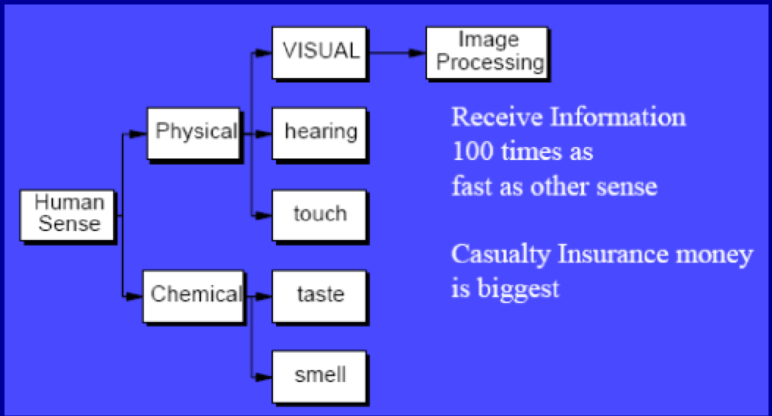
Image Processing in Remote Sensing

Present by:
Dr.Weerakaset Suanpaga
D.Eng(RS&GIS)

Contents

1. Introduction

- Image Processing
    - Treat visuals as engineering



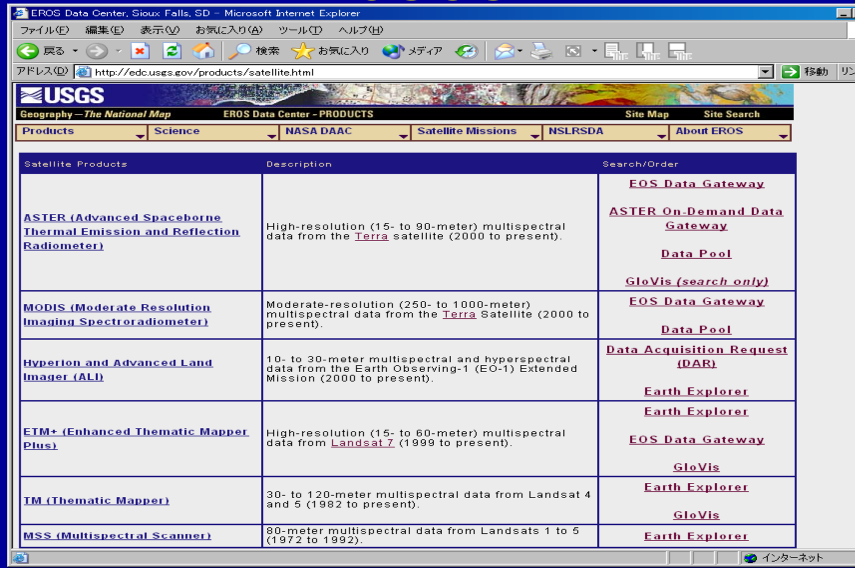
Application Field of Digital Image Processing

1. Document/Drawing Auto Recognition
2. Medical Field
    - X-ray film Interpretation, cell count/classify,CT
3. Industrial Field
    - eye of industrial robot
    - Inspection of products
4. Remote Sensing
    - Land cover/Land use Classification
5. Automated Mapping
    - Building/Road/Signboard Recognition
6. Transportation
    - Vehicle count, Car license plate Recognition,speed measurement

Characteristics of Digital Image in R/S

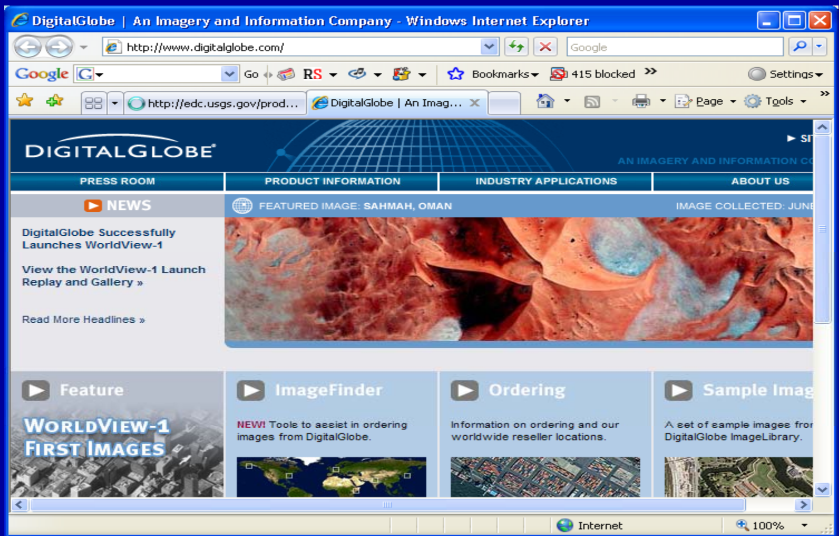
1. Brightness in Numerical Data (Usually in integer)
   - Scaled from radiometer Instrument ($W/m_2/sr$)
2. 3D Datae
   - X-Y Spatially Distributed Geo-Coded Data
   - Z Multi-Channel
3. Big Volume
4. A lot of useful information
5. A lot of Distortions involved
   - Radiometric
   - Geo-location

USGS



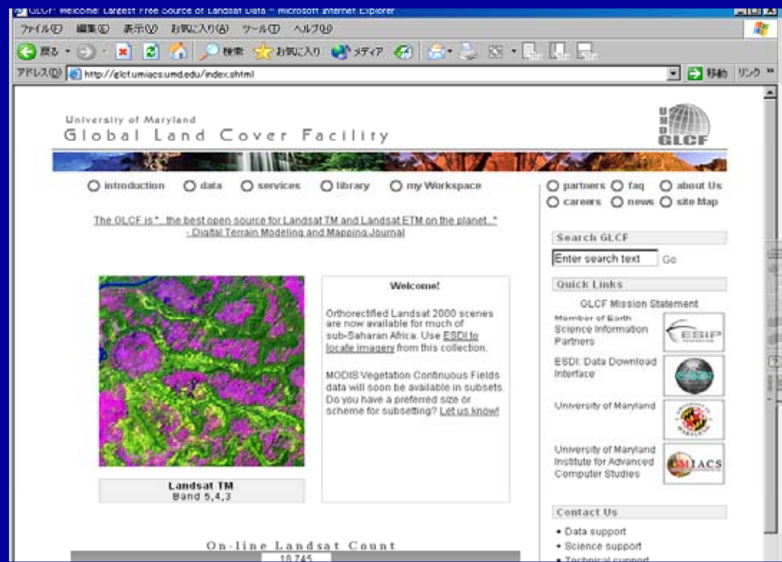
http://edc.usgs.gov/products/satellite.html

