

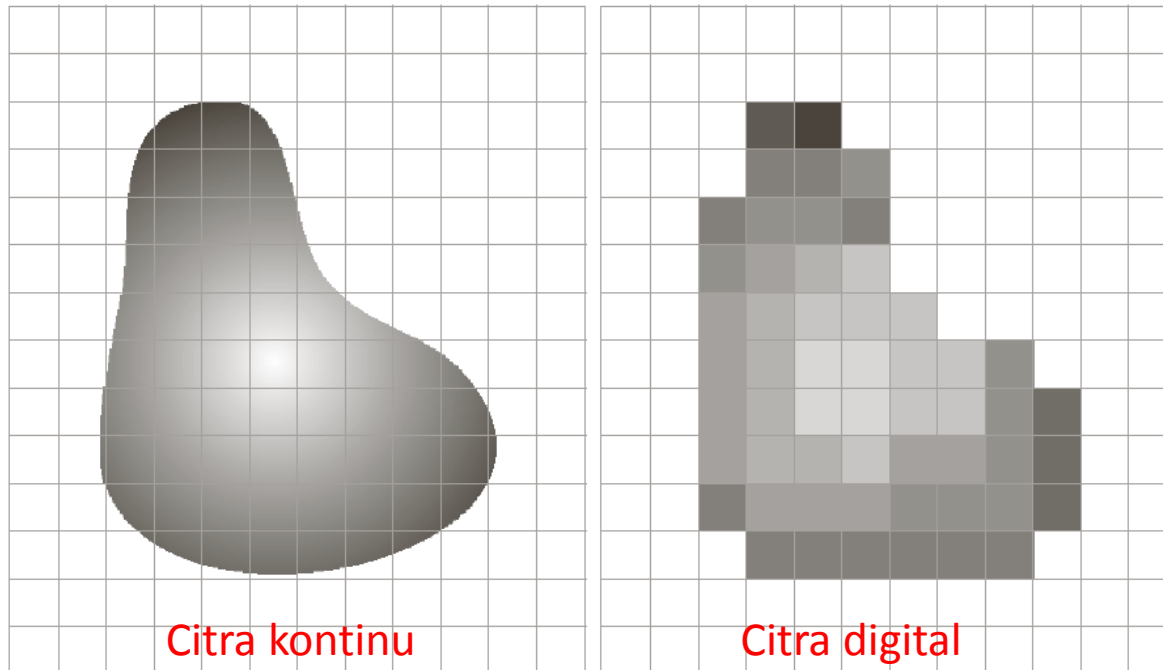
Digitalisasi Citra

IF4073 Interpretasi dan Pengolahan Citra

Oleh: Rinaldi Munir

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2019

- Citra adalah sinyal kontinu dwimatra, $f(x, y)$
- $f(x, y)$ menyatakan intensitas cahaya pada posisi (x, y)
- Agar citra dapat diolah oleh komputer digital, maka citra perlu di-digitalisasi (atau di-digitisasi) menjadi citra digital



Sebuah citra digital adalah versi diskrit dari citra kontinu

a b

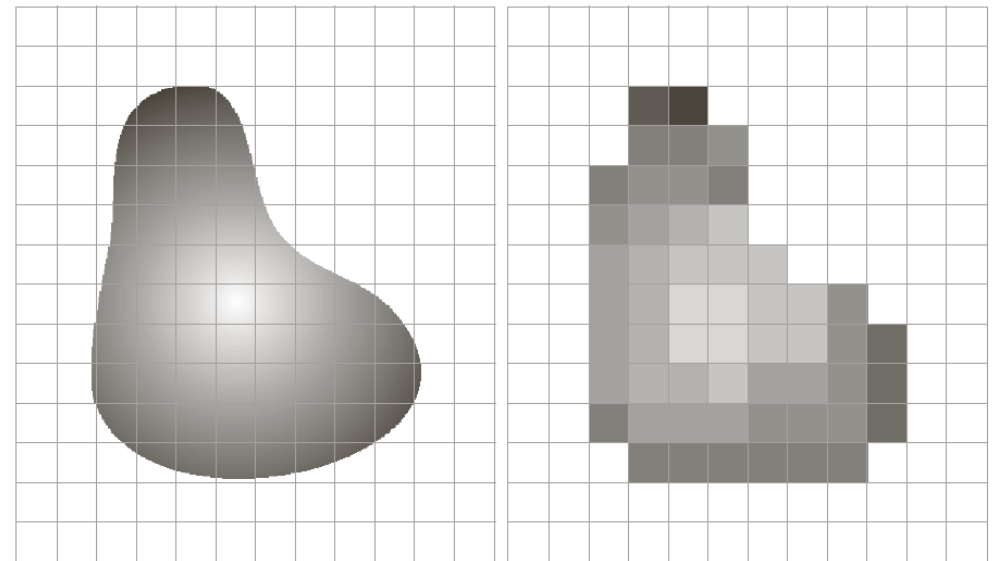
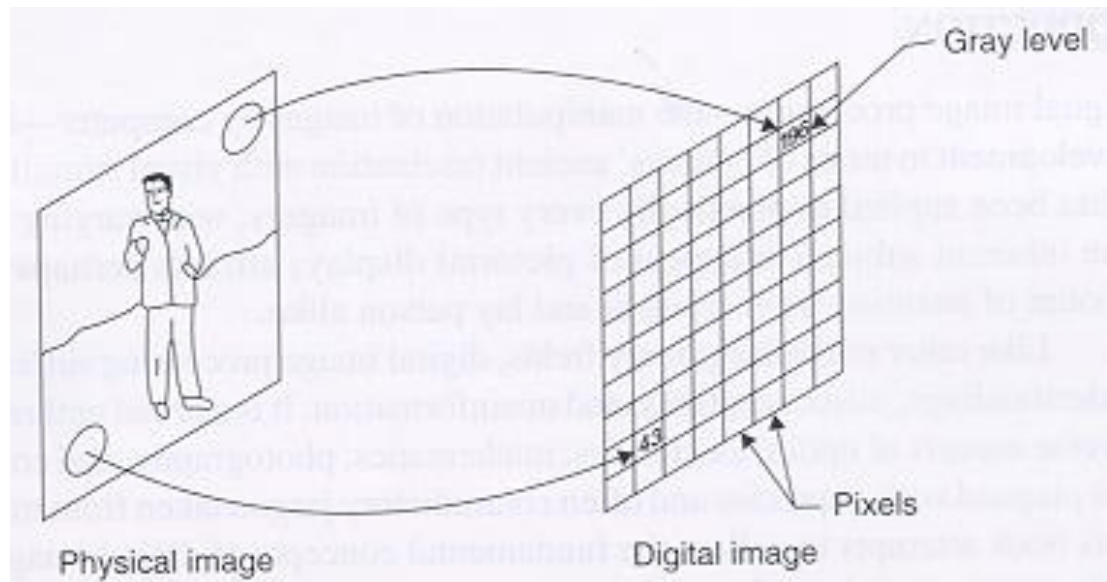
FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

- Citra digital $f(x, y)$ direpresentasikan sebagai matriks berukuran $N \times M$

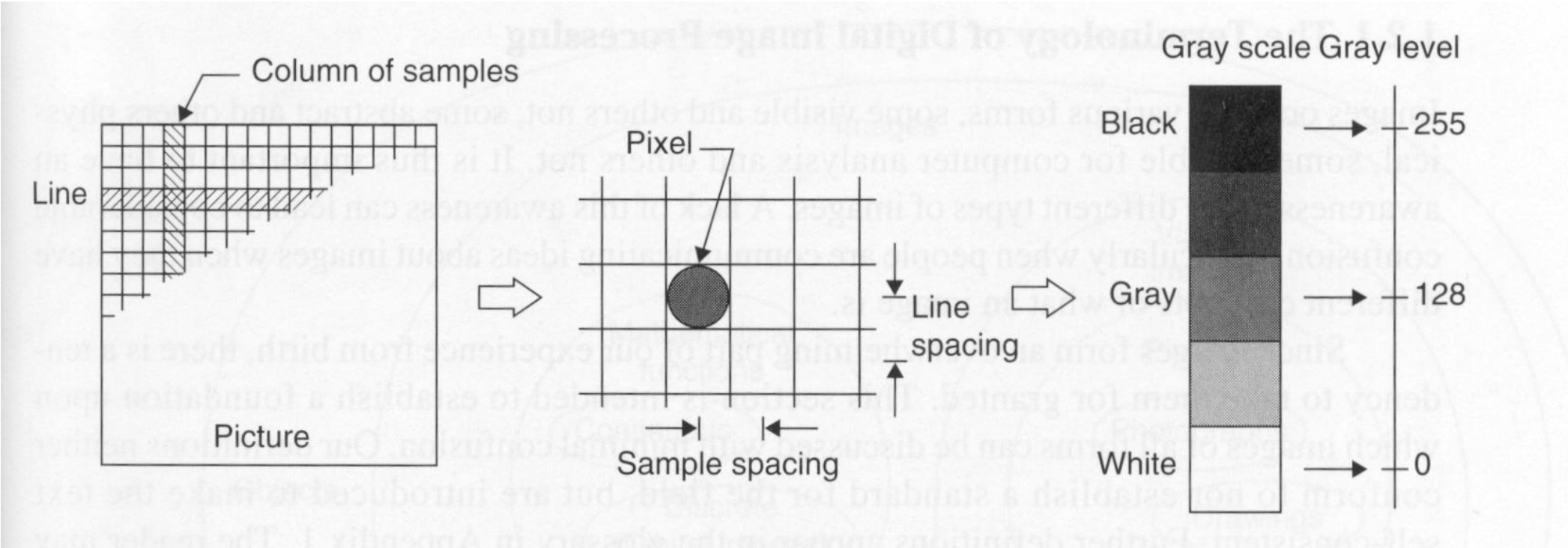
$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

