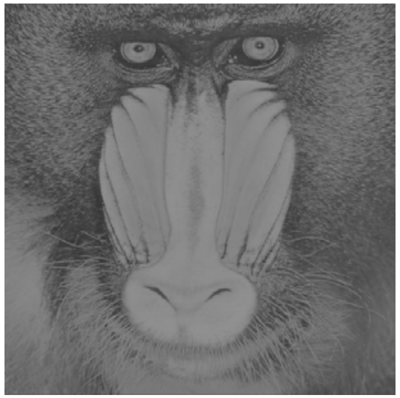


B

H



S



I

2.3 Color Vision

The answer is:

2.3 Color Vision

2.3.2 Color Model: converting colors from RGB to HSI

$$I = \frac{1}{3}(R + G + B) \qquad (2.3.5)$$

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)] \qquad (2.3.6)$$

$$\theta = \arccos\left\{\frac{[(R-G)+(R-B)]/2}{[(R-G)^2 + (R-B)(G-B)]^{1/2}}\right\}$$

$$H = \begin{cases} \theta & B \le G \\ 360 - \theta & \text{others} \end{cases} \qquad (2.3.7)$$

2.3 Color Vision

2.3.2 Color Model: converting colors from HSI to RGB

RG Sector: $0^o \le H < 120^o$

$$B = I(1-S) \qquad (2.3.8)$$

$$R = I\left[1+\frac{S\cos H}{\cos(60^\circ - H)}\right] \qquad (2.3.9)$$

$$G = 3I-(B+R) \qquad (2.3.10)$$

2.3 Color Vision

2.3.2 Color Model: converting colors from HSI to RGB

GB Sector: $120^o \le H < 240^o$

$$R = I(1 - S) \qquad (2.3.11)$$

$$G = I\left[1 + \frac{S\cos\left(H - 120^\circ\right)}{\cos(180^\circ - H)}\right] \qquad (2.3.12)$$

$$B = 3I - (R + G) \qquad (2.3.13)$$

2.3 Color Vision

2.3.2 Color Model: converting colors from HSI to RGB

BR Sector: $240^o \le H < 360^o$

$$G = I(1 - S) \qquad (2.3.14)$$

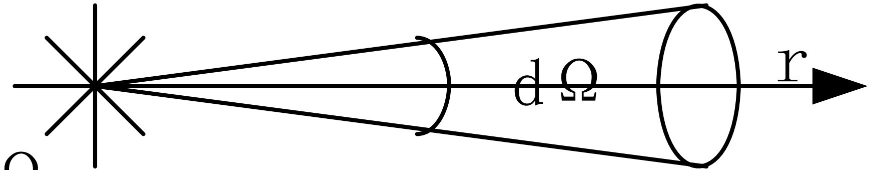
$$B = I\left[1 + \frac{S\cos(H - 240^\circ)}{\cos(300^\circ - H)}\right] \qquad (2.3.15)$$

$$R = 3I - (G + B) \qquad (2.3.16)$$

2.4 Photometry and Imaging Model

2.4.1 Point Source

点光源

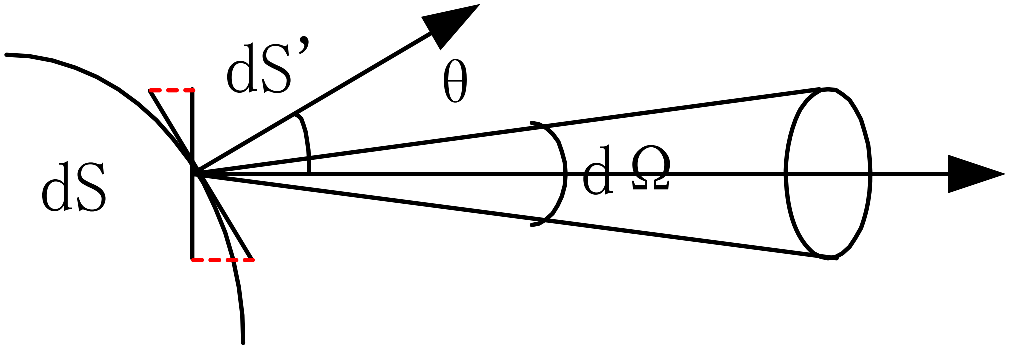


以r为轴取1个立体角元dΩ，
设dΩ中的光通量为Φ，则在沿r方向的发光强度为 $I = \frac{d\Phi}{d\Omega}$ cd 1 cd = lm/sr

2.4 Photometry and Imaging Model

2.4.2 Distributed light Source

扩展光源



亮度： $B \equiv \frac{dI}{dS'} \equiv \frac{dI}{dS\cos\theta} \equiv \frac{d\Phi}{d\Omega dS\cos\theta}$ cd/m^2

照度： $E = \frac{d\Phi}{dS}$

2.4 Photometry and Imaging Model

2.4.3 Imaging Model

We denote images by 2-D functions of the form .

$$0 < f(x, y) < \infty$$

f(x,y) can be characterized by two components called the illumination and reflectance components which are denoted by i(x,y) and r(x,y).

$$f(x, y) = i(x, y)r(x, y)$$

where

$$0 < i(x, y) < \infty$$

$$0 < r(x, y) < 1$$

2.5 Imaging Transformation

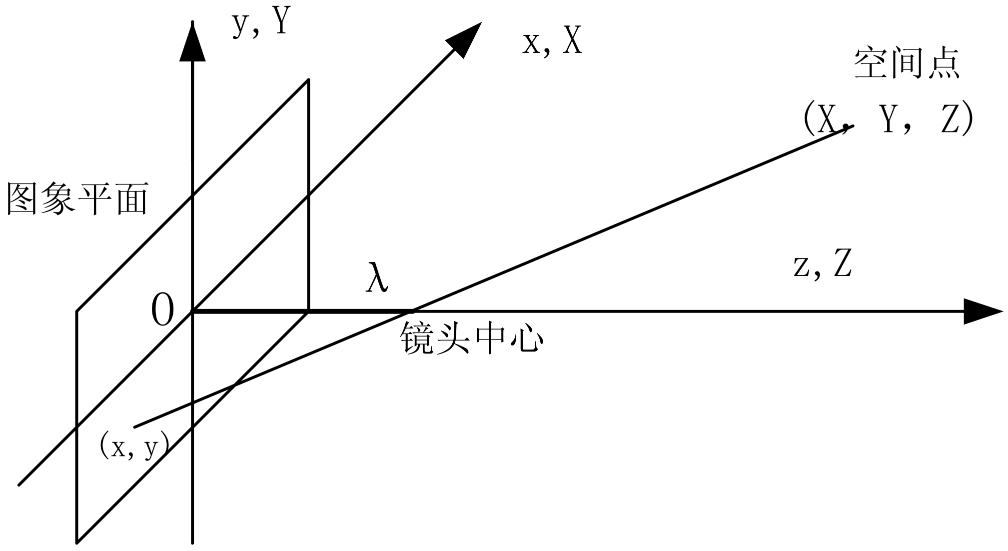


图2.5.1　投影成象示意图

2.5 Imaging Transformation

$$\frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda - Z} \quad \Rightarrow \quad x = \frac{\lambda X}{\lambda - Z}$$

$$\frac{y}{\lambda} = -\frac{Y}{Z-\lambda} = \frac{Y}{\lambda - Z} \quad \Rightarrow \quad y = \frac{\lambda Y}{\lambda - Z}$$

2.5 Imaging Transformation

A point in Cartesian coordinates can be represented as a vector:

$$w = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$$

it's homogeneous coordinates is

$$w_h = \begin{bmatrix} kX & kY & kZ & k \end{bmatrix}^T$$

If we define perspective projection transform matrix as:

$$p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix}$$

2.5 Imaging Transformation

then

$$c_h = Pw_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/\lambda & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ -kZ/\lambda + k \end{bmatrix}$$

The coordinates of a point in camera coordinate are:

$$c = \begin{bmatrix} x & y & z \end{bmatrix}^T = \begin{bmatrix} \dfrac{\lambda X}{\lambda - Z} & \dfrac{\lambda Y}{\lambda - Z} & \dfrac{\lambda Z}{\lambda - Z} \end{bmatrix}^T$$

2.5 Imaging Transformation

Inverse transformation can map a point in image into 3-D space:

$$w_h = P^{-1} c_h$$

Where P^{-1} is:

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/\lambda & 1 \end{bmatrix}$$

2.5 Imaging Transformation

If the coordinates of a point in image is(x’,y’,0), then it’s homogeneous vector is:

$$c_h = \begin{bmatrix} kx' & ky' & 0 & k \end{bmatrix}^T$$

And it’s homogeneous coordinates is

$$w_h = \begin{bmatrix} kx' & ky' & 0 & k \end{bmatrix}^T$$

Coordinate vector in Cartesian coordinates is

$$w = \begin{bmatrix} X & Y & Z \end{bmatrix}^T = \begin{bmatrix} x' & y' & 0 \end{bmatrix}^T$$

2.5 Imaging Transformation

$$\begin{cases} X = \dfrac{x'}{\lambda}(\lambda - Z) \\ Y = \dfrac{y'}{\lambda}(\lambda - Z) \\ c_h = \begin{bmatrix} kx' & ky' & kz & k \end{bmatrix}^T \end{cases}$$

$$w_h = \begin{bmatrix} kx' & ky' & kz & \dfrac{kz}{\lambda} + k \end{bmatrix}^T$$

Change to Cartesian coordinates

$$w = \begin{bmatrix} X & Y & Z \end{bmatrix}^T = \begin{bmatrix} \dfrac{\lambda x'}{\lambda + z} & \dfrac{\lambda y'}{\lambda + z} & \dfrac{\lambda z}{\lambda + z} \end{bmatrix}^T$$

2.5 Imaging Transformation

In other words:

$$X = \frac{\lambda x'}{\lambda + z}$$

$$Y = \frac{\lambda y'}{\lambda + z}$$

$$Z = \frac{\lambda z}{\lambda + z}$$

$\Longrightarrow$

$$X = \frac{x'}{\lambda}(\lambda - Z)$$

$$Y = \frac{y'}{\lambda}(\lambda - Z)$$

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data presentation

Assume that an image f(x,y) is sampled so that the resulting digital image has M rows and N columns.We use integer values for those of the coordinates (x,y).

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data presentation

We often use a more traditional matrix notation to denote a digital image and its elements.

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

- Clearly, $a_{ij} = f(i, j)$

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data presentation

two dimensional array	**A[M][N]**
Image size	**M*N**
Pixel at point (i,j)	A[i][j]

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image Types

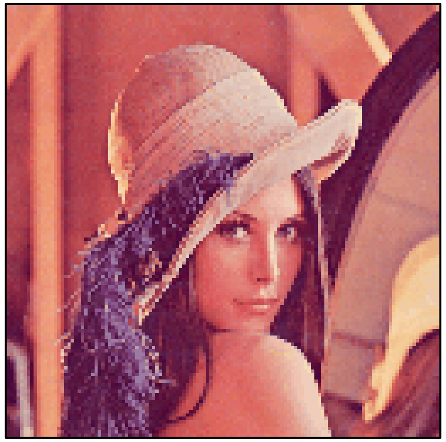
Type	Bits/pixel	color levers
Binary image	1	2
Gray Image	8	256
Color image	24	16777216
Multispectral Image	8*n	

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image Types

Binary Image

Gray Image



Color Image

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image statistic parameters

Max value: $ma\_value = \max(A(i,j))$

Min value: $mi\_value = \min(A(i,j))$

Mean value:
$$me\_value = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} A(i,j)$$

Variance:
$$\sigma^2 = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(A(i,j) - me\_value)^2$$

Where: $i = 0,1,\ldots M-1$ $\quad j = 0,1,\ldots N-1$

2.6 Sampling and Quantization

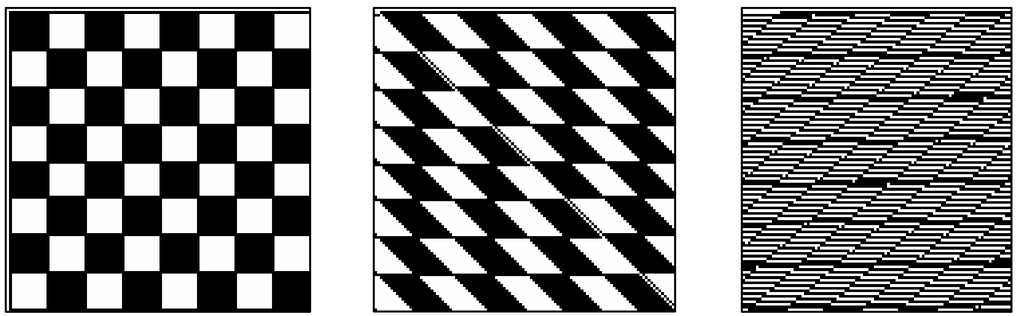
2.6.1 Image Fundaments: Image data file formats

Name	type	usage
raw data format	*.dat, *.raw	Dos, UNIX and Macintosh image
Bit-mapped format	*.bmp	Microsoft Windows format
Tagged file format	*.tif	Dos, UNIX and Macintosh image

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Raw data file: simple but no any attached information.
If you don't know the width of image, then:



113 **112** **100**

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Bit-mapped file:

File head	Bitmap head	Color map	data

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Bit-mapped file: file head

Offset	**Length**	**Name**	**Description**
0	2	bfType	“BM”
2	4	bfSize	Size of file
6	2	bfReserved1	0
8	2	bfReserved2	0
10	4	bfOffBits	Offset after file head

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Bit-mapped file: bitmap head

Offset	Length	Name	Description
14	4	biSize	Size of bitmap head, 40
18	4	biWidth	Width of Image
22	4	biHeight	Height of Image
26	2	biPlanes	Always “1”
28	2	biBitCount	Bits/pixel, 1,4,8 or 24
30	4	biCompression	Size of compressed file
34	4	bfSizeImag	Offset after file head

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Bit-mapped file: bitmap head

Offset	**Length**	**Name**	**Description**
38	4	biXPelsPerMeter	Resolution in horizon direction
42	4	biYPelsPerMeter	Resolution in vertical direction
46	4	biClrUsed	Color number used
50	4	biClrImportant	Important color's number
54	4*N	bmiColors	Color Mapping table

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Bit-mapped file: Color map

Offset	Length	Name	Description
0	1	rgbBlue	Value of blue
1	1	rgbGreen	Value of green
2	1	rgbRed	Value of red
3	1	rgbReserved	0

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

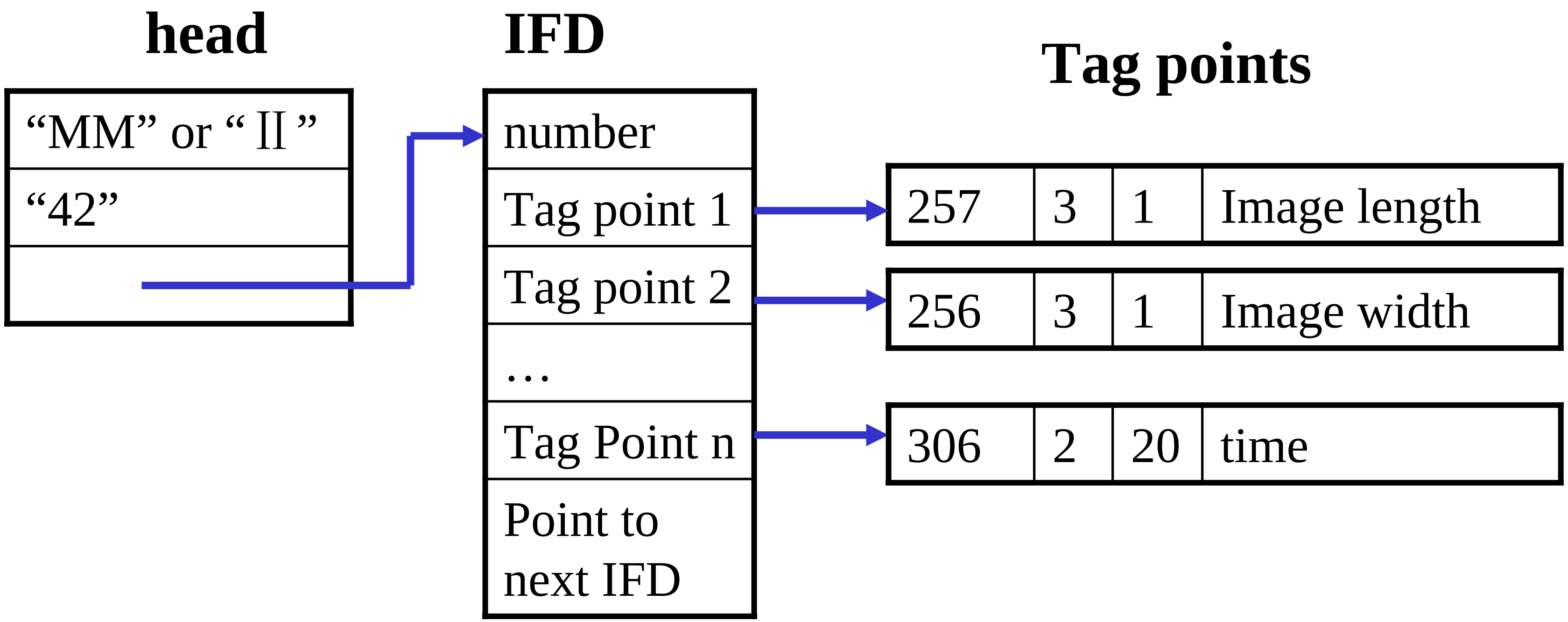
Bit-mapped file: example of Color map biBitCount=8

Number	**rgbBlue**	**rgbGreen**	**rgbRed**
0	0	0	0
1	1	1	1
. . .	. . .	. . .	. . .
255	255	255	255

2.6 Sampling and Quantization

2.6.1 Image Fundaments: Image data file formats

Tagged file format



IFD: image file directory

2.6 Sampling and Quantization

2.6.2 Spatial resolution

The number G of discrete gray levels and the size of image are typically integer power of 2. The range of values spanned by the gray levels is called the dynamic range of an image

$$G = 2^k \qquad M = 2^m \qquad N = 2^n$$

The number b of bits required to store a digital image is :

$$b = M \times N \times k$$

When $M{=}N$:

$$b = N^2 k$$

2.6 Sampling and Quantization

2.6.3 Gray level resolution :uniform quantization

8 bits　　4 bits　　2 bits　　1 bit

2.6 Sampling and Quantization

2.6.3 Zooming and Shrinking Digital Images

Zooming requires two steps:

(1) the creation of new pixel locations

(2) the assignment of gray levels to those new locations.

Methods:

(1) Nearest neighbor interpolation

(2) Bilinear interpolation

2.7 Relationship between Pixels

2.7.1 Neighborhood of a pixel

A pixel p at coordinates (x,y) has four horizontal and vertical neighbors whose coordinates are given by This set of pixels,called the 4-neighbors of p,is denoted by $N_4(p)$

0	$(x, y-1)$	0
$(x-1, y)$	(x, y)	$(x+1, y)$
0	$(x, y+1)$	0

2.7 Relationship between Pixels

2.7.1 Neighborhood of a pixel

The four *diagonal* neighbors of *p* have coordinates. and are denoted by

$N_D(p)$

$(x-1, y-1)$	0	$(x+1, y-1)$
0	(x, y)	0
$(x-1, y+1)$	0	$(x+1, y+1)$

$N_D(p)$ together with $N_4(p)$,are called the *8-neighbors* of *p*,denoted by $N_8(p)$

2.7 Relationship between Pixels

2.7.2 Connectivity

To establish if two pixels are connected,it must be determined if they are neighbors and if their gray levels satisfy a specified criterion of similarity. If *V* is defined as the set of gray levels, then

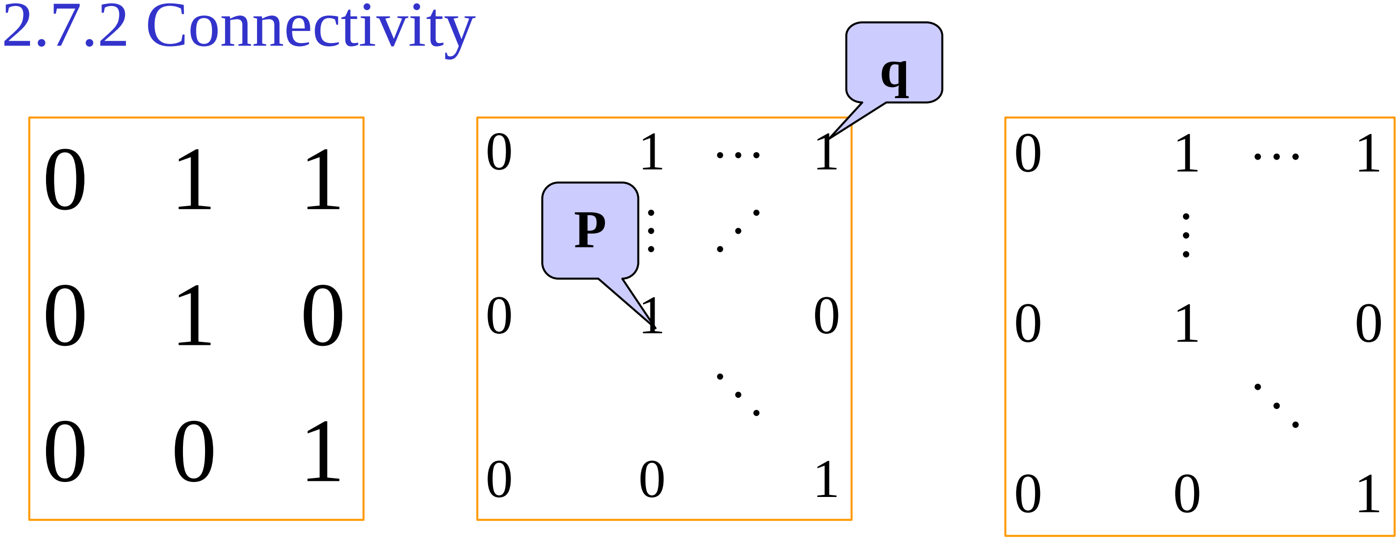
- *4-adjacency:* two pixels *p* and *q* with values from *V* are 4-adjacent if *q* is in the set $N_4(p)$
- *8-adjacency* : two pixels *p* and *q* with values from *V* are 8-adjacent if *q* is in the set $N_8(p)$
- *m-adjacency* : two pixels *p* and *q* with values from *V* are 4-adjacent if

  (i)*q* is in $N_4(p)$,or

  (ii) *q* is in $N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from *V*.

2.7 Relationship between Pixels

2.7.2 Connectivity



a *b* *c*

Figure 2.26 (a) Arrangement of pixels;(b) pixels that are 8-adjacent;(c)m-adjacent

2.7 Relationship between Pixels

Problem: consider the two image subsets, *S* and *T* shown in the following figure. For *V*={1}, determine whether these Two subsets are (a) 4-adjacent, (b)8-adjacent or (c) m-adjacent

S				T			
1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	0
0	0	1	0	1	1	0	0
0	1	0	1	0	0	0	0

2.7 Relationship between Pixels

q

S **T**

1	0	1	0	0	0	1	1
0	1	1	0	0	0	1	0
0	0	1	0	1	1	0	0
0	1	0	1	0	0	0	0

p

Solution: Let p and q be as shown in the following figure. Then, (a) S and T are not 4-connected because q is not in the set $N_4(p)$. (b) S and T are 8-connected because q is in the set $N_8(p)$. (c) S and T are m-connected because (i) q is in $N_D(p)$, and (ii) the set $N_4(p) \cap N_4(q)$ is empty.

2.7 Relationship between Pixels

2.7.3 Distance Measures: definitions

For pixels p,q and z ,with coordinates (x,y),(s,t),and(u,v),respectively, D is a *distance function or metric* if

(a) $D(p,q) \geq 0 \quad (D(p,q)=0, if \;\; p=q)$

(b) $D(p,q) = D(q,p), and$

(c) $D(p,z) \leq D(p,q) + D(q,z)$

2.7 Relationship between Pixels

2.7.3 Distance Measures: definitions

- The Euclidean distance is

$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{\frac{1}{2}}$$

- And the city-block and chessboard distances are:

$$D_4(p,q) = |x-s| + |y-t|$$

$$D_8(p,q) = \max(|x-s|, |y-t|)$$

- The D_m distance is defined as the shortest m-path between the points.

2.7 Relationship between Pixels

2.7.3 Distance Measures: example 1

City-block distance(D_4)

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

(a)

Chessboard distance (D_8)

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

(b)

图2.7.4　等距离轮廓实例

2.7 Relationship between Pixels

2.7.3 Distance Measures: example 2

$$\begin{array}{ccccc} & & 0 & & 1(q) \\ & & & \iddots & \\ 0 & & 1 & & \\ & \iddots & & & \\ 1(p) & & & & \end{array}$$

$D_m(p,q)=2$

$$\begin{array}{ccccc} & & 0 & & 1(q) \\ & & & \iddots & \\ 1 & \cdots & 1 & & \\ \vdots & & & & \\ 1(p) & & & & \end{array}$$

$D_m(p,q)=3$

$$\begin{array}{ccccc} & & 1 & \cdots & 1(q) \\ & & \vdots & & \\ 0 & & 1 & & \\ & \iddots & & & \\ 1(p) & & & & \end{array}$$

$D_m(p,q)=3$

$$\begin{array}{ccccc} & & 1 & \cdots & 1(q) \\ & & \vdots & & \\ 1 & \cdots & 1 & & \\ \vdots & & & & \\ 1(p) & & & & \end{array}$$

$D_m(p,q)=4$

2.7 Relationship between Pixels

2.7.3 Distance Measures: example 3

Problem: consider the image segment shown
Let *V*={0,1} and compute the lengths of the shortest 4- 8- and m-path between *p* and *q*. if a particular path does not exist between these two points, explain why.

q

0	1	2	1
2	2	0	2
2	3	1	1
1	0	1	2

p

2.7 Relationship between Pixels

2.7.3 Distance Measures: example 3

0	1	2	1
2	2	0	2
2	3	1	1
1	0	1	2

a) When $V = \{0,1\}$, 4-path does not exist between p and q because it is impossible to get from p to q by traveling along points that are both 4-adjacent and also have values from V .

b) The shortest 8-path is shown in right Figure, its length (shown magenta) is 4. The length of shortest m-path (shown blue) is 5.

0	1	2	1
2	2	0	2
2	3	1	1
1	0	1	2

2.8 Arithmetic and Logic Operation

2.8.1 Arithmetic operation

Arithmetic operations between two pixels *p* and *q* include:

Addition: $p+q$

Subtraction: $p-q$

Multiplication: $p*q$

Division: p/q

2.8 Arithmetic and Logic Operation

2.8.2 Logical operation:

AND OR NOT

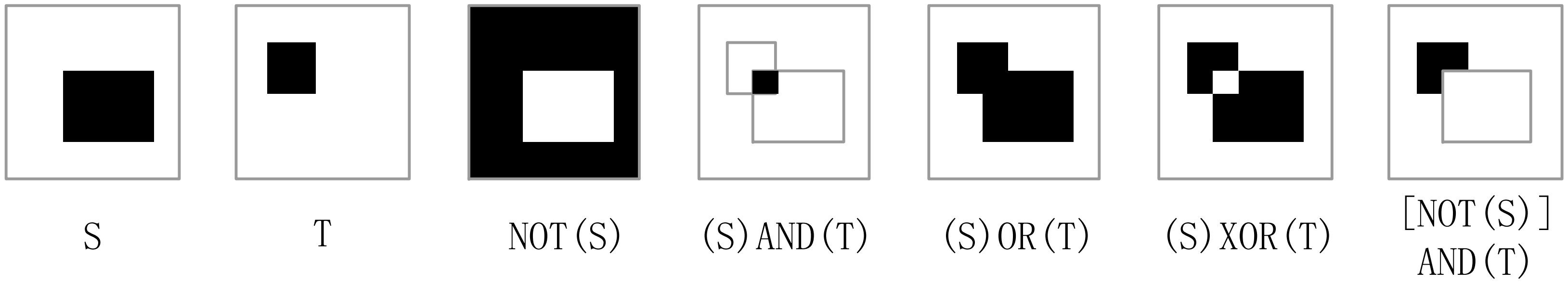


图2.8.1 二值图象的逻辑运算

2.8 Arithmetic and Logic Operation

2.8.3 Point operation and Local operation:

A(x,y) $\longrightarrow$ ***B(x,y)***

Point Operation: $B(x, y) = f(A(x, y))$

Local Operation: $B(x, y) = f(A(x \pm s, y \pm t))$

Windows: $s = 0,1,\ldots S-1 \quad t = 0,1,\ldots T-1$

Usually, *S=T=1,3,5,7…*

2.8 Arithmetic and Logic Operation

2.8.4 Local operation:

$$z = \frac{1}{9}(z_1 + z_2 + \cdots + z_9) = \frac{1}{9}\sum_{i=1}^{9} z_i$$

$$z = w_1 z_1 + w_2 z_2 + w_n z_n = \sum_{i=1}^{9} w_i z_i$$

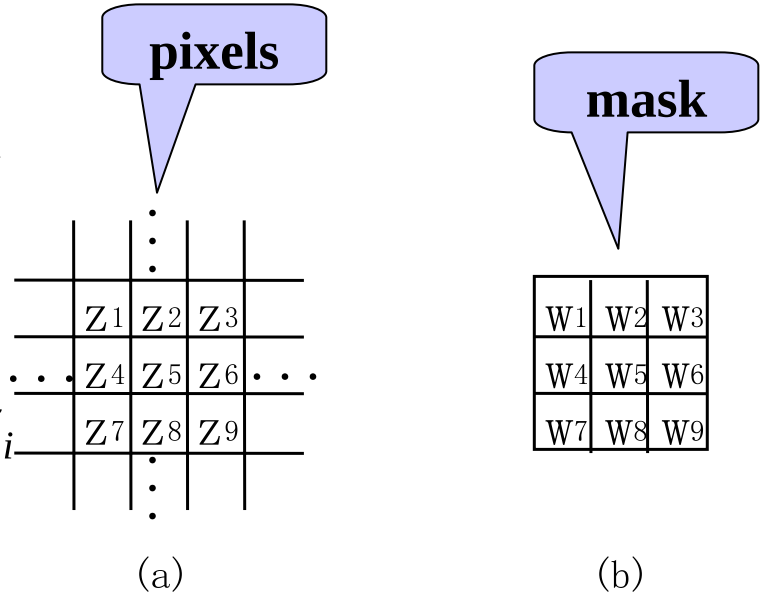


图2.8.2　模板运算示例

summary

- Basic idea of the eye in perceiving pictorial information
- Color fundamentals and color models
- Fundamentals of images, include presentation, read, write display, sampling and quantization, relationship between pixels, and so on.

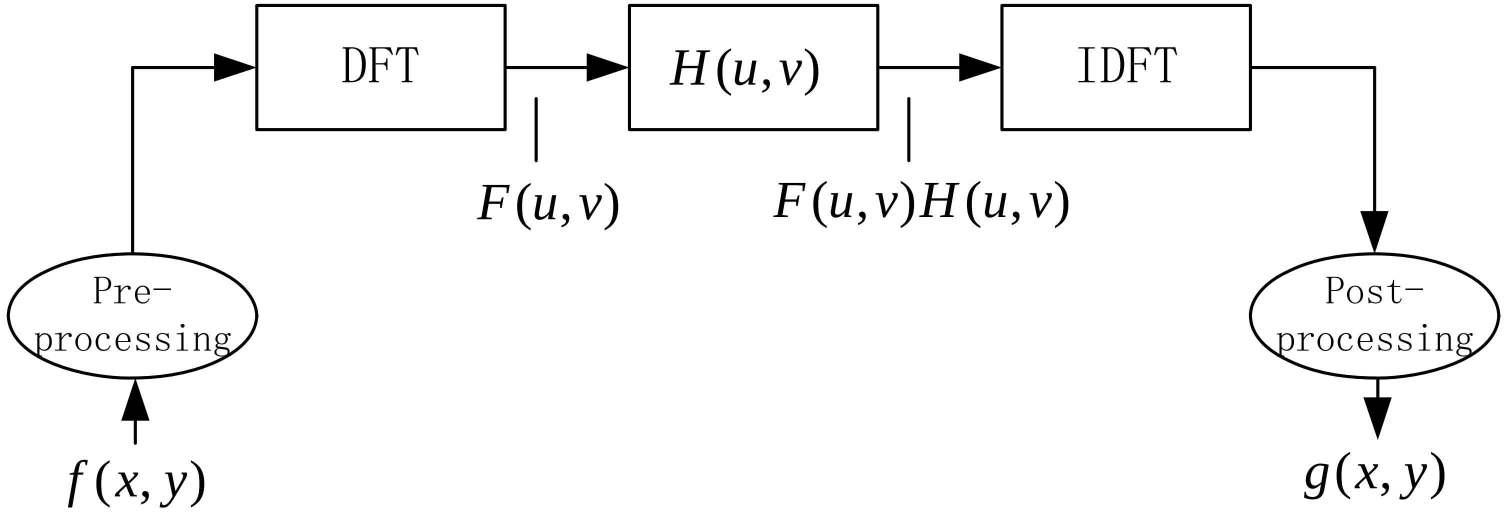
Chapter3 Image Transforms

- 3.1General Introduction and Classification
- 3.2 The Fourier Transform and Properties
- 3.3 Other Separable Image Transforms
- 3.4 Hotelling Transform

Chapter3 Image Transforms

Preview

General steps of operation in frequency domain



3.1 General Introduction and Classification

3.1.1 introduction

- image transforms are the bases of image processing and analysis
- this chapter deals with two-dimensional transforms and their properties
- image transforms are used in image enhancement, restoration, reconstruction, encoding and description

3.1 General Introduction and Classification

3.1.1 classification

$$\text{image transforms}\begin{cases}\text{separable transforms}\begin{cases}\text{DFT and its properties}\\ \text{others}\begin{cases}\text{DCT}\\ \text{Hadamard Transform}\\ \text{Walsh Transform}\\ \text{Slant Transform}\\ \text{Haar Transform}\end{cases}\end{cases}\\ \text{statistical transforms: Hotelling transform}\end{cases}$$

3.2 Fourier Transform and Properties

3.2.2 definitions: 1-D CFT

The One-Dimensional Continuous Fourier Transform and its Inverse

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{j2\pi xu} du$$

3.2 Fourier Transform and Properties

3.2.2 definitions:1-D DFT

The One-Dimensional Discrete Fourier Transform and its Inverse

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-j\frac{2\pi}{N}ux} \qquad u = 0,1,\cdots N-1$$

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{j\frac{2\pi}{N}ux} \qquad x = 0,1,\cdots N-1$$

3.2 Fourier Transform and Properties

3.2.2 definitions (2) :1-D DFT

Other expressions:

$$F(u) = R(u) + jI(u)$$

or

$$F(u) = \left|F(u)\right| \exp\left[j\phi(u)\right]$$

Euler's formula:

$$\exp\left[-j2\pi ux\right] = \cos 2\pi ux - j\sin 2\pi ux$$

3.2 Fourier Transform and Properties

3.2.2 definitions: spectrum, Phase angle Power spectrum

Magnitude or spectrum

$$|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2}$$

Phase angle or phase spectrum

$$\phi(u) = \arctan\left[I(u)/R(u)\right]$$

Power spectrum
（Spectral density)

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

3.2 Fourier Transform and Properties

3.2.2 definitions:1-D DFT example

If a signal is expressed as *f(x)={2,3,4,4}*, its DFT are:

$$F(0)=\frac{1}{4}\sum_{x=0}^{3} f(x)\exp(0)=\frac{1}{4}[f(0)+f(1)+f(2)+f(3)]=3.25$$

$$F(1)=\frac{1}{4}\sum_{x=0}^{3} f(x)\exp(-j2\pi x/4)=\frac{1}{4}[2e^{0}+3e^{-j\pi/2}+4e^{-j\pi}+4e^{-j3\pi/2}]=\frac{1}{4}(-2+j)$$

$$F(2)=\frac{1}{4}\sum_{x=0}^{3} f(x)\exp(-j4\pi x/4)=\frac{1}{4}[2e^{0}+3e^{-j\pi}+4e^{-2j\pi}+4e^{-j3\pi}]=-\frac{1}{4}(1+j0)$$

$$F(3)=\frac{1}{4}\sum_{x=0}^{3} f(x)\exp(-j6\pi x/4)=\frac{1}{4}[2e^{0}+3e^{-j3\pi/2}+4e^{-j3\pi}+4e^{-j9\pi/2}]=-\frac{1}{4}(2+j)$$

3.2 Fourier Transform and Properties

3.2.2 definitions:1-D DFT example

And the Fourier spectra are :

$$|F(0)| = 3.25$$

$$|F(1)| = [(2/4)^2 + (1/4)^2]^{1/2} = \sqrt{5}/4$$

$$|F(2)| = [(1/4)^2 + (0/4)^2]^{1/2} = 1/4$$

$$|F(3)| = [(2/4)^2 + (1/4)^2]^{1/2} = \sqrt{5}/4$$

$$\phi(0) = 0$$

$$\phi(1) = 0.85\pi$$

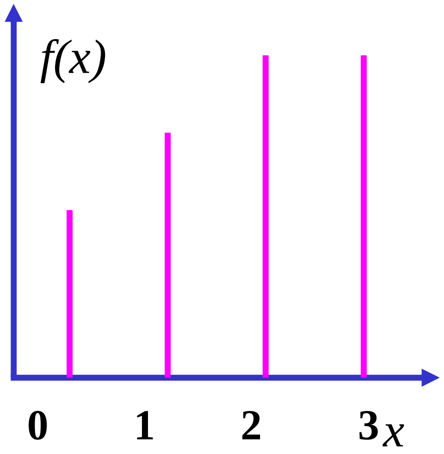
$$\phi(3) = \pi$$

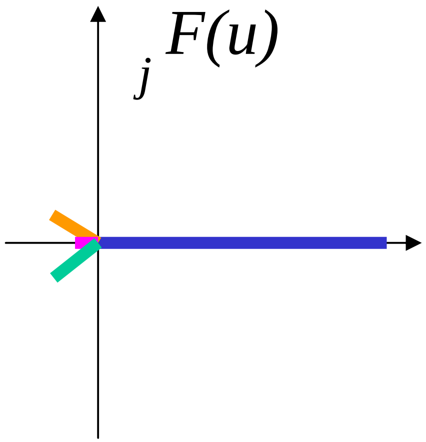
$$\phi(3) = -0.85\pi$$

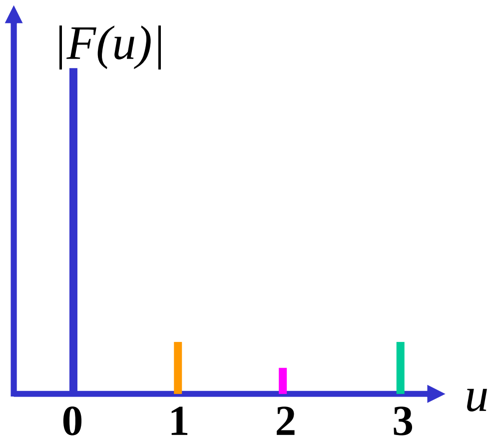
3.2 Fourier Transform and Properties

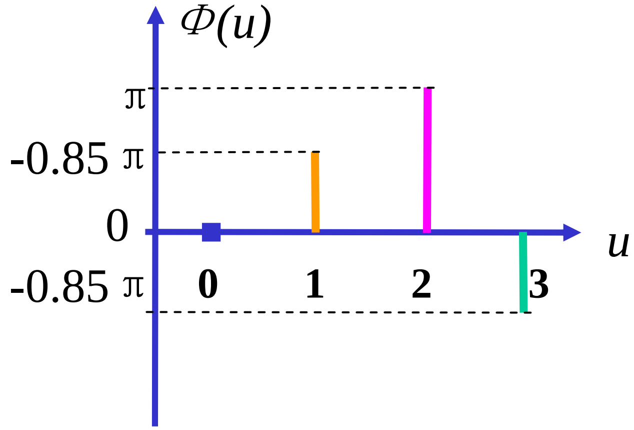
3.2.2 definitions:1-D DFT example

Graphic illustration ：









3.2 Fourier Transform and Properties

3.2.2 definitions: 2D-DFT

The Two-Dimensional Discrete Fourier Transform and its Inverse

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)} \qquad \begin{array}{l} u = 0,1,\cdots M-1 \\ v = 0,1,\cdots N-1 \end{array}$$

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)} \qquad \begin{array}{l} x = 0,1,\cdots M-1 \\ y = 0,1,\cdots N-1 \end{array}$$

3.2 Fourier Transform and Properties

3.2.2 definitions: 2D-DFT

Magnitude or spectrum

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

Phase angle or phase spectrum

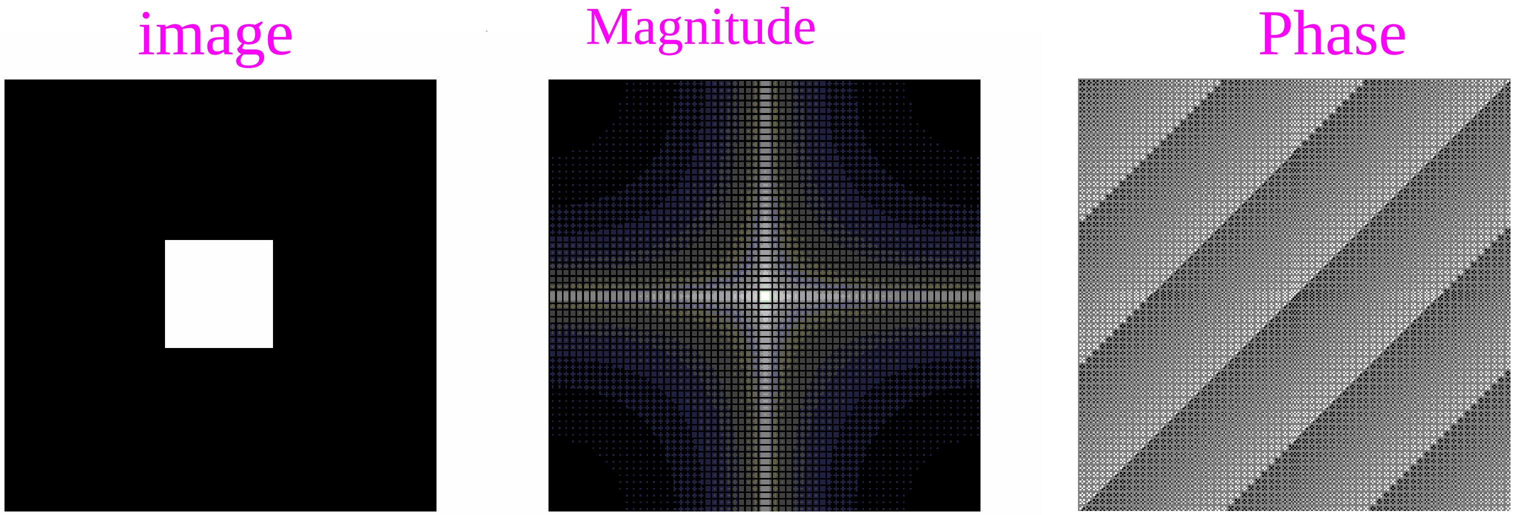
$$\phi(u,v) = \arctan\left[I(u,v)/R(u,v)\right]$$

Power spectrum
（Spectral density)

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

3.2 Fourier Transform and Properties

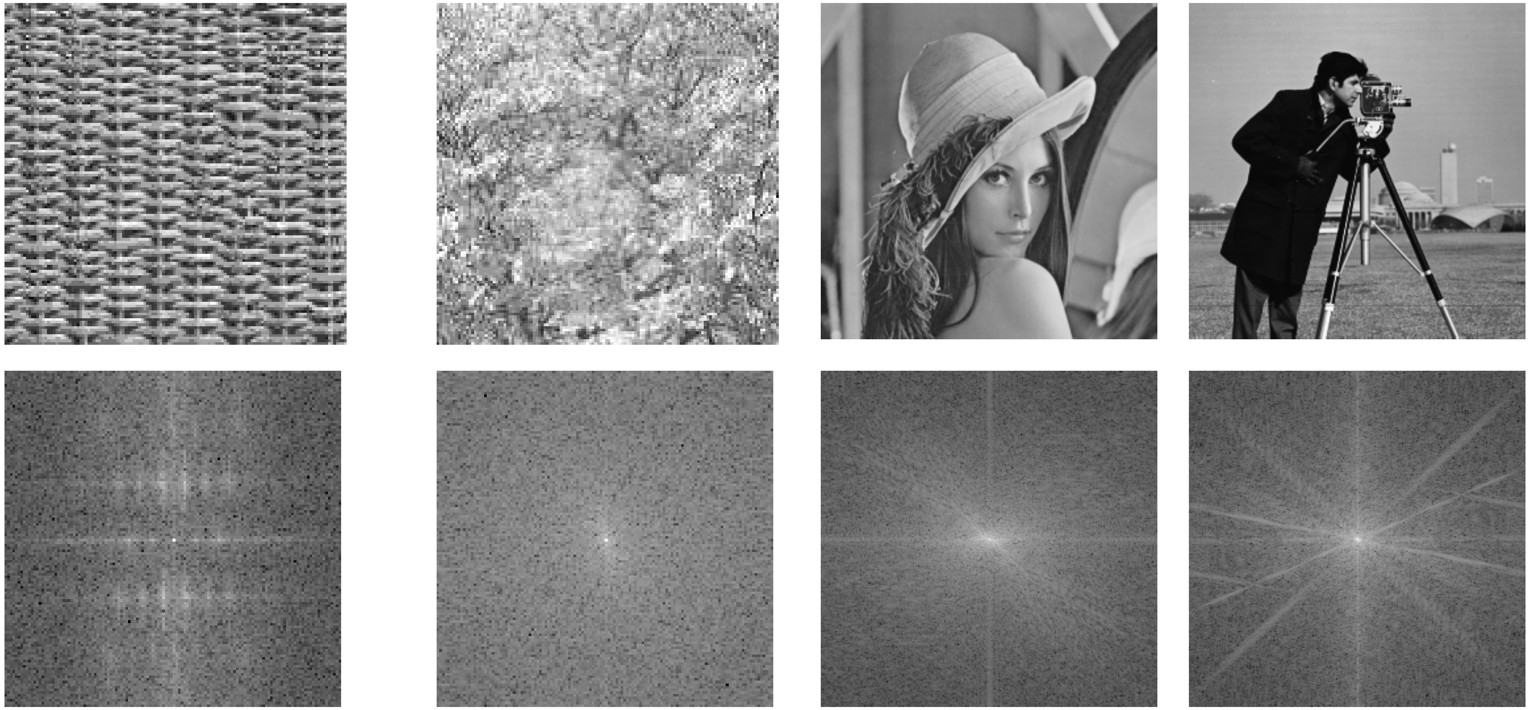
3.2.2 definitions: 2D-DFT



3.2 Fourier Transform and Properties

3.2.2 definitions: 2D-DFT

Typical images and their spectra



3.2 Fourier Transform and Properties

3.2.3 Properties: Display

- Usually the Fourier spectra are displayed as intensity function.
- many image spectra decrease rather rapidly as a function of increasing frequency
- their high-frequency terms have a tendency to become obscured when displayed in image.