Commercial Satellites/ 60cm



http://www.digitalglobe.com/

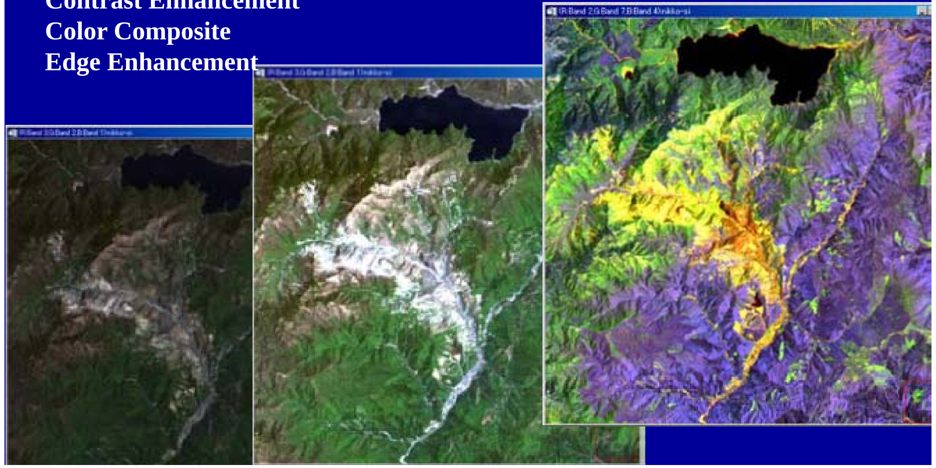
Free download Site



Preprocessing I

Visualization for better visual interpretation / understanding
Contrast Enhancement
Color Composite
Edge Enhancement

Original Data is not suitable for visual interpretation. This is just data which represent radiance of earth surface



Preprocessing II

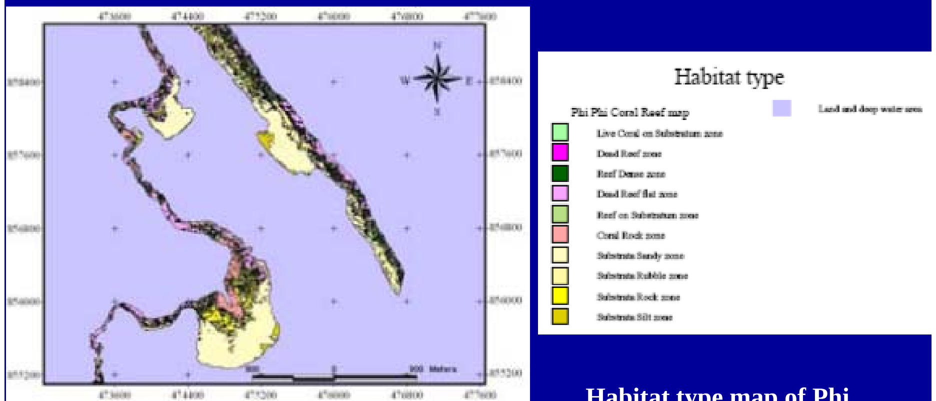
Geometric Correction: to know the exact position and overlay with maps.

Other correction: Radiometric Correction / Atmospheric Correction

Classification

- To divide images into several number of classes. –Landuse/Landcover

Map Publishing



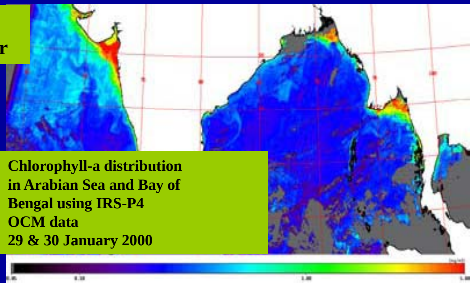
Habitat type map of Phi Phi Don Island.

Further Analysis

Calculating Physical Parameters using Models

Water discharge
Biomass, CO_2
SST (Sea Surface Temperature)
Chlorophyll-a Concentration
Suspended Sediment
and etc.

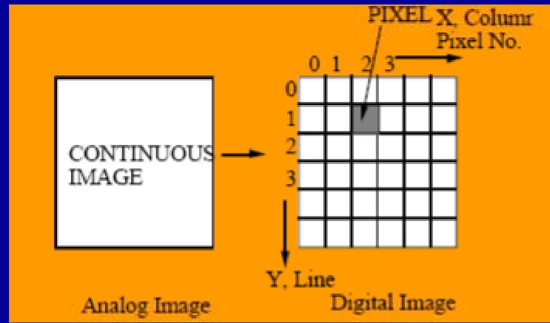
Water Runoff Model
White: Saturated by Water



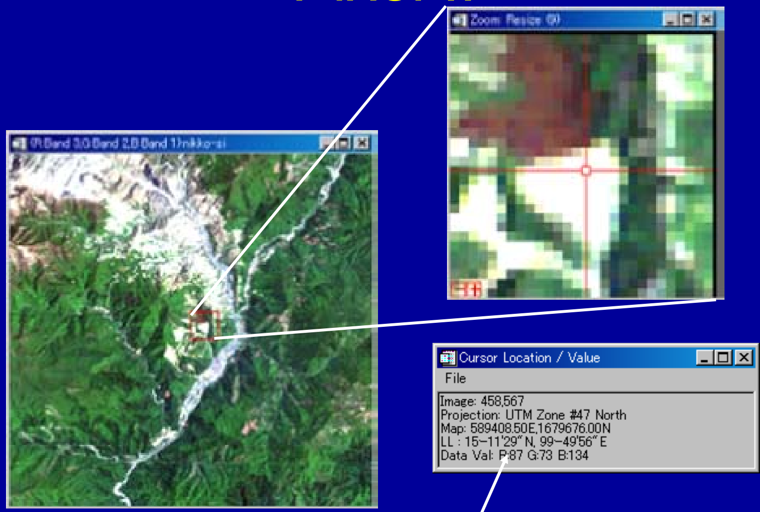
Chlorophyll-a distribution in Arabian Sea and Bay of Bengal using IRS-P4 OCM data 29 & 30 January 2000

Digital Image Data Pixel

Pixel (Picture Element) ,pixel has a value f(x,y)
x,y:integer, f: brightness in most case, integer



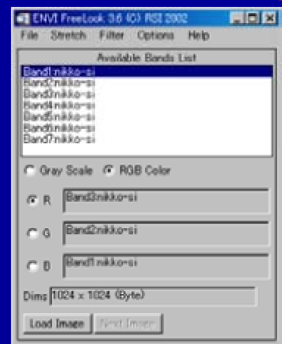
Pixel II

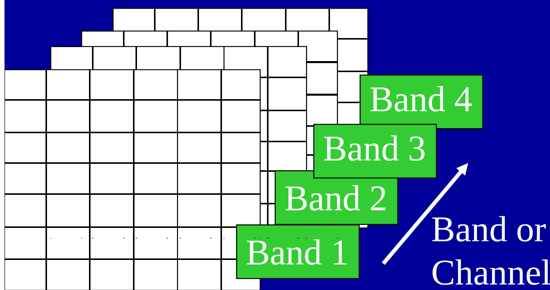


Pixel Value

Multi Channel Image

Color Image: 3 channel for R,G,B
Landsat TM 7 Channel





Bit and Binary System

The gray level of each pixel is recorded and stored as a finite number of bits.