- $N \times M$ menyatakan resolusi citra, N baris dan M kolom
- Setiap elemen matriks menyatakan sebuah *pixel* (*picture element*)
- Nilai $f(i, j)$ menyatakan nilai intensitas *pixel* pada posisi (i, j) , yang dinamakan *graylevel* (derajat keabuan).

- Proses digitalisasi citra ada dua tahap:
 1. Penerokan (*sampling*): yaitu digitalisasi secara spasial (x, y).
 2. Kuantisasi: yaitu pengangkaan nilai intensitas $f(x, y)$ menjadi *integer*.
- Kedua proses di atas berkaitan dengan *diskritisasi* tetapi dalam ranah berbeda

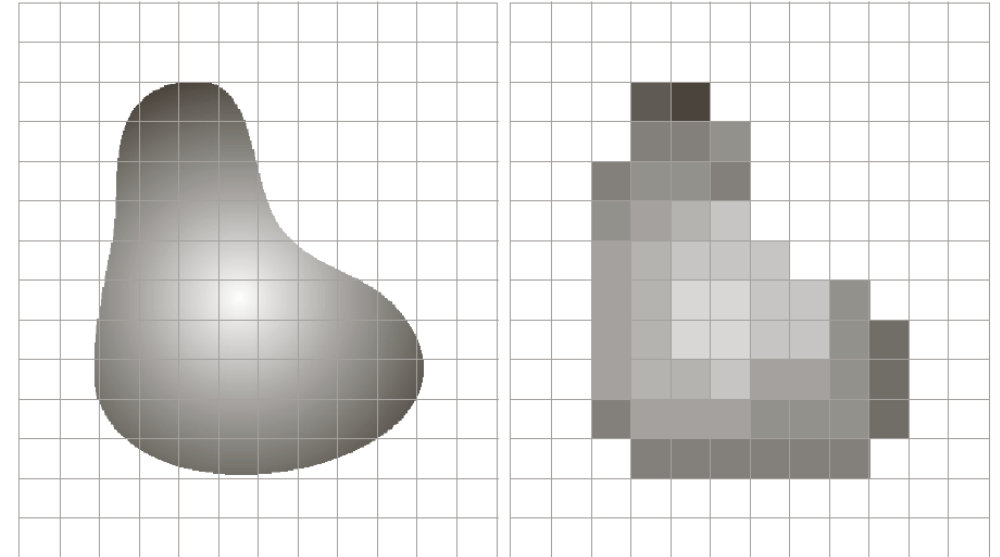
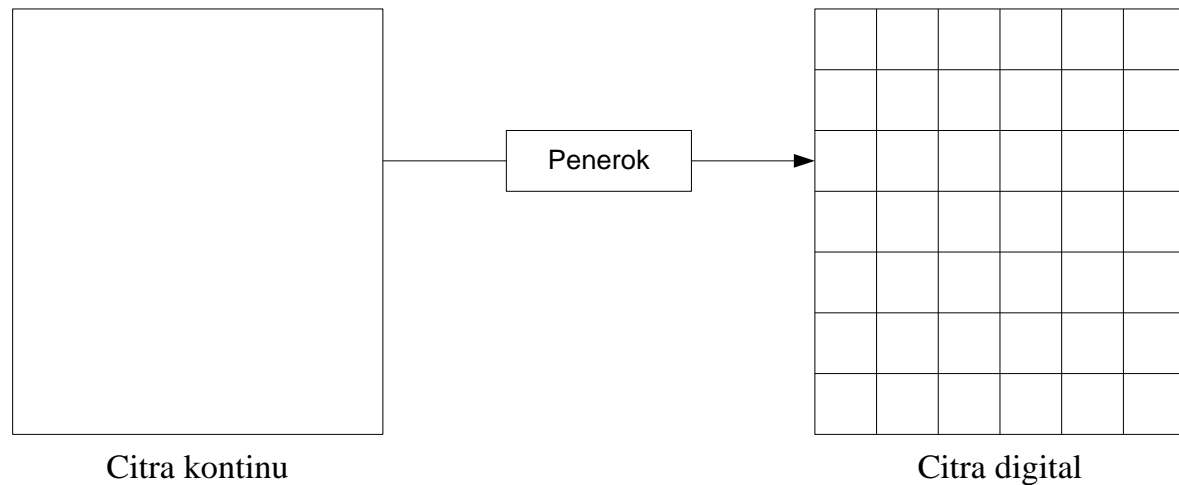


Proses digitalisasi citra

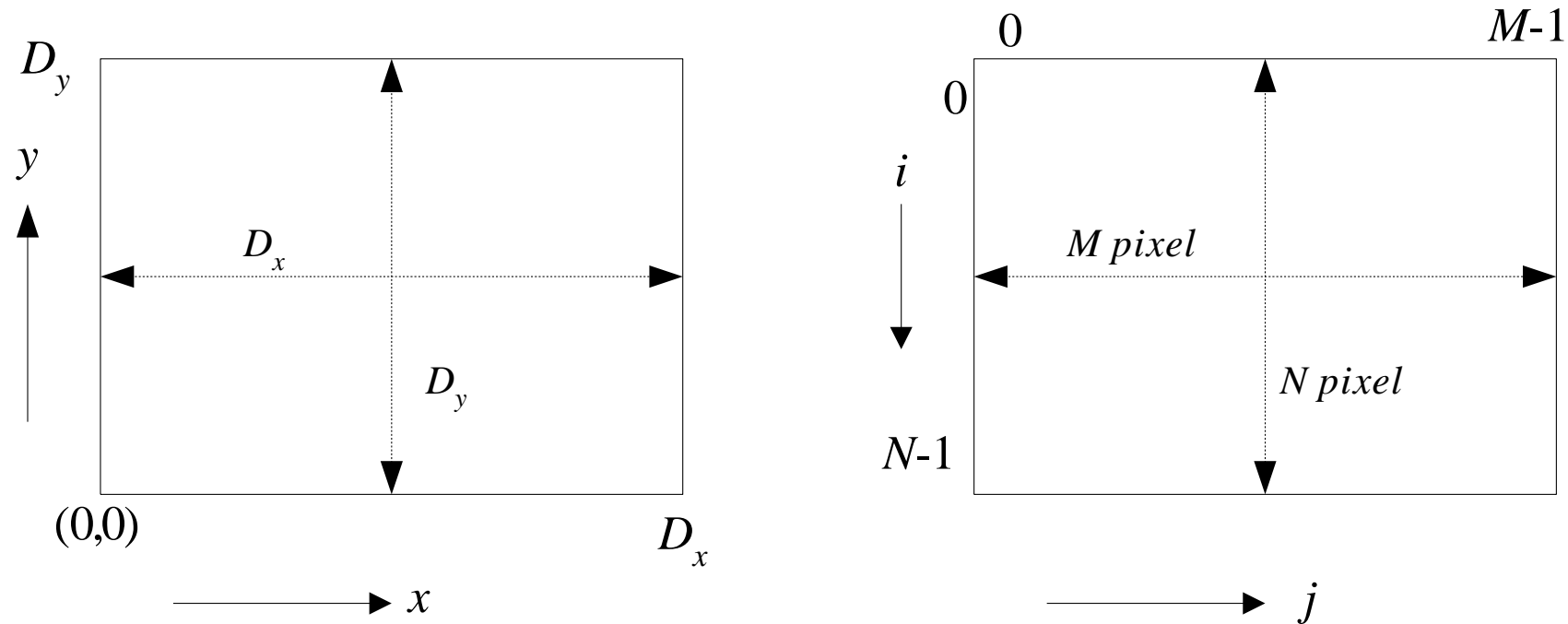


Penerokan (*sampling*)

