UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia
FAKULTAS ILMU KOMPUTER

Basic Image Processing 1 – Spatial Domain

CSCE604133 Computer Vision
Fakultas Ilmu Komputer
Universitas Indonesia

Acknowledgements

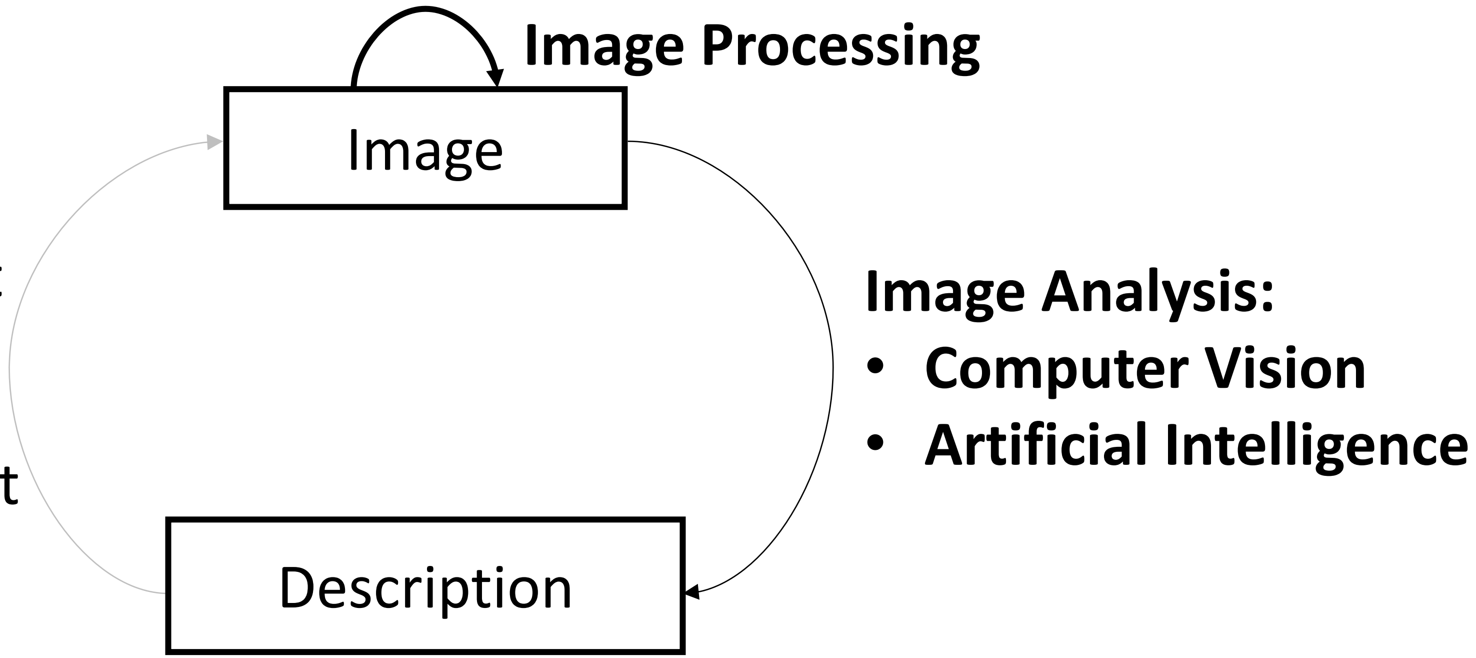
- These slides are created with reference to:
  - Computer Vision: Algorithms and Applications, 2nd ed., Richard Szeliski https://szeliski.org/Book/
  - Digital Image Processing, Gonzales and Woods, 3rd ed, 2008.
  - Course slides for CSCE604133 Image Processing – Faculty of Computer Science, Universitas Indonesia
  - Introduction to Computer Vision, Cornell Tech https://www.cs.cornell.edu/courses/cs5670/2024sp/lectures/lectures.html
  - Computer Vision, University of Washington https://courses.cs.washington.edu/courses/cse576/08sp/

Image and Description

prinsip cv dari image -> description (dari suatu gambar mau dapet informasi tertentu)

terkadang gambar belum siap untuk di deskripsiin, makanya harus di persiapin

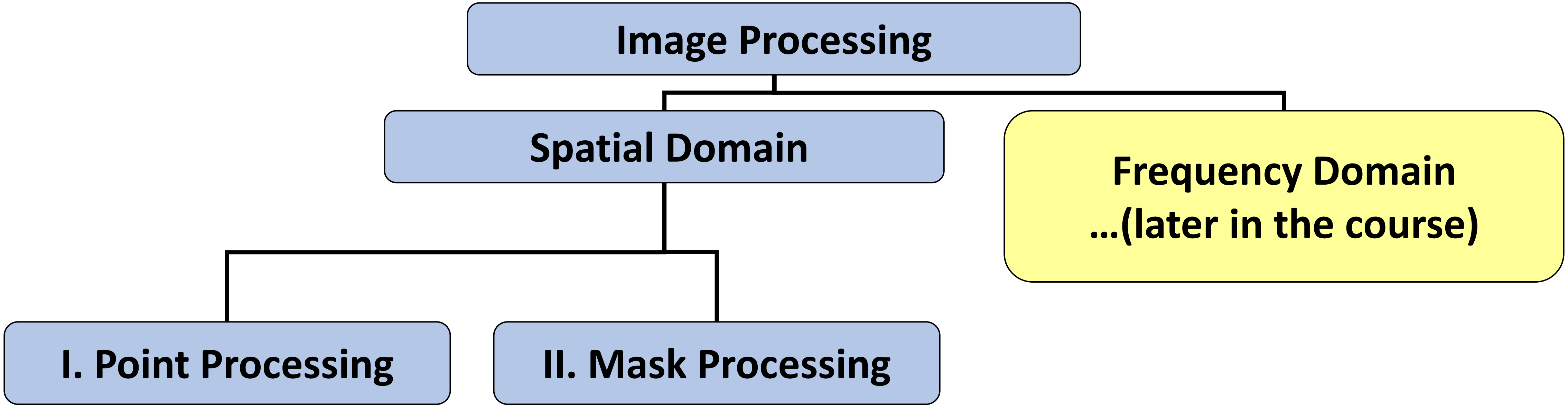
- Image processing in general is the process with images as both input and output
- The image processing is usually done to support some sort of computer vision task, e.g., Denoising an image so it can be used for OCR



FAKULTAS ILMU KOMPUTER

Basic Image Processing in the Spatial Domain

Basic Image Processing

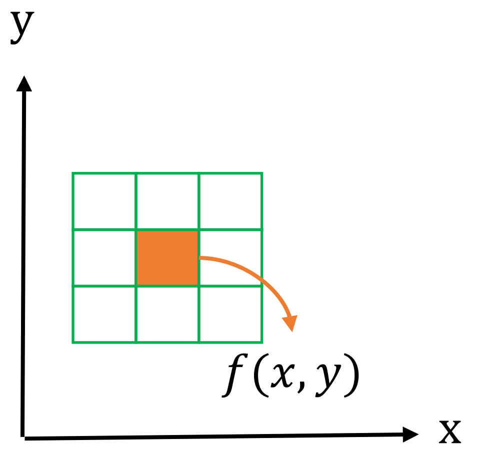


Spatial Intensity Transformation

- Spatial transformation can be denoted as T

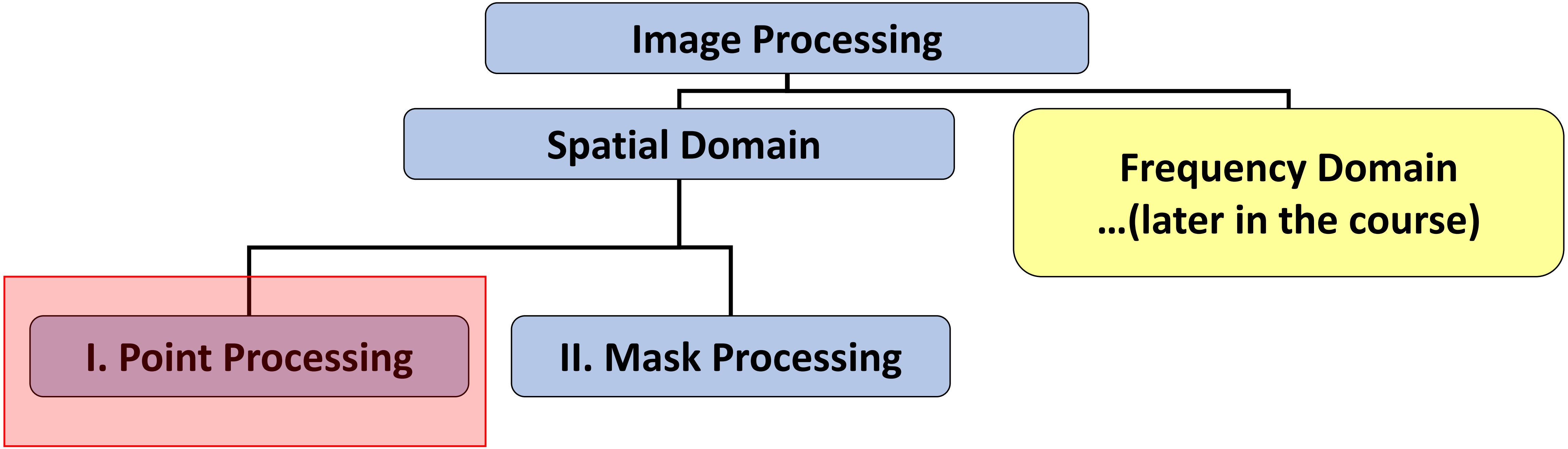
Hasil perubahan g, dipengaruhi oleh
- Nilai pixel (point processing)
- Mempertimbangkan area disekelilingnya (mask processing)

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



- The function T is applied to the neighborhood of point (x, y)
  - Can be the pixel **itself**
  - Can be an **area** surrounding the pixel
    - Usually the area is rectangular, centered on (x, y), and much smaller than the image

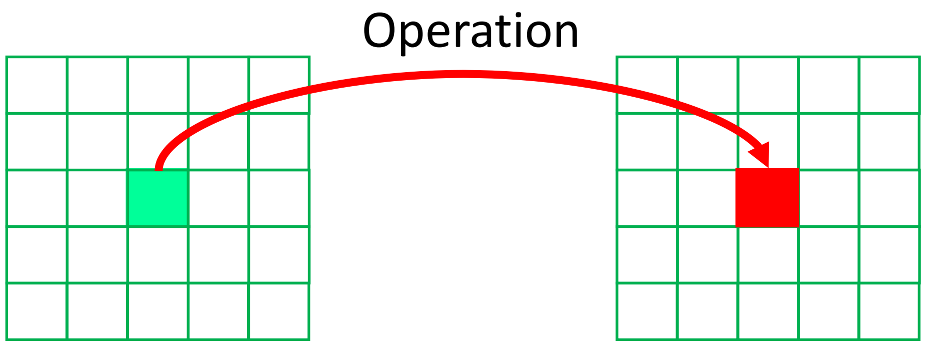
Point Processing



Point Processing

Yang tadinya tinggi dijadiin rendah, yang tadinya rendah jadi tinggi

- Intensity transformation with an operation that only involves a single pixel



- Some Basic Intensity Transformations

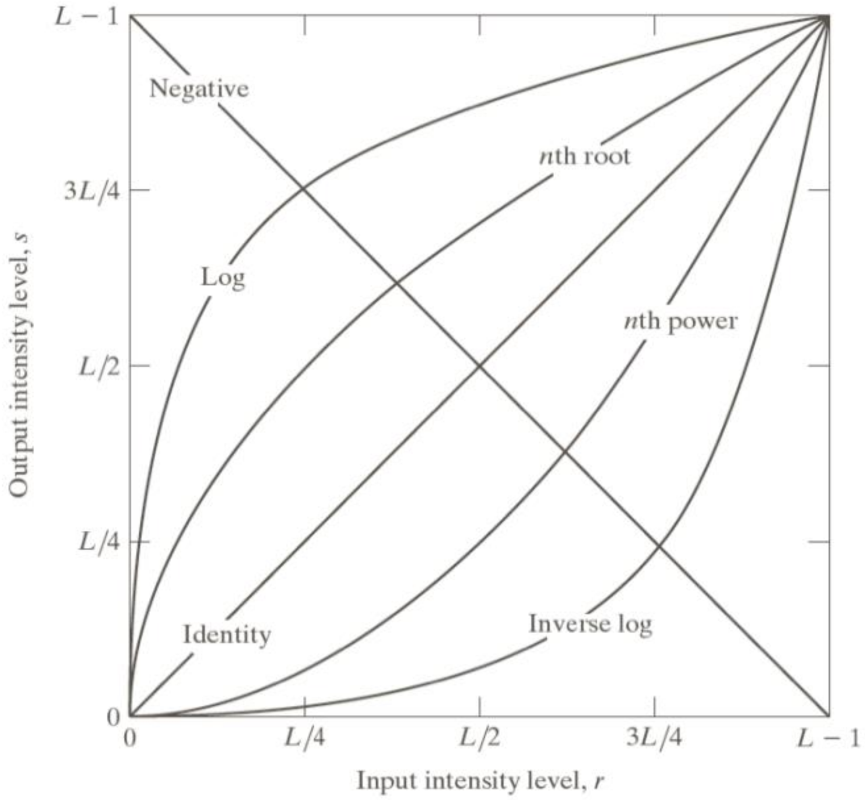


Image Negative

Yang tdinya gelap jadi terang (karena di inverse)

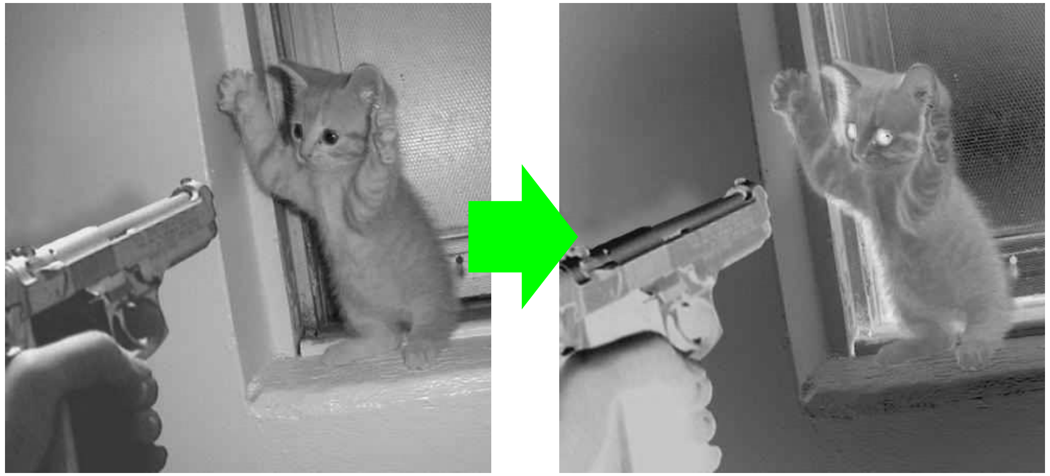
Benefits:
- Object Extraction: segmentation lebih baik di dark bacgkroud dan light foreground.
- Visualization: improve visualization
- Preprocessing for machine learning

- Changes the gray-level pixel of the input image:

$$G_{baru} = 255 - G_{lama}$$

Ada kasus dimana gambar lebih baik dijadiin negative (misalnya object extraction lebih baik black on white)

*) max value of pixel, 255 is for for 8-bit image (0-255)