3.2 Fourier Transform and Properties

3.2.3 Properties: Display

• A useful processing technique which compensates for this difficulty consists of displaying the function

$$D_1(u,v) = \log(1 + |F(u,v)|)$$

Or

$$D_2(u,v) = \begin{cases} |F(u,v)| + 100 \\ 255 \quad \text{if } |F(u,v)| + 100 > 255 \end{cases}$$

•instead of $|F(u,v)|$

3.2 Fourier Transform and Properties

3.2.3 Properties: separability

The DFT pair can be expressed in the separable forms:

$$F(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\exp\left[\frac{-j2\pi ux}{N}\right]\sum_{y=0}^{N-1}f(x,y)\exp\left[\frac{-j2\pi vy}{N}\right]$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\exp\left[\frac{j2\pi ux}{N}\right]\sum_{v=0}^{N-1}F(u,v)\exp\left[\frac{j2\pi vy}{N}\right]$$

3.2 Fourier Transform and Properties

3.2.3 Properties: separability

The principal of the separability property is that $f(x,y)$ or $F(u,v)$ can be obtained in two steps by successive applications of the 1-D Fourier transform or its inverse

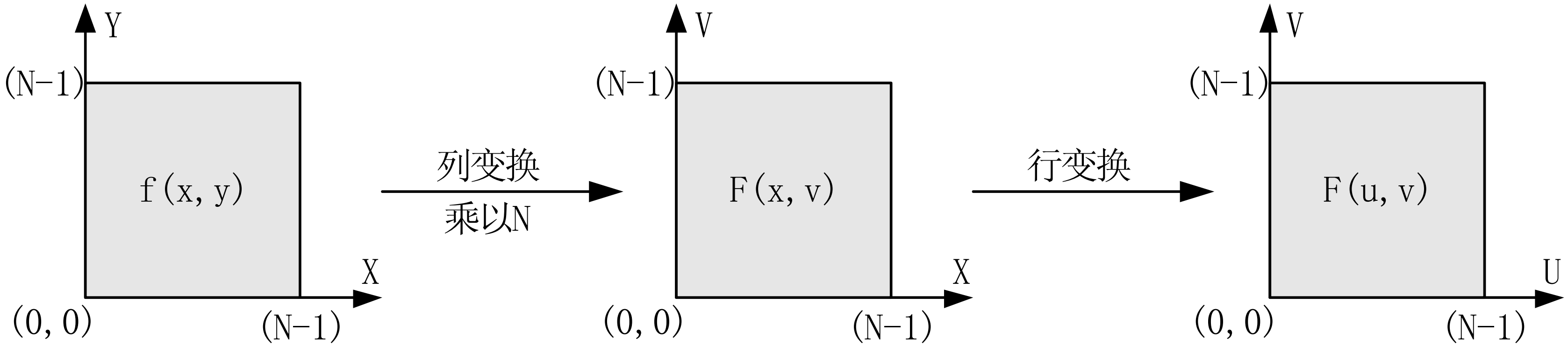
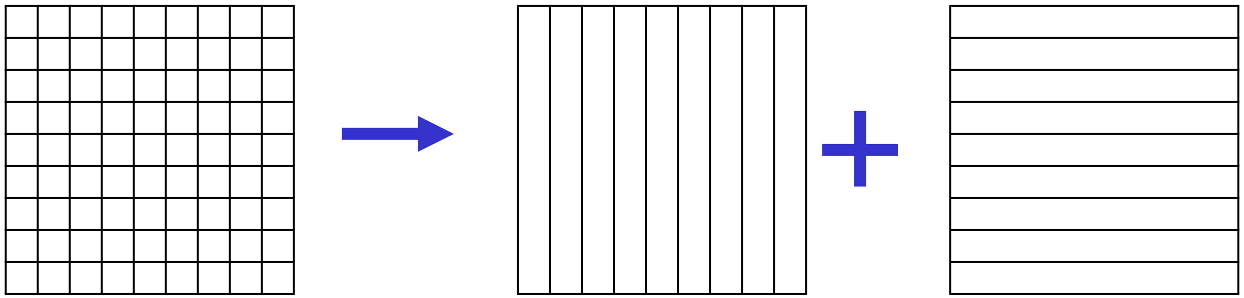


图3.2.2　由2步1-D变换计算2-D变换

3.2 Fourier Transform and Properties

3.2.3 Properties: separability

For a N*N image, it can be separated into 2N 1D-DFT



3.2 Fourier Transform and Properties

3.2.3 Properties: periodicity and conjugate symmetry

The discrete Fourier transform and its inverse are *periodic* with Period *N*

$$F(u,v) = F(u+N,v) = F(u,v+N) = F(u+N,v+N)$$

The Fourier transform also exhibits conjugate symmetry since

$$F(u,v) = F^*(-u,-v)$$

Or more interestingly,

$$\left|F(u,v)\right| = \left|F(-u,-v)\right|$$

3.2 Fourier Transform and Properties

3.2.3 Properties: translation

The translation properties of the Fourier transform pair are given by

$$f(x,y)\exp\left[j2\pi(u_0x+v_0y)/N\right]\Leftrightarrow F(u-u_0,v-v_0)$$

and

$$f(x-x_0,y-y_0)\Leftrightarrow F(u,v)\exp\left[-j2\pi(ux_0+vy_0)/N\right]$$

3.2 Fourier Transform and Properties

3.2.3 Properties: translation

A shift in *f(x,y)* does not affect the magnitude of its Fourier transform since

$$|F(u,v)\exp\left[-j2\pi(ux_0+vy_0)/N\right]|=|F(u,v)|$$

3.2 Fourier Transform and Properties

3.2.3 Properties: translation

Example: in the case $u_0=v_0=N/2$, it follows that

and

$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-N/2, v-N/2)$$

3.2 Fourier Transform and Properties

3.2.3 Properties: rotation

If we introduce the polar coordinates

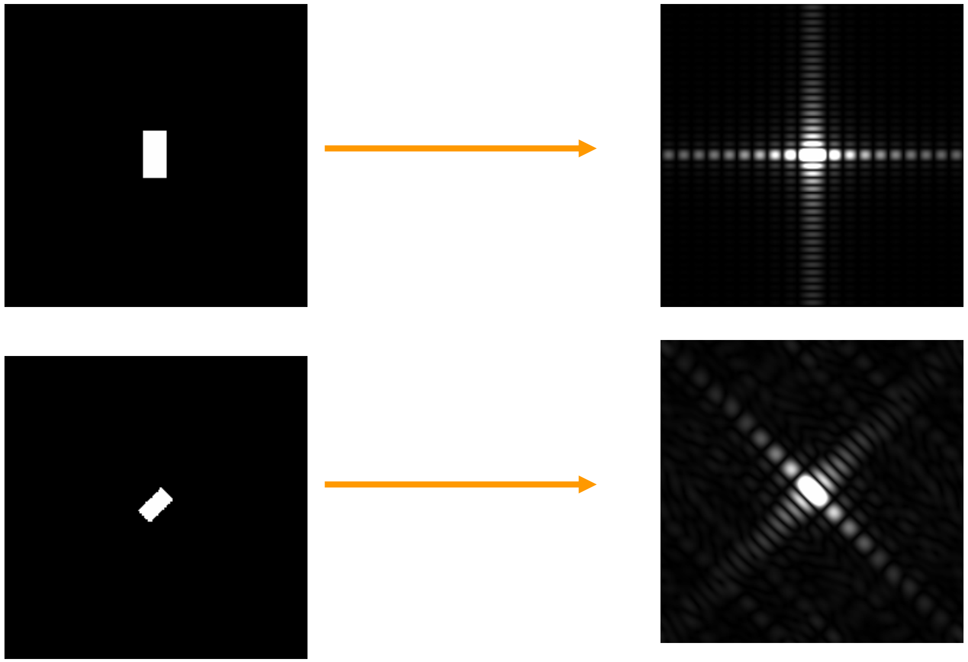
$$x = r\cos(\theta) \quad y = r\sin(\theta) \; u = \omega\cos(\phi) \; v = \omega\sin(\phi)$$

Then f(x,y) and F(u,v)become $f(r,\theta)$ and $F(\omega,\phi)$ respectively

$$f(r,\theta+\theta_0) \Leftrightarrow F(w,\phi+\theta_0)$$

3.2 Fourier Transform and Properties

3.2.3 Properties: rotation



3.2 Fourier Transform and Properties

3.2.3 Properties: Distributivity

It follows directly from the definition of the transform pair that,

$$F\left\{f_1(x,y)+f_2(x,y)\right\}=F\left\{f_1(x,y)\right\}+F\left\{f_2(x,y)\right\}$$

And, in general that,

$$F\left\{f_1(x,y)\cdot f_2(x,y)\right\}\neq F\left\{f_1(x,y)\right\}\cdot\left\{f_2(x,y)\right\}$$

3.2 Fourier Transform and Properties

3.2.3 Properties: scaling

It is also easy to show that for two scalar *a* and *b*

$$af(x,y) \Leftrightarrow aF(u,v)$$

and

$$f(ax,by) \Leftrightarrow \frac{1}{|ab|} F\left(\frac{u}{a},\frac{v}{b}\right)$$

3.2 Fourier Transform and Properties

3.2.3 Properties: average value

A widely-used definition of the average value of a 2D Discrete function is given by the expression

$$\tilde{f}(x,y) = \frac{1}{N^2}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)$$

Substitution of u-v-0 in definition of 2D DFT yields

$$F(0,0) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y) \longrightarrow \tilde{f}(x,y) = \frac{1}{N}F(0,0)$$

3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

The convolution of two functions *f(x)* and *g(x)*, denoted by *f(x)*g(x)*, is defined by the integral:

$$f(x)*g(x)=\int_{-\infty}^{\infty} f(z)g(x-z)dz$$

Where *z* is a dummy variable of integration

3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

Example1: graphic illustration of convolution $f(x)*g(x)$

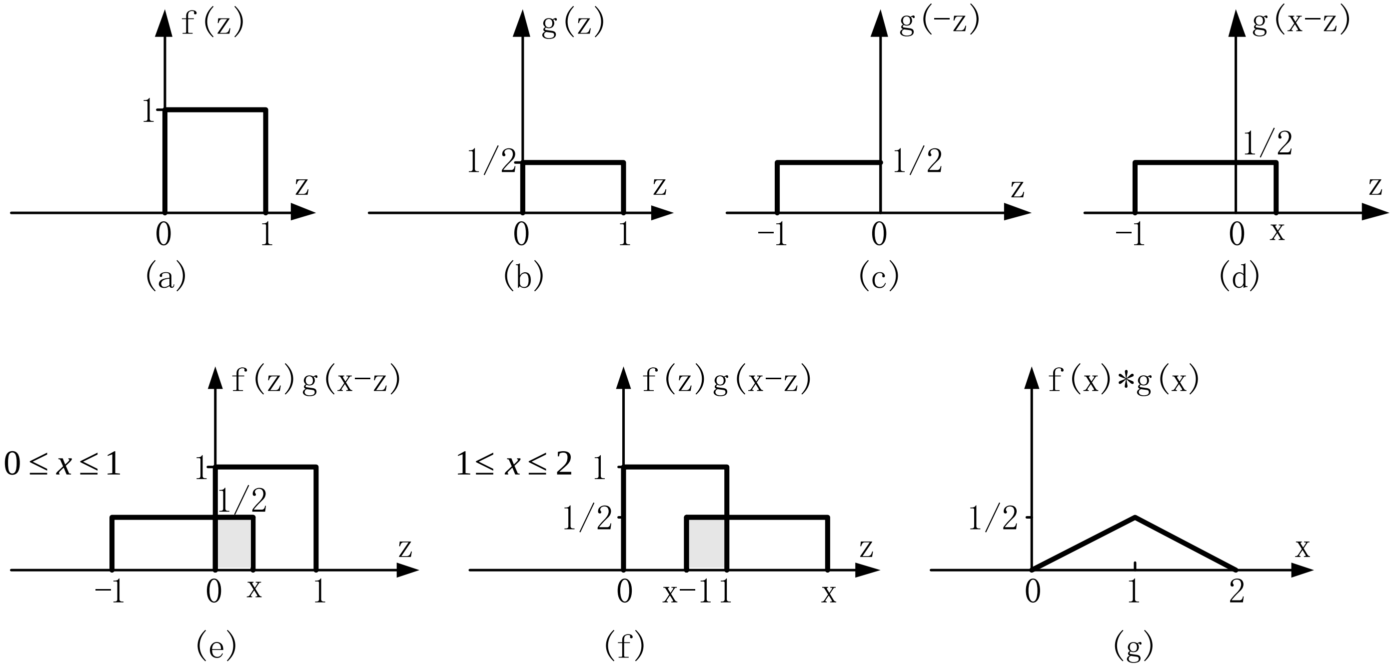
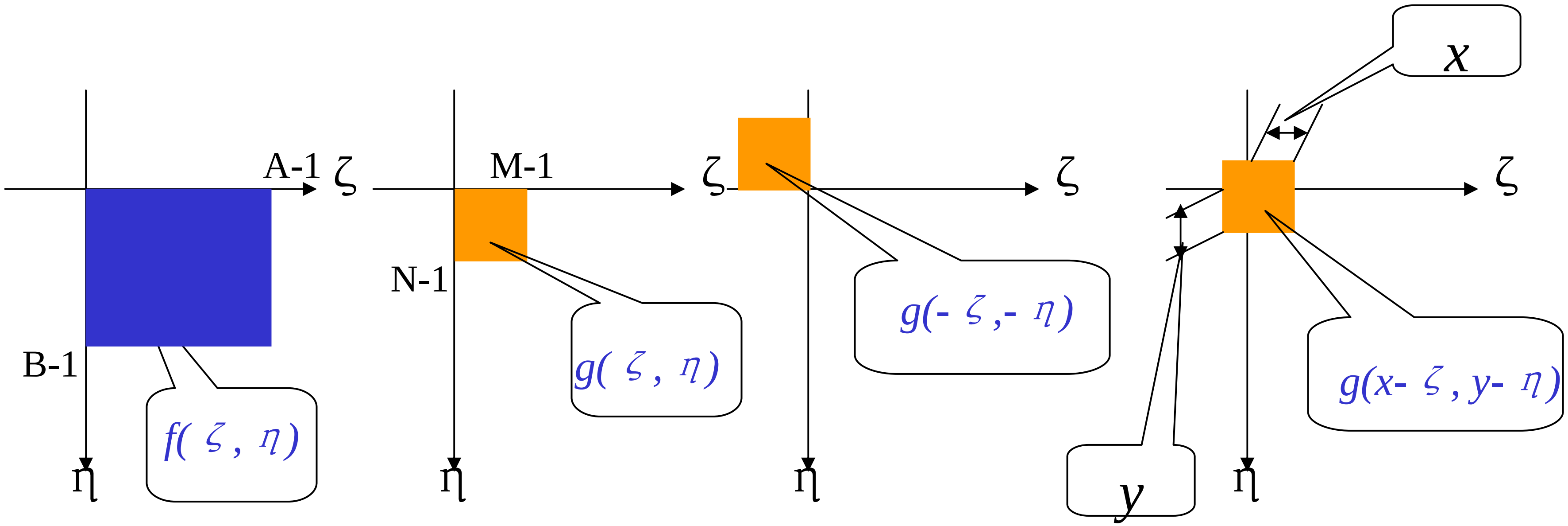


图3.2.3　1-D函数卷积示例

3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

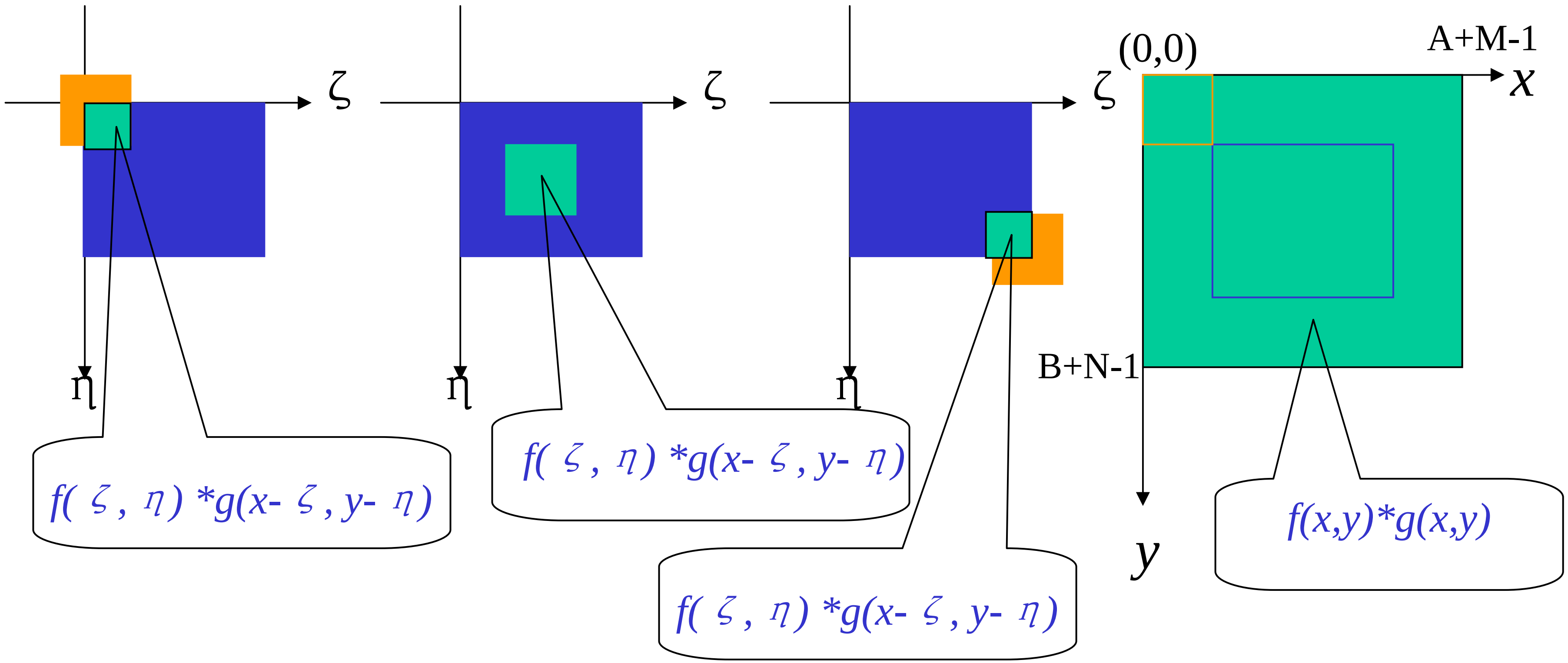
Example2: graphic illustration of convolution *f(x,y)*g(x,y)*



3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

Example2: graphic illustration of convolution *f(x,y)*g(x,y)*



3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

If *f(x,y)* has the Fourier transform *F(u,v)* and *g(x,y)* has the Fourier Transform *G(u,v)*, then *f(x,y)*g(x,y)* has the Fourier transform *F(u,v)G(u,v)*.This result, formally stated as:

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

And the convolution in *frequency* domain reduces to multiplication In the *spatial*-domain

$$f(x,y)g(x,y) \Leftrightarrow F(u,v) * G(u,v)$$

3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

Definition of 1D-discrete convolution

$$f_e(x) = \begin{cases} f(x) & 0 \le x \le A-1 \\ 0 & A \le x \le M-1 \end{cases}$$

$$g_e(x) = \begin{cases} g(x) & 0 \le x \le B-1 \\ 0 & B \le x \le M-1 \end{cases}$$

$$f_e(x) * g_e(x) = \frac{1}{M} \sum_{m=0}^{M-1} f_e(m) g_e(x-m) \qquad \text{x=0,1,…,M-1}$$

3.2 Fourier Transform and Properties

3.2.3 Properties: convolution

Definition of 2D-discrete convolution

$$f_e(x,y)=\begin{cases} f(x,y) & 0\le x\le A-1 \text{ and } 0\le y\le B-1 \\ 0 & A\le x\le M-1 \text{ or } B\le y\le N-1 \end{cases}$$

$$g_e(x,y)=\begin{cases} g(x,y) & 0\le x\ge C-1 \text{ and } 0\le y\le D-1 \\ 0 & C\le x\le M-1 \text{ or } D\le y\le N-1 \end{cases}$$

$$f_e(x,y)*g_e(x,y)=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)g_e(x-m,y-n) \qquad \begin{array}{l} x=0,1,\ldots,M-1 \\ y=0,1,\ldots,N-1 \end{array}$$

3.2 Fourier Transform and Properties

3.2.3 Properties: 1-D correlation

The correlation of two functions *f(x)* and *g(x)* is defined as:

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f^*(z) g(x+z) dz$$

The discrete equivalent is defined as:

$$f_e(x) \circ g_e(x) = \frac{1}{M} \sum_{m=0}^{M-1} f_e^*(m) g_e(x+m) \qquad x = 0,1,\ldots, M-1$$

3.2 Fourier Transform and Properties

3.2.3 Properties: 2-D correlation

The correlation of two functions *f(x,y)* and *g(x,y)* is defined as:

$$f(x,y) \circ g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^*(p,q)g(x+p,y+q)\,dp\,dq$$

The discrete equivalent is defined as:

$$f_e(x,y) \circ g_e(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e^*(m,n)g_e(x+m,y+n) \qquad \begin{array}{l} x = 0,1,\ldots,M-1 \\ y = 0,1,\ldots,N-1 \end{array}$$

3.2 Fourier Transform and Properties

3.2.3 Properties: 2-D correlation

It can be shown for both continuous and discrete cases That the following *correlation theorem* holds:

$$f(x,y) \circ g(x,y) \Leftrightarrow F^*(u,v)G(u,v)$$

$$f^*(x,y)g(x,y) \Leftrightarrow F(u,v) \circ G(u,v)$$

3.3 Other Separable Transforms

3.3.1 General formulations

Definition1: if X is an *N-by-1* vector and T is an *N-by-N* matrix then:

$$y_i = \sum_{j=0}^{N-1} t_{i,j} x_j$$

polynomial expression

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} t_{0,0} & t_{0,1} & \cdots & t_{0,N-1} \\ t_{1,0} & t_{1,1} & \cdots & t_{1,N-1} \\ \vdots & & \ddots & \\ t_{N-1,0} & t_{N-1,1} & \cdots & t_{N-1,N-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \longrightarrow Y = TX$$

matrix expression

3.3 Other Separable Transforms

3.3.1 General formulations

Definition2: inversion $X = T^{-1}Y$

If *rank(T)= N* then it is a *linear* transform

If $T^{-1} = T^{*t}$ then it is a *Unitary* transform $TT^{*t} = TT^{-1} = I$

If $T^{-1} = T^{t}$ then it is a *orthogonal* transform $TT^{t} = TT^{-1} = I$

3.3 Other Separable Transforms

3.3.1 General formulations

Example: 1-D Discrete Fourier Transform (DFT)

$$F(u) = \sum_{x=0}^{N-1} f(x) e^{-j\frac{2\pi}{N}ux} \longrightarrow F = Tf$$

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{j\frac{2\pi}{N}ux} \longrightarrow f = T^{-1}F$$

$$\Downarrow$$

$$T^{-1} = T^{*t}$$

It is a *Unitary* transform

3.3 Other Separable Transforms

3.3.1 General formulations

Definition3: 2-D transformation

$$y_{m,n} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} \Phi(i, j, m, n)$$

$\Phi(i, j, m, n)$ Can be thought of as a *MN-by-MN* block matrix have *M* rows of *M* blocks, each of which is an *N-by-N* matrix

3.3 Other Separable Transforms

3.3.1 General formulations

Definition4: if $\Phi(i, j, m, n)$ can be separated into the product of rowwise and columnwise component function, that is

$$\Phi(i, j, m, n) = T_r(i, m) T_c(j, n)$$

then the transformation is called *separable*

$$y_{m,n} = \sum_{i=0}^{M-1} \left[\sum_{j=0}^{N-1} x_{i,j} T_c(j, n) \right] T_r(i, m)$$

3.3 Other Separable Transforms

3.3.1 General formulations

Definition5: if two component are identical:

$$\Phi(i, j, m, n) = T(i, m) T(j, n)$$

then the transformation is called *symmetric*

$$y_{m,n} = \sum_{i=0}^{M-1} \left[\sum_{j=0}^{N-1} x_{i,j} T(j,n) \right] T(i,m)$$

or

$$Y = TXT$$

3.3 Other Separable Transforms

3.3.1 General formulations

Example: 2-D Discrete Fourier Transform (DFT)

Separable and *Symmetric Unitary* transform

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\exp\left[\frac{-j2\pi ux}{N}\right]\sum_{y=0}^{N-1} f(x,y)\exp\left[\frac{-j2\pi vy}{N}\right]$$

lets

$$W_M = \exp(j2\pi / M)$$

$$W_N = \exp(j2\pi / N)$$

then

3.3 Other Separable Transforms

3.3.1 General formulations

Example: Matrix expression of 2-D DFT

$$
\begin{bmatrix}
F(0,0) & F(0,1) & \cdots & F(0,N-1) \\
F(1,0) & F(1,1) & \cdots & F(1,N-1) \\
\vdots & \vdots & \ddots & \vdots \\
F(M-1,0) & F(M-1,1) & \cdots & F(M-1,N-1)
\end{bmatrix}
= \frac{1}{\sqrt{M}}
\begin{bmatrix}
W_M^0 & W_M^0 & W_M^0 & \cdots & W_M^0 \\
W_M^0 & W_M^{-1} & W_M^{-2} & \cdots & W_M^{-(M-1)} \\
W_M^0 & W_M^{-2} & W_M^{-4} & \cdots & W_M^{-2(M-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
W_M^0 & W_M^{-(M-1)} & W_M^{-2(M-1)} & \cdots & W_M^{-(M-1)^2}
\end{bmatrix}
$$

$$
\times
\begin{bmatrix}
f(0,0) & f(0,1) & \cdots & f(0,N-1) \\
f(1,0) & f(1,0) & \cdots & f(N-1,0) \\
\vdots & \vdots & \ddots & \vdots \\
f(N-1,0) & f(N-1,0) & \cdots & f(N-1,N-1)
\end{bmatrix}
\times \frac{1}{\sqrt{N}}
\begin{bmatrix}
W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\
W_N^0 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\
W_N^0 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
W_N^0 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)^2}
\end{bmatrix}
$$

3.3 Other Separable Transforms

3.3.1 General formulations

Example: Matrix expression of 2-D DFT

It's denoted as

$$G = W_M F W_N$$

And the inverse is:

$$F = W_M^{-1} G W_N^{-1}$$

3.3 Other Separable Transforms

3.3.1 General formulations

Example: Matrix expression of 2-D DFT

Where:

$$W_M^{-1} = \frac{1}{\sqrt{M}} \begin{bmatrix} W_M^0 & W_M^0 & W_M^0 & \cdots & W_M^0 \\ W_M^0 & W_M^1 & W_M^2 & \cdots & W_M^{(M-1)} \\ W_M^0 & W_M^2 & W_M^4 & \cdots & W_M^{2(M-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_M^0 & W_M^{(M-1)} & W_M^{2(M-1)} & \cdots & W_M^{(M-1)^2} \end{bmatrix}$$

$$W_N^{-1} = \frac{1}{\sqrt{N}} \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_N^0 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix}$$

3.3 Other Separable Transforms

3.3.2 Discrete Cosine Transform (DCT)

The 1-D DCT pair is given by the expression:

$$C(u) = a(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad x=0,1,\ldots N\text{-}1$$

$$f(x) = \sum_{u=0}^{N-1} a(u)C(u)\cos\left[\frac{(2x+1)u\pi}{2N}\right] \qquad u=0,1,\ldots N\text{-}1$$

where

$$a(u) = \begin{cases} \sqrt{1/N} & \text{when } u = 0 \\ \sqrt{2/N} & \text{when } u = 1, 2, \cdots, N-1 \end{cases}$$

3.3 Other Separable Transforms

3.3.2 Discrete Cosine Transform (DCT)

Basis function matrix

$$C_N = \sqrt{\frac{2}{N}} \begin{array}{cc} & x \\ u & \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \cdots & 1/\sqrt{2} \\ \cos \pi/2N & \cos 3\pi/2N & \cdots & \cos(2N-1)\pi/2N \\ \cos 2\pi/2N & \cos 6\pi/2N & \cdots & \cos 2(2N-1)\pi/2N \\ \cos 3\pi/2N & \cos 9\pi/2N & \cdots & \cos 3(2N-1)\pi/2N \\ \cos 4\pi/2N & \cos 12\pi/2N & \cdots & \cos 4(2N-1)\pi/2N \\ \vdots & \vdots & \ddots & \vdots \\ \cos(N-1)\pi/2N & \cos 3(N-1)\pi/2N & \cdots & \cos(2N-1)(N-1)\pi/2N \end{bmatrix} \end{array}$$

3.3 Other Separable Transforms

3.3.2 Discrete Cosine Transform (DCT)

The 2-D DCT pair is given by the expression:

$$C(u,v) = a(u)a(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

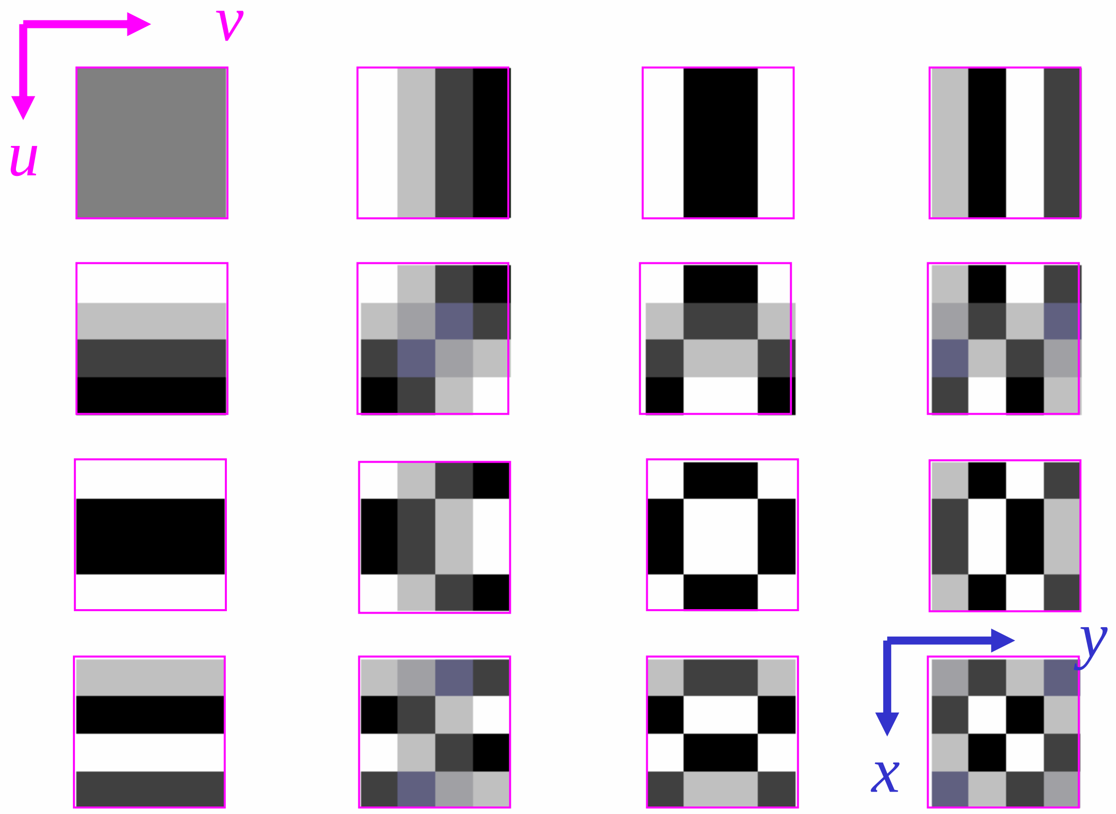
u,v=0,1,…N-1

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} a(u)a(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

x,y=0,1,…N-1

3.3 Other Separable Transforms

3.3.2 Discrete Cosine Transform (DCT)

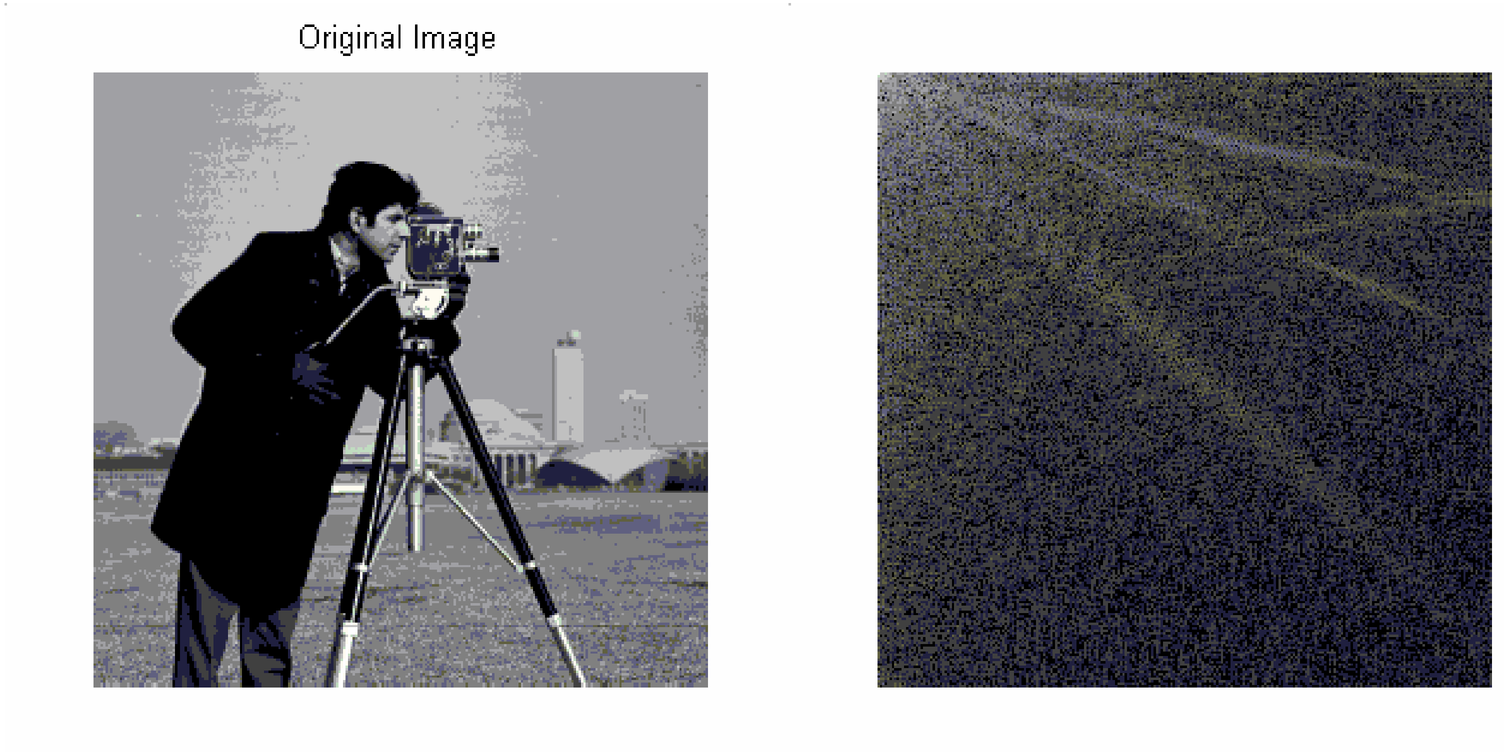


The 2D-DCT basis images for N=4

3.3 Other Separable Transforms

3.3.2 Discrete Cosine Transform (DCT)

An example of 2D-DCT



3.3 Other Separable Transforms

3.3.2 Walsh transform

When $N=2^n$, the kernel function is:

$$g(x,u) = \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

the discrete Walsh transform of a function *f(x)*, denote by *W(u)*, is:

$$W(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

Where $b_k(z)$ is the *k*th bit in the binary representation of *z*.
Eg: $n=3$, $z=6$ (110 in binary), we have that

$b_0(z)=0$, $b_1(z)=1$, and $b_2(z)=1$

3.3 Other Separable Transforms

3.3.2 Walsh transform： 1-D transform

The values of $g(x,u)$ are list in below

$u \backslash x$	0	1	2	3	4	5	6	7
0	+	+	+	+	+	+	+	+
1	+	+	+	+	-	-	-	-
2	+	+	-	-	+	+	-	-
3	+	+	-	-	-	-	+	+
4	+	-	+	-	+	-	+	-
5	+	-	+	-	-	+	-	+
6	+	-	-	+	+	-	-	+
7	+	-	-	+	-	+	+	-

3.3 Other Separable Transforms

3.3.2 Walsh transform： 1-D transform

Inverse kernel and transform:

$$h(x,u)=\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

$$f(x)=\sum_{u=0}^{N-1}W(u)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

3.3 Other Separable Transforms

3.3.2 Walsh transform： 1-D matrix expression N=8

$$
\begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix}
=
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}
\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}
$$

3.3 Other Separable Transforms

3.3.2 Walsh transform: 1-D transform

The values of *g(x,u)* are list in below for *N*=8

$u \backslash x$	0	1	2	3	4	5	6	7	Sequency of the row
0	+	+	+	+	+	+	+	+	0
1	+	+	+	+	-	-	-	-	1
2	+	+	-	-	+	+	-	-	3
3	+	+	-	-	-	-	+	+	2
4	+	-	+	-	+	-	+	-	7
5	+	-	+	-	-	+	-	+	6
6	+	-	-	+	+	-	-	+	4
7	+	-	-	+	-	+	+	-	5

3.3 Other Separable Transforms

3.3.2 Walsh transform： 1-D ordered Walsh transform

The values of *g(x,u)* are list in below for *N*=8

$u \backslash x$	0	1	2	3	4	5	6	7	Sequency of the row
0	+	+	+	+	+	+	+	+	0
1	+	+	+	+	-	-	-	-	1
2	+	+	-	-	-	-	+	+	2
3	+	+	-	-	+	+	-	-	3
4	+	-	-	+	+	-	-	+	4
5	+	-	-	+	-	+	+	-	5
6	+	-	+	-	-	+	-	+	6
7	+	-	+	-	-	+	+	-	7

3.3 Other Separable Transforms

3.3.2 Walsh transform： 2-D transform

The direct and inverse kernel functions are expressed as:

$$g(x,y,u,v)=\frac{1}{N}\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

$$h(x,y,u,v)=\frac{1}{N}\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

And the direct and inverse transforms are:

$$W(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}W(u,v)\prod_{i=0}^{n-1}(-1)^{[b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)]}$$

3.3 Other Separable Transforms

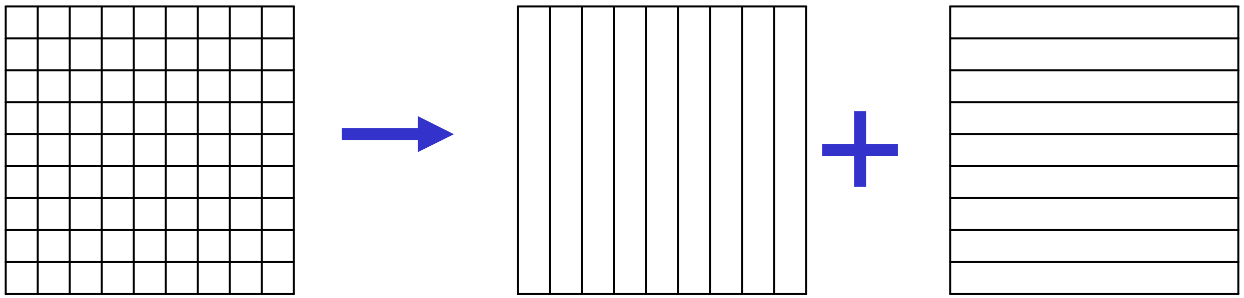
3.3.2 Walsh transform： 2-D transform

The direct and inverse kernel functions are

Separable and *Symmetric*

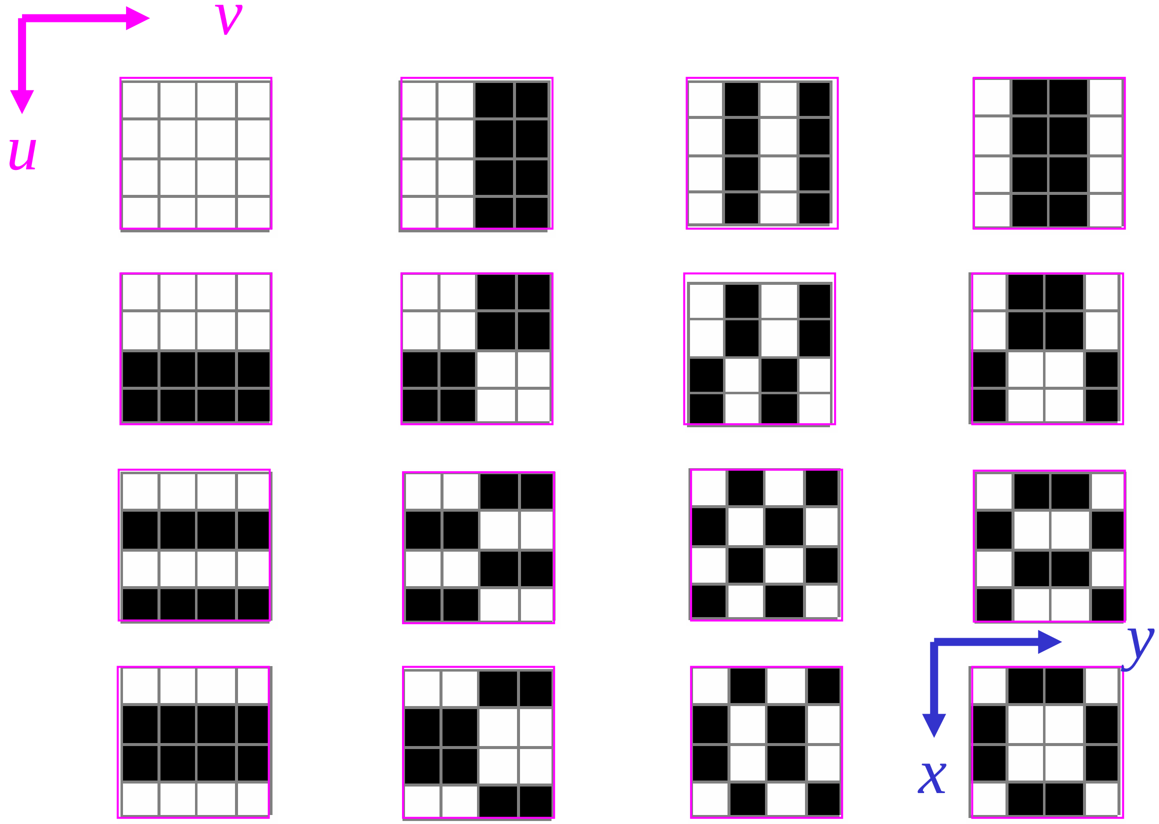
$$g(x,y,u,v) = g_1(x,u)g_1(y,v) = h_1(x,u)h_1(y,v)$$

So it can be implemented in two steps



3.3 Other Separable Transforms

3.3.2 Walsh transform： 2-D transform



The Walsh transform basis images for *N*=4

3.3 Other Separable Transforms

3.3.2 Hadamard transform: 1-D transform

When $N=2^n$, the kernel function is:

$$g(x,u)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

Where the summation in the exponent is performed in modulo 2

1-D Hadamard transform of a function *f(x)*, denote by H(*u*), is:

$$H(u)=\frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)}$$

3.3 Other Separable Transforms

3.3.2 Hadamard transform: 1-D inverse transform

Inverse kernel and transform:

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

$$f(x) = \sum_{u=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(x) b_i(u)}$$

3.3 Other Separable Transforms

3.3.4 hadamard transform： 1-D matrix expression N=8

$$
\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \end{bmatrix}
=
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{bmatrix}
\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}
$$

3.3 Other Separable Transforms

3.3.2 Hadamard transform: 1-D transform

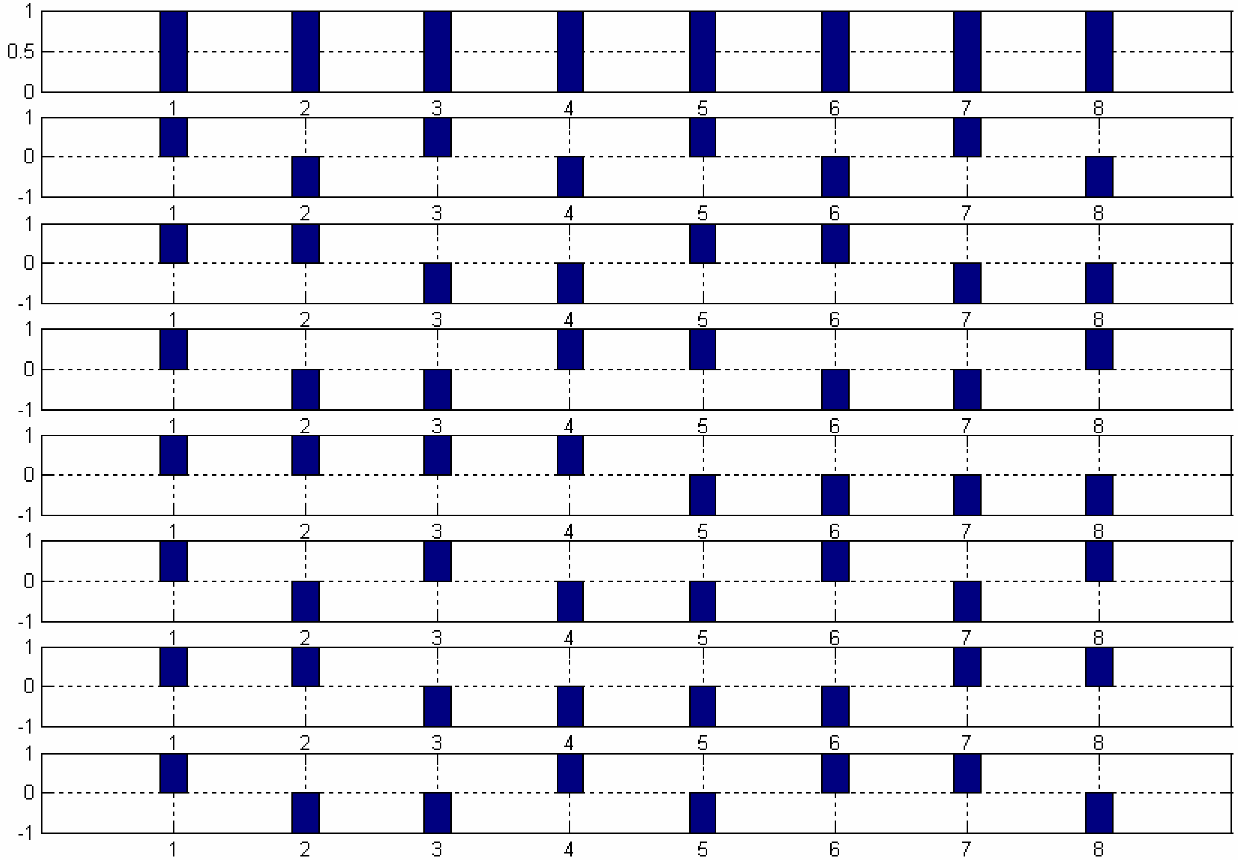
The values of $g(x,u)$ are list in below for N=8

x →, u ↓

$u \backslash x$	0	1	2	3	4	5	6	7	Sequency of the row
0	+	+	+	+	+	+	+	+	0
1	+	-	+	-	+	-	+	-	7
2	+	+	-	-	+	+	-	-	3
3	+	-	-	+	+	-	-	+	4
4	+	+	+	+	-	-	_	-	1
5	+	-	+	-	-	+	-	+	6
6	+	+	-	-	-	-	+	+	2
7	+	-	-	+	-	+	+	-	5

3.3 Other Separable Transforms

3.3.2 Hadamard transform： 1-D transform



The 1-D Hadamard transform basis function for N=8

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 1-D transform

Another way for generating kernel matrix

For the two-by-two case, the kernel matrix is

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

And for successively larger N, these can be generated from The block matrix from

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} = H_2 \otimes H_N$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform: 1-D transform

Another way for generating kernel matrix

For examples

$$H_4 = \frac{1}{\sqrt{4}}\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right]$$

$$H_8 = \frac{1}{\sqrt{8}}\left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}\right]$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 1-D ordered Hadamard transform

Lets kernel function:

$$g(x,u) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)p_i(u)}$$

where

$$\left.\begin{aligned} p_0(u) &= b_{n-1}(u) \\ p_1(u) &= b_{n-1}(u) + b_{n-2}(u) \\ &\vdots \\ p_{n-1}(u) &= b_1(u) + b_0(u) \end{aligned}\right\}$$

And inverse kernel function:

$$h(x,u) = (-1)^{\sum_{i=0}^{n-1} b_i(x)p_i(u)}$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform: 1-D ordered Hadamard transform

The values of $g(x,u)$ are list in below for N=8

Same with the Ordered Walsh transform

Sequency of the row

$u \backslash x$	0	1	2	3	4	5	6	7	
0	+	+	+	+	+	+	+	+	0
1	+	+	+	+	-	-	-	-	1
2	+	+	-	-	-	-	+	+	2
3	+	+	-	-	+	+	-	-	3
4	+	-	-	+	+	-	_	+	4
5	+	-	-	+	-	+	+	-	5
6	+	-	+	-	-	+	-	+	6
7	+	-	+	-	-	+	+	-	7

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 1-D ordered Hadamard transform

Then the ordered Hadamard transform pair is

$$H(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)p_i(u)}$$

$$f(x) = \sum_{u=0}^{N-1} H(u)(-1)^{\sum_{i=0}^{n-1} b_i(x)p_i(u)}$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 2-D transform

The kernel functions of 2-D Hadamard are:

$$h(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

$$g(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

Both the direct and inverse kernel function are *Separable* and *Symmetric,* because

$$g(x,y,u,v)=g_1(x,u)g_1(y,v)=h_1(x,u)h_1(y,v)$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 2-D transform

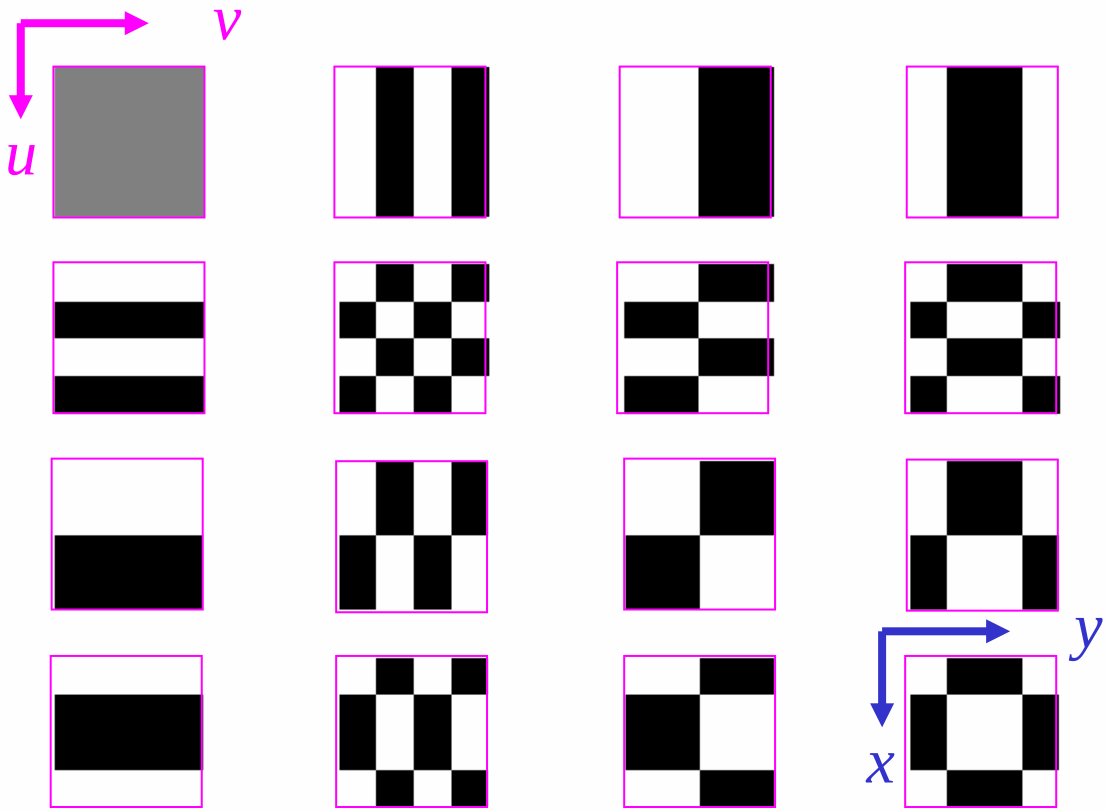
And the 2-D Hadamard transform is defined as:

$$H(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

$$f(x,y) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} H(u,v)(-1)^{\sum_{i=0}^{n-1}[b_i(x)b_i(u)+b_i(y)b_i(v)]}$$