- Citra kontinu diterok menjadi grid-grid yang berbentuk bujursangkar
- Penerokan bertujuan untuk menentukan seberapa banyak *pixel* yang diperlukan untuk merepresentasikan citra kontinu, dan bagaimana pengaturannya



Hubungan antara elemen gambar dan elemen matriks



$$0 \leq i \leq N - 1$$

$$0 \leq j \leq M - 1$$

N = jumlah maksimum *pixel* dalam satu baris

M = jumlah maksimum *pixel* dalam satu kolom

D_x = lebar gambar (dalam inchi)

D_y = tinggi gambar (dalam inchi)

- Jumlah terokan biasanya diasumsikan perpangkatan dari dua,

$$N = 2^n$$

yang dalam hal ini,

N = jumlah penerokan pada suatu baris/kolom

n = bilangan bulat positif

- Contoh ukuran penerokan: 256×256 *pixel*,

128×256 *pixel*,

512×1024 *pixel*

dst

Contoh penerokan:

original image, $n = 8$ (256 x 256)



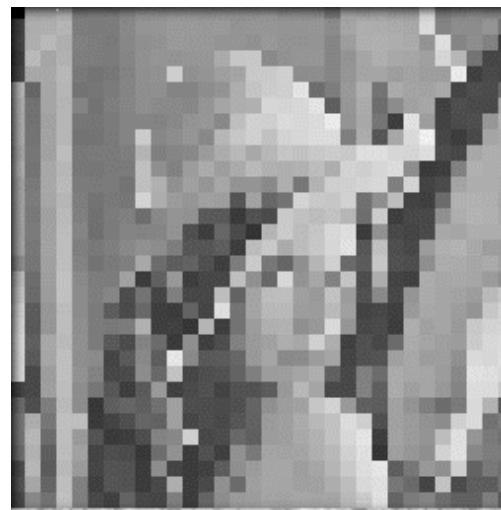
$n = 7$ (128 x 128)



$n = 6$ (64 x 64)



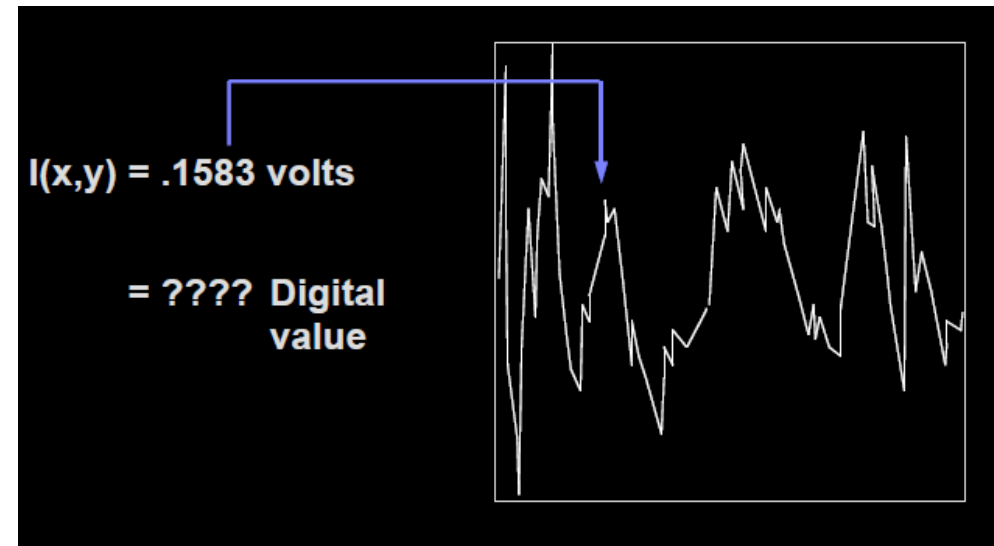
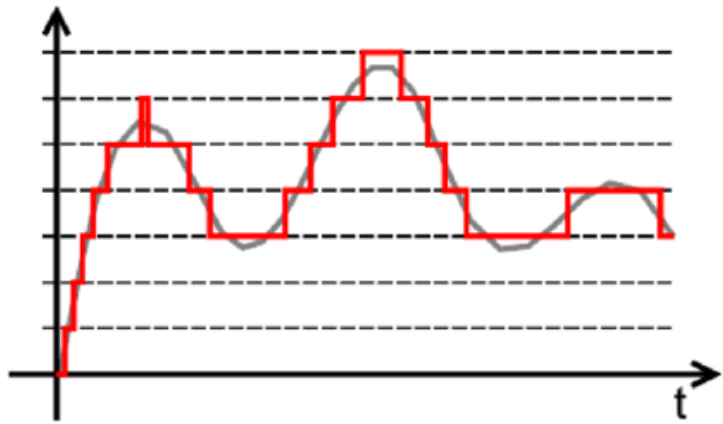
$n = 5$ (32 x 32)



Catatan: Citra telah disamakan ukurannya untuk perbandingan

Kuantisasi

- Kuantisasi berkaitan dengan diskritisasi nilai intensitas cahaya pada koordinat (x, y).
- Tujuan kuantisasi adalah memetakan nilai dari sinyal kontinu menjadi G buah nilai diskrit



- Nilai intensitas *pixel* dalam integer dinyatakan dalam selang $[0, L]$.
- Selang $[0, L]$ disebut skala keabuan (*graylevel*)
- Proses kuantisasi membagi skala keabuan $[0, L]$ menjadi G buah level.
- Biasanya G diambil perpangkatan dari 2,

$$G = 2^m$$

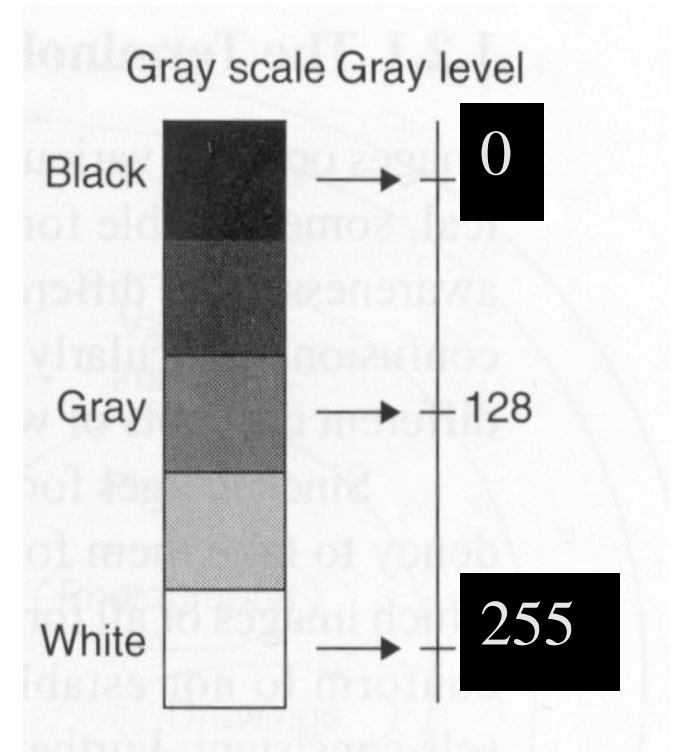
yang dalam hal ini,

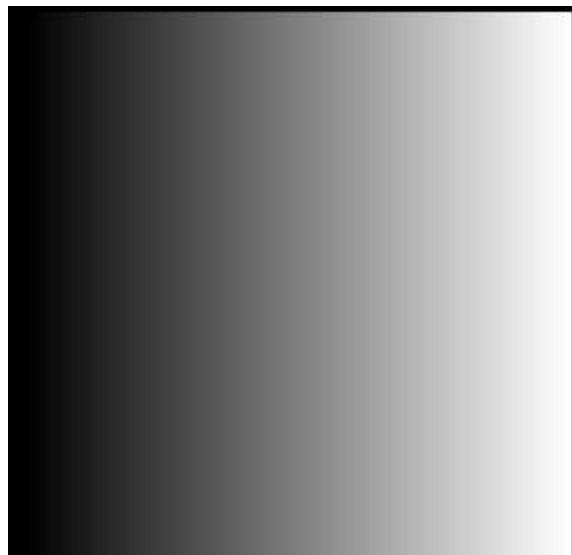
G = derajat keabuan

m = bilangan bulat positif

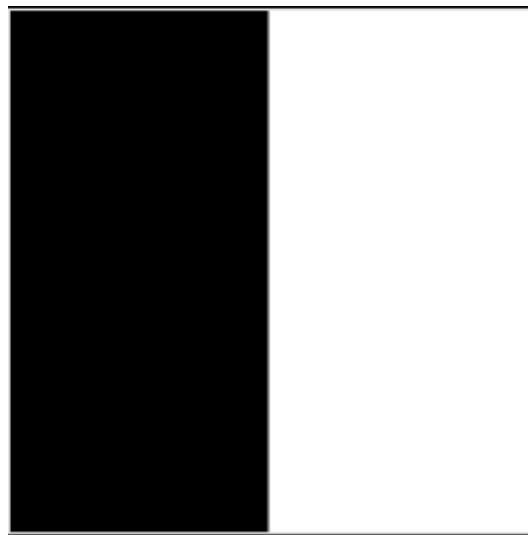
$$G = 2^m$$

Skala Keabuan	Rentang Nilai Keabuan	<i>Pixel Depth</i>
2^1 (2 nilai)	0, 1	1 bit
2^2 (4 nilai)	0 sampai 3	2 bit
2^3 (16 nilai)	0 sampai 15	3 bit
2^8 (256 nilai)	0 sampai 255	8 bit