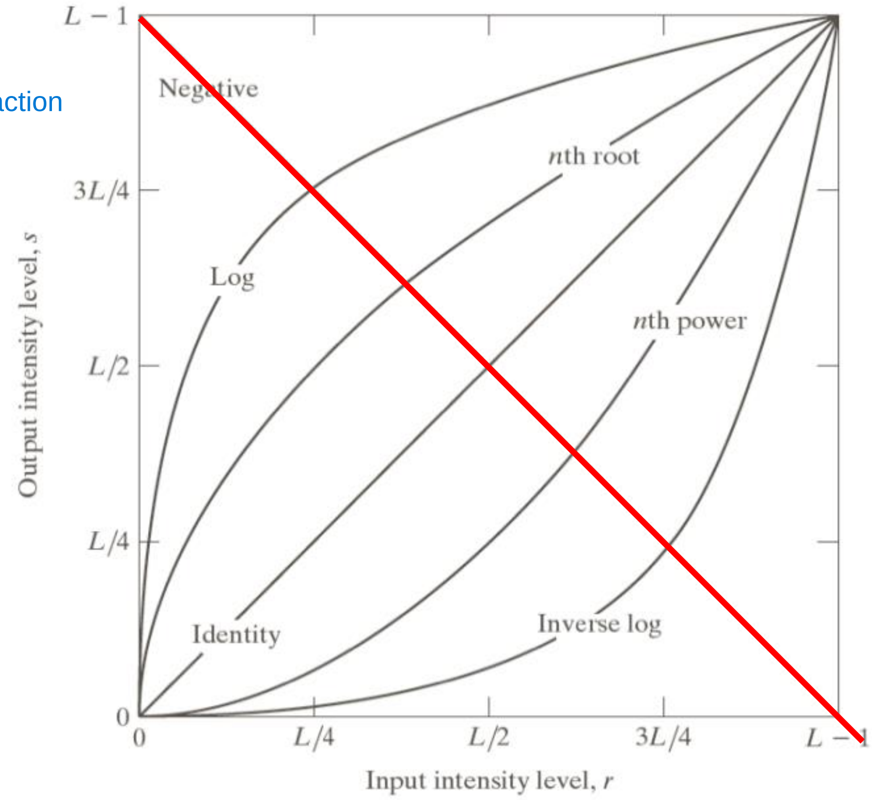


Image Negative Example

a b

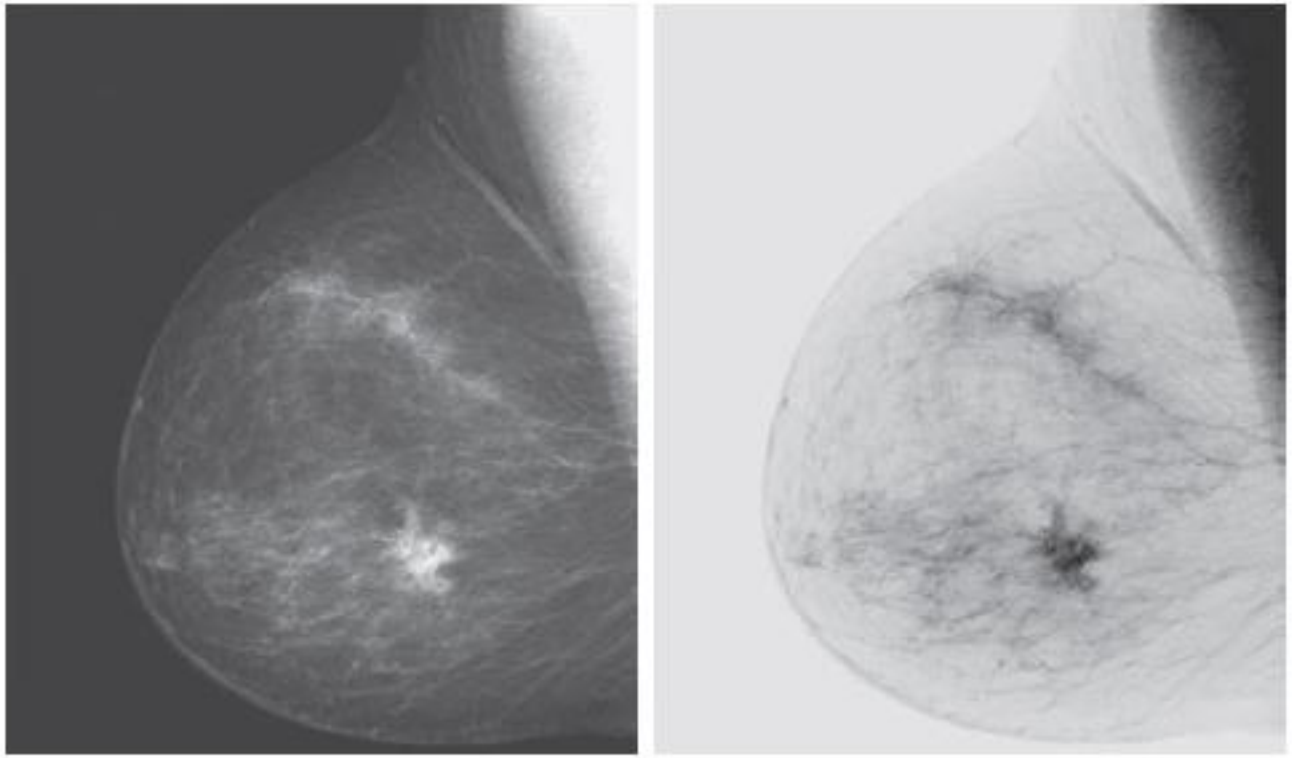


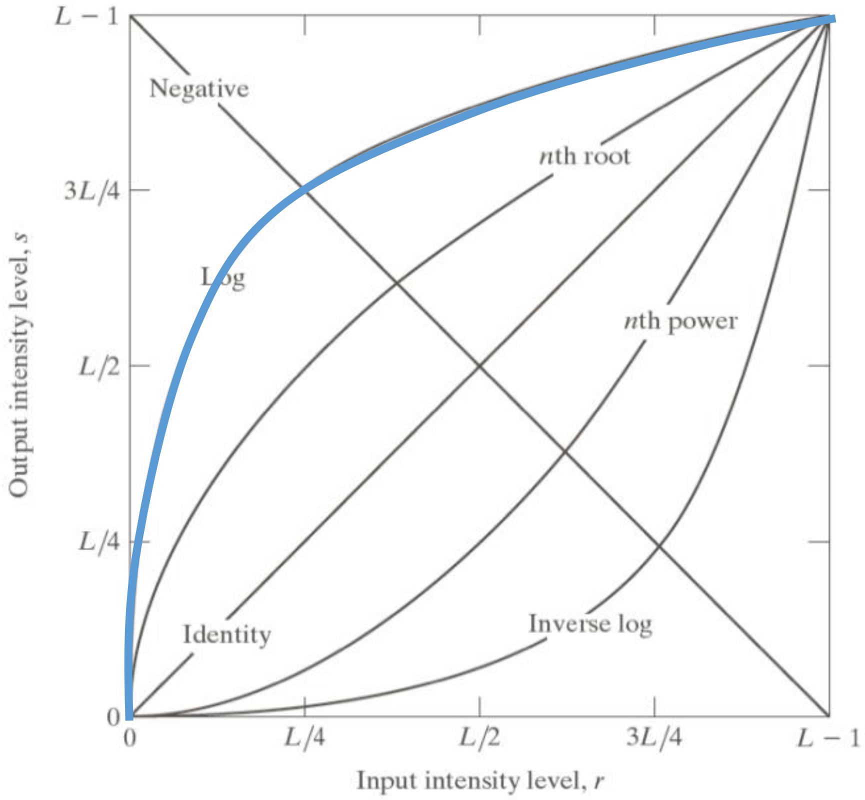
FIGURE 3.4 (a) A digital mammogram. (b) Negative image obtained using Eq. (3-3). (Image (a) Courtesy of General Electric Medical Systems.)

Log Transformation

- Log transformation

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

  - For image input r and output image s, where c is constant and $r \geq 0$
- The goal is to expand the dark pixel values and compress the light pixel values, limiting the dynamic range
  - a narrow range of low intensity values → a wider range of high intensity values
  - a wide range of high intensity values → a narrower range of low intensity values.



Log Transformation

Biasa digunain untuk citra medis

- Log transformation

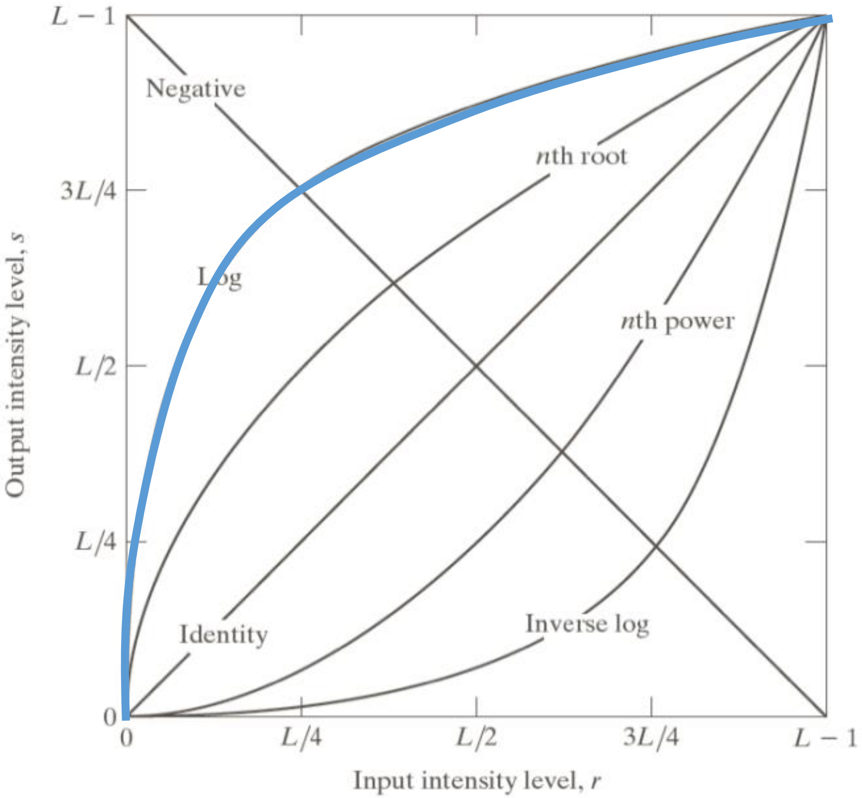
$$s = c\ \log(1 + r)$$

  - For image input r and output image s, where c is constant and $r \geq 0$

Input image

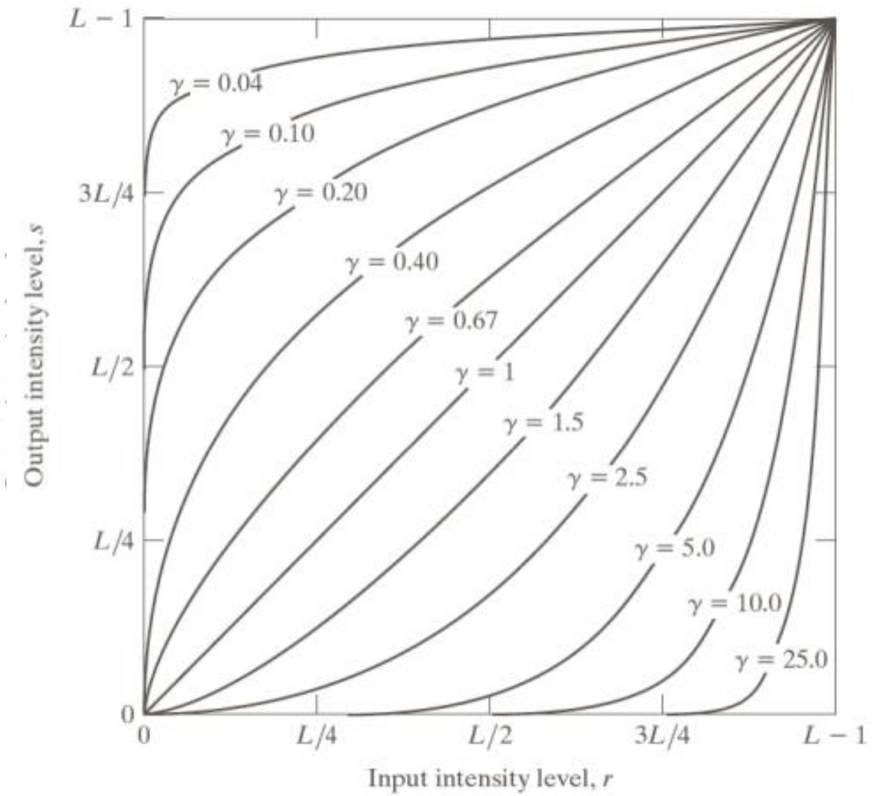
Its log transform c=1

Source: Image from Hamid Laga, Telecom Lille 1



Gamma (Power) Transformations

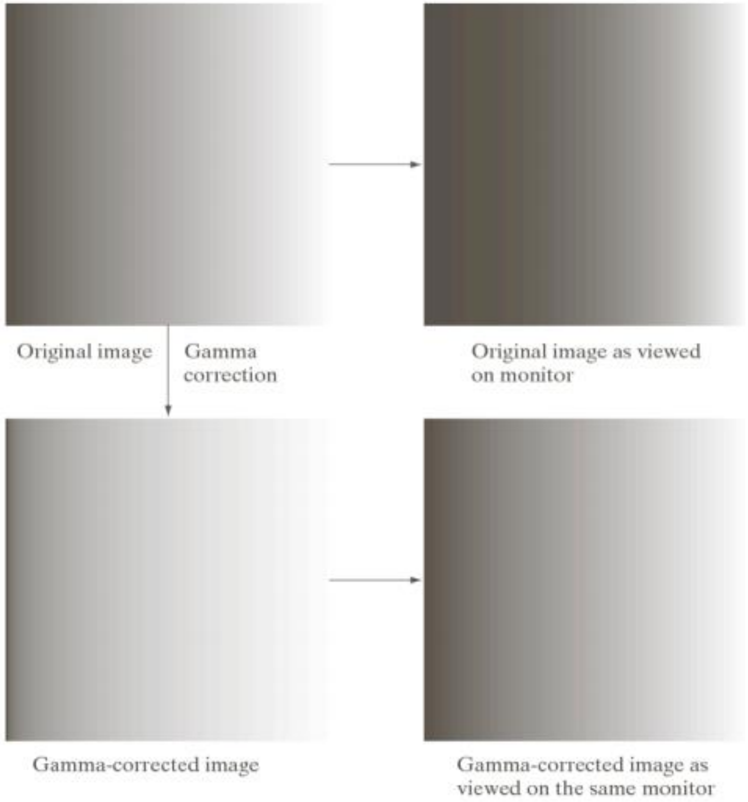
- General equation $$s = cr^{\gamma}$$
  - For input r ,output s, with c and γ constant
- Similar to log transform
  - Possible transformations just by varying γ
  - $\gamma < 1$
  - $\gamma > 1$
  - $\gamma = 1$



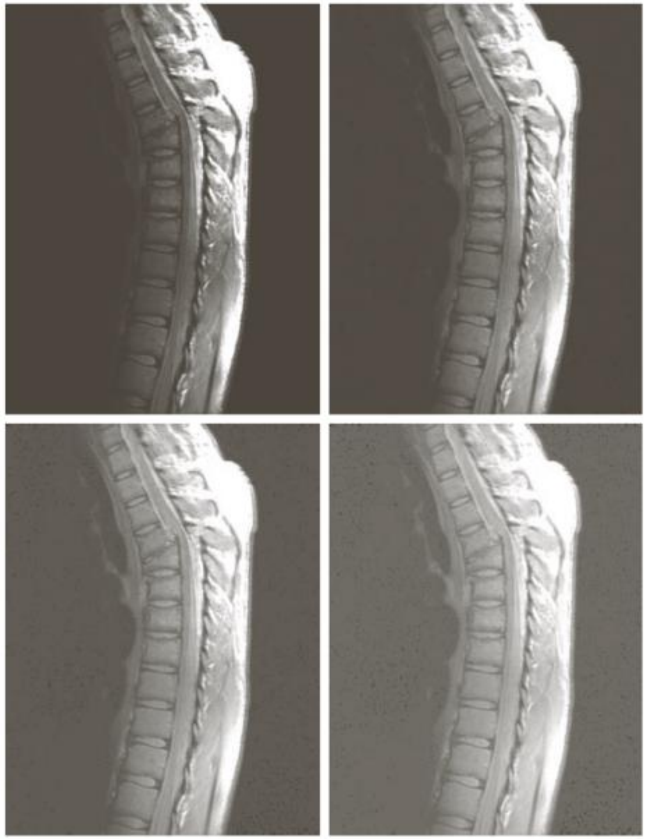
Gamma Transformation Applications

Images from Department of Radiology and Radiological Sciences, Vanderbilt University

- Gamma correction for displays



- Improving contrast



Contrast Stretching

- Expanding the pixel intensity range that was limited before to a full-range
- Input image with a limited grey level range 0 – $\max(f_{in})$ (low contrast) → full range intensity image 0 – 255 (high contrast)

$$f_{out} = ((f_{in} - a) * b$$

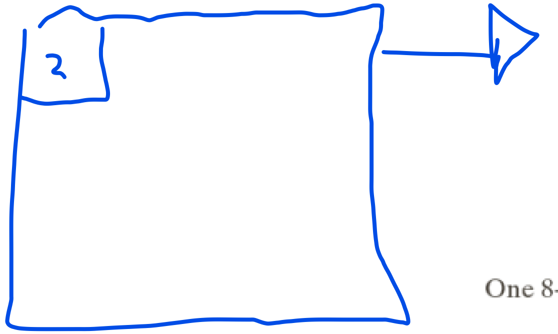
  - where $a = \min((f_{in}))$, and $b = 255/(\max((f_{in}) - \min((f_{in}))$

Biasanya contrast stretching hasilnya gak natural

Original image with a limited intensity range (ex: 0-100 on an 8-bit image)

Image after *contrast stretching* with a full intensity range (ex: 0-255 on 8 bit)

Bit-Plane Slicing



- Each pixel consists of 8 bits (for 8-bit images)
- Instead of considering the intensity *values*, we can consider the contribution of each bit

Perlu kalo misalnya perlu kompresi data

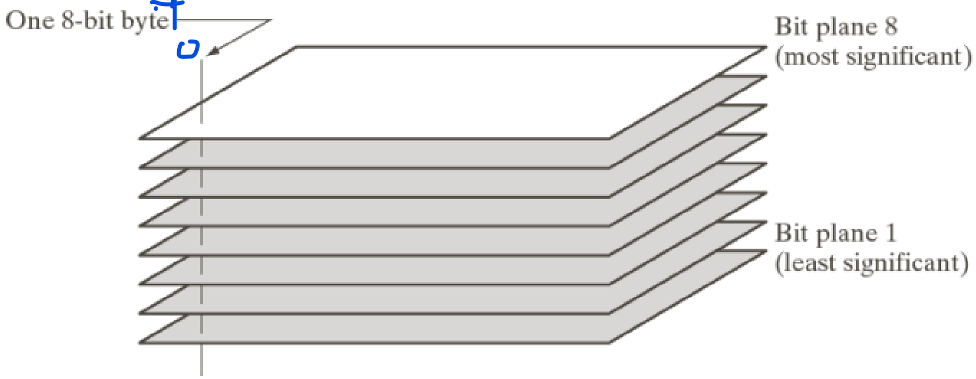


Image Histograms

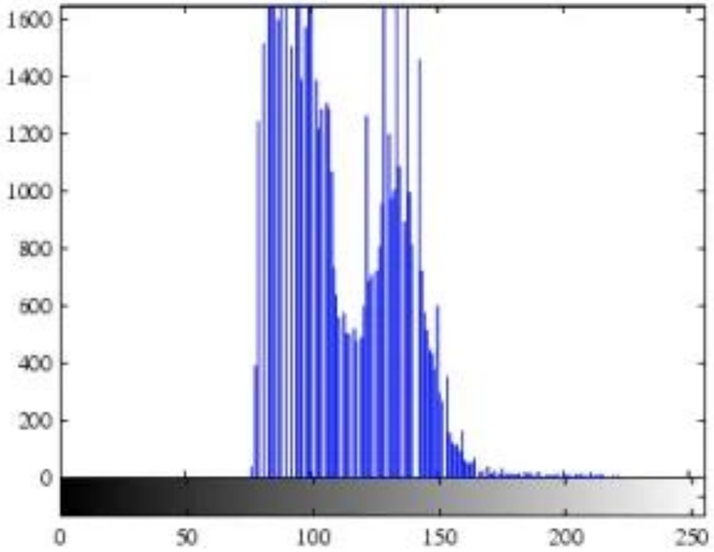
Histogram bisa digunakan untuk merepresentasikan intensity dari gambar