If there are k bits/pixel, total of 2k gray levels over the range 0 to 2^k -1

Exmaple of 3 bits image

bit map			graylevel	bitmap			graylevel
bit2	bit1	bit0		bit2	bit1	bit0	
0	0	0	0	1	0	0	4
0	0	1	1	1	0	1	5
0	1	0	2	1	1	0	6
0	1	1	3	1	1	1	7

if k equals 8, the group of bits is called byte.

Binary System in Computer Memory

Pixel value is stored in limited space in a computer memory. 1unit = 1byte = 8bits

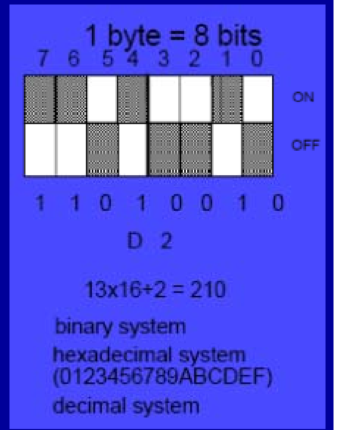
8 bits has 2^8=2*2*2*2*2*2*2*2 = 256 combinations of on/off at bits.

Thus k bits unsigned integer has 0 to 2^k-1 of data range.

- **8bits (1byte) / pixel 0 -> 255**
- **16 bits (2bytes)/pixel 0-> 65535**
- **1024 bytes = 1KB**
- **1024 KB = 1MB**
- **1024 MB = 1GB**

Image Size in Bytes

1024 width *1024 height *7 bands / 1 byte/pixel -> 7MB



How to store numerical value in limited number of Bits

Typical Computer Word Length
8 bits, 16bits, 32bits, 64bits

We usually use 8,16,32,64bits to store pixel values.

8bits	unsigned integer	0-255 most common
16bits	unsigned integer	0-65535 Optical, radar image
32bits	unsigned integer	0-4,294,967,295
8 bits	signed integer	-128 to +127
16 bits	signed integer	-32768 to 32767

IEEE format floating point value

32bits float	significant figures	7
64bits float	complex	
64bits double	significant figures	15
128bits double complex		

Byte Order

- If byte/ pixel is 2 or more,
- Byte order depends on the type of CPU(Central Processing Unit) There are 2 types in byte order
  - 1. Motorola etc. (Little endian CPUs, LSB First)
    - 680x0, PowerPC Macintosh
    - Sparc: SUN WorkStation
  - 2. Intel (Big endian CPUs, MSB First)
    - 80x86, Pentium IBM Compatible Personal Computer

Text Data

Text (Character) is being stored as an integer number following certain Character Code Set,Most of the case, ASCII(American Standard Code for Information Interchange). Or ISO 8859-1is used. Sometime we just say ASCII File, or Text File, which means you can read it. Compare with image data which we call binary data, or binary file.

http://www.asciitable.com/

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.asciitable.com

Sampling & Quantization

Digitization of Gray Scale Images

- (Analog to Digital Image) Conversion of a continuous picture into a discrete form