3.3 Other Separable Transforms

3.3.2 Hadamard transform： 2-D transform



The Hadamard transform basis images for N=4

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 2-D ordered transform

The direct and inverse kernel functions of 2-D ordered Hadamard are same as:

$$g(x,y,u,v) = h(x,y,u,v) = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]}$$

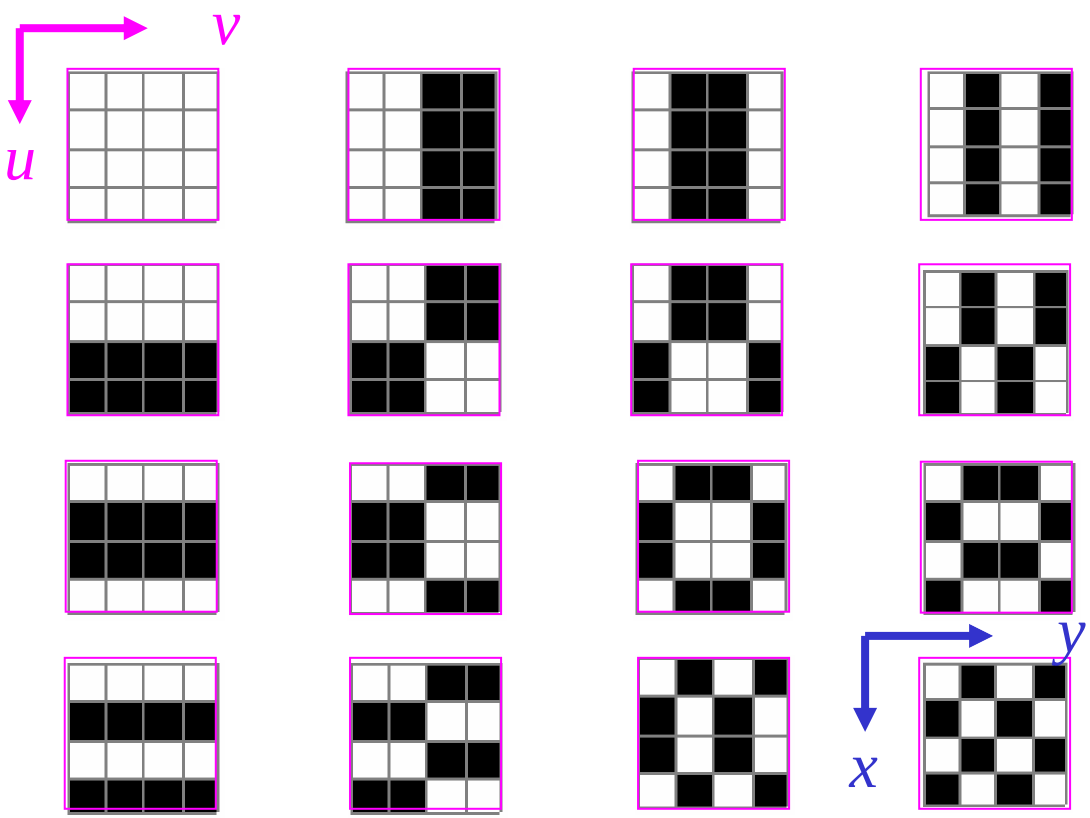
And the 2-D ordered Hadamard transform is defined as:

$$H(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]}$$

$$f(x,y) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} H(u,v)(-1)^{\sum_{i=0}^{n-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]}$$

3.3 Other Separable Transforms

3.3.4 Hadamard transform： 2-D ordered transform



Basis image of 2-D ordered Hadamard transform

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

1. Haar function

The Haar functions $h_k(z)$ are defined on the interval [0,1], and $k=0,1\ldots N-1$, $N=2^n$. Let theInteger $0 \le k \le N-1$ be specified (uniquely) by two other integers, p and q, as

$$k = 2^p + q - 1$$

Where 2^p is the largest power of 2 such that $2^p \le k$ and q-1 is The remainder

For example,

$k=1=2^0+1-1,$ $\Rightarrow$ $p=0, q=1$

$k=23=2^4+8-1,$ $\Rightarrow$ $p=4, q=8$

$k=100=2^6+37-1,$ $\Rightarrow$ $p=6, q=37$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

1. Haar function

The Haar functions are defined by

$$h_0(z) = h_{00}(z) = 1/\sqrt{N} \qquad z \in [0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \le z < \frac{q-1/2}{2^p} \\ -2^{p/2} & \frac{q-1/2}{2^p} \le z < \frac{q}{2^p} \\ 0 & \text{others} \end{cases}$$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

2. Haar transform matrix

$$H = \begin{bmatrix} h_0(0/N) & h_0(1/N) & \cdots & h_0(N-1/N) \\ h_1(0/N) & h_1(1/N) & \cdots & h_1(N-1/N) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1}(0/N) & h_{N-1}(1/N) & \ldots & h_{N-1}(N-1/N) \end{bmatrix}$$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

2. Haar transform matrix

For example: N=2

$$H_2 = \begin{bmatrix} h_0(0/2) & h_0(1/2) \\ h_1(0/2) & h_1(1/2) \end{bmatrix}$$

k=0	*p=0, q=0*
k=1	*p=0, q=1*

$$h_0(0/2) = h_{00}(0/2) = \frac{1}{\sqrt{2}}$$

$$h_0(1/2) = h_{00}(1/2) = \frac{1}{\sqrt{2}}$$

$$h_1(0/2) = h_{01}(0/2) = \frac{1}{\sqrt{2}}$$

$$h_1(1/2) = h_{01}(1/2) = -\frac{1}{\sqrt{2}}$$

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

2. Haar transform matrix

For example: N=4

$$H_4 = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

For example: N=8

$$H_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

2. Haar transform matrix

$$C_N = \sqrt{\frac{2}{N}}\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & \cdots & 1/\sqrt{2} \\ \cos \pi/2N & \cos 3\pi/2N & \cdots & \cos(2N-1)\pi/2N \\ \cos 2\pi/2N & \cos 6\pi/2N & \cdots & \cos 2(2N-1)\pi/2N \\ \cos 3\pi/2N & \cos 9\pi/2N & \cdots & \cos 3(2N-1)\pi/2N \\ \cos 4\pi/2N & \cos 12\pi/2N & \cdots & \cos 4(2N-1)\pi/2N \\ \vdots & \vdots & \ddots & \vdots \\ \cos(N-1)\pi/2N & \cos 3(N-1)\pi/2N & \cdots & \cos(2N-1)(N-1)\pi/2N \end{bmatrix}$$

$$w_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \qquad H_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

3.3 Other Separable Transforms

3.3.5 Haar transform： definitions

3. Haar transform

Direct transform: $G = HF$

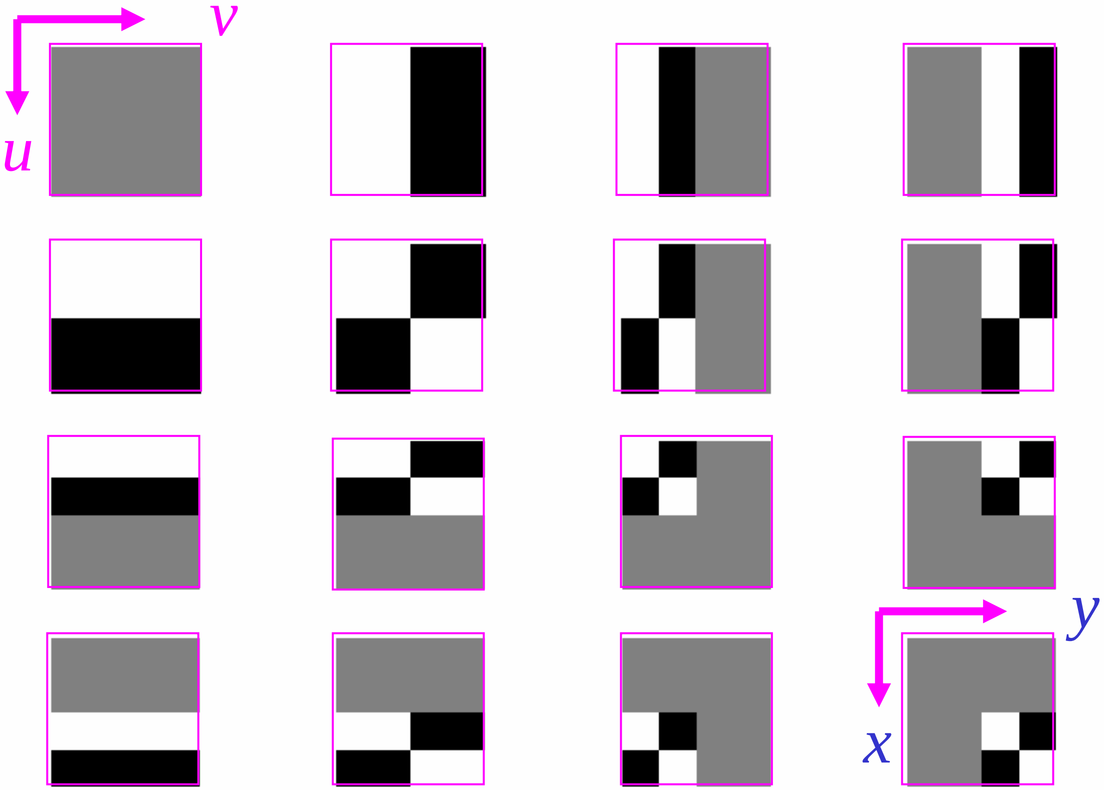
Inverse transform: $F = H^{-1}G$

$$
\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \end{bmatrix}
= \frac{1}{\sqrt{8}}
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
\sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\
2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & -2
\end{bmatrix}
\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}
$$

3.3 Other Separable Transforms

3.3.5 Haar transform： properties

3. Haar transform



The Haar transform basis images for N=4

3.3 Other Separable Transforms

3.3.5 Haar transform： example

Edge detecting ability of the Haar transform

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.83 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

image **Haar transform**

3.3 Other Separable Transforms

3.3.6 Slant transform： definitions

Unitary kernel matrix starting with:
$$S_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

And iterating it according to the schema:

$$S_N = \frac{1}{\sqrt{2}}\left[\begin{array}{cc|c|c|cc|c|c} 1 & 0 & \vdots & 0 & \vdots & 1 & 0 & \vdots & 0 \\ a_N & b_N & & & & -a_N & b_N & & \\ \multicolumn{2}{c}{\cdots} & \cdots & \cdots & \cdots & \multicolumn{2}{c}{\cdots} & \cdots & \cdots \\ \multicolumn{2}{c}{0} & \vdots & I_{(N/2)-2} & \vdots & \multicolumn{2}{c}{0} & \vdots & I_{(N/2)-2} \\ \multicolumn{2}{c}{\cdots} & \cdots & \cdots & \cdots & \multicolumn{2}{c}{\cdots} & \cdots & \cdots \\ 0 & 1 & \vdots & 0 & \vdots & 0 & -1 & \vdots & 0 \\ -b_N & a_N & & & & b_N & a_N & & \\ \multicolumn{2}{c}{\cdots} & \cdots & \cdots & \cdots & \multicolumn{2}{c}{\cdots} & \cdots & \cdots \\ \multicolumn{2}{c}{0} & \vdots & I_{(N/2)-2} & \vdots & \multicolumn{2}{c}{0} & \vdots & -I_{(N/2)-2} \end{array}\right]\left[\begin{array}{ccc} S_{N/2} & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & S_{N/2} \end{array}\right]$$

3.3 Other Separable Transforms

3.3.6 Slant transform： definitions

Where *I* is the identity matrix of order *N/2-2* and

$$a_N = \left[\frac{3N^2}{4(N^2-1)} \right]^{1/2} \qquad b_N = \left[\frac{N^2-4}{4(N^2-1)} \right]^{1/2}$$

For example: N=4

$$S_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3/\sqrt{5} & 1/\sqrt{5} & -1/\sqrt{5} & -3/\sqrt{5} \\ 1 & -1 & -1 & -1 \\ 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} \end{bmatrix}$$

3.3 Other Separable Transforms

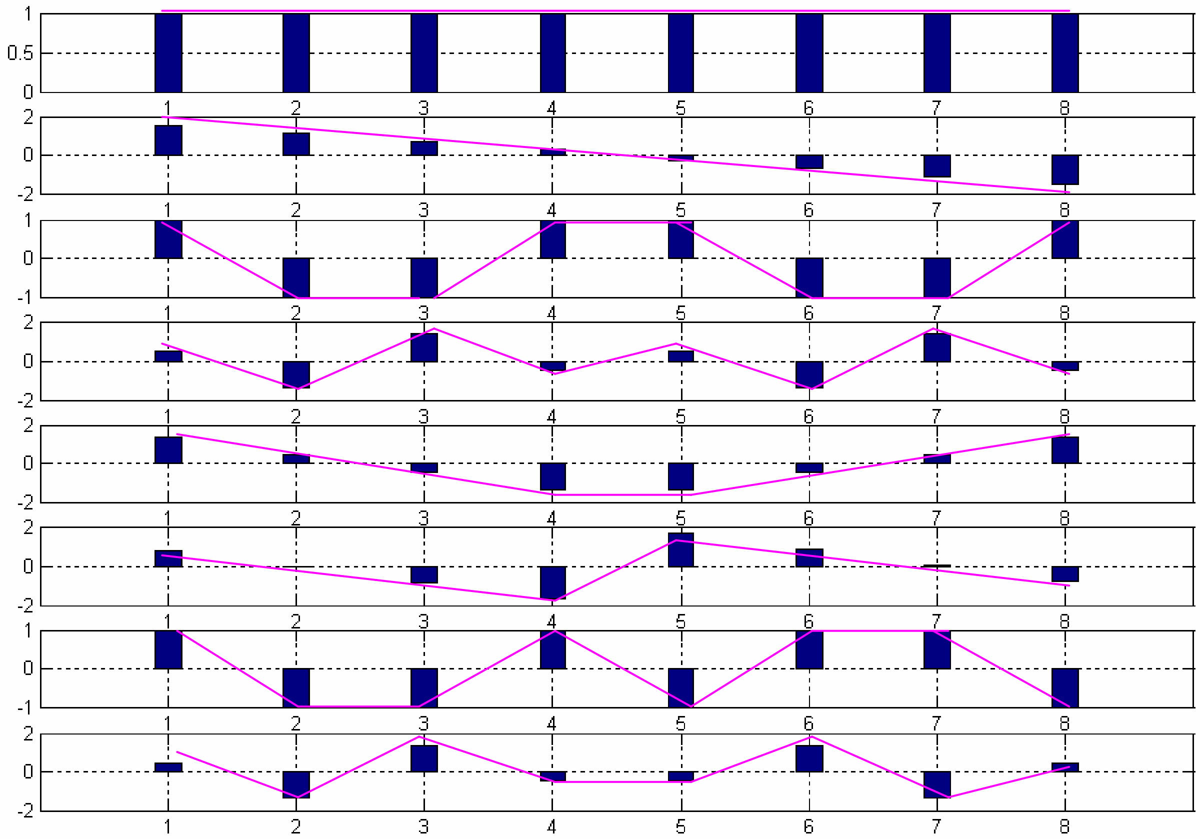
3.3.6 Slant transform： definitions

For example: *N=8*

$$S_8 = \frac{1}{\sqrt{8}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7/\sqrt{21} & 5/\sqrt{21} & 3/\sqrt{21} & 1/\sqrt{21} & -1/\sqrt{21} & -3/\sqrt{21} & -5/\sqrt{21} & -7/\sqrt{21} \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} & 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} \\ 3/\sqrt{5} & 1/\sqrt{5} & -1/\sqrt{5} & -3/\sqrt{5} & -3/\sqrt{5} & -1/\sqrt{5} & 1/\sqrt{5} & 3/\sqrt{5} \\ 7/\sqrt{105} & -1/\sqrt{105} & -9/\sqrt{105} & -17/\sqrt{105} & 17/\sqrt{105} & 9/\sqrt{105} & 1/\sqrt{105} & -7/\sqrt{105} \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1/\sqrt{5} & -3/\sqrt{5} & 3/\sqrt{5} & -1/\sqrt{5} & -1/\sqrt{5} & 3/\sqrt{5} & -3/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}$$

3.3 Other Separable Transforms

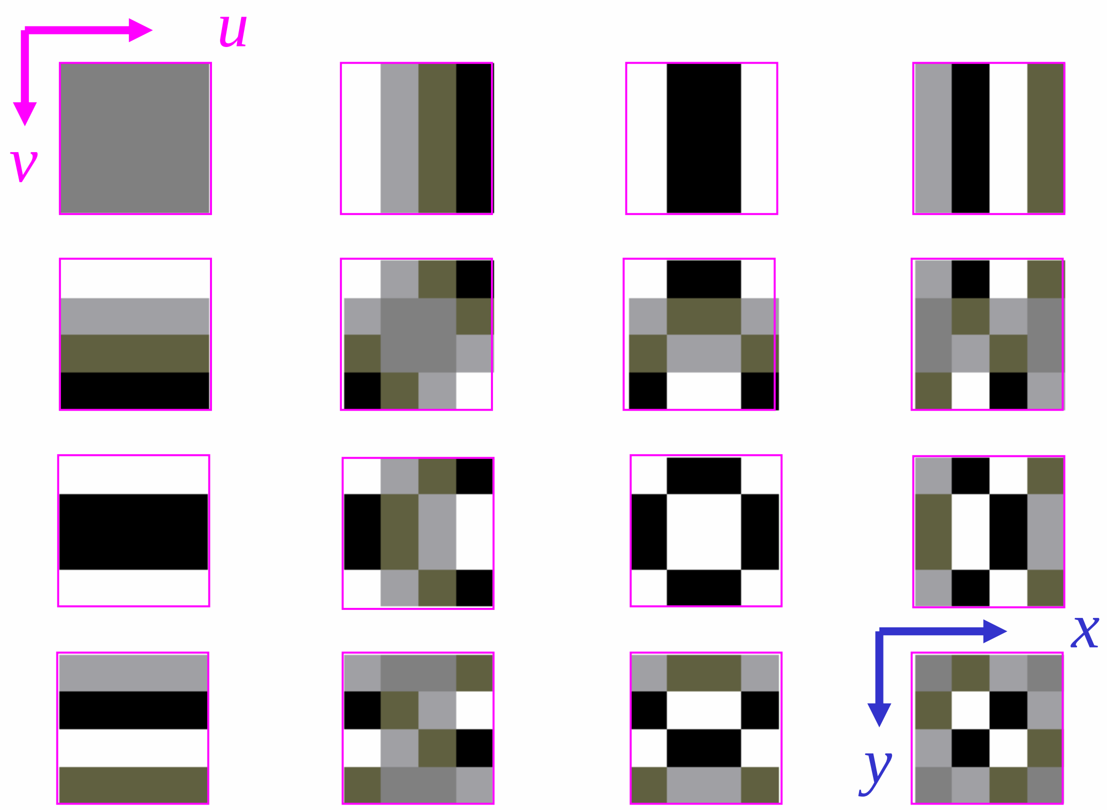
3.3.6 Slant transform： 1-D basis functions



The 1-D Slant transform basis functions for N=8

3.3 Other Separable Transforms

3.3.6 Slant transform： 2-D basis images



The 2-D Slant transform basis functions for N=4

3.4 Hotelling Transforms

3.3.7 Hotelling transform： definitions

KLT: Karhunen-Loeve Transfrom
PCA:Principal Components Analysis

The *kth* image in a image set can be expressed as a vector:

…

$$x^k = \begin{bmatrix} x_0^k & x_1^k & \cdots & x_{N-1}^k \end{bmatrix}^T \quad k=0,1,\ldots M\text{-}1$$

The covariance matrix of the **x** vector is defined as

$$C_x = E\{(x - m_x)(x - m_x)^T\}$$

where

$$m_x = E\{x\}$$

is the mean vector, *E* is the expected value

3.4 Hotelling Transforms

3.3.7 Hotelling transform： definitions

They can be approximated from the samples by using the relations

$$m_x = \frac{1}{M}\sum_{k=0}^{M-1} x^k$$

and

$$C_x = \frac{1}{M}\sum_{k=0}^{M-1}(x^k - m_x)(x^k - m_x)^T$$
$$= \frac{1}{M}\sum_{k=0}^{M-1} x^k x^{kT} - m_x m_x^T$$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： definitions

let $|C_x - \lambda I| = 0$ → Calculated N eigenvalues and arranged as $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1}$

let $[C_x - \lambda_i I] T_i = 0$ → Calculated N eigenvectors T_i and arranged as

$$A = \begin{bmatrix} T_0^T \\ T_1^T \\ \vdots \\ T_{N-1}^T \end{bmatrix}$$

→ Hotelling transform $\boxed{Y = A(X - m_x)}$

Inverse Hotelling transform $\boxed{X = A^T Y + m_x}$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： properties

Relationship between the eigenvaluse and eigenvectors:

$$[C_x - \lambda_i I]T_i = 0 \Rightarrow C_x T_i = \lambda_i T_i$$

$$A = \begin{bmatrix} T_0^T \\ T_1^T \\ \vdots \\ T_{N-1}^T \end{bmatrix} \Rightarrow \boxed{C_x A^T = A^T \wedge}$$

where $\wedge = \begin{bmatrix} \lambda_0 & & & 0 \\ & \lambda_1 & & \\ & & \ddots & \\ 0 & & & \lambda_{N-1} \end{bmatrix}$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： properties

mean vector of $\boldsymbol{y}$ $m_y = E\{y\} = E\{(Ax - Am_x)\} = AE\{x\} - Am_x$

$$\boxed{m_y = 0}$$

The covariance matrix of the ***Y*** vector is given by

$$C_y = E\{(Y - m_y)(Y - m_y)^T\}$$

$$= E\left\{(AX - Am_x)(AX - Am_x)^T\right\}$$

$$= E\left\{A(X - m_x)(X - m_x)^T A^T\right\}$$

$$= AE\left\{(X - m_x)(X - m_x)^T\right\} A^T$$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： properties

$$AA^T = I \Rightarrow \boxed{\begin{aligned} C_y &= AC_xA^T \\ &= AA^T \wedge \\ &= \wedge \end{aligned}}$$

$$\boxed{C_xA^T = A^T \wedge}$$

C_y is a diagonal matrix with elements equal to the eigenvalues of C_x, that is

$$C_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix}$$

That's means elements of Y are *uncorrelated*

3.4 Hotelling Transforms

3.3.7 Hotelling transform： inverse transform

Since Hotelling transform is *orthogonal* , so

$$A^{-1} = A^T \qquad \text{and} \qquad X = A^T Y + m_x$$

If we form A from K eigenvectors corresponding to the largest eigenvalues as A_K, the recovered vector will be

$$\hat{X} = A_K^T Y + m_x$$

It can be shown that the mean square error, e_{ms}, between X and $\hat{X}$ is given by the expression

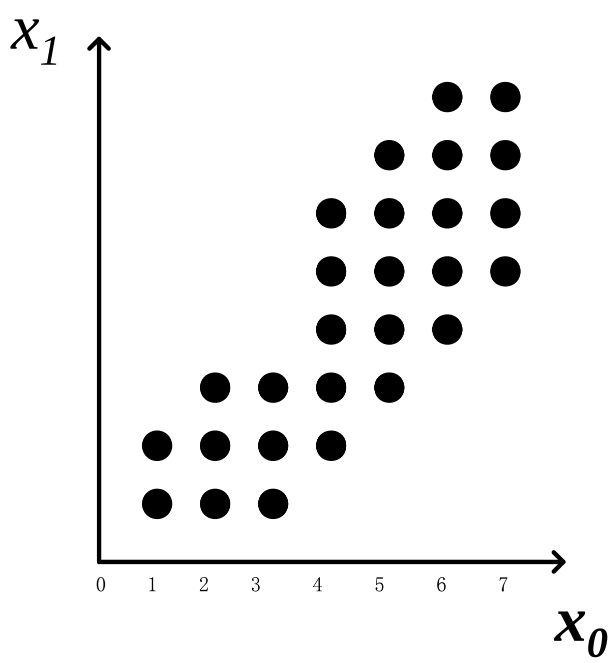
$$e_{ms} = \sum_{j=0}^{N-1} \lambda_j - \sum_{j=0}^{k-1} \lambda_j = \sum_{j=K}^{N-1} \lambda_j$$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： example

Given the samples of 2-dimension vectors shown as below, calculate its Hotelling transform. $N=2$, $M=27$

(1,1) (1,2) (2,1) (2,2) (2,3)
(3,1) (3,2) (3,3) (4,2) (4,3)
(4,4) (4,5) (4,6) (5,3) (5,4)
(5,5) (5,6) (5,7) (6,4) (6,5)
(6,6) (6,7) (6.8) (7,5) (7,6)
(7,7) (7,8)



3.4 Hotelling Transforms

3.3.7 Hotelling transform： example

Let

$$x^k = \begin{bmatrix} x_0^k & x_1^k \end{bmatrix}^T \qquad k=0,1,\ldots 26$$

$$m_x = \frac{1}{27}\sum_{k=0}^{25} x_k = \begin{bmatrix} 4.444 & 4.2963 \end{bmatrix}$$

$$C_x = \frac{1}{27}\sum_{k=0}^{25}(x_i - m_x)(x_i - m_x)^T$$
$$= \begin{bmatrix} 3.4103 & 3.2479 \\ 3.2479 & 4.7550 \end{bmatrix}$$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： example

let $|C_x - \lambda I| = 0$ $\Rightarrow$ $\boxed{\begin{vmatrix} 3.4103-\lambda & 3.2479 \\ 3.2479 & 4.7550-\lambda \end{vmatrix} = 0}$

$\boxed{\lambda_0 = 7.3993}$ $\boxed{\lambda_1 = 0.7659}$

let $[C_x - \lambda_i I]T_i = 0$ $\Rightarrow$ $T_0 = \begin{bmatrix} 0.6314 \\ 0.7755 \end{bmatrix}$ $T_1 = \begin{bmatrix} -0.7755 \\ 0.6314 \end{bmatrix}$

$$\boxed{A = \begin{bmatrix} 0.6314 & 0.7755 \\ -0.7755 & 0.6314 \end{bmatrix}}$$

$$\boxed{y = A(x - m_x)}$$

3.4 Hotelling Transforms

3.3.7 Hotelling transform： example

y =(-4.7309 0.5899) (-3.9555 1.2212) (-4.0996 -0.1856) (-3.3241 0.4458)
(-2.5486 1.0771) (-3.4682 -0.9611) (-2.6927 -0.3297) (-1.9172 0.3017)
(-2.0613 -1.1052) (-1.2859 -0.4738) (-0.5104 0.1576) (0.2651 0.7890)
(1.0406 1.4203) (-0.6545 -1.2493) (0.1210 -0.6179) (0.8965 0.0135)
(1.6719 0.6449) (2.4474 1.2762) (0.7524 -1.3934) (1.5279 -0.7620
(2.3033 -0.1306) (3.0788 0.5008) (3.8543 1.1322) (2.1592 -1.5375)
(2.9347 -0.9061) (3.7102 -0.2747) (4.4857 0.3567)

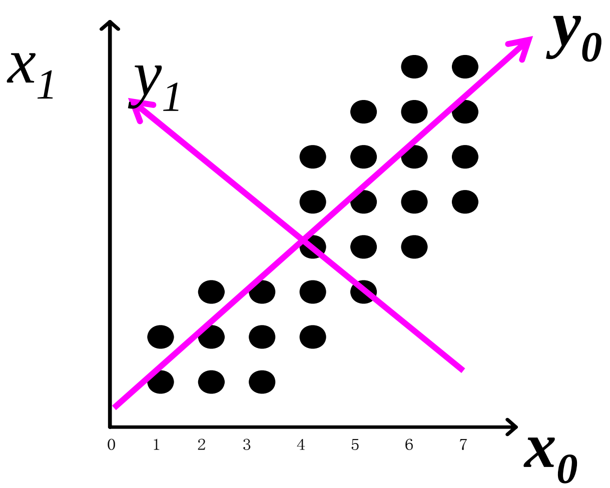
my = 1.0e-015 *

-0.6908 -0.1069

cy =

7.3993 0.0000

0.0000 0.7659



corry =

1.0000 0.0000

0.0000 1.0000

Summary

Basic function

Sinusoidal transforms

(a) Discrete Fourier Transform $\quad e^{i\theta} = \cos\theta + i\sin\theta$

(b) Discrete Cosine Transform $\quad \cos\theta$

(c) Discrete Sine Transform $\quad \sin\theta$

(d) Hartly Transform $\quad \cos\theta + \sin\theta$

Summary

Rectangular wave transforms

(a) Hadamard Transform

(b) Walsh Transform

(c) Slant Transform

(d) Haar Transform

Summary

Eigenvector-based transforms

(a) Hotelling Transform (K-LT)

(b) SVD Transform

Chapter4 Image Enhancement

- 4.1 General introduction and Classification
- 4.2 Enhancement by Spatial Transforming(contrast enhancement)
- 4.3 Enhancement by Spatial Filtering (image smoothing)
- 4.4 Enhancement by Frequency Filtering (image sharpening)
- 4.5 Color Enhancement
- Summary

4.1 General Introduction and Classification

4.1.1 Purposes

- improve the visual effects
- easy to edge extracting

4.1.2 Methods

- spatial domain: point operations, local operations
- frequency domain: DFT $\Rightarrow$ Filter $\Rightarrow$ IDFT

4.1 General Introduction and Classification

4.1.3 contents

- contrast enhancement: linear transform
  non-linear transform,
  histogram equalization
  histogram matching
  local enhancement
- image smoothing: averaging mask,
  order-statistics filter
  lowpass filter.
- image sharpening: derivatives,
  highpass filter
- color image enhancement: pseudo color processing,
  full color processing

4.2 Contrast Enhancement

4.2.1. Introduction: General expression

f(x,y) → *g(x,y)*

Local operator

$$g(x,y) = T[f(x \pm i, y \pm j)] \quad i, j = 0, \pm 1, \pm 2 \cdots$$

where T is a operator on f, defined over some neighborhood of (x,y)

when the neighborhood is of size 1*1,(a single pixel), we define

$$r = f(x,y) \qquad s = g(x,y)$$

Point operator

and $$s = T(r)$$

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histogram gives an estimate of the probability of the occurrence of gray levels

$$p(s_k) = n_k / n \qquad\qquad k = 0, 1, \cdots, L-1$$

Where s_k is the the kth gray level and the n_k is the number of Pixels in the image having gray level s_k

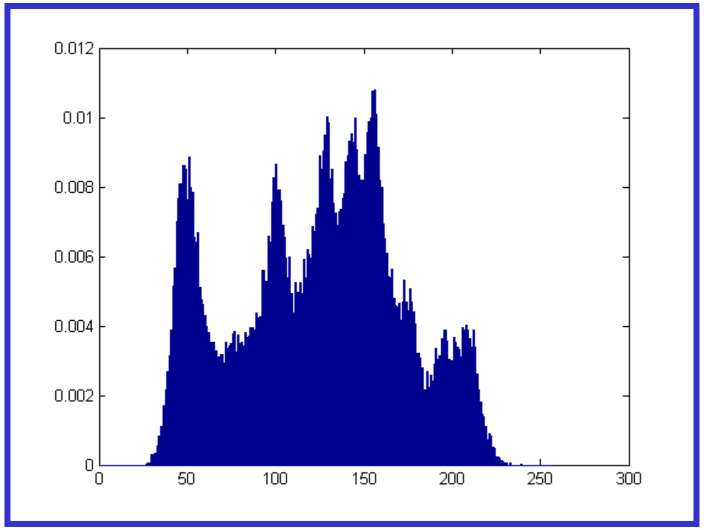
apparently

$$\sum_{k=0}^{L-1} p(s_k) = 1$$

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Horizontal axis: gray level values
Vertical axis: probability of gray level values



4.2 Contrast Enhancement

4.2.1. Introduction: Classification

- Direct gray level transformations
- Histogram processing
- Operations among images

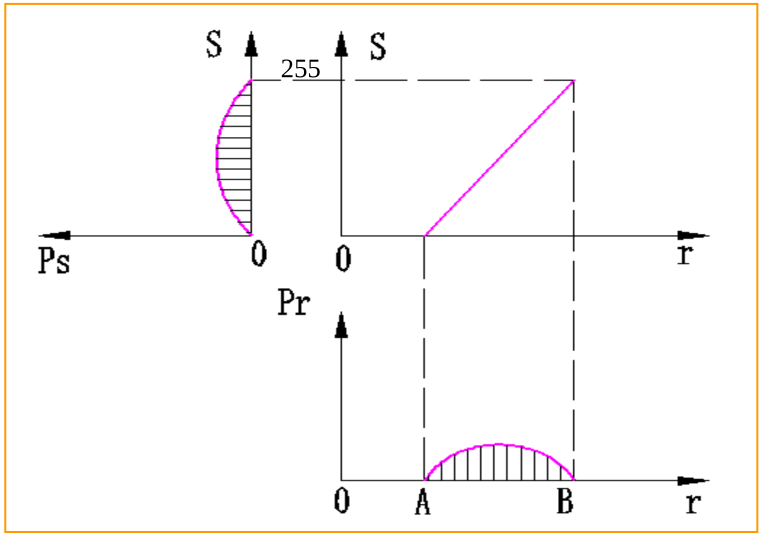
4.2 Contrast Enhancement

4.2.2 Direct gray level transformations: Linear transformations

Expression: $s = T(r) = ar + b \quad r \in [A, B] \quad s \in [C, D]$

Formulation: $$s = \frac{D - C}{B - A} r + \frac{BC - AD}{B - A}$$

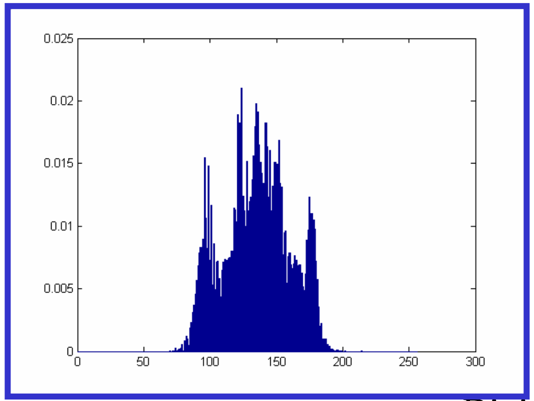
Example:
C=0
D=255

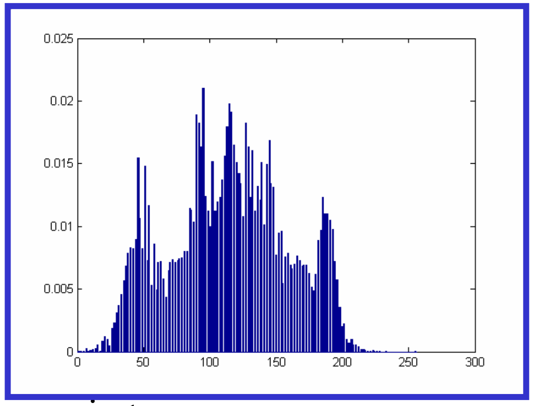


4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Linear transformations

example: $A= 69, B=213, C=0, D=255$ → $s = 1.7r - 122.2$



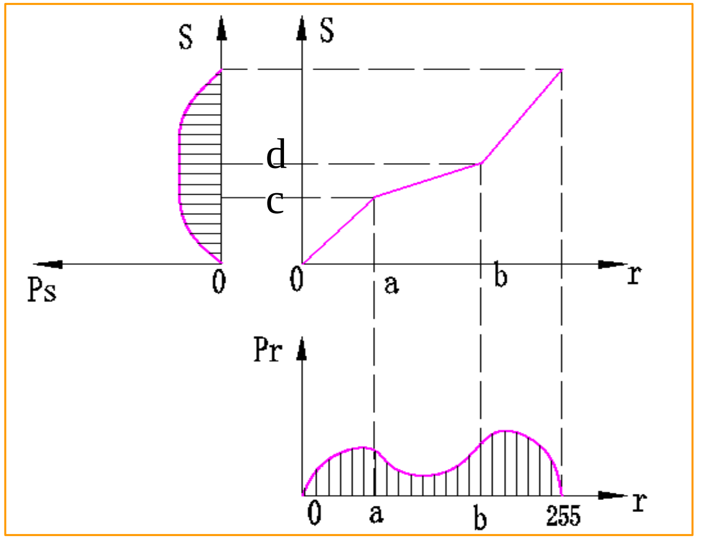


4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Piecewise-linear transformations

Formulation:

$$s = \begin{cases} \dfrac{c}{a} r & r \in [0, a) \\ \dfrac{d-c}{b-a} r + c & r \in [a, b) \\ \dfrac{255-d}{255-b}(r-b)+d & r \in [b, 255) \end{cases}$$



4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Log transformations

Formulation: $$s = c\log(1+r)$$

c is a constant and it is assumed that $r \geq 0$

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Power transformations

Formulation: $$s = cr^{\gamma}$$

Where c and γ are positive constants.

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

If a transform has the form:

$$s = T(r) \qquad 0 \le r \le 1$$

We hope the Probability Density Function (PDF) of s is

$$p_s(s) = 1$$

Inverse transform exist

and, if $r_1 < r_2$ then $s_1 < s_2$

$T(r)$ satisfies the following conditions:

(a) is single-valued and monotonically in the interval $0 \le r \le 1$

(b) $0 \le s \le 1$ for $0 \le r \le 1$

s has the same range as the r

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Continuous condition

$$s = T(r) \Rightarrow p_s(s) = p_r(r)\frac{dr}{ds}$$

$$\left.\begin{aligned} \frac{ds}{dr} &= \frac{p_r(r)}{p_s(s)} \\ p_s(s) &= 1 \end{aligned}\right\} \Rightarrow s = \int_0^r p_r(x)dx$$

namely $T(r) = \int_0^r p_r(x)dx$

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Discrete condition

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0,1,2,\ldots,L-1$$

In practice

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)

Histogram equalization： $r \longrightarrow s$

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \qquad k = 0,1,2\cdots L-1$$

Histogram equalization： $z \longrightarrow v$

$$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) \qquad k = 0,1,2\cdots L-1$$

let $\quad v_k = s_k \quad$ then $\quad r \longrightarrow z$

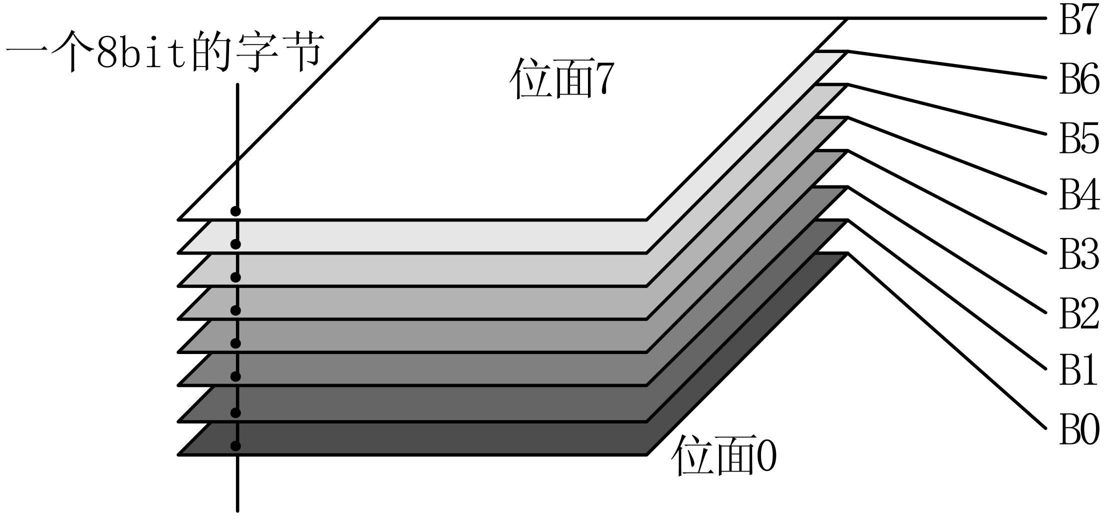
$$z_k = G^{-1}(v_k) = G^{-1}(s_k) = G^{-1}(T(r_k))$$

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Bit-plane slicing

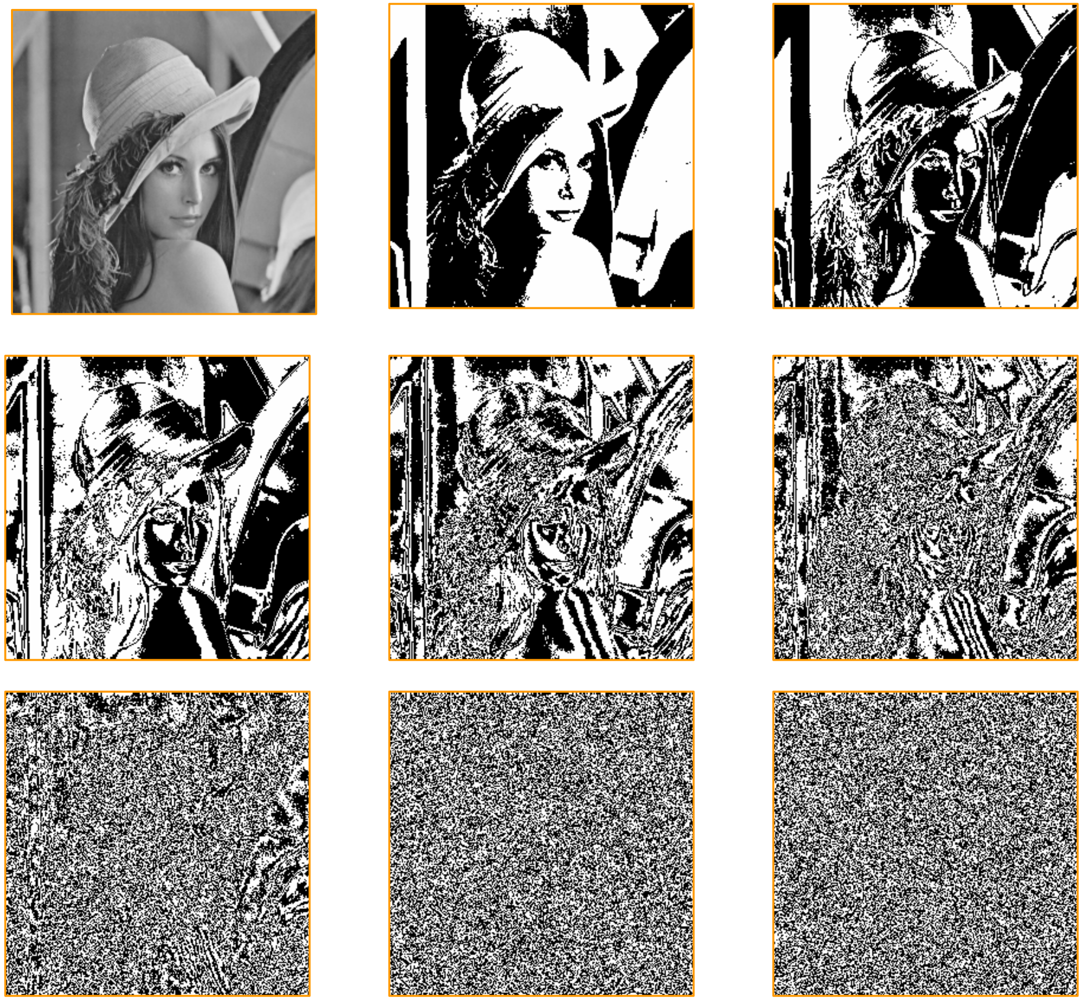
eg:

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
1	1	1	1	1	1	1	1	255
1	1	1	1	1	1	1	0	254
1	0	0	0	0	0	0	1	129



4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Bit-plane slicing



	b_7	b_6
b_5	b_4	b_3
b_2	b_1	b_0

4.2 Contrast Enhancement

4.2.4 Operations among Images: Image averaging

Original image： $f(x,y)$ noise: $\eta(x,y)$

$$g(x,y) = f(x,y) + \eta(x,y)$$

Averaging:

$$\overline{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

Expected value:

$$E\{\overline{g}(x,y)\} = f(x,y)$$

Variances:

$$\sigma^2_{\overline{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

Standard deviation:

$$\sigma_{\overline{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$

4.3 Image Smoothing

- Introduction
- Smoothing linear filters
- Order- statistic filters
- Low-pass filter in frequency domain

4.3 Image Smoothing

4.3.1 Introduction: Smoothing filters

(1) Smoothing linear filters: neighbor averaging
out range pixel smoothing
Maximum homogeneity smoothing

(2)Order- statistic filters: Max filters
Min filters
Midpoint filters
Median filters
Alpha-trimmed mean filter

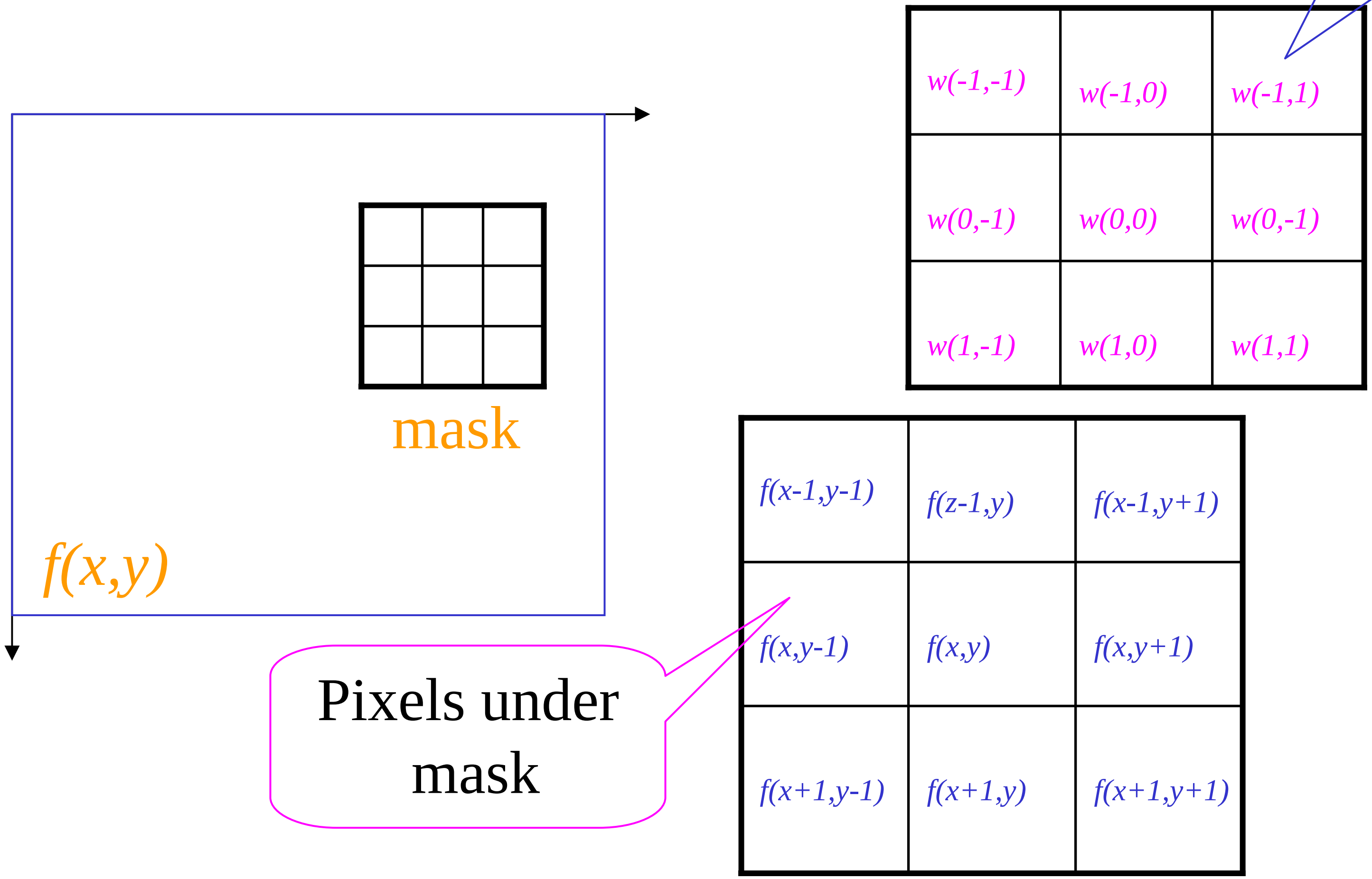
(3)Low-pass filters: Idea low-pass filter (ILPF)
Butterworth low-pass filter (BLPF)
Gaussian low-pass filter (GLPF)

4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Spatial filter: mask, kernel, template, window

Mask coefficients



4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Spatial filter: operating steps: (page 83)