Kekurangan:
- Kehilangan informasi mengenai letak (gak tau pixelnya ada dimana)

- For an image with intensity levels $[0, L-1]$, its histogram is

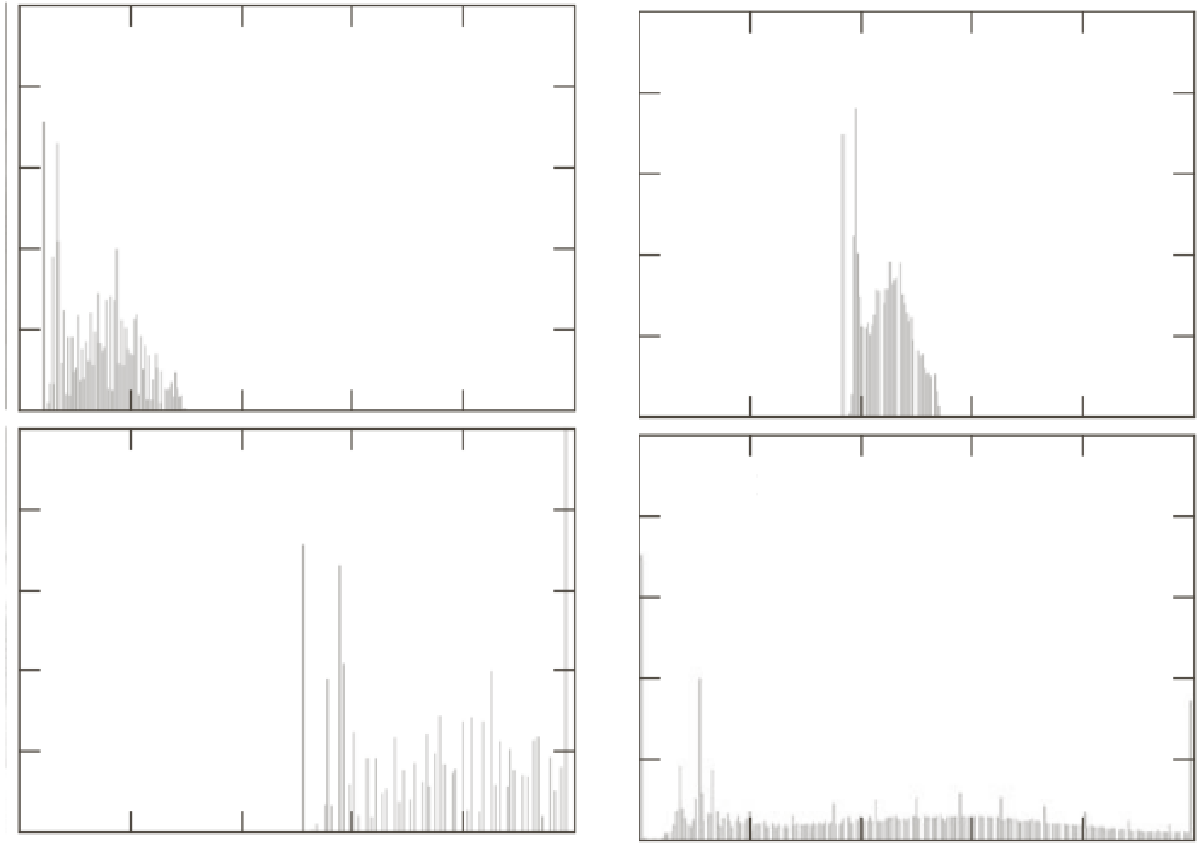
$$h(r_k) = n_k$$

  - Where r_k is the k-th intensity value and n_k is the number of pixels with intensity r_k
- The histogram plots the distribution of intensity values throughout the image



Learning from an Image’s Histogram

- What can we deduce about the image from their histograms?



Dark image

Low contrast

Bright image

Contrast tinggi makin keliatan perbedaan warna

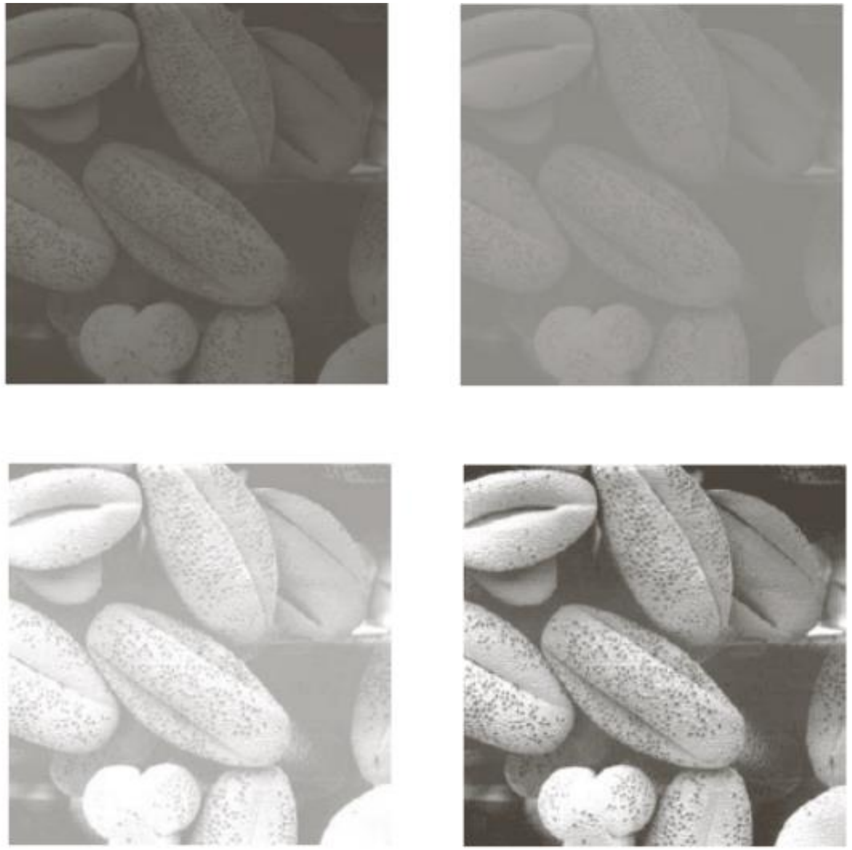
High contrast

Learning from an Image’s Histogram (2)

Sebelum melakukan preprocessing:

- Biasanya melakukan data mengenai image dalam sisi histogram untuk menentukan gelap atau terang.

- What can we deduce about the image from their histograms?



Dark image

Low contrast

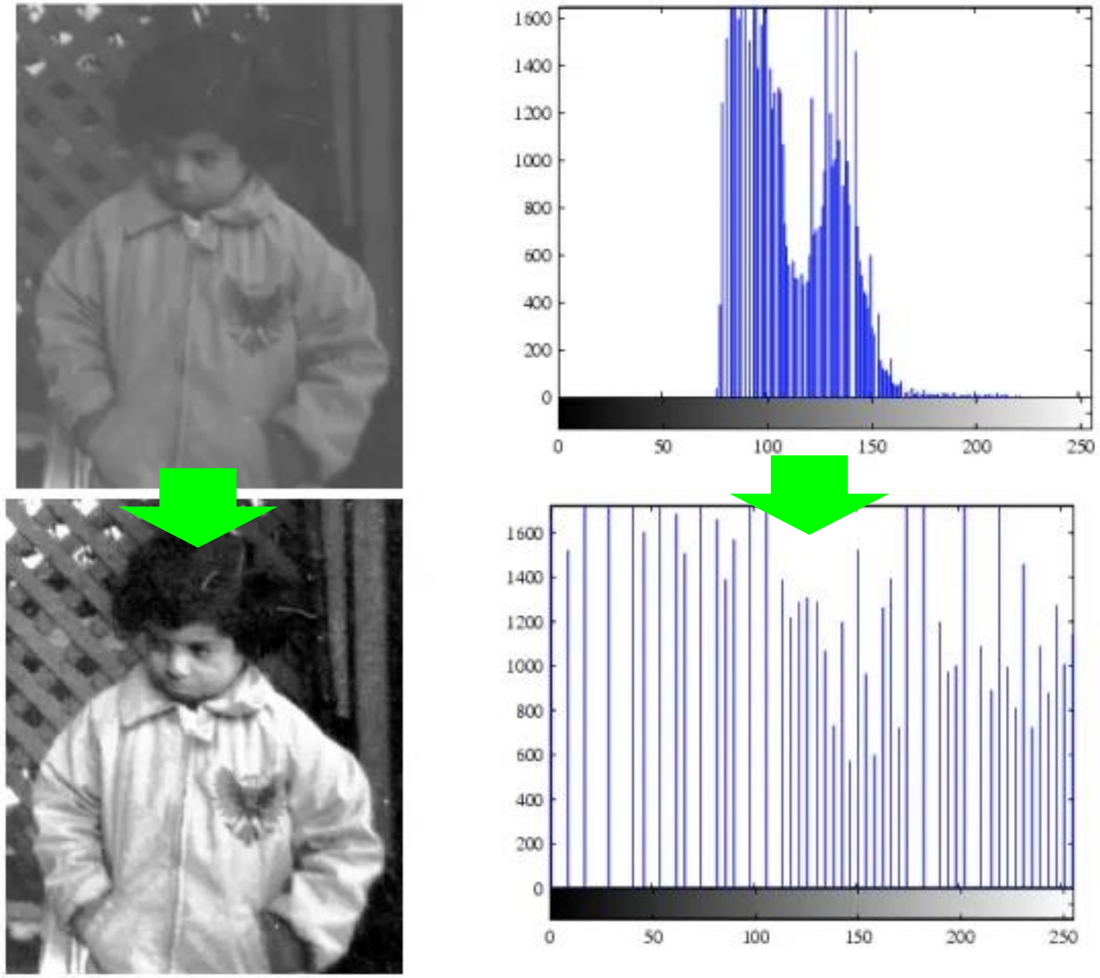
Bright image

High contrast

Histogram Equalization

ini di strech out

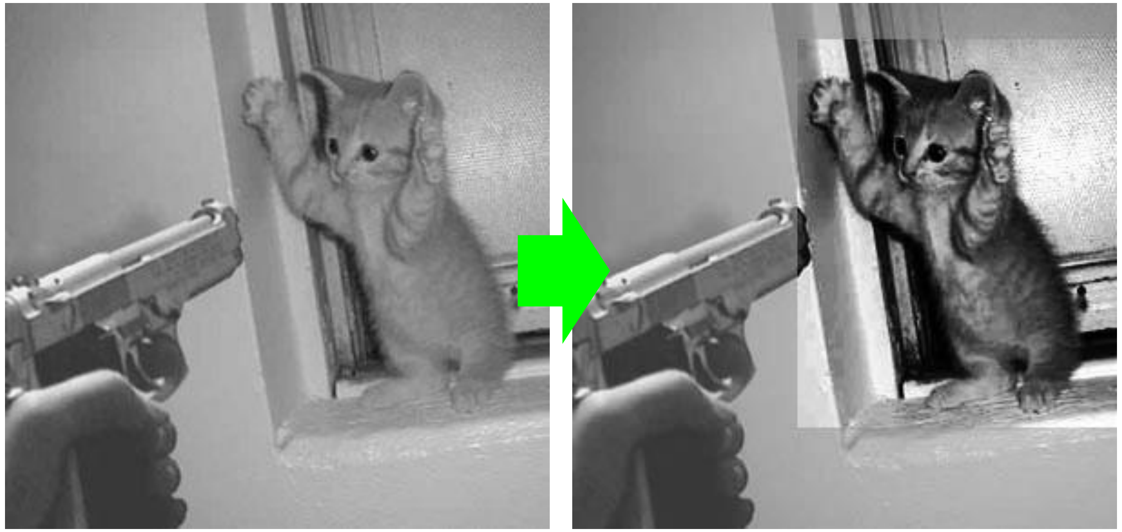
- Change the gray level mapping so the contrast is better distributed on the entire range (0-255 for 8-bit image)
  - Grey levels that often occur to be spread out from the gray levels near it
  - Grey levels that rarely occur to be compressed nearer to the gray levels near it
  - Ensure the new histogram reaches the highest possible value (ex: 255 for 8-bit images)



Histogram Equalization (2)

- Global HE

- Local HE

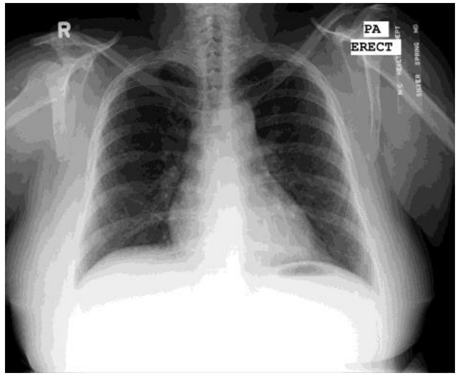


Extracting Areas of an Image

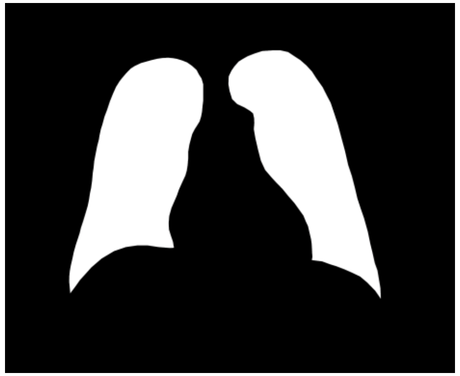
Menggunakan data sebelumnya untuk menentukan background, misalnya kalo CCTV, kita bisa analisis background dimana gak ada orang, terus disubstract backgroundnya.

- Image Subtraction

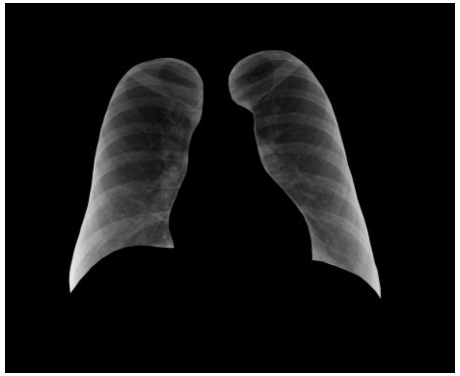
- Image Masking



X-Ray (A)



Mask (B)

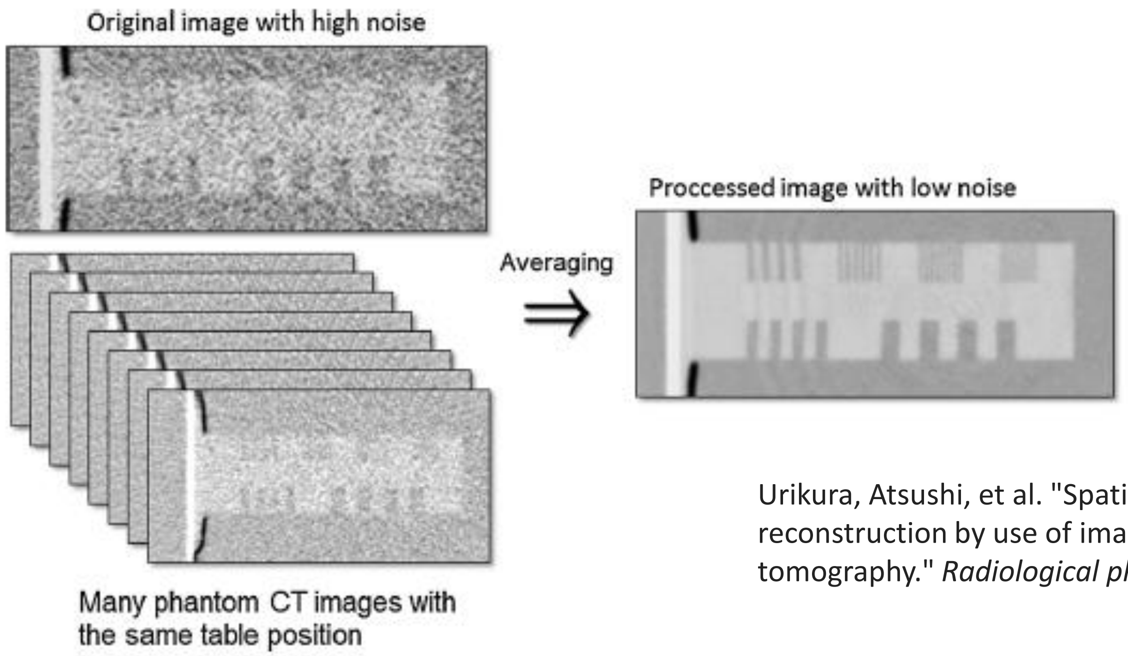


$A\ AND$ B

Image Averaging

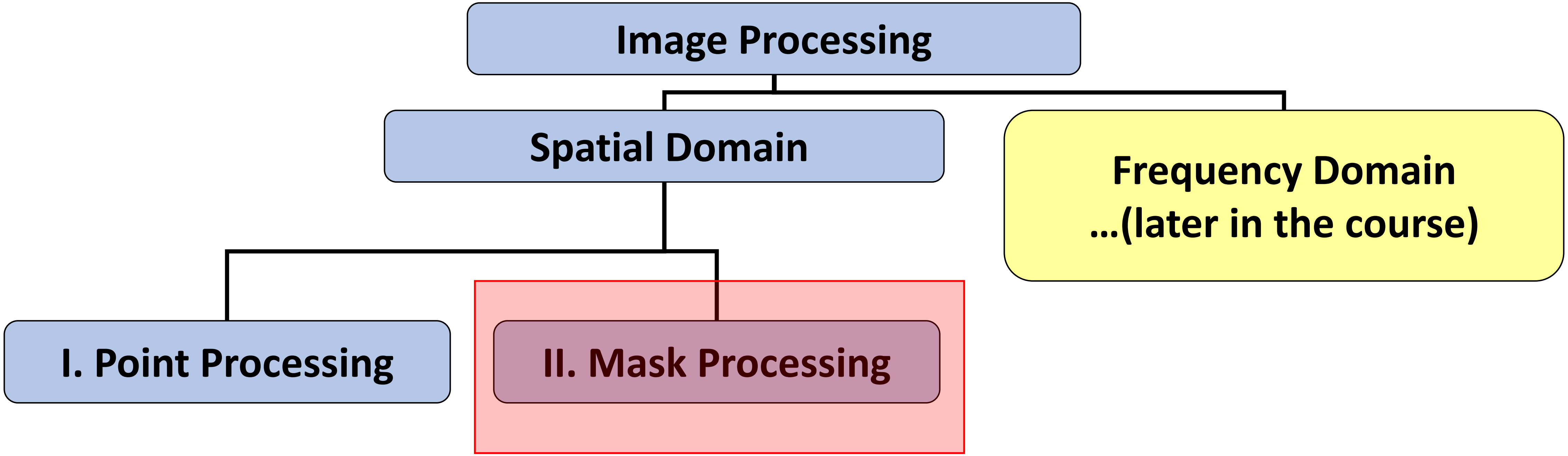
- Done when we have some images with the same image, but they all have noise
- The noise of each image is different and uncorrelated
- Noise can be removed by averaging all images

Remove noise kayak outlier gitu biar dikecilin



Urikura, Atsushi, et al. "Spatial resolution measurement for iterative reconstruction by use of image-averaging techniques in computed tomography." *Radiological physics and technology* 7 (2014): 358-366.