1. Sampling : Selection of a discrete grid to represent an image (usually square grid)

2. Quantization: Mapping of the brightness into a numerical value (How many levels ? usually 8 bits – 16 bits)

Sampling

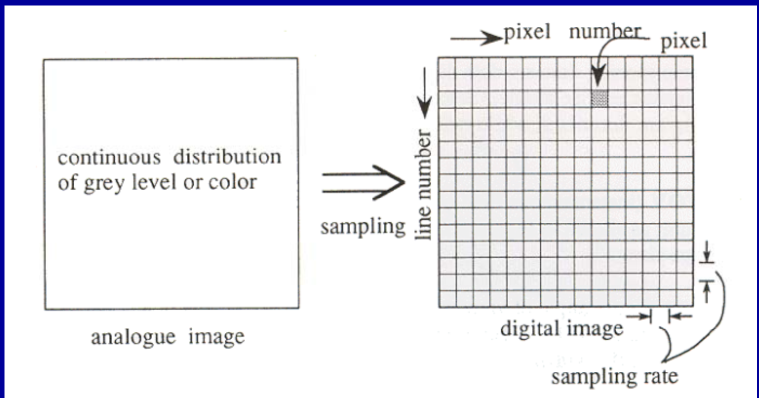


Figure 6.1.1 Concept of sampling

FOV and IFOV

- FOV: Field of View
- IFOV: Instantaneous Field of View for 1 pixel

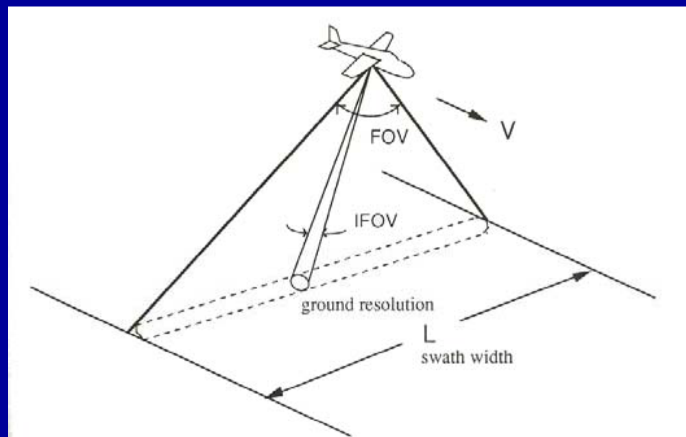


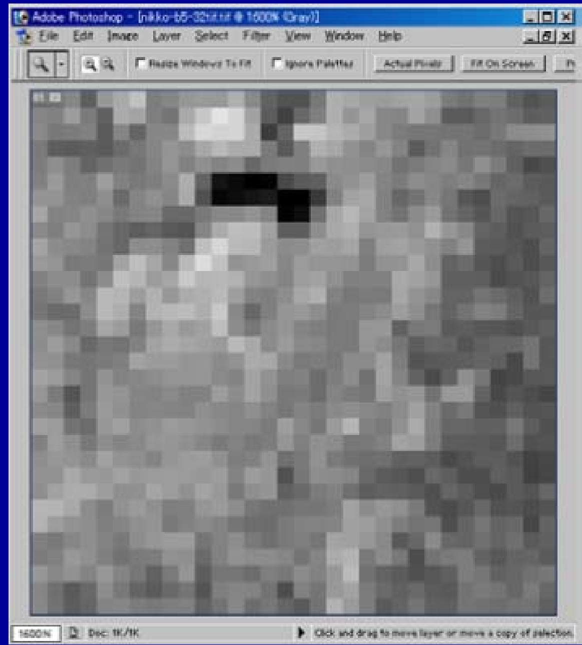
Figure 6.2.1 FOV and IFOV

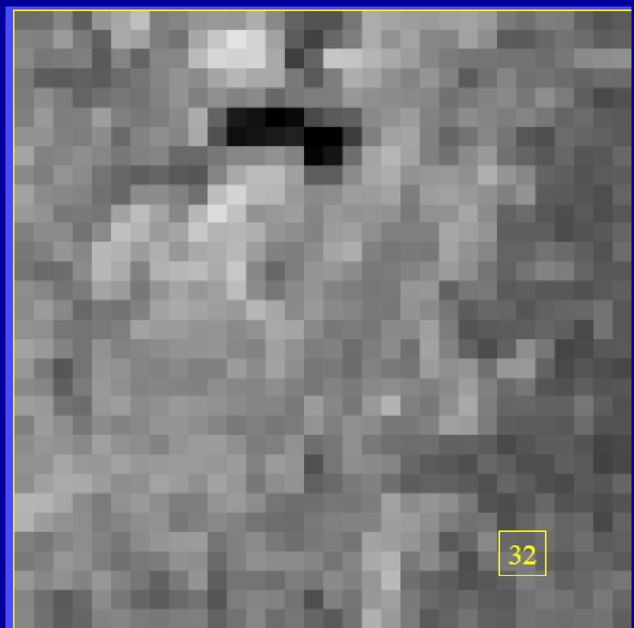
Sampling Policy

Pixel Size or Sampling Rate

Shannon's sampling thoerem

" There will be no loss of information if sampling is taken with a period of half of the reciprocal of the frequency involved in the original analog frequency wave”





Quantization

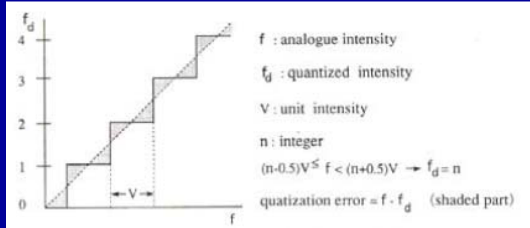


Figure 6.1.2 Concept of quantization

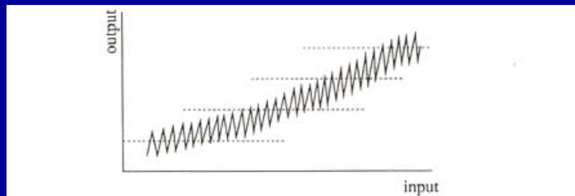
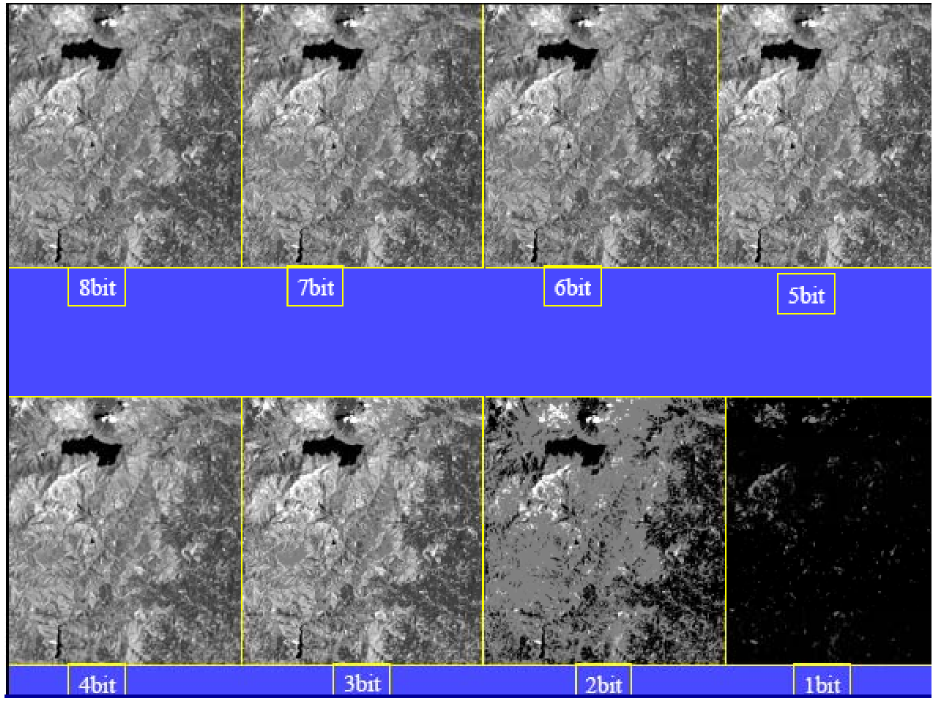


Figure 6.1.3 Quantization in the case of a signal containing a noise



Records and Files

- Image data is stored in secondary memory(Floppy, Hard Disks ,many types of removable disks, Tape, etc.)
- Each line of image pixels is usually stored as a logical record,which is implemented on physical records on media.The total set of records which constructing an image is called file.
- In case of tape media, only sequential access can be done, and logical record is same as physical record
- Gap separates physical record. TM(tape mark) separates files

File Format

(1) Band-Sequential (BSQ)

(2) Band-interleaved by line (BIL)

(3) Band-interleaved by pixel (BIP)

If the processing is a pixel-by-pixel, the BIP format is convenient because the pixel gray levels in each band are stored contiguously within a data record

If the processing is only on a single band from the multispectral image, the BSQ format is most attractive because it minimize the amount of data that must be read to access a single band

The BIL format represents a good compromise of efficiency and convenience for general application and is probably used more widely than either of the other formats.
File

