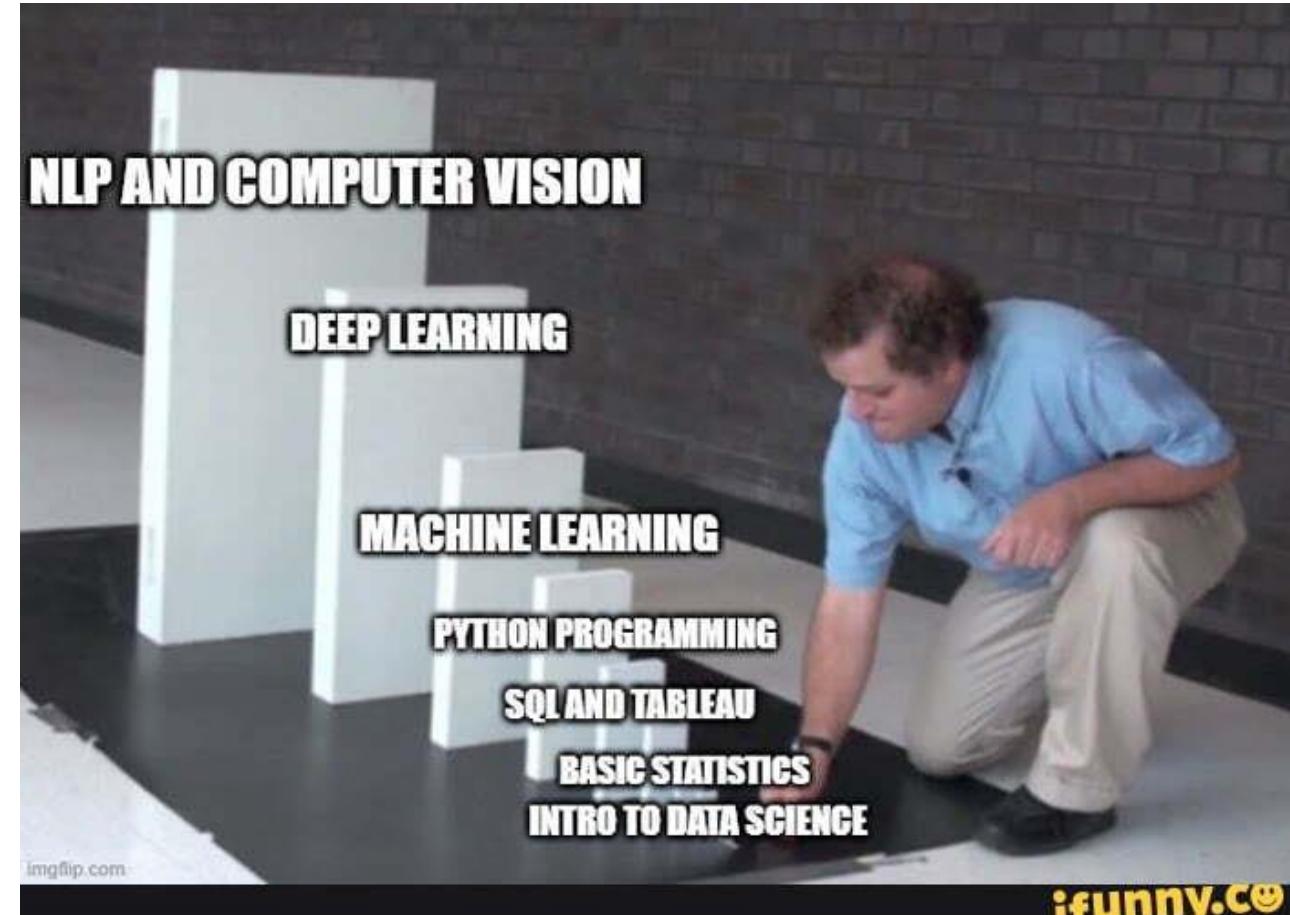


# Pengantar

Ghandy, M.Sc.

IBDA4311 – *Computer Vision*

# Computer Vision

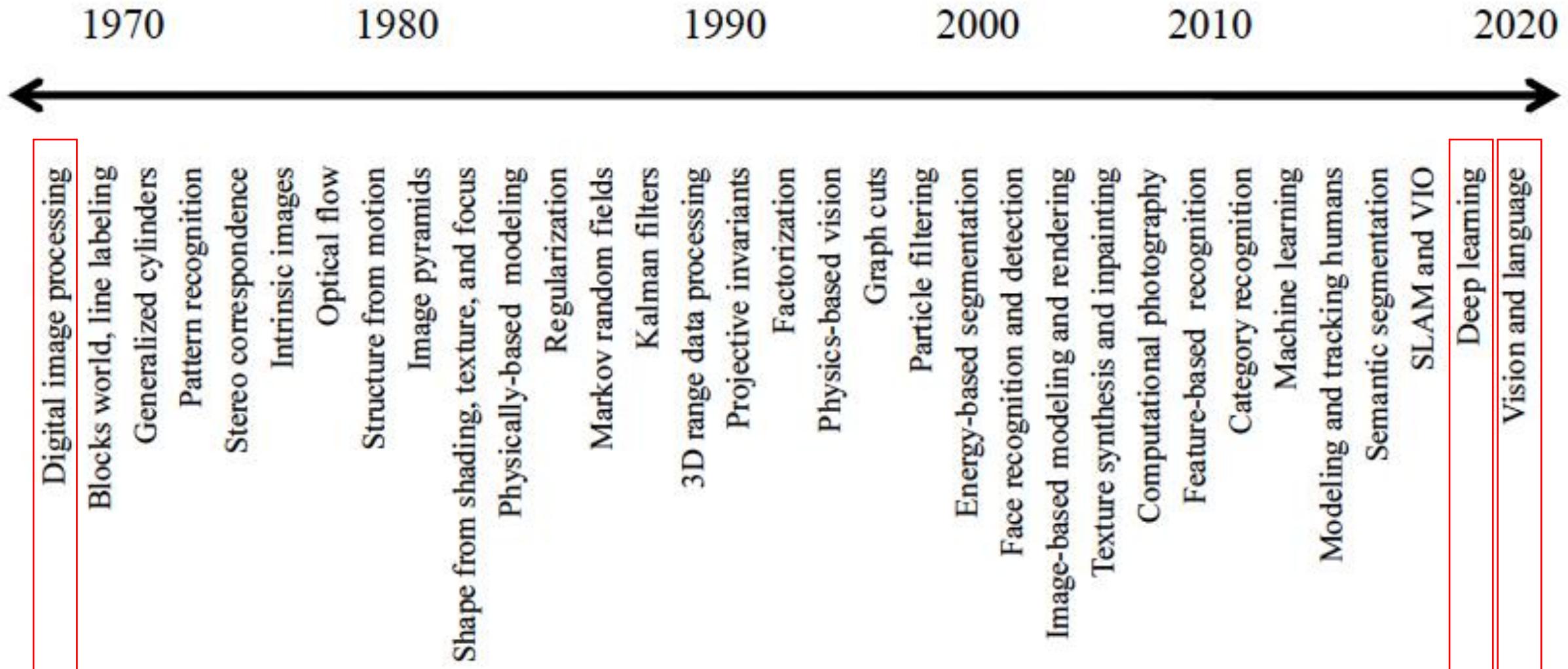


**COMPUTER VISION PROBLEMS**



hit the button

# Computer Vision

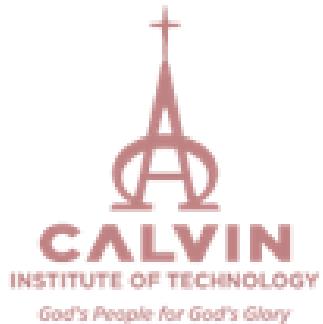


# Capaian Pembelajaran

- Silabus
- Instalasi
- Visi

# Silabus

# Silabus



**Fakultas Sains dan Teknik  
Course Syllabus  
Semester Ganjil 2023/2024**

---

<b>Mata Kuliah</b>	: Computer Vision
<b>Kode Mata Kuliah</b>	: IBDA4311
<b>Bobot SKS</b>	: 3

# Logistik

- Jam kuliah : Selasa (09.00 – 12.00 WIB, 3x50 menit)
- Tempat : Lab Komputer 1 (Fisik)
- Harap menggunakan laptop ketika menghadiri kelas
- *Office hours* : by appointment
- E-mail : [ghandy.salim@calvin.ac.id](mailto:ghandy.salim@calvin.ac.id)
- Komunikasi : Canvas (silabus, materi kuliah, tugas, *announcement*) maupun e-mail

# Pustaka

## Wajib

- [1] R. Szeliski, *Computer vision: algorithms and applications*. 2<sup>nd</sup> ed, London ; New York: Springer, 2022.  
<https://szeliski.org/Book/>

## Rekomendasi

- [2] Stephen-Palmer, *Vision Science: Photons to Phenomenology*. MIT Press, 1999.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning.*, Cambridge, Massachusetts: The MIT Press, 2016. [deeplearningbook.org/](http://deeplearningbook.org/)

# Tips untuk berhasil dalam MK ini

- Konsisten mengerjakan tugas dan proyek sebaik mungkin.
- Bertanya dan partisipasi secara aktif di dalam kelas, sehingga dosen juga mendapat *feedback* jika terlalu cepat atau ada yang kurang jelas.
- Tidak tertinggal pelajaran: jika tidak paham di awal, segera menghubungi dosen atau bertanya kepada mahasiswa lainnya.
- Gunakan *office hours* dosen untuk lebih memahami materi dan tugas.
- Mata kuliah ini banyak memakai pemrograman dengan Python dengan *libraries* seperti numPy, scikit-learn, pandas, keras, pytorch, tensorflow. Bacalah dokumentasi dari *libraries* untuk lebih terbiasa menggunakannya. Anda juga dapat menggunakan ChatGPT untuk memahami kode-kode pemrograman yang diberikan

# Setelah Anda menyelesaikan perkuliahan ini, Anda akan:

- Memahami dasar matematis, fisika, filosofis, dan teoretis yang digunakan di dalam *computer vision*
- Mampu membuat model *computer vision* untuk menyelesaikan berbagai macam permasalahan

# Instalasi

# Anaconda

- Anaconda adalah sistem manajemen paket python
- Melalui Anaconda, maka versi dari paket python yang ada akan dikelola oleh package management system conda.



# Instalasi Anaconda

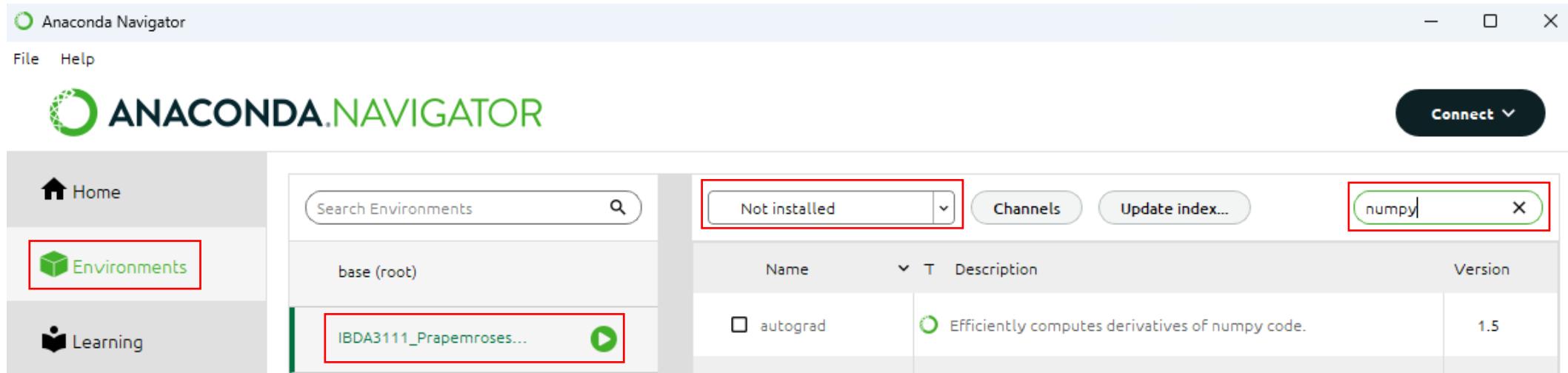
Gunakan tutorial di bawah ini untuk instalasi Anaconda:

1. [https://www.youtube.com/watch?v=itVBw\\_rNYjl&t=315s&ab\\_channel=GeekyScript](https://www.youtube.com/watch?v=itVBw_rNYjl&t=315s&ab_channel=GeekyScript)

Gunakan tutorial di bawah ini untuk membuat “*new environment*” pada Anaconda (**Gunakan Python 3.9.17 untuk mata kuliah ini**):

1. [https://www.youtube.com/watch?v=svOca-6liM8&ab\\_channel=LigentTutorials](https://www.youtube.com/watch?v=svOca-6liM8&ab_channel=LigentTutorials)
2. [https://www.youtube.com/watch?v=A8dlq-AUDpc&ab\\_channel=TheEducationMachine](https://www.youtube.com/watch?v=A8dlq-AUDpc&ab_channel=TheEducationMachine)

# Paket Python



Instalasi *library* berikut ini pada *environment* baru anda:

- |                 |                 |
|-----------------|-----------------|
| 1. numpy        | 7. tensorflow   |
| 2. scipy        | 8. keras        |
| 3. matplotlib   | 9. pytorch      |
| 4. seaborn      | 10. jupyterLab  |
| 5. pandas       | 11. statsmodels |
| 6. scikit-learn | 12. gensim      |
|                 | 13. nltk        |

# GPU cloud

- CIT (hanya untuk riset)
- GCP, AWS (mahal)
- Azure (free \$200)
- Google colabs (gratis dengan limit)
- Gradient (gratis dengan limit): <https://gradient.run/free-gpu>
- Deepnote (gratis dengan limit): <https://deepnote.com/>

Perbandingan CPU dan GPU:

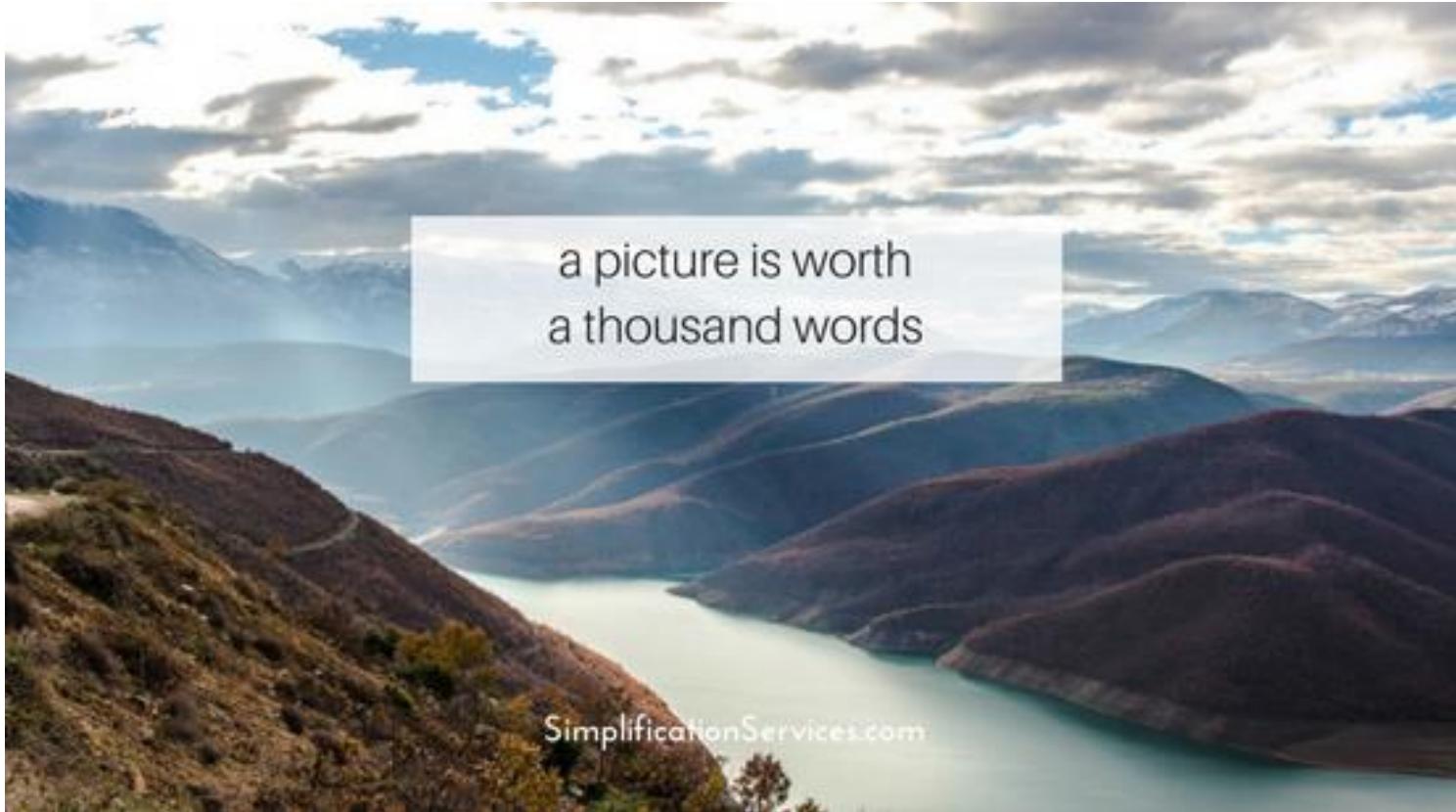
<https://colab.research.google.com/notebooks/gpu.ipynb>

# Visi



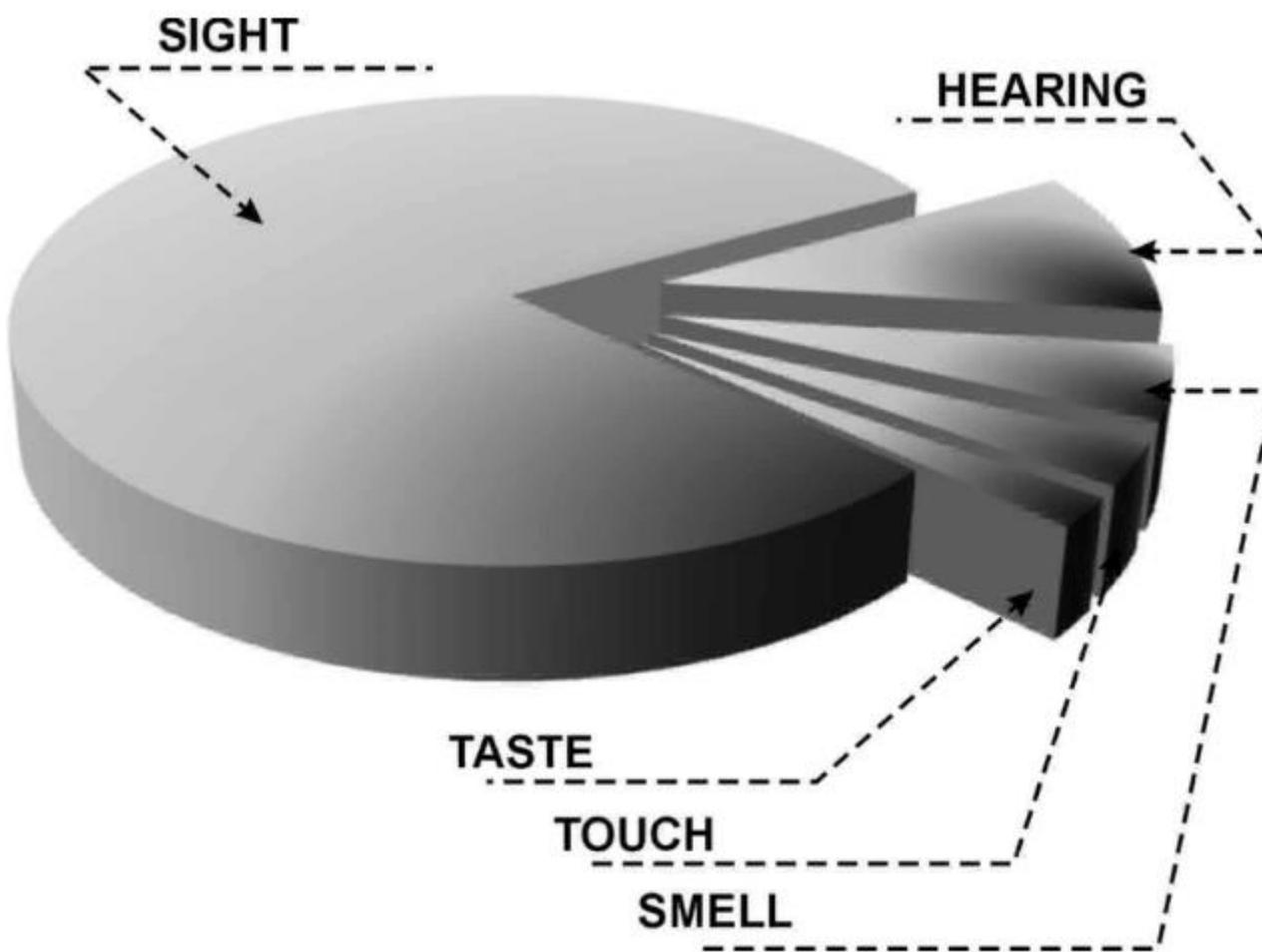
# Apa itu visi?

- "A picture is worth a thousand words"

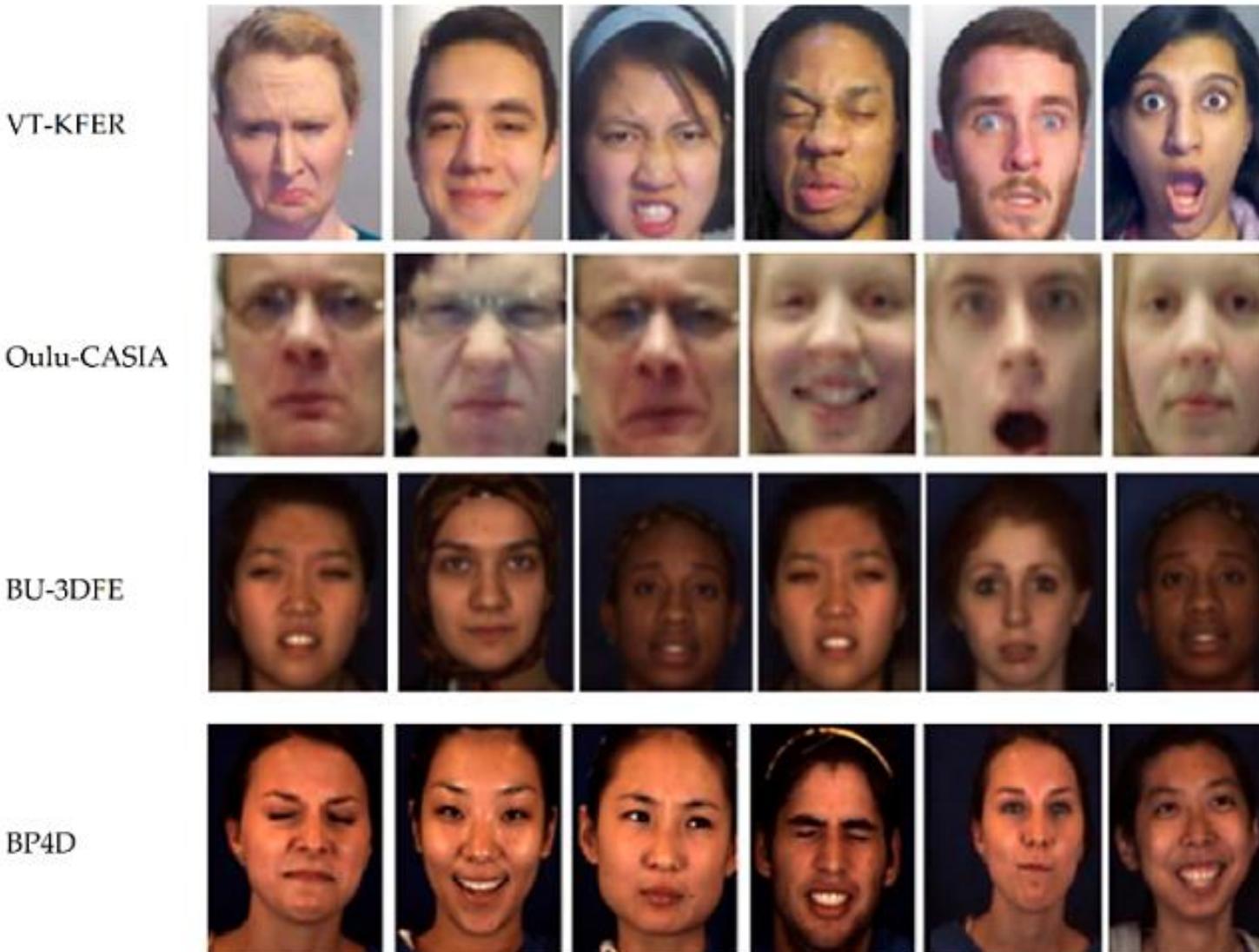


- Visi menentukan aksi

# Human Sensory



# Emotion Detection Based on Images



<https://images.app.goo.gl/arDkySuQ5iXMgDLU8>

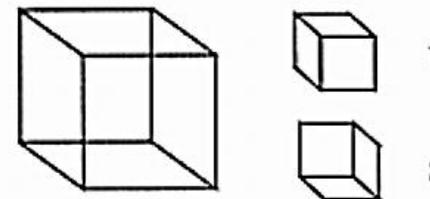
© Copyright 2020 Calvin Institute of Technology.  
The information contained herein shall not be used, copied, published, quoted, and/or released without prior approval.

# Persepsi Visual

- Pemodelan lingkungan



A. Vase/Faces



B. Necker Cube



C. Duck/Rabbit

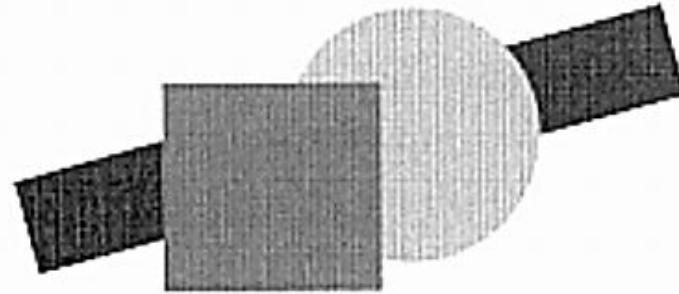
Figure 1.1.5

C. Duck/Rabbit

Ambiguous figures. Figure A can be seen either as a white vase against a black background or as a pair of black faces against a white background. Figure B can be seen as a cube viewed from above or below. Figure C can be seen as a duck (facing left) or a rabbit (facing right).

# Persepsi Visual

- Pemodelan lingkungan



A



B

Figure 1.1.6

Visual completion behind partly occluding objects. Figure A is perceived as consisting of a square, a circle, and a rectangle even though the only visible regions are those shown separated in Figure B.

# Persepsi Visual

- Konstruksi Makna



Figure 1.1.9

Many kinds of dogs. Visual perception goes beyond the physical description of objects to classify them into known categories. Despite the substantial physical differences in their appearance, all these animals are readily perceived as belonging to the category of dogs.

# Limitasi visi

- Lukisan impressionist berusaha memvisualkan aspek yang melampaui batasan visual



<https://images.app.goo.gl/NYdj3zQMqHpyst9SA>



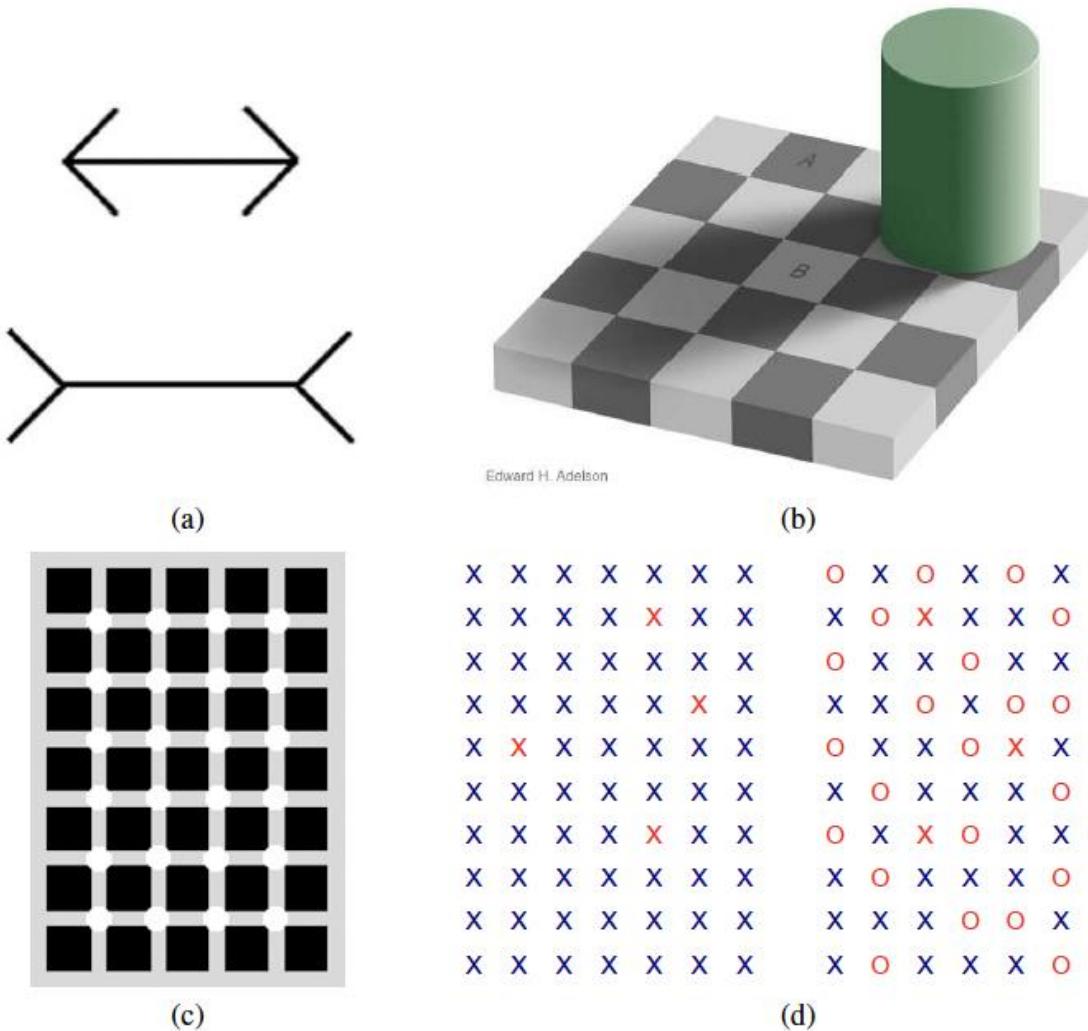
<https://perfectpicturelights.com/blog/impressionism-light-and-shadow>

# Limitasi visi

- Lukisan impressionist berusaha memvisualkan aspek yang melampaui batasan visual



# Ilusi Optik

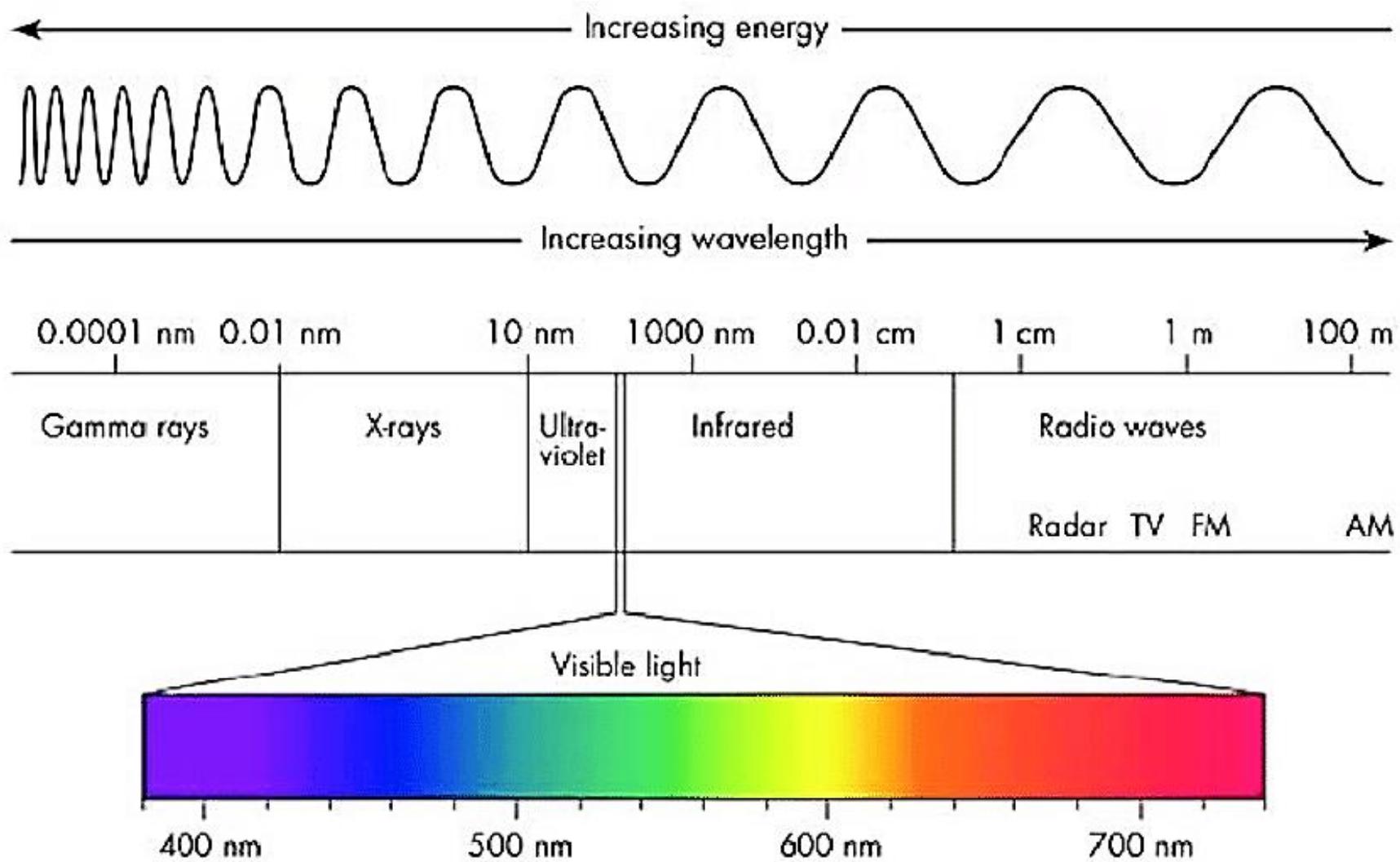


**Figure 1.3** Some common optical illusions and what they might tell us about the visual system: (a) The classic Müller-Lyer illusion, where the lengths of the two horizontal lines appear different, probably due to the imagined perspective effects. (b) The “white” square B in the shadow and the “black” square A in the light actually have the same absolute intensity value. The percept is due to *brightness constancy*, the visual system’s attempt to discount illumination when interpreting colors. Image courtesy of Ted Adelson, <http://persci.mit.edu/gallery/checkershadow>. (c) A variation of the Hermann grid illusion, courtesy of Hany Farid. As you move your eyes over the figure, gray spots appear at the intersections. (d) Count the red Xs in the left half of the figure. Now count them in the right half. Is it significantly harder? The explanation has to do with a *pop-out* effect (Treisman 1985), which tells us about the operations of parallel perception and integration pathways in the brain.

# Informasi Optikal

- Persepsi visual tergantung pada 3 hal:
  - Cahaya
  - Objek yang memantulkan cahaya
  - Sistem visual pengamat

# Spektrum Cahaya

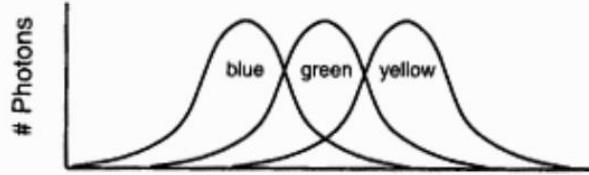


# Istirahat 15 Menit

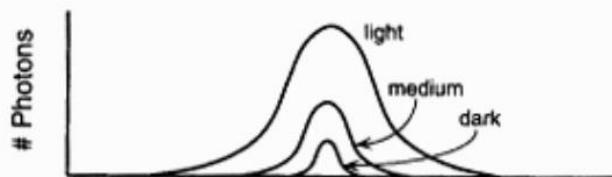
# Color

- Color as physiological and psychological properties

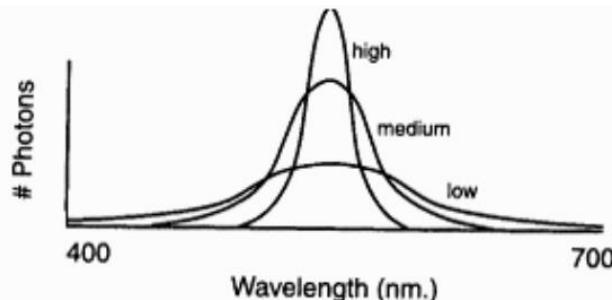
- Hue



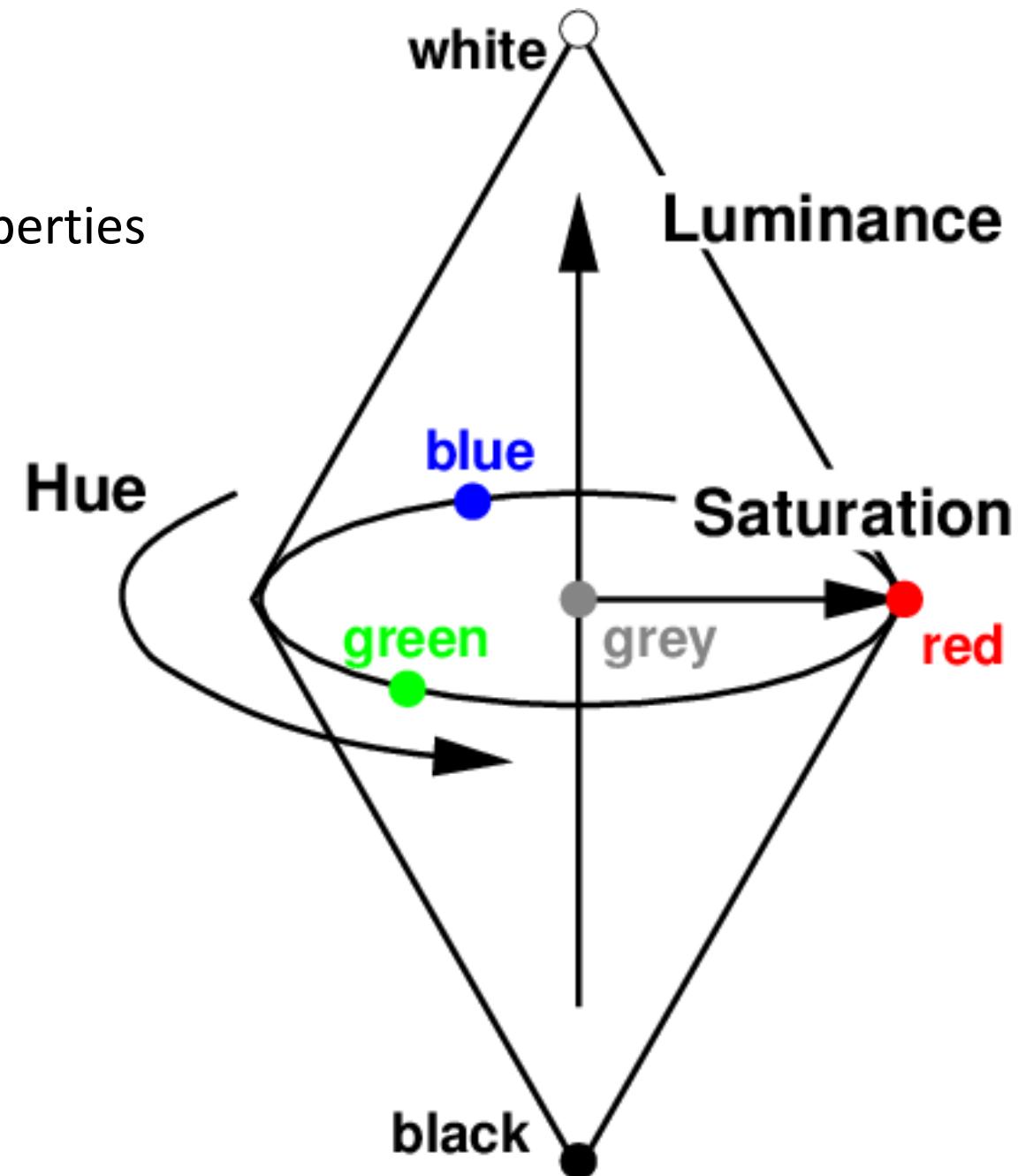
A. Mean  $\leftrightarrow$  Hue



B. Area  $\leftrightarrow$  Lightness



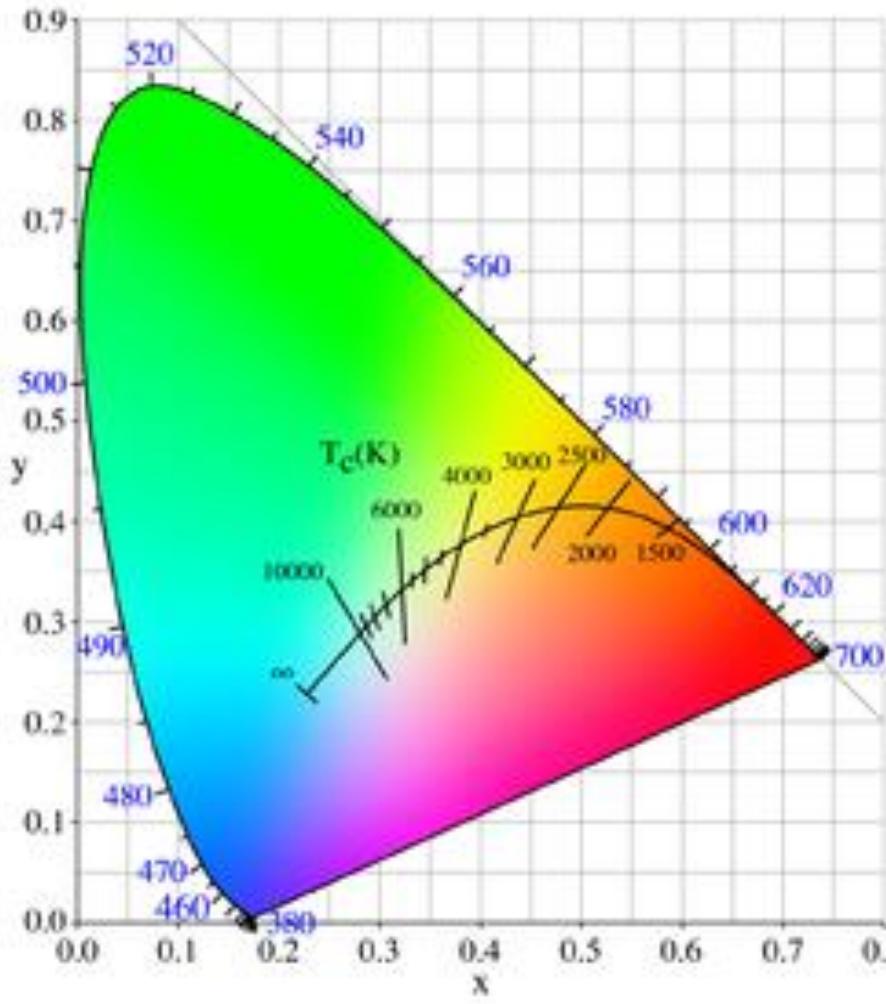
C. Variance  $\leftrightarrow$  Saturation





# Color Space

- Color space dibentuk dari kombinasi hue dan saturation

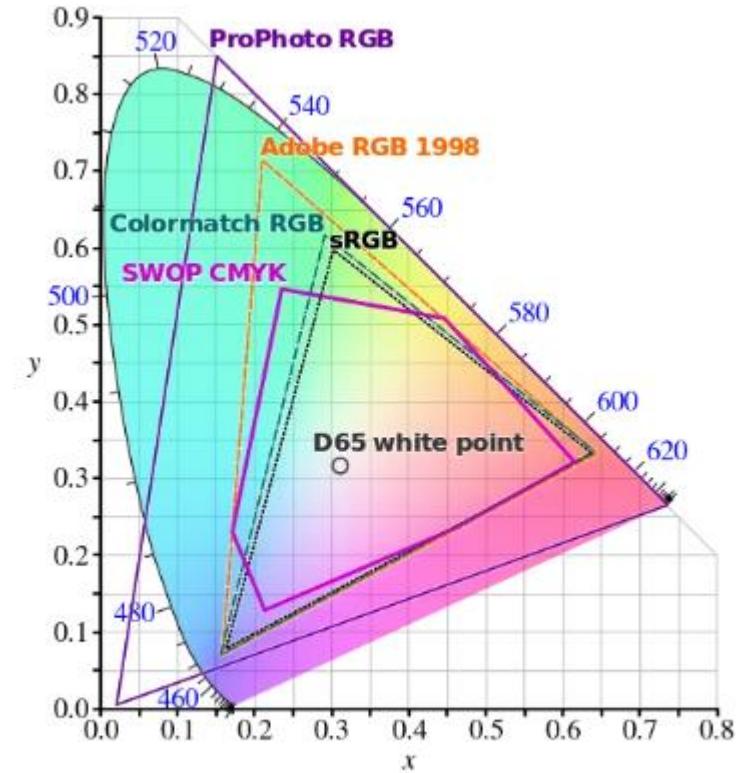


# Color Space

## Apa itu *Color Space*?

Warna adalah bagian integral dari sistem visi komputer. Color Space adalah kombinasi warna secara terorganisir dan terutama digunakan saat kita mereproduksi warna dalam media yang berbeda. Misalnya, Anda memiliki layar laptop dan monitor besar 52". Tentunya, akan ada perbedaan total piksel dan kombinasi warna. Ruang warna yang berbeda membantu membuatnya kompatibel dengan resolusi layar untuk memberikan pengalaman kombinasi warna yang lebih baik.

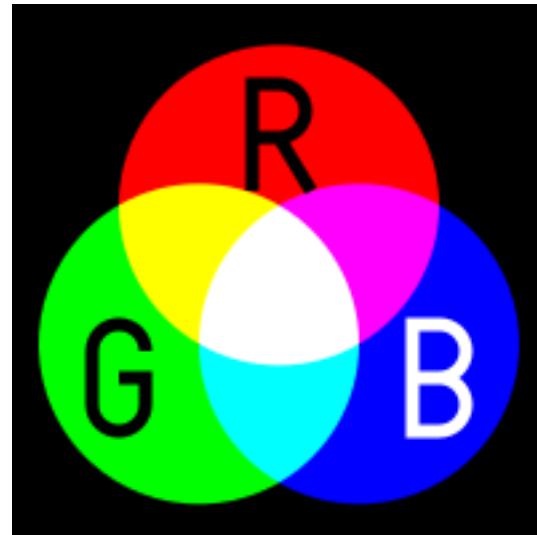
Diagram kromatisitas di samping ini mewakili Color Space. Ini menunjukkan segmen gelombang untuk nilai sumbu x dan y yang berbeda dengan berbagai panjang gelombang. Ruang membuat semua kemungkinan kombinasi warna terlihat oleh sistem penglihatan manusia. Ada banyak ruang warna seperti RGB, sRGB, dll. Kombinasi warna di bawah setiap ruang warna (ditandai dengan segitiga) menyimpan warna yang ditentukan.



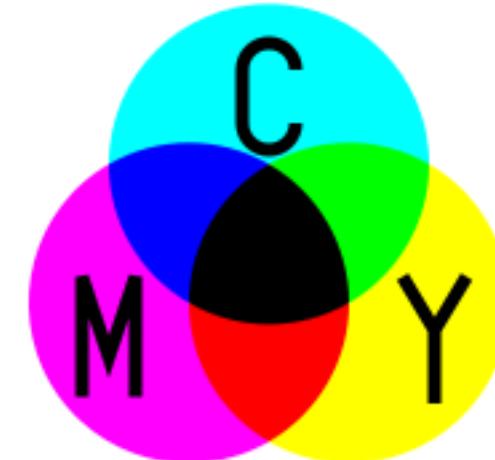
[CIE 1931 xy chromaticity diagram](#) (Under Wikimedia Commons free license CC BY-SA 3.0)

# Color Space

- Color addition vs subtraction
- Cara mudah untuk mengingat perbedaan pencampuran warna aditif dan subtraktif adalah **pencampuran warna aditif** terjadi ketika kita **mencampur lampu** dengan warna berbeda sedangkan **pencampuran warna subtraktif** terjadi ketika kita **mencampur cat** atau bahan berwarna lainnya.

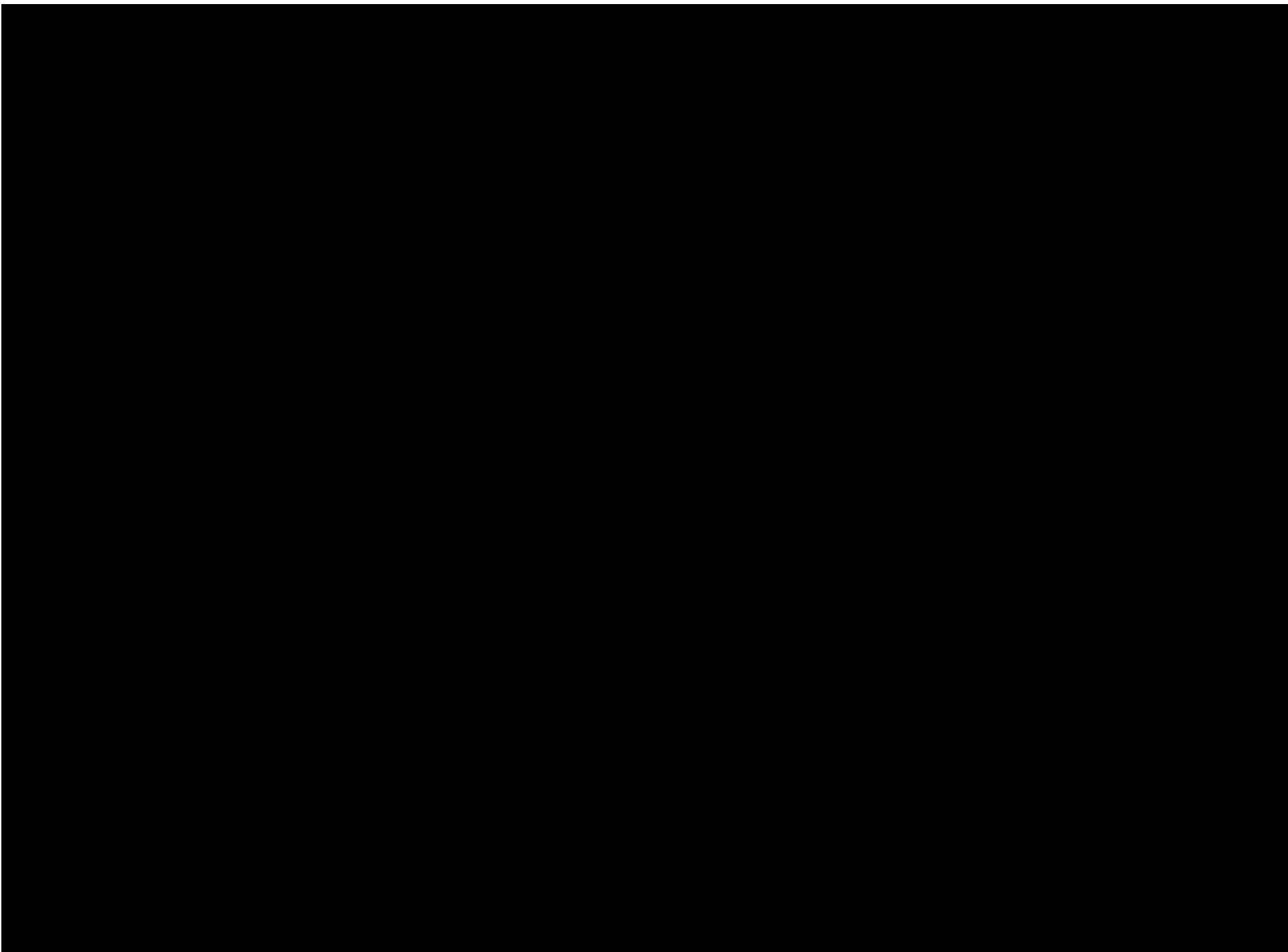


Color Addition



Color Subtraction





# Trichromatic Theory

- Young-Hemholtz: Teori trikromatik penglihatan warna adalah teori yang menyatakan ada **tiga reseptor warna yang berbeda di retina**. Sel kerucut di retina inilah yang memberi penglihatan warna. Teori ini mengatakan **sel kerucut** sensitif terhadap tiga warna berbeda: **hijau, biru, dan merah**

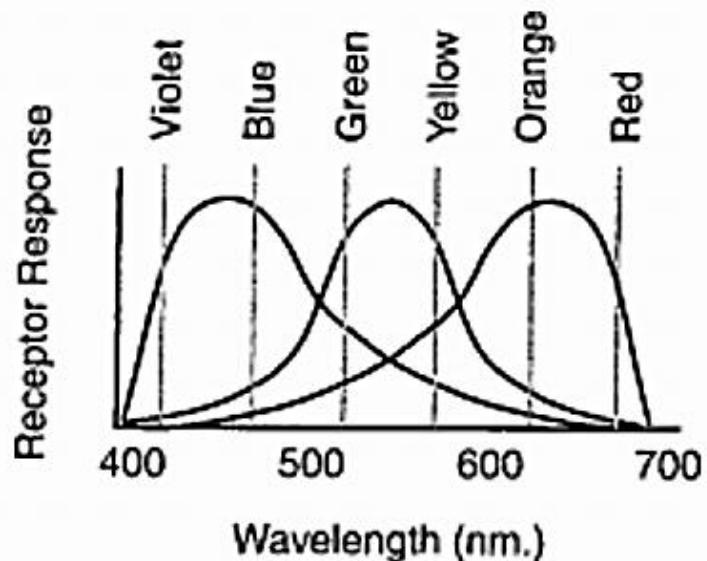


Figure 3.2.8  
Patterns of activation in trichromatic theory.  
Different color perceptions were proposed to result from different balances of activity in the three receptor systems, as illustrated here for six basic colors.

# Trichromatic Theory

# Opponent Process Theory

- Ewald Hering
- Warna komplementer: 2 warna bersebrangan, jika digabung akan saling cancel (hue)

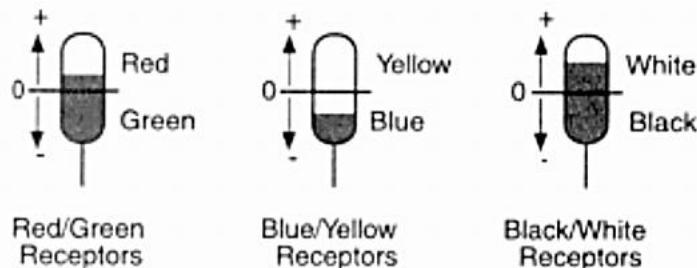
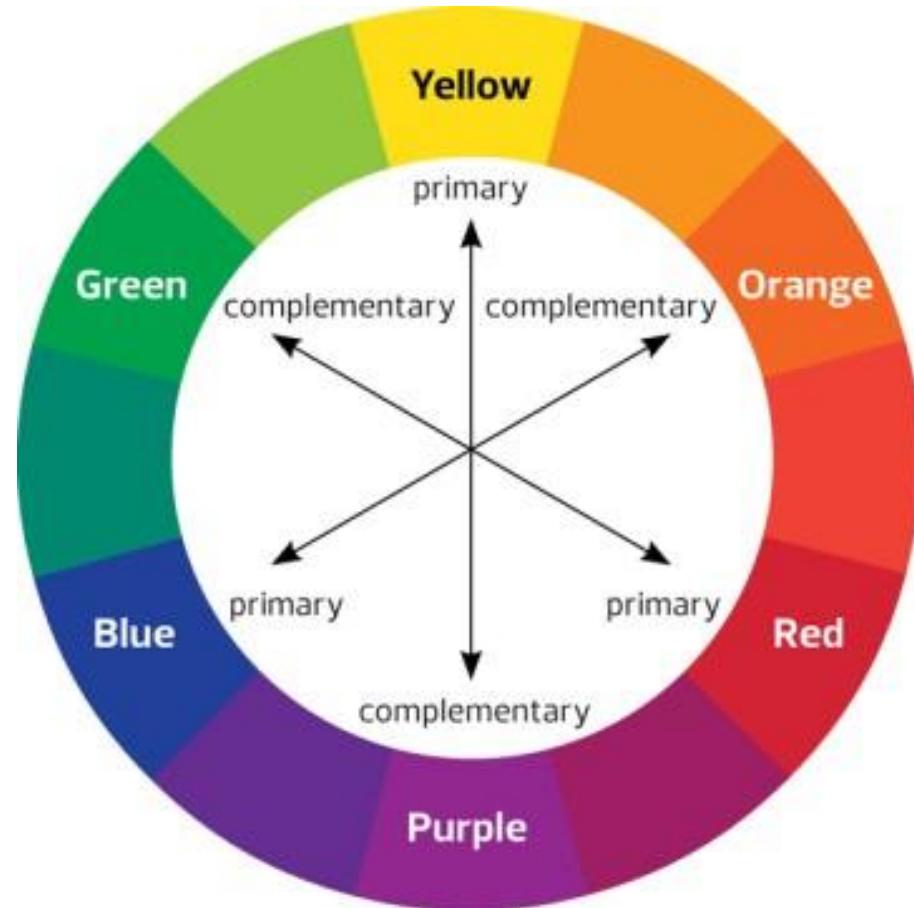


Figure 3.2.10

Hering's opponent process theory of color vision. Hering suggested that three types of receptors could respond in two opposite directions (+ versus -) from a neutral point (0) to signal red versus green, blue versus yellow, and black versus white.



# Opponent Process Theory

+

# Aplikasi warna komplementer



# Dual Process Theory

- Hurvich & Jameson

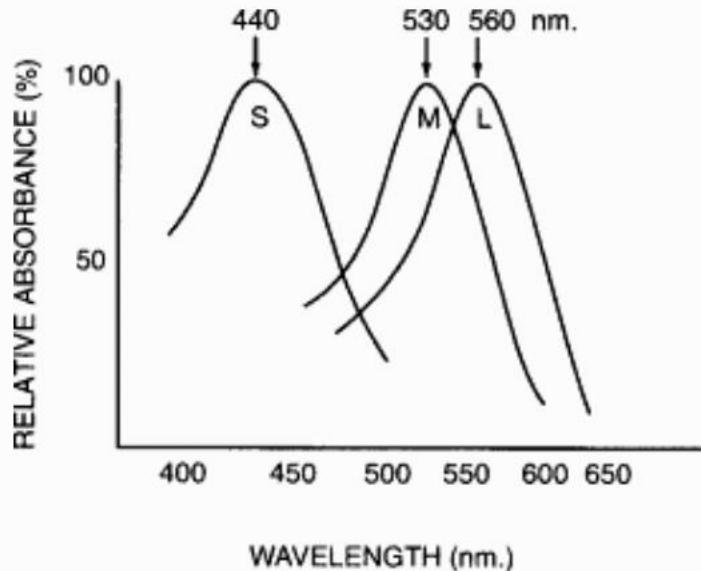
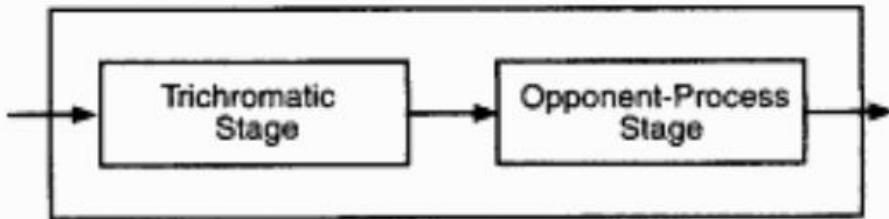
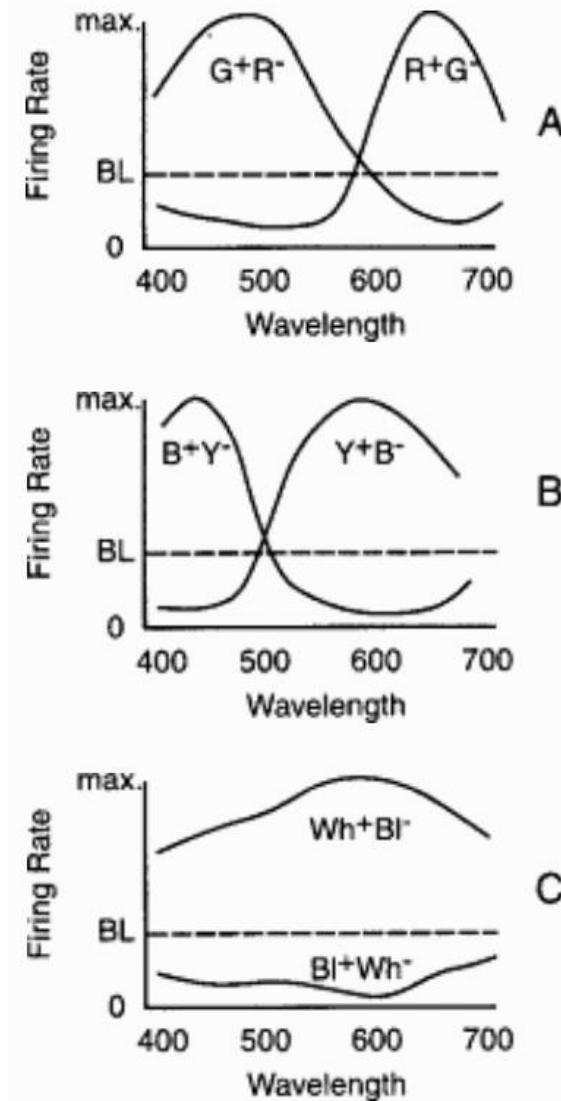


Figure 3.2.13

Neural response curves for the three types of cones. Short- (S), medium- (M), and long- (L) wavelength cones provide overlapping but differential responses to light of different wavelengths, as predicted by trichromatic theory. These curves are defined by the absorption spectra of the three pigments found in normal cones. (After Schnapf, Kraft, & Baylor, 1987.)

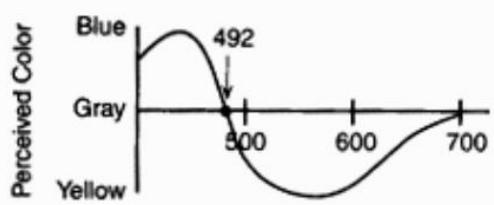


DUAL PROCESS THEORY

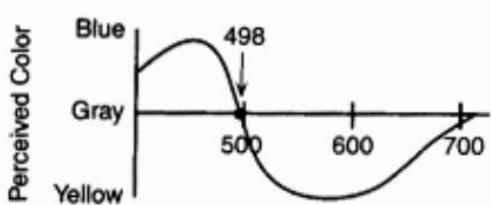


# Buta warna

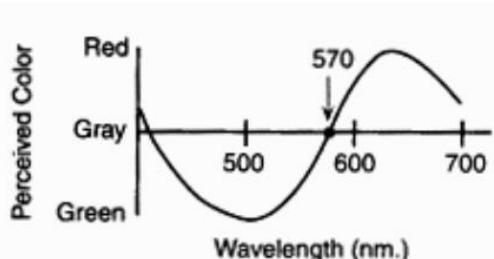
- Protanopia are unable to perceive any 'red' light
- Deuteranopia are unable to perceive 'green' light
- Tritanopia are unable to perceive 'blue' light.



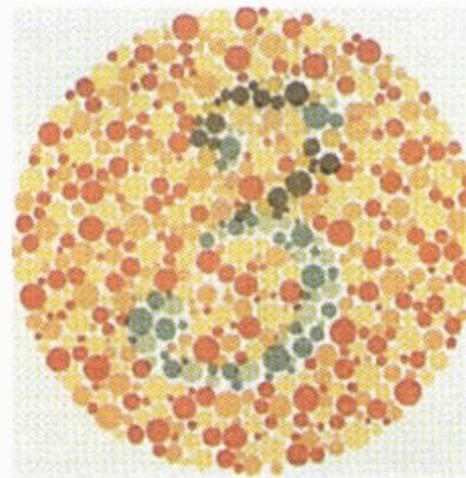
A. Protanopia



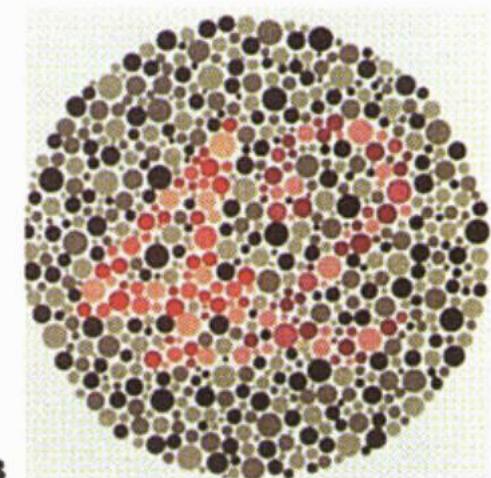
B. Deuteranopia



C. Tritanopia



A



B

## Plate 3.5

A test for color blindness. People with normal color vision see the number 3 in A and 42 in B, whereas protanopes see no number in A and a 2 in B, and deutanopes see no number in A and a 4 in B.  
(Courtesy of Graham-Field Surgical Company.)

# Lintasan cahaya

- Cahaya menempuh lintasan terpendek (geodesik)

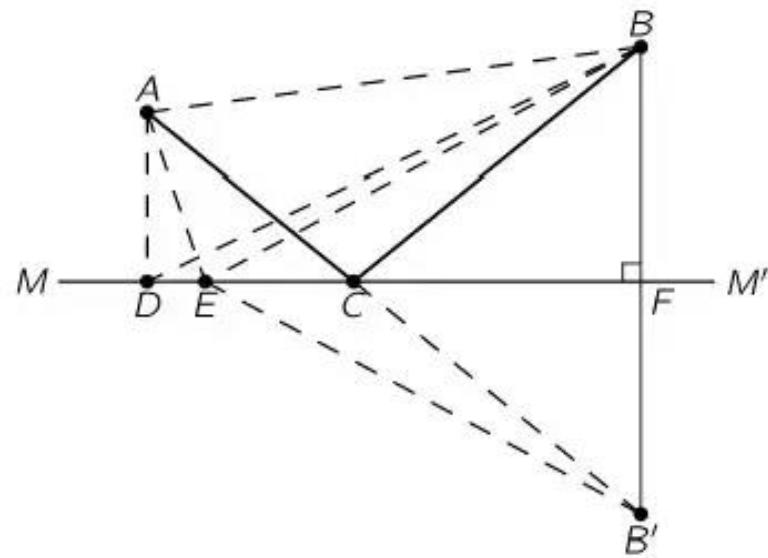
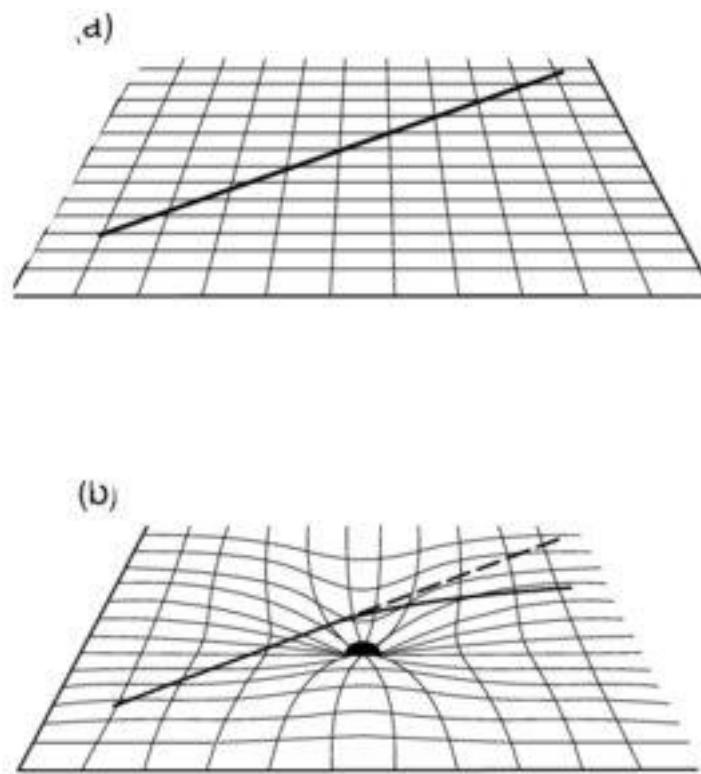
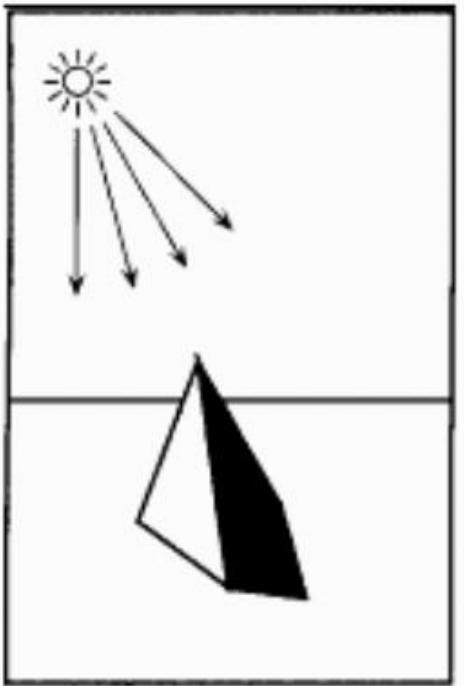


Illustration of the principle of least time.  
We construct on the other side of  $MM'$  an artificial point  $B'$ , which is the same distance below the plane  $MM'$  as the point  $B$  is above the plane.  
Then we draw the line  $EB'$ .

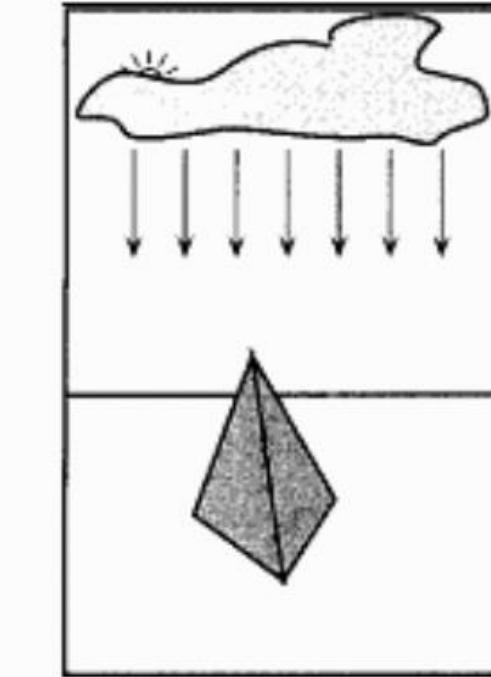
[http://www.feynmanlectures.caltech.edu/I\\_26.html](http://www.feynmanlectures.caltech.edu/I_26.html)



# Illuminasi



A. Point-Source Illumination



B. Spatially Extended Illumination

# Interaksi dengan permukaan objek

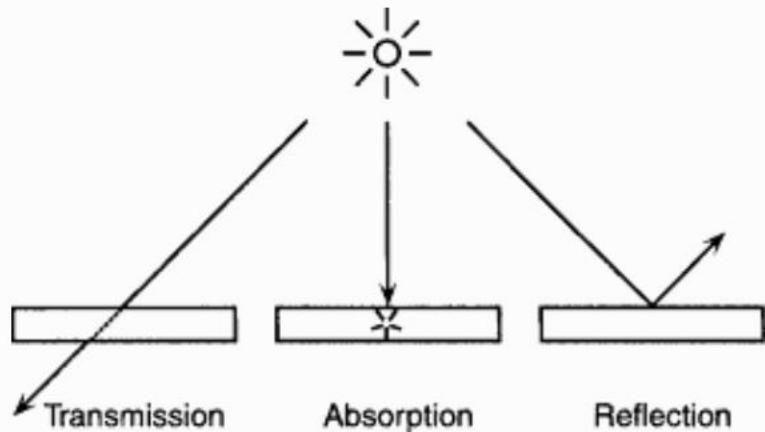


Figure 1.2.2

Interactions between light and surfaces. A photon can be transmitted through a surface, absorbed by it, or reflected off it.



Figure 1.2.3

Refraction of light. When light is transmitted through an object, it can be bent (refracted), leading to erroneous perceptions, such as the misaligned appearance of the spoon handle in this glass of water.

# Mata

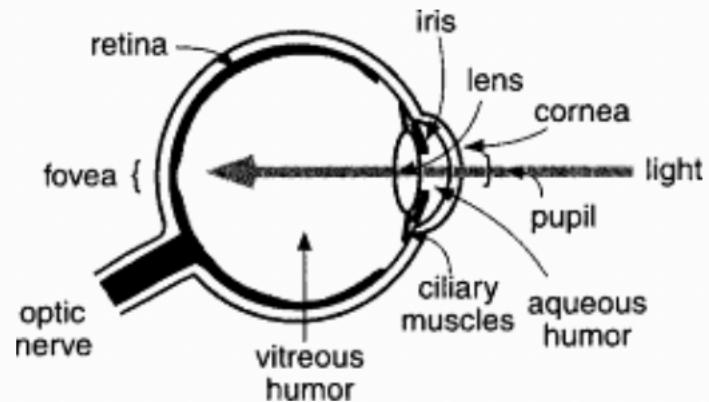
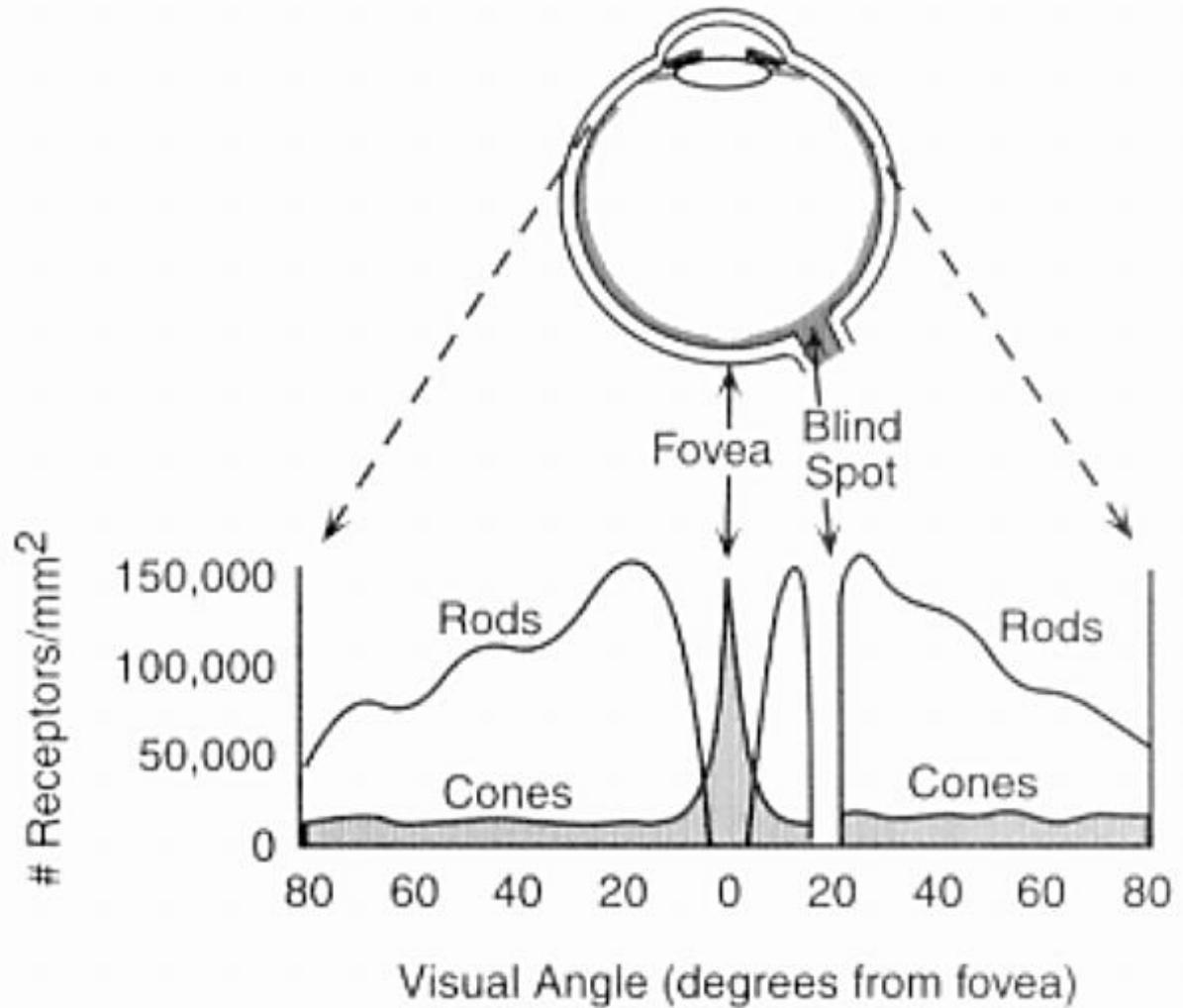


Figure 1.3.4

A cross section of the human eye. Light enters the eye through the cornea, aqueous humor, lens, and vitreous humor before striking the light-sensitive receptors of the retina, where light is converted into electrochemical signals that are carried to the brain via the optic nerve.



# Sistem syaraf

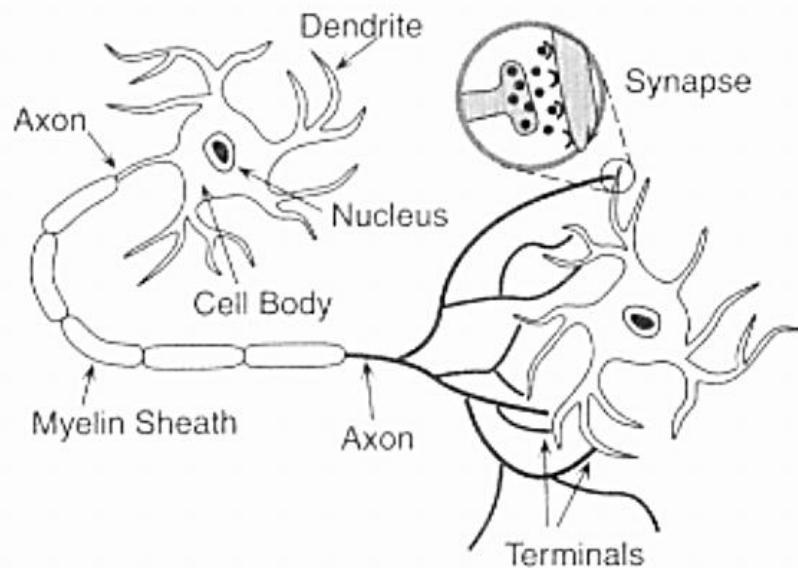
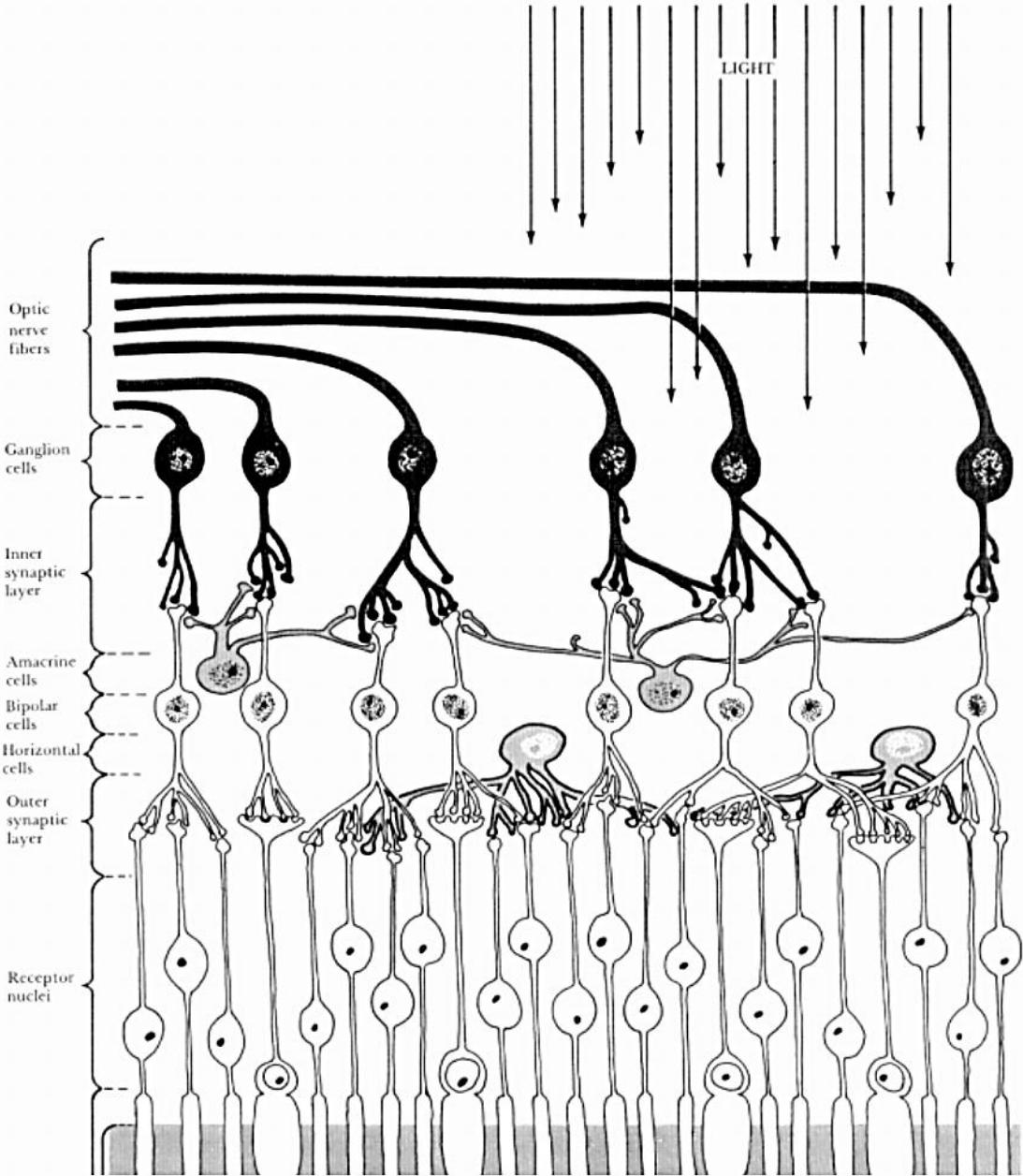


Figure 1.3.6

A typical neuron. A neuron is a cell that consists of a cell body that integrates graded electrical signals from its dendrites and transmits the result via discrete action potentials. These spikes travel along an axon, which is encased in a myelinated sheath, to terminals, where neurotransmitters are released at synapses to stimulate the dendrites of other neurons.



# Sistem visual manusia

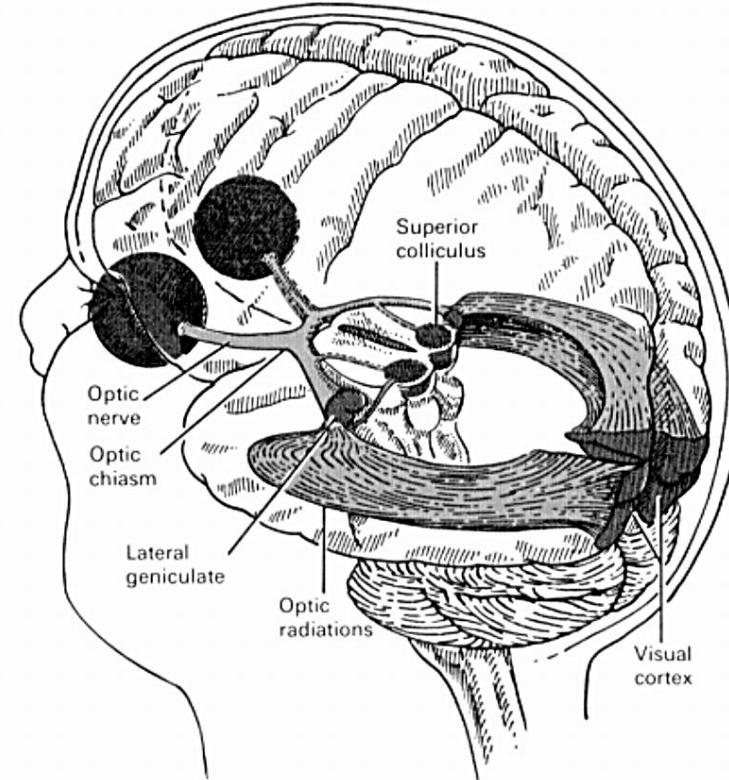
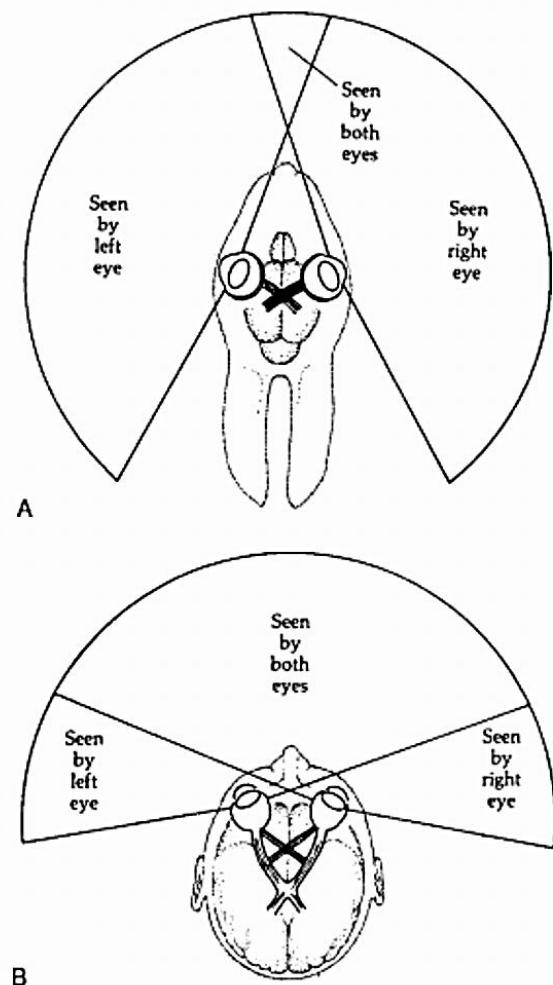


Figure 1.3.1

The human visual system. Visual processing begins in the eyes and is relayed to the brain by the optic nerve. The primary visual pathway then goes from the lateral geniculate nucleus to occipital cortex via the optic radiations. From there, visual information travels to other parts of the brain. A secondary pathway goes from the optic nerve to the superior colliculus and then to other brain centers. (From Rosenzweig & Leiman, 1982.)

# Classical Theory of Vision

THEORY NATIVISM vs.  
EMPIRICISM

Structural-Empiricism  
ism

Gestalt

Ecological  
Optics

ATOMISM vs.  
HOLISM

Atomism

Holism

Holism

ORGANISM vs.  
ENVIRONMENT

Organism

Organism

Environment

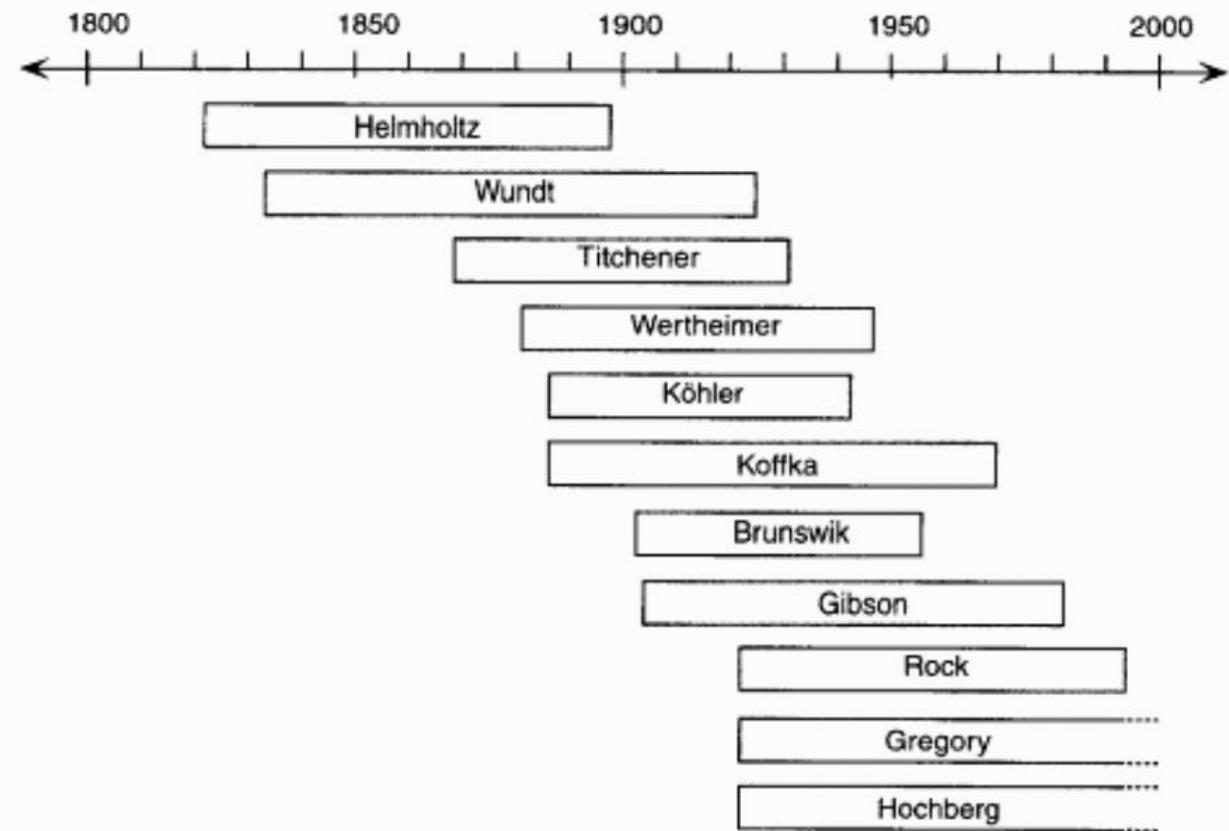


Figure 2.1.1

A timeline of visual theorists. The life spans of some of the most important psychological theorists of vision are represented over the past two centuries. Wundt and Titchener were structuralists; Wertheimer, Köhler, and Koffka were Gestaltists; Brunswick and Gibson were ecologists; and Helmholtz, Rock, Gregory, and Hochberg were constructivists.

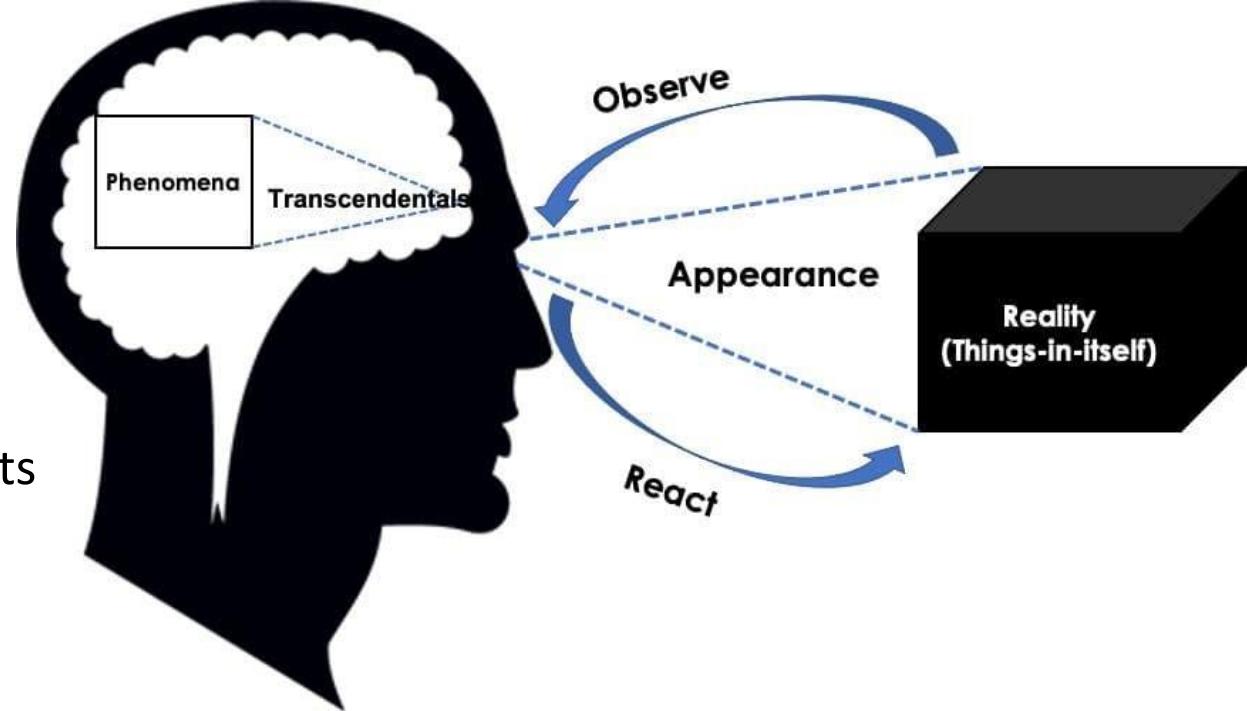
# German Idealism

- Immanuel Kant: Phenomena, Noumena
- Gestaltism/holism: Wertheimer
  - The whole is different from the sum of its parts



Figure 2.1.5

Emergent properties of a configuration. The arrangement of several dots in a line gives rise to emergent properties, such as length, orientation, and curvature, that are different from the properties of the dots that compose it.



# Phenomenology

- Edmund Husserl: Natural Experience
- Ecological Optics: James Gibson
  - “Ask not what's inside your head, but what your head's inside of”
  - foreshadow modern work in computational vision
- Constructivism: Helmholtz
  - global percepts are constructed from local information

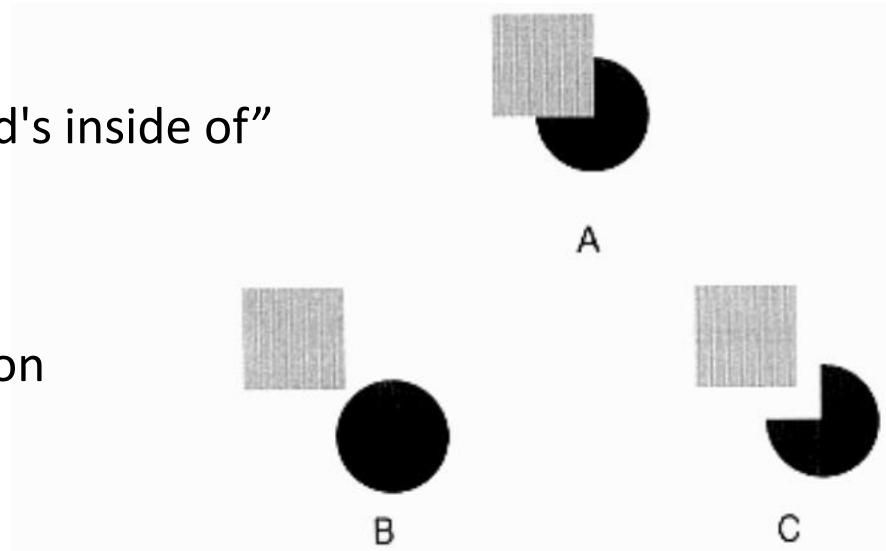
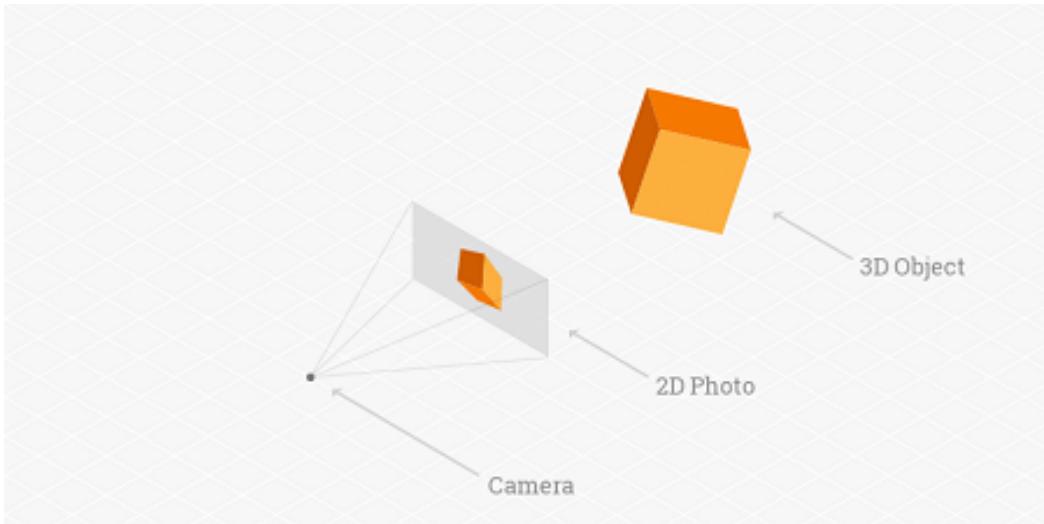


Figure 2.1.12  
Two possible interpretations of a simple image. Image A can be interpreted either as a circle behind a square (interpretation B) or as a three-quarter circle next to a square (interpretation C). The fact that the visual system strongly prefers interpretation B to C can be explained by the principles of likelihood and/or Prägnanz.

# Reformational Philosophy

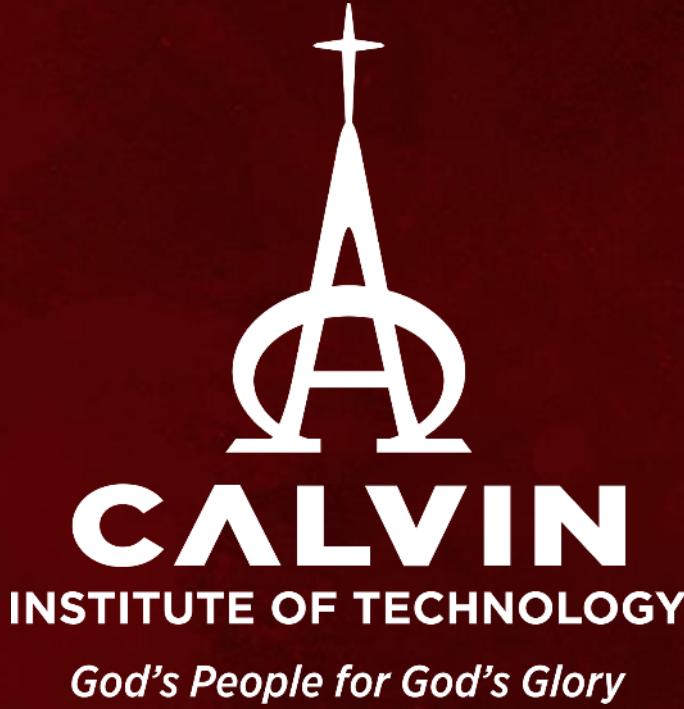
- Realita diciptakan dalam keragaman aspek → setiap aspek ciptaan bukan Pencipta
- Pre-theoretical vs theoretical thought:
  - Pre-theoretical thinking takes in reality as a whole
  - Theoretical thinking analyses reality from one aspect.
- Example:
  - a gardener may look at a plant in its unity as a plant; whereas a biologist may analyse the plant in terms of its biotic aspect.
  - When we see a sunset, we admire it as a sunset rather than in terms of the movement of the earth on its axis and its movement around the sun.

# Summary

- Silabus
- Instalasi
- Visi: fondasi dan filsafat

# Tuhan Memberkati





# Pemrosesan Gambar

Ghandy, M.Sc.

IBDA4311 – *Computer Vision*

# Capaian Pembelajaran

- Pengantar Paket Python

1. Numpy
2. PyTorch
3. Matplotlib
4. Opencv

- Pemrosesan Gambar

- Membaca gambar
- Channel dan format
- Operasi dasar
- Menulis Gambar
- Video
- Thresholding
- Blurring

# Pengantar Paket Python

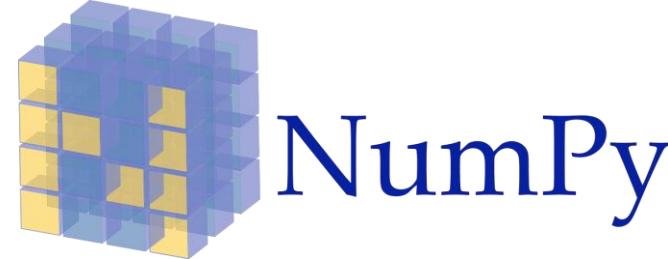
# Python



- Python adalah Bahasa pemrogrammang yang sifatnya high-level
- Python code hampir seperti pseudocode (bisa mengeskpresikan ide dengan bebas)
- Python yang akan digunakan adalah python 3.9



# NumPy



- NumPy adalah paket untuk melakukan perhitungan numerik (vs Scipy lebih untuk matematika sains)
- Array berbentuk objek
- Sangat berguna untuk problem linear algebra

# Array and Matrices

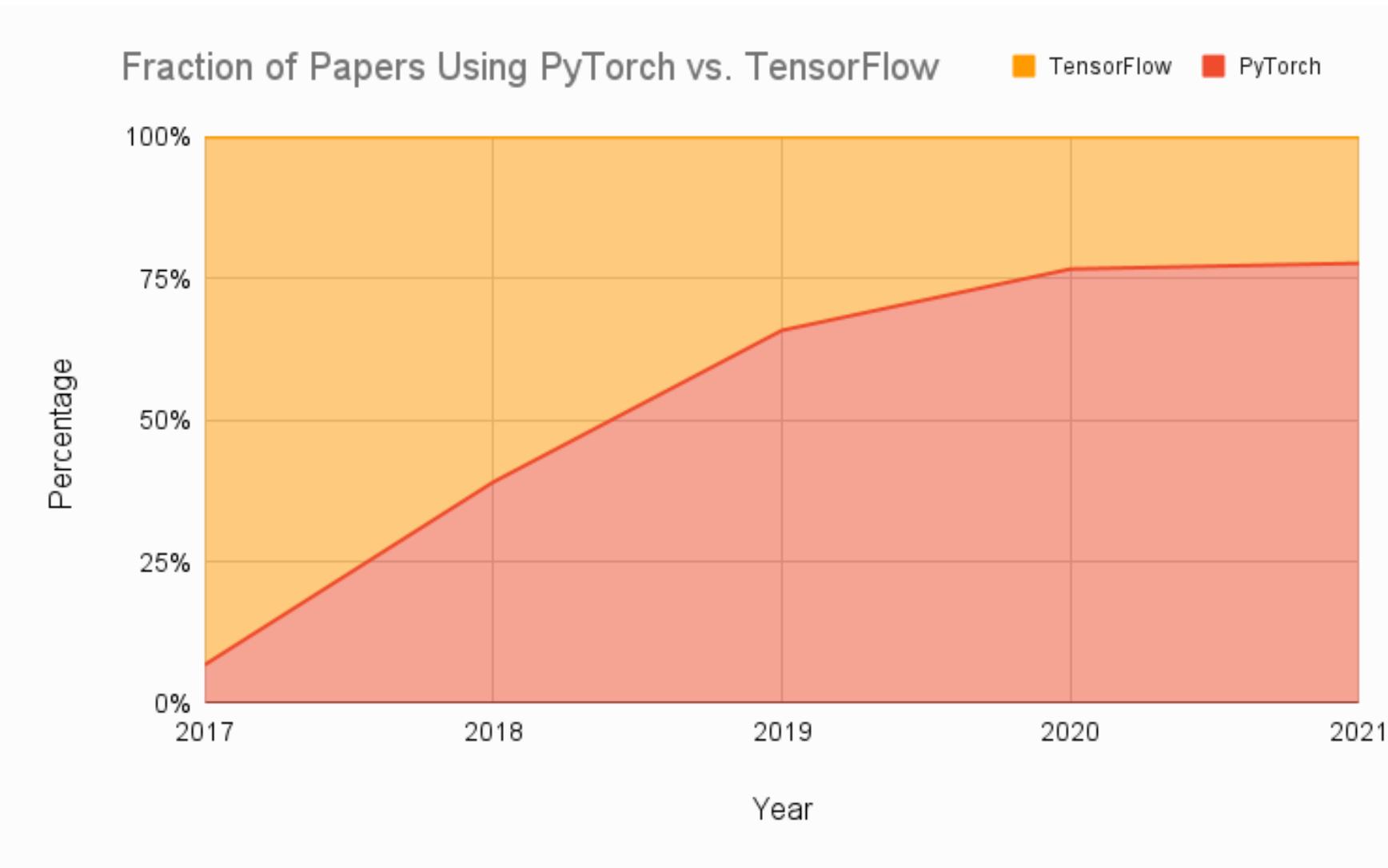
- Array di numpy (np.array) merupakan kumpulan daftar elemen
- Setiap array harus memiliki tipe yang sama
- Beberapa datatype:
  - Float (float16, float32, float64)
  - Int (int8, int16, int32, int64)
  - Uint
  - Bool
  - Complex
- Matrices adalah array yang memiliki 2 jenis dimensi
- Jika memiliki lebih dari 2 jenis dimensi disebut Tensor

# Operasi Matriks

- Keunggulan numpy terletak pada operasi matriks
- Operasi matematika dasar (+,-,/,\*,%) dianggap sebagai operasi elemen
- Operasi matriks yang umum dapat menggunakan:
  - Matrix multiplication: np.matmul
  - Transpose: property objek numpy .T
  - Inverse: np.linalg.inv
  - Penjumlahan sepanjang axis: np.sum
  - Menumpuk 2 array: np.stack
  - Menyambung 2 array: np.concatenate

- PyTorch adalah paket untuk deeplearning dan optimisasi operasi tensor
- Mengapa numpy dan opencv tidak cukup?
  - Karena PyTorch dapat menggunakan GPU untuk operasi tensor
  - Karena PyTorch dapat melakukan turunan parsial otomatis
- Ada beberapa deep learning frameworks yang populer:
  - TensorFlow (google) → compatible dengan KERAS (banyak digunakan di industri)
  - PyTorch (facebook) → lebih mudah untuk mengutak-atik tensornya (banyak digunakan untuk riset)

# PyTorch vs TensorFlow



# Matplotlib



- Matplotlib adalah paket untuk menghasilkan grafik dengan fleksibel
- Terdapat beberapa toolkit tambahan yang bisa dibangun diatas matplotlib:
  - Basemap dan cartopy: untuk proyeksi dan mapping
  - Bokeh: untuk plot interaktif pada browser
  - Seaborn: untuk high level interface visualization

# OpenCV



- OpenCV (open source computer vision library) adalah paket untuk aplikasi computer vision
- Memiliki berbagai macam interfaces (c++, python, java, matlab)
- Mendukung berbagai macem OS (windows, linux, mac, android)
- Dirancang untuk aplikasi real-time

# Pemrosesan Gambar

# Membaca gambar

- Membaca terhadap filepath tertentu dapat menggunakan cv2.imread
- Gambar pada opencv akan direpresentasikan dalam bentuk numpy array

# Channel dan format

- Shape dari gambar tersebut berbentuk (tinggi, lebar, warna BGR)
- Ini dikarenakan opencv memperlakukan gambar seperti rows (tinggi) dan column (lebar) pada numpy
- Konvensi ini mungkin akan berbeda dengan kesepakatan paket tensor lain!!!
- Warna apapun dikonstruksi sebagai kombinasi 3 warna dasar (subjective color) dengan urutan: blue, green, red

# Menampilkan gambar

- Untuk menampilkan gambar dapat menggunakan cv2.imshow (tidak berlaku untuk jupyter)
- Untuk jupyter gunakan plt.imshow

# Operasi dasar

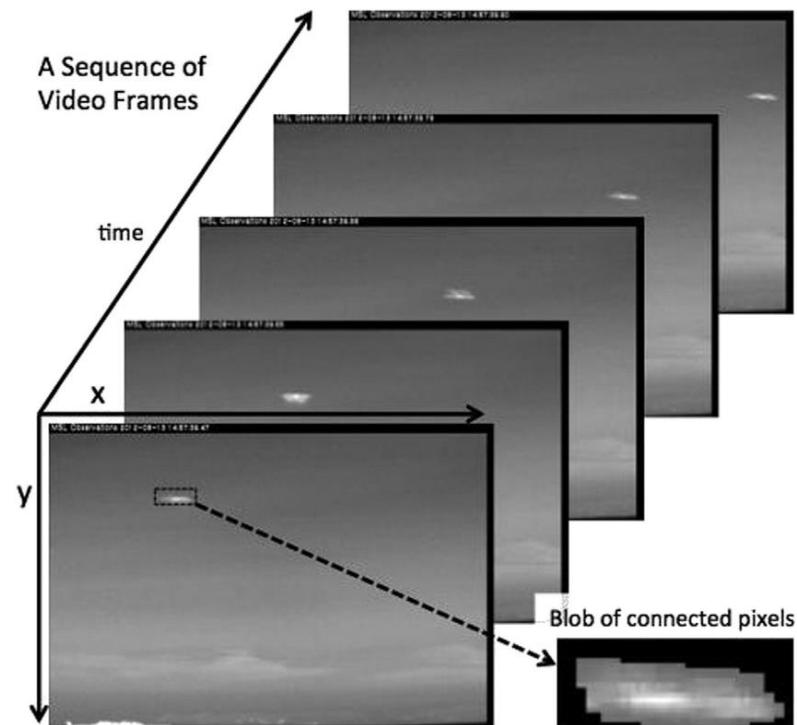
- Untuk mengubah gambar berwarna menjadi abu-abu gunakan cvtColor
- Grayscale tidak mempunyai dimensi warna, namun setiap pixel memiliki luminosity diantara 0 dan 255
- Untuk mengubah ukuran gambar gunakan resize (tapi kali ini menggunakan konvensi lebar lalu tinggi)

# Menulis Gambar

- Gambar dapat ditulis kedalam sebuah file atau variable dengan imwrite

# Video

- Video sebenarnya hanyalah sebuah urutan-urutan gambar
- Untuk membaca video dari webcams, IP cameras, atau file, kita dapat menggunakan VideoCapture
- Untuk menuliskannya dalam file, kita dapat menggunakan VideoWriter

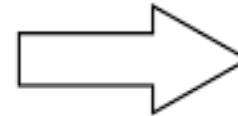


# Thresholding

- Binary thresholding memaksa gambar menjadi 2 nilai.
- Binary thresholding dapat digunakan untuk memperoleh mask.

104	223	50
12	255	0
46	120	100

$T=100$

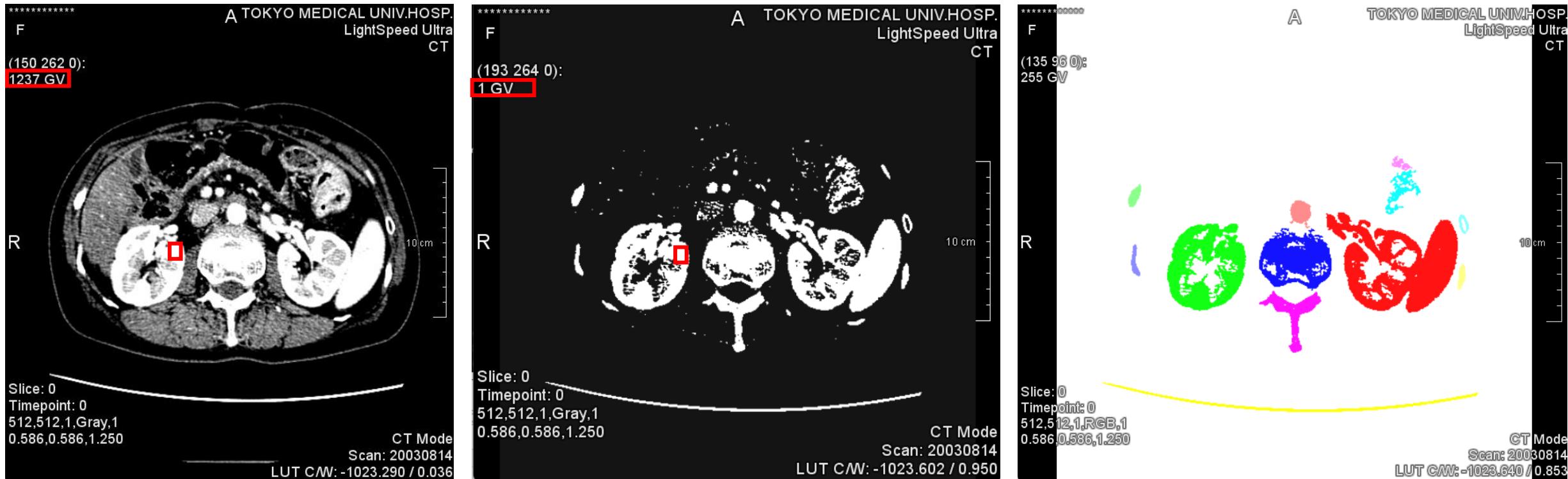


1	1	0
0	1	0
0	1	1



# Thresholding

- Thresholding juga dapat digunakan untuk keperluan segmentasi



# Gaussian Blur

- Fungsi gaussian dapat dipakai untuk merata-rata suatu posisi dengan nilai terdekatnya
- Ini akan menghasilkan efek blur pada gambar

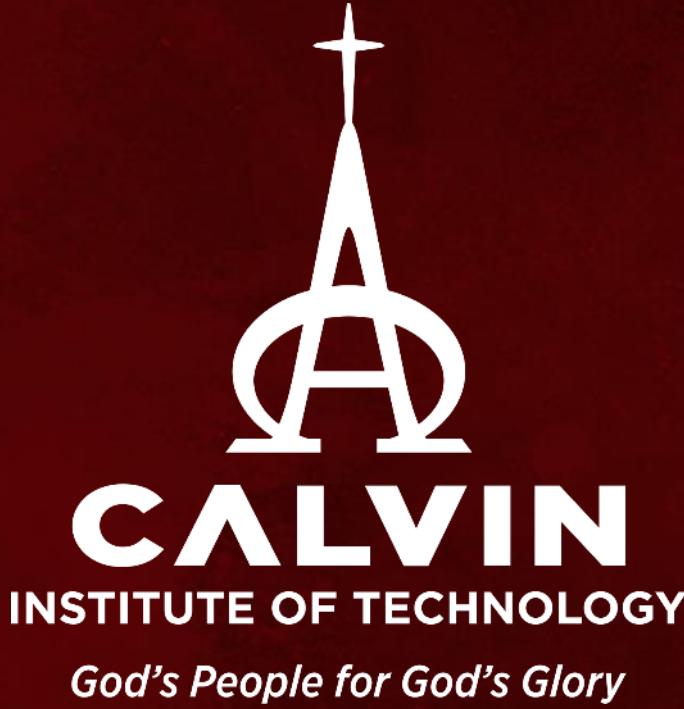


Original image



Gaussian Blur filter applied

# Summary



# Deteksi Batas dan Garis

IBDA4311 – *Computer Vision*

# Capaian Pembelajaran

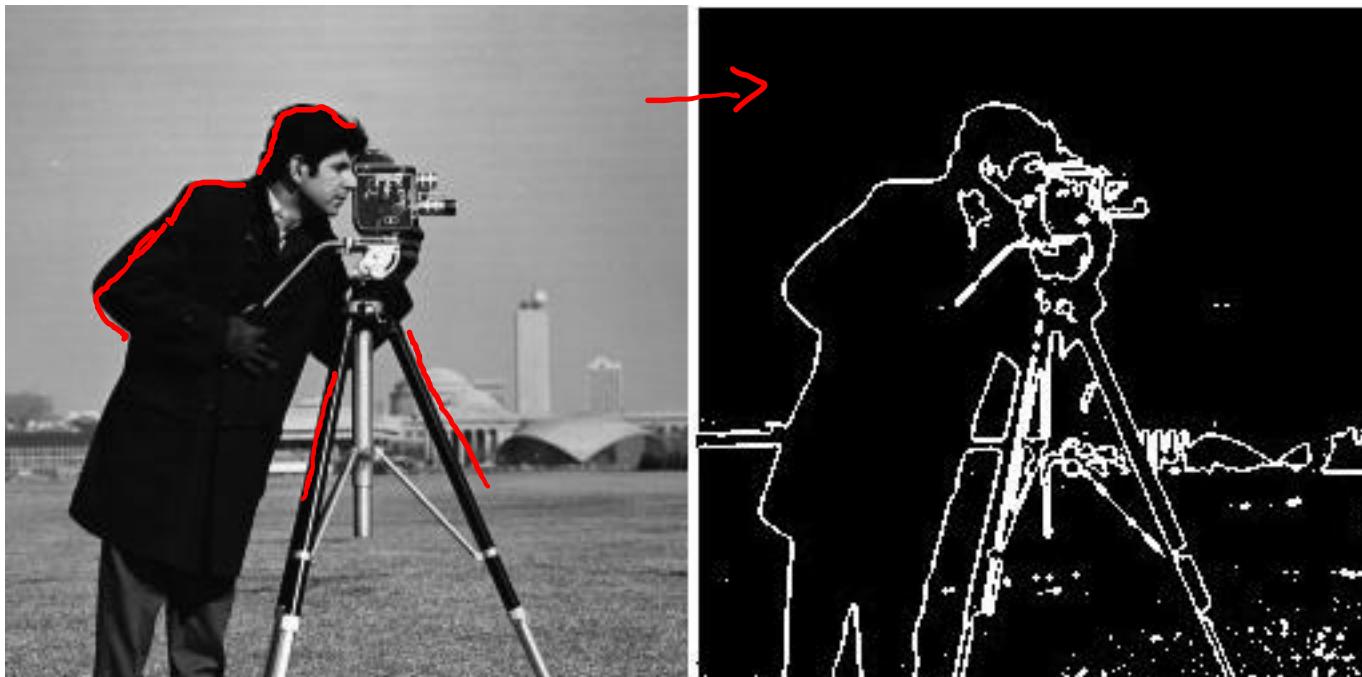
- Deteksi Batas
- Deteksi Garis

Edge Detection  
line Detection

# Deteksi Batas

# Garis Batas

- Sistem visual manusia berdasarkan garis batas
- Kita dapat mengidentifikasi sebuah objek dengan hanya menggunakan garis batas

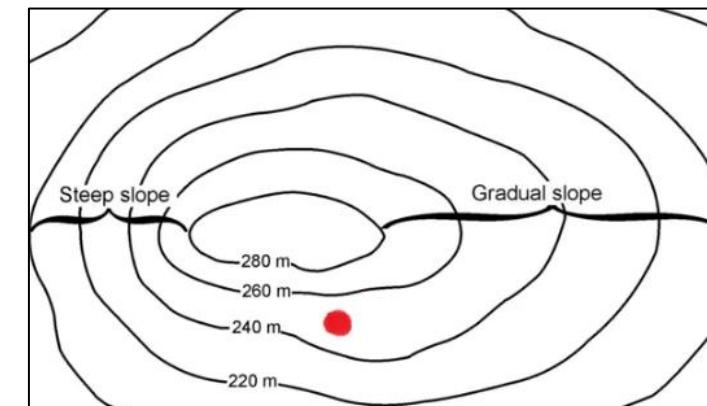
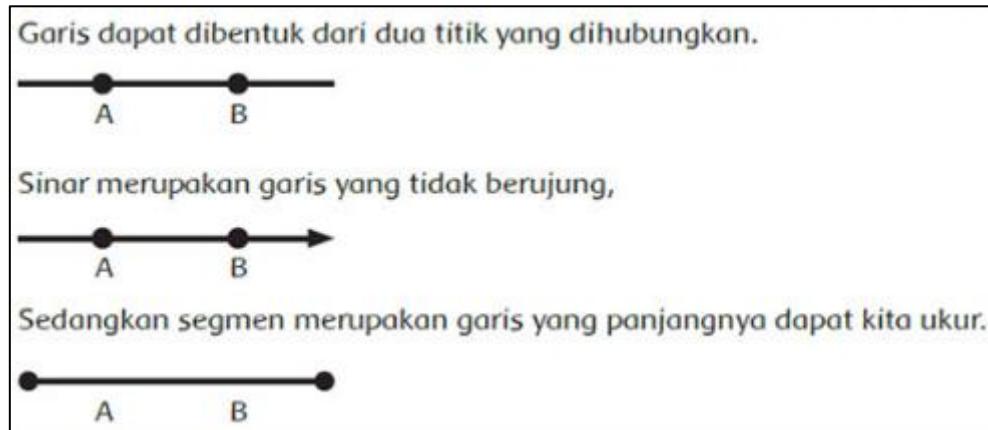


# Tujuan Deteksi Batas

*Tujuan*

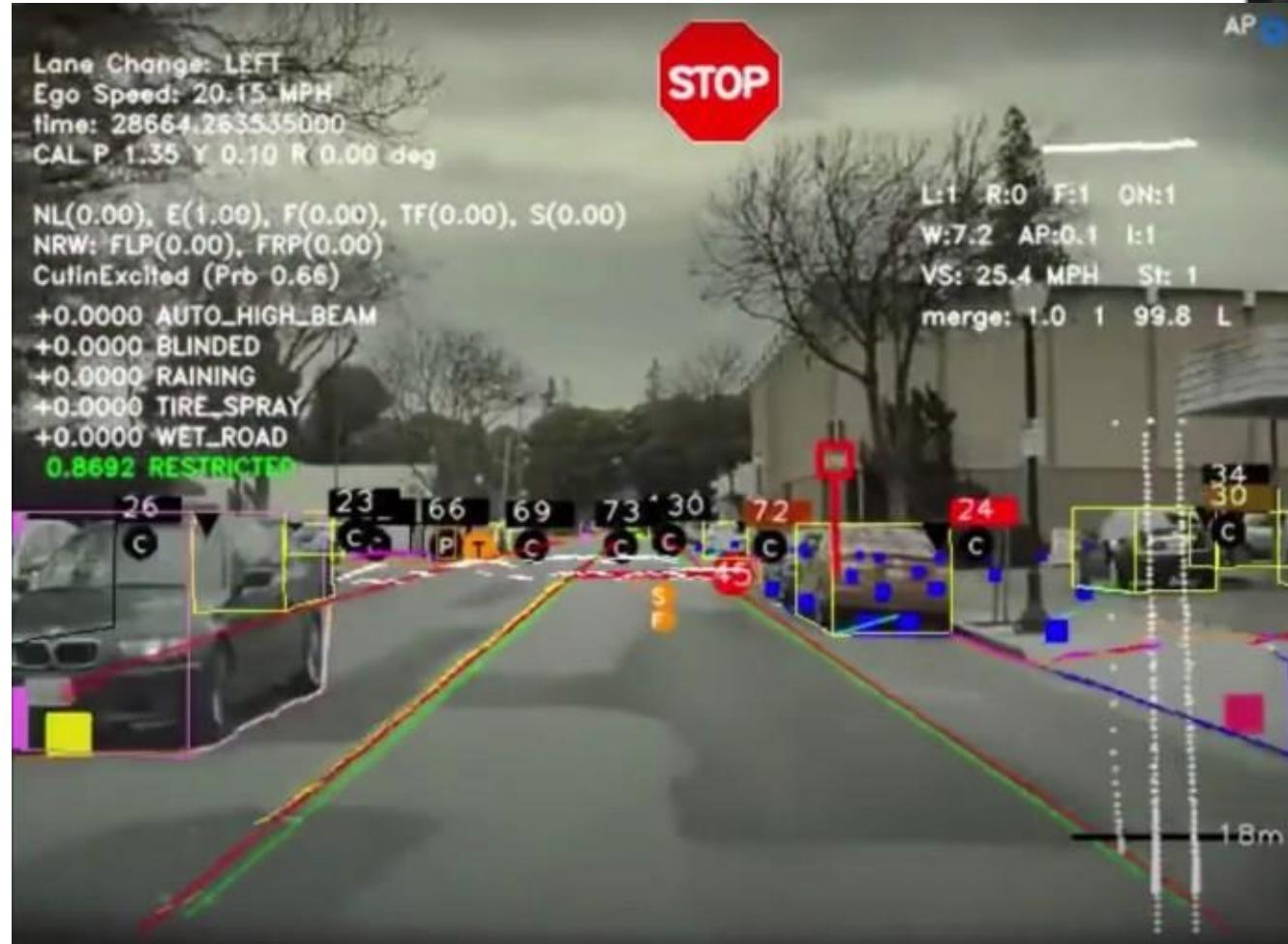
*Curvk*

1. Memetakan gambar (kumpulan pixel 2D) menjadi **himpunan kurva, segmen garis, atau kontur**
2. Representasi lebih padat dari sekedar gambar yang lengkap

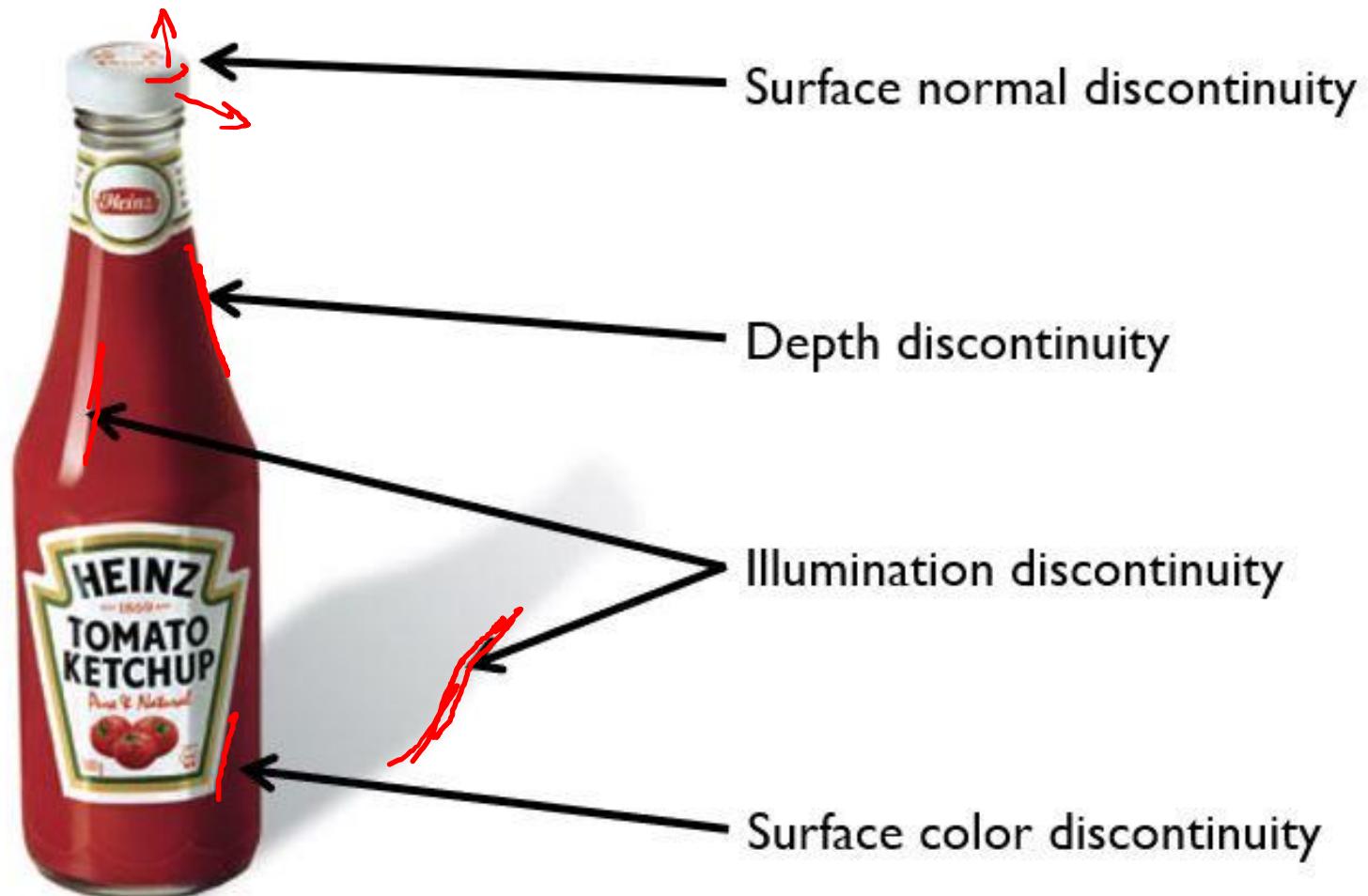


# Aplikasi Modern

1. Lane Detection (self driving car)
2. Sistem Navigasi Helikopter



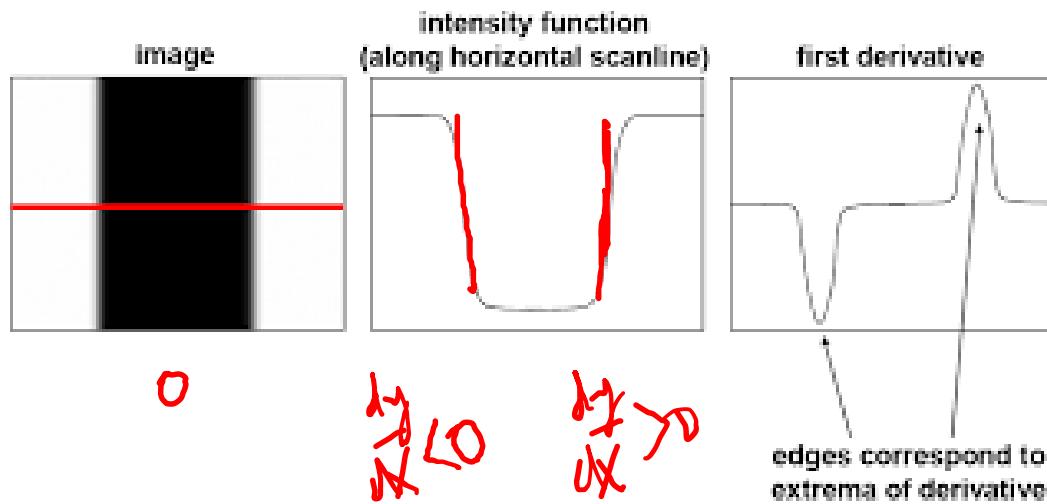
# Apa saja yang menyebabkan batas



# Turunan dan Batas

edge

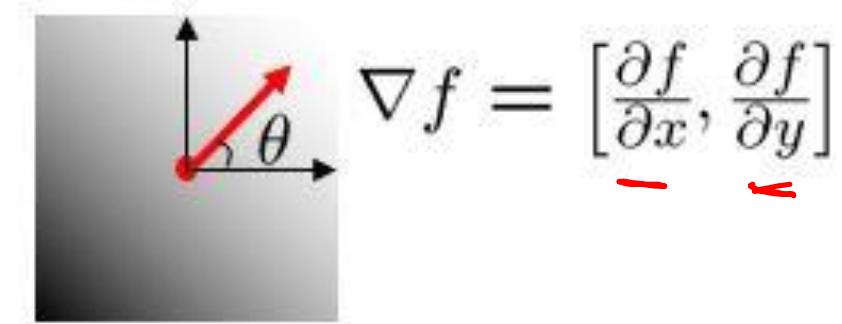
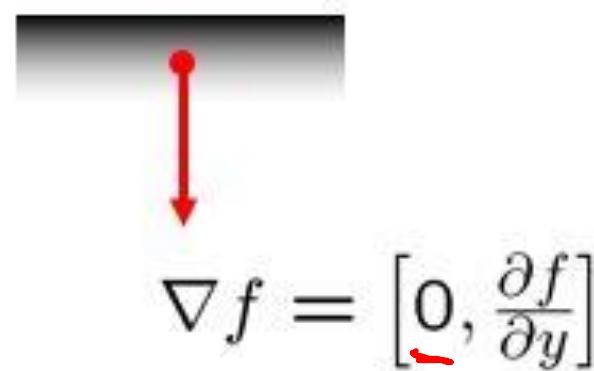