Mask Processing



Mask Processing

- Mask processing processes the surrounding pixels as a window
- We then apply convolution using a mask on the whole window
- Masks are also called filters or kernels.
- Example:
  - 3x3 sized pixel window,
  - The transformation of pixel x will be affected by all 8 neighbors
  - As opposed to point processing : in point processing, a pixel's value does not depend on the neighbors

X akan terpengaruhi dengan neighbord ida yang lain

1	2	3
8	x	4
7	6	5

Mask Processing (2)

- If this 3x3 mask was used for convolution on the image, then for every 3x3 window

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

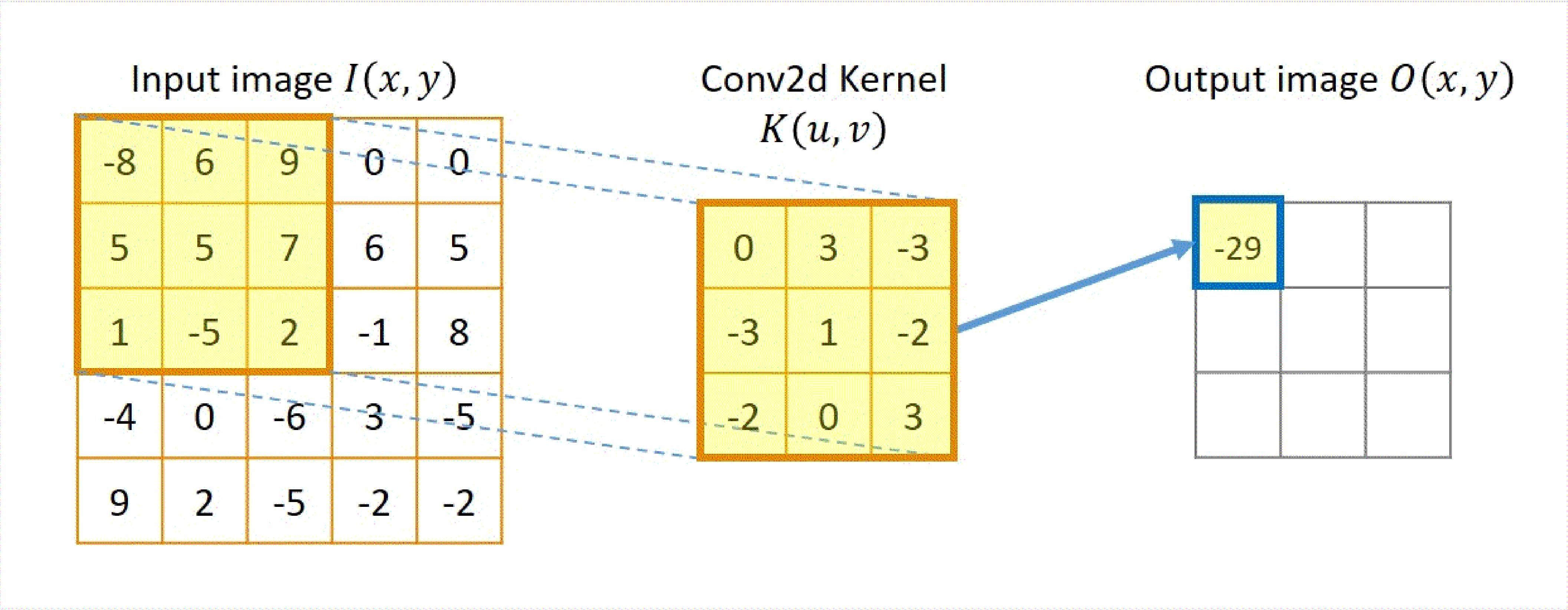
G_{11}	G_{12}	G_{13}	G_{14}	G_{15}
G_{21}	G_{22}	G_{23}	G_{24}	G_{25}
G_{31}	G_{32}	G_{33}	G_{34}	G_{35}
G_{41}	G_{42}	G_{43}	G_{44}	G_{45}
G_{51}	G_{52}	G_{53}	G_{54}	G_{55}

$$G22' = w1\,G11 + w2\,G12 + w3\,G13 + w4\,G21 + w5\,G22 + w6\,G23 + w7\,G31 + w8\,G32 + w9\,G33$$

BisaPadding dengan teknik seperti zero Padding jadinya Kita bisa cari G11

Tanpa padding, kita gak bisa cari G11

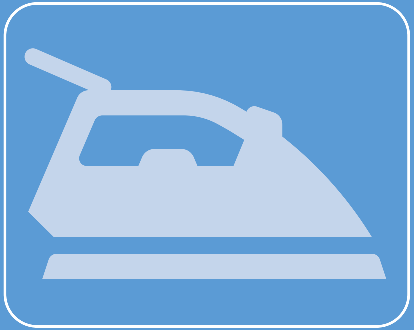
Mask Processing (3)



https://coolgpu.github.io/coolgpu_blog/assets/images/Conv2d_0p_1s_1inCh.gif

Spatial Filters

yang banyak noise, itu di smooth noisenya



Smoothing filters:

- Lowpass filter (linear, takes the average value in the window)
- Median filter (non-linear filter, takes the median in every window)

Sharpening filters:

- Roberts, Prewitt, Sobel (edge detection)
- Highpass filter

Smoothing

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

- Average lowpass filter

$$\frac{1}{n} \times [A]$$

  - Where A is a matrix with n elements of 1s

(a) Original

(b)-(f) results of spatial lowpass filtering with a mask size of 3,5,9,15,35

Mean Filters

Arithmetic Mean Filter

- Smoothing and noise reduction
- Loss of detail

important

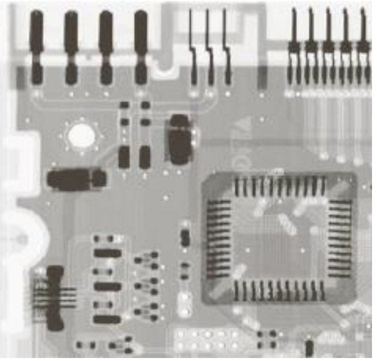
$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{x,y}} g(s,t)$$

- $m \times n$ image
- $S_{x,y}$ kernel centered at (x,y)

Geometric Mean Filter

- Smoothing and noise reduction
- Less loss of detail compared to arithmetic mean filter

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{x,y}} g(s,t) \right]^2$$



Clear Image

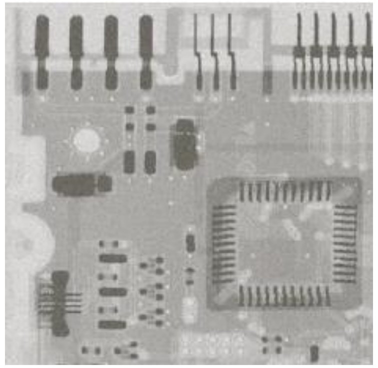
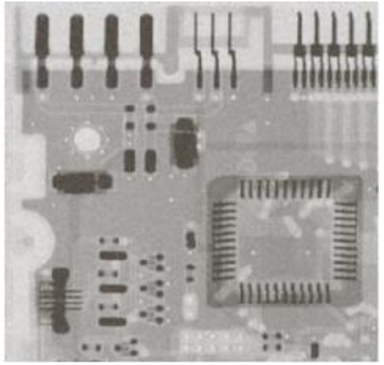
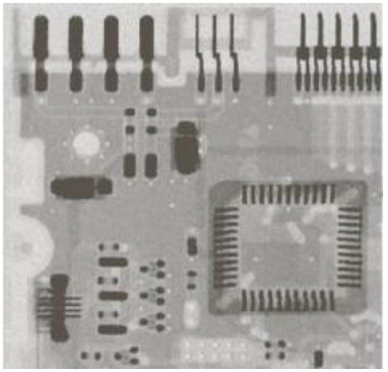


Image w/ Gaussian Noise

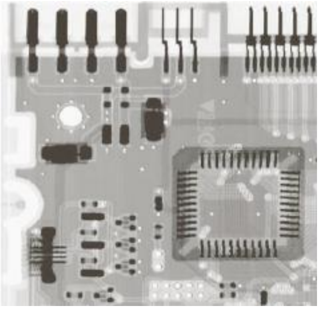


Restored Image with Arithmetic Mean Filter



Restored Image with Geometric Mean Filter

Order-Statistic Filters


Clear Image

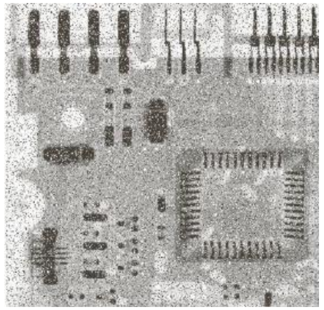
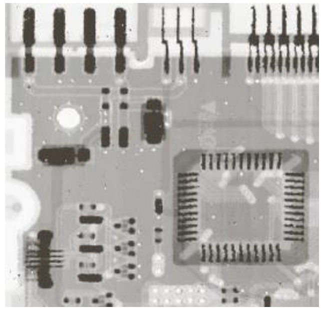
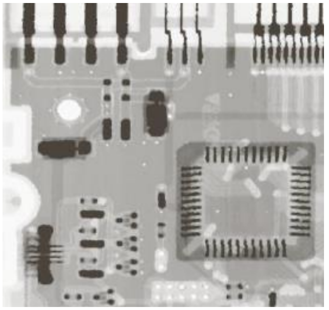
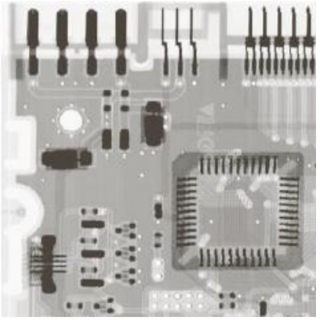


Image w/ Salt and Pepper Noise


Restored with 1 pass of Median Filter


Restored with 3 passes of Median Filter


Clear Image

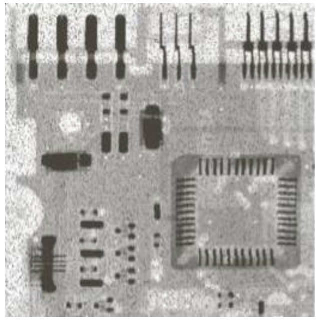
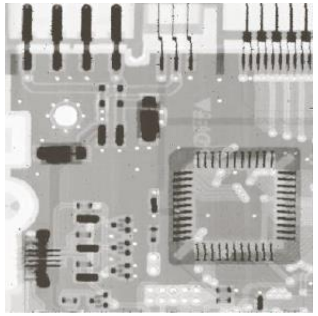
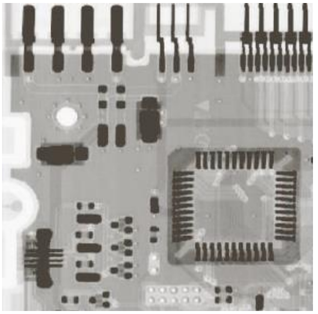
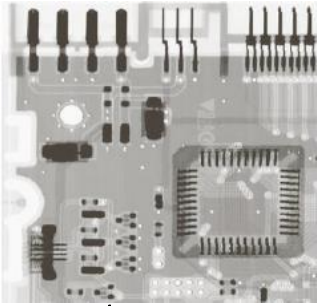


Image w/ Pepper Noise


Restored with Max Filter


Restored with Min Filter


Clear Image

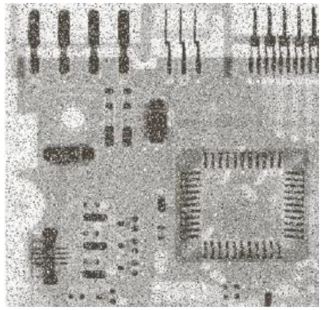
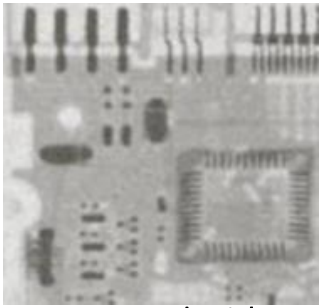


Image w/ Salt and Pepper Noise


Restored with Alpha-Trimmed Filter

Median Filters

$$\hat{f}(x,y) = \underset{(s,t)\in S_{x,y}}{\text{median}}\{g(s,t)\}$$

Alpha Trimmed Mean Filter

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{x,y}} g(s,t)$$

Min / Max Filters

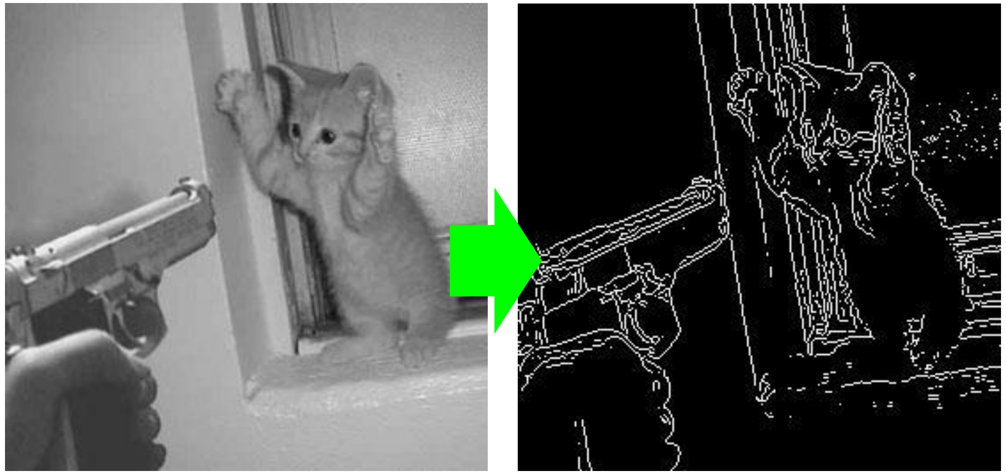
$$\hat{f}(x,y) = \underset{(s,t)\in S_{x,y}}{\text{min/max}}\{g(s,t)\}$$

- $m \times n$ image
- $S_{x,y}$ kernel centered at (x,y)
- d : between 0 and $mn-1$

Edge Detection

Menajamkan perbedaan pada intensitas

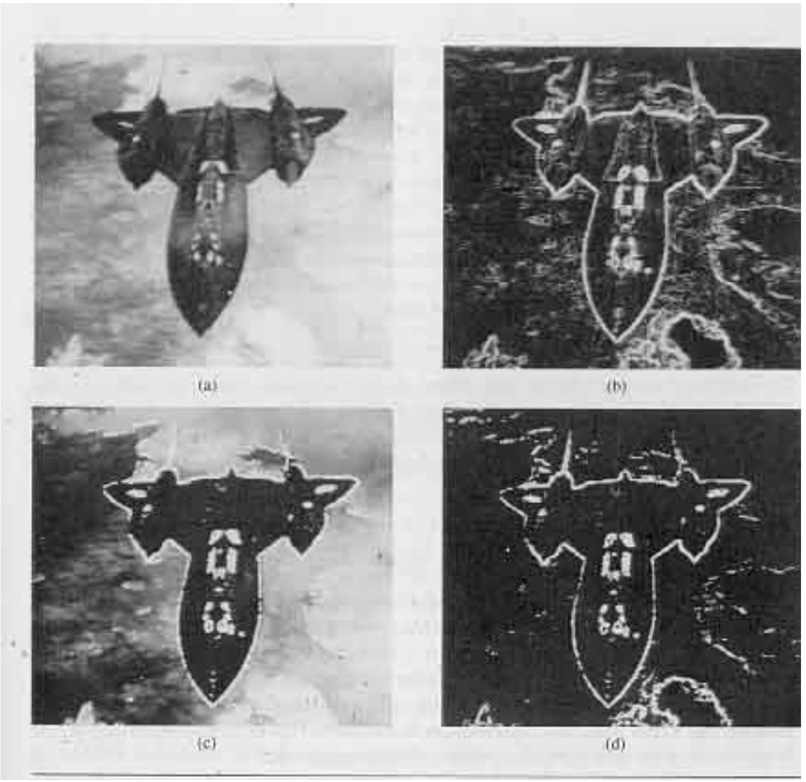
- An edge (limit of an object) is denoted by a **large difference** in intensity
- How to detect large difference in intensity?
  - Make the difference more apparent!
  - Thresholding to distinguish which differences are edges



Buat nangkep perbedaan horizontal

Sobel and Prewitt Mask

dikali positive/negative agar dia bisa
max perbedaan yanga da.



di kali negative semua (turun)

-1	-2	-1
0	0	0
1	2	1

dikali positive semua (naik)