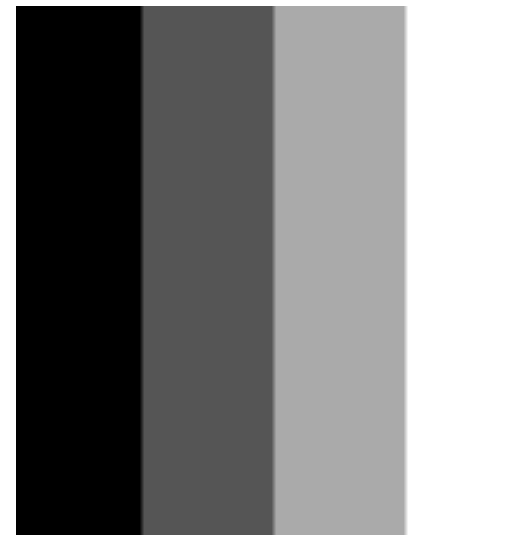




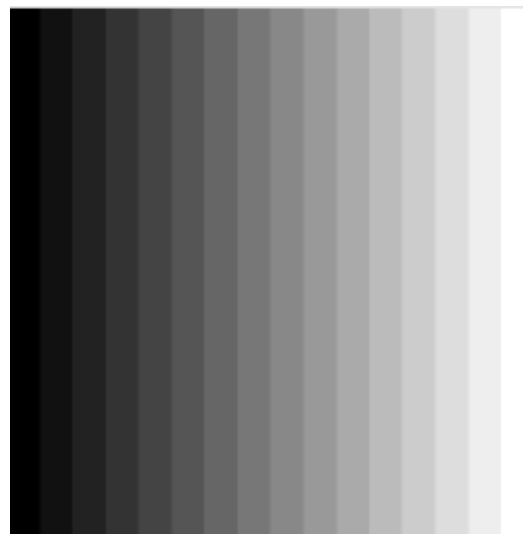
Original image



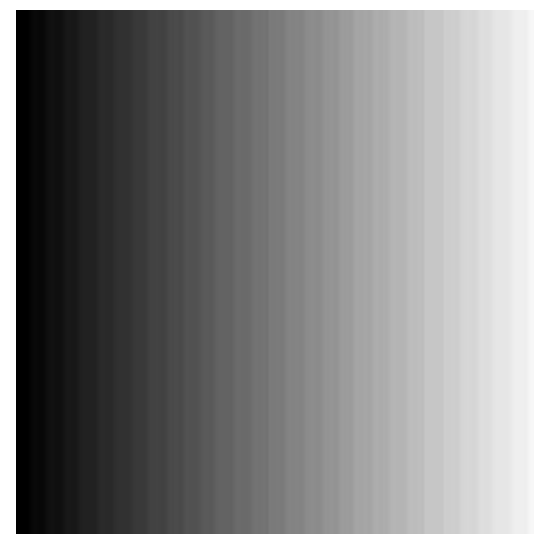
2 graylevel



4 graylevel



16 graylevel



32 graylevel

256 gray levels (8bits/pixel)



32 gray levels (5 bits/pixel)



16 gray levels (4 bits/pixel)



- 8 gray levels (3 bits/pixel)



4 gray levels (2 bits/pixel)



2 gray levels (1 bit/pixel)





(a) 256 level



(b) 128 level

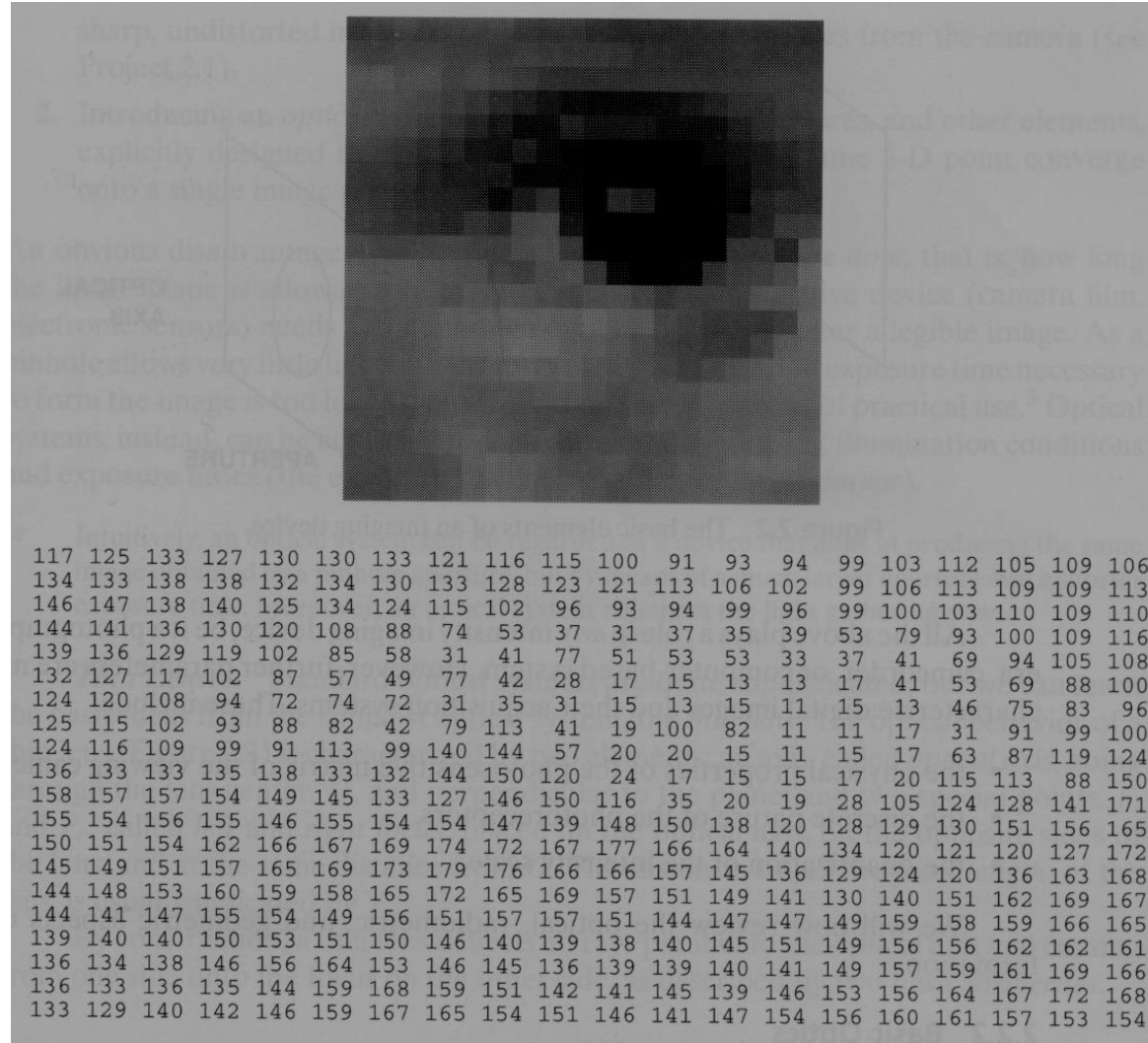
Citra Lena yang dikuantisasi ke dalam 256 level dan 128 level keabuan