(1) Moving the filter mask from point to point in an image

(2) Multiplying the filter coefficient and the corresponding image pixels

(3) Adding all the products

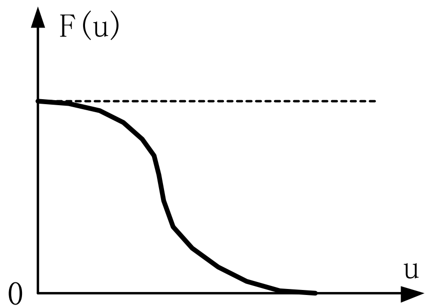
(4) The sum is the response of the filter at a given point

$$g(x,y)=\sum_{i=-a}^{a}\sum_{j=-b}^{b} w(i,j)\,f(x+i,y+j)$$

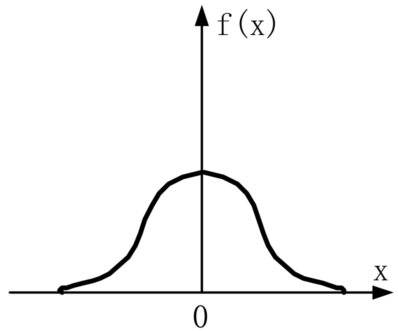
4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Spatial filter: the sign of filter coefficients



(a)



(b)

positive

4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

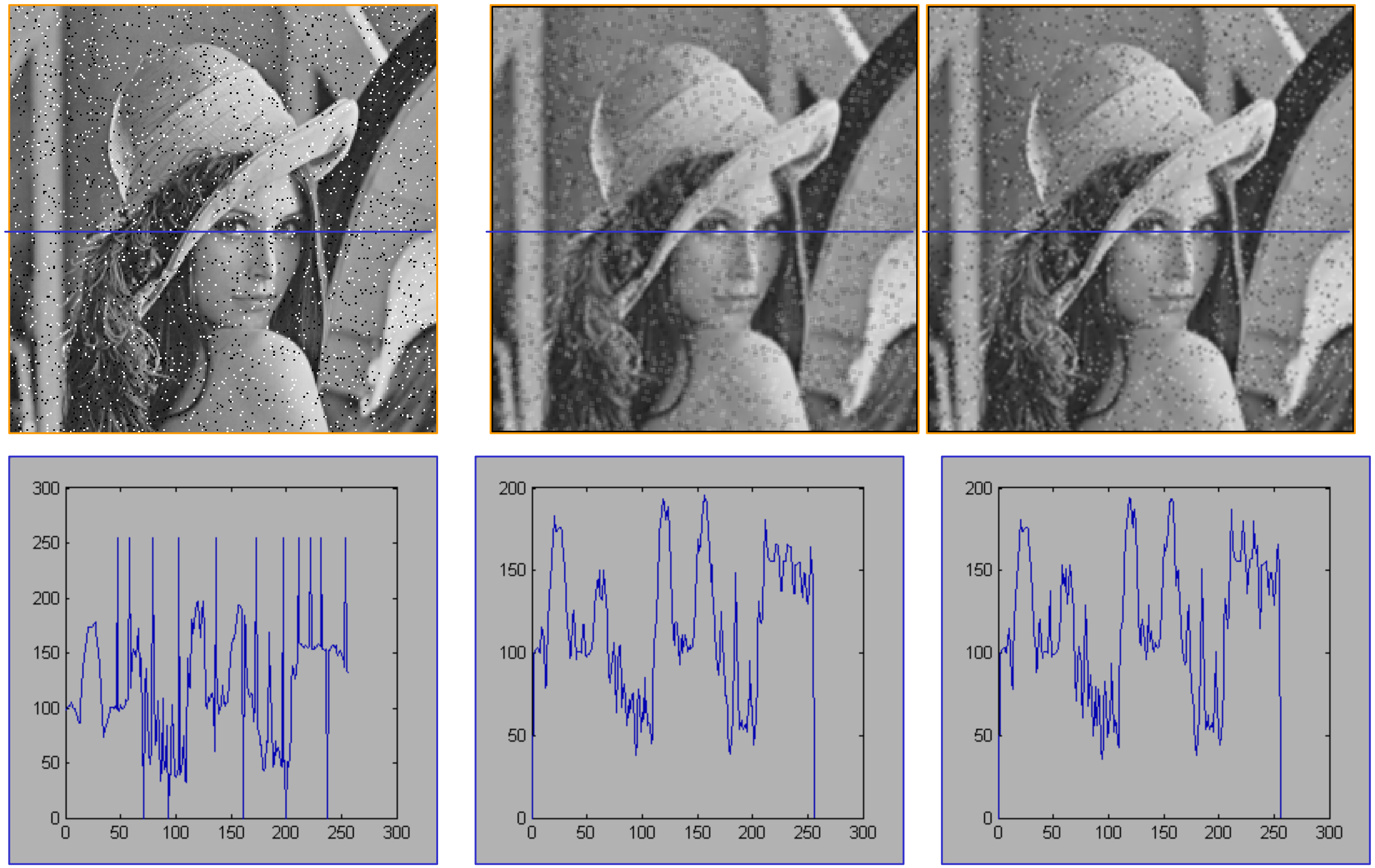
Typical mask:

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

4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

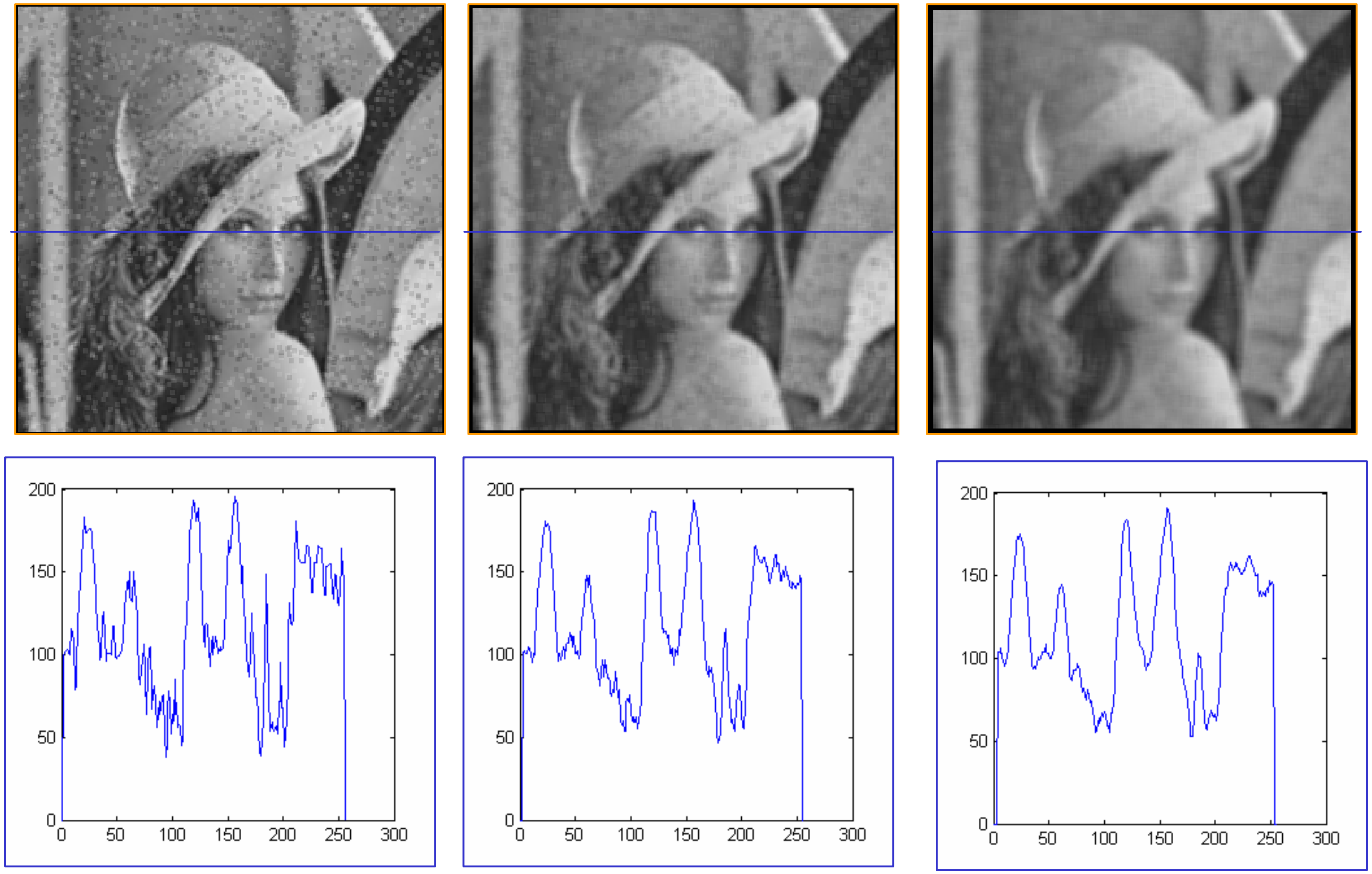
Experimental results: different masks



4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Experimental results:different sizes



4.3 Image Smoothing

4.3.2 Smoothing linear filters: out range pixel smoothing

formulate

$$g(x,y)=\sum_{i=-a}^{a}\sum_{j=-b}^{b} w(i,j)\,f(x+i,y+j)$$

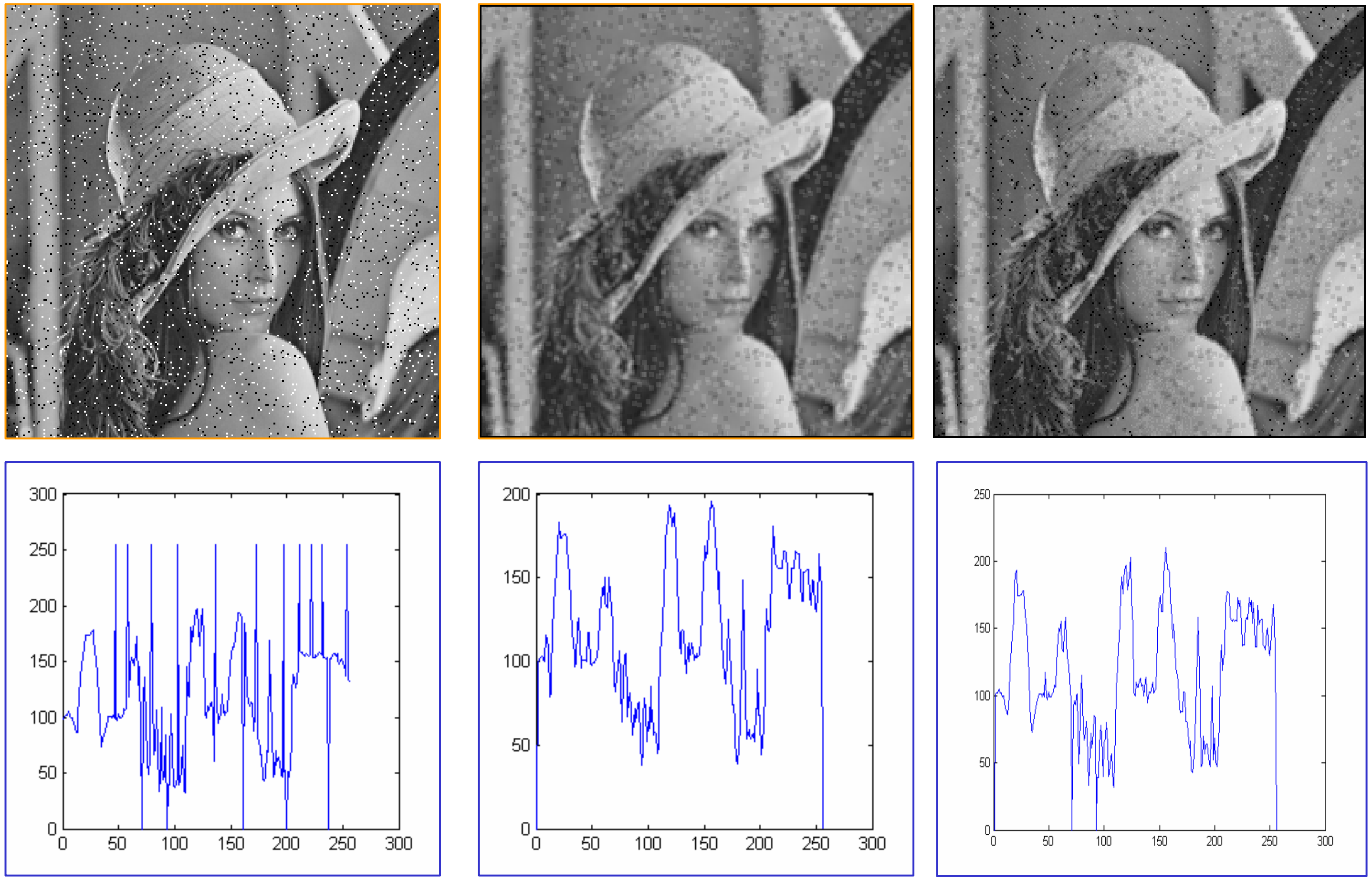
$$\hat{g}(x,y)=\begin{cases} g(x,y) & |g(x,y)-f(x,y)|>T \\ f(x,y) & \text{others} \end{cases}$$

Advantage: keep details in the image with salt-pepper noise

4.3 Image Smoothing

4.3.2 Smoothing linear filters: out range pixel smoothing

Experimental results

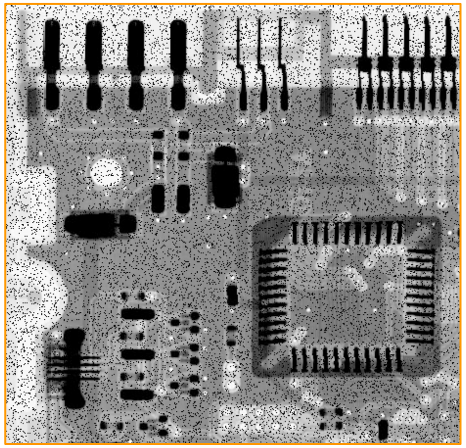


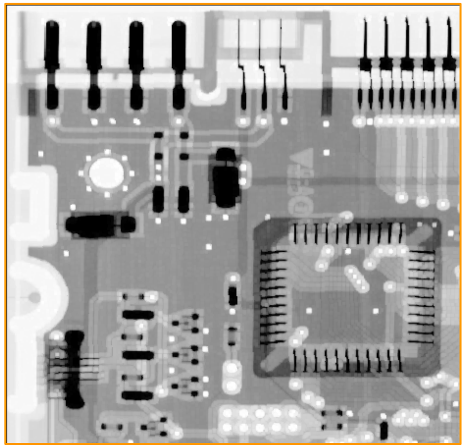
4.3 Image Smoothing

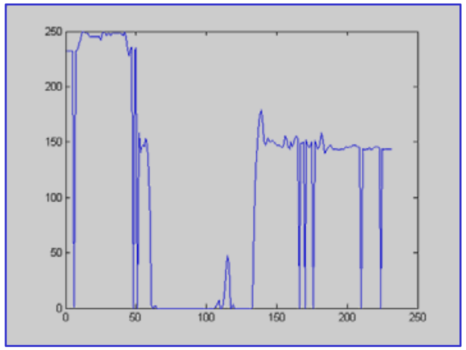
4.3.3 Order- statistic filters: Max filters

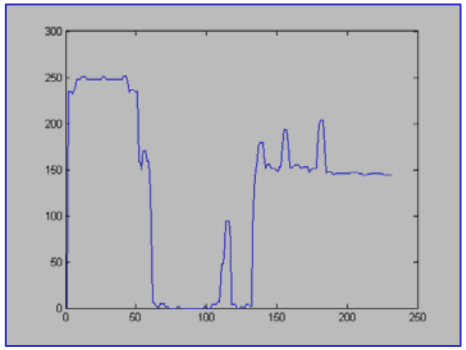
formulate

$$g(x,y) = \max_{(i,j=0,\pm 1\cdots)} \{ f(x+i, y+j) \}$$







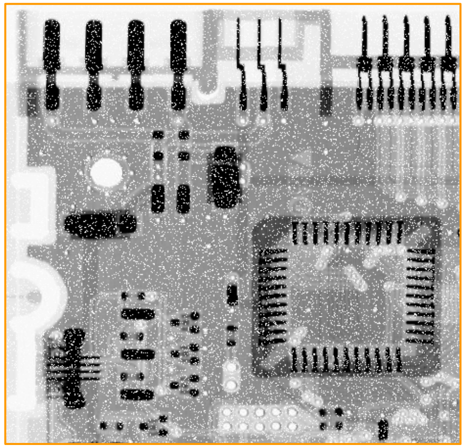


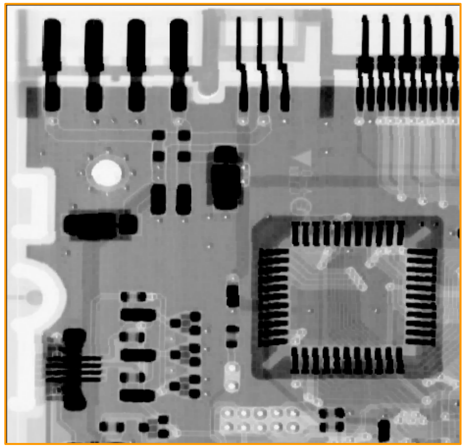
Advantage: removes pepper noise

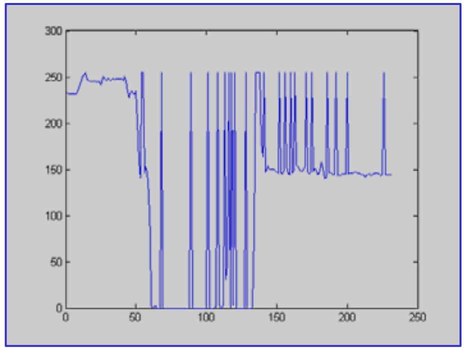
4.3 Image Smoothing

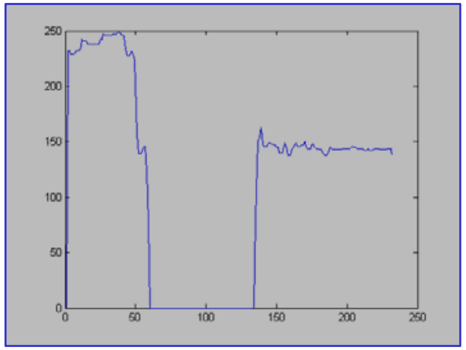
4.3.3 Order- statistic filters: Min filters

formulate $$g(x,y)=\min_{(i,j=0,\pm 1\cdots)}\{f(x+i,y+j)\}$$







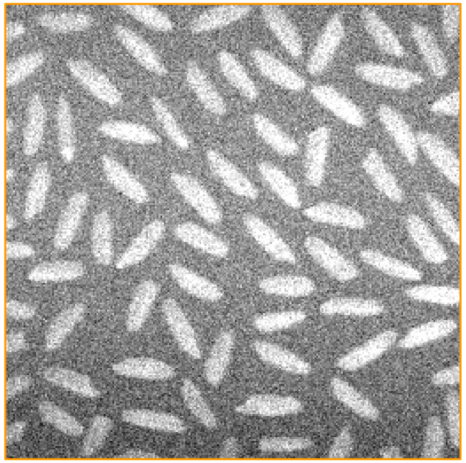


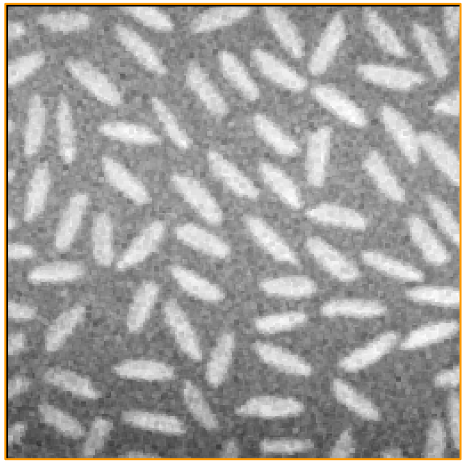
Advantage: removes salt noise

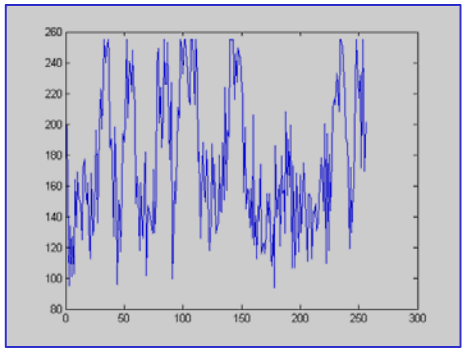
4.3 Image Smoothing

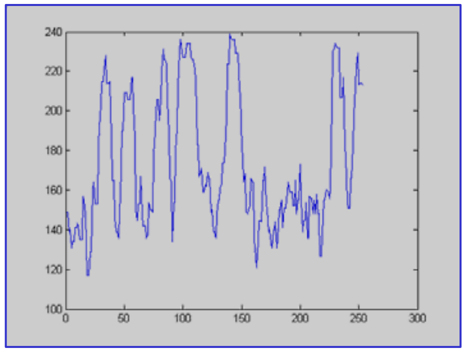
4.3.3 Order- statistic filters: Midpoint filter

formulate $$g(x,y)=\frac{1}{2}[\max_{(i,j=0,\pm1\cdots)}\{f(x+i,y+j)\}+\min_{(i,j=0,\pm1\cdots)}\{f(x+i,y+j)\}]$$









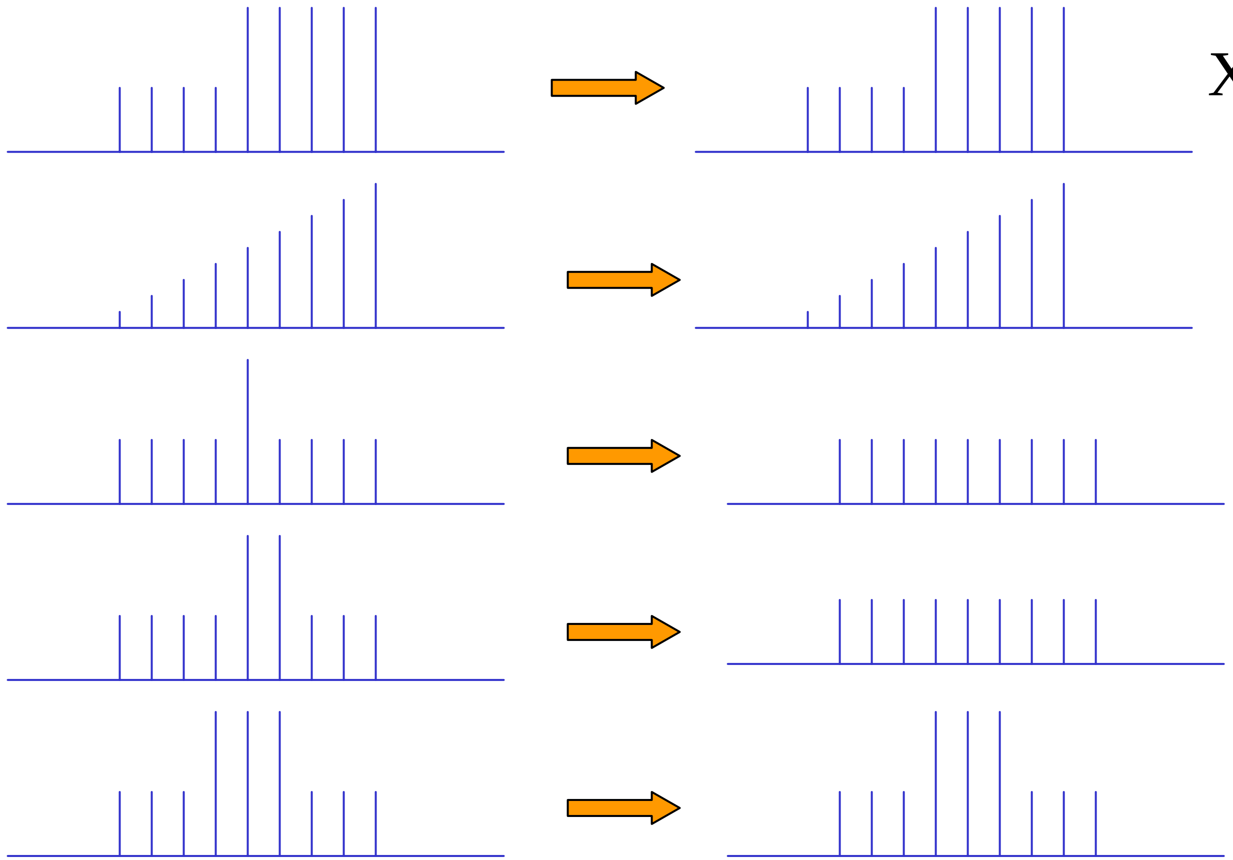
Advantage: removes Gausian noise

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

1-D: replace the signal value by the median value of neighborhood

L=5



X=[4 5 9 5 4]

[4 4 5 5 9]

[4 5 5 5 4]

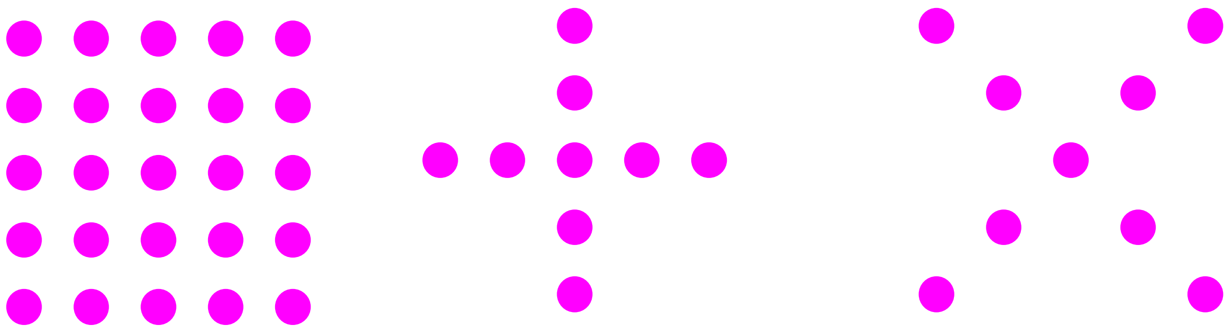
Advantage: keep edges

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

Replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel

$$g(x,y) = \underset{(i,j=0,\pm 1\cdots)}{median}\{f(x+i, y+j)\}$$



Shapes of 2-D filter

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

Experiment result

4.3 Image Smoothing

4.3.3 Order- statistic filters: alpha-trimmed mean filter

Delete the α lowest and α highest gray-level values of a sub-image, averaging the remaining pixels as output

Let the pixels in a sub-image: $A_0 \le A_1 \cdots \le A_{N-1}$

Then:

$$g(x,y) = \frac{1}{N-2\alpha}\sum_{i=\alpha}^{N-\alpha} A_i$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

formulate

$$g(x, y) = h(x, y) * f(x, y)$$

$$G(u, v) = H(u, v) F(u, v)$$

D_0: *cutoff frequency*

where

$$H(u,v) = \begin{cases} 1 \quad if \quad D(u,v) \le D_0 \\ 0 \quad if \quad D(u,v) > D_0 \end{cases}$$

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Properties: total image power

$$P_T = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} P(u,v) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \left|F(u,v)\right|^2$$

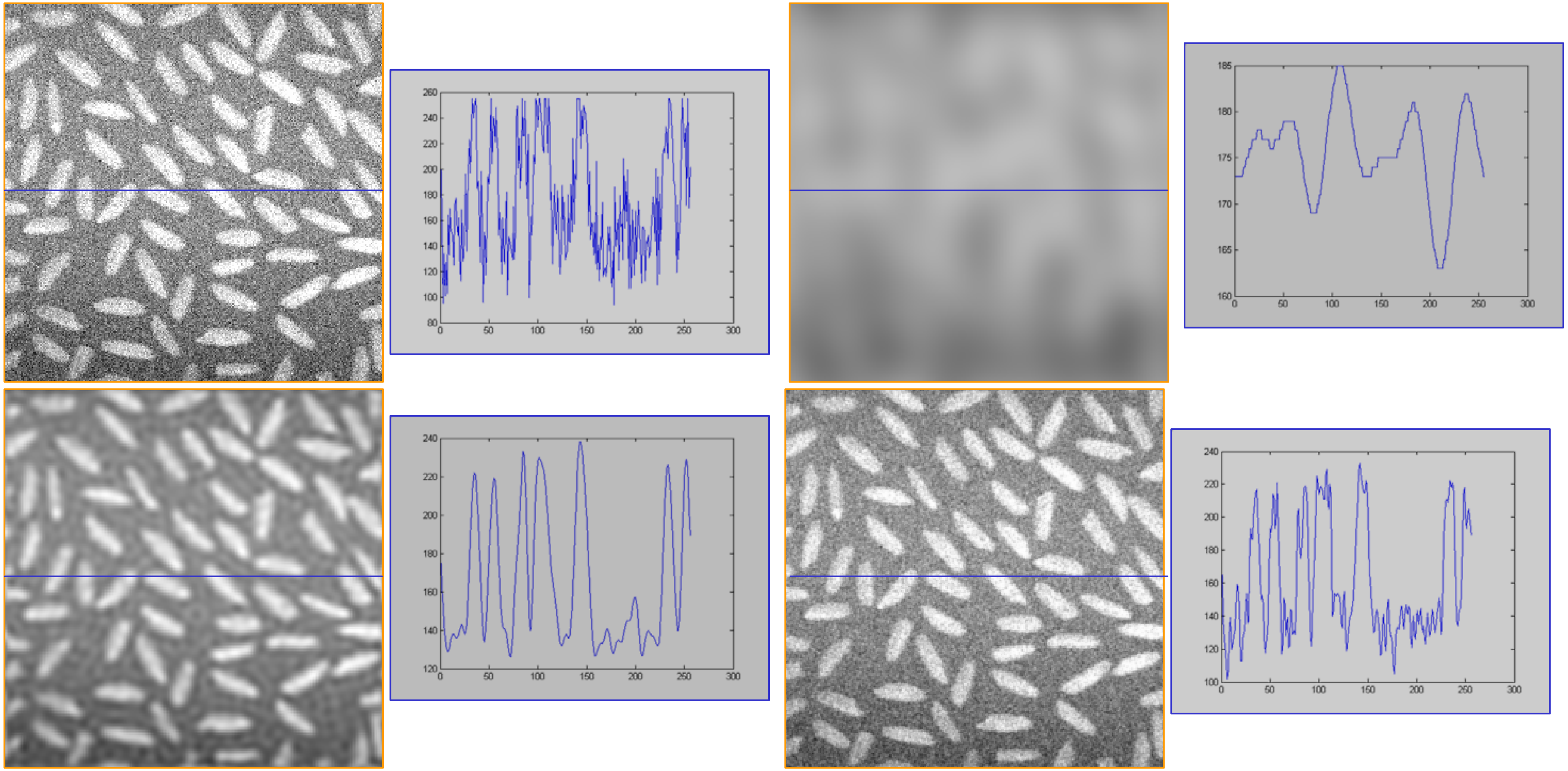
Power percent in a circle

$$\alpha = 100 \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} P(u,v) / P_T$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Experiment result cutoff frequencies set at radii values of 5、30、80



4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

formulate

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

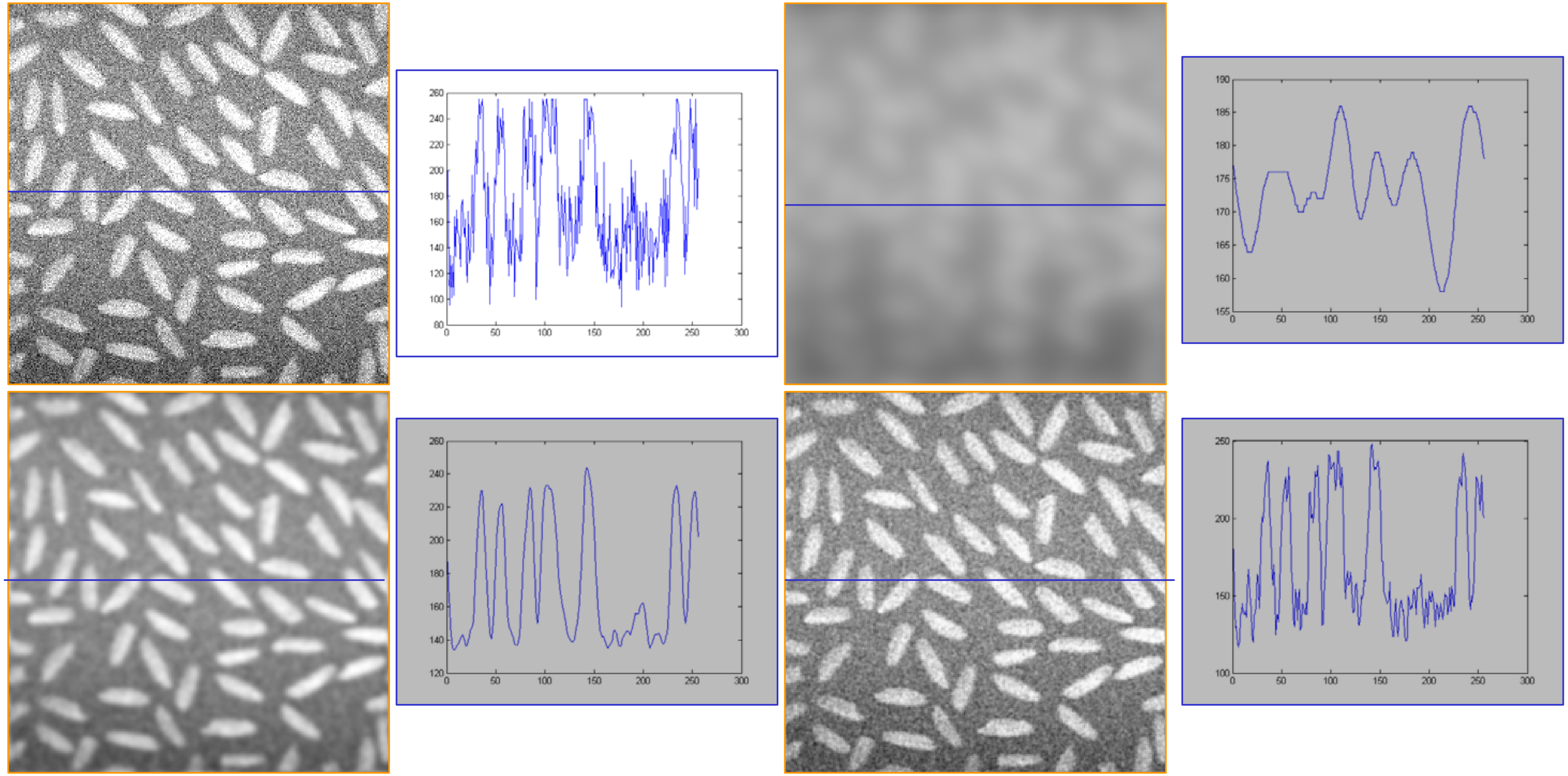
where

$$D(u,v) = [(u-M/2)^2 + (v-N/2)^2]^{1/2}$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

Experiment result cutoff frequencies set at radii values of 5、30、80



4.3 Image Smoothing

4.3.4 Low-pass filters: Gaussian low-pass filter

formulate

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

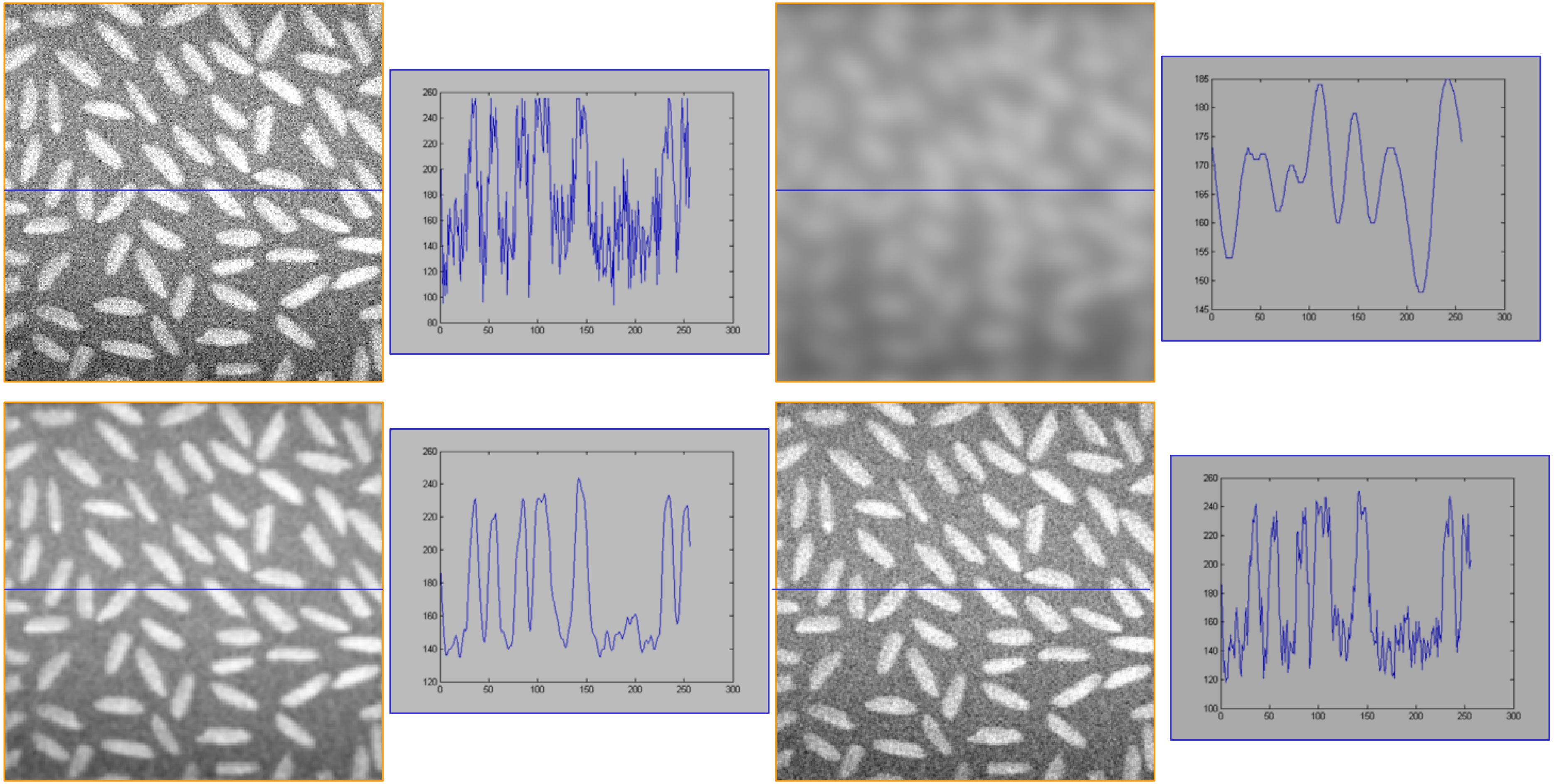
where

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

4.3 Image Smoothing

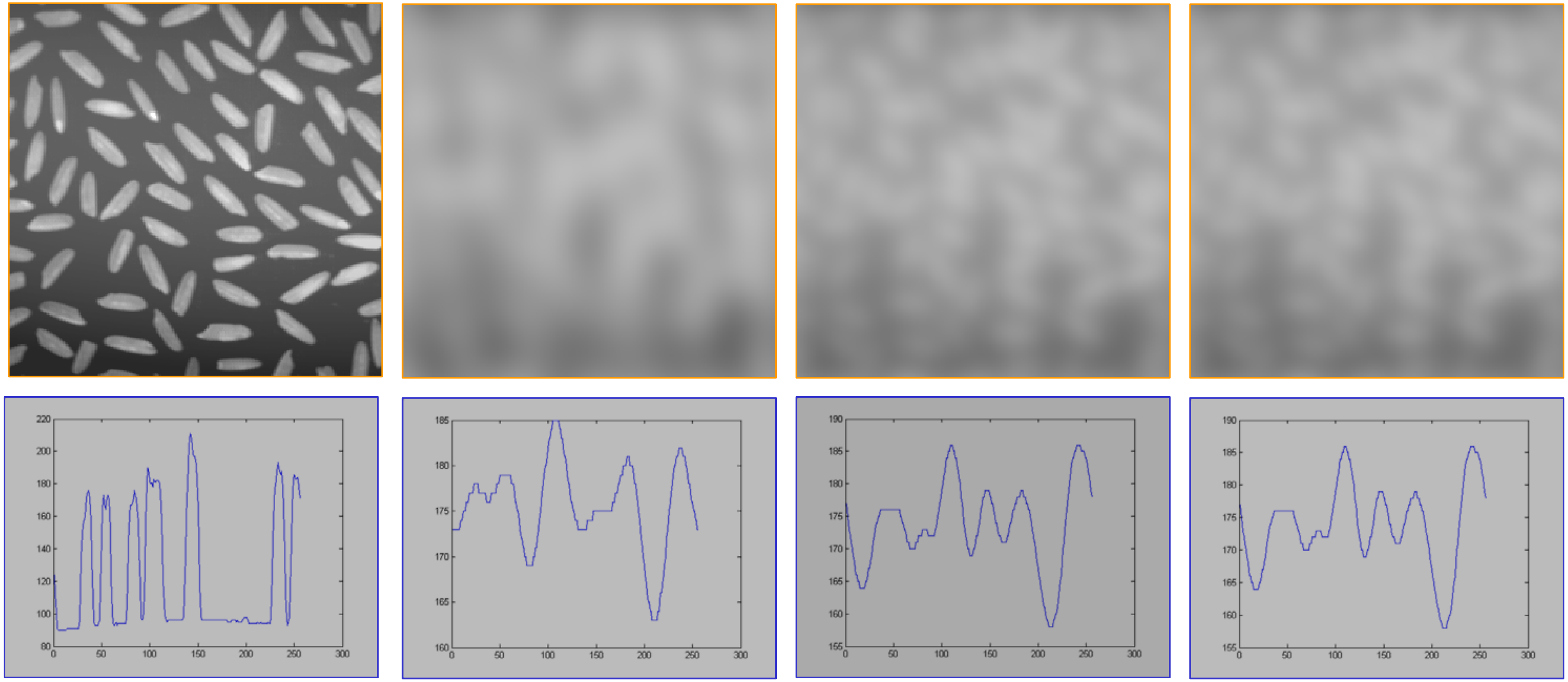
4.3.4 Low-pass filters: Gaussian low-pass filter

Experiment result cutoff frequencies set at radii values of 5、30、80



4.3 Image Smoothing

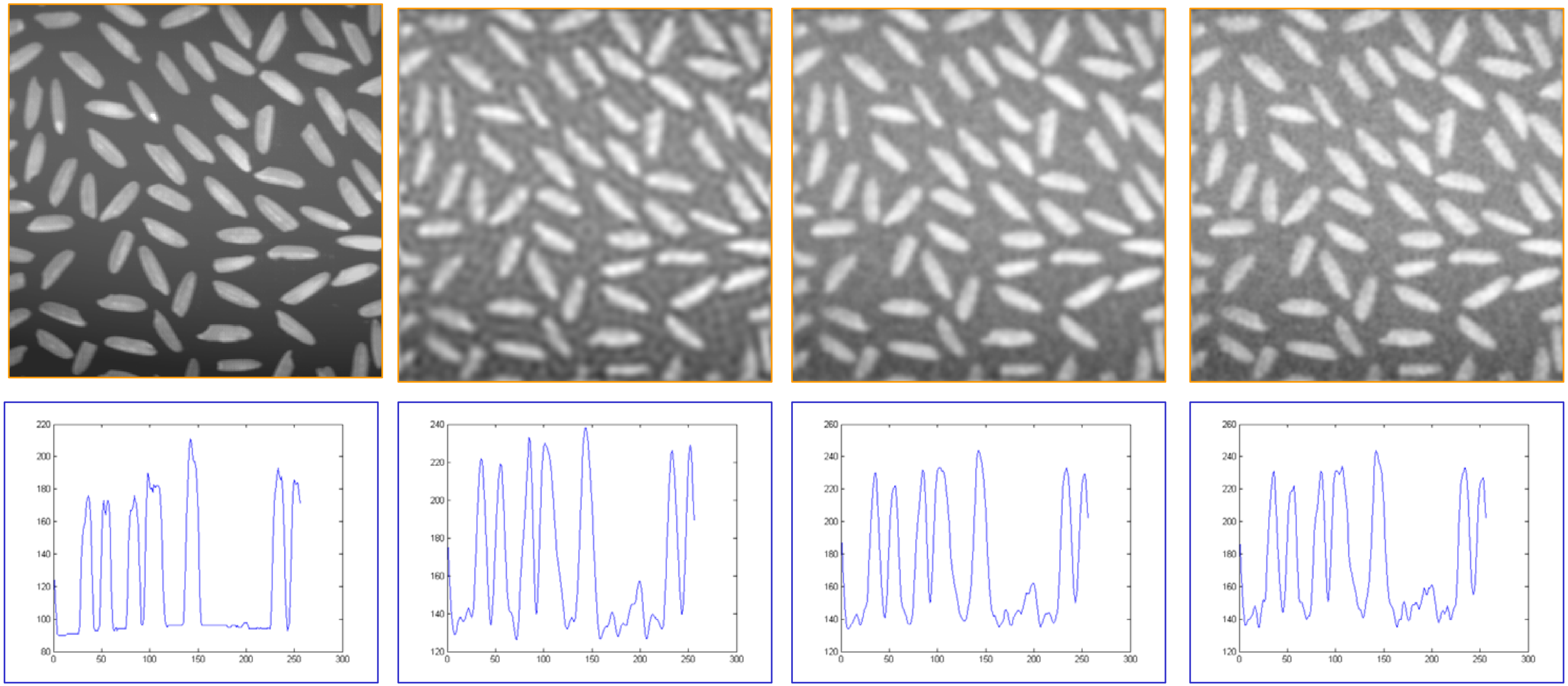
4.3.4 Low-pass filters: Comparisons (cutoff 5)



Origin Ideal filter Butterworth filter Gaussian filter

4.3 Image Smoothing

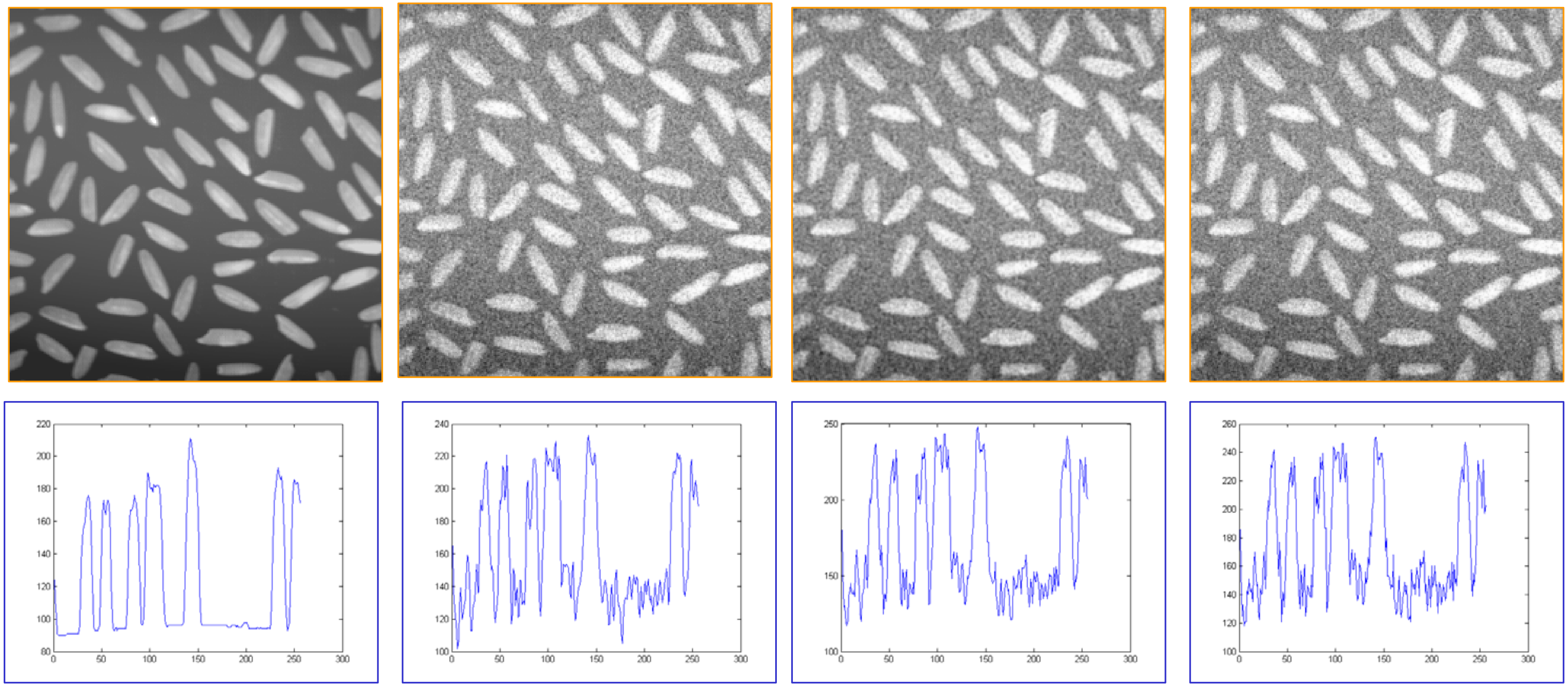
4.3.4 Low-pass filters: Comparison (cutoff 30)



Origin Ideal filter Butterworth filter Gaussian filter

4.3 Image Smoothing

4.3.4 Low-pass filters: Comparison (cutoff 80)



Origin | Ideal filter | Butterworth filter | Gaussian filter

4.4 Image Sharpening

- Introduction
- First-order derivative (in spatial domain)
- Second-order derivative (in spatial domain)
- High-pass filter in frequency domain

4.4 Image Sharpening

4.4.1 Introduction

Purposes: easy to detect edges

highlight fine detail

Request: insensitive to noise

4.4 Image Sharpening

4.4.1 Introduction: Sharpening filters

(1) First-order derivative : Robert
Prewitt Sobel
Robinson Kirsch

(2) Second-order derivative : Laplacian
LoG

(3) High-pass filter in frequency domain :
Idea high-pass filter (IHPF)
Butterworth high-pass filter (BHPF)
Gaussian high-pass filter (GHPF)