-1	0	1
-2	0	2
-1	0	1

buat nangkep perbedaan
vertical

Sobel

-1	-1	-1
0	0	0
1	1	1

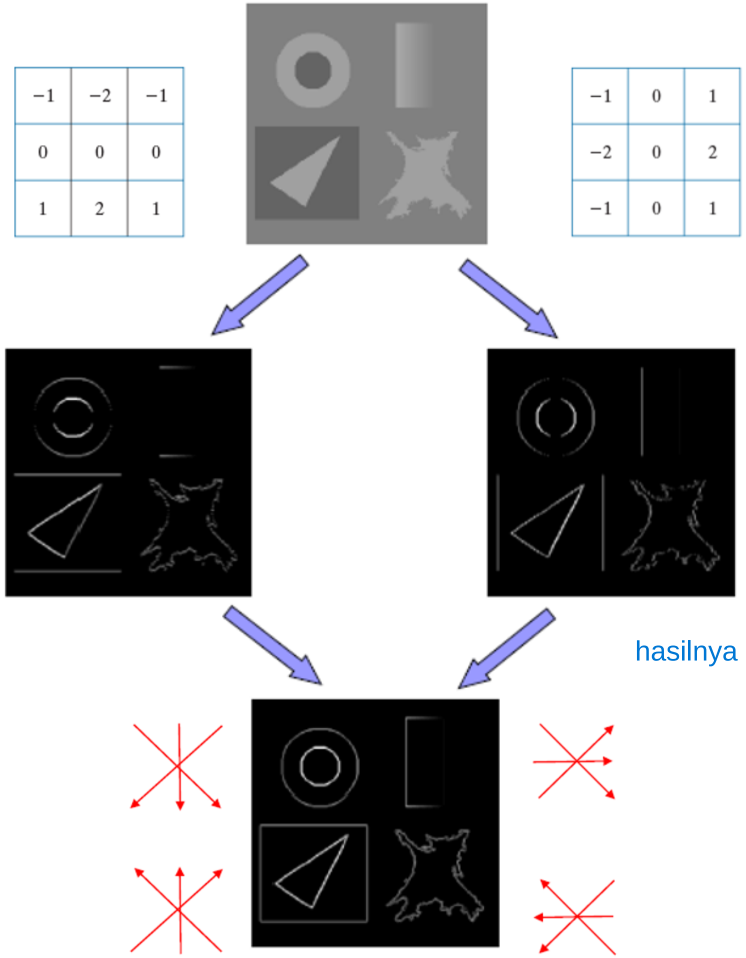
-1	0	1
-1	0	1
-1	0	1

Prewitt

(a) Original image, (b) result of filtering with Prewitt mask, (c) thresholding of (b) for values > 25 (white)
(d) thresholding of (b) pfor values >25 (white) and < 25 (black)

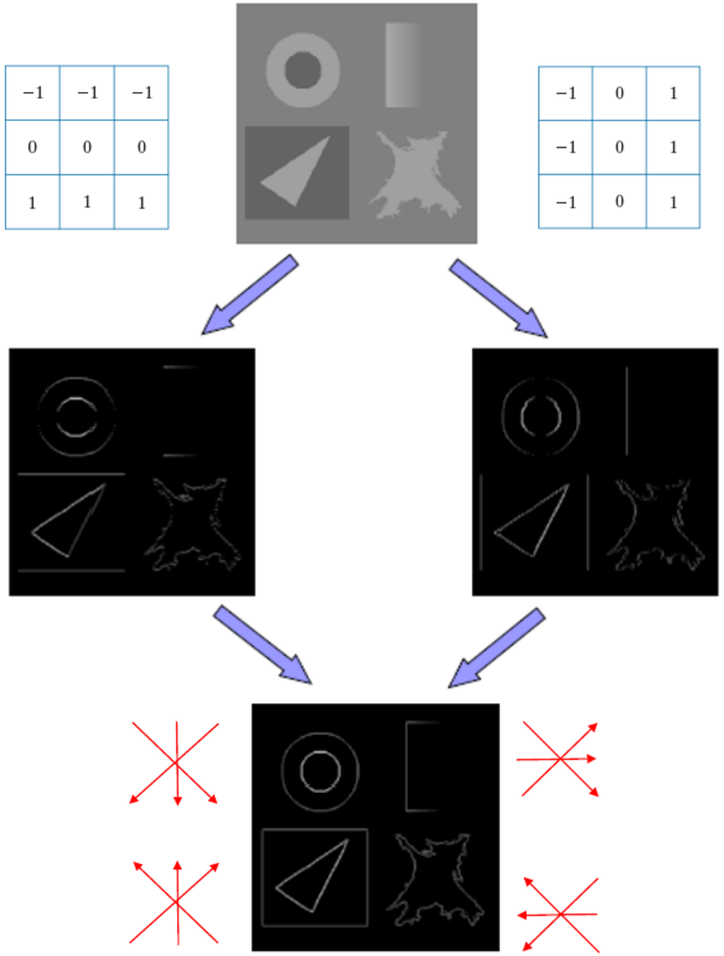
Sobel and Prewitt Mask for Edge Detection

Harus 2-2nya yang lakuin perkalian negative



$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

hasilnya di merge



$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Try this!

- [Image Kernels explained visually (setosa.io)]

Image Gradients

Perubahan intensitas = gradien

Kalau ada perubahan, maka bisa dicari gradien nya

- Brightness gradient of image $f(x, y)$

$$\Delta f = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right)$$

- Digital derivative:

$$\Delta x = f(x+n, y) - f(x, y)$$

$$\Delta y = f(x, y+n) - f(x, y)$$

commonly $n = 1$.

- Magnitude of Gradient Vector
  - How different areas are

$$\| \nabla f \| = \sqrt{\Delta x^2 + \Delta y^2}$$

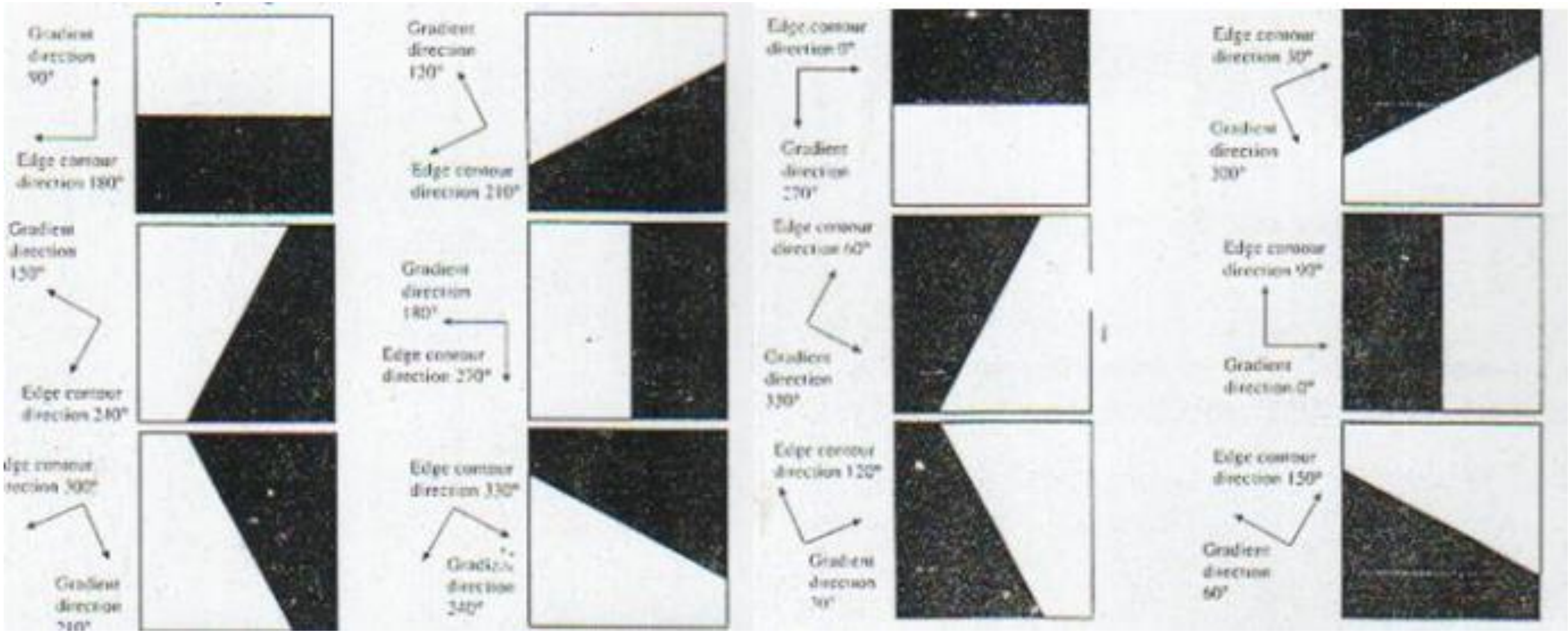
$$\| \nabla f \| = \max(abs(\Delta x), abs(\Delta y))$$

$$\| \nabla f \| = abs(\Delta x) + abs(\Delta y)$$

- Direction of Gradient Vector
  - Shows the edge contour direction

$$\phi = \tan^{-1} \mathit{atau} \arctan \frac{\Delta y f(i,j)}{\Delta x f(i,j)}$$

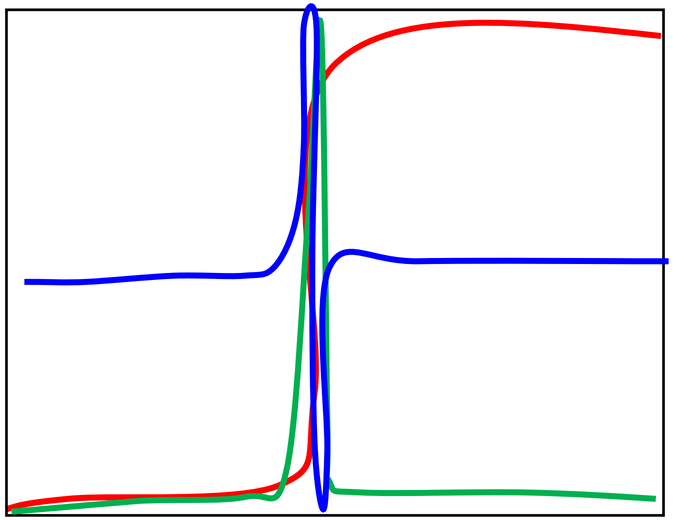
Direction of Gradient Vector (2)



Zero - Crossing

Turunan pertama, itu menandakan ada perubahaan intensitas

Turunan kedua: menunjukkan perubahan warna (misal dari terang jadi gelap)



1D image : change of intensity

1st derivative: a peak at the point of change of intensity

2nd derivative: zero-crossing

Laplacian of Gaussian Filtering

- Laplacian operator (HPF):

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

- Laplacian itself attempts to enhance the quality of detail in the image
- Laplacian of Gaussian filtering attempts to remove noise *and* enhance the quality of details.

- Laplacian of Gaussian:

$$\nabla^2 G_\sigma(x, y, \sigma) = \nabla^2 G_\sigma * F(x, y)$$

$$\nabla^2 G_\sigma(x, y, \sigma) = \left(\frac{r^2 - 2\sigma^2}{\sigma^4}\right) \exp\left(\frac{-r^2}{2\sigma^2}\right)$$

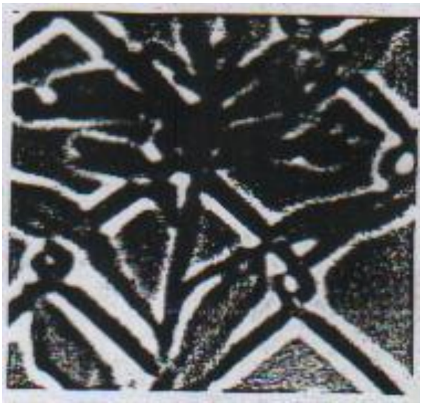
with $r = \sqrt{x^2 + y^2}$

Laplacian of Gaussian Filtering (2)

Original image (320 x 320)

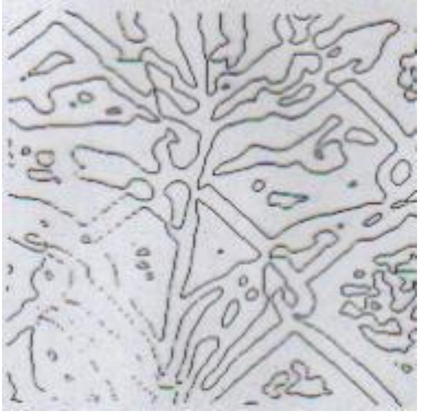
Gaussian filtering $\sigma = 8$

Gaussian filtering $\sigma = 4$



Laplacian of Gaussan

Positive (white) Negative (black)



Zero-Crossings

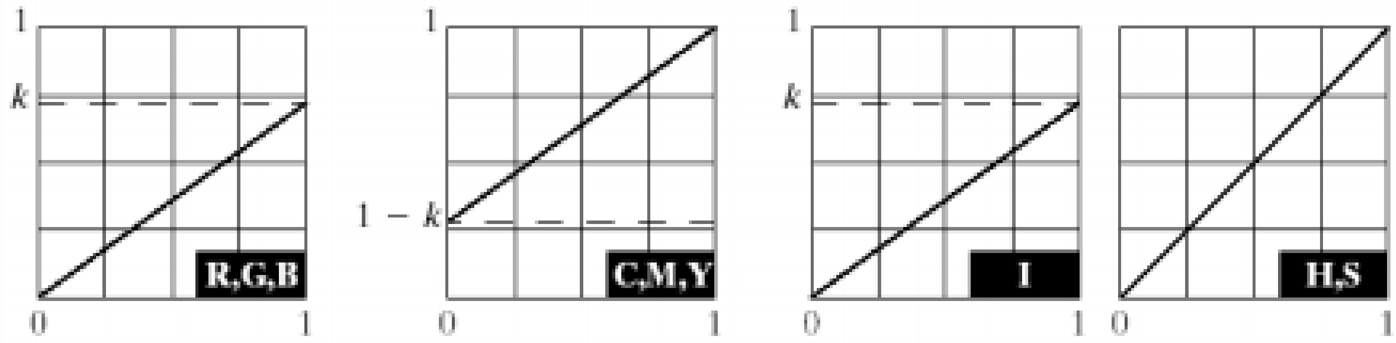
Color Image Processing

Color Transformation

- Within 1 color model (ex: only in RGB, not between color models)

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

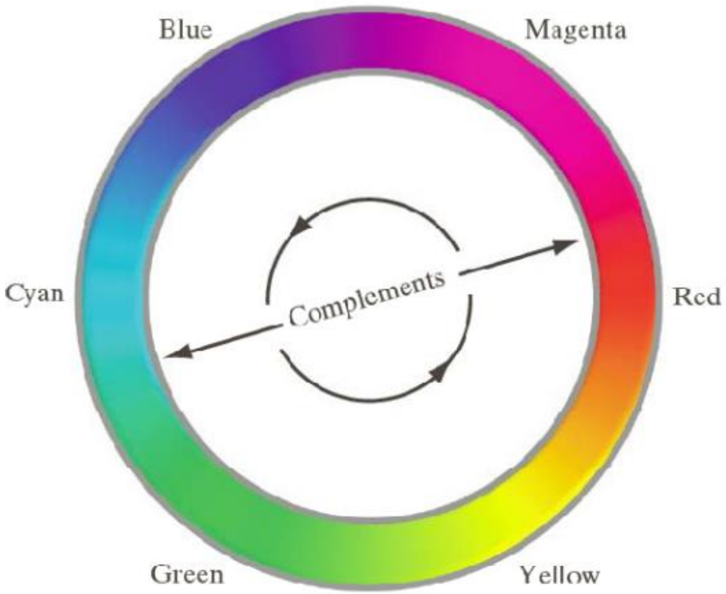
  - Depends on color model
- For example: $g(x,y) = k.f(x,y) \;\; for\; k = 0.7$

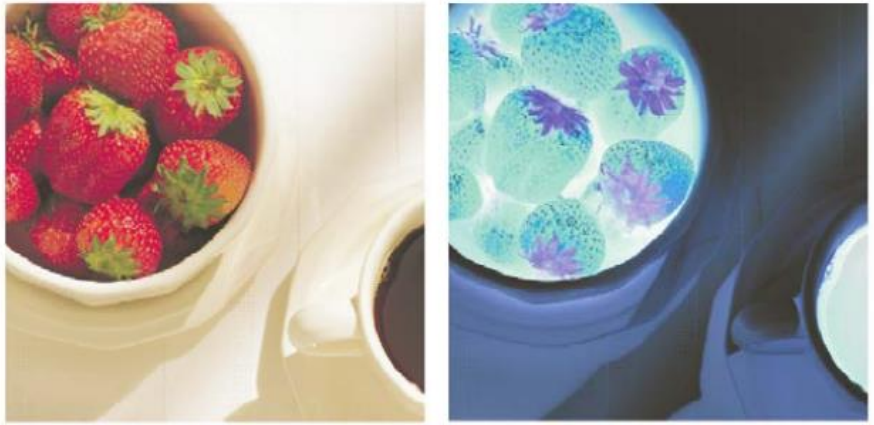


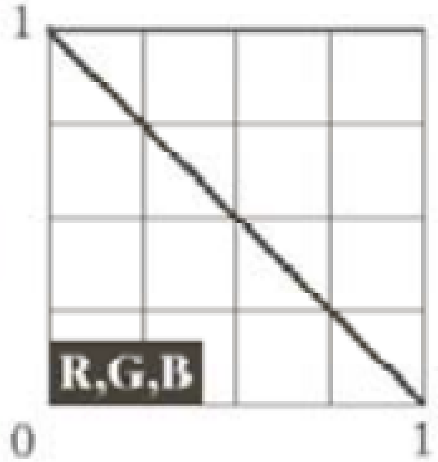
Color Complements

beda sama negative

- The hues directly opposite of one another on Newton's color circle







karena membantu membedakan objek

Color Slicing

- Attempts to highlight a specific range of colors.
  - Select colors of interest so that they stand out, neutralize other colors
- Set a color of interest $(a_1, a_2, a_3 .. a_n)$ and allow a threshold of colors near it to remain, while cancelling out the rest