File Format

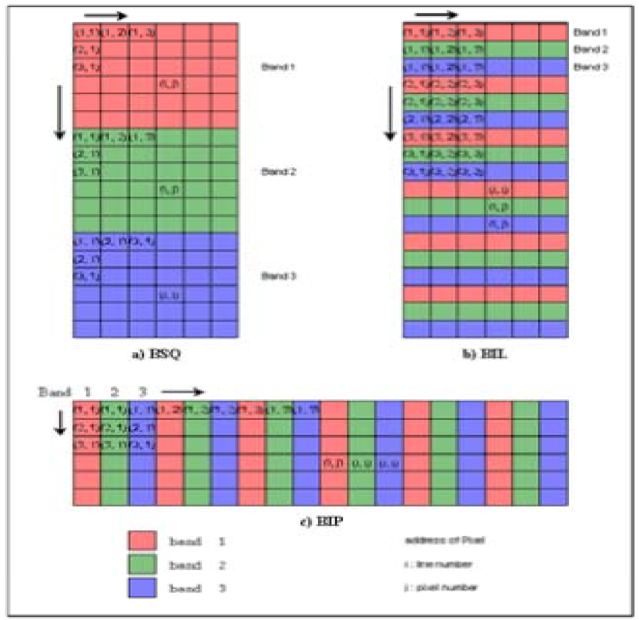


Figure 6.4.1 Image data format (in the case of 3 separate bands)

Digital Image in Computer RAM

- If the quantities of image data are so large, it is impossible to store the whole image data on main memory of a computer(RAM). Image data are usually stored on the secondary memory(Hard disks,etc.).
- When specific data are necessary in the processing procedure, these data are transfered from disk to RAM line-by-line. Usually, the quantity of image data in one line is not so larg

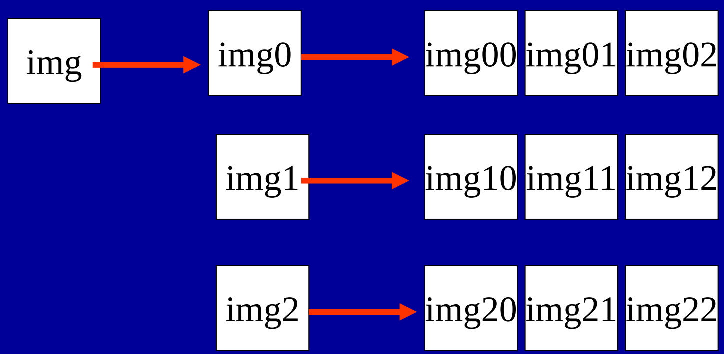
Allocating RAM for image data

➔If the images are not so huge, it is convenient to store whole image in the RAM. Because accessing RAM is fast, and it is easy to write image processing programs.

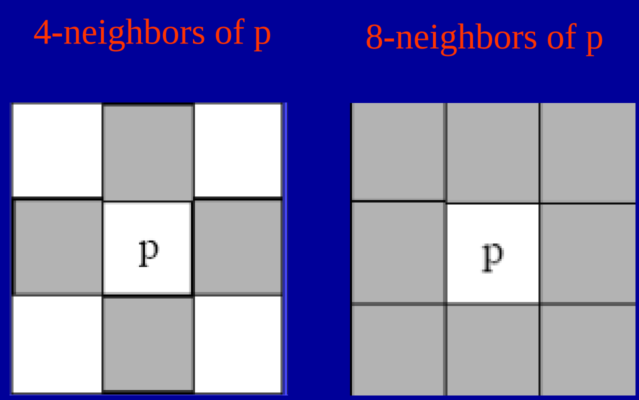
➔In C language , the pointer is used to allocate memory for image data storage in the main memory

```
unsigned char **img;                     /* 2D array (single band image) */
int h,w;                                 /* height and width of image */
int i,j;                                 /* counter declaration */
img=malloc(sizeof( unsigned char * ) * h ); /* allocate pointer table for each line*/
for(i=0;i<h;i++)                         /* allocate memory for each line */
        img[i] = malloc( sizeof( unsigned char ) * w );
for(i=0;i<h;i++)                         /* accessing to image data */
        for(j=0;j<w;j++)
                img[i][j] = 0;
```


Pointer Structure for Allocating Memory

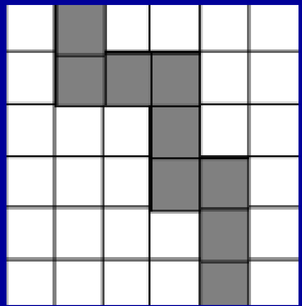


Neighbors of a Pixel

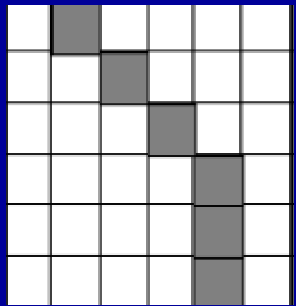


Connectivity

continuous
at 4-connectivity



continuous
at 8-connectivity



Contents

3. Visualization of Remote Sensing Data

1. Visualization is Important
   - Human's extremely high capability on image recognition
   - Better Understanding by visual interpretation
     - Identify objects and their status, patterns, guess what is going on on the ground
   - Utilize Human's capability to distinguish several thousands of colors
2. Image data is not always ready for visualization
   - Pixel has Digital Number (DN), which represents radiance from the ground.
   - Low contrast for visualization
   - Appropriate Enhancement and Color Composite is needed to visualize data on a computer display

3.1 Contrast Enhancement
Image Histogram

- The image histogram describes the statistical distribution of gray levels in an image in terms of the number of pixels(or percentage against the total number of pixels) at each gray level.
- An image histogram only specifies the total number of pixels at each gray level; it contains no information about the spatial distribution of gray levels through out the image.

1	3	2	4
4	5	2	3
3	2	3	2
3	3	4	3

Histogram

Pixel Value	Number
1	1
2	4
3	7
4	3
5	1
Total	16

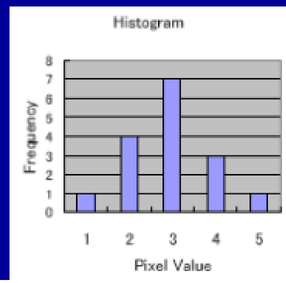
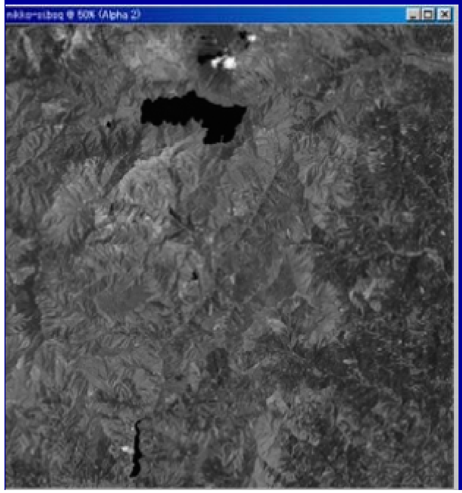
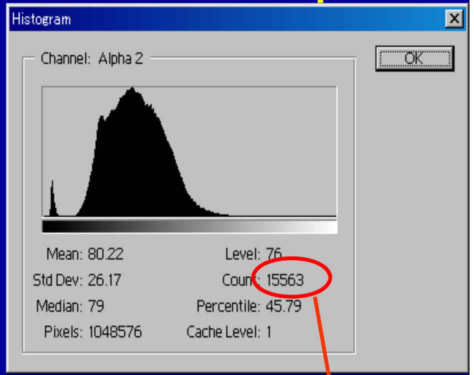