4.4 Image Sharpening

4.4.2 first-order derivative : foundation

The derivatives of a digital function are defined in terms of differences

For first derivative it must be

(1) zero in flat segment

(2) nonzero at the onset of a gray-level step or ramp

(3) nonzero along ramp

For second derivative it must be

(1) zero in flat segment

(2) nonzero at the onset and end of a gray-level step or ramp

(3) zero along ramp of constant slope

4.4 Image Sharpening

4.4.2 first-order derivative : foundation

A basic definition of the first-order derivative of a 1-D function *f(x)* is the difference

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

Similarly, we define a second-order derivative as the difference

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

4.4 Image Sharpening

4.4.2 first-order derivative : foundation

First-order derivatives of a digital image are based on various approximations of the 2-D gradient. It is defined as the vector:

$$\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T$$

Magnitude:

$$\left|\nabla f\right| = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right]^{1/2} \Longrightarrow \left|\nabla f\right| \approx \left|\left(\frac{\partial f}{\partial x}\right)\right| + \left|\left(\frac{\partial f}{\partial y}\right)\right|$$

$$\Downarrow$$

$$\left|\nabla f\right| \approx \max\left\{ \left|\frac{\partial f}{\partial x}\right|, \left|\frac{\partial f}{\partial y}\right| \right\}$$

Direction:

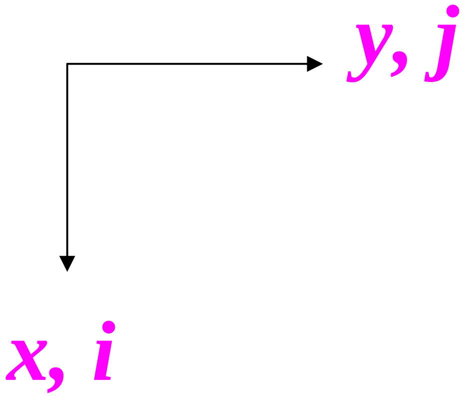
$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

4.4 Image Sharpening

4.4.2 first-order derivative : foundation

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

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



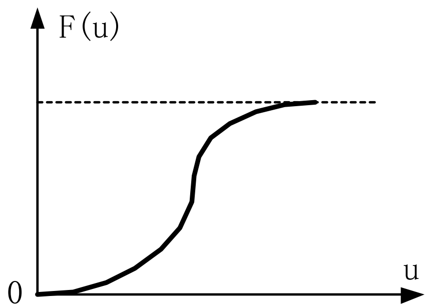
These equations can be implemented using masks

$$G_x = \begin{vmatrix} -1 & 0 \\ 1 & 0 \end{vmatrix} \qquad G_y = \begin{vmatrix} -1 & 1 \\ 0 & 0 \end{vmatrix}$$

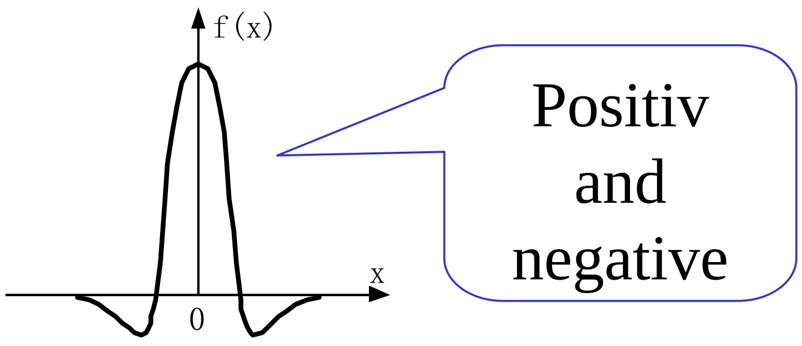
4.4 Image Sharpening

4.4.2 first-order derivative : foundation

Spatial filter: the sign of filter coefficients



(c)

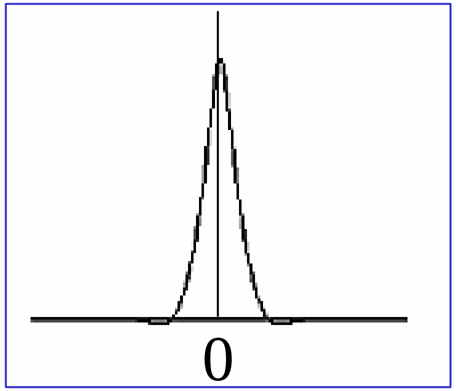


(d)

4.4 Image Sharpening

4.4.2 first-order derivative : display

$f(x,y)$ ⟶ $g(x,y)=\nabla f$



(1) $g(x,y)=\left|\nabla f\right|$

(2) $g(x,y)=f(x,y)+\left|\nabla f\right|$

(3) Linear scaling $g(x,y)$

(4) $g(x,y)=\begin{cases} L_b & \left|\nabla f\right| < \mathrm{T} \\ L_t & \text{othewise} \end{cases}$

4.4 Image Sharpening

4.4.2 first-order derivative : gradient operators

Roberts operator

$$G_x = \begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix} \qquad G_y = \begin{vmatrix} 0 & 1 \\ -1 & 0 \end{vmatrix}$$

Prewitt operator

$$G_x = \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{vmatrix} \qquad G_y = \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

Sobel operator

$$G_x = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \qquad G_y = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix}$$

4.4 Image Sharpening

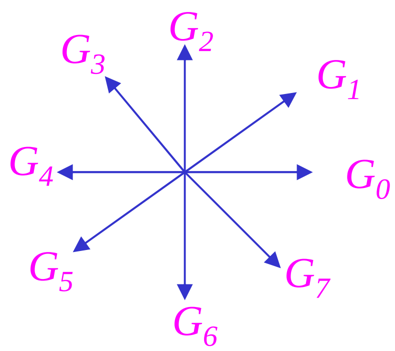
4.4.2 first-order derivative : gradient operators

Robinson

$$G_0 = \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} \quad G_1 = \begin{vmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{vmatrix} \quad G_2 = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} \quad G_3 = \begin{vmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{vmatrix}$$

$$G_4 = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix} \quad G_5 = \begin{vmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{vmatrix} \quad G_6 = \begin{vmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{vmatrix} \quad G_7 = \begin{vmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{vmatrix}$$

$$\left|\nabla f\right| \approx \max_{i=0,1\ldots7} \left\{ G_i \right\}$$



4.4 Image Sharpening

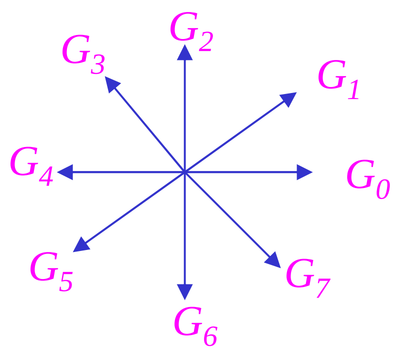
4.4.2 first-order derivative : gradient operators

Kirsch

$$G_0=\begin{vmatrix}5 & -3 & -3\\ 5 & 0 & -3\\ 5 & -3 & -3\end{vmatrix}\quad G_1=\begin{vmatrix}-3 & -3 & -3\\ 5 & 0 & -3\\ 5 & 5 & -3\end{vmatrix}\quad G_2=\begin{vmatrix}-3 & -3 & -3\\ -3 & 0 & -3\\ 5 & 5 & 5\end{vmatrix}\quad G_3=\begin{vmatrix}-3 & -3 & -3\\ -3 & 0 & 5\\ -3 & 5 & 5\end{vmatrix}$$

$$G_4=\begin{vmatrix}-3 & -3 & 5\\ -3 & 0 & 5\\ -3 & -3 & 5\end{vmatrix}\quad G_5=\begin{vmatrix}-3 & 5 & 5\\ -3 & 0 & 5\\ -3 & -3 & -3\end{vmatrix}\quad G_6=\begin{vmatrix}5 & 5 & 5\\ -3 & 0 & -3\\ -3 & -3 & -3\end{vmatrix}\quad G_7=\begin{vmatrix}5 & 5 & -3\\ 5 & 0 & -3\\ -3 & -3 & -3\end{vmatrix}$$

$$\left|\nabla f\right| \approx \max_{i=0,1\ldots7}\left\{G_i\right\}$$



4.4 Image Sharpening

4.4.3 second-order derivative : definitions

The second-order derivative is defined as

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

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

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

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

4.4 Image Sharpening

4.4.3 second-order derivative : Laplacian

It is well known as *Laplacian* operator. It can be implemented as a mask

$$G = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

4.4 Image Sharpening

4.4.3 second-order derivative : Laplacian

Experiment result $g(x, y) = f(x, y) + \nabla^2 f(x, y)$

4.4 Image Sharpening

4.4.3 second-order derivative : LoG

Smoothing first

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

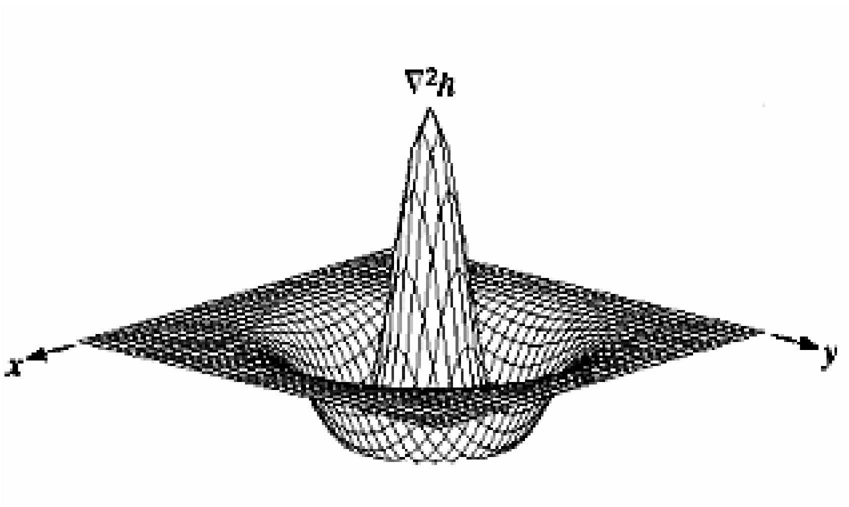
$$r = x^2 + y^2$$

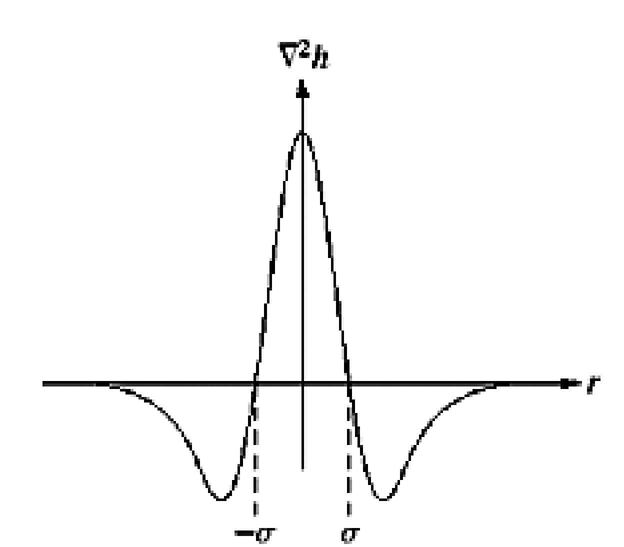
Laplacian sharpening

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

4.4 Image Sharpening

4.4.3 second-order derivative : LoG





0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

4.4 Image Sharpening

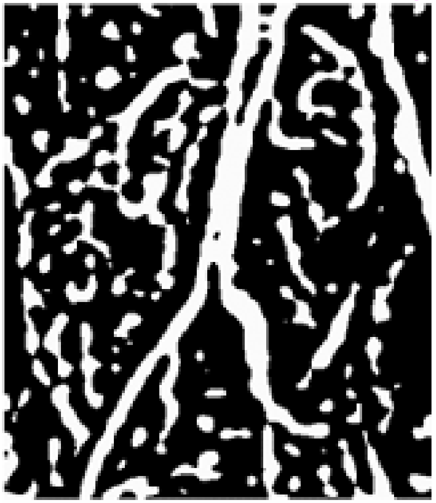
4.4.3 second-order derivative : LoG

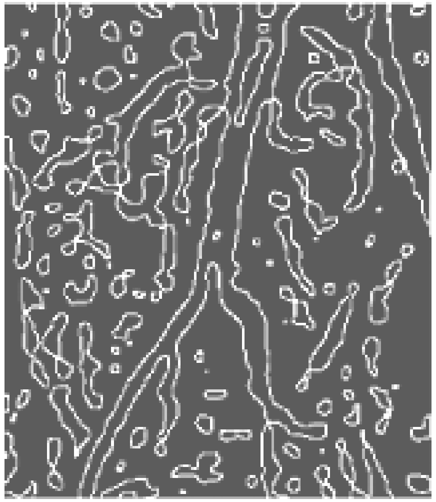

original


Sobel


LoG

Thresholded LoG



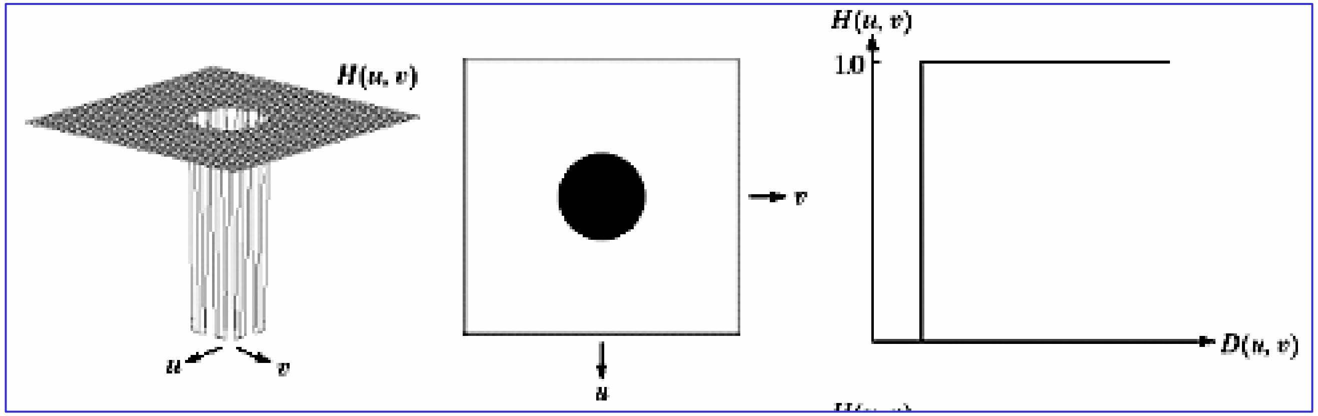

Zero cross

4.4 Image Sharpening

4.4.3 High-pass filter: Ideal highpass filter (IHPF)

formula

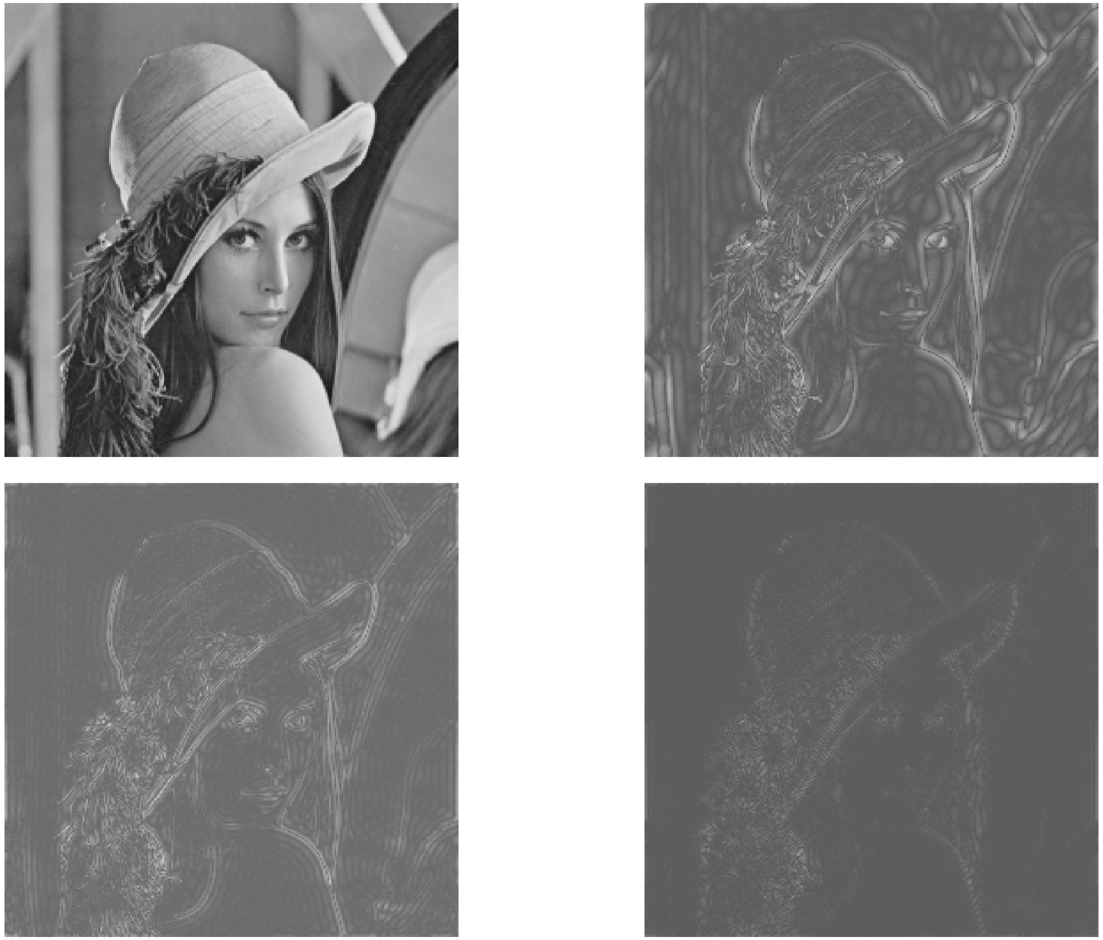
$$H(u,v) = \begin{cases} 0 & if \quad D(u,v) \le D_0 \\ 1 & if \quad D(u,v) > D_0 \end{cases}$$



4.4 Image Sharpening

4.4.3 High-pass filter: Ideal highpass filter (IHPF)

cutoff frequencies set at radii values of 15、30、80

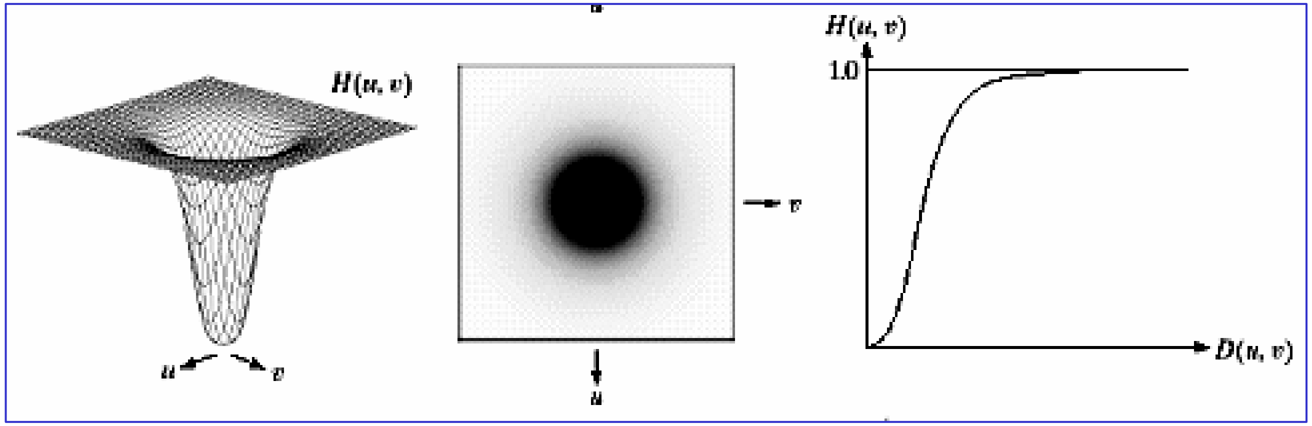


4.4 Image Sharpening

4.4.3 High-pass filter: Butterworth highpass filter (BHPF)

formula

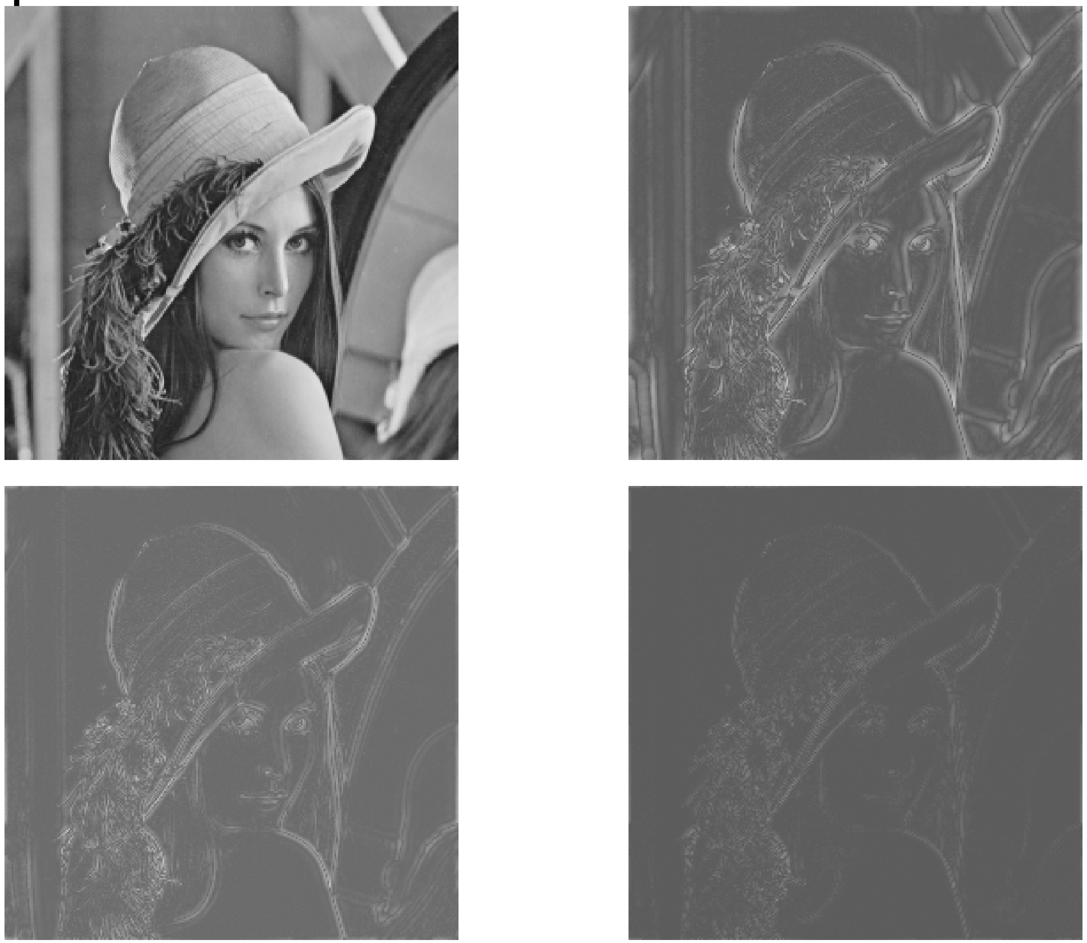
$$H(u,v)=\frac{1}{1+[D_0/D(u,v)]^{2n}}$$



4.4 Image Sharpening

4.4.3 High-pass filter: Butterworth highpass filter (BHPF)

cutoff frequencies set at radii values of 15、30、80.

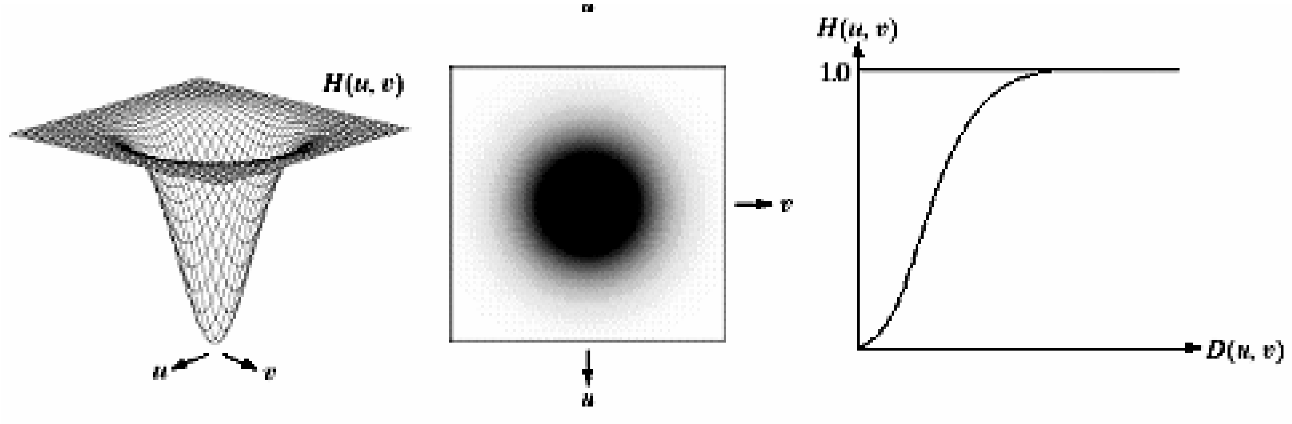


4.4 Image Sharpening

4.4.3 High-pass filter: Gaussian highpass filter (GHPF)

formula

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$



4.4 Image Sharpening

4.4.3 High-pass filter: Gaussian highpass filter (GHPF)

cutoff frequencies set at radii values of 15、30、80.

4.4 Image Sharpening

4.4.3 High-pass filter: Homomorphic filter

An image f(x,y) can be expressed as the product of illumination and reflectance components

$$f(x,y) = i(x,y)r(x,y)$$

let

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$$

4.4 Image Sharpening

4.4.3 High-pass filter: Homomorphic filter

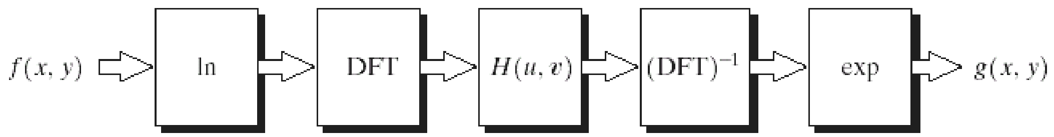


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

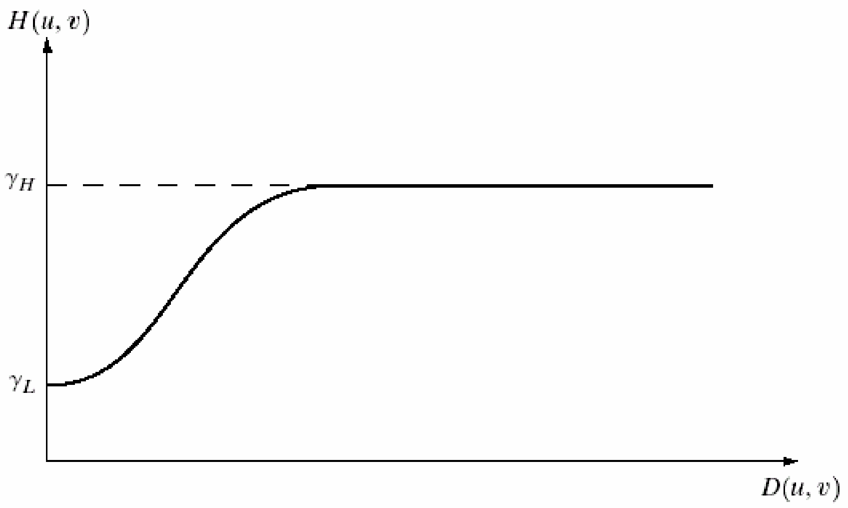


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

4.5 Color Image Enhancement

- Introduction
- Pseudo color image processing
- Full color enhancement
- Noise in color image

4.5 Color Image Enhancement

4.5.2 Pseudo color image processing : Intensity Slicing

formula

$$f(x, y) = c_k \quad \text{if} \quad f(x, y) \in V_k$$

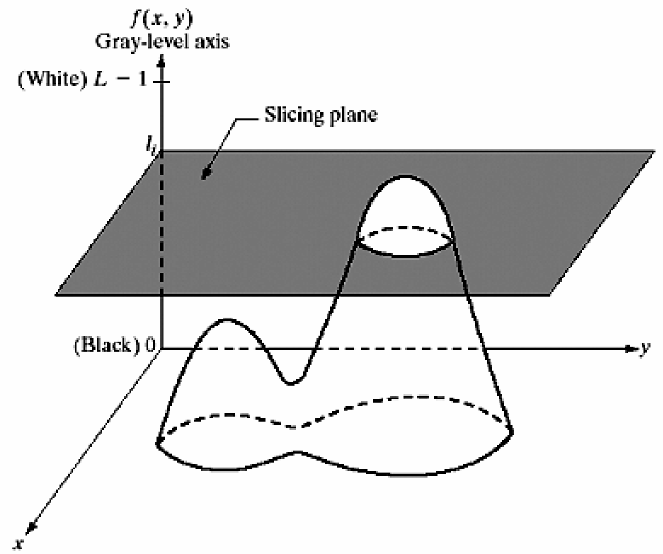


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

4.5 Color Image Enhancement

4.5.2 Pseudo color image processing : Gray level to color

Combine several monochrome images into a single color composite

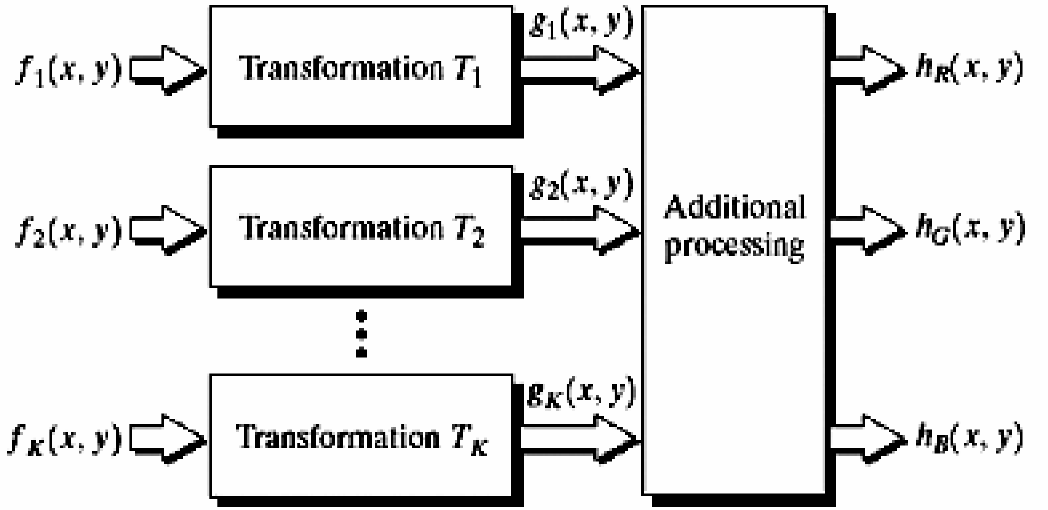


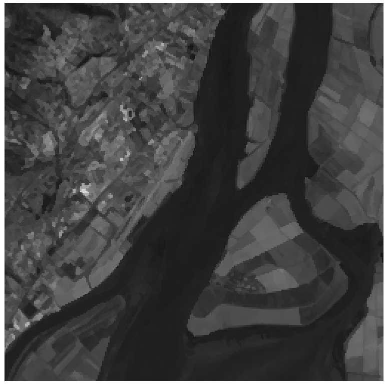
FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

4.5 Color Image Enhancement

4.5.2 Pseudo color image processing : Gray level to color

band1

band2

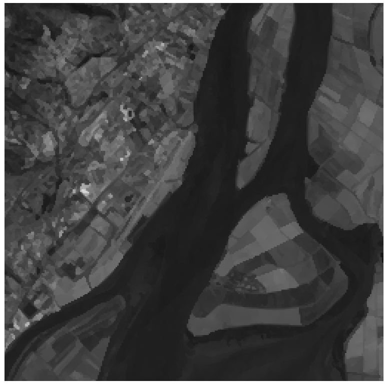


band3

band1,2,3 false color

band5

band4



band3

band5,4,3 false color

4.5 Color Image Enhancement

4.5.3 Full color enhancement : Fundamentals

A full color image can be expressed as

$$c(x,y) = \begin{bmatrix} c_R(x,y) \\ c_G(x,y) \\ c_B(x,y) \end{bmatrix} = \begin{bmatrix} R(x,y) \\ G(x,y) \\ B(x,y) \end{bmatrix}$$

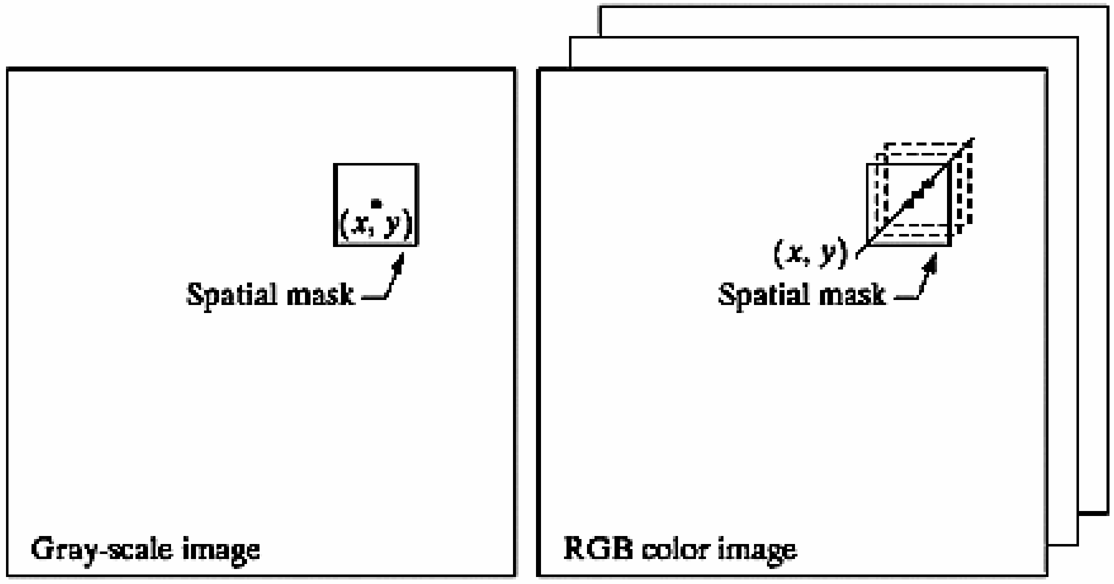
A pixel in color image interpreted as vector

$$c = \begin{bmatrix} c_R \\ c_G \\ c_B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

4.5 Color Image Enhancement

4.5.3 Full color enhancement : Fundamentals

Mask operation in color image



4.5 Color Image Enhancement

4.5.3 Full color enhancement : Color transformations

Adjusting the intensity of an image:

in RGB color space three components must be transformed:

$$s_i = kr_i \quad i = 1,2,3 \quad 0 < k < 1$$

in HSI color space only intensity components is modified:

$$s_3 = kr_3$$

$$s_1 = r_1$$

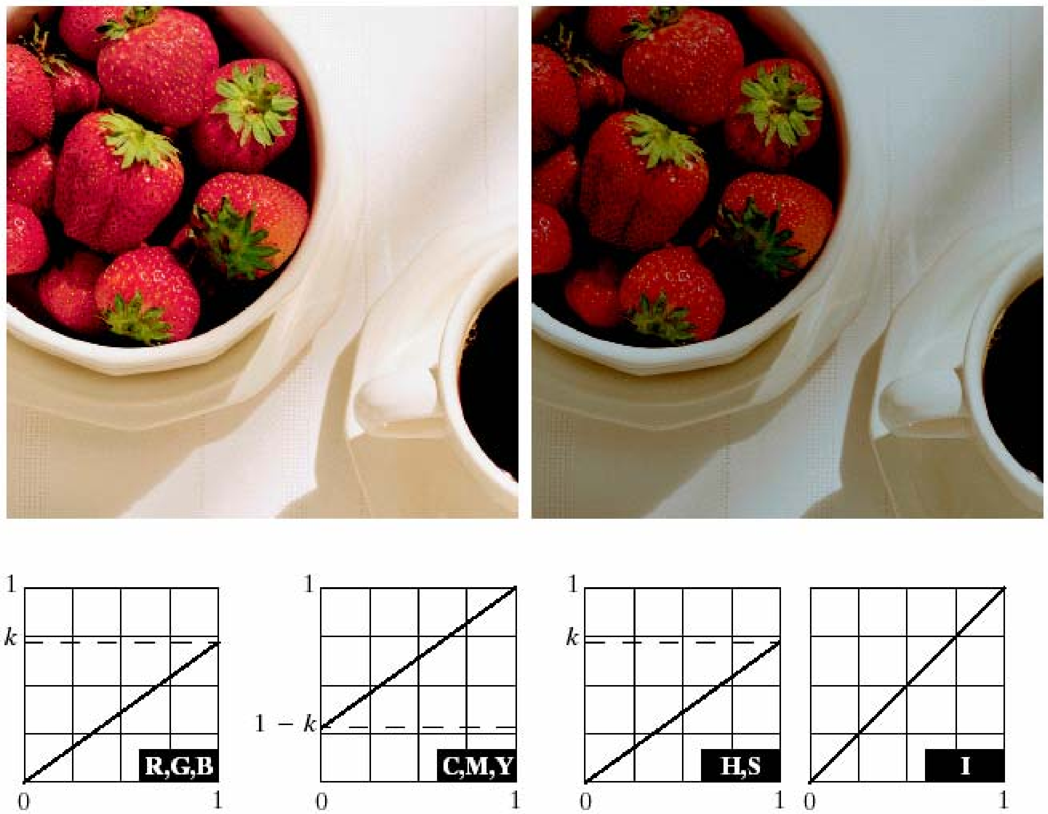
$$s_2 = r_2$$

4.5 Color Image Enhancement

4.5.3 Full color enhancement : Color transformations

a b
c d e

FIGURE 6.31 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)



I HS

S HI

4.5 Color Image Enhancement

4.5.3 Full color enhancement : Color Complement

in RGB color space three components must be transformed:

$$s_i = 1 - r_i \quad i = 1,2,3 \quad 0 < k < 1$$

in HSI color space only intensity components is modified:

$$s_1 = (r_1 + 0.5)$$

$$s_2 = r_2$$

$$s_3 = 1 - r_3$$

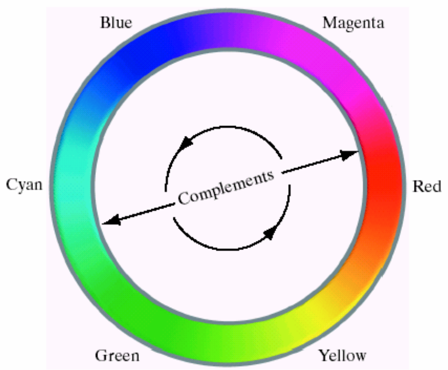
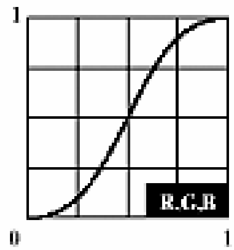


FIGURE 6.32 Complements on the color circle.

4.5 Color Image Enhancement

4.5.3 Full color enhancement : color corrections

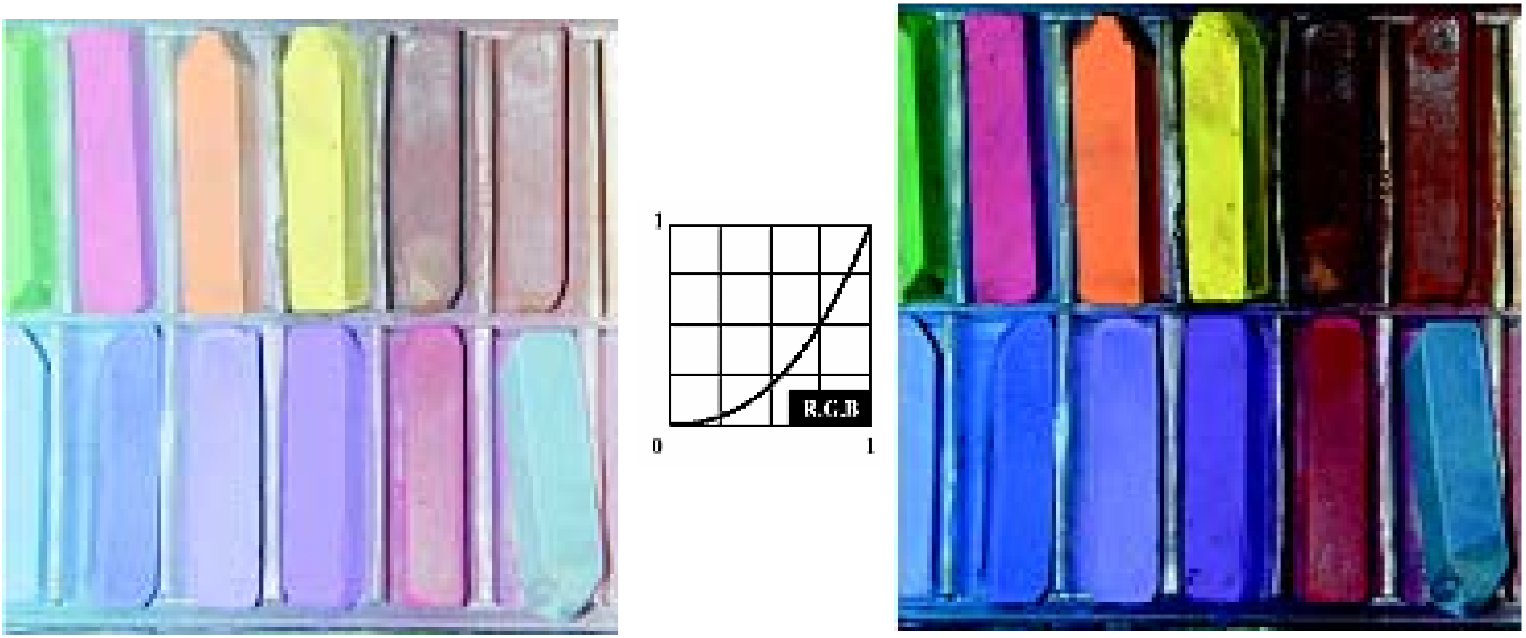
In RGB and CMY(K) space, map all tree (or four) color components with the same transform function



4.5 Color Image Enhancement

4.5.3 Full color enhancement : color corrections

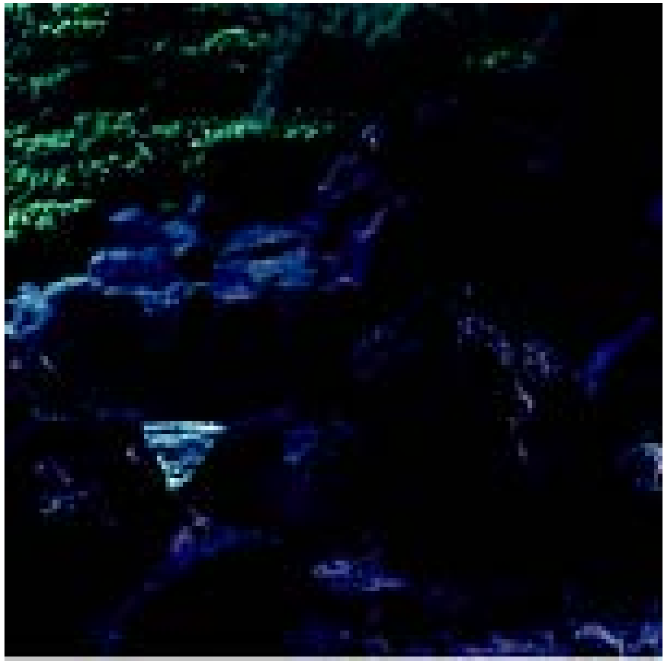
example

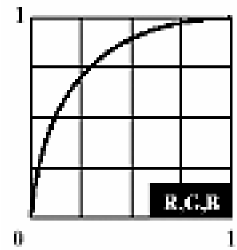


4.5 Color Image Enhancement

4.5.3 Full color enhancement : color corrections

example





Chapter5 Image Restoration

- 5.1 Introduction
- 5.2 Diagonalization
- 5.3 Unconstrained Restoration(inverse filtering)
- 5.4 Constrained Restoration(wiener filtering)
- 5.5 Estimating the Degradation Function
- 5.6 Geometric Distortion Correction
- 5.7 image inpaiting

5.1 Introduction

5.1.1 Purpose

"compensate for" or "undo" defects which degrade an image.