1 pixel = 8 bit = 256 graylevel → 0 sampai 255

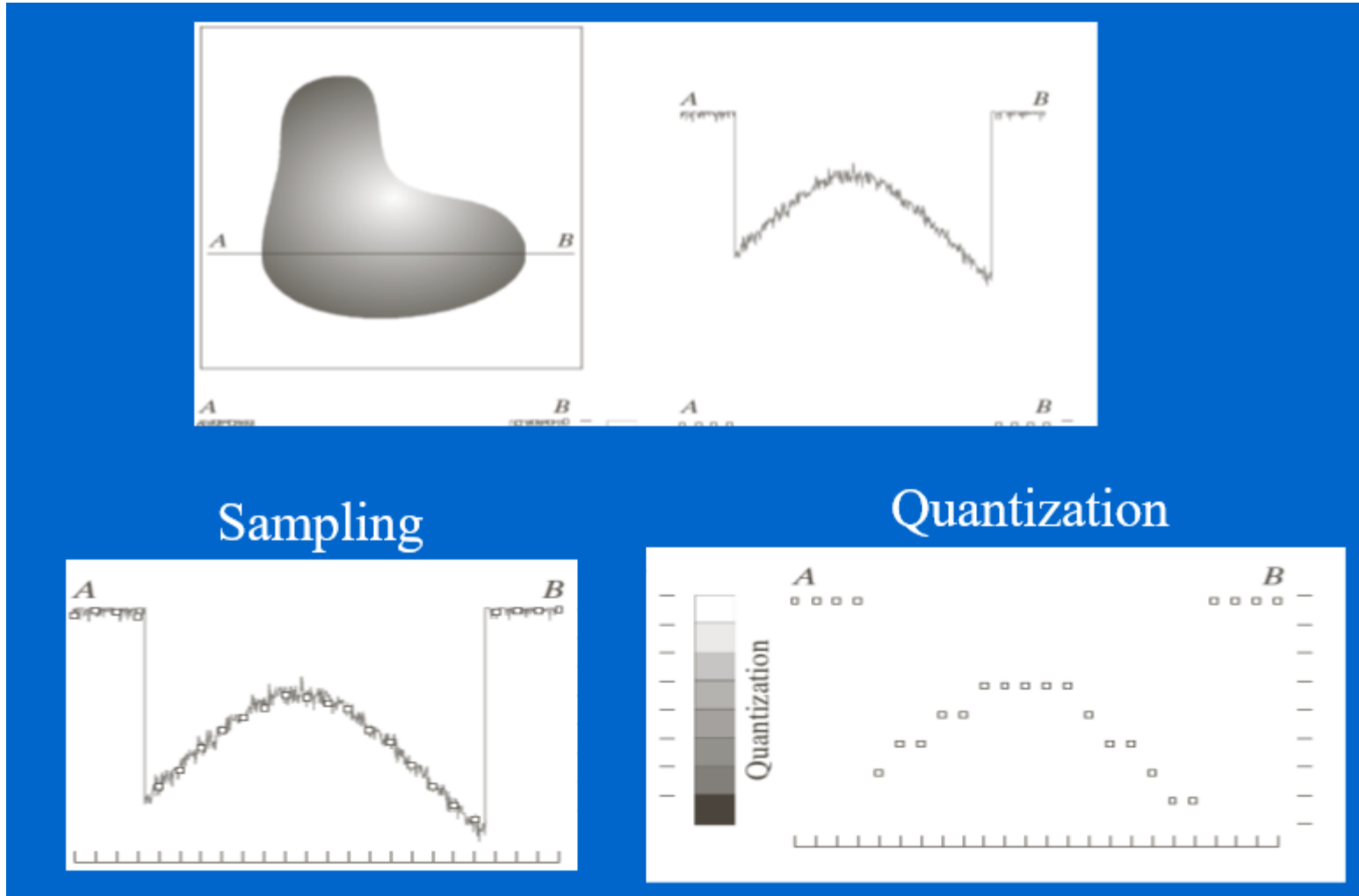


127	127	129	124	127	133	131	133	127
130	133	128	128	130	130	127	128	137
128	125	130	129	127	130	127	123	130
129	132	130	128	126	131	129	131	130
124	130	129	127	122	128	131	129	131
123	127	129	128	129	130	127	131	132

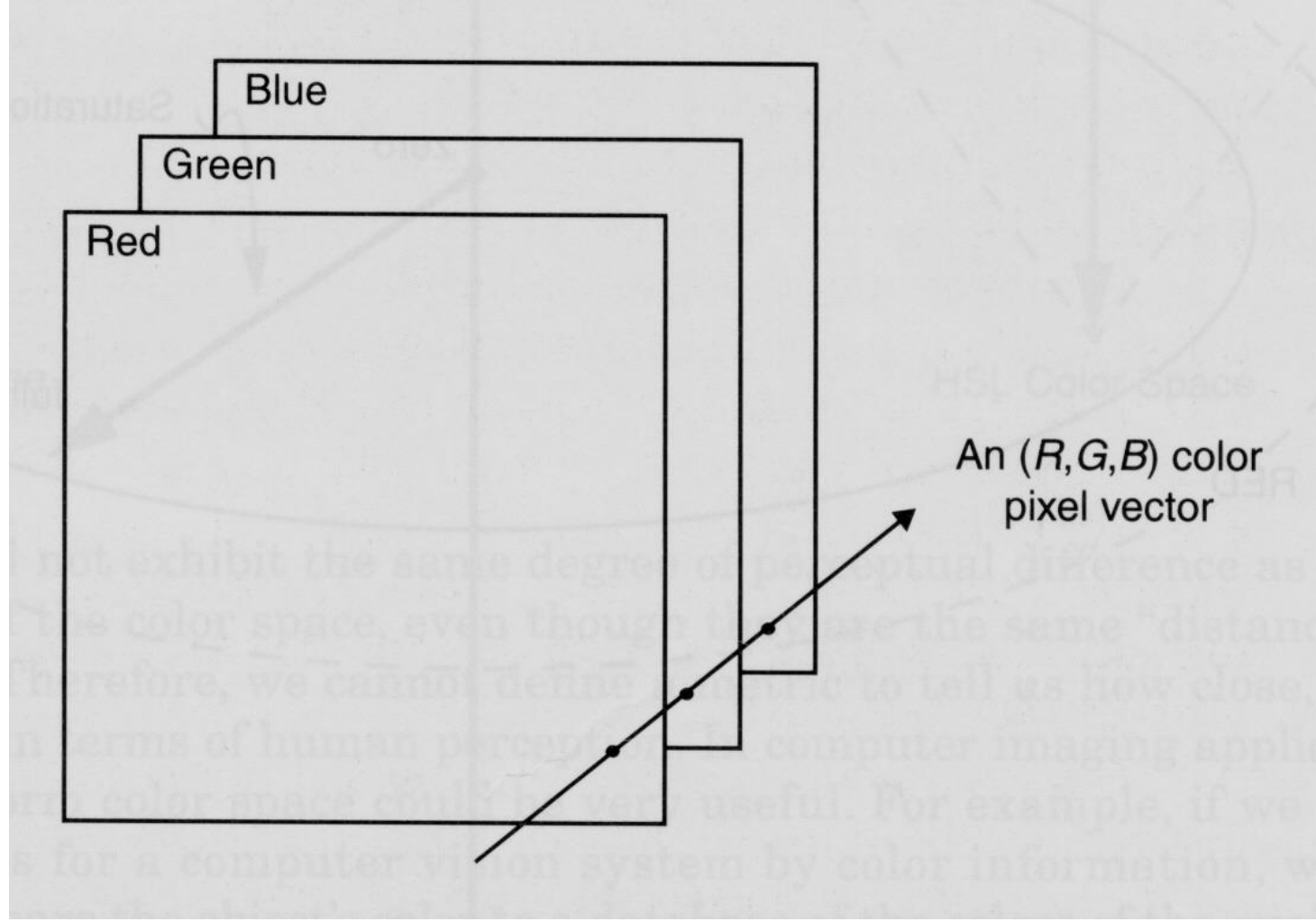
1 pixel = 8 bit = 256 graylevel → 0 sampai 255



Digitalisasi citra secara keseluruhan:



- Umumnya 256 level (*8 bit/pixel*) sudah cukup untuk merepresentasikan nilai intensitas pixel.
- Jika citra digital berukuran $N \times M$ dan setiap *pixel* kedalamannya b bit, maka kebutuhan memori untuk mereresentasikan citra adalah
$$N \times M \times b \text{ bit}$$
- Untuk citra berwarna, 256 level digunakan untuk setiap warna.
- Citra berwarna terdiri dari tiga kanal warna: – **red**, **green**, and, **blue** – Kombinasi ketiga warna menghasilkan persepsi warna-warna yang kita lihat.