$a = (0.6863, 0.1608, 0.1922)$ to detect edible objects

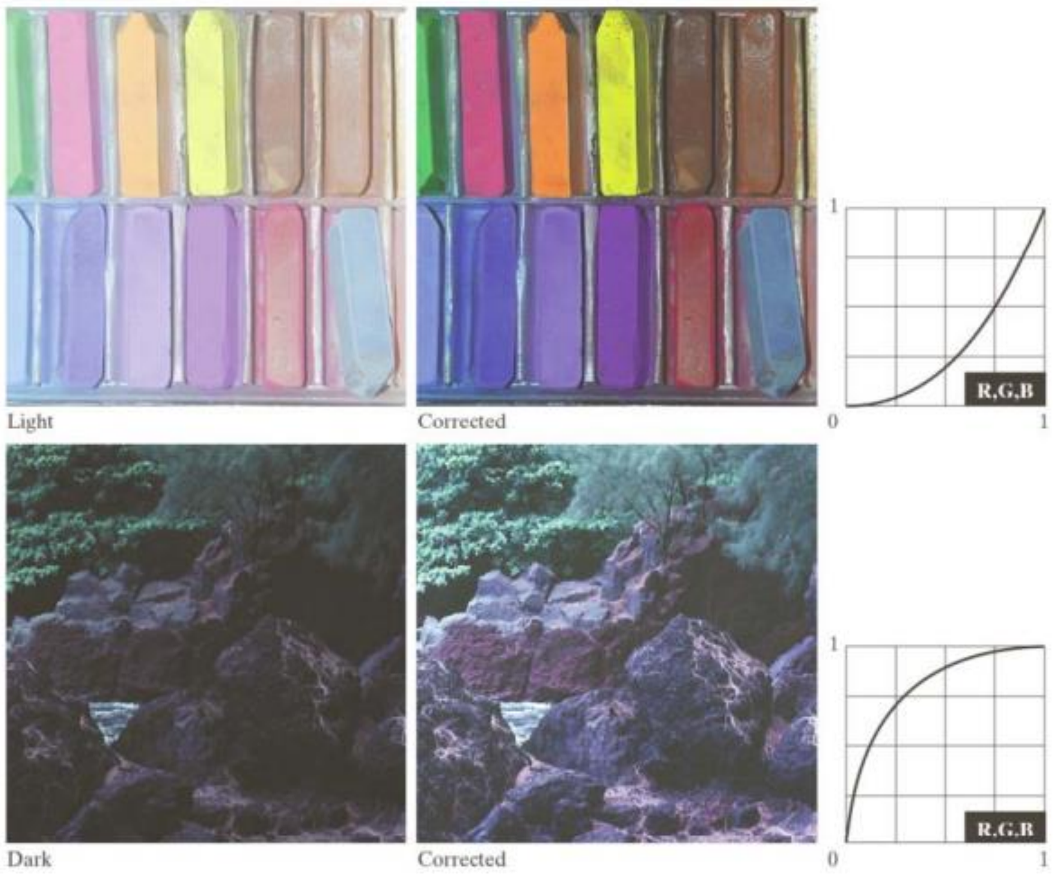
Reds with a radius of 0.2549 from a.

Reds with a radius of 0.1765 from a.

Color Correction

- Most common usage:
  - Photo enhancement
  - Color reproduction (across devices)
- Subjective
  - Attempting to obtain the most *visually pleasing* color image.
- To create neutrality, use a **device-independent color model**
  - For example CIELAB*

seperti HSV



Grayscale vs RGB Color

Grayscale

RGB

R

G

B

- Image processing methods on grayscale images on principle can be applied on color images
- But they may have different results

Histogram Equalization

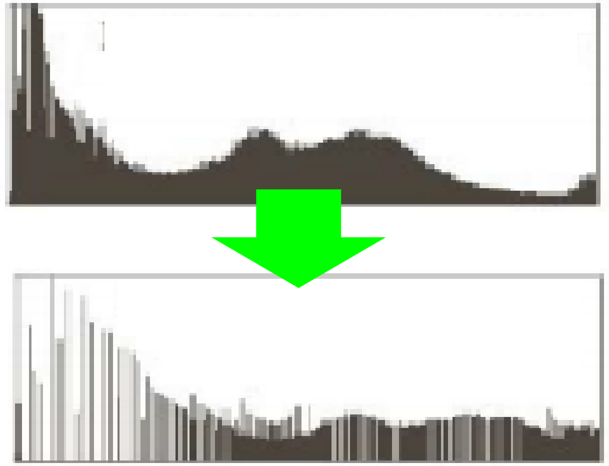
Color Histogram Equalization

Yang mengubah distribusi equalization, jagngan digunakan di RGB, ganti aja di HSI, dll

Karena hasilnya bakal beda

- For color images, HSI’s intensity channel I is particularly suited for this process

Histogram of I before



Histogram of I after

- Usually followed up with saturation correction

Color Image Smoothing

$$\bar{\boldsymbol{c}}(x,y) = \frac{1}{K} \sum_{(s,t)\in S_{xy}} \boldsymbol{c}(s,t)$$

- 2 approaches based on RGB / HSI color spaces
  - RGB: Smoothing on each channel, and join back into new R’G’B’ image
  - HIS: Smoothing on I channel only

Smoothing RGB

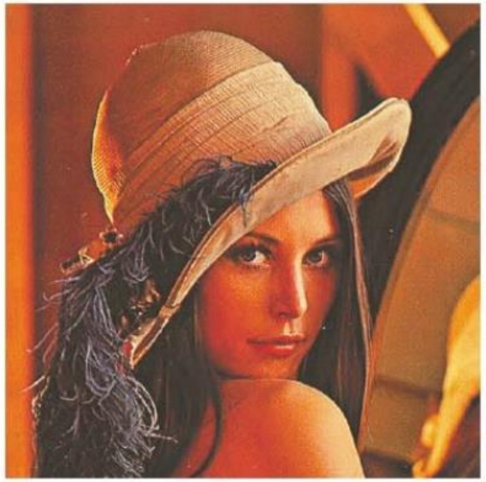
Smoothing HSI

Image Difference

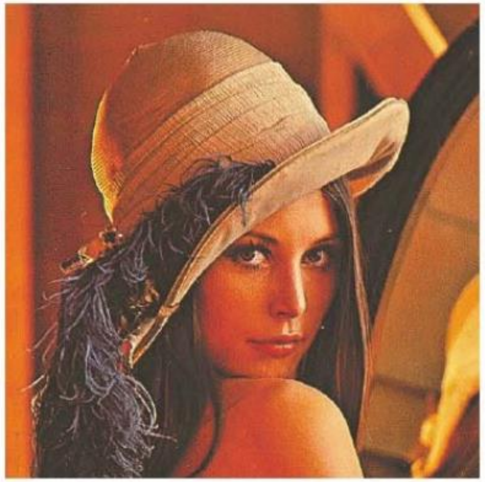
Color Image Sharpening

$$\nabla^2[\boldsymbol{c}(x, y)]$$

- Based on the Laplacian operator (recall grayscale operations)
  - 2 approaches based on RGB / HSI color spaces



Smoothing RGB



Smoothing HSI

Image Difference

Edge Detection

- Per channel vs RGB color vector approach

R

G

B

Color Vector

Per Channel

Image Difference

FAKULTAS ILMU KOMPUTER

UNIVERSITAS INDONESIA

Image Restoration

Image Restoration vs Enhancement

Enchancement, itu bisa trial and error

Both restoration and enhancement do this!

- Image enhancement is done heuristically, via **trial and error,** until we obtain a good image
- In image restoration **we attempt to estimate the model of distortion**

Restoration gak bisa trial and error, maka harus mengestimasi model of distortionnya

Estimation By Mathematical Modeling

- Study the actual physical process that results in the degradation!
- Can be used to estimate environmental effects to image capture

- How did this image happen??

kayak gimana cara satelit dapetin gambar kita padahal jauh banget

[1]Hufnagel, R. E., and N. R. Stanley. "Modulation transfer function associated with image transmission through turbulent media." *JOSA* 54.1 (1964)

Modeling: Atmospheric Effects

- Physical characteristics of atmospheric turbulence[1].

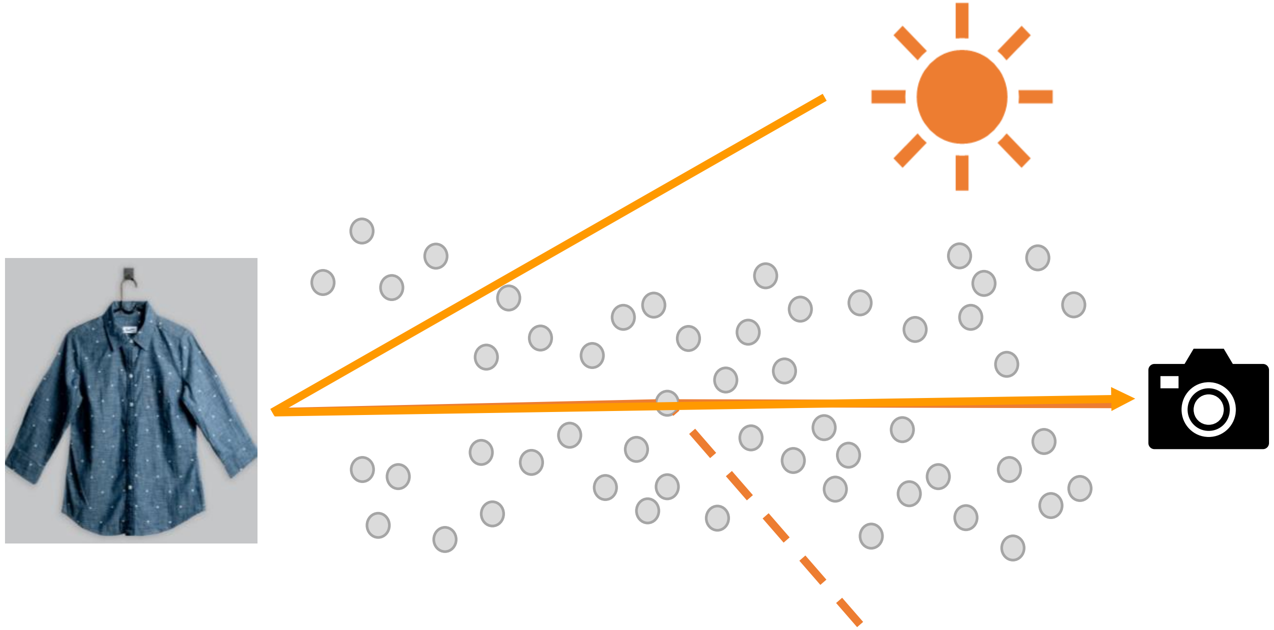
$$H(u,v) = e^{-k(u^2+v^2)^{\frac{5}{6}}}$$

tinggal di inverse distortion modelny

[2]He, Kaiming, Jian Sun, and Xiaoou Tang. "Single image haze removal using dark channel prior." IEEE Transactions on Pattern Analysis and Machine Intelligence 33.12 (2010).

Modeling: Photometric Scattering[2]

- Physical light propagation in media



c : Attenuation coefficient of media
d : Distance of object and camera
J : Original intensity of scene
A : *airlight* (the color of the ambient light)
τ : transmission
β : scattering coefficient

$$I = J\tau + A(1-\tau)$$

$$\tau = e^{-\beta d}$$

Kita gak tau transmissionnya berapa

Image Restoration

- If we know $y = f(x)$, we now then can apply $x = f^{-1}(y)$

Estimate

- Estimate the distortion / degradation model

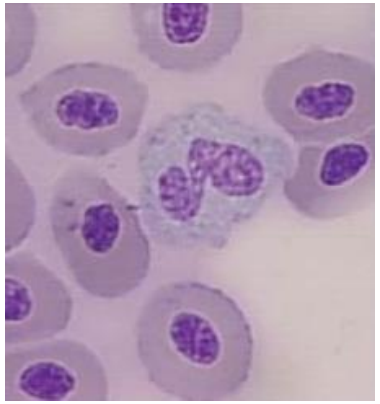
Restore

- Try to invert the model to recover the original image

Morphological Processing

What is Morphology?

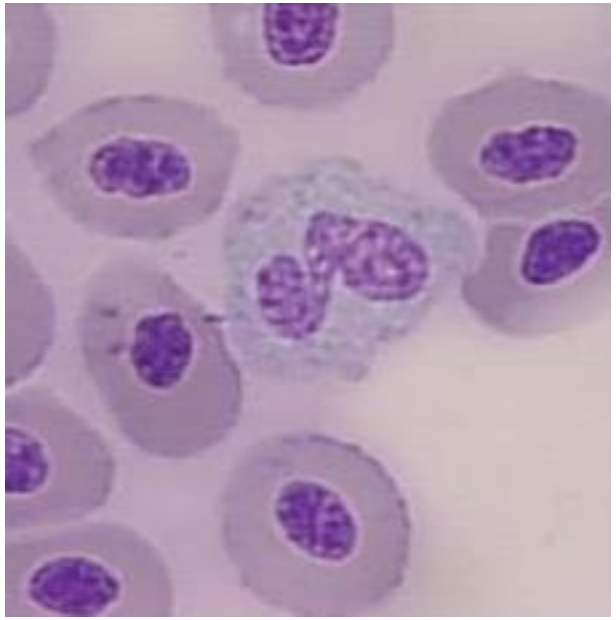
- The study of the **form** of things.
  - [Biology] the form and structure of animals and plants.
  - [Linguistics] the study of the forms of words.
- What about the form and structure of an image?
  - Shapes
  - Boundaries



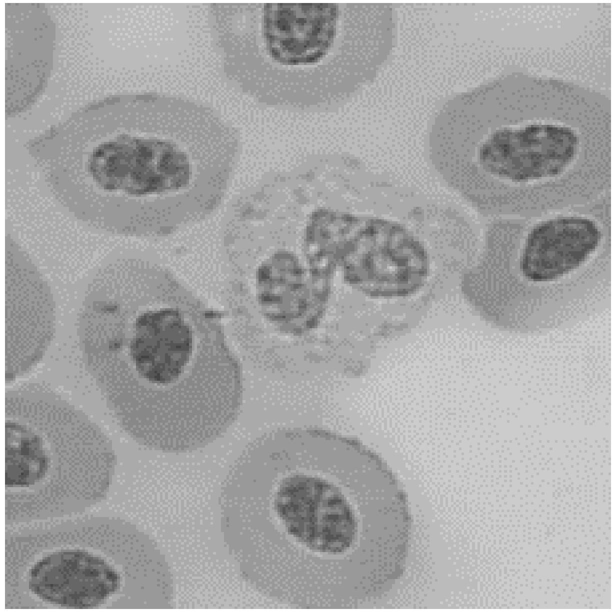
- Mathematical morphology can be used as a tool for **extracting image components** that are useful in the **representation and description** of region shapes such as areas, boundaries, etc.
- We can use morphological techniques for:
  - image pre-processing
  - intermediate and post-processing: segmentation or classification

Detecting Objects

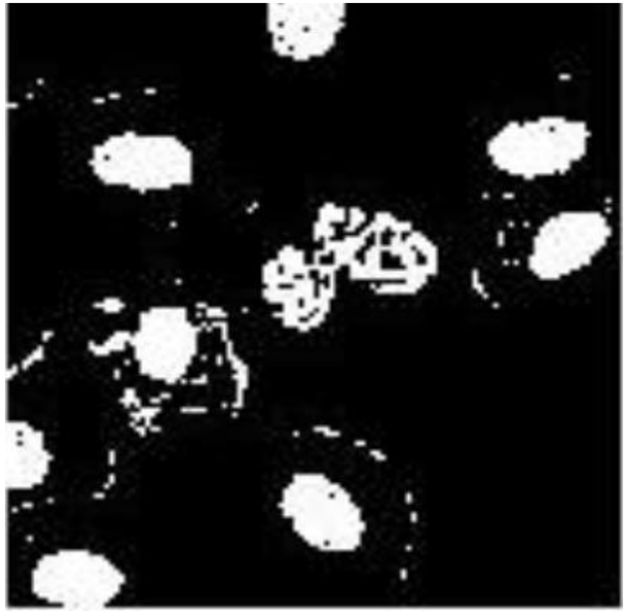
- How many bacteria are in this image?