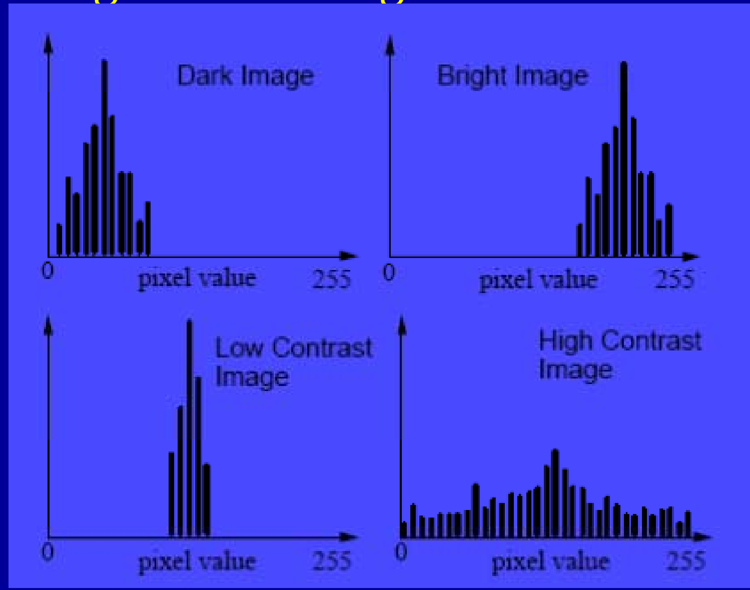
Image Histogram – Example



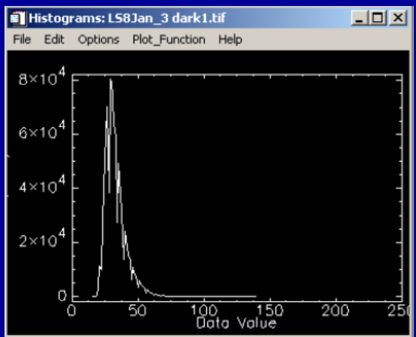


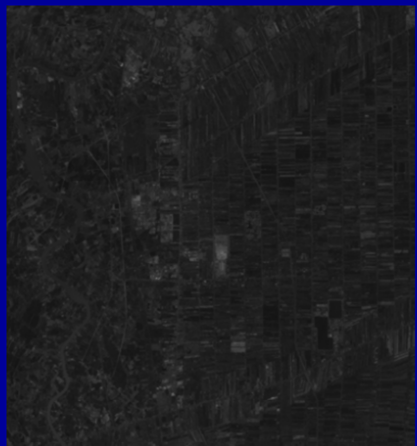
There are 15,563 pixels which have value 76

Histogram and Image Characteristics

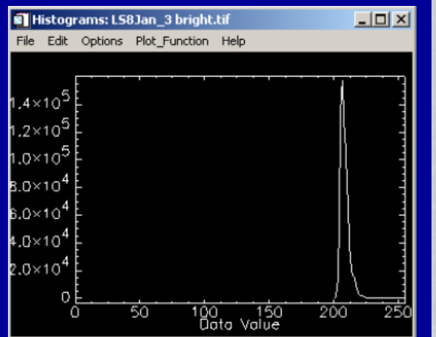


Histogram and Image Characteristics
Dark Image

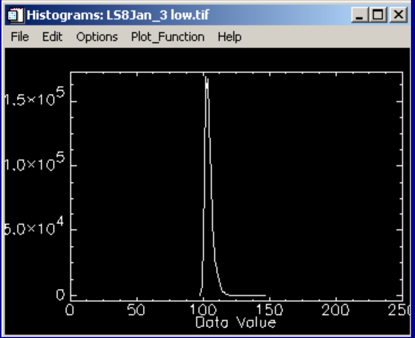




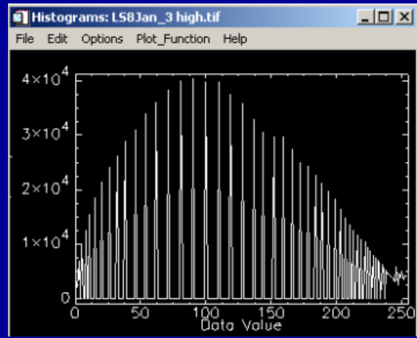
Histogram and Image Characteristics
Bright Image