5.1.2 Degrade Causes

(1) atmospheric turbulence

(2) sampling, quantization

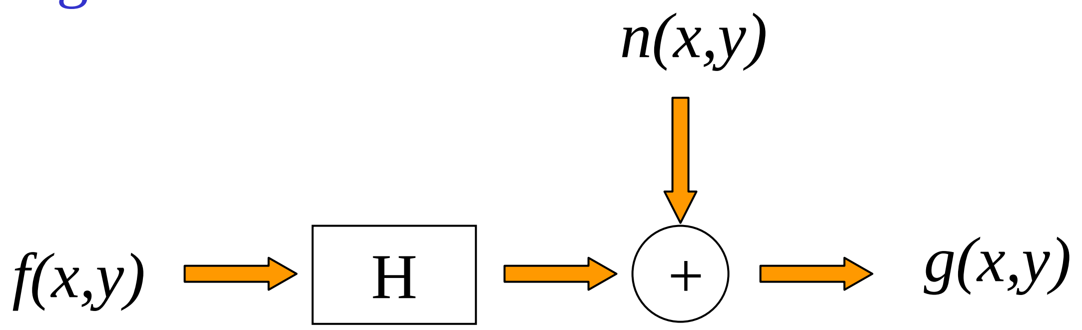
(3) motion blur

(4) camera misfocus

(5) noise

5.1 Introduction

5.1.3 degradation model



Assume it is a linear, position- invariant system, We can model a blurred image by

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

Where *h(x,y)* is called as Point Spread Function (PSF)

5.1 Introduction

5.1.4 Methods

Unconstrained Restoration: inverse filtering

Constrained Restoration: wiener filtering

5.1.5 problem expression

Estimate a true image $f(x,y)$ from a degraded image $g(x,y)$ based on prior knowledge of PSF $h(x,y)$ and the statistical properties of noise $n(x,y)$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

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

$$f_e(x) = \begin{cases} f(x) & 0 \le x \le A-1 \\ 0 & \text{else} \end{cases}$$

$$h_e(x) = \begin{cases} h(x) & 0 \le x \le B-1 \\ 0 & \text{else} \end{cases}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

$$g_e(x)=\sum_{m=0}^{M-1} f_e(m)h_e(x-m)+n_e(x) \qquad M=A+B \qquad x=0,1,\ldots,M-1$$

$$g=Hf+n=\begin{bmatrix} g_e(0) \\ g_e(1) \\ \vdots \\ g_e(M\text{-}1) \end{bmatrix}=\begin{bmatrix} h_e(0) & h_e(-1) & \cdots & h_e(-M+1) \\ h_e(1) & h_e(0) & \cdots & h_e(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \cdots & h_e(0) \end{bmatrix}\begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(M\text{-}1) \end{bmatrix}+\begin{bmatrix} n_e(0) \\ n_e(1) \\ \vdots \\ n_e(M-1) \end{bmatrix}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 1-D

$$h_e(x) = h_e(x+M)$$

$$H = \begin{bmatrix} h_e(0) & h_e(M-1) & \cdots & h_e(1) \\ h_e(1) & h_e(0) & \cdots & h_e(2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(M-1) & h_e(M-2) & \cdots & h_e(0) \end{bmatrix}$$

H is circulant

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

$$f_e(x)=\begin{cases} f(x,y) & 0\le x\le A-1 \text{ and } 0\le y\le B-1 \\ 0 & \mathrm{A}\le x\le M-1 \text{ or } \mathrm{B}\le y\le N-1 \end{cases}$$

$$h_e(x)=\begin{cases} h(x,y) & 0\le x\le C-1 \text{ and } 0\le y\le D-1 \\ 0 & \mathrm{A}\le x\le M-1 \text{ or } \mathrm{B}\le y\le N-1 \end{cases}$$

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

$$g_e(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f_e(m,n)h_e(x-m,y-n) \qquad x=0,1,\ldots,M-1 \quad y=0,1,\ldots,N-1$$

$$g = Hf + n = \begin{bmatrix} H_0 & H_{M-1} & \cdots & H_1 \\ H_1 & H_0 & \cdots & H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{M-1} & H_{M-2} & \cdots & H_0 \end{bmatrix} \begin{bmatrix} f_e(0) \\ f_e(1) \\ \vdots \\ f_e(MN\text{-}1) \end{bmatrix} + \begin{bmatrix} n_e(0) \\ n_e(1) \\ \vdots \\ n_e(MN-1) \end{bmatrix}$$

H is block-circulant

5.2 Diagonalization

5.2.1 Matrix expression of degradation model: 2-D

where

$$H_i = \begin{bmatrix} h_e(i,0) & h_e(i,N-1) & \cdots & h_e(i,1) \\ h_e(i,1) & h_e(i,0) & \cdots & h_e(i,2) \\ \vdots & \vdots & \ddots & \vdots \\ h_e(i,N-1) & h_e(i,N-2) & \cdots & h_e(i,0) \end{bmatrix}$$

5.2 Diagonalization

5.2.2 Diagonliztion: 1-D

The eigenvector and eigenvalue of a circulant matrix *H* are

$$w(k) = \left[1 \quad \exp\left(j\frac{2\pi}{M}k \right) \quad \cdots \quad \exp\left(j\frac{2\pi}{M}(M-1)k \right) \right]^T$$

$$\lambda(k) = h_e(0) + h_e(M-1)\exp\left(j\frac{2\pi}{M}k \right) + \cdots + h_e(1)\exp\left(j\frac{2\pi}{M}(M-1)k \right)$$

Combine the *M* eigenvectors to a matrix

$$W = [w(0) \quad w(1) \quad \cdots \quad w(M-1)]$$

then the *H* can be expressed as

$$H = WDW^{-1} \qquad \text{where} \qquad D(k,k) = \lambda(k)$$

5.2 Diagonalization

5.2.2 Diagonliztion: 1-D

for $$g = Hf + n$$

$$W^{-1}g = W^{-1}Hf + W^{-1}n$$
$$= W^{-1}WDW^{-1}f + W^{-1}n$$
$$= DW^{-1}f + W^{-1}n$$

$$G(u) = H(u)F(u) + N(u)$$

5.2 Diagonalization

5.2.2 Diagonliztion: 2-D

$$W(i,m) = \exp\left(j\frac{2\pi}{M}im \right) W_N$$

$$W_N(k,n) = \exp\left(j\frac{2\pi}{N}kn \right)$$

$$H = WDW^{-1} \Longrightarrow G(u,v) = H(u,v)F(u,v) + N(u,v)$$

$g(x,y) = f(x,y)*h(x,y) + n(x,y)$ in Spatial Coordinates

$G(u,v) = F(u,v)H(u,v) + N(u,v)$ in frequency domain

$g = Hf + n$ in vector form

5.3 Inverse Filtering

5.3.1 assumption

H is given, and the noise is negligible

5.3.2 degradation model

$$F(u,v) \rightarrow \boxed{H} \rightarrow G(u,v)$$

$$G(u,v) = H(u,v)F(u,v)$$

5.3 Inverse Filtering

5.3.3 restoration

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} = G(u,v)H_I(u,v)$$

$$H_I(u,v) = \frac{1}{H(u,v)}$$

deconvolution

5.3.4 properties

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$= \frac{H(u,v)F(u,v) + N(u,v)}{H(u,v)}$$

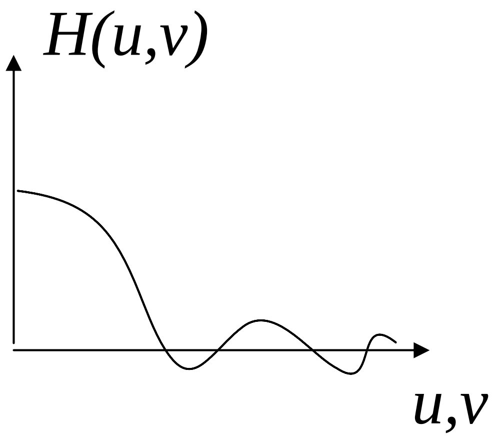
$$= F(u,v) + \frac{N(u,v)}{H(u,v)}$$

5.3 Inverse Filtering

5.3.4 properties

sensitive to additive noise: if *H(u,v)* has zero or very small value, the *N(u,v)/H(u,v)* could easily dominate the estimate

5.3.5 improvements



$$M(u,v)=\begin{cases} 1/H(u,v) & \text{if } u^2+v^2<w_0^2 \\ 1 & \text{else} \end{cases}$$

or

$$M(u,v)=\begin{cases} k & \text{if } H(u,v)<d \\ 1/H(u,v) & \text{else} \end{cases}$$

5.3 Inverse Filtering

5.3.4 examples: without noise

Original image

Blurred image

Restored image

5.3 Inverse Filtering

5.3.4 examples: with noise

Blurred and noised image

Restored image

5.4 Wiener Filtering

5.4.1 assumption

H is given, and consider the image and noise as random processes

5.4.2 request

The mean square error between uncorrupted image and estimated image is minimized. This error measure is given by

$$e^2 = E\left\{(f - \hat{f})^2\right\}$$

5.4 Wiener Filtering

5.4.3 restoration

$$\hat{F}(u,v) = G(u,v)H_W(u,v)$$

$$H_W(u,v) = \frac{1}{H(u,v)} \times \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

$$= \frac{H(u,v)^*}{|H(u,v)|^2 + S_n(u,v)/S_f(u,v)}$$

Wiener filter, 1942

where

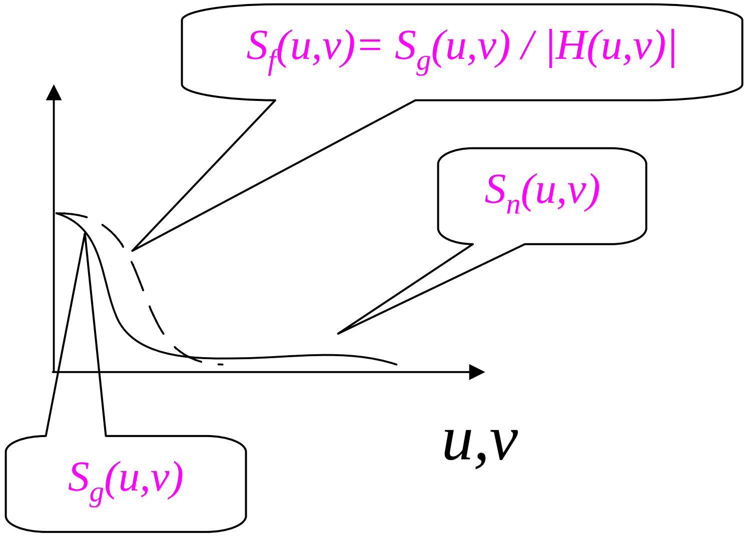
$S_n(u,v) = |N(u,v)|^2$ Power spectrum of the noise

$S_f(u,v) = |F(u,v)|^2$ Power spectrum of the undegraded image

5.4 Wiener Filtering

5.4.5 estimate the power spectrum

$$S_g(u,v) = |H(u,v)|^2 S_f(u,v) + S_n(u,v)$$

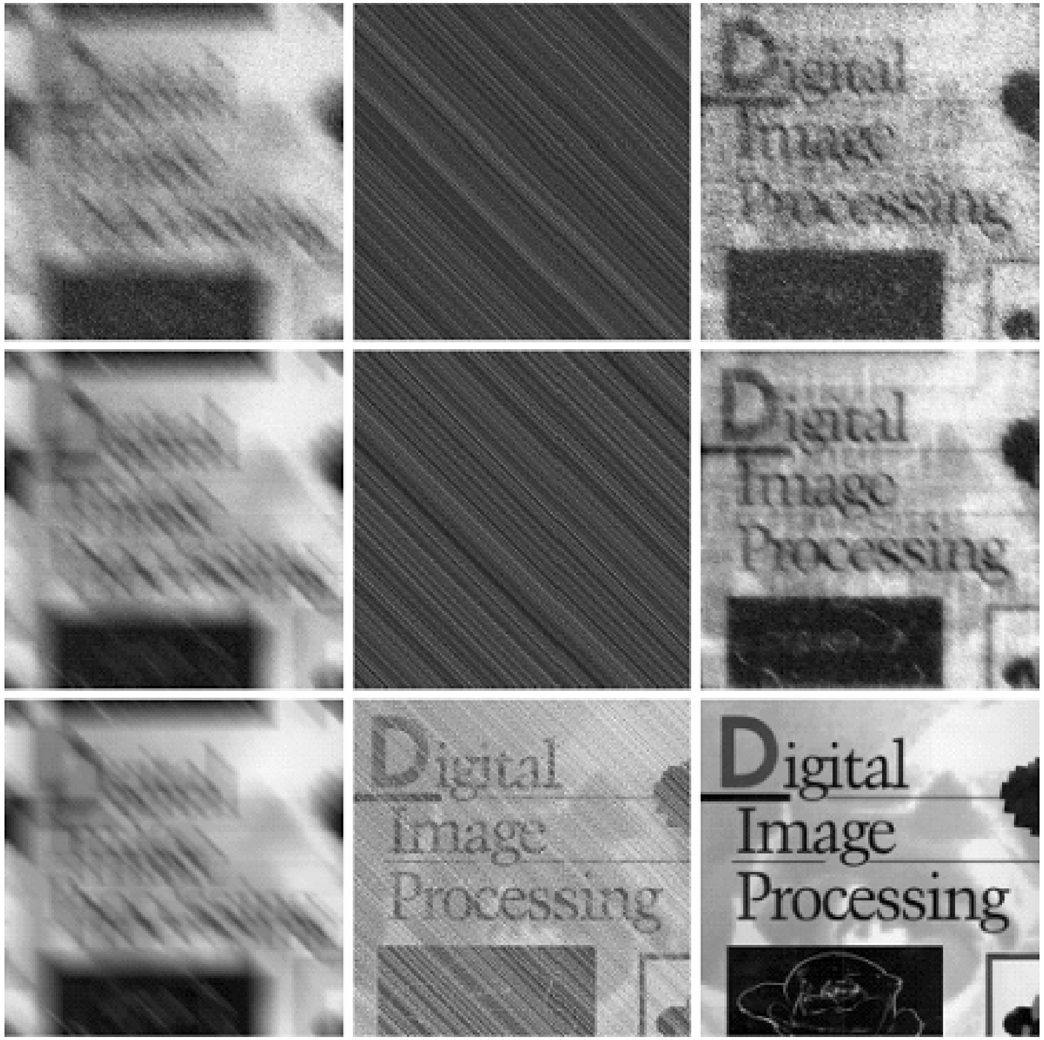


5.4.4 properties

- optimal in terms of the mean square error
- When $H(u,v)=0$, $H_w(u,v)=0$

5.4 Wiener Filtering

5.4.5 experimental results:



The first raw:
Image corrupted by motion blur and additive noise

The second raw:
Results of inverse filtering

The third raw:
Results of Wiener filtering

5.5Estimating the Degradation Function

(blind deconvolution)

5.5.1 Estimation by image observation

(1) Choose observed sub-image $g_s(x, y)$

(2) Denote the constructed sub-image as $\hat{f}_s(x, y)$

(3) Assume noise is negligible

then

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

5.5Estimating the Degradation Function

5.5.2 Estimation by experimentation

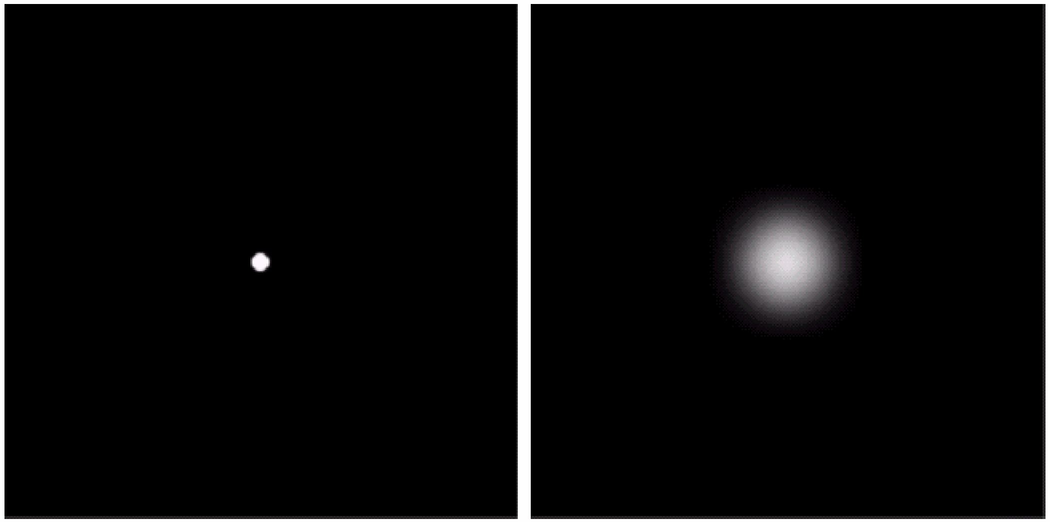
If $f(x,y) = \delta(x,y)$

$$\delta(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

Impulse response

then

$$g(x,y) = \delta(x,y) * h(x,y) = h(x,y)$$



a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling

An atmospheric turbulence model based on the physical characteristics

Hufnagel and Stanley, 1964

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

where k is a constant

it has the same form as the Gaussian lowpass filter

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

If T is the duration of the exposure, the effect of image motion follows that

$$g(x,y)=\int_0^T f\left[x-x_0(t),y-y_0(t)\right]dt$$

where $x_0(t)$ and $y_0(t)$ are the time varying components of motion in the x-direction and y-direction. Its Fourier transform is

$$G(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}g(x,y)e^{-2\pi j(ux+vy)}dxdy$$

$$=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x-x_0(t),y-y_0(t)]dt\right]e^{-2\pi j(ux+vy)}dxdy$$

5.5 Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

Reversing the order of integration:

$$G(u,v)=\int_0^T\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x-x_0(t),y-y_0(t)]e^{-2\pi j(ux+vy)}dxdy\right]dt$$

Using the translation Properties of Fourier transformation, then

$$G(u,v)=\int_0^T F(u,v)e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

$$=F(u,v)\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

By defining $$H(u,v)=\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$

then $$G(u,v)=H(u,v)F(u,v)$$

Suppose that the image in question undergoes uniform linear motion in the x-direction only, at a rate given by $x_0(t)=at/T$

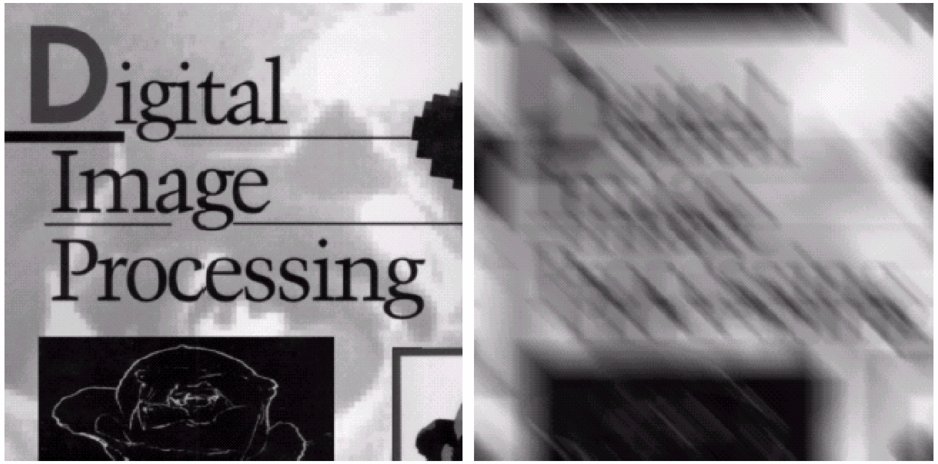
$$H(u,v)=\int_0^T e^{-2\pi j[ux_0(t)+vy_0(t)]}dt$$
$$=\int_0^T e^{-2\pi juat/T}dt$$
$$=\frac{T}{\pi ua}\sin(\pi ua)e^{-j\pi ua}$$

5.5Estimating the Degradation Function

5.5.3 Estimation by modeling :restoration of uniform linear motion

If we allow the y-component to wary as well with the motion given by $y_0(t) = bt/T$ then the degradation function becomes

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

5.6 Geometric Transformation

5.6.introduction

$$f(x,y) \longrightarrow \boxed{T} \longrightarrow g(x,y)$$

$$g(x,y) = T[(f(x,y)] = f(x',y')$$

where

$$x' = r(x,y) \quad y' = s(x,y)$$

for example: zoom out $\quad x' = x/2 \quad y' = y/2$

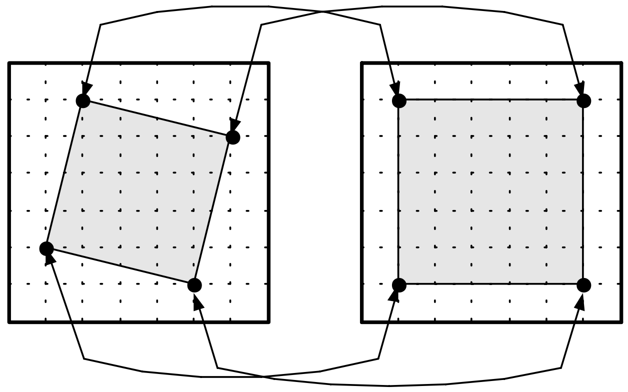
zoom in $\quad x' = 2x \quad y' = 2y$

5.6 Geometric Distortion Correction

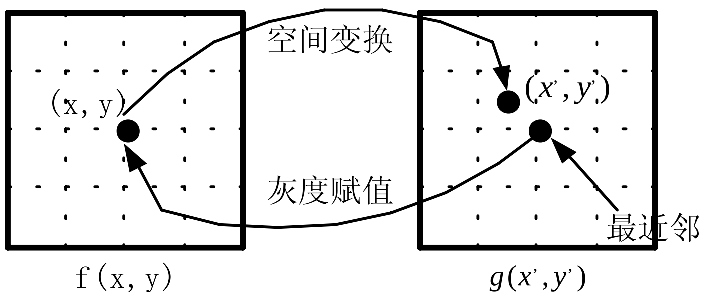
5.6.1 introduction

A geometric transformation consists of two basic operations:

(1) Spatial transformation



(2) Gray-level interpolation



Direct transform

Inverse transform

5.6 Geometric Distortion Correction

5.6.2 spatial transformations

Linear correction

$$r(x,y) = a_1 x + a_2 y + a_3$$
$$s(x,y) = b_1 x + b_2 y + b_3$$

Quadratic correction

$$r(x,y) = a_1 x^2 + a_2 y^2 + a_3 xy + a_4 x + a_5 y + a_6$$
$$s(x,y) = b_1 x^2 + b_2 y^2 + b_3 xy + b_4 x + b_5 y + b_6$$

5.6 Geometric Distortion Correction

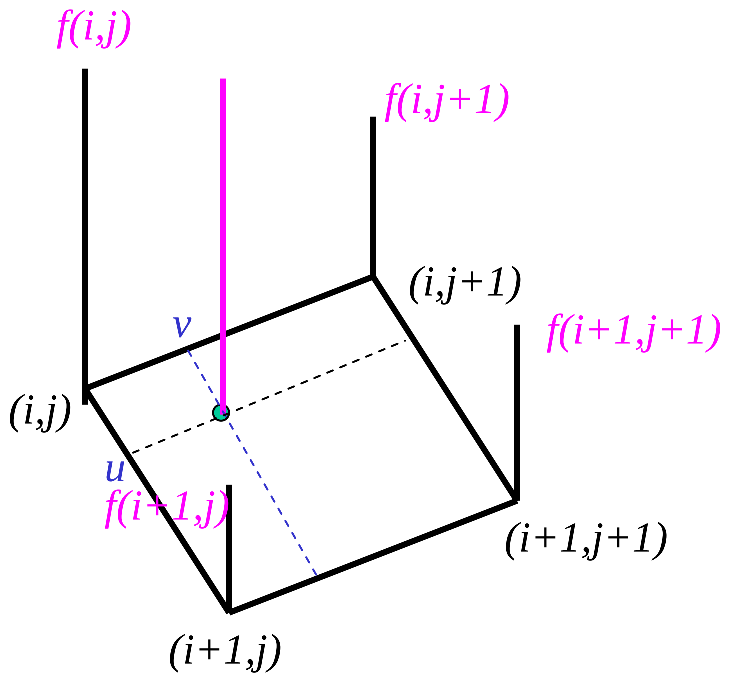
5.6.3 gray-level interpolation

- Nearest neighbor interpolation

$$g(x,y) = f(x',y') = f(i+u,\ j+v)$$

$$u,v \in (0,1) \qquad i,\ j \in Z$$

$$x' = round(x') \qquad y' = round(y')$$



5.6 Geometric Distortion Correction

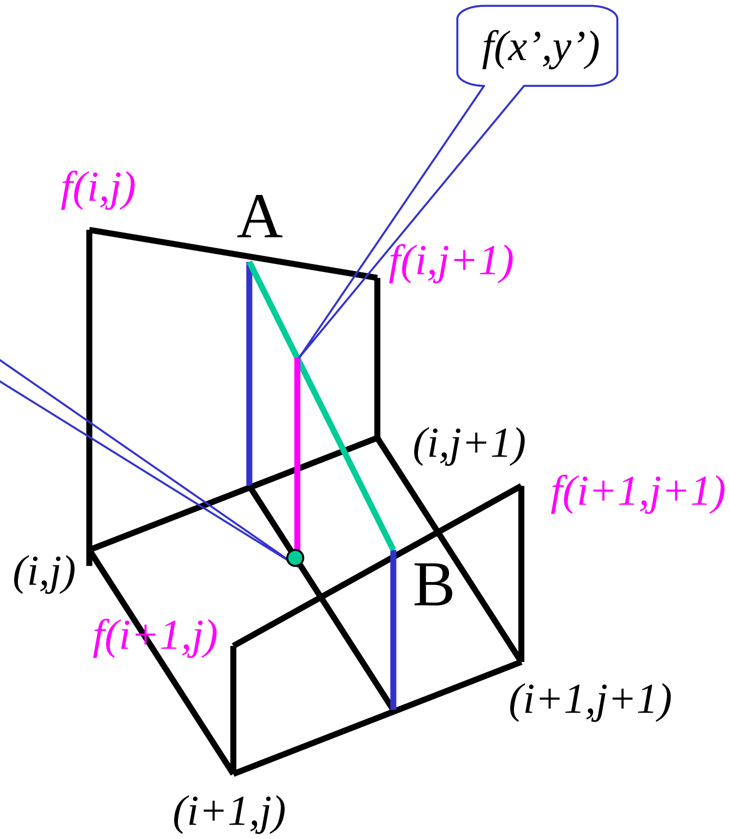
5.6.3 gray-level interpolation

- Bi-linear interpolation

$$f(x', y') = f(i+u, j+v)$$

$$u, v \in (0,1) \qquad i, j \in Z$$

$$= (1-u)(1-v)f(i,j) + (1-u)vf(i,j+1) + u(1-v)f(i+1,j) + uvf(i+1,j+1)$$



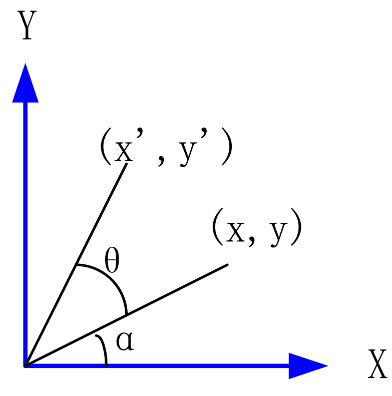
It can be operated by a mask

$$\begin{vmatrix} (1-u)(1-v) & (1-u)v \\ u(1-v) & uv \end{vmatrix}$$

5.6 Geometric Distortion Correction

5.6.3 gray-level interpolation

Rotation in 2D plane



$$x' = r\cos(\alpha+\theta) = r\cos\alpha\cos\theta - r\sin\alpha\sin\theta$$

$$y' = r\sin(\alpha+\theta) = r\cos\alpha\sin\theta + r\sin\alpha\cos\theta$$

The original coordinated of the point in polar coordinates are:

$$x = r\cos\alpha \qquad y = r\sin\alpha$$

Summary

- Inverse filtering is a very easy and accurate way to restore an image provided that we know what the blurring filter is and that we have no noise

- Wiener filtering is the optimal tradeoff between the inverse filtering and noise smoothing

- It is possible to restore an image without having specific knowledge of degradation filter and additive noise. However, not knowing the degradation filter **h** imposes the strictest limitations on our restoration capabilities.

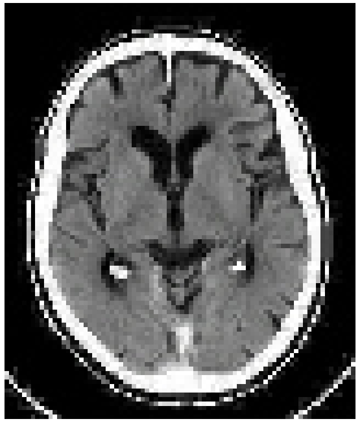
Chapter6 Image Reconstruction

- 6.1 Introduction
- 6.2 Reconstruction by Fourier Inversion
- 6.3 Reconstruction by convolution and backprojection
- 6.4 Finite series-expansion

Preview

CT reconstruction

CT machine

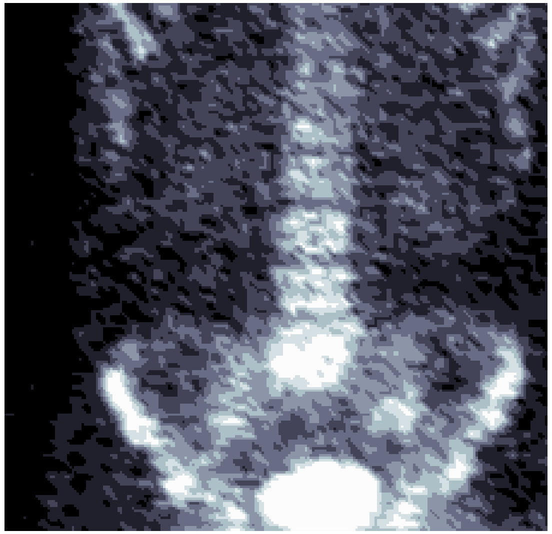


TCT

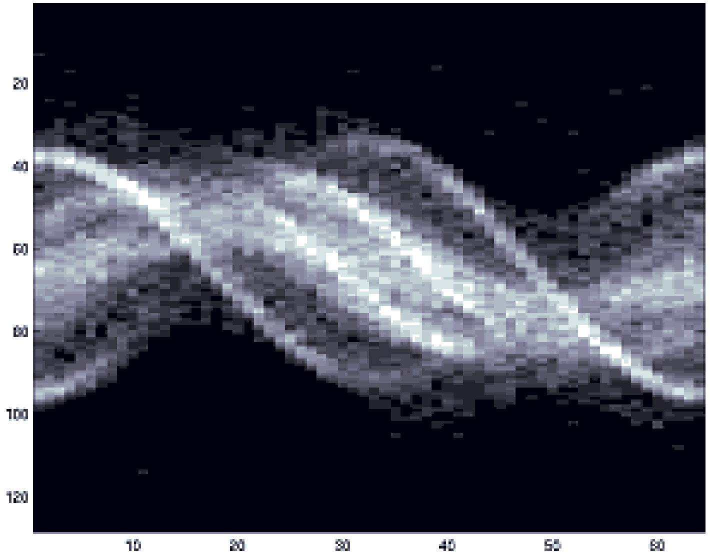
MRI

Preview

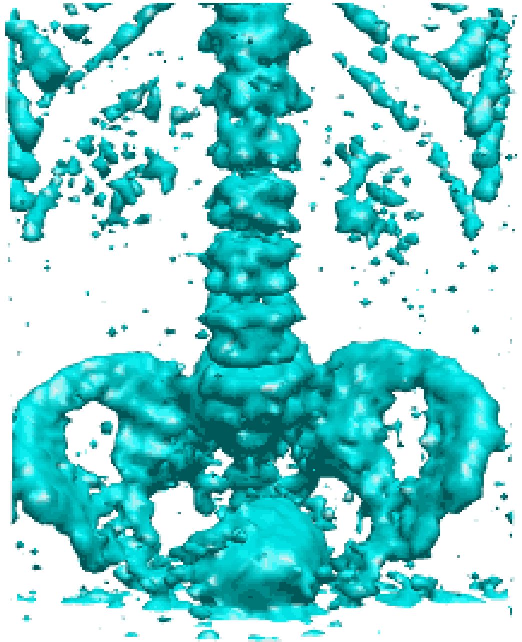
CT reconstruction



SPECT



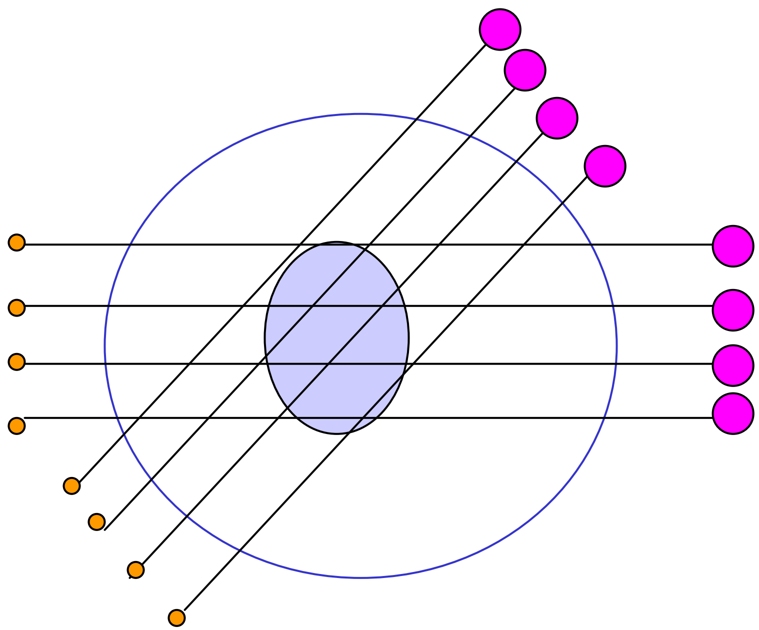
projections



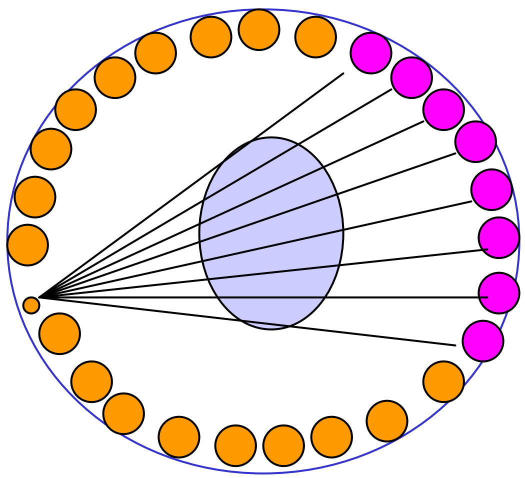
3D reconstruction

6.1 Introduction

6.1.1 Overview of Computed Tomography (CT)



parallel-beam



fan-beam

6.1 Introduction

6.1.2 Classification

(1) Transmission Computed Tomography TCT

(2) Emission Computed Tomography, ECT

(3) Reflection Computed Tomography, RCT

(4) Magnetic resonance imaging, MRI

6.1 Introduction

6.1.3 Physical basis of projection

Let us consider the simplest case, a single block of *homogeneous* tissue and a monochromatic beam of X-rays. The linear attenuation coefficient μ is defined by

$$I = I_0 e^{-\mu L}$$

where

L is the length of the block

I and I_0 are incident and attenuated intensities of the X-ray, respectively

6.1 Introduction

6.1.3 Physical basis of projection

Let $\mu(x,y)$ denote the sectional attenuation variation. For an infinitely thin beam of monochromatic X-rays, the detected intensity of the X-ray along a straight line ***L*** is expressed as

$$I = I_0 e^{-\int_L \mu(x,y)dl}$$

$$\Downarrow$$

$$-\ln\frac{I}{I_0} = \int_L \mu(x,y)dl$$

6.1 Introduction

6.1.4 Purpose

Reconstruction $\mu(x,y)$ from its projections

6.1.5 Methods

Fourier Inversion (frequency domain)

convolution and backprojection (spatial domain)

Finite series-expansion (spatial domain)

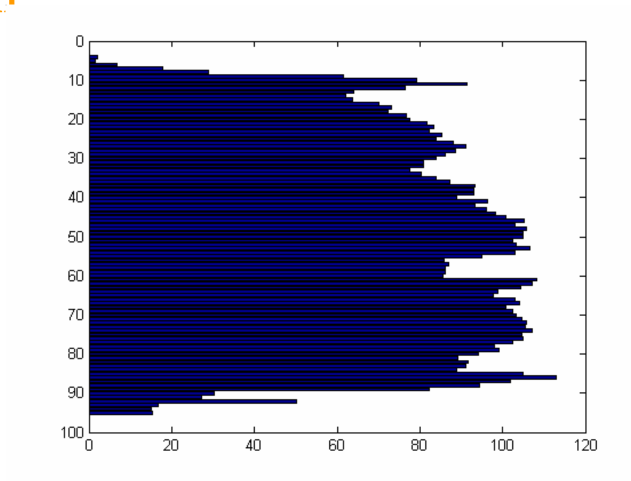
6.2 Reconstruction by Fourier Inversion

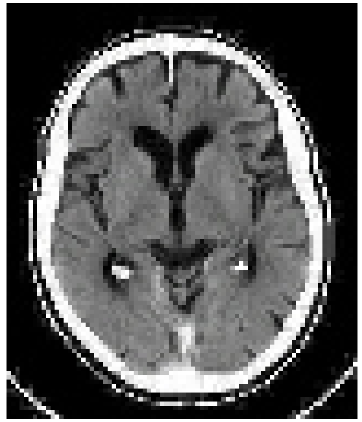
6.2.1 Mathematical expression of projection

Projection in x- and y-axial

$$p(x) = \int_{-\infty}^{\infty} f(x, y)dy$$

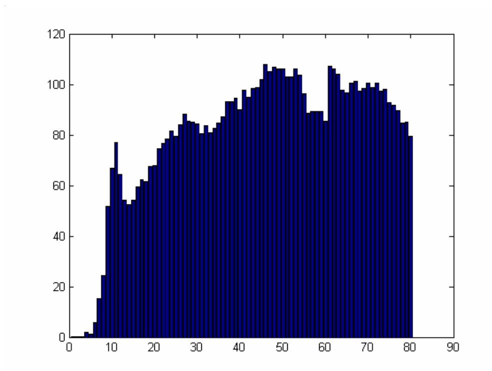
$$p(y) = \int_{-\infty}^{\infty} f(x, y)dx$$





Projection in arbitrary direction

$$p_{\theta}(t) = \int_{-\infty}^{\infty} f(x, y)ds$$



6.2 Reconstruction by Fourier Inversion

6.2.1 Mathematical expression of projection

where

$$t = y\sin\theta + x\cos\theta$$

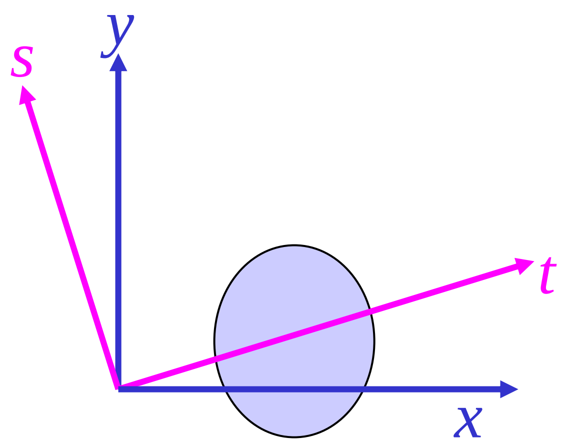
$$s = y\cos\theta - x\sin\theta$$

$$\Downarrow$$

$$x = t\cos\theta - s\sin\theta$$

$$y = t\sin\theta + s\cos\theta$$

$$\Downarrow$$



$$p_\theta(t) = \int_{-\infty}^{\infty} f(x, y)ds$$

$$= \int_{-\infty}^{\infty} f(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds$$

6.2 Reconstruction by Fourier Inversion

6.2.2 Fourier Slice Theorem

If *F(u,v)* is the Fourier Transform of *f(x,y)*, $p_\theta(t)$ is the projection of *f(x,y)* in θ *–direction*, and the $S_\theta(\omega)$ is the Fourier transform of $p_\theta(t)$ then, $S_\theta(\omega)$ is a slice of *F(u,v)* in the θ *–direction*

$$S_\theta(\omega) = F(\omega, \theta)$$

where $F(\omega, \theta)$ is the polar coordinate expression of *F(u,v)*

6.2 Reconstruction by Fourier Inversion

6.2.2 Fourier Slice Theorem: proof

When $\theta = 0$

$$p(x) = \int_{-\infty}^{\infty} f(x,y)dy \Rightarrow P(u) = \int_{-\infty}^{\infty} p(x)\exp(-2\pi jux)dx$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)dy\exp(-2\pi jux)dx$$

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp[-2\pi j(ux+vy)]dxdy$$

Let v=0 $\Rightarrow$ $F(u,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp(-2\pi jux)dxdy$

$$\therefore \quad F(u,0) = P(u)$$

In other words, the Fourier transform of the vertical projection of an image is the horizontal radial profile of the 2D Fourier transform of the image.

6.2 Reconstruction by Fourier Inversion

6.2.2 Fourier Slice Theorem **:**proof

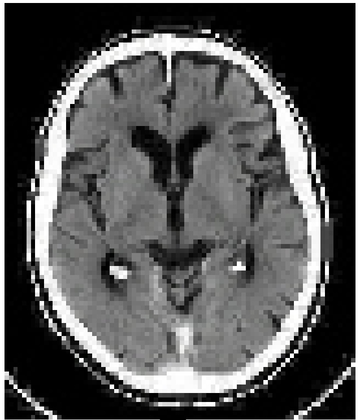
In general

By the nature of the Fourier transform, if an image ***f***(***x***,***y***) is rotated by an angle with respect to the ***x*** axis, the Fourier transform ***F***(***u***,***v***) will be correspondingly rotated by the same angle with respect to the ***u*** axis.

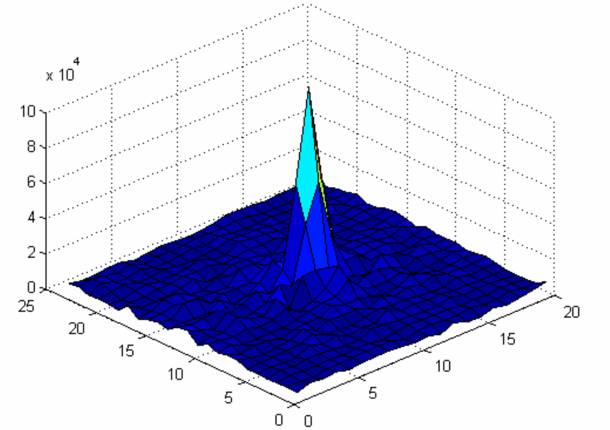
$$S_\theta(\omega) = F(\omega, \theta)$$

6.2 Reconstruction by Fourier Inversion

6.2.2 Fourier Slice Theorem

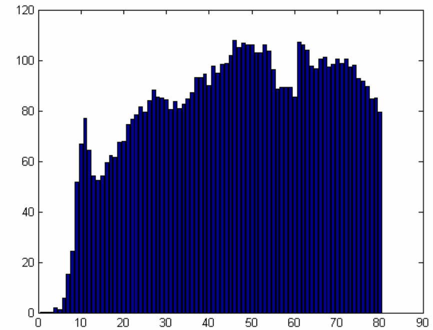


2D-DFT

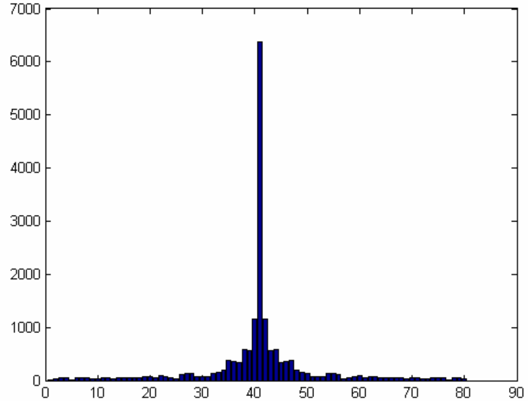


$$F(u,0) = P(u)$$

projection



1D-DFT



6.2 Reconstruction by Fourier Inversion

6.2.3 Arithmetic of Fourier inversion

Step1: calculate the projection's DFT $S_\theta(\omega)$ in θ_m –direction, $m=0,1, M-1$

Step2: combine $S_\theta(\omega)$ into $F(\omega, \theta)$

Step3: interpolation

Step4: 2D-IDFT

6.3 Reconstruction by convolution and backprojection

6.3.1 Parallel-Beam Reconstruction

With the inverse Fourier transform, an image *f(x,y)* can be expressed as

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}F(u,v)\exp\left[2\pi j(ux+vy)\right]dudv$$

Let $\qquad u=\omega\cos\theta \quad v=\omega\sin\theta$

we have

$$f(x,y)=\int_{0}^{2\pi}\int_{0}^{\infty}F(\theta,\omega)\exp\left[2\pi j(x\cos\theta+y\sin\theta)\omega\right]\omega d\omega d\theta$$

Because $\qquad F(\theta+\pi,\omega)=F(\theta,-\omega)$

we have

$$f(x,y)=\int_{0}^{\pi}\int_{-\infty}^{\infty}F(\theta,\omega)\left|\omega\right|\exp\left[2\pi j(x\cos\theta+y\sin\theta)\omega\right]d\omega d\theta$$

6.3 Reconstruction by convolution and backprojection

6.3.1 Parallel-Beam Reconstruction

Using the Fourier slice theorem, we have

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}S_\theta(w)\left|\omega\right|\exp\left[2\pi j(x\cos\theta+y\sin\theta)\omega\right]d\omega d\theta$$

let $t=x\cos\theta+y\sin\theta$

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}S_\theta(w)\left|\omega\right|\exp\left[2\pi j\omega t\right]d\omega d\theta$$

$$S_\theta(\omega)\left|\omega\right| \Leftrightarrow p_\theta(t)*h(t)$$

where

$$h(t)=F^{-1}(\left|\omega\right|) \Rightarrow f(x,y)=\int_0^{\pi}d\theta\int_{-\infty}^{\infty}p_\theta(t)h(t-\tau)d\tau$$

6.3 Reconstruction by convolution and backprojection

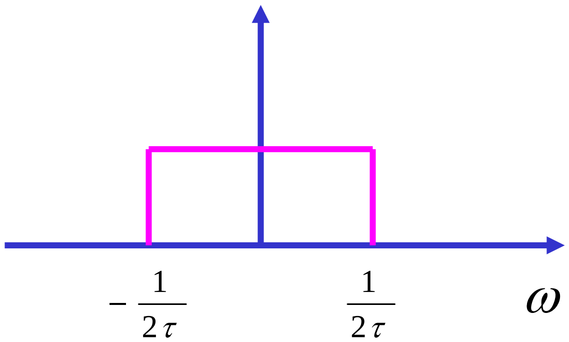
6.3.1 Parallel-Beam Reconstruction

Note that $h(t)$ does not exist in an ordinary sense, but $S_{\theta}(\omega)$ is essentially bandlimited, $h(t)$ can be accurately evaluated within the maximum bandwidth of $S_{\theta}(\omega)$.

$$h(t) = F^{-1}(|\omega|) \quad = \int_{-\omega c}^{\omega c} \omega \exp(2\pi j\omega t)\, d\omega$$

If $\omega c = \dfrac{1}{2\tau}$

$$h(t) = \frac{1}{2\tau^2}\left(\frac{\sin(2\pi t/2\tau)}{2\pi t/2\tau}\right) - \frac{1}{4\tau^2}\left(\frac{\sin(2\pi t/2\tau)}{2\pi t/2\tau}\right)^2$$



6.3 Reconstruction by convolution and backprojection

6.3.2 Arithmetic of Fourier inversion

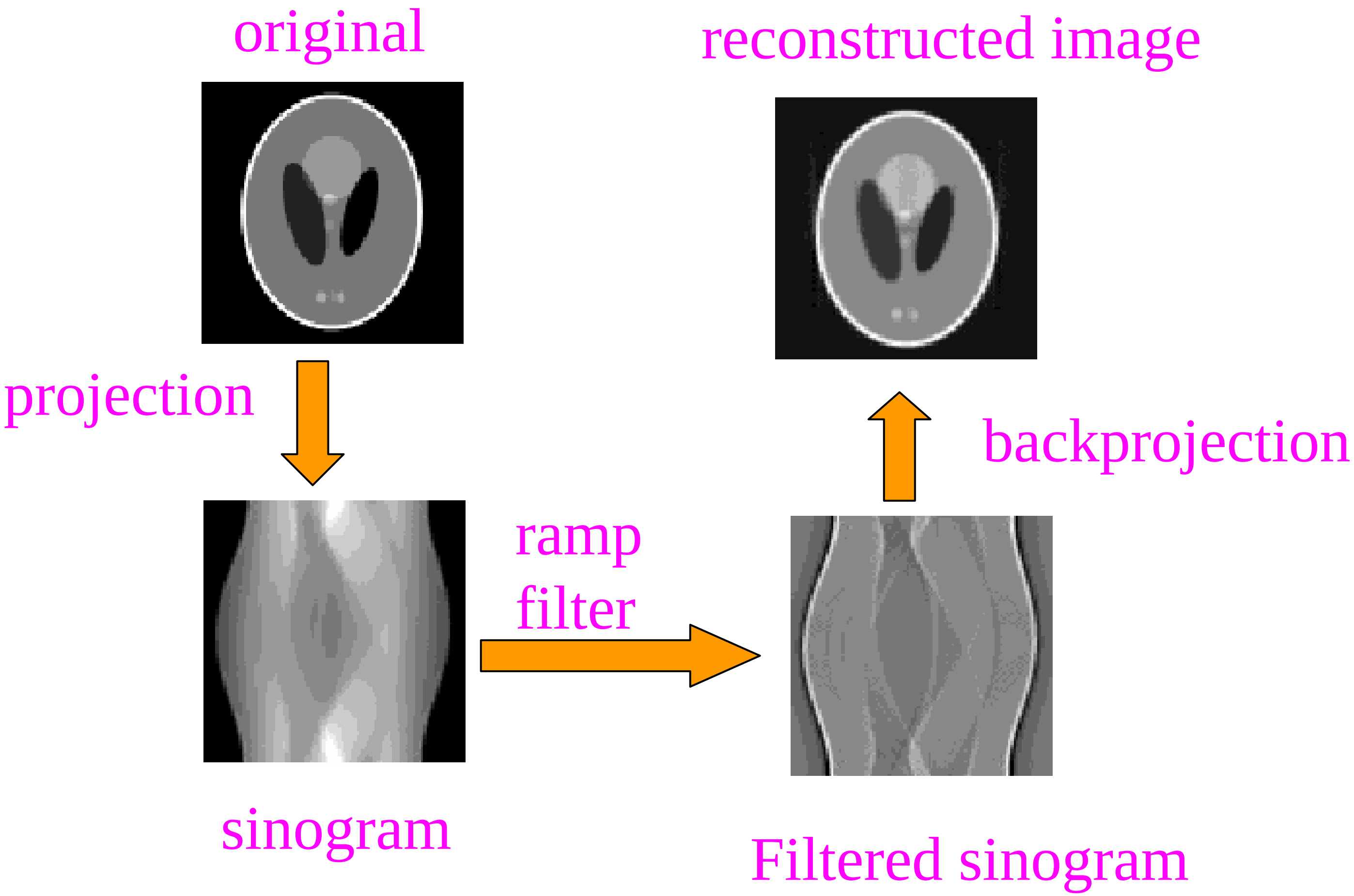
Step1: applying the filter *h(t)* to $p_\theta(t)$, get $p'_\theta(t)$,

Step2: calculate the integration

$$f(x,y) = \int_0^{\pi} p'_\theta(t)\,d\theta$$

6.3 Reconstruction by convolution and backprojection

6.3.3 Experimental results

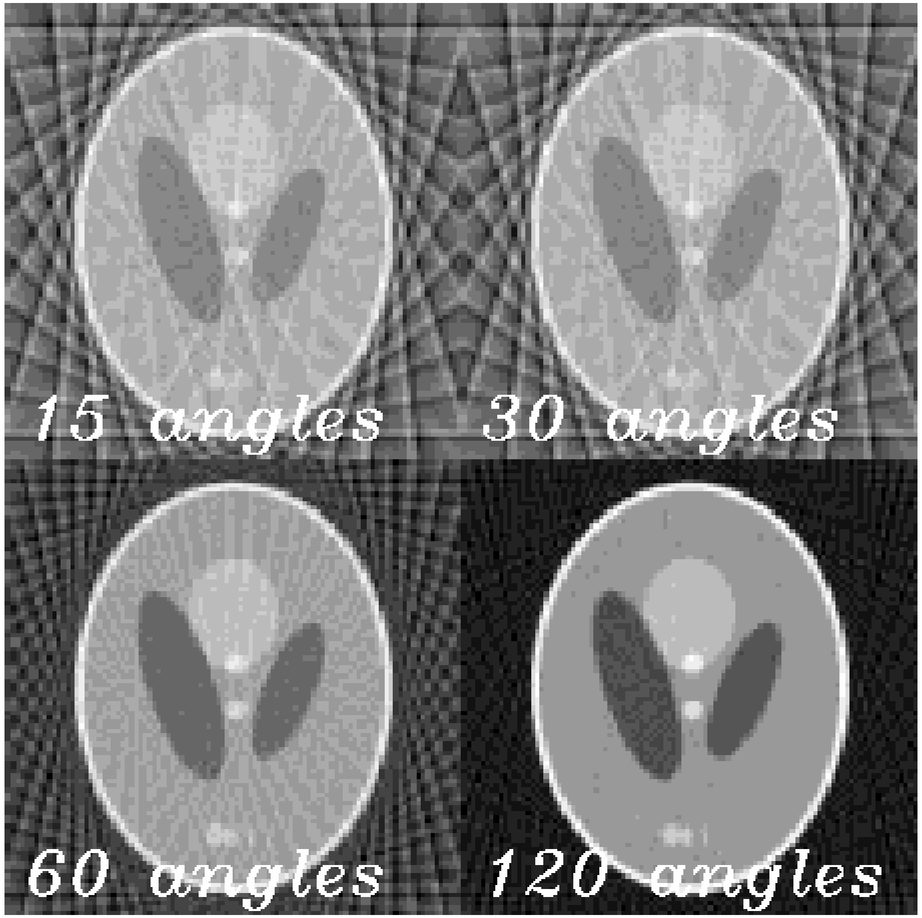


6.3 Reconstruction by convolution and backprojection

6.3.4 Practical problems: Aliasing - Insufficient angular sampling

•reducing scanning time

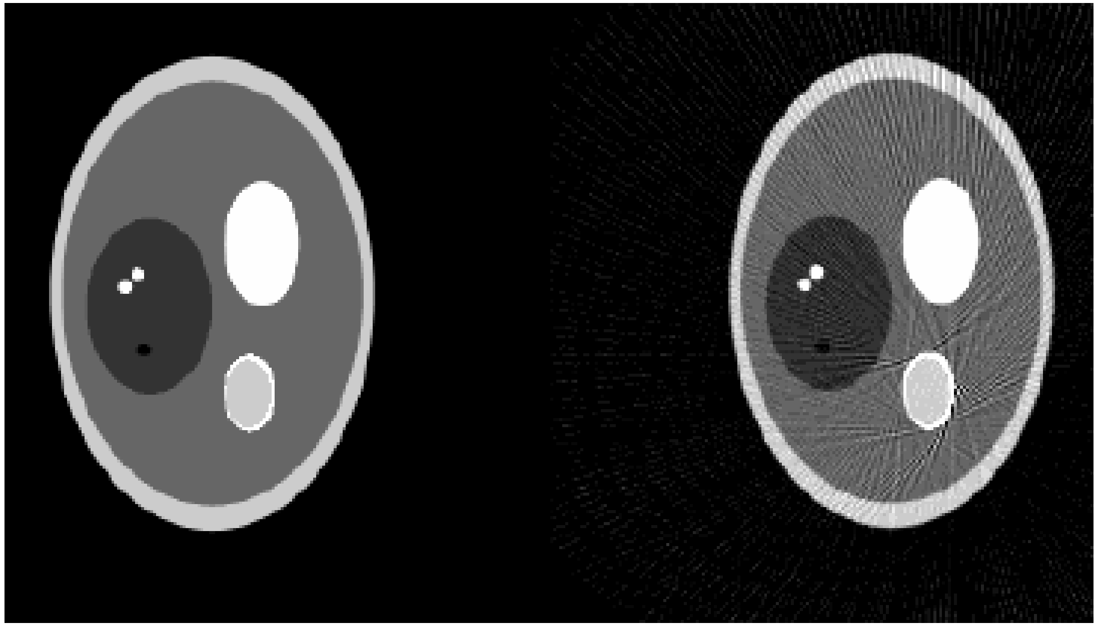
•reducing patient dose



6.3 Reconstruction by convolution and backprojection

6.3.4 Practical problems: Aliasing - Insufficient radial sampling

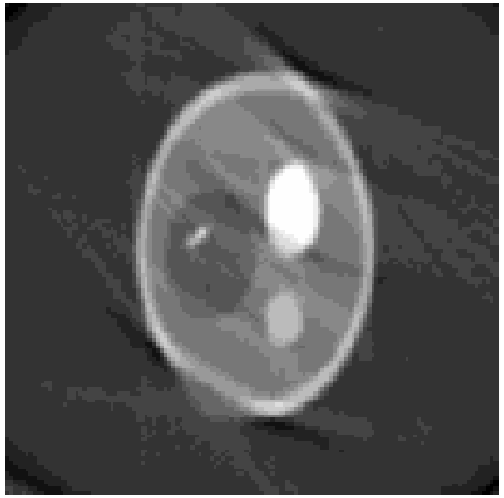
occurs when there is a sharp intensity change caused by, for example, bones.