Original Image



Grayscale Image

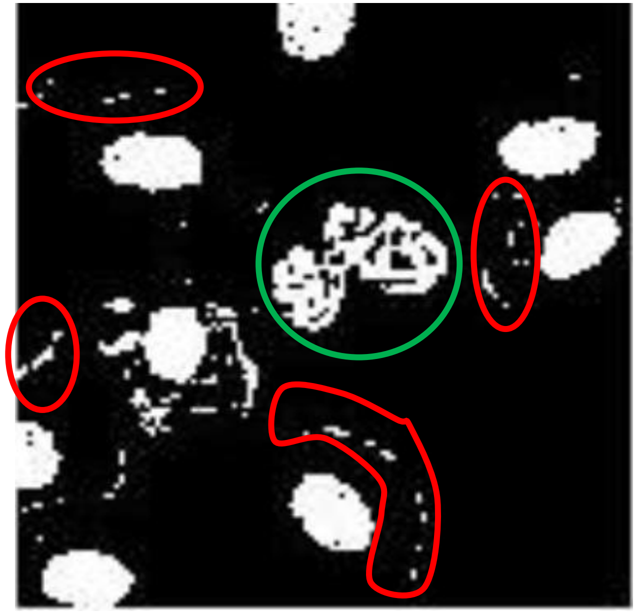


Thresholded Image

Detecting Objects (2)

- How many bacteria are in this image?
  - Too small objects → need to remove them
  - 1 object is separated into multiple parts → need to connect these parts

Kita perlu melakukan probing berdasarkan structural elenmentnya



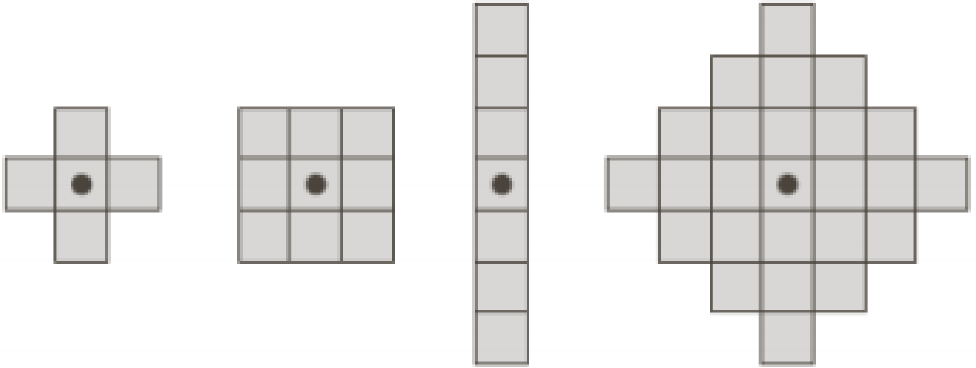
Thresholded Image

9 objects detected

Structural Elements (Strel)

Di probe

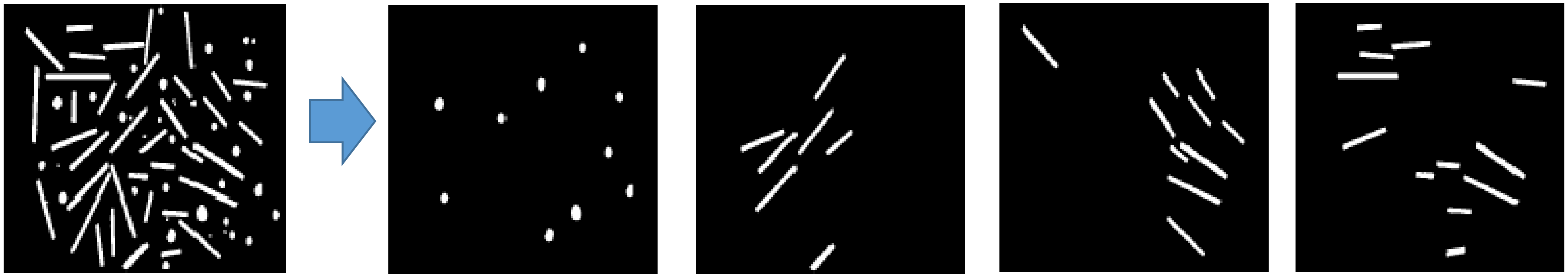
- Morphology formulates operations based on *structural elements* (**SE/strel**)
- Structural elements: Small sets or sub-images **to probe** an image for properties / forms of interest - should be much **smaller** than the image.
- The shape of the structured element is arbitrary, as long as it can be represented as a binary image of a given size.
  - Ex: buildings may be found with an elongated-square-shape strel



Morphological Operations

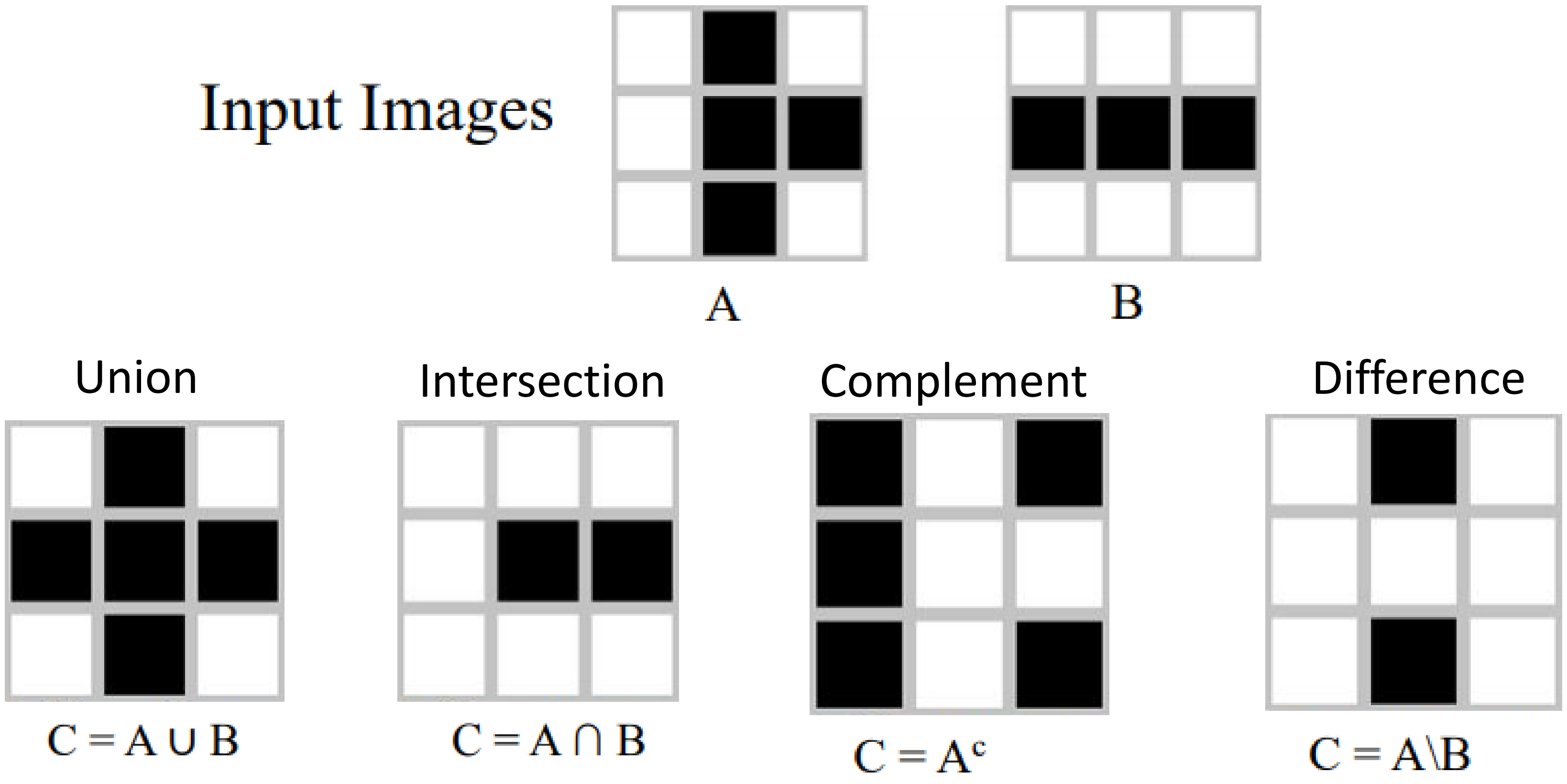
- The shapes detected/found is based on the SE/strel used
- The image is probed by passing the structural elements across the image
- Effects of different SEs:

Misalnya di probe dengan bentuknya beda-beda nanti algoritma akan bisa mengklasifikasikan gambar sesuai dengan bentuknya.



We obtain the detected objects through operations using the SE.

Recall: Basic Set Operations

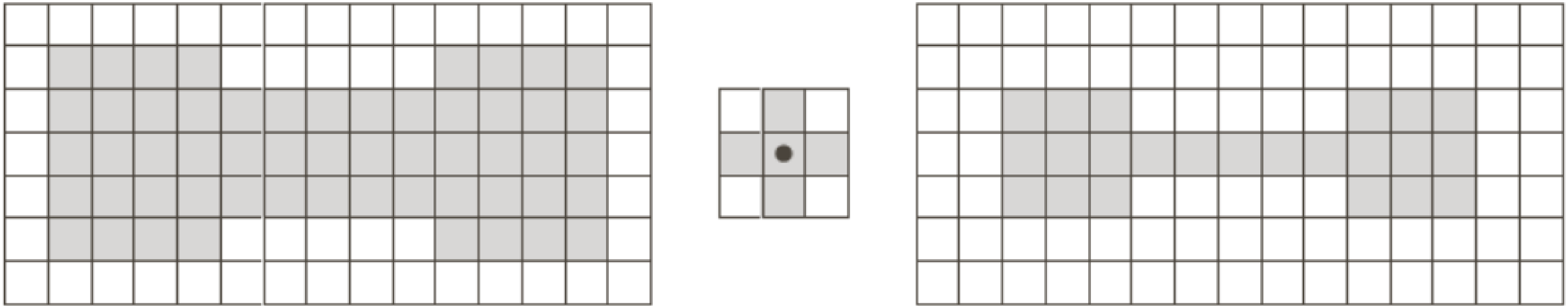


Erosion

Menghilangkan gambar yang kecil (tidak signifikan)

Erosion, ktia pic yang boleh nyala apa engga

- With A and B sets in Z^2, the erosion of A by B is $A \ominus B = \{z | (B)_z \subseteq A\}$
- The structural element is applied to all pixels of the image, and the pixel is turned on only if **the entire structural element** falls within the foreground area.
- B must be contained in A -- hence B doesn't share components with the background.

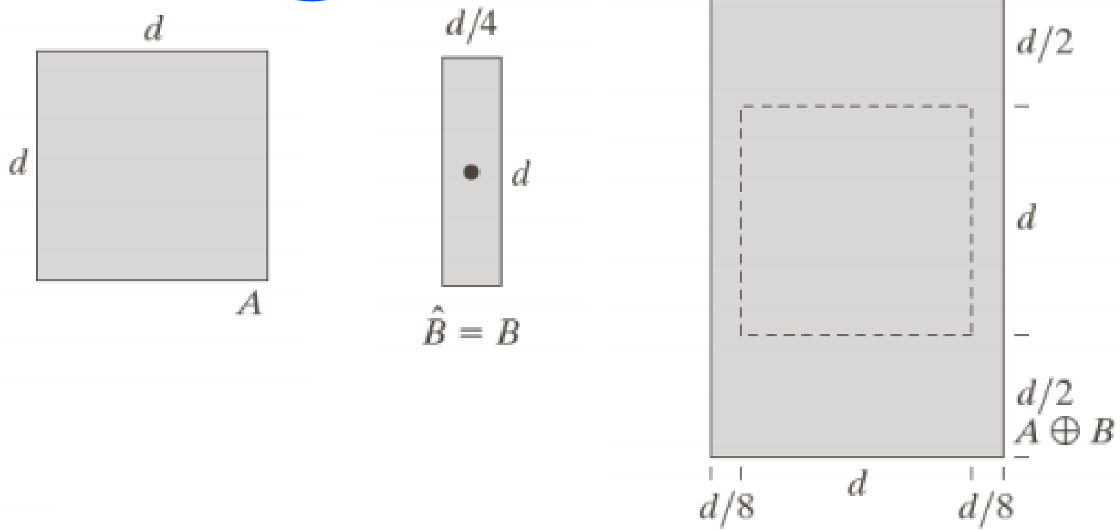


$$A \ominus B = \{z | (B)_z \cap A^C = \emptyset\}$$

Dilation

- With A and B sets in Z^2, the dilation of A by B is $A \oplus B = \{z | [(\hat{B})_z \cap A] \subseteq A\}$
- The structural element, SE, is applied to all pixels of the image
- The pixel is turned on if **at least 1 pixel of the structural element** falls within foreground area (pixels).

Mengabbungngkan yang relevan
dengan dilasi)



$$A \ominus B = \{z | (\hat{B})_z \cap A^C \neq \emptyset\}$$

Opening and Closing

erosi dulu baru dilasi

Opening

- Opening: Erosion then dilation with the same structuring element
- Erosion removes small details and darkens the image
- Dilation increases the overall intensity
- The opening-region image results in a collection of foreground parts (objects of interest) that fit a particular strel

$$f \circ b = (f \ominus b) \oplus b$$

dilasi dulu baru erosi

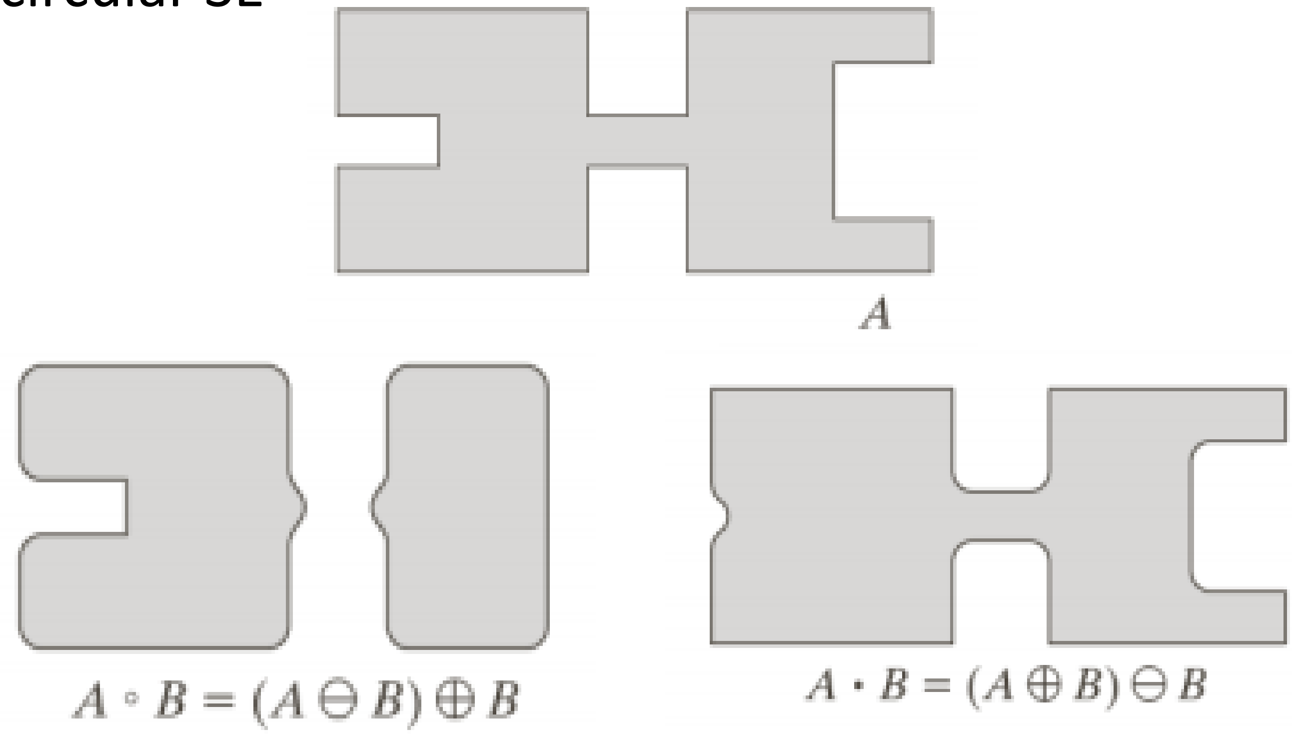
Closing

- Closing: Dilation then erosion using the same structuring element
- The dilation removes dark details and brightens the image,
- The erosion darkens the image
- The closing-region image is a collection of background parts that fit a particular strel

$$f * b = (f \oplus b) \ominus b$$

Opening and Closing (2)

- Using a small circular SE



Opening Closing

Filtering

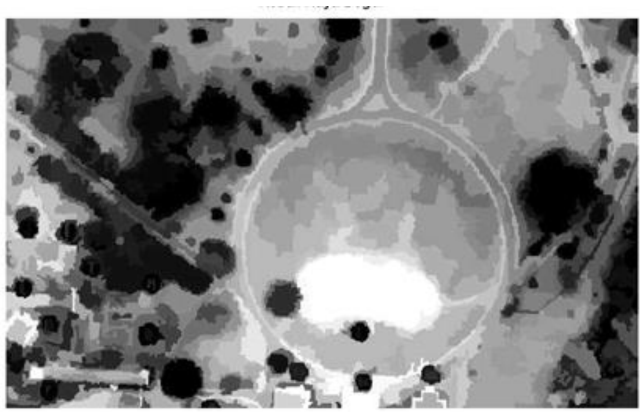
closing dulu baru dikurangin

Top-hat filtering $f - (f \circ b)$

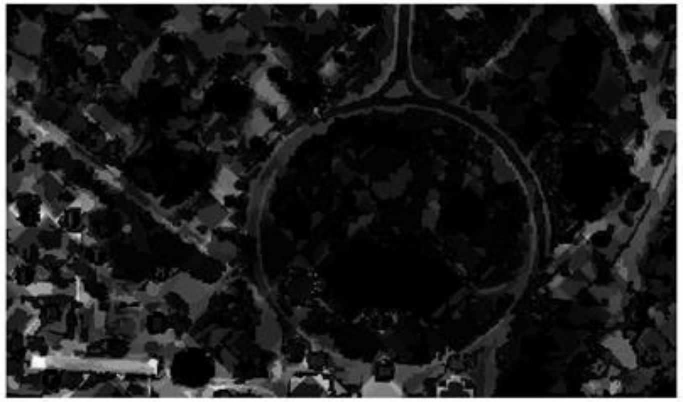
- contains the ‘peaks’ of objects of interest that fit a particular structuring element

Bottom-hat Filtering $(f * b) - f$

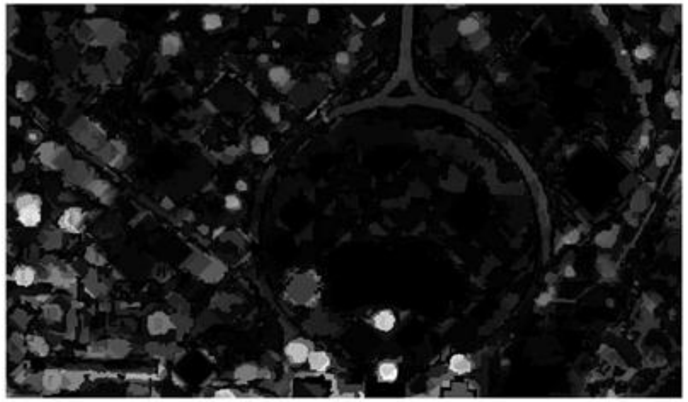
- contains the ‘gaps’ between the objects of interest that fit a particular structuring element