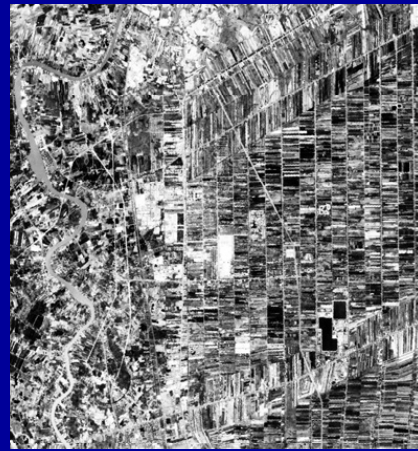


Histogram and Image Characteristics
Low Contrast Image

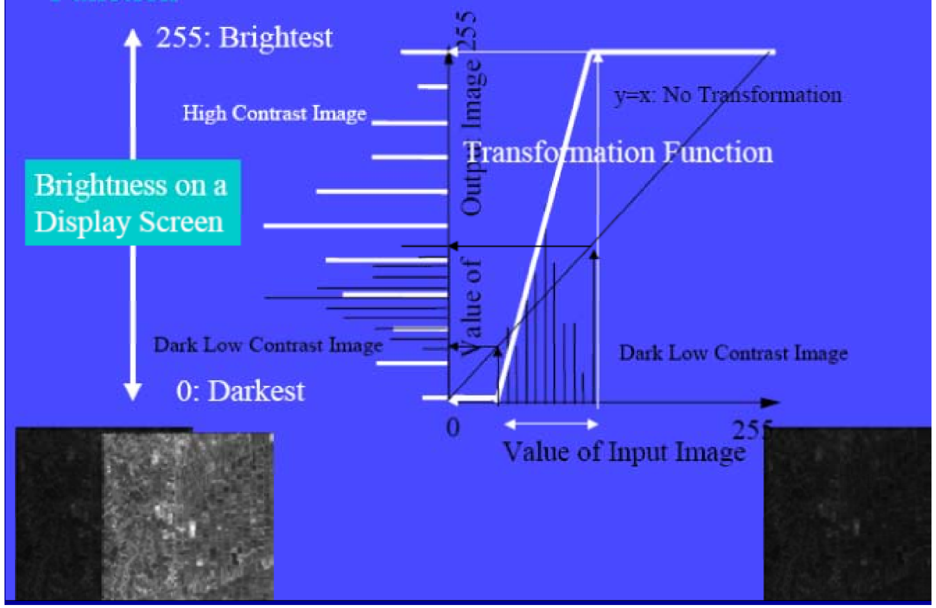


Histogram and Image Characteristics
High Contrast Image

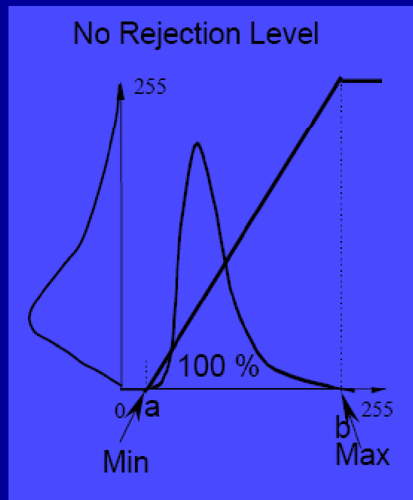


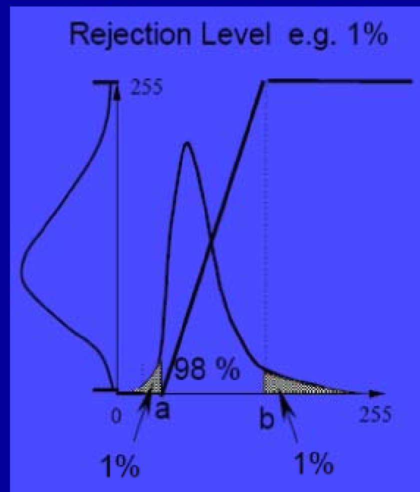


Contrast Enhancement by A Linear Transformation Function



Deciding range of input level



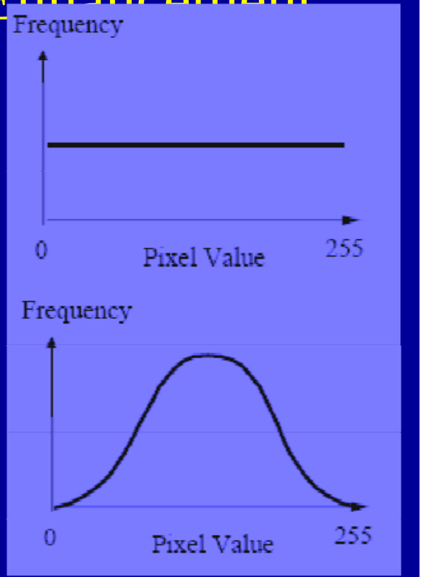


Other Contrast Enhancement

1. Linear Piecewise
   - To give equal number of pixel to brightness
   - To c onvert histogram to Gausian distribution
   - Natural Perception

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

2. Piecewise Linear



Various Transformation

- •Linear transformation
- •Piecewise linear transformation
- •Cyclic transformation
- •Continuous function transformation
- •Histogram Equalization
- •Local contrast enhancement

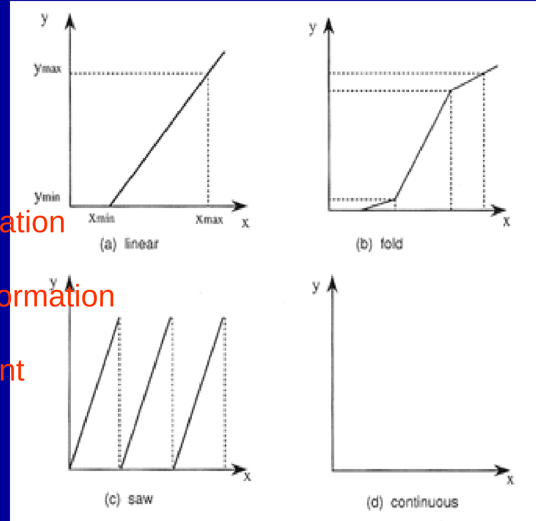
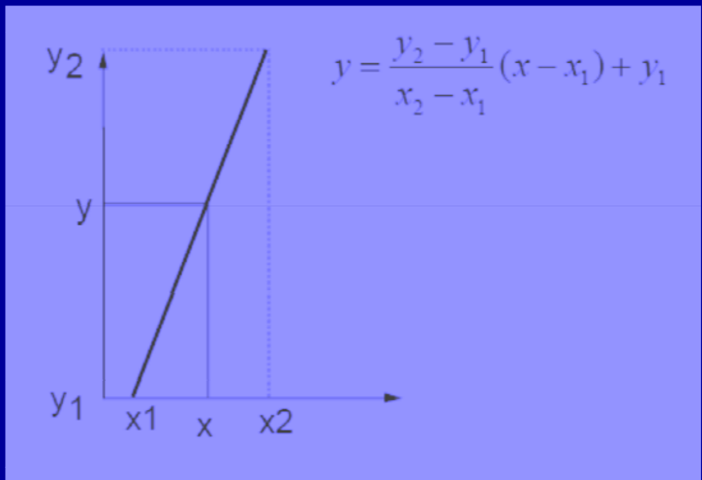