=



Red



Green



Blue



Red

148	162	175	182	189	194	195	193	195	195	197
148	164	174	176	185	189	191	191	196	194	195
144	159	167	176	178	185	188	191	196	194	197
128	147	157	168	173	179	182	184	191	191	192
119	134	148	160	164	170	179	176	181	189	185
145	124	142	151	160	168	169	174	180	182	183
172	120	140	153	157	169	171	178	180	182	182
196	120	129	144	152	158	167	170	177	176	178
204	144	116	134	142	149	155	165	165	170	171

Green

42	43	48	50	53	56	56	53	54	54	54
50	49	51	47	53	55	56	55	59	55	54
51	48	47	49	49	51	50	52	54	51	54
53	48	45	49	50	52	50	48	51	50	50
59	43	43	48	47	48	54	47	49	55	50
100	42	41	42	44	46	45	46	50	52	50
142	47	43	42	39	46	44	48	49	51	49
185	65	44	42	42	43	48	46	50	48	49
209	106	44	42	41	42	44	50	48	50	49

Blue

16	24	32	35	37	40	40	37	37	38	36
19	25	31	28	34	37	38	37	40	35	33
17	23	27	33	32	35	33	36	39	35	37
20	19	23	31	33	34	34	32	36	35	35
29	16	24	33	32	34	39	30	31	38	34
71	11	18	24	30	33	30	30	34	36	34
113	14	16	21	24	32	30	32	33	35	33
156	32	13	20	25	28	33	31	35	33	32
177	72	9	16	22	26	30	35	32	33	32

Kode program Matlab untuk dekomposisi R, G, B dari citra berwarna

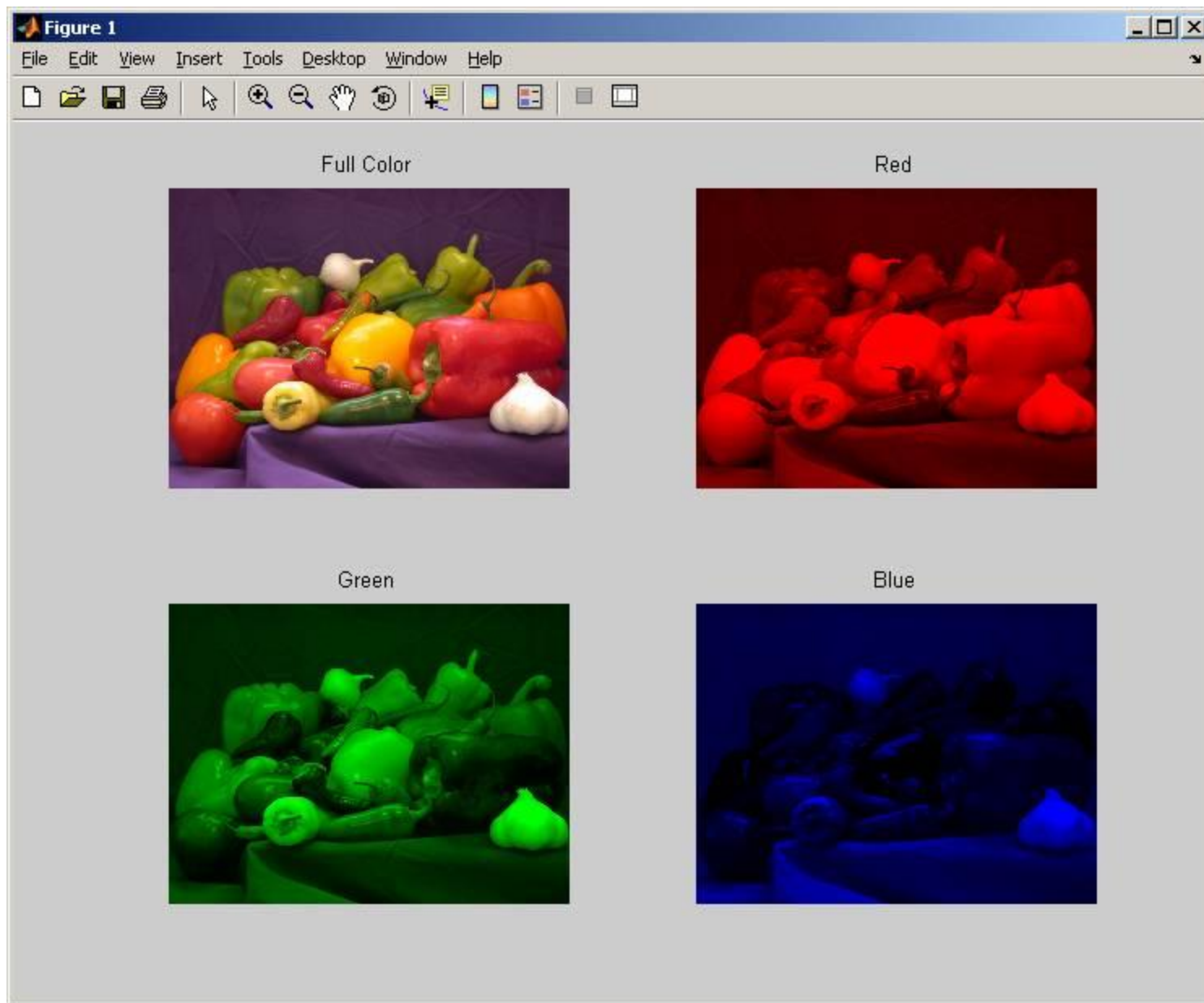
```
function fig = imColorSep(A)
% IMCOLORSEP Displays the RGB decomposition of a full-color image
%
% Syntax: fig = imColorSep(A);
%
% Example: A = imread('peppers.png');
% fig = imColorSep(A);
%
% Written by: Rick Rosson, 2007 December 26
%
% Revised: Rick Rosson, 2008 January 2
%
% Copyright (c) 2007-08 Richard D. Rosson. All rights reserved.
%
% Number of gray scale values:
N = 256;
% Make sure data type of image array is 'uint8':
A = im2uint8(A);
% Create figure window:
fig = figure;
% Display full color image:
subplot(2,2,1);
imshow(A);
title('Full Color');

% Cell array of color names:
ColorList = { 'Red' 'Green' 'Blue' };

% Gray-scale column vector: % range [ 0 .. 1 ]
gr = 0:1/(N-1):1; % increment 1/(N-1)

% Display each of the three color components:
for k = 1:3
    % color map:
    cMap = zeros(N,3);
    cMap(:,k) = gr;

    % Display monochromatic image:
    subplot(2,2,k+1);
    imshow(ind2rgb(A(:,:,k),cMap));
    title(ColorList{k});
end
end
```



Secara umum dikenal 3 jenis citra digital:

1. **Citra biner** (1 *pixel* = 1 bit)

Graylevel hanya 0 dan 1 (hitam dan putih)

2. **Citra grayscale** (1 *pixel* umumnya 8 bit)

Graylevel dari 0 sampai 255 (hitam ke putih)

3. **Citra berwarna** (24-bit RGB)

Terdiri dari tiga kanal warna: *red* (R), *green* (G), dan *blue*(B)

Graylevel pada setiap kanal warna panjangnya 8 bit



Color image (24-bit RGB)



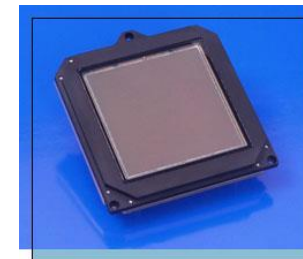
Grayscale image (8-bit)



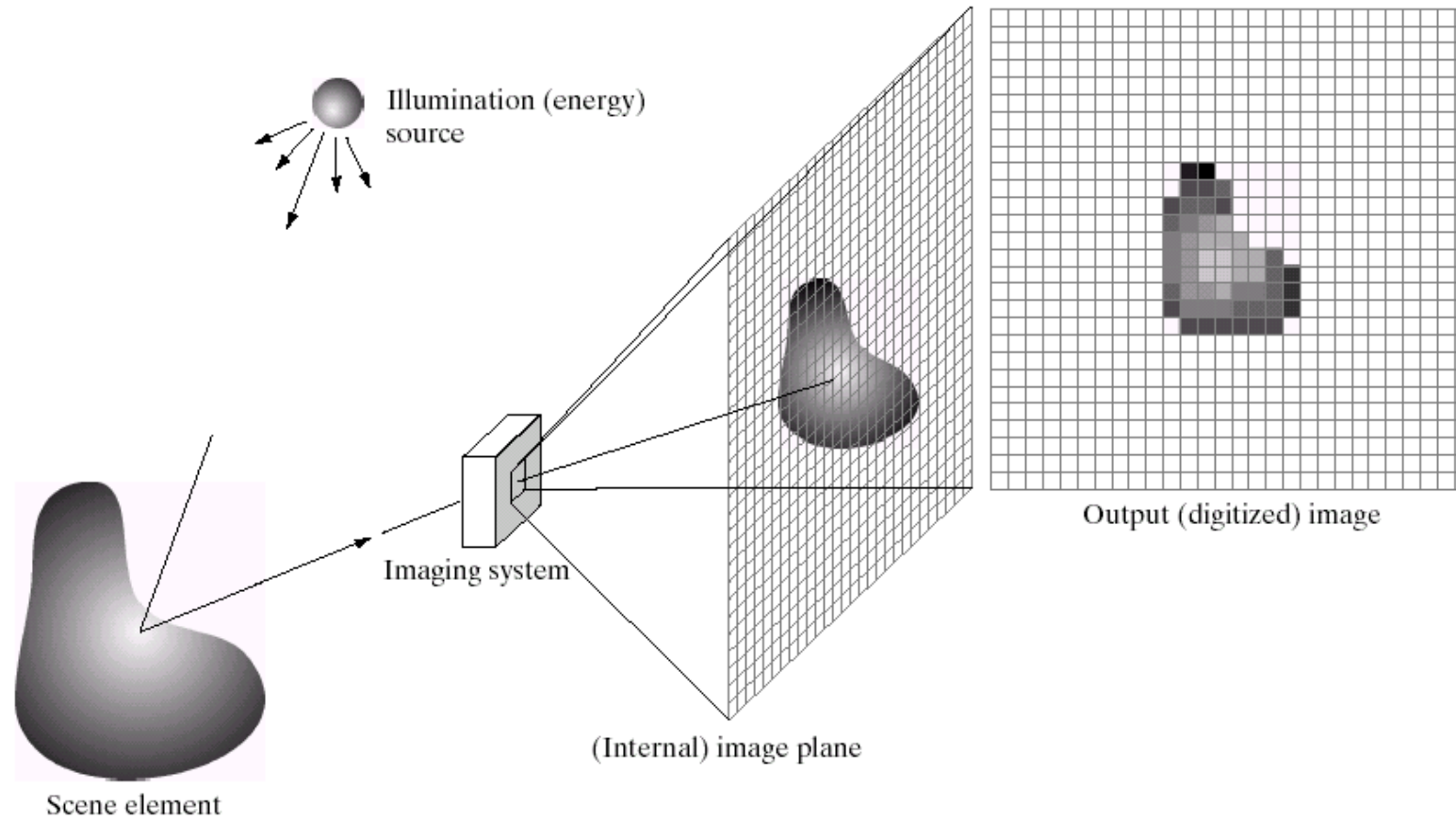
Binary image (1-bit)

Digital cameras

- A digital camera replaces film with a sensor array.
 - Each cell in the array is light-sensitive diode that converts **photons** to **electrons**
- Two common types
 - Charge Coupled Device (CCD)
 - Complementary metal oxide semiconductor (CMOS)

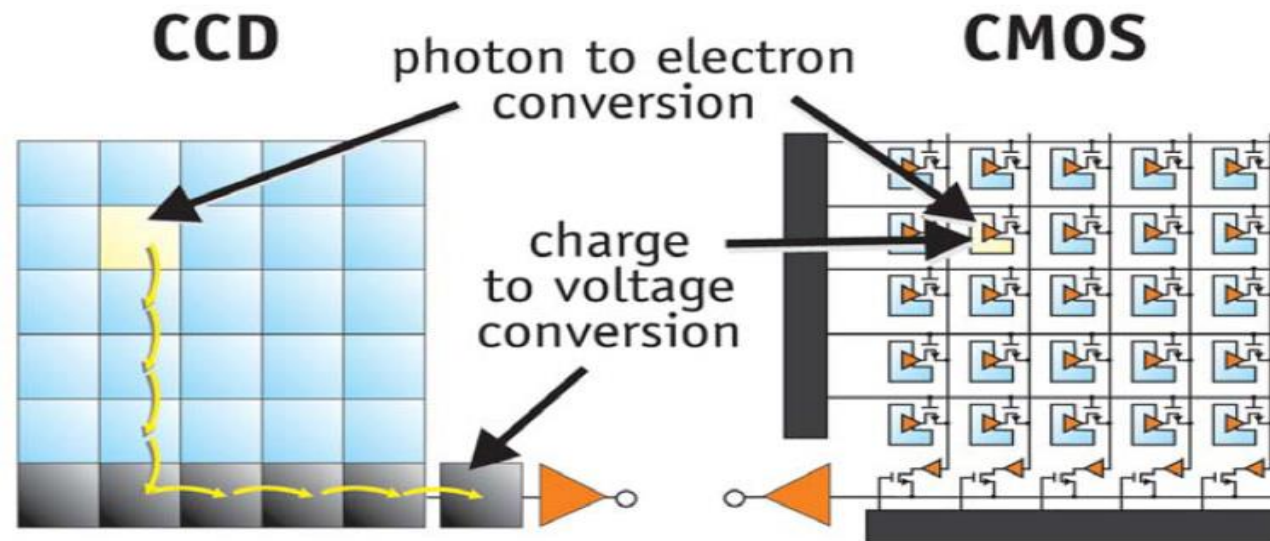


Proses di-digitisasi dengan kamera digital



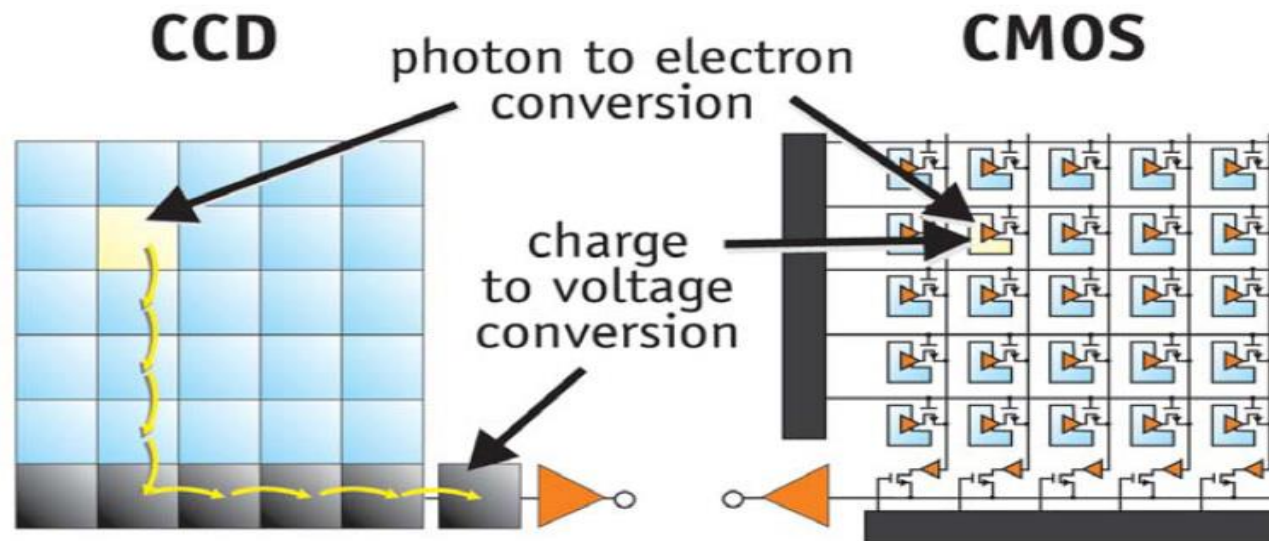
CCD Cameras

- CCDs move photogenerated charge from pixel to pixel and convert it to voltage at an output node.
- An **analog-to-digital converter (ADC)** then turns each pixel's value into a digital value.



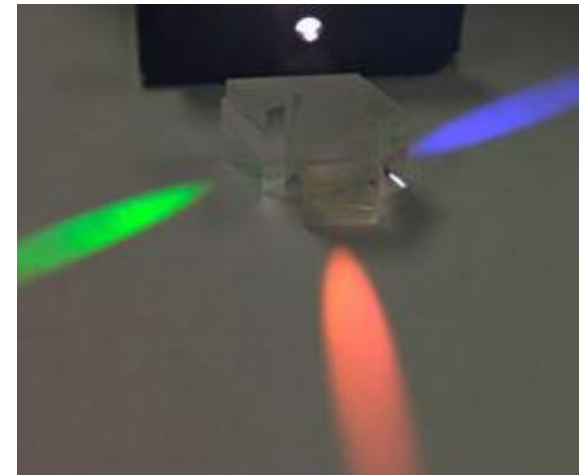
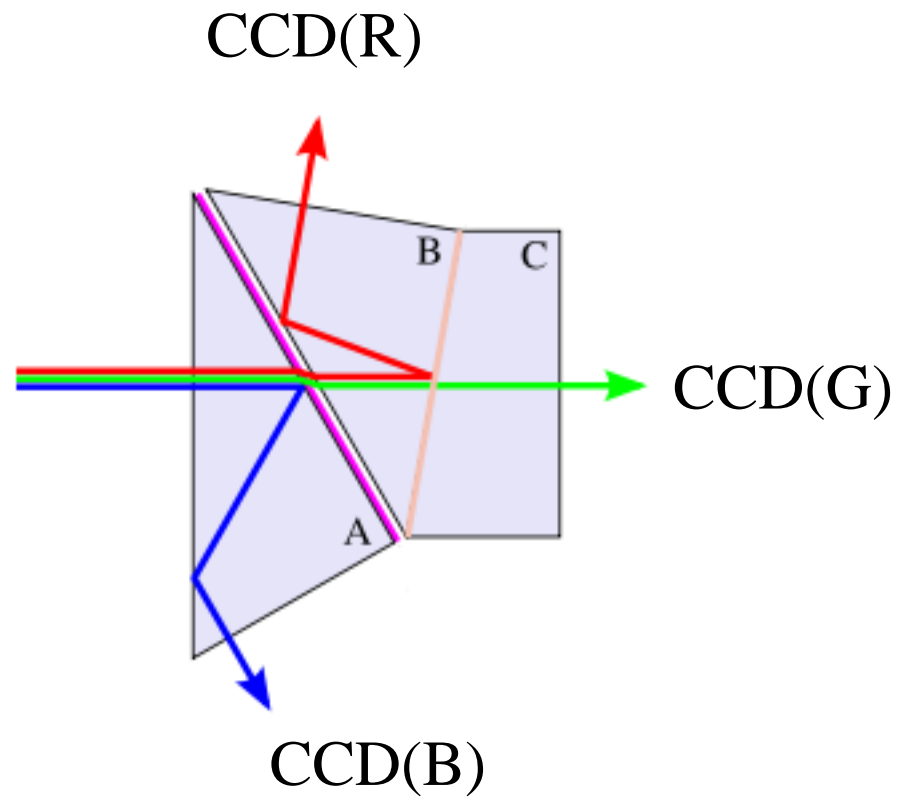
CMOS Cameras

- CMOSs convert charge to voltage inside each element.
- Uses several transistors at each pixel to amplify and move the charge using more traditional wires.
- The CMOS signal is digital, so it needs no ADC.



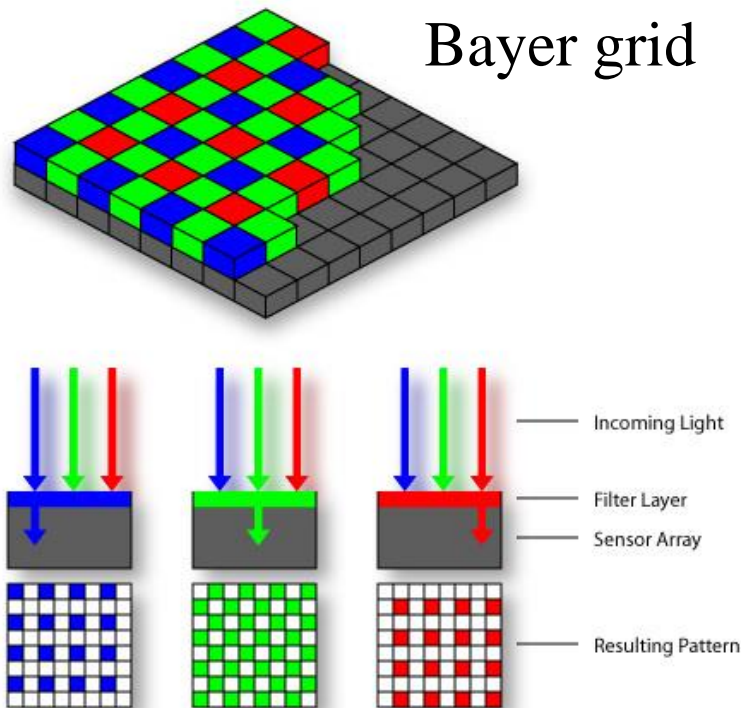
Color sensing in camera: Prism

- Requires three chips and precise alignment.

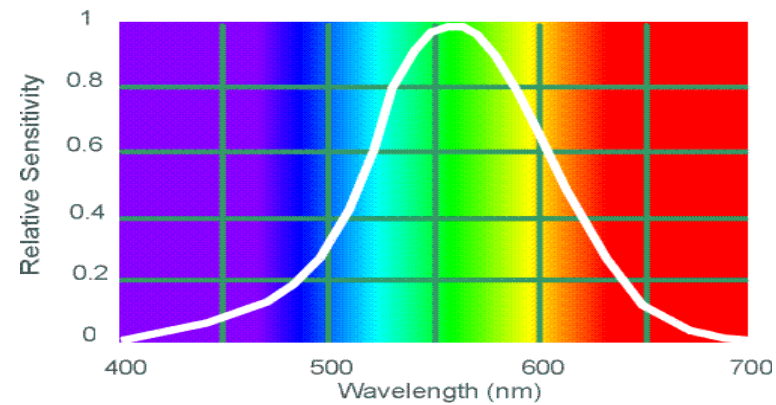


Color sensing in camera: Color filter array

- In traditional systems, color filters are applied to a single layer of photodetectors in a tiled mosaic pattern.

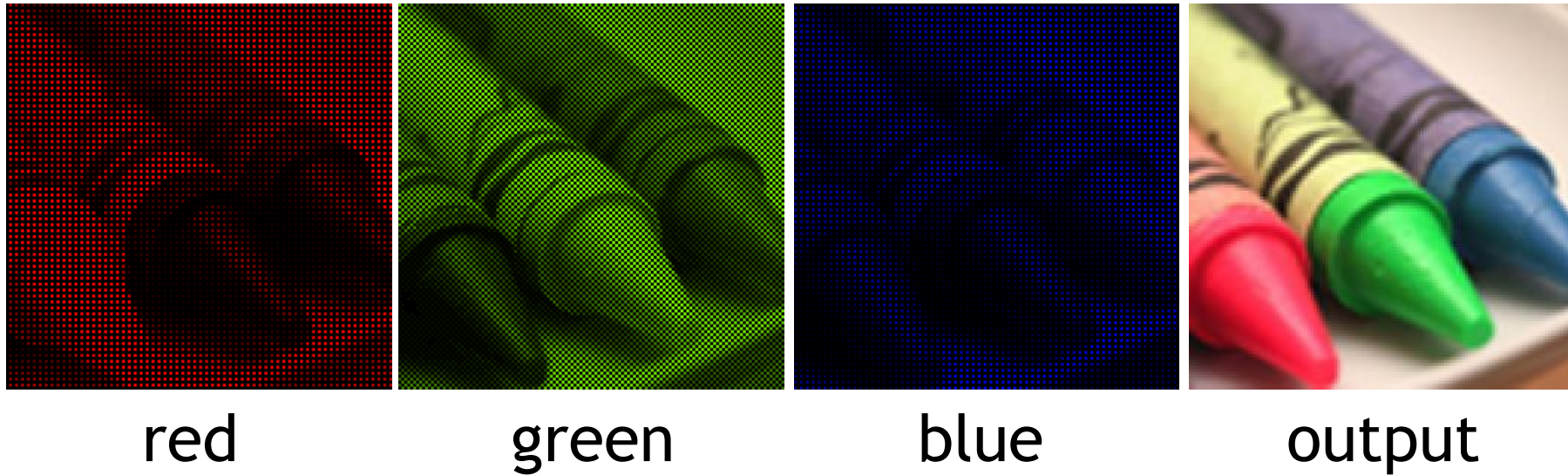


Why more green?



Human Luminance Sensitivity Function

Color sensing in camera: Color filter array



G	R	G	R
B	G	B	G
G	R	G	R
B	G	B	G

demosaicing
(interpolation)



rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb
rGb	Rgb	rGb	Rgb
rgB	rGb	rgB	rGb

Referensi

1. Dr. George Bebis, *Image Formation and Representation*, CS485/685
Computer Vision
2. Antonio, R. C., Paiva, *Image representation, sampling, and quantization*,
ECE 6962 – Fall 2010
3. Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*,
Prentice Hall