Original Image



Top-hat filtered image

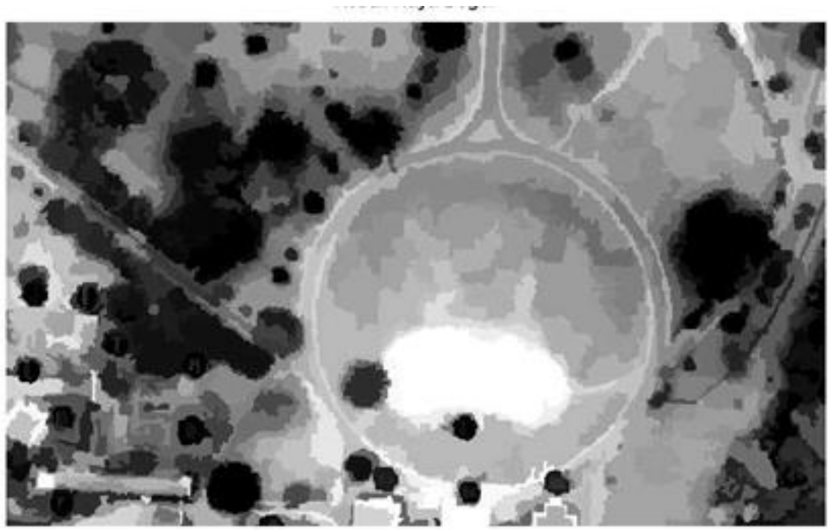


Bottom-hat filtered image

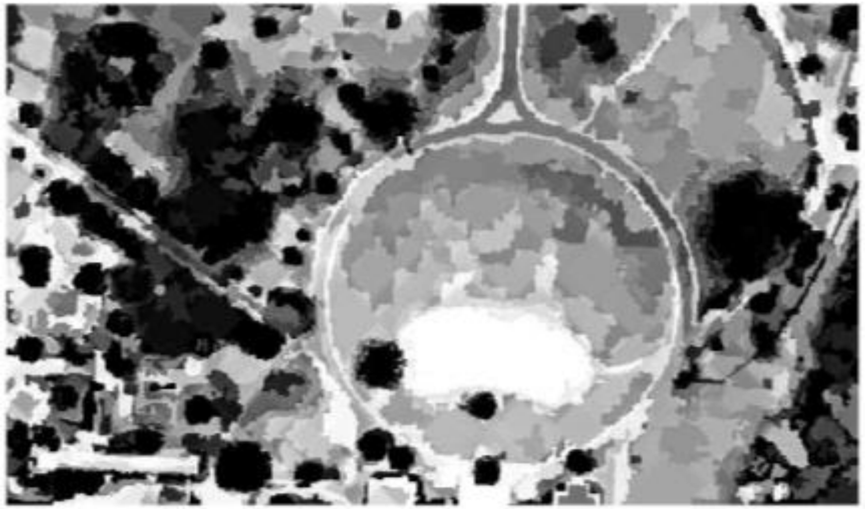
Contrast Enhancement

$$3f - (f \circ b) - (f * b)$$

- Maximizes the contrast between the objects and the gaps (original + top-hat filtered – bottom-hat filtered)



Original Image



Contrast Enhanced Image

Boundary Extraction

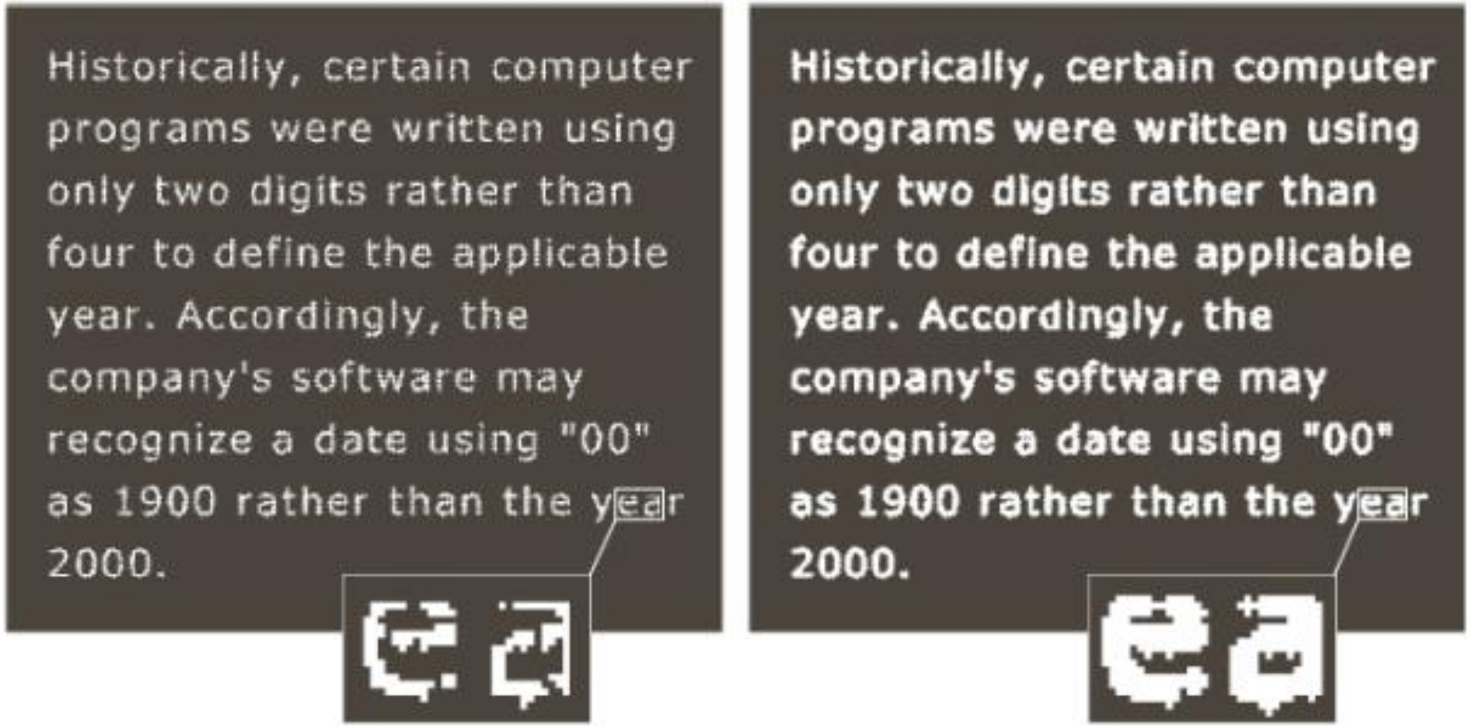
Bisa menggunakan erosi, biar dapet boundarynya.

$$\beta(A) = A - (A \ominus B)$$

Dilation to Clean Up Shapes

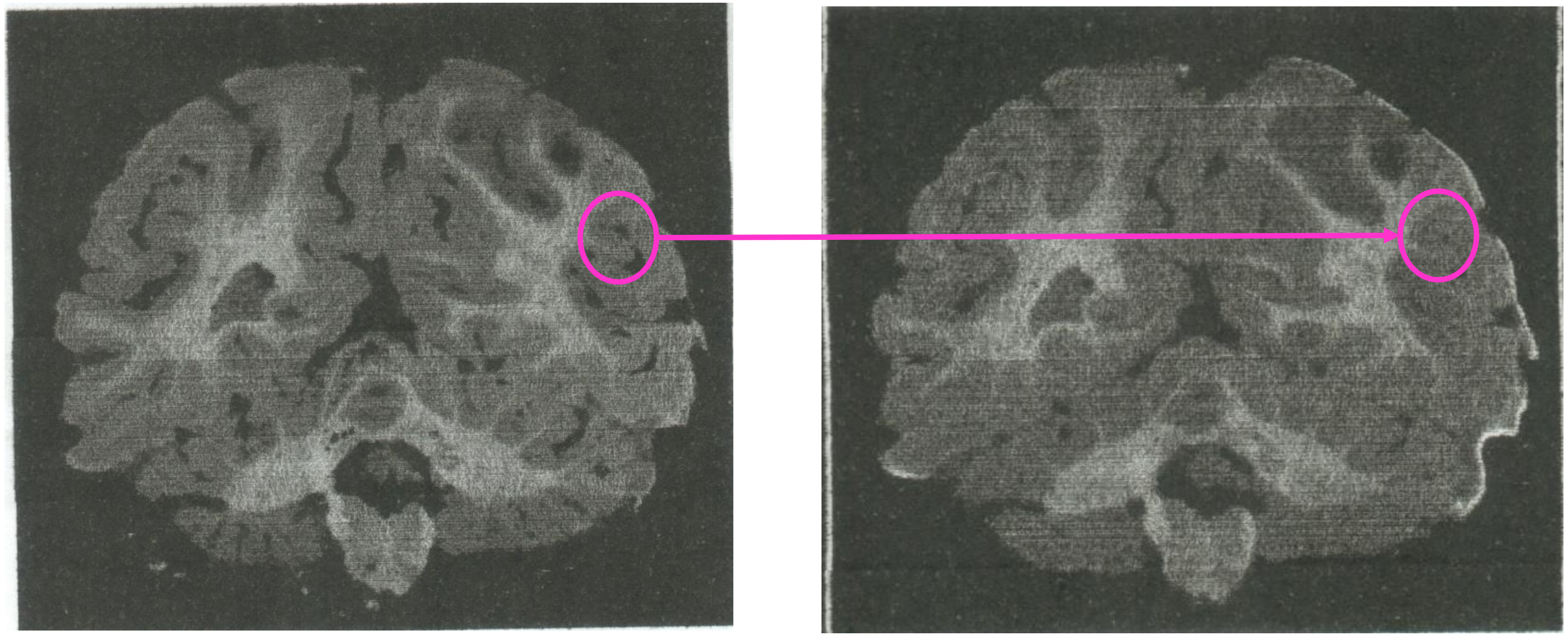
- Dilation by a strel

0	1	0
1	1	1
0	1	0

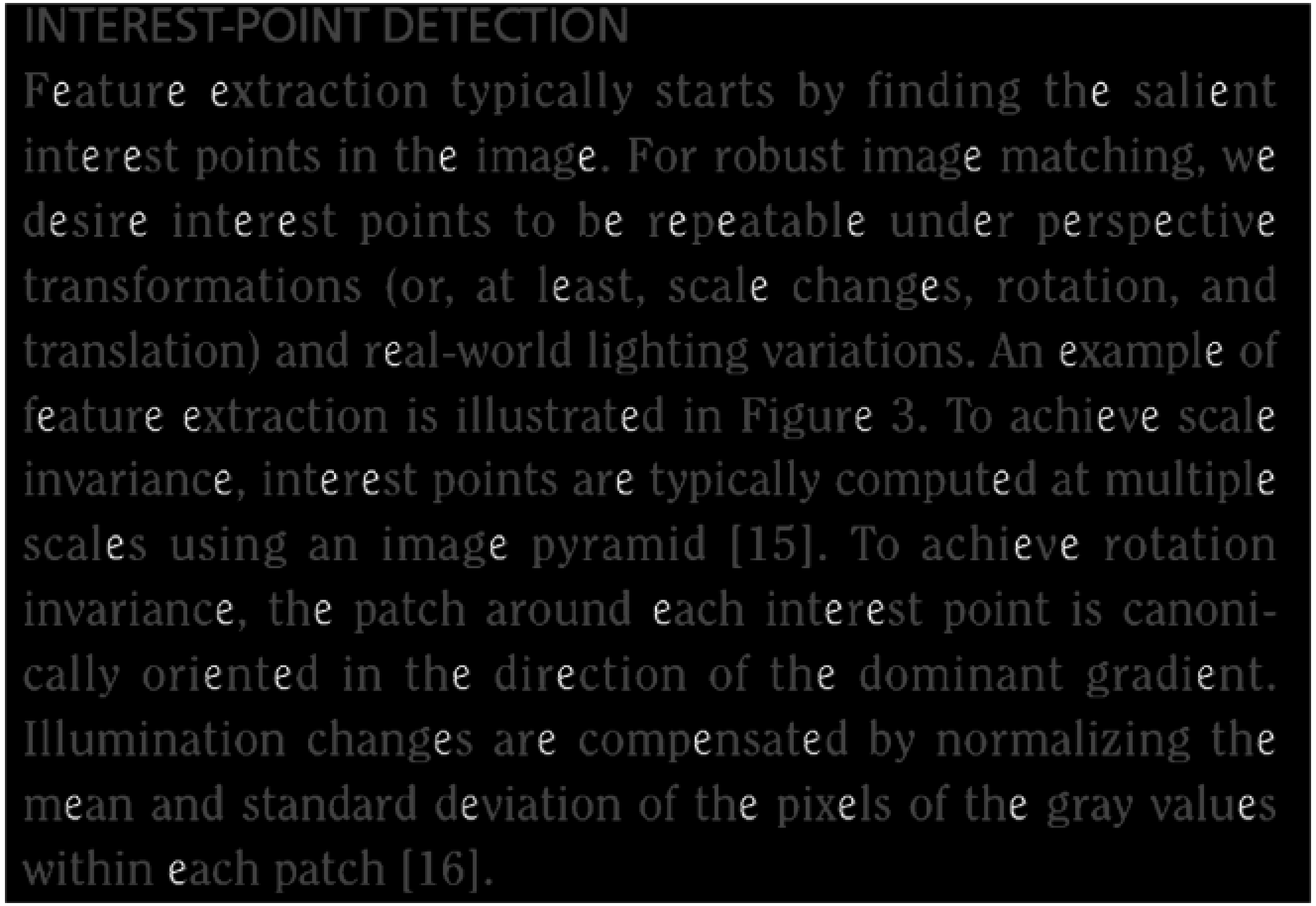


Dilation on Medical Images

- 9x9 circular structuring elements
- Fills in the gaps in the brain and smooths contour



Detection and Recognition

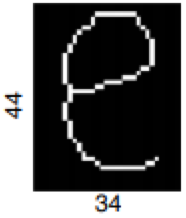


INTEREST-POINT DETECTION

Feature extraction typically starts by finding the salient interest points in the image. For robust image matching, we desire interest points to be repeatable under perspective transformations (or, at least, scale changes, rotation, and translation) and real-world lighting variations. An example of feature extraction is illustrated in Figure 3. To achieve scale invariance, interest points are typically computed at multiple scales using an image pyramid [15]. To achieve rotation invariance, the patch around each interest point is canoni-cally oriented in the direction of the dominant gradient. Illumination changes are compensated by normalizing the mean and standard deviation of the pixels of the gray values within each patch [16].

Mengekstrasi seluruh huruf e

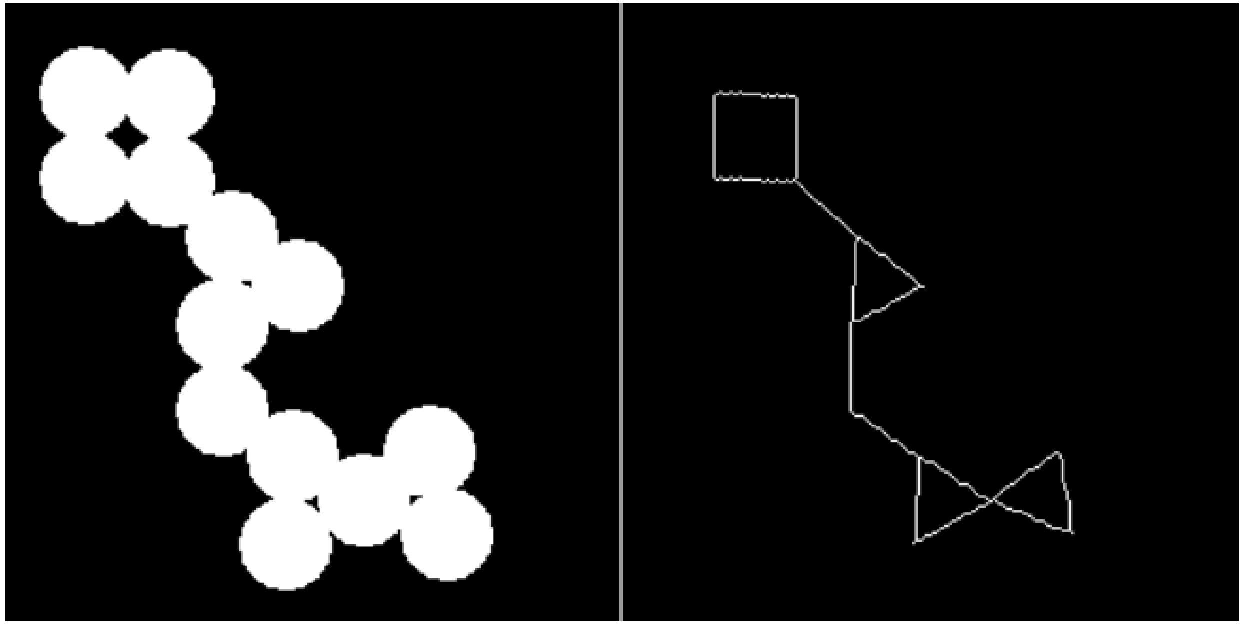
Structuring element W



Skeletonization

Intinya cuman perlu tau rangka dari gambar yang ada.

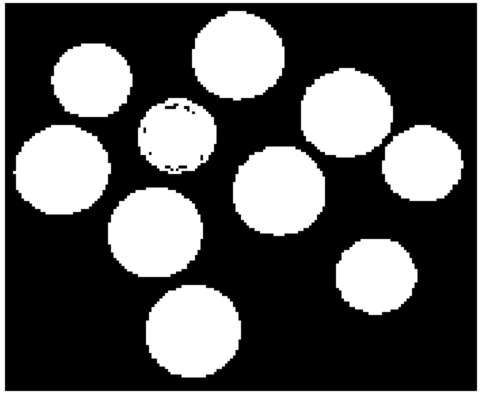
- Skeletonize objects in a binary image.
- Erosion to make all objects into centerlines --- without changing the essential structure of the objects, such as the existence holes and branches



Convex Hull

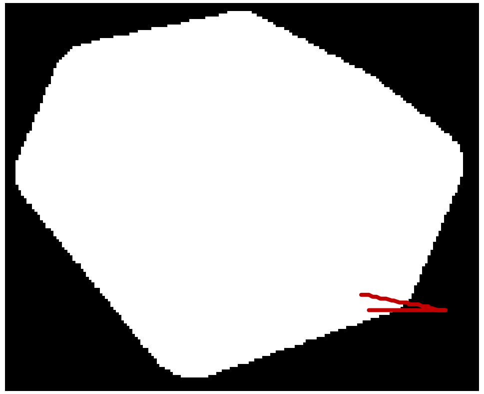
Original

Binary



Kita cuman perlu tau area of hull (area yang perlu diperhatikan)

Union Convex Hull



Objects Convex Hull