6.3 Reconstruction by convolution and backprojection

6.3.4 Practical problems:Motion artifact

caused by patient motion, such as respiration and heart beat, during data acquistion



6.4 Finite series-expansion

6.4.1 Expression of Discrete projection

A 2-D array can be expressed as a vector

$$f(x,y) = \{ f_0, f_1 \cdots f_{N-1} \}$$

The projection of *i*-th ray is

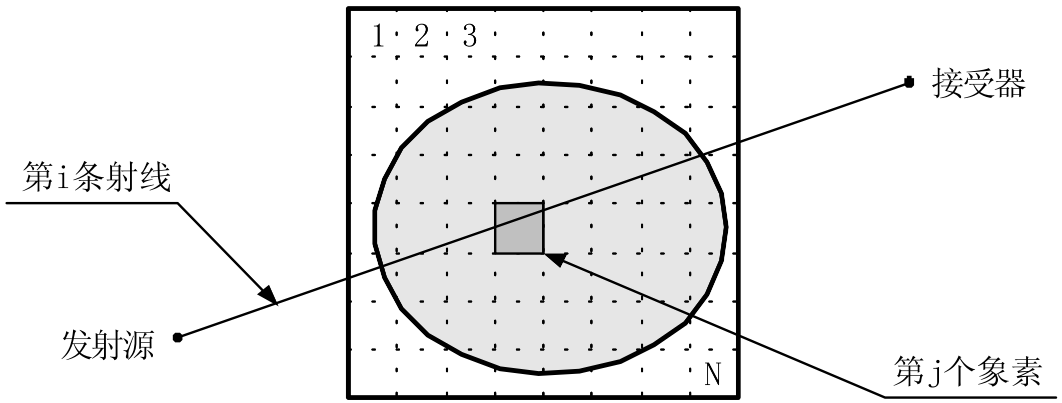
$$p_i = \sum_{j=0}^{N-1} w_{i,j} f_j \qquad i = 0,1,\cdots M-1$$

where

$$w_{i,j} = \begin{cases} 1 & \text{if the } i\text{-th ray cross the } j\text{-th point} \\ 0 & \text{else} \end{cases}$$

6.4 Finite series-expansion

6.4.1 Expression of Discrete projection



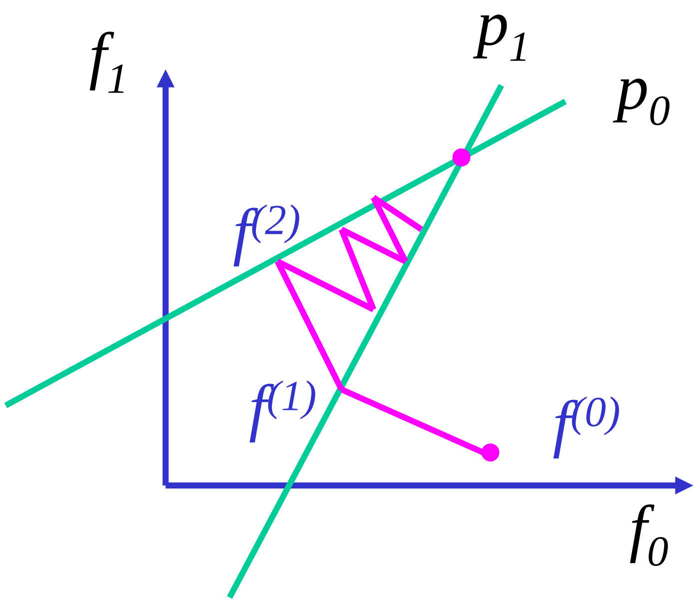
$$
\begin{cases}
w_{00} f_0 + w_{01} f_1 + \cdots + w_{0(N-1)} f_{N-1} = p_0 \\
w_{10} f_0 + w_{11} f_1 + \cdots + w_{1(N-1)} f_{N-1} = p_1 \\
\vdots \\
w_{(M-1)0} f_0 + w_{(M-1)1} f_1 + \cdots + w_{(M-1)(N-1)} f_{N-1} = p_{M-1}
\end{cases}
$$

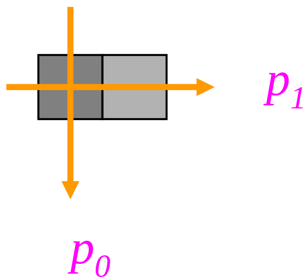
6.4 Finite series-expansion

6.4.2 Solution by iteration

f(x,y) can be seen as a point in a *N*-D space, each projection equation is a super-plane of this *N*-D space. If that equation set has a unique solution, then all the super-planes intersect in a point

For example *N=2*





$$\begin{cases} w_{00} f_0 + w_{01} f_1 = p_0 \\ w_{10} f_0 + w_{11} f_1 = p_1 \end{cases}$$

6.4 Finite series-expansion

6.4.2 Solution by iteration

$$f^{(1)} = f^{(0)} - \frac{w_0 f^{(0)} - p_0}{w_0 \bullet w_0} w_0$$

$$f^{(k+1)} = f^{(k)} - \frac{w_k f^{(k)} - p_k}{w_k \bullet w_k} w_k$$

6.4.3 Arithmetic of finite series-expansion

Step1: given an initial estimation $f^{(0)}$, $k=0$

Step2: adjust $f^{(k)}$ by a projection equation to get $f^{(k+1)}$, $k=k+1$

Step3: repeat step2 until the adjust value less than δ

Chapter7 Image Compression

- Preview
- 7.1 Introduction
- 7.2 Fundamental concepts and theories
- 7.3 Basic coding methods
- 7.4 Predictive coding
- 7.5 Transform coding
- 7.6 Introduction to international standards

7.1 Introduction

7.1.1 why code data?

- To reduce storage volume
- To reduce transmission time
  - One colour image
    - 760 by 580 pixels
    - 3 channels, each 8 bits
    - 1.3 Mbyte
  - Video data
    - same resolution
    - 25 frames per second
    - 33 Mbyte/second

7.1.2 Classification

Lossless compression: reversible

Lossy compression: irreversible

7.1 Introduction

7.1.3 Compression methods

- Entropy coding: Huffman coding, arithmetic coding
- Run length coding: G3, G4
- LZW coding: arj(DOS), zip(Windows), compress(UNIX)
- Predictive coding: linear prediction, non-linear prediction
- Transform coding: DCT, KLT, Walsh-T, WT(wavelet)
- Others: vector quantization (VQ),
  Fractal coding
  the second generation coding

7.2 Fundamental concepts and theories

7.2.1 Data redundancy

If n_1 denote the original data number, and n_2 denote the compressed data number, then the *data redundancy* can be defined as

$$R_D = 1 - \frac{1}{C_R}$$

where C_R, commonly called the *compression ratio*, is

$$C_R = \frac{n_1}{n_2}$$

For example $n_2 = n_1$ ⇨ $R_D = 0$

$n_2 \ll n_1$ ⇨ $C_R \to \infty, R_D \to 1$

7.2 Fundamental concepts and theories

7.2.1 Data redundancy

•Other expression of *compression ratio*

$$C_R = \frac{n_2}{n_1} \quad \text{or} \quad \overline{L} = \frac{n_2}{M \times N} bits / pixel, \quad C_R = \frac{8bits / pixel}{\overline{L}}$$

•Three basic image redundancy

Coding redundancy

Interpixle redundancy

psychovisual redundancy

7.2 Fundamental concepts and theories

7.2.1 Data redundancy: coding redundancy

If r_k represents the gray levels of an image and each r_k occurs with probability $p_r(r_k)$, then

$$p_r(r_k) = \frac{n_k}{n}, k = 0,1,\cdots L-1$$

where L is the number of gray levels, n_k is number of times that the kth gray level appears in the image, and n is the total number of pixels in the image

7.2 Fundamental concepts and theories

7.2.1 Data redundancy: coding redundancy

If the number of bits used to represent each value of r_k is $l(r_k)$,then the average number of bits required to represent each pixel is

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

For example

r_k	$P_r(r_k)$	*Code1*	$l_1(r_k)$	*Code2*	$l_2(r_k)$
r_0	0.19	000	3	11	2
r_1	0.25	001	3	01	2
r_2	0.21	010	3	10	2
r_3	0.16	011	3	001	3
r_4	0.08	100	3	0001	4
r_5	0.06	101	3	00001	5
R_6	0.03	110	3	000001	6
r_7	0.02	111	3	000000	7

7.2 Fundamental concepts and theories

7.2.1 Data redundancy: coding redundancy

$$L_{avg} = \sum_{k=0}^{7} l_2(r_k) p_r(r_k)$$

$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16)$$

$$+ 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02)$$

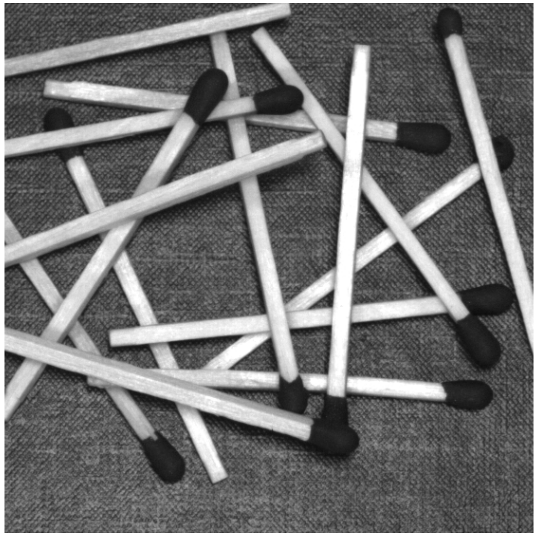
$$= 2.7 bits$$

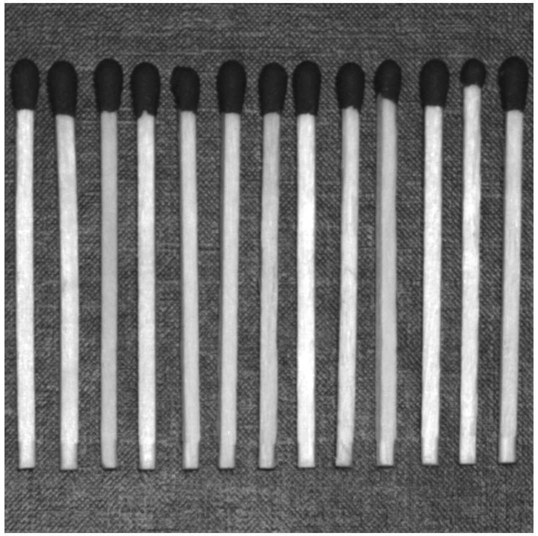
$$C_R = 3/2.7 = 1.11$$

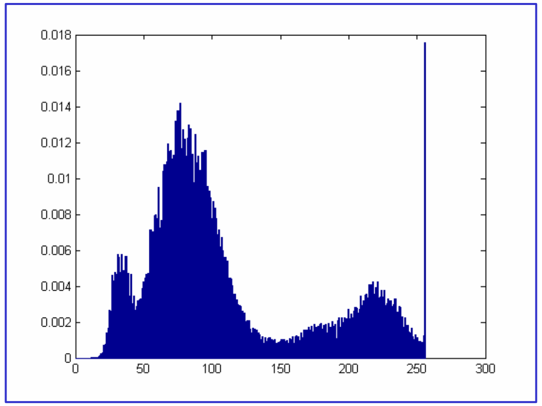
$$R_D = 1 - \frac{1}{1.11} = 0.099$$

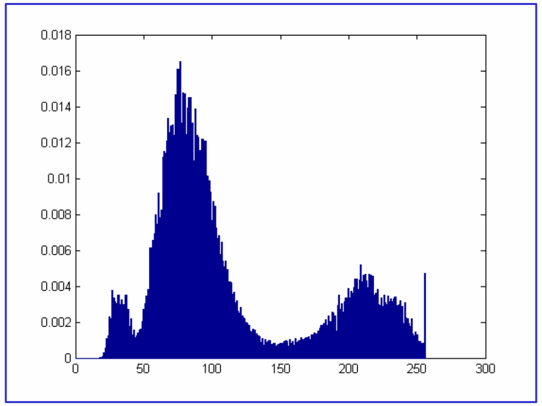
7.2 Fundamental concepts and theories

7.2.1 Data redundancy: interpixel redundancy









7.2 Fundamental concepts and theories

7.2.1 Data redundancy: interpixel redundancy

Autocorrelation coefficients

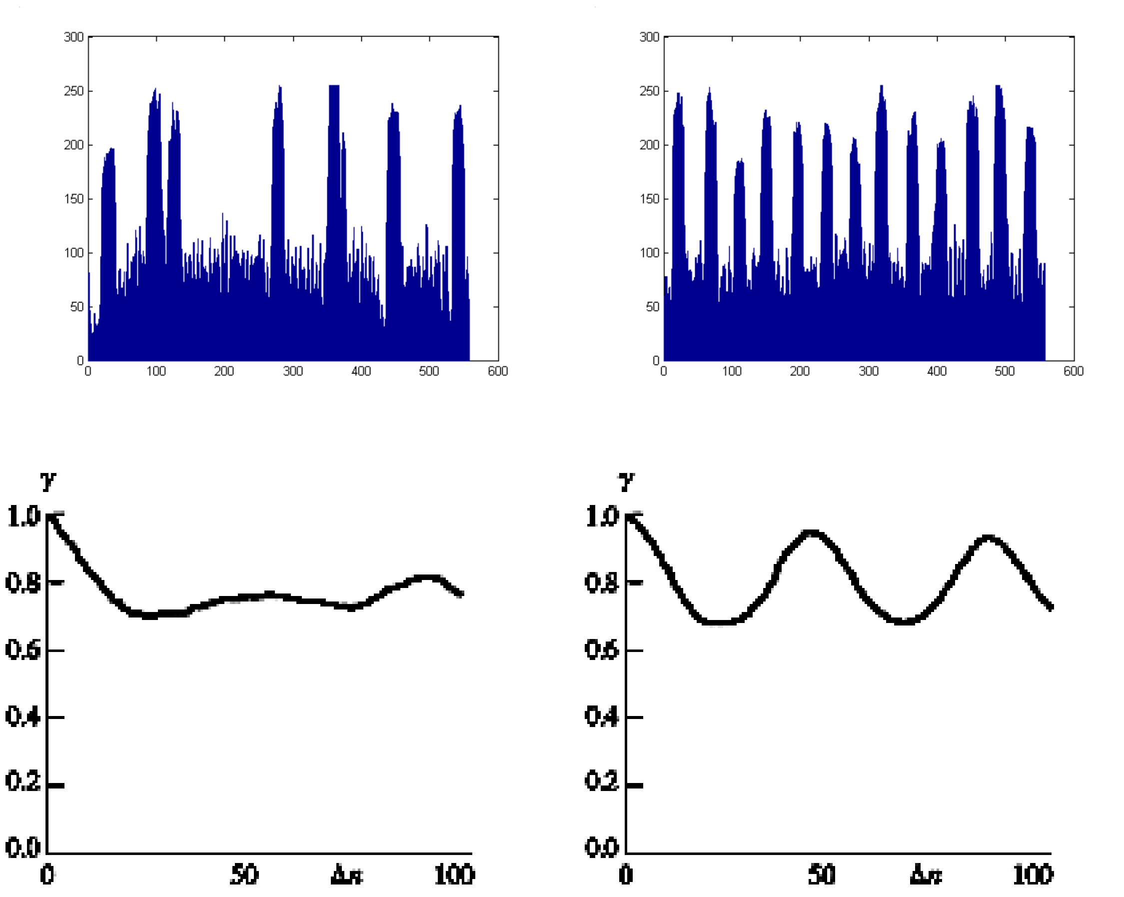
$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}$$

where

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n)$$

7.2 Fundamental concepts and theories

7.2.1 Data redundancy: interpixel redundancy



7.2 Fundamental concepts and theories

7.2.1 Data redundancy: psychovisual redundancy

- Psychovisual redundancy is associated with real or quantifiable visual information. The information itself is not essential for normal visual processing

8bits/pixel

4bits/pixel

2bits/pixel

7.2 Fundamental concepts and theories

7.2.2 Fidelity criteria: objective

Let $f(x,y)$ represent an input image and $\hat{f}(x,y)$ denote the Decompressed image. For any value *x* and *y*, the error *e(x,y)* bewteen $f(x,y)$ and $\hat{f}(x,y)$ can be defined as

$$e(x,y) = \hat{f}(x,y) - f(x,y)$$

So that the *total error* between the two images is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]$$

7.2 Fundamental concepts and theories

7.2.2 Fidelity criteria: objective

The *root-mean-square error*, e_{ems}, between $f(x,y)$ and $\hat{f}(x,y)$ is

$$e_{rms} = \{\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2\}^{1/2}$$

The *mean-square signal noise ratio* is defined as

$$SNR_{rms} = 10\lg \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2}$$

7.2 Fundamental concepts and theories

7.2.2 Fidelity criteria: objective

The *signal noise ratio* is defined as

$$SNR = 10\lg \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - \bar{f}(x,y) \right]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2}$$

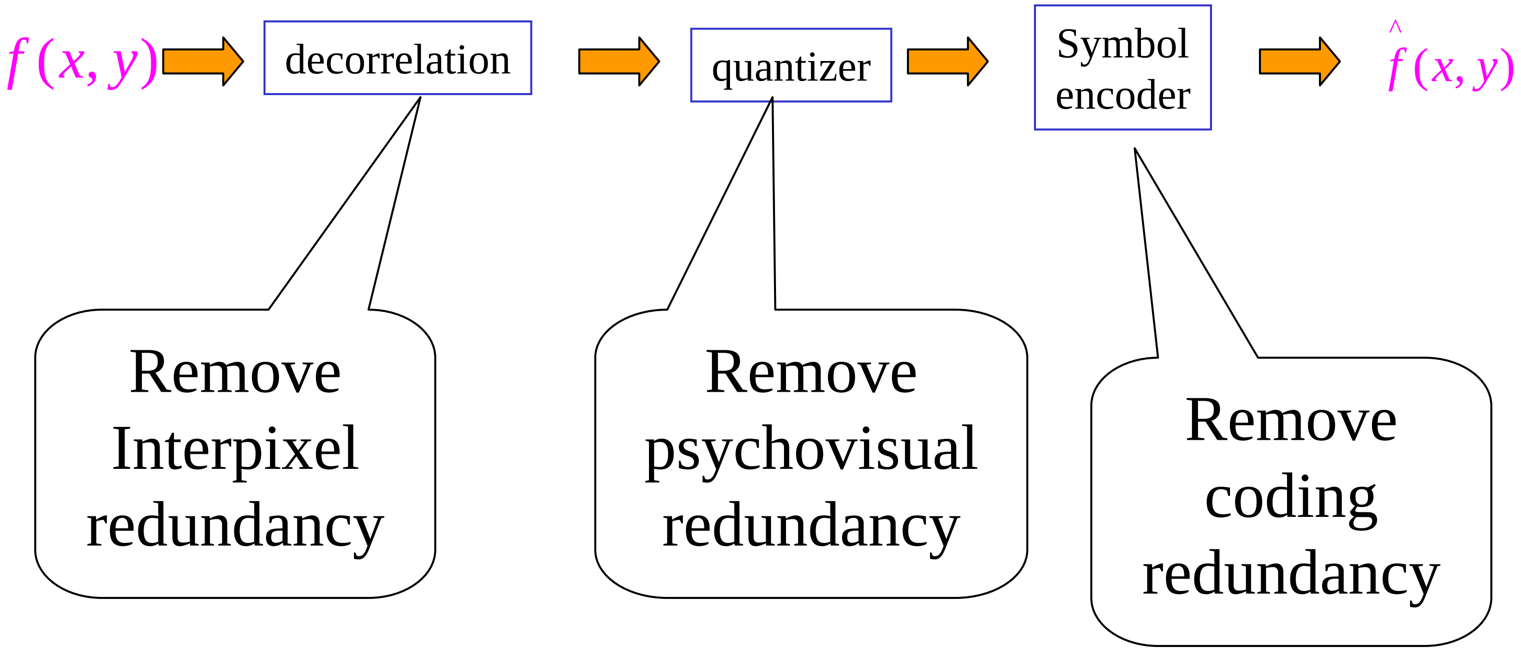
The *peak signal noise ratio* is defined as

$$PSNR = 10\lg \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} {f_{\max}}^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2} = 10\lg \frac{{f_{\max}}^2}{e_{rms}^2}$$

7.2 Fundamental concepts and theories

7.2.3 image compression models

Source encoder model



7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: terms

1948, Cloude Shannon, A mathematical theory of communication

News: data

information: contents

Information source: symbols

Memoryless source

Markov source

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Self-information

A random event E that occurs with probability $P(E)$ is said to contain $I(E)$ units of information.

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

For example $P(E) = 0, \quad I(E) = \infty$

$P(E) = 1, \quad I(E) = 0$

Unit: Base=2, bits (binary digits)

Base=*e*, nats (nature digits)

Base=10, Hartly

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Entropy of the source

definition

Suppose $X=\{x_0, x_1, \ldots x_{N-1}\}$ is a discrete random variable, and the probability of x_j is $P(x_j)$, namely

$$\begin{bmatrix} X \\ P(X) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \\ p(x_0) & p(x_1) & \cdots & p(x_1) \end{bmatrix}$$

then

$$H(X) = \sum_{j=0}^{N-1} p(x_j) \log \frac{1}{p(x_j)} = -\sum_{j=0}^{N-1} p(x_j) \log p(x_j)$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Entropy of the source

properties

(1) $H(X) \geq 0$

(2) $H(X) = H(1,0,\cdots 0) = H(0,1,\cdots 0) = \cdots = H(0,0,\cdots 1) = 0$

(3) $H(X) = H(x_0, x_1 \cdots x_{N-1}) \leq \log N$

When $P(x_j)=1/N$ for all j,, $H(X)=logN$

example $X=\{1,2,3,4,5,6,7,8\}$, $p_j=1/8$ for each j

$$H(X) = \sum_{j=1}^{8} \frac{1}{8} \log_2 8 = 3 bits / symbol$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Conditional entropy

definitions

$$H_0(X) = \sum_i p(x_i) \log \frac{1}{p(x_i)}$$

$$H_1(X) = \sum_i p(x_i) \sum_i p(x_i \mid x_{i-1}) \log \frac{1}{p(x_i \mid x_{i-1})}$$

$$H_2(X) = \sum_i p(x_i) \sum_i p(x_i \mid x_{i-1} x_{i-2}) \log \frac{1}{p(x_i \mid x_{i-1}\, x_{i-2})}$$

$$\vdots$$

$$H_m(X) = \sum_i p(x_i) \sum_i p(x_i \mid x_{i-1} \cdots x_{i-m}) \log \frac{1}{p(x_i \mid x_{i-1} \cdots x_{i-m})}$$

property

$$H_\infty = \cdots = H_m < H_{m-1} < \cdots < H_2 < H_1 < H_0$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: using information theory

A simple 8-bits image:

21	**21**	**21**	**95**	**169**	**243**	**243**	**243**
21	**21**	**21**	**95**	**169**	**243**	**243**	**243**
21	**21**	**21**	**95**	**169**	**243**	**243**	**243**
21	**21**	**21**	**95**	**169**	**243**	**243**	**243**

Probability of the source symbols:

Gray-levelGa	Count	probability
21	12	3/8
95	4	1/8
169	4	1/8
243	12	3/8

Zero-order entropy:

$$H_0(X) = \sum_i p(x_i) \log \frac{1}{p(x_i)}$$

$$= 1.81 \, bits / pixel$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: using information theory

Relative frequency of pairs of pixels:

Gray-level	Count	probability
(21,21)	8	2/8
(21,95)	4	1/8
(95,169)	4	1/8
(169,243)	4	1/8
(243,243)	8	2/8
(243,21)	4	1/8

First-order entropy:

$$H_1(X) = \sum_i p(x_i) \sum_i p(x_i \mid x_{i-1}) \log \frac{1}{p(x_i \mid x_{i-1})}$$

$$= 1.25 bits / pixel$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Noiseless coding theorem

Also called *Shannon first theorem*. Defines the minimum average code word length per source symbol that can be achieved

$$\boxed{H(X) \le L_{avg}}$$

For memoryless (zero-memory) source:

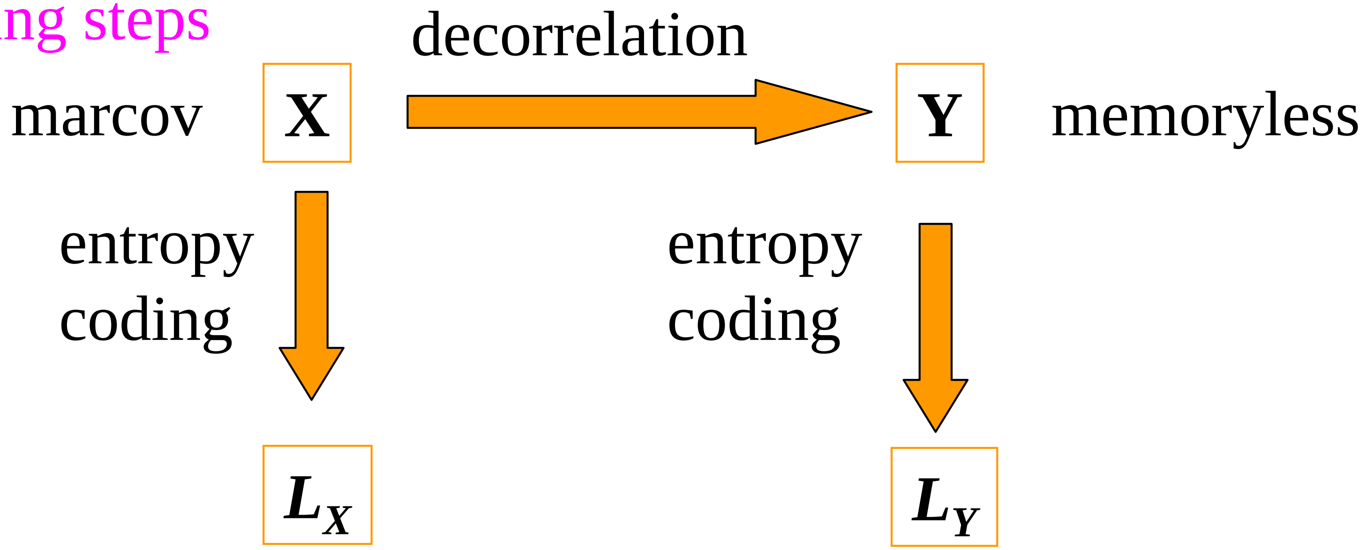
$$H(X) = \sum_i p(x_i) \log \frac{1}{p(x_i)}$$

But, the image data usually are Markov source, how to code?

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Noiseless coding theorem

Coding steps



$$H(X)=H_m(X)<H_0(X), \qquad L_X= H_0(X)$$

$$H(Y)=H_0(Y),\ L_Y= H_0(Y)$$

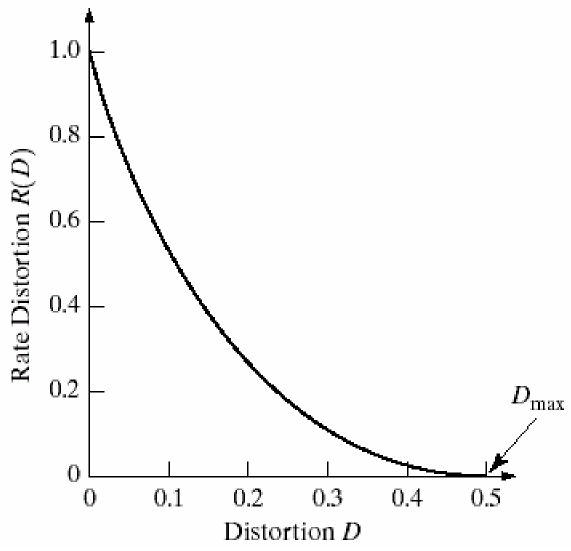
$$H(X)=H(Y) \Longrightarrow L_Y= H_m(X)<L_X$$

7.2 Fundamental concepts and theories

7.2.4 Elements of information theory: Noise coding theorem

For a given rate distortion *D*, the rate distortion function *R(D)* is less than *H(X)*, and *R(D)* is the minimum of coding length L_{avg}

FIGURE 8.10 The rate distortion function for a binary symmetric source.



7.3 Basic coding methods

7.3.1 Huffman coding

Huffman, 1952

- code the low probabilities symbols with long word
  code the high probabilities symbols with short word

- smallest possible number of code symbols per source symbol

- the source symbols must be coded *one at a time*

7.3 Basic coding methods

7.3.1 Huffman coding

step1:create a series of source reduction by ordering the probabilities

step2:combine the lowest probability symbols into a single symbol that replaces them in the next source reduction

step3:repeat step2 until the lowest probability is 1

For example

Symbol	Probability	1	2	3	4
A	0.4	0.4	0.4	0.4	0.6
B	0.3	0.3	0.3	0.3	0.4
C	0.1	0.1	0.2	0.3	
D	0.1	0.1	0.1		
E	0.06	0.1			
F	0.04				

7.3 Basic coding methods

7.3.1 Huffman coding

step4:code each reduced source with 0 and 1, staring with the smallest source and working back to the original source

Symbol	Probability		1		2		3		4	
A	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
B	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
C	0.1	011	0.1	011	0.2	010	0.3	01		
D	0.1	0100	0.1	0100	0.1	011				
E	0.06	01010	0.1	0101						
F	0.04	01011								

$$L_{avg} = 0.4\times 1 + 0.3\times 2 + 0.1\times 3 + 0.1\times 4$$
$$+ 0.06\times 5 + 0.04\times 5$$
$$= 2.2 bits/symbol$$

$$H_0(A) = -\sum_{i=1}^{6} p(a_i)\log p(a_i)$$
$$= 2.14 bits/symbol$$

$$L_{avg} > H_0(X)$$

7.3 Basic coding methods

7.3.2 Arithmetic coding

•An entire sequence of source symbols(or message) is assigned as a single arithmetic code word

• the code word itself defines an interval of real numbers between 0 and 1

The encoding process of an arithmetic coding can be explained through the following example

7.3 Basic coding methods

7.3.2 Arithmetic coding: example

encoding

- Let us assume that the source symbols are {$a_1\ a_2\ a_3 a_4$} and the probabilities of these symbols are {0.2, 0.2, 0.4, 0.2}
- To encode a message of a sequence: $a_1\ a_2\ a_3 a_3 a_4$
- The interval [0,1) can be divided as four sub-intervals: [0.0, 0.2), [0.2, 0.4), [0.4, 0.8), [0.8, 1.0),

symbols	probabilites	Initial intervals
a_1	0.2	[0.0,0.2)
a_2	0.2	[0.2,0.4)
a_3	0.4	[0.4,0.8)
a_4	0.2	[0.8,1.0)

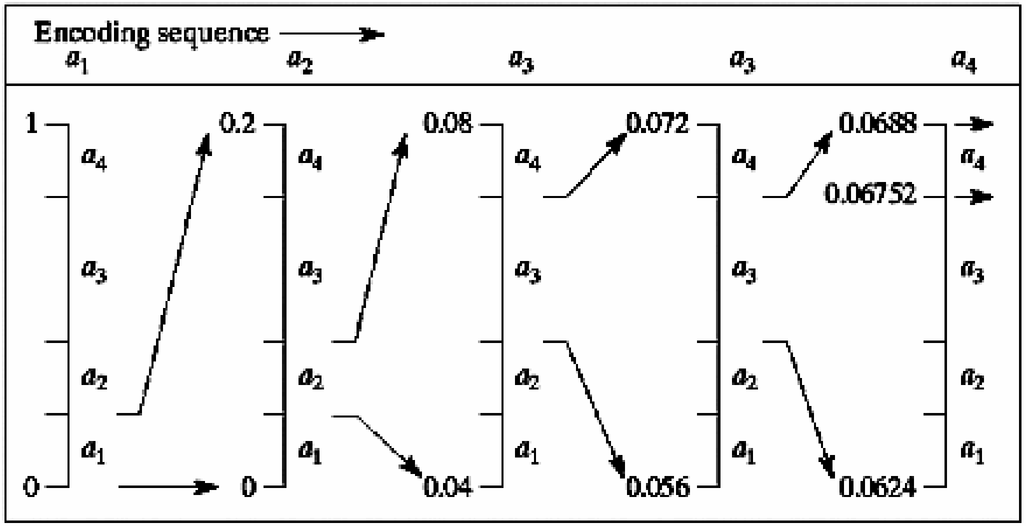
7.3 Basic coding methods

7.3.2 Arithmetic coding: example

•We take the first symbol a_1 from the message and find its encoding range is [0.0, 0.2).

•The second symbol *a2*is encodedby taking the 20th-40th of interval [0.0, 0.2) as the new interval [0.02,0.04).

•And so on. Visually, we can use the figure:



•Finally, choose a number from the interval of [0.0688,0.06752) as the output: 0.06800

7.3 Basic coding methods

7.3.2 Arithmetic coding : example

$\boxed{\text{decoding}}$

$0 < 0.068800 < 0.2 \Rightarrow a_1$

$0.04 < 0.068800 < 0.08 \Rightarrow a_2$

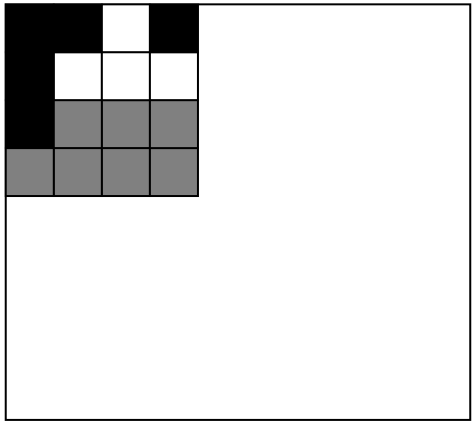
$0.056 < 0.068800 < 0.072 \Rightarrow a_3$

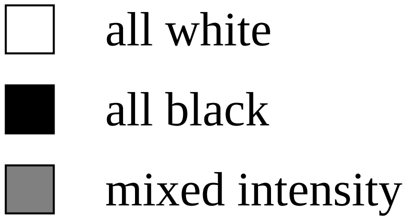
$0.0624 < 0.068800 < 0.0688 \Rightarrow a_3$

$0.06752 < 0.068800 < 0.0688 \Rightarrow a_4$

7.3 Basic coding methods

7.3.4 Binary image coding: constant area coding





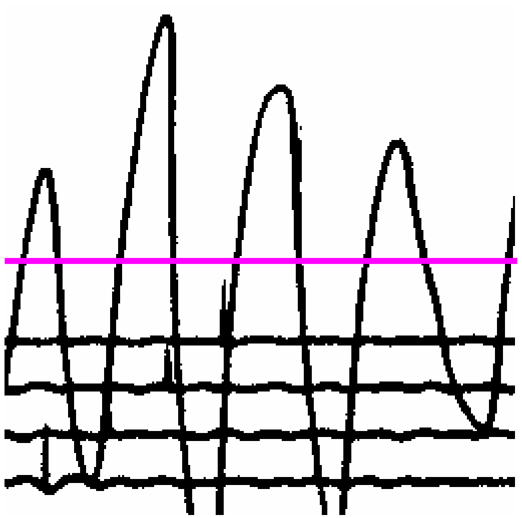
- The most probable category is coded with 1-bit code word 0
- The other two category is coded with 2-bit code word 10 and 11
- The code assigned to the mixed intensity category is used as a prefix

7.3 Basic coding methods

7.3.4 Binary image coding: run length coding

- Represent each row of a image by a sequence of lengths that describe runs of black and white pixels

- the standard compression approach in facsimile (FAX) coding. CCITT, G3,G4.



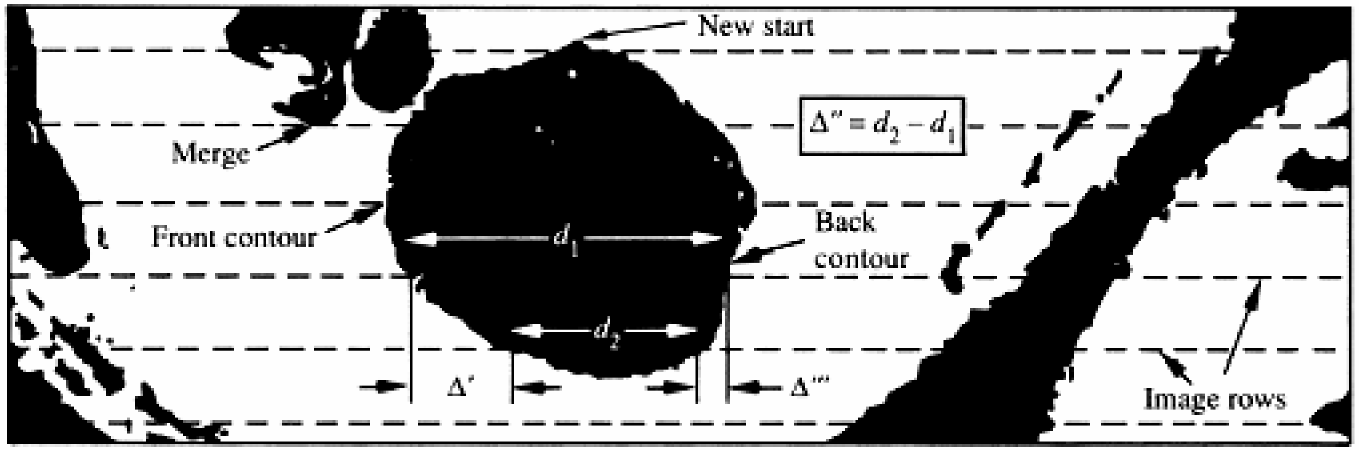
(1,8)	**(0,3)**	**(1,14)**	**(0,4)**	**(1, 28)**	**(0,3)**	**(1,23)**	**(0,3)**	**(1,29)**
(0, 3)	**(1,28)**	**(0,3)**	**(1,31)**	**(0,4)**	**(1,25)**	**(0,4)**	**(1,38)**	**(0,4)**

8	**3**	**14**	**4**	**28**	**3**	**23**	**3**	**29**	**3**	**28**
9	**3**	**31**	**4**	**25**	**4**	**38**	**4**			

7.3 Basic coding methods

7.3.4 Binary image coding: contour tracing and coding

- Δ' is the difference between the starting coordinates of the front contours adjacent lines
- Δ'' is the difference between the front-to-back contour lengths contours adjacent lines
- code: new start, Δ' and Δ''



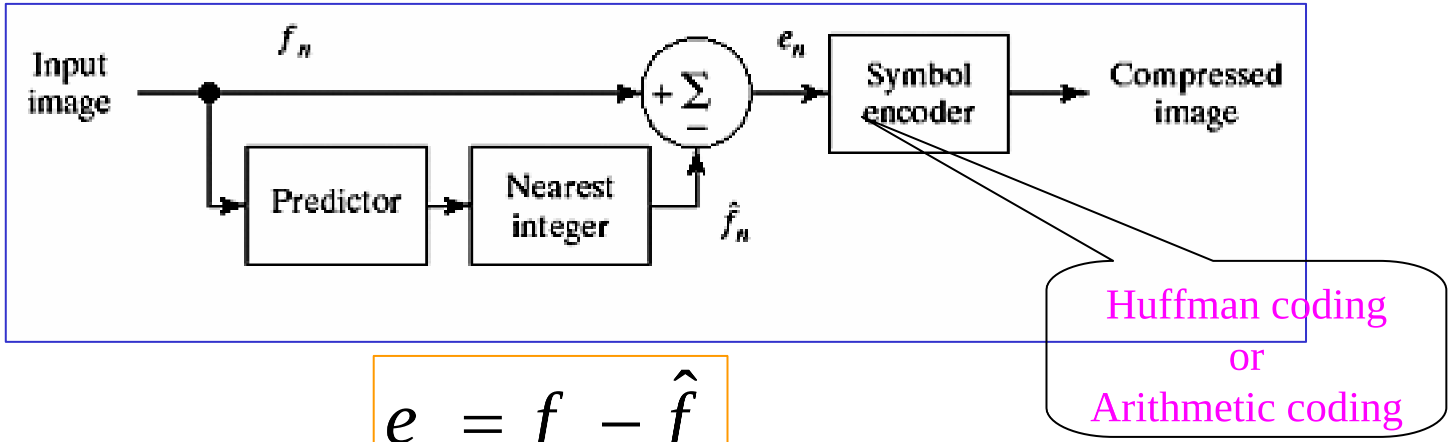
But for gray image…

162	161	159	161	162	160	158	156	156	161
162	161	159	161	162	160	158	156	156	161
163	158	159	159	160	158	155	155	156	158
159	157	159	156	159	159	154	152	155	153
155	157	156	156	158	157	156	155	154	156
156	157	155	151	157	156	155	156	156	154
157	156	156	156	156	156	154	156	155	155
158	157	155	155	156	155	155	155	155	155
156	155	156	153	156	155	156	155	154	156
155	155	157	154	157	155	157	158	158	158

7.4 Predictive coding

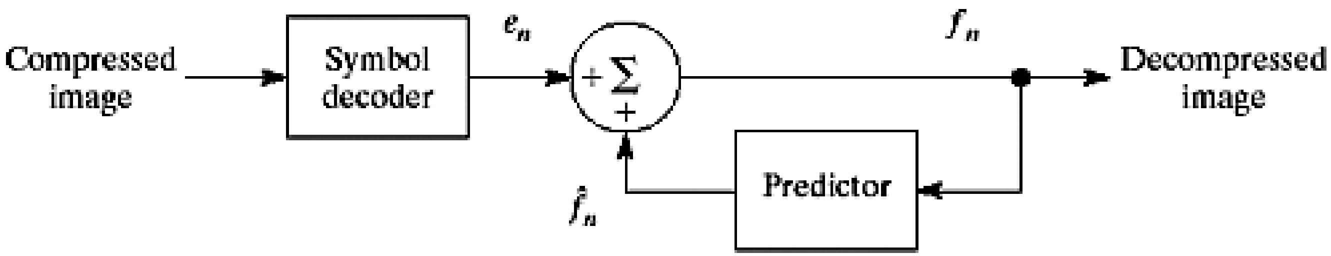
7.4.1 lossless coding

encoder



$$e_n = f_n - \hat{f}_n$$

decoder



$$f_n = e_n + \hat{f}_n$$

7.4 Predictive coding

7.4.1 lossless coding

In most case, the prediction is formed by a linear combination of m previous pixels. That is,

$$\hat{f}_n = round\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

where m is the order of the linear predictor

α_i are prediction coefficients

For example, 1-D linear predictive can be written

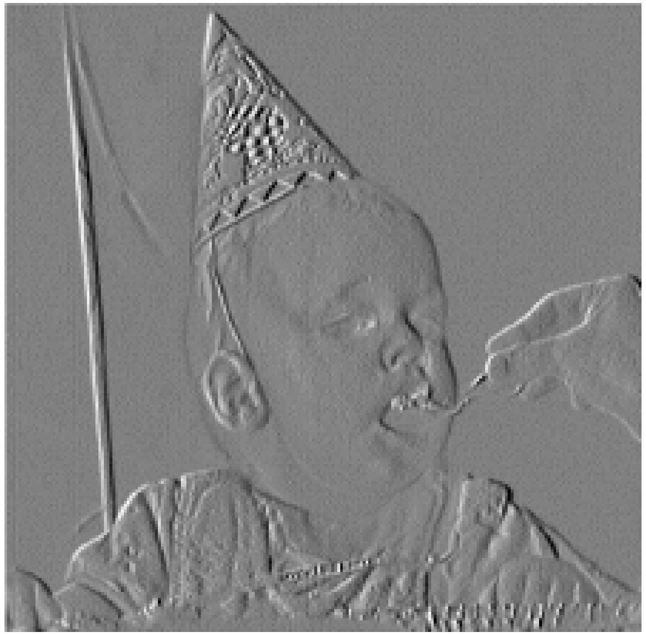
$$\hat{f}(x,y) = round\left[\sum_{i=1}^{m} \alpha_i f(x, y-i)\right]$$

7.4 Predictive coding

7.4.1 lossless coding: Experimental result

m=1, α =1 $$\hat{f}(x,y) = round\left[f(x,y-1) \right]$$

Original image



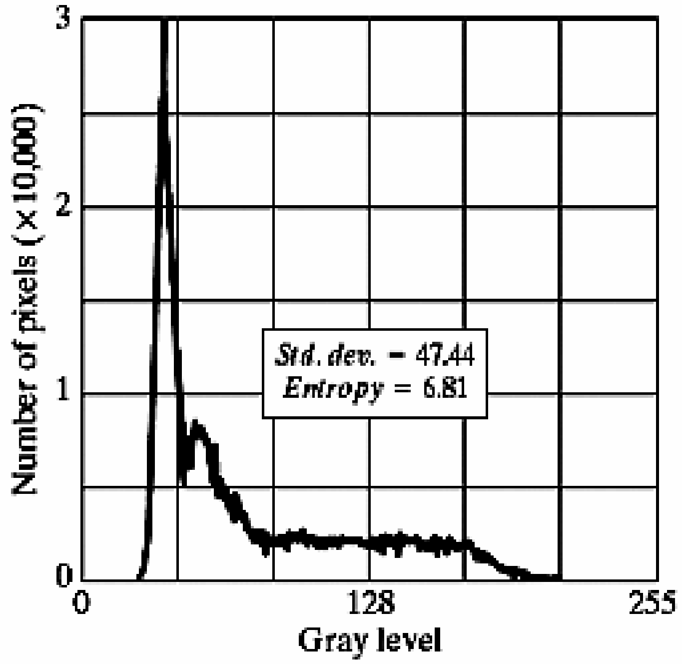
Residual image
(128 represents 0)