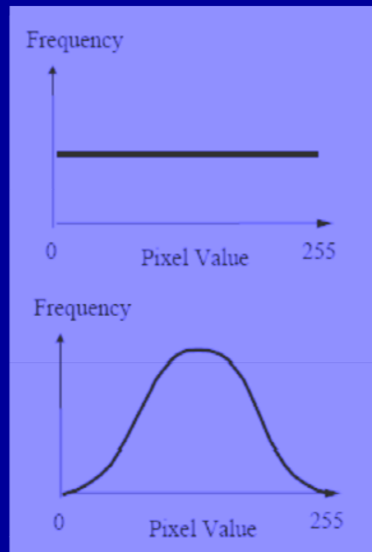
Fig 10.2.1 Typical Density Conversion

Linear Transformation

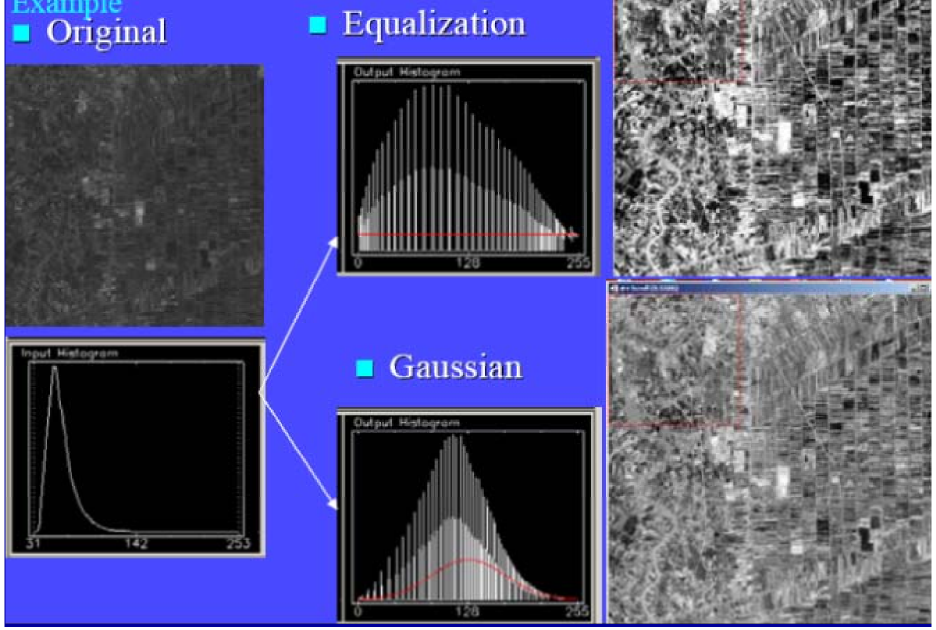


$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$

- Equalization
  - ➢To give equal number of pixel to every brightness ranges
- Gaussian Histogram Conversion
  - ➢To convert histogram to Gaussian distribution
- Natural Perception

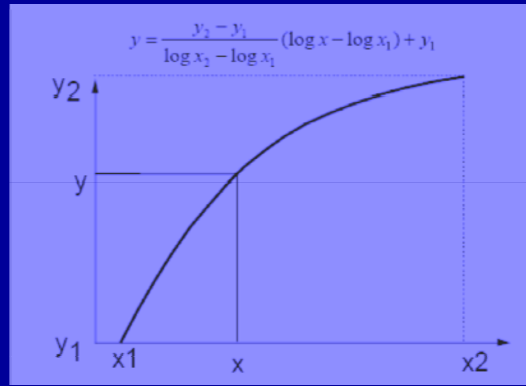


Other Contrast Enhancement Example



Non Linear Continuous Function Transformation

- If dynamic range of the original image is very big.



Fold or Piecewise Linear & Saw

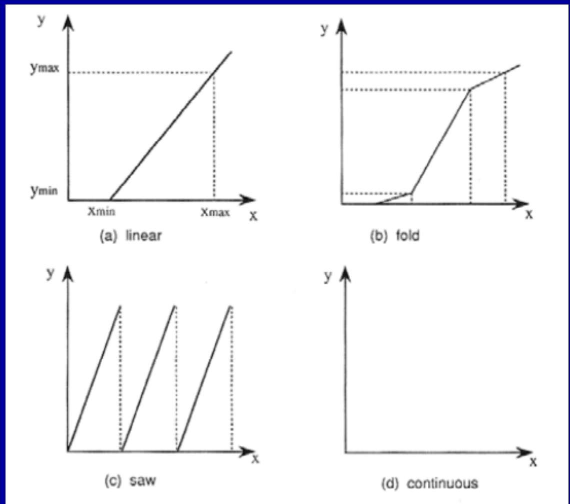


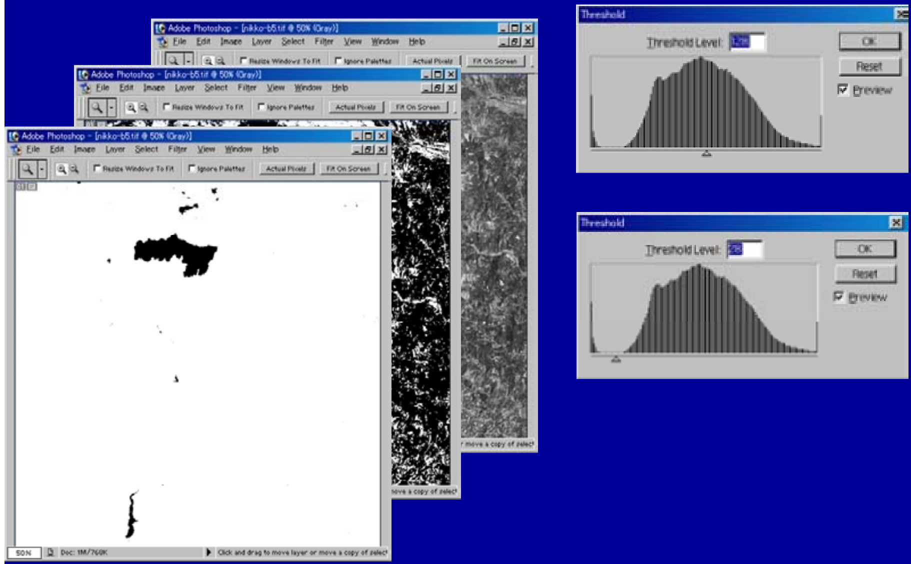
Fig 10.2.1 Typical Density Conversion

Thresholding

- It segments an image into two classes defined by a single gray level threshold

1. Simple classification algorithm
2. Divide the pixel into two using a value T
   - If x is less than T class 1 otherwise class 2
3. Multiple thresholding
   - If x is in the range of T1 to T2, then class 1 otherwise class2
4. Change detection in a pair of multi-temporal images

Threshold is a simple classification



LUT

Look Up Table

No Math Operation -> Fast

Input 2 → Output 12

LUT element No.	LUT
0	4
1	8
2	12
3	16
.	.
.	.
254	255
255	255

```
unsigned char **out_img, **imp_img;
int h, w, i, j;
/* Declaration of LUT array */
static int lut[256];
/* Set up LUT */
for(i=0;i<255;i++){
lut[i] = i*4+4;
if( lut[i] > 255 )
      lut[i] = 255;
}
/* Conversion */
for(i=0;i<h;i++)
for(j=0;j<w;j++)
out_img[i][j] = lut[ inp_img[i][j] ];
```

Hardware or 1-dim Array in Progra

END