Peningkatan Kualitas Citra

Domain Spasial

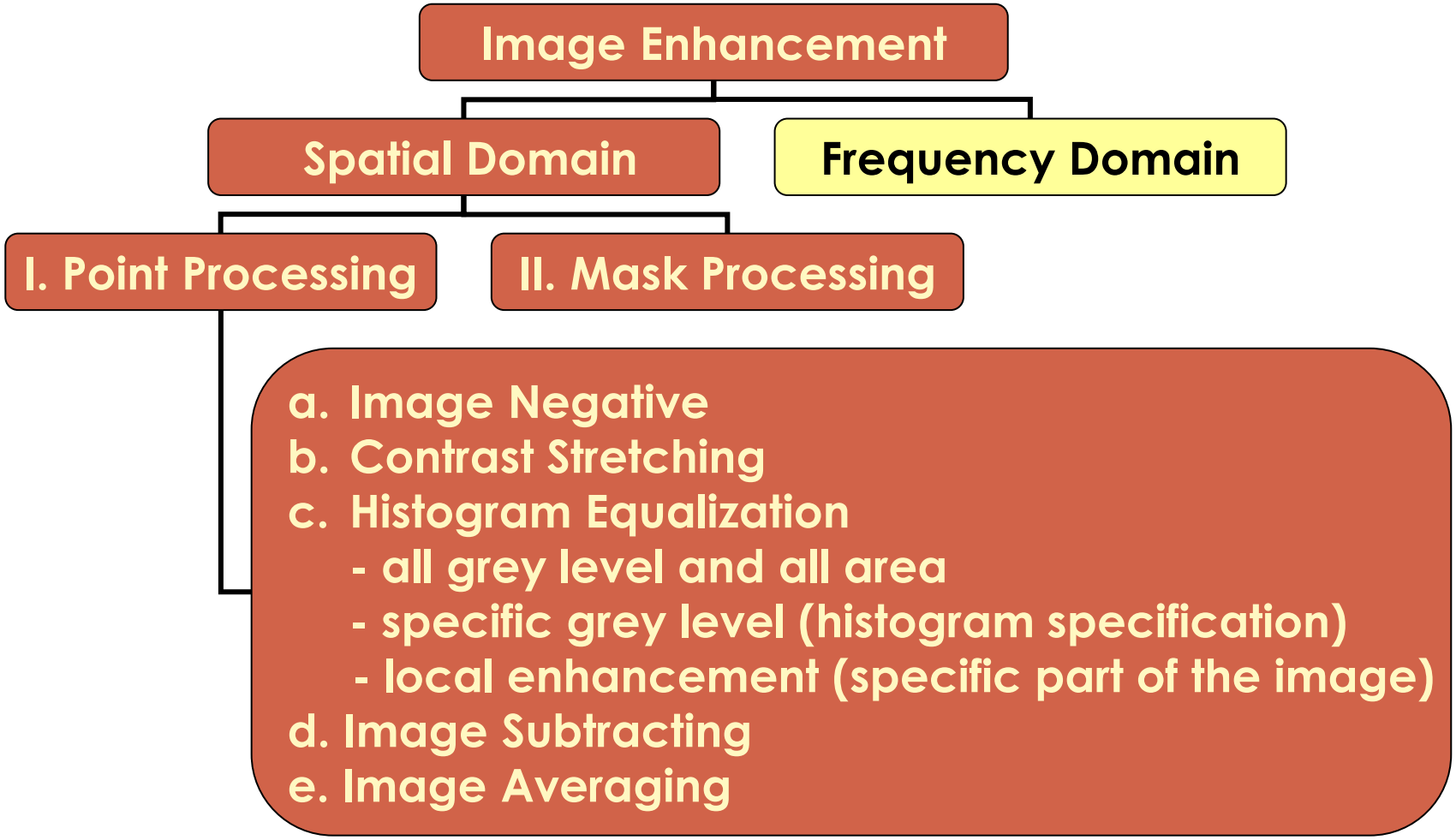
Tujuan Perbaikan Citra

- Tujuan dari teknik peningkatan mutu citra adalah untuk melakukan pemrosesan terhadap citra agar hasilnya mempunyai kwalitas relatif lebih *baik* dari citra awal untuk aplikasi tertentu.
- Kata *baik* disini tergantung pada jenis aplikasi dan problem yang dihadapi

Jenis Teknik Peningkatan Kualitas

- Teknik peningkatan kualitas citra dapat dibagi menjadi dua:
  - Peningkatan kualitas pada domain spasial
    - Point processing
    - Mask processing
  - Peningkatan kualitas pada domain frekuensi

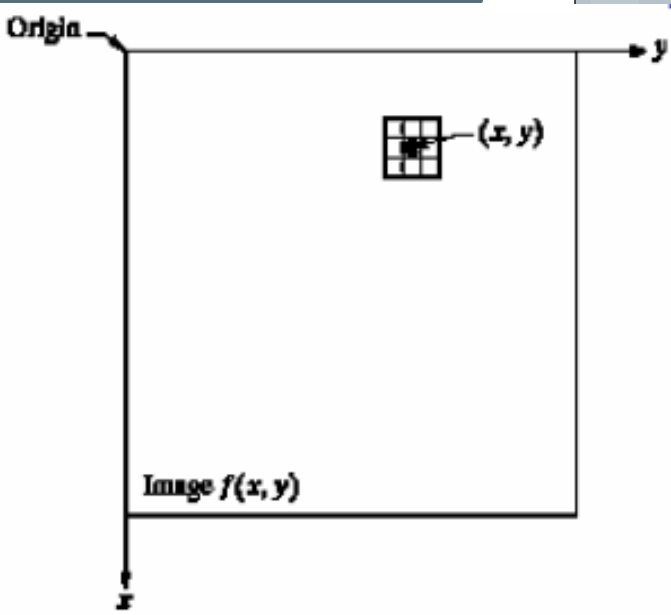
Lingkup Pembahasan



Point Processing

- Cara paling mudah untuk melakukan peningkatan kualitas citra pada domain spasial adalah dengan melakukan pemrosesan yang hanya melibatkan satu piksel saja (tidak menggunakan jendela ketetanggaan)
- Pengolahan menggunakan histogram juga termasuk dalam bagian point processing

Domain Spasial



- Prosedur yang secara langsung memanipulasi pixel.

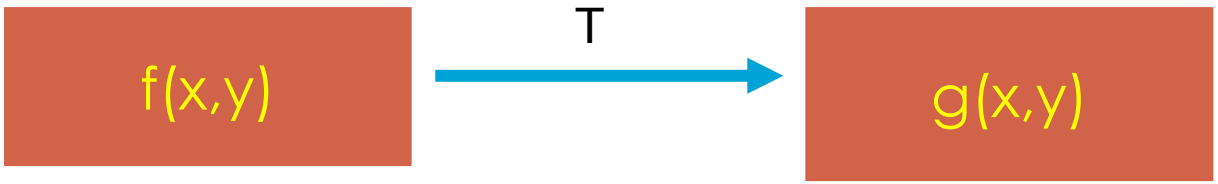
$$g(x,y) = T[f(x,y)]$$

dimana

f(x,y) adalah image input

g(x,y) adalah image yang diproses

T adalah sebuah operator pada f yang didefinisikan berdasar nilai neighborhood dari (x,y)



Operator T

- Operator T dapat berupa :
  - Kumpulan pixels (x,y) dari image
  - Kumpulan dari ‘neighborhoods’ N(x,y) dari setiap pixel
  - Kumpulan dari images f1,f2,f3,…

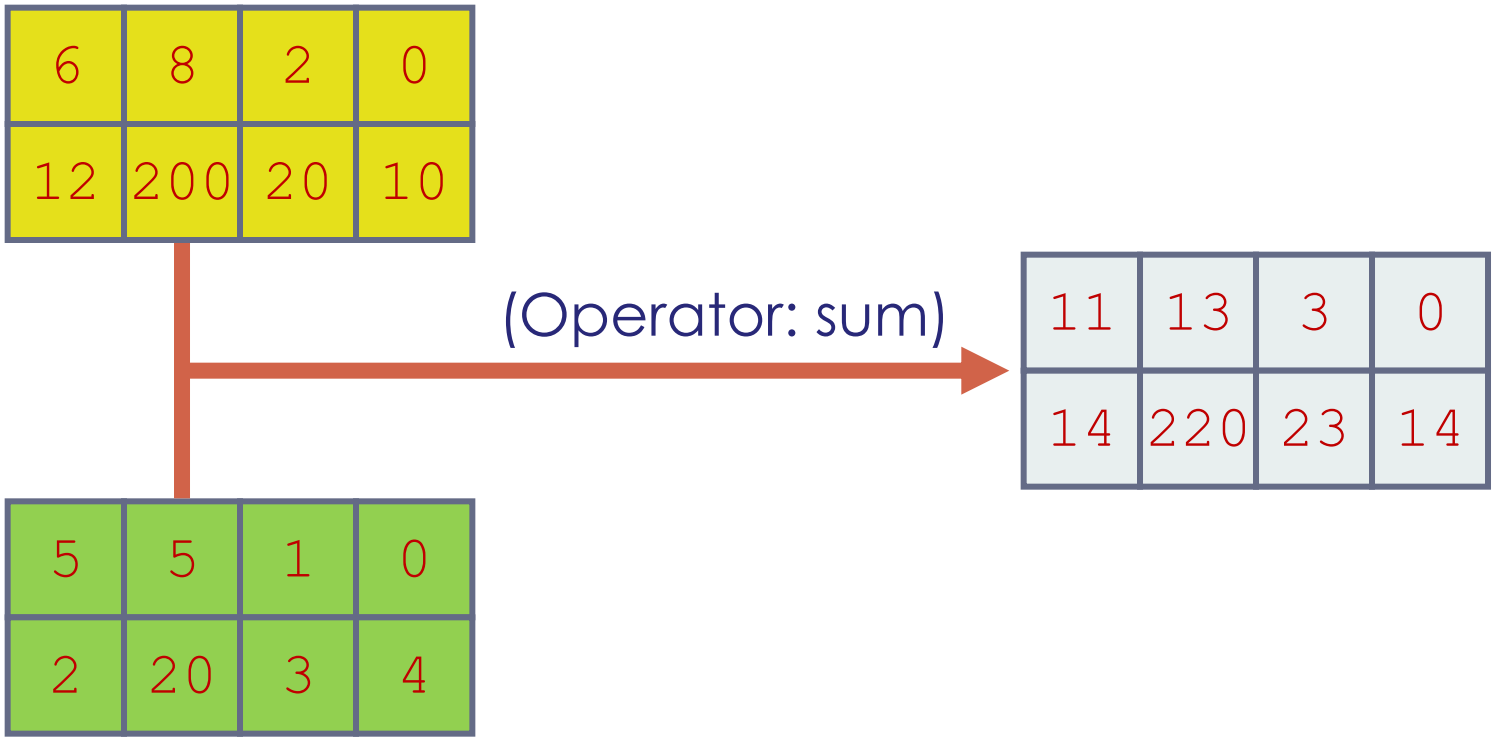
Contoh Operator : div by 2

- Operasi terhadap himpunan pixel dari image

(Operator: Div. by 2)

Contoh Operator : sum

- Operasi terhadap kumpulan image f1,f2,…



Point Processing

- Point processing, tetangga 1x1 piksel
  - Output pixel pada titik tertentu hanya bergantung pada input pixel pada titik tersebut dan tidak bergantung pada nilai pixel tetangganya
- g hanya bergantung pada nilai f pada posisi (x,y)
- T = fungsi transformasi gray level (atau intensitas mapping)

$$g(x,y) = T[f(x,y)]$$

Konversi RGB ke Graylevel

- Ada tiga pendekatan
  - Lightness Method

    Gray = (max(R, G, B) + min(R, G, B)) / 2

  - Average Method

    Gray = (R + G + B) / 3

  - Luminosity Method

    Gray = 0.21 R + 0.71 G + 0.07 B

Transformasi Graylevel Dasar

- Teknik perbaikan citra sederhana
- Nilai piksel sebelum dan sesudah proses dinotasikan dengan **r** dan **s**, dimana **s=T(r)**
  - Image Negative
  - Log Transformations
  - Power-Law Transformations
  - Piecewise-Linear Transformation Functions

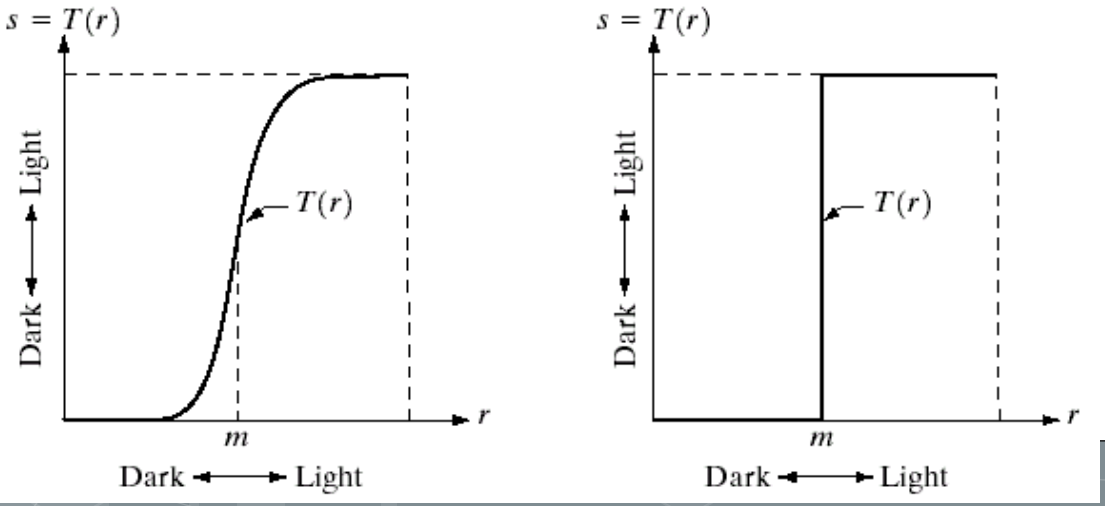
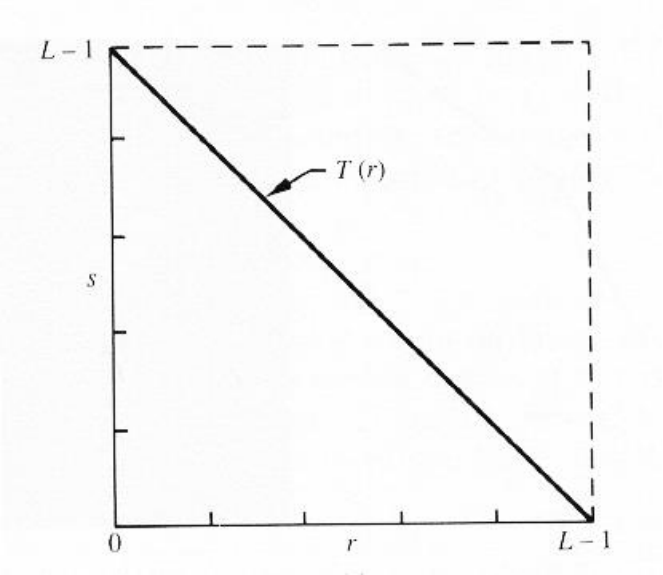


Image Negative

- Didapat dengan menerapkan fungsi transformasi s = T(r) = L- 1 – r; dimana L = Max Gray Value



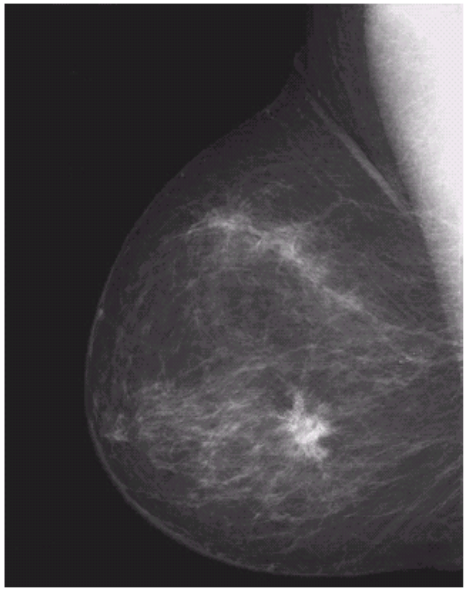


Image Negative : Algoritma

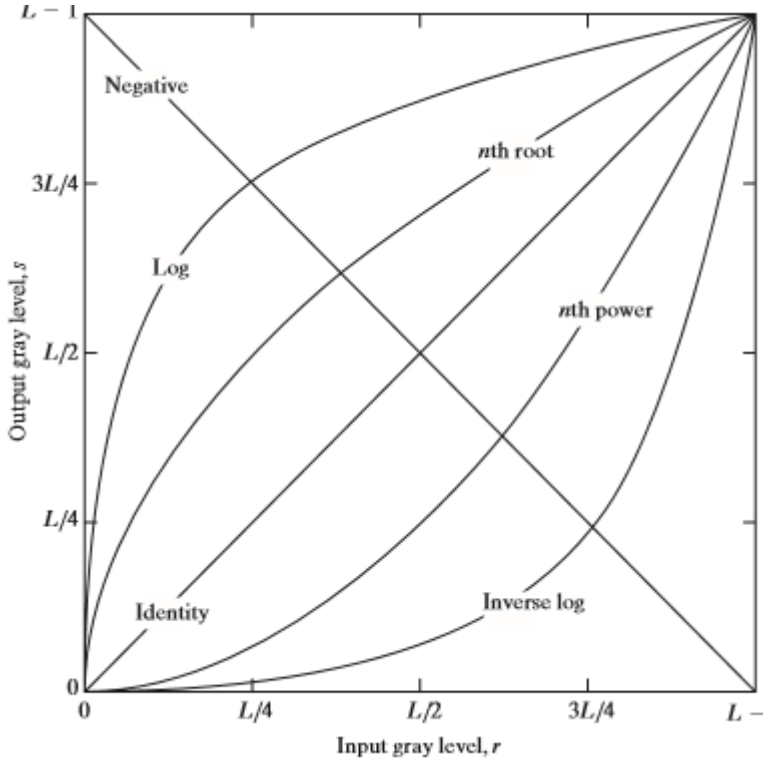
- Algoritma proses transformasi image negative
  - Ambil Citra (I)
  - Jika I dalam RGB konversikan ke Graylevel
  - Hitung nilai maksimum (Max)dari Graylevel pada I
  - Untuk setiap baris pada indeks x, lakukan
  - Untuk setiap kolom pada indeks y, lakukan
  - Hasil(x,y) = Max – I(x,y)
  - Tampilkan Hasil

Log Transformations

- Bentuk umum :

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

dimana c = konstanta dan diasumsikan $r \geq 0$



Log Transformations

InvLog

Log

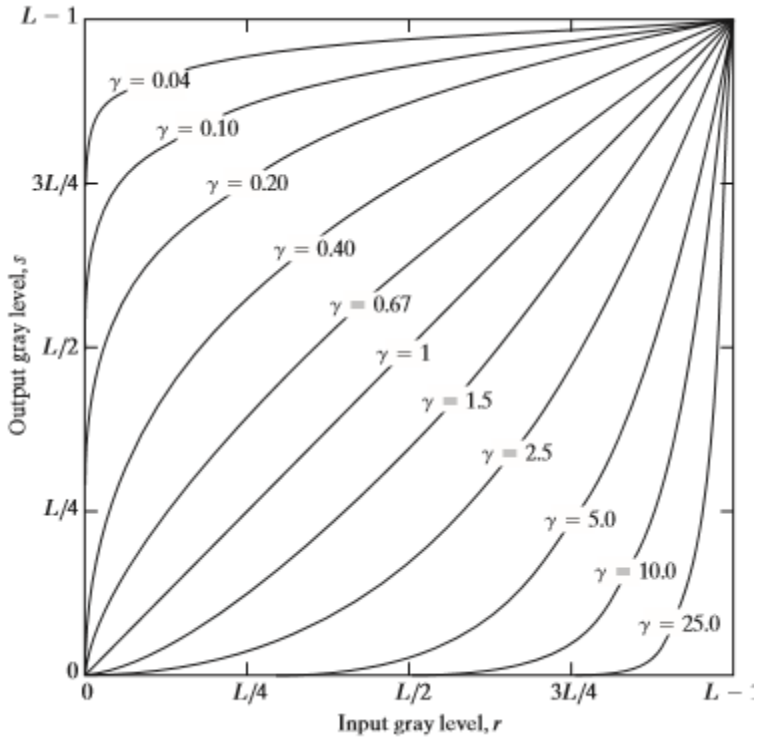
Log Transf : Algoritma

- Algoritma proses transformasi log
  - Ambil Citra (I)
  - Jika I dalam RGB konversikan ke Graylevel
  - Tentukan nilai konstanta c
  - Untuk setiap baris pada indeks x, lakukan
  - Untuk setiap kolom pada indeks y, lakukan
  - Hasil(x,y) = c * log(1+I(x,y))
  - Tampilkan Hasil

Power-Law Transformations

- Bentuk umum : $s = cr^{\gamma}$

dimana c dan γ = konstanta positif



Power-Law Transformations

- Image terlalu banyak didominasi oleh warna cerah, maka diperlukan ekspansi nilai gray value dengan $\gamma > 1$

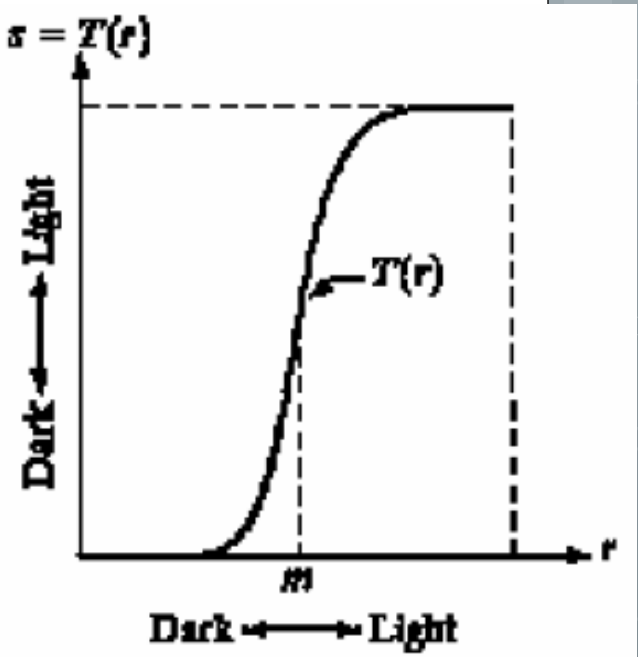
C=1 dan γ=3, 4, dan 5

Power-Law Transf : Algoritma

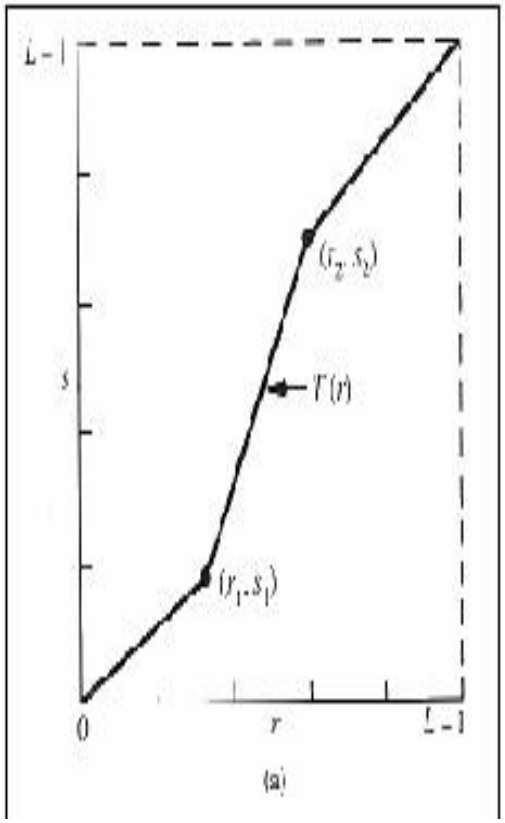
- Algoritma proses transformasi Power-Law
  - Ambil Citra (I)
  - Jika I dalam RGB konversikan ke Graylevel
  - Tentukan nilai konstanta c
  - Tentukan nilai konstanta γ
  - Untuk setiap baris pada indeks x, lakukan
  - Untuk setiap kolom pada indeks y, lakukan
  - Hasil(x,y) = c *(I(x,y)^ γ)
  - Tampilkan Hasil

Piecewise-Linear Transformation Functions

- Contrast Stretching
- Menghasilkan nilai contrast yang lebih besar dari image original, dengan cara :
  - Menggelapkan (darkening) level dibawah m dari image asli.
  - Mencerahkan (Brightening) level y
    atas m dari image asli.

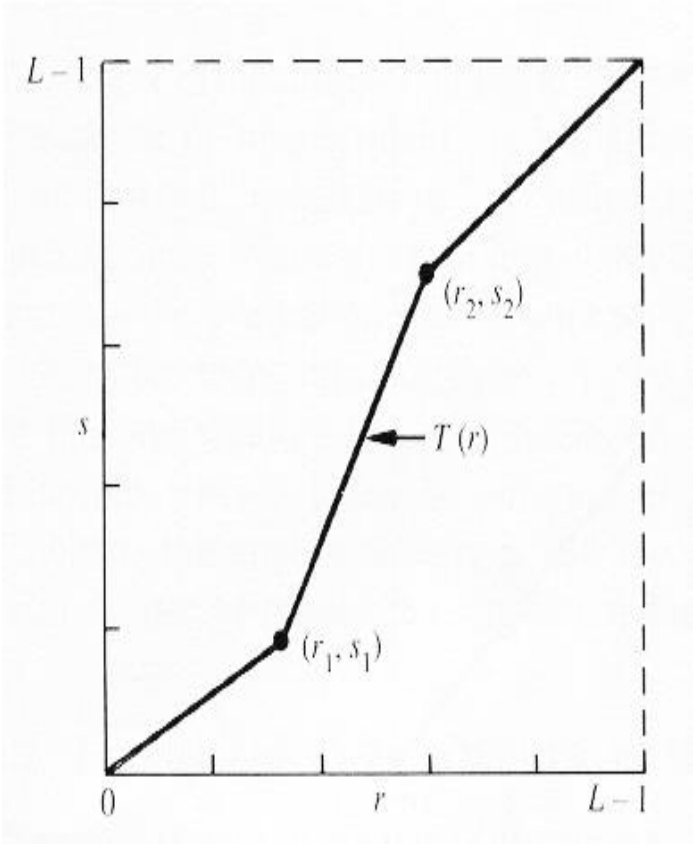


Contrast Stretching



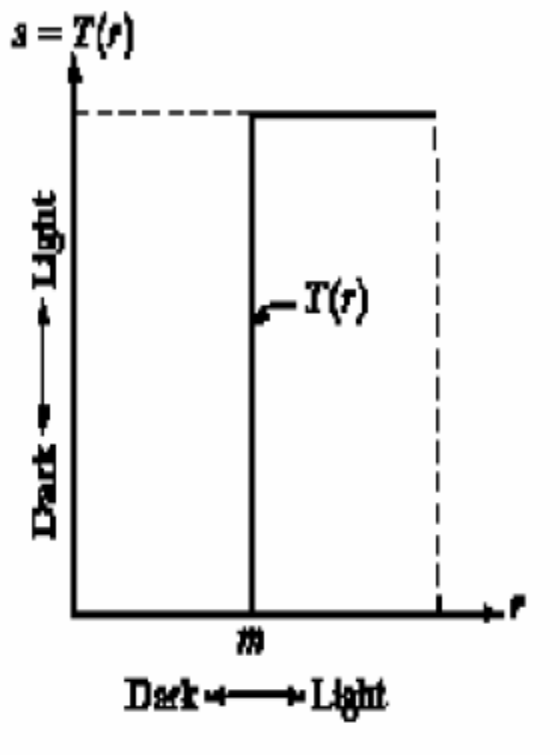
Fungsi Transformasi

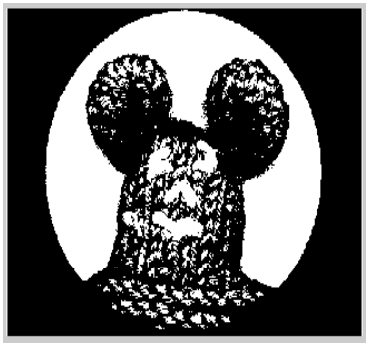
- Lokasi dari (r_1,s_1) dan (r_2,s_2) menjadi kontrol dari bentuk fungsi transformasi.
  - Bila $r_1= s_1$ dan $r_2= s_2$ transformasi adalah fungsi linear dan hasilnya adalah tidak ada perubahan image.
  - Bila $r_1=r_2$, $s_1=0$ dan $s_2=L-1$, transformasi akan berubah menjadi sebuah fungsi thresholding yang menghasilkan sebuah binary image.

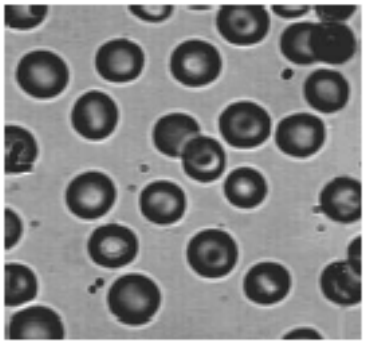


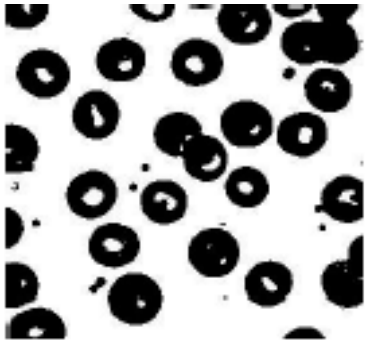
Binerisasi

- Menghasilkan image biner (two level image)







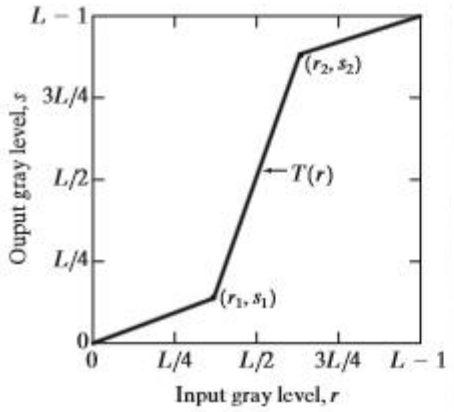


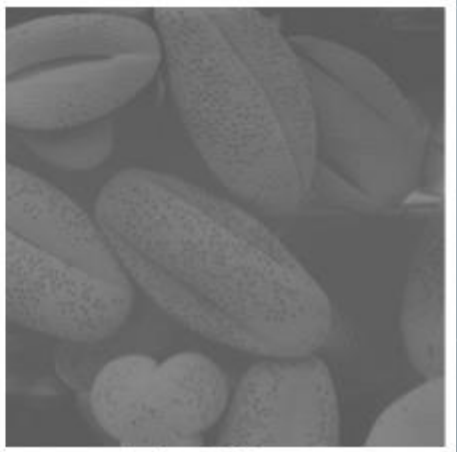
Contoh Contrast Stretching

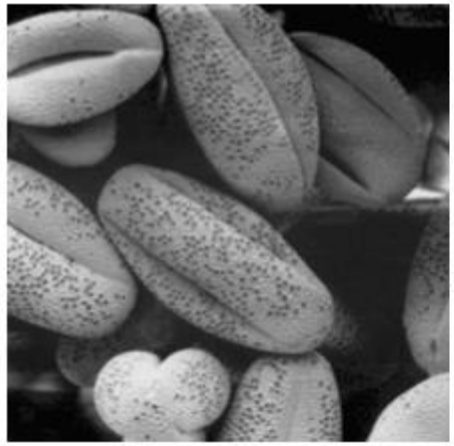
- Meningkatkan range dinamis dari gray level dalam image.
- Contoh sebuah image yang bersifat low contrast. Image semacam ini dapat dihasilkan proses iluminasi yang jelek atau setting lensa yang kurang baik saat proses akuisisi.
- Hasil dari contrast Stretching :

  (r1,s1) = (rmin,0) dan

  (r2,s2) = (rmax,L-1)
- Hasil dari proses thresholding





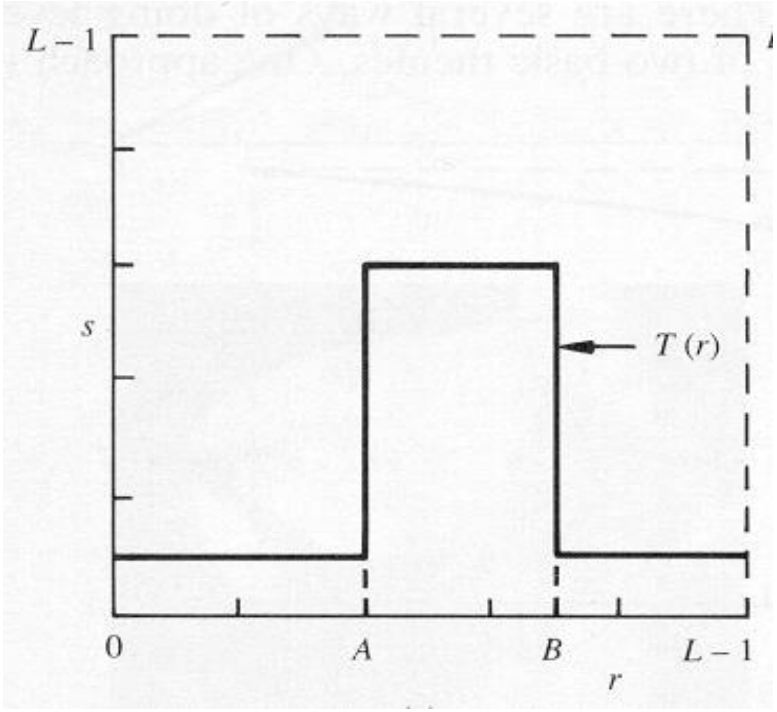


Contrast Stretching : Algoritma

- Algoritma proses Contrast Stretching
  - Ambil Citra (I)
  - Jika I dalam RGB konversikan ke Graylevel
  - Tentukan nilai (r1,s1) dan (r2,s2)
  - Buat garis dari titik (0,0) ke (r1,s1)→ A
  - Buat garis dari titik (r1,s1) ke (r2,s2)→ B
  - Buat garis dari titik (r2,s2) ke (rMax,sMax)→ C
  - Untuk setiap baris pada indeks x, lakukan
  - Untuk setiap kolom pada indeks y, lakukan
  - Jika nilai graylevel I(x,y) < r1 maka gunakan A
  - Jika nilai graylevel I(x,y) >= r1 dan I(x,y) <= r2 maka gunakan B
  - Jika nilai graylevel I(x,y) > r2 maka gunakan C
  - Tampilkan Hasil

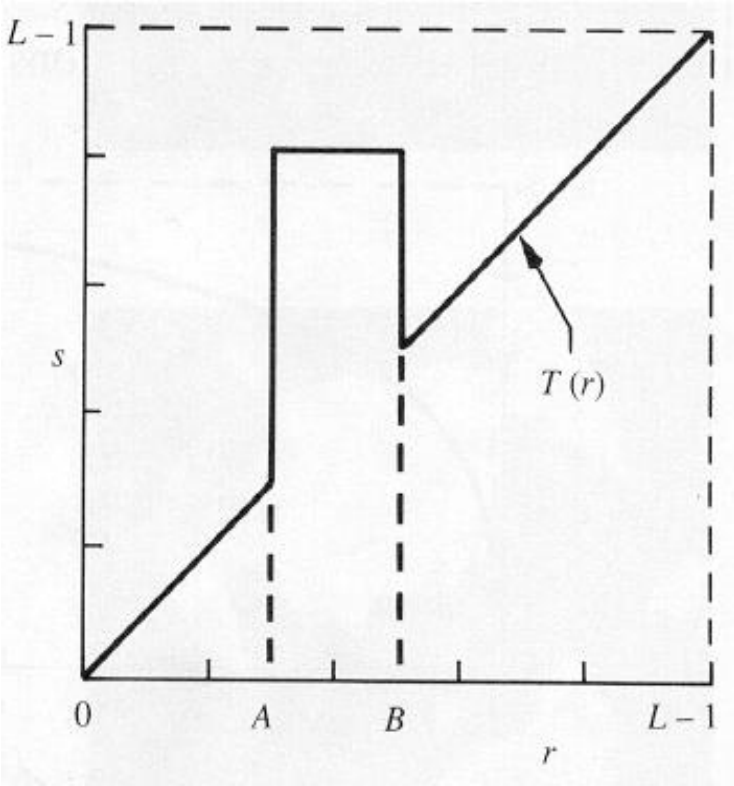
Gray Level Slicing

- Pemilihan range gray level tertentu dari image (misalnya untuk meningkakan penampakan feature tertentu).
- Salah satu cara yang dapat dilakukan adalah menampilkan gray level dari feature tertentu sebagai nilai high value dan sisanya sebagai low value sehingga didapat bentuk binary image).

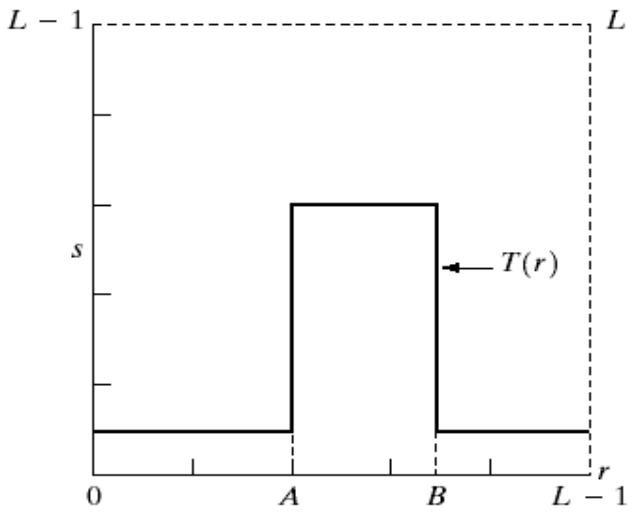


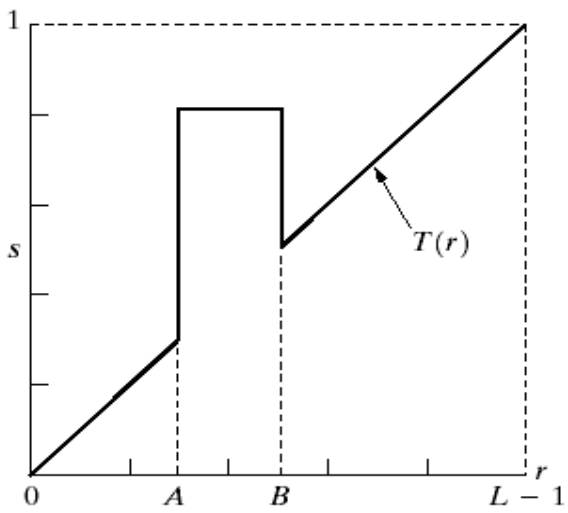
Gray Level Slicing

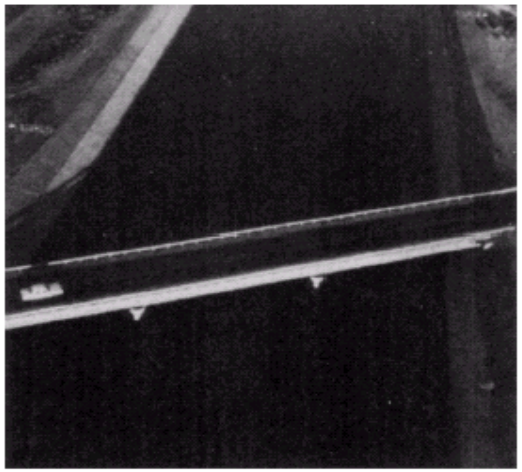
- Pendekatan yang lain adalah menerangkan bagian feature yang menjadi fokus namun tetap mempertahankan nilai backgroud dan nilai gray level image bagian image lainnya.

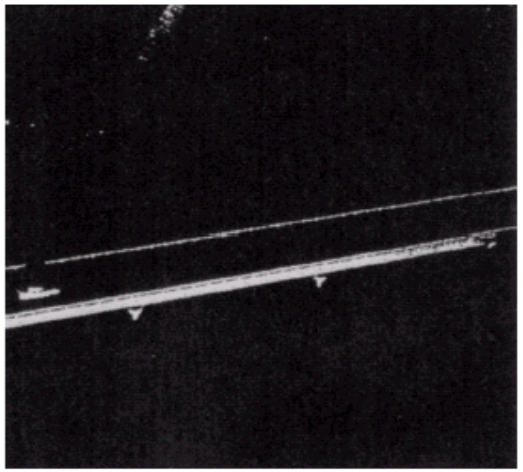


Contoh penerapan graylevel slicing







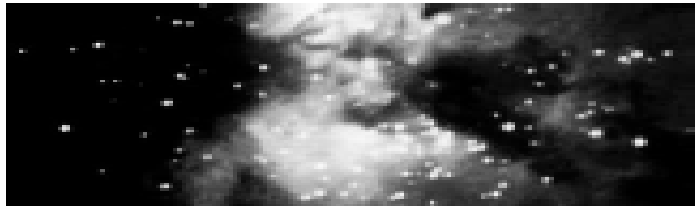


a b
c d

FIGURE 3.11
(a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

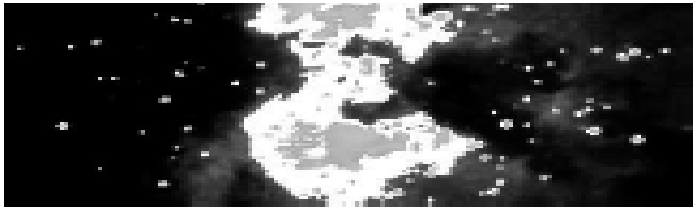
Contoh penerapan graylevel slicing

Original Image



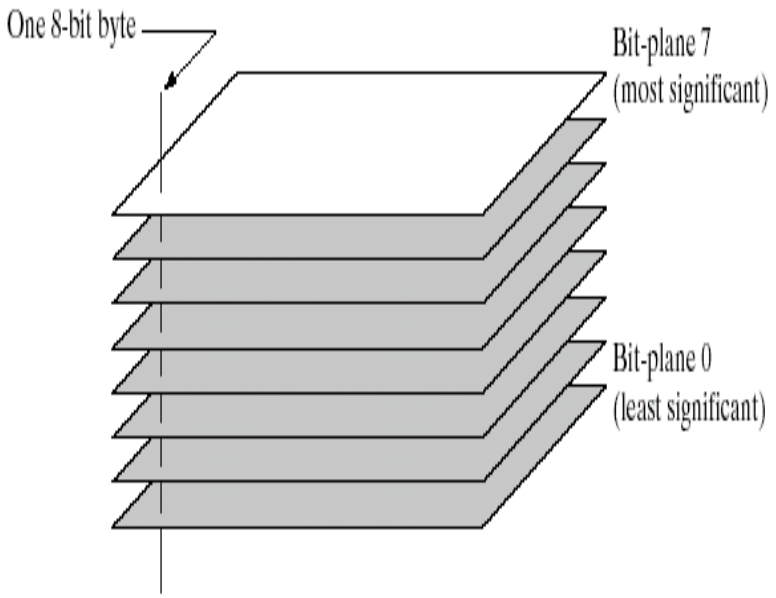
Highlighted Image with no background

Highlighted Image with background



Bit-Plane Slicing

- Untuk melihat fokus terhadap kontribusi dari penampilan image berdasar bit yang spesifik.
  - Diasumsikan bahwa setiap pixel direpresentasikan oleh 8 bits, dan sebuah image terdiri dari 8 1-bit planes.
  - Plane 0 memuat lowest order bits dari byte yang menyusun pixel dan plane 7 memuat highest-order bits.
  - Hanya 5 highest order bits yang memuat data visual yang signifikan. Plane 7 berkorespodensi langsung dengan treshold image pada gray level 128.



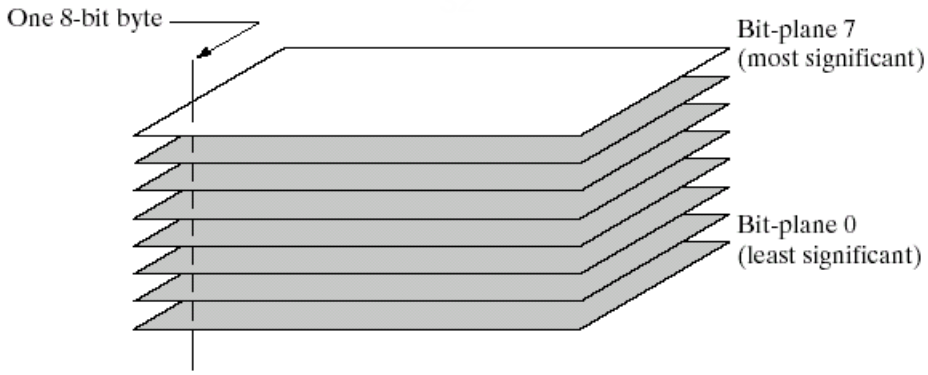


FIGURE 3.12 Bit-plane representation of an 8-bit image.

255 138 30
65 12 201
180 111 85

MSB plane
1 1 0
0 0 1
1 0 0

LSB plane
1 0 1
1 0 1
1 1 1

	LSB							MSB
255	1	1	1	1	1	1	1	1
138	0	1	0	1	0	0	0	1
30	1	1	1	1	1	0	0	0
65	1	0	0	0	0	0	1	0
12	0	0	1	1	0	0	0	0
201	1	0	0	1	0	0	1	1
183	1	1	1	0	1	1	0	1
111	1	1	1	1	0	1	1	0
85	1	0	1	0	1	0	1	0

Bit-Plane Slicing

- Menampilkan bit-bit penyusun image sebagai individual binary image.

Bit-Plane Slicing : Algoritma

- Algoritma proses Bit-Plane Slicing
  - Ambil Citra (I) dalam 8 bit
  - Jika I dalam RGB konversikan ke Graylevel
  - Masing-masing nilai graylevel dalam I ubah ke bentuk notasi bit
  - Pisahkan deretan bit ke dalam 8 array
  - Bangun citra untuk setiap bit array tersebut ke dalam citra Hasil
  - Tampilkan Hasil

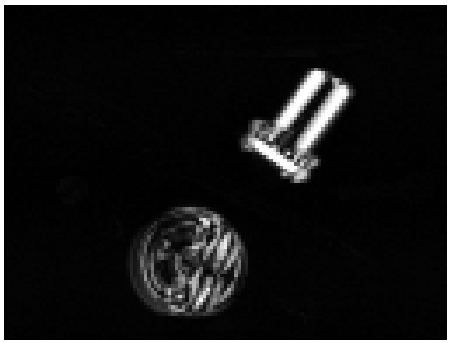
Operasi Aritmatika dan Logika

- Operasi Arithmetic/logic melibatkan dua atau lebih image yang masing-masing dikenakan operasi pixel per pixel.
- Arithmetic Operations
  - Addition, Subtraction, Multiplication, and Division
- Logic Operations
  - AND, OR, NOT

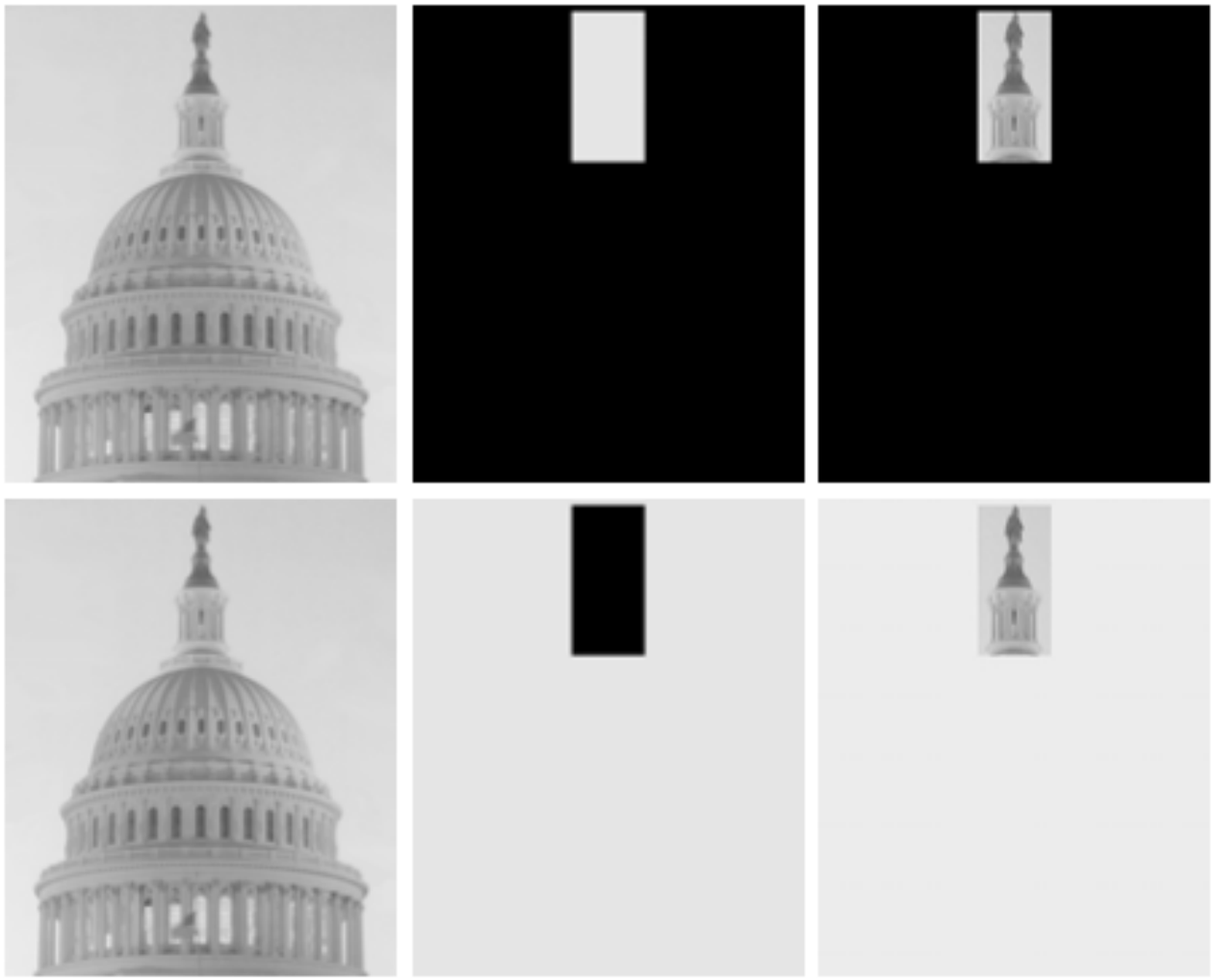
Aritmatika : Image Substraction

- Selisih/difference dari dua imageyang hampir sama untuk melihat feature tertentu dari image dengan melihat perbedaan dari dua image tersebut.
- Notasi

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



Logika: AND dan OR



a b c
d e f

FIGURE 3.27
(a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

Latihan Implementasi 1

- Buat program untuk melakukan transformasi piksel dari RGB ke Gray, dengan :
  - Lightness Method : (max(R, G, B) + min(R, G, B)) / 2
  - Average Method : (R + G + B) / 3
  - Luminosity Method : 0.21 R + 0.71 G + 0.07 B

Tugas 1: Demo Pertemuan 4

- Buat program untuk melakukan proses:
  - Form 1
    - Image Negative
    - Log Transformations
    - Power-Law Transformations
    - Piecewise-Linear Transformation Functions
  - Form 2 : Bit-plane slicing
  - Form 3 : Operasi aritmatika (Substraction)
  - Form 4 : Operasi logika
    - AND
    - OR
    - XOR

Thank You !