7.4 Predictive coding

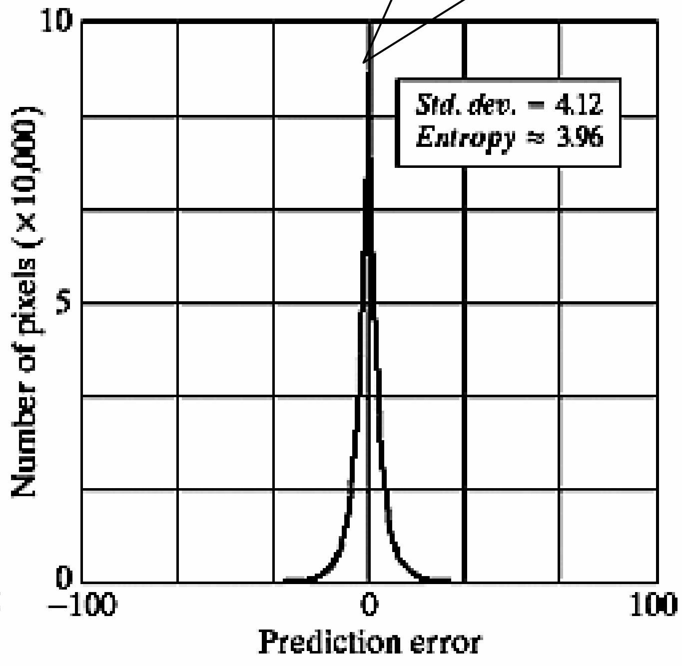
7.4.1 lossless coding: Experimental result

m=1, α =1

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$



Histogram of original image



Histogram of residual image

7.4 Predictive coding

7.4.1 lossless coding: JPEG lossless compression standard

c	***b***
a	***x***

Location of pixels

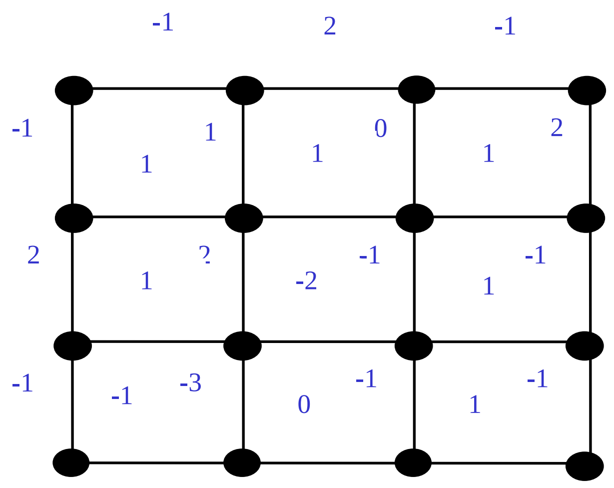
Prediction schemes

Prediction Scheme	**Prediction value**
0	Null
1	a
2	b
3	c
4	$a+b-c$
5	$a+(b-c)/2$
6	$b+(a-c)/2$
7	$(a+b)/2$

7.4 Predictive coding

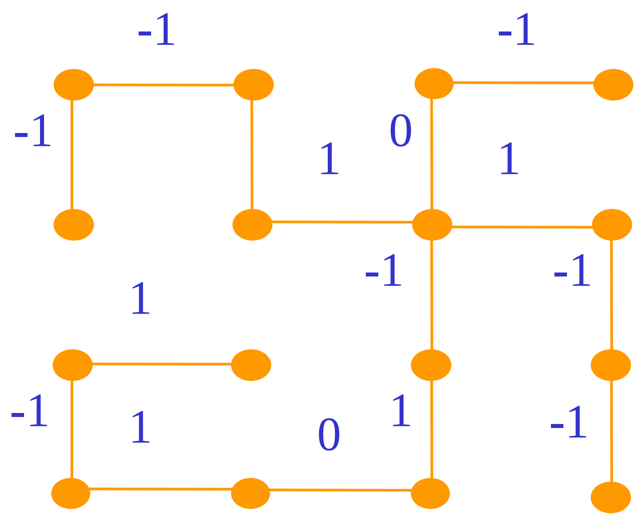
7.4.1 lossless coding: predictive tree

21	20	22	21
20	21	22	23
22	23	21	22
21	20	20	21

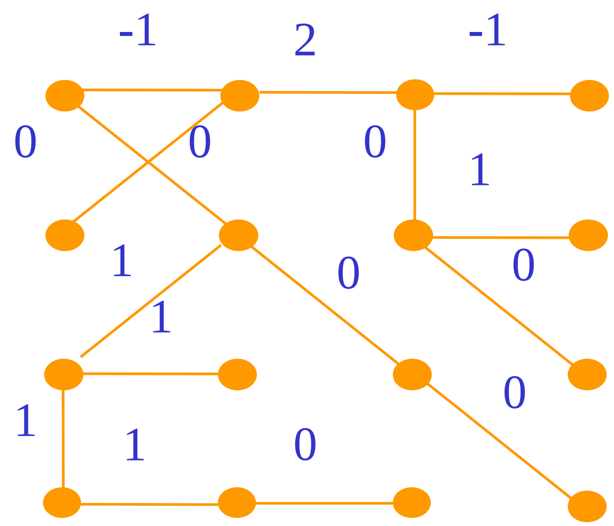


7.4 Predictive coding

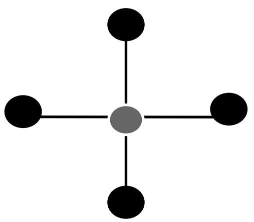
7.4.1 lossless coding: predictive tree

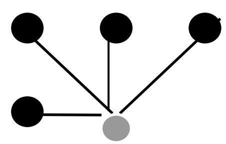


MAW预测树



SNMAW预测树

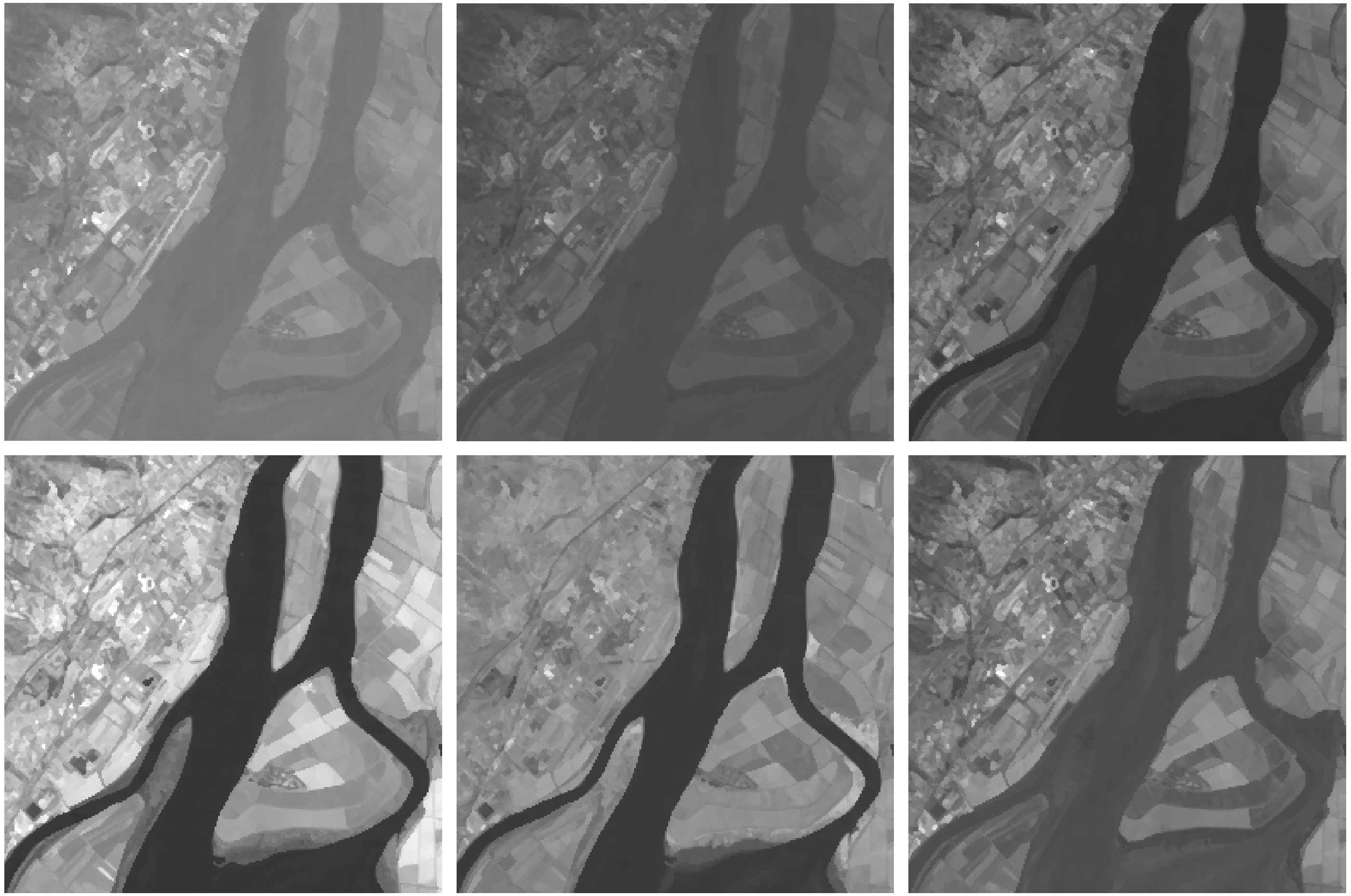




Cost?

7.4 Predictive coding

7.4.1 lossless coding: predictive tree



7.4 Predictive coding

7.4.1 lossless coding: predictive tree

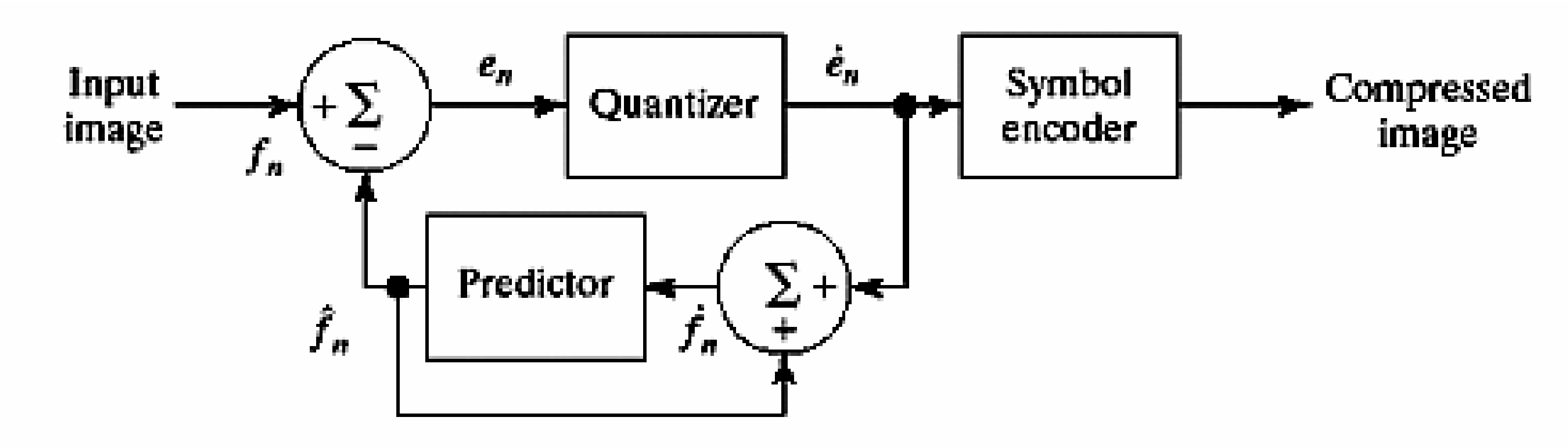
波段	Orignal (bits/pixel)	Best JPEG (bits/pixel)	MAW预测树 ER(bits/pixel) T(s)		SNMAW预测树 ER(bits/pixel) T(s)	
1	5.74	3.39	2.24	0.99	1.66	0.09
2	5.19	2.96	1.89	0.78	1.36	0.09
3	6.24	3.71	2.49	0.97	1.91	0.09
4	6.04	3.71	2.53	1.04	1.96	0.09
5	6.48	4.20	2.91	1.22	2.32	0.09
6	6.02	3.58	2.40	0.91	1.83	0.09
平均	**6.04**	**3.59**	**2.41**	**0.99**	**1.84**	**0.09**

7.4 Predictive coding

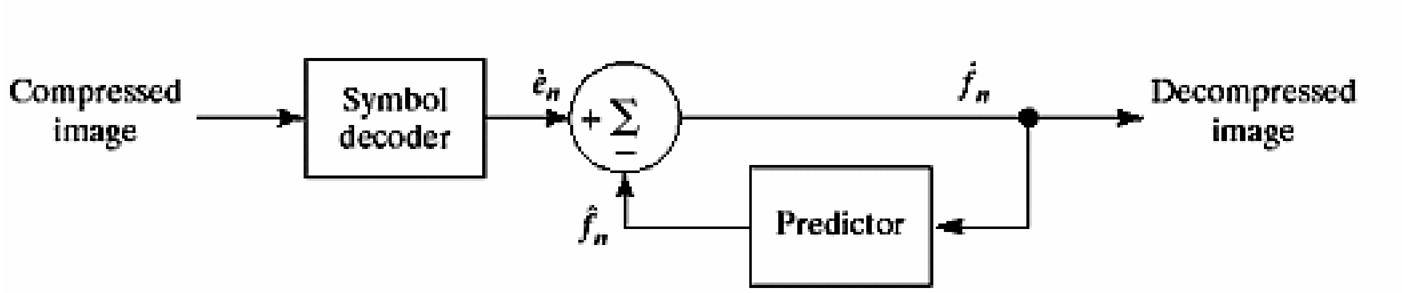
7.4.2 lossy coding: model

encoder

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$



decoder



7.4 Predictive coding

7.4.2 lossy coding: Optimal predictors

differential pulse code modulation(DPCM)

- Minimizes the encoder's mean-square prediction error

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

Subject to the constraint that

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}$$

$$\Rightarrow \quad E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^{m} \alpha_i f_{n-1}\right]^2\right\}$$

7.4 Predictive coding

7.4.2 lossy coding: Optimal predictors

Experiment: four predictors:

(1) $\hat{f}(x,y) = 0.97 f(x,y-1)$

(2) $\hat{f}(x,y) = 0.5 f(x,y-1) + 0.5 f(x-1,y)$

(3) $\hat{f}(x,y) = 0.75 f(x,y-1) + 0.75 f(x-1,y) - 0.5 f(x-1,y-1)$

(4) $$\hat{f}(x,y) = \begin{cases} 0.97 f(x,y-1) & \text{if } \Delta h \le \Delta v \\ 0.97 f(x-1,y) & \text{otherwise} \end{cases}$$

where

$$\Delta h = \left| f(x-1,y) - f(x-1,y-1) \right|$$

$$\Delta v = \left| f(x,y-1) - f(x-1,y-1) \right|$$

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	

7.4 Predictive coding

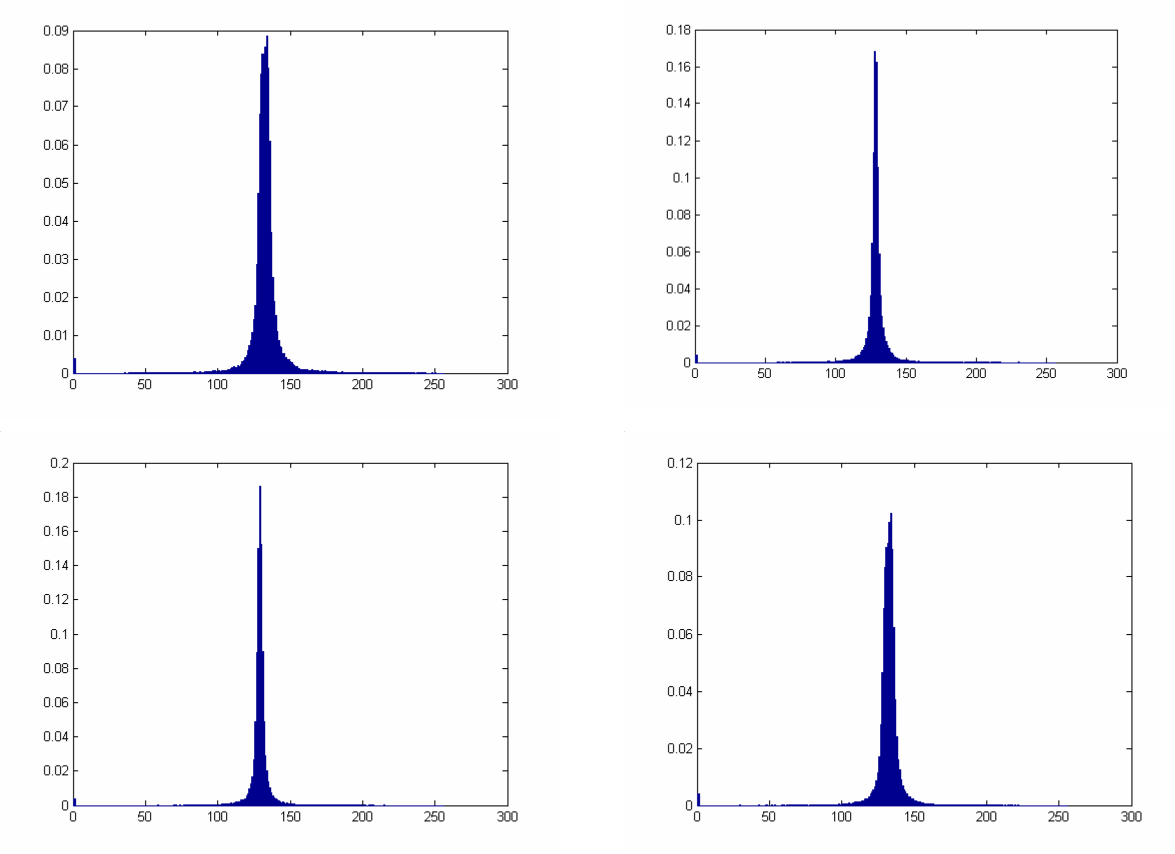
7.4.2 lossy coding: Optimal predictors

Experiment

original

7.4 Predictive coding

7.4.2 lossy coding: Optimal predictors

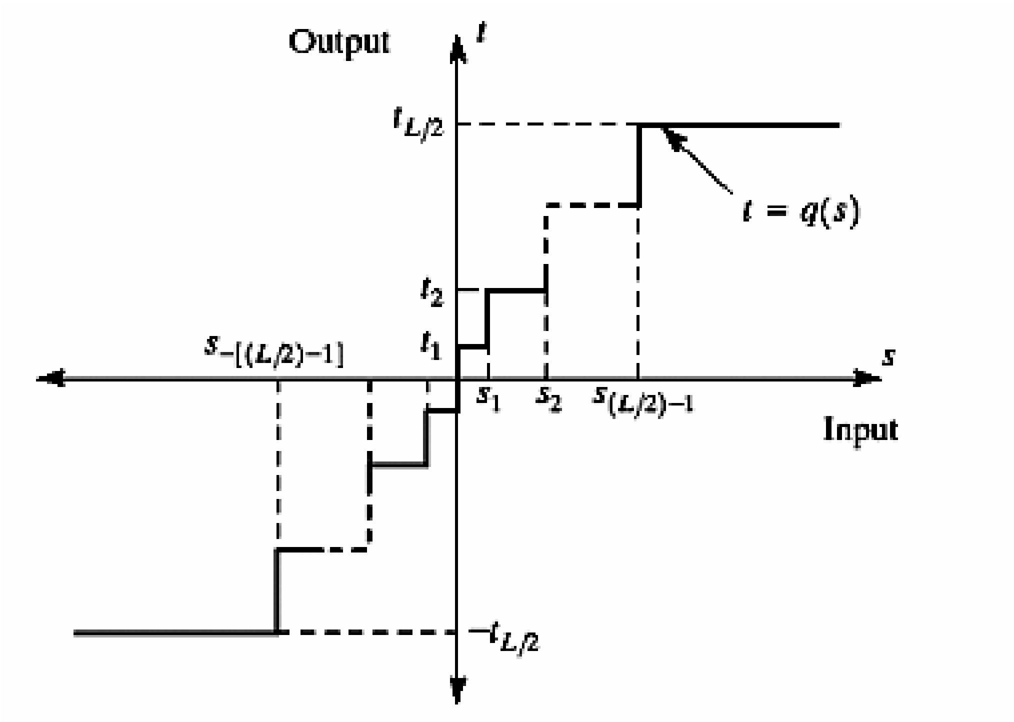


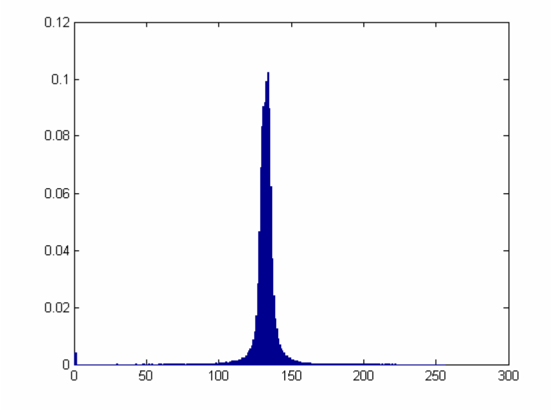
Conclusion:Error decreases as the order of the predictor increases

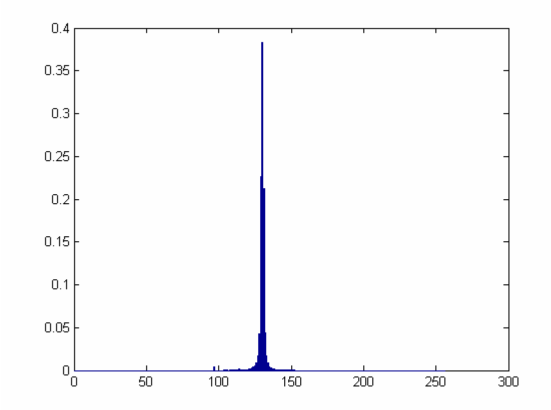
7.4 Predictive coding

7.4.2 lossy coding: Optimal quantization

The staircase quantization function *t=q(s)* is shown as







7.5 Transform Coding

7.5.1 Review of image transform: definitions

$$\mathbf{X} \Rightarrow \boxed{\boldsymbol{T}} \Rightarrow \mathbf{Y}$$

Direct transform $Y = TX$ $\Rightarrow$ $y_{m,n} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} x_{i,j} \Phi(i, j, m, n)$

Inverse transform $X = T^{-1}Y$

Vector expression

polynomial expression

Question: why transformation can compress data?

7.5 Transform Coding

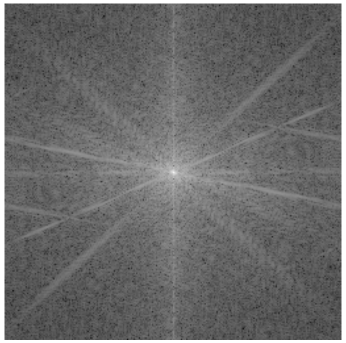
7.5.1 review of image transform: properties

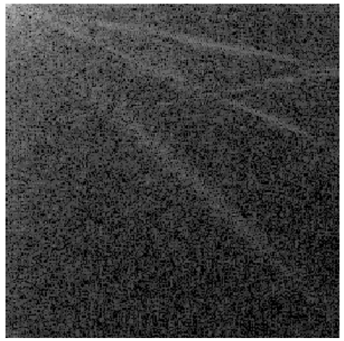
- **Entropy keeping:** $H(X)=H(Y)$
- **Energy keeping:** $$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|X(m,n)|^2=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}|Y(k,l)|^2$$
- **decorrelation:** $$H_\infty=\cdots=H_m<H_{m-1}<\cdots<H_2<H_1<H_0$$

$$H(Y)=H_0(Y) \qquad H(X)=H_m(X) \qquad H_0(X)>H_0(Y)$$

- **Energy re-assigned:**

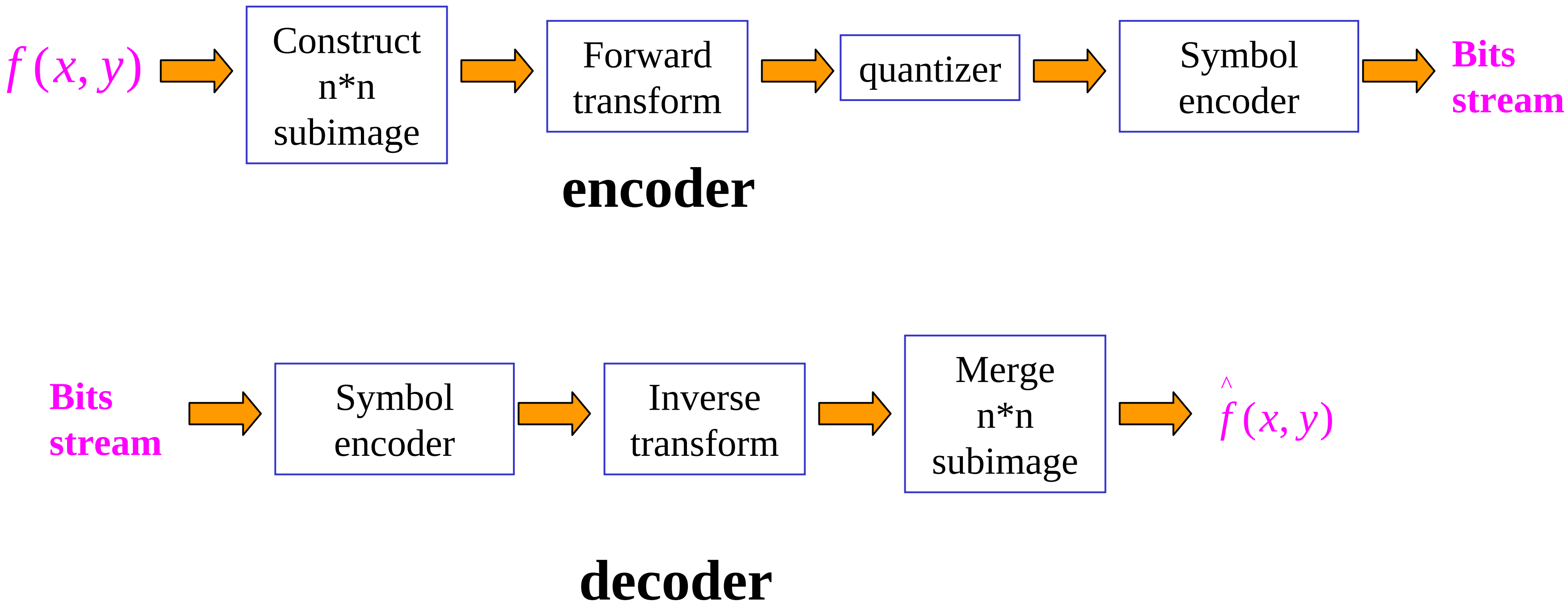

original


DFT


DCT

7.5 Transform Coding

7.5.2 Transform coding system



7.5 Transform Coding

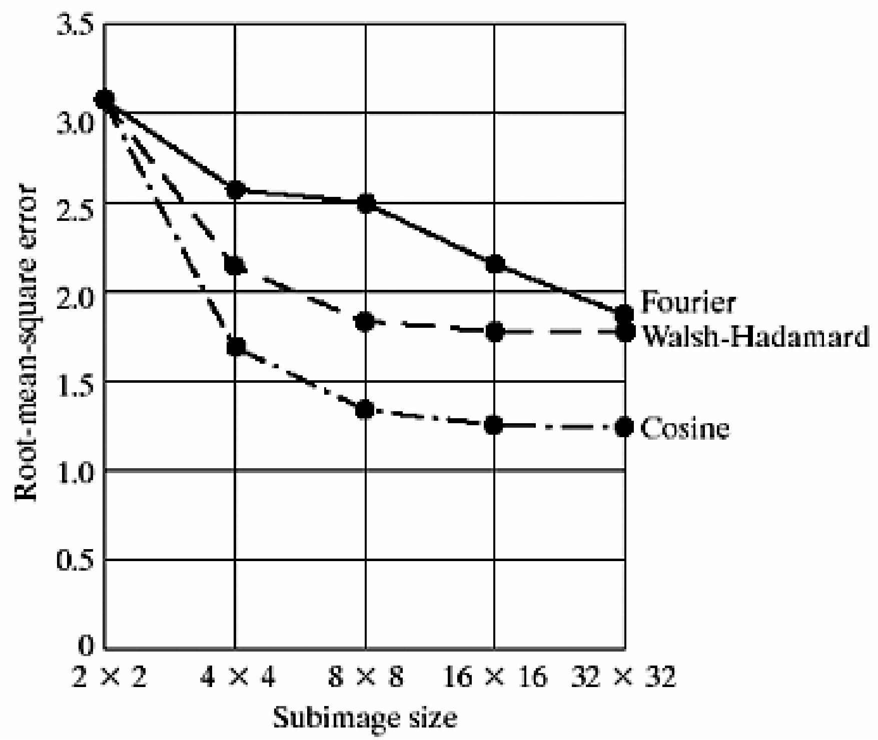
7.5.3 Transform selection

Information packing ability: *KLT>DCT>DFT>WHT*

Computational complexity: *WHT<DCT<DFT<KLT*

7.5.4 sub-image size selection

- Computational complexity increase as the subimage size increase
- Correlation decrease as the subimage size increase
- The most popular subimage size are 8*8, and 16*16



7.5 Transform Coding

7.5.5 bit allocation: zonal coding

- transform coefficients of maximum variance carry the most information and should be retained

1	*1*	*1*	*1*	*1*	*0*	*0*	*0*
1	*1*	*1*	*1*	*0*	*0*	*0*	*0*
1	*1*	*1*	*0*	*0*	*0*	*0*	*0*
1	*1*	*0*	*0*	*0*	*0*	*0*	*0*
1	*0*	*0*	*0*	*0*	*0*	*0*	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*

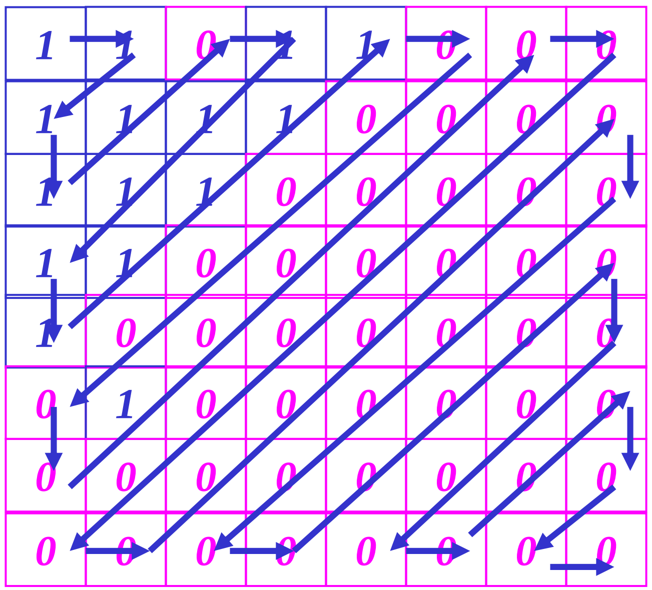
Zonal mask

8	*7*	*6*	*4*	*3*	*2*	*1*	*0*
7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
6	*5*	*4*	*3*	*3*	*1*	*1*	*0*
4	*4*	*3*	*3*	*2*	*1*	*0*	*0*
3	*3*	*3*	*2*	*1*	*1*	*0*	*0*
2	*2*	*1*	*1*	*1*	*0*	*0*	*0*
1	*1*	*1*	*0*	*0*	*0*	*0*	*0*
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*

Bit allocation

7.5 Transform Coding

7.5.5 bit allocation: threshold coding



threshold mask

Zig-zag

7.5 Transform Coding

7.5.5 bit allocation: threshold coding

There are three basic ways to threshold a transformed coefficients

- A single *global threshold* can be applied to all subimages
- A *different threshold* can be used for each subimage
- The threshold can be varied as a *function* of the location of each coefficient within the subimage

$$\hat{T}(u,v) = round\left[\frac{T(u,v)}{Z(u,v)}\right]$$

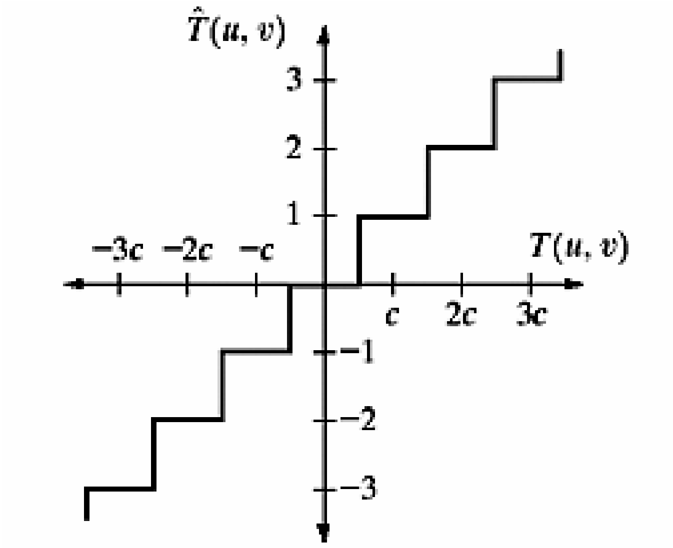
where *T(u,v):* transform coefficients

Z(u,v): quantization matrix

7.5 Transform Coding

7.5.5 bit allocation: threshold coding

Z(u,v) is assigned a particular value *c*



Z(u,v) used in JPEG stardard

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

7.5 Transform Coding

7.5.5 bit allocation: threshold coding

Experiment results

Left column: quantize with $Z(u,v)$

Right column: quantize with $4*Z(u,v)$

7.5 Transform Coding

7.5.3 JEPG lossy compression standard

$f(x, y)$ → Construct 8*8 subimage → DCT → quantizer → Zig-zag → Huffman coding → **Bits stream**

encoder

Question: why there are mosaics in JEPG images?

7.6 Introduction to international standards

- Image compression standards:

  -JPEG,

  -JPEG2000

- Video compression standards:

  -MPEG-1,MPEG-2,MPEG-4,

  -H.261,H.263,H.263+,H.264

  -H.264/AVC

Video compression standards

ITU-T Video Coding Experts Group (VCEG)

– H.261 (1990) $\rightarrow$ H.263 (1995) $\rightarrow$ H.263+ (1998) $\rightarrow$ H.26L

ISO Motion Picture Experts Group (MPEG)

– MPEG1(1991) $\rightarrow$ MPEG2 (1994) $\rightarrow$ MPEG4 (1999)

Joint Video Team (JVT) (VCEG/MPEG) 2001

– H.264/MPEG4 part 10. Official title: Advanced Video Coding (AVC)

– International standard: December 2002

H.264

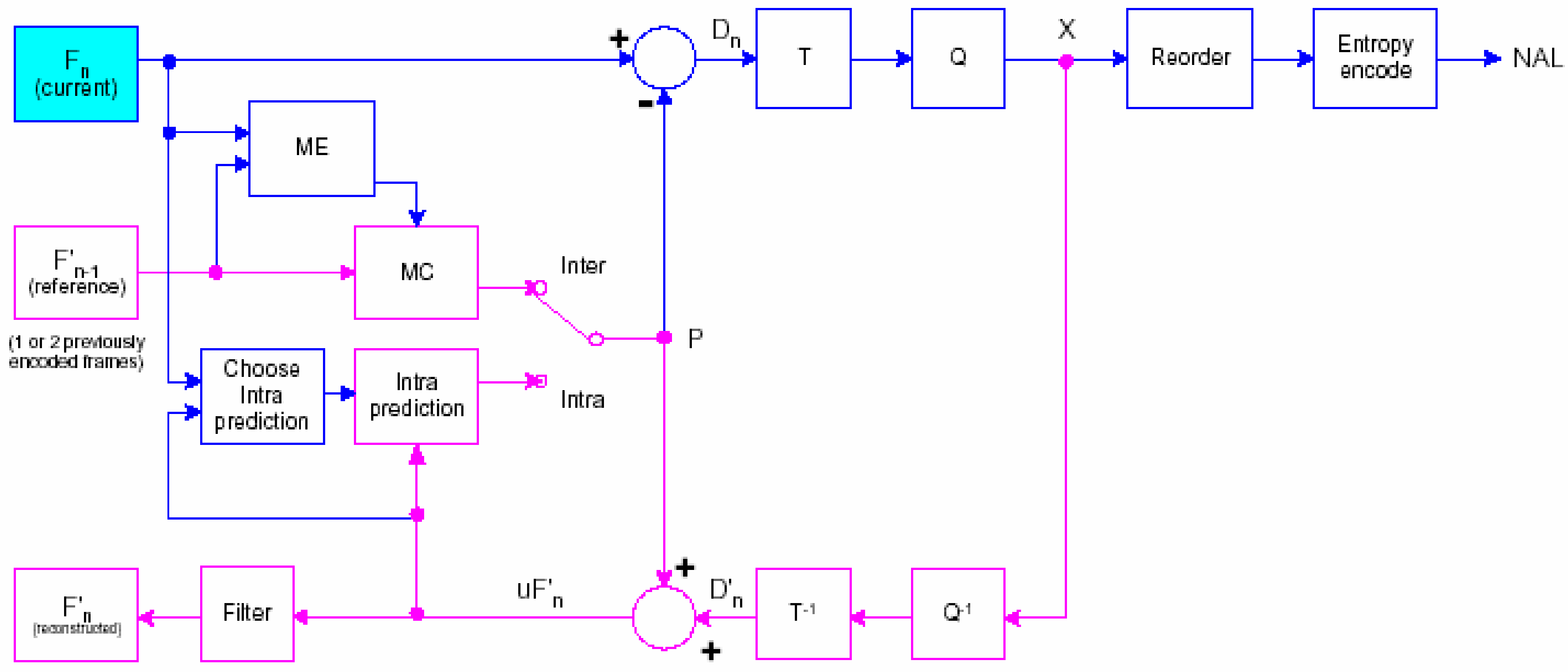
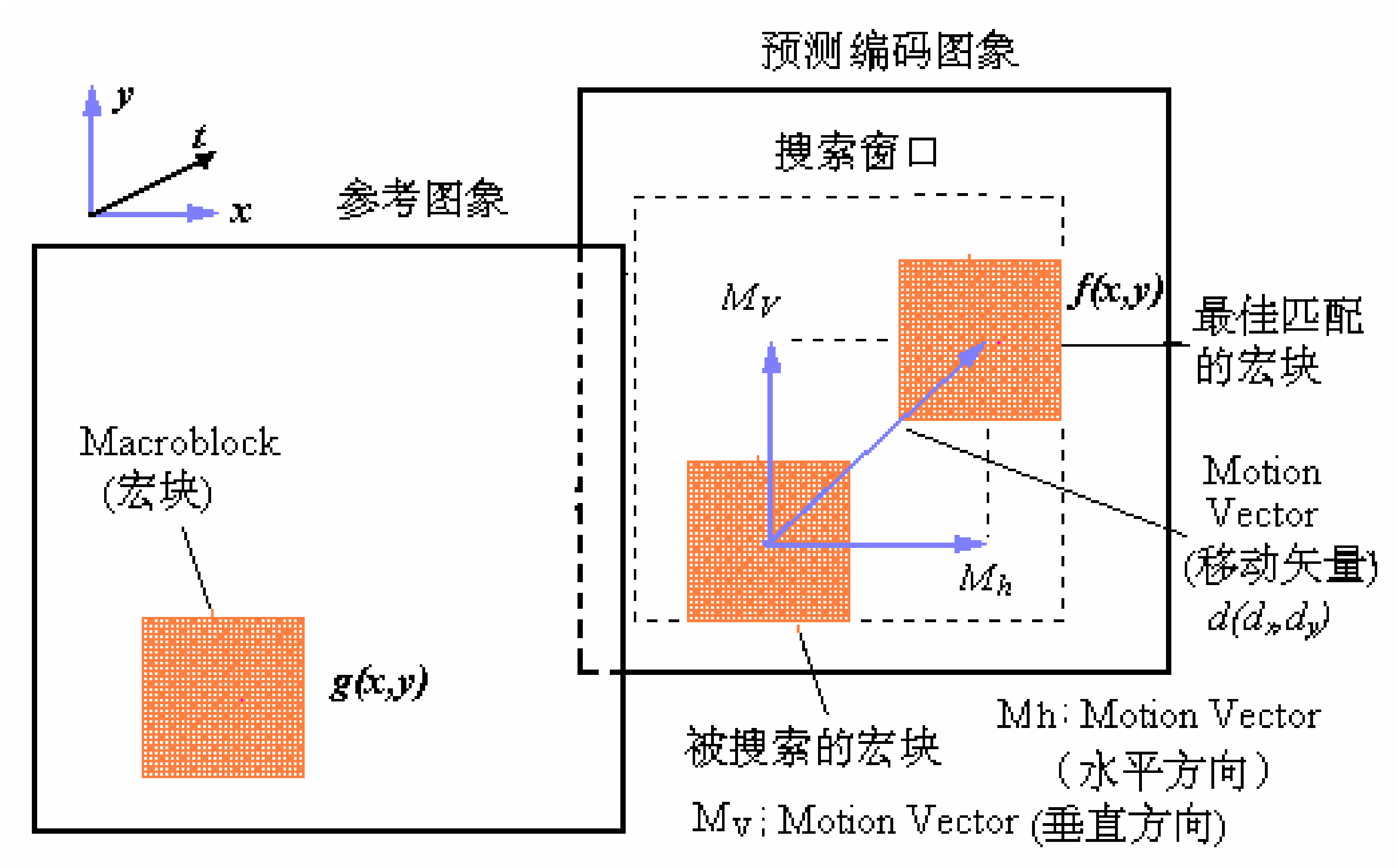


Figure 2-1 AVC Encoder

ME

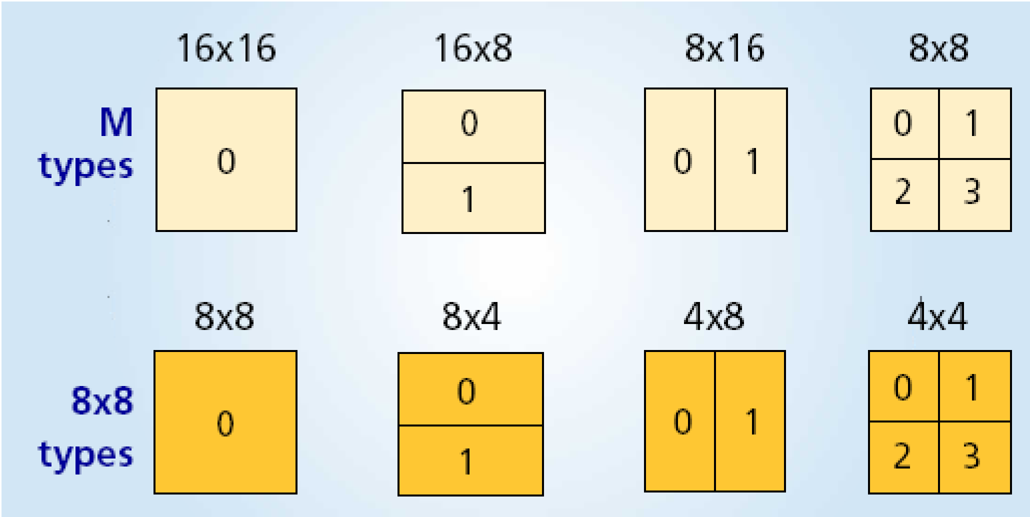


预测编码图象
y
t
x
参考图象
搜索窗口
M_V
f(x,y)
最佳匹配
的宏块
Macroblock
(宏块)
Motion
Vector
(移动矢量)
$d(d_x,d_y)$
M_h
g(x,y)
被搜索的宏块
Mh: Motion Vector
（水平方向）
Mv: Motion Vector (垂直方向)

Prediction of inter Macroblocks –tree structured MC

16*16 block → 16*16, 16*8, 8*16, 8*8

8*8 block → 8*8



These partition and sub-partition give rise to a large mumber of possible combinations within a macroblock

Example:

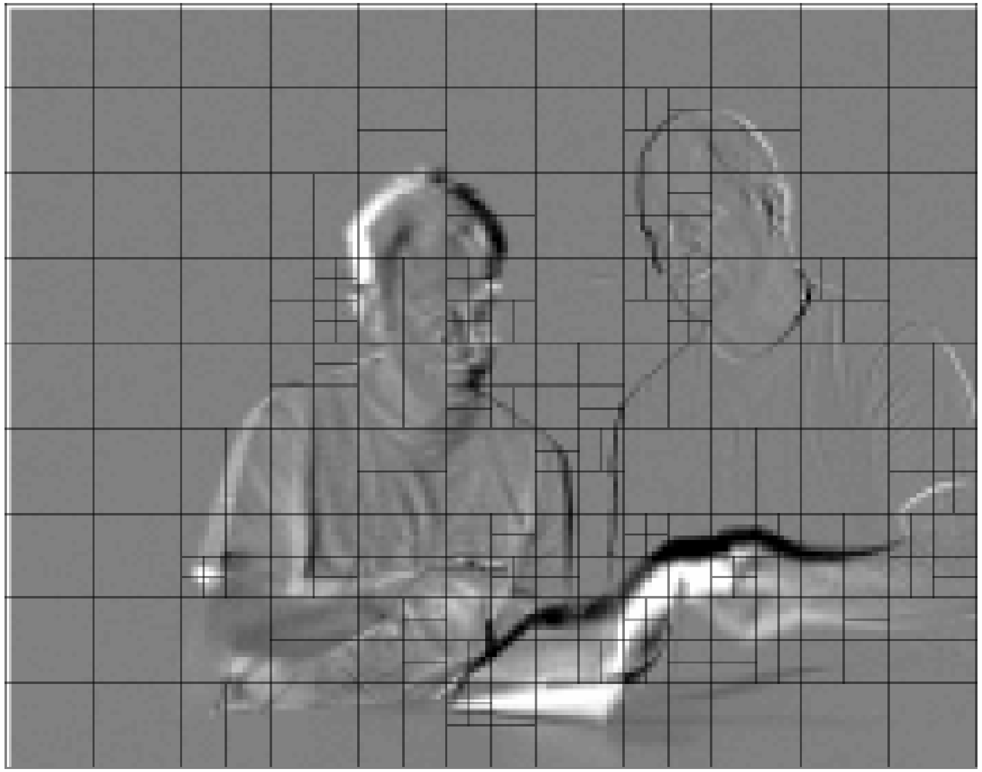
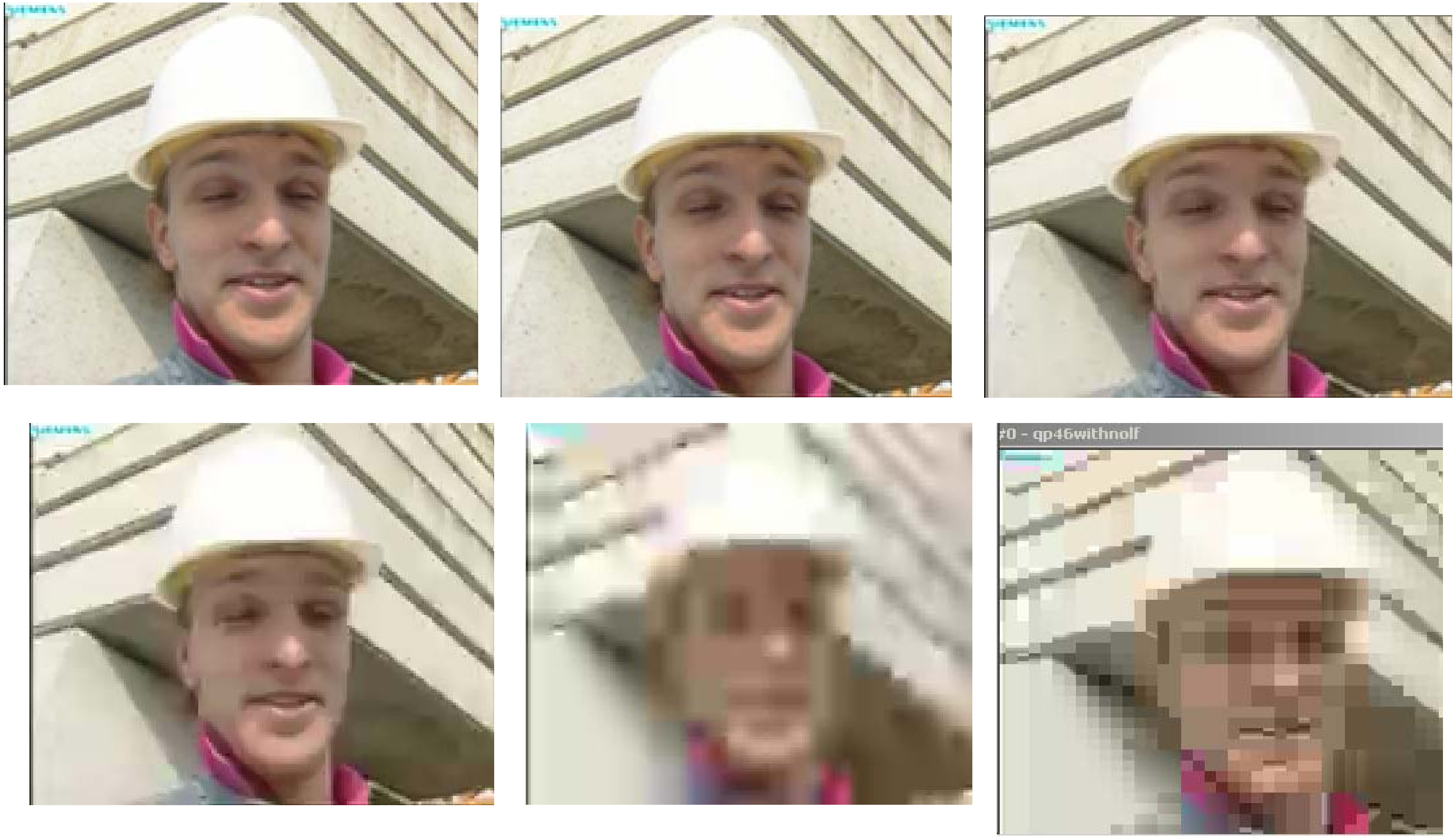


Figure 2-3 Residual (without MC) showing optimum choice of partitions

quantization



QP＝5，15，25，35，50

deblocking

fig6 the comparing of deblocking and non-deblocking（QP=36）

科大西区

West campus of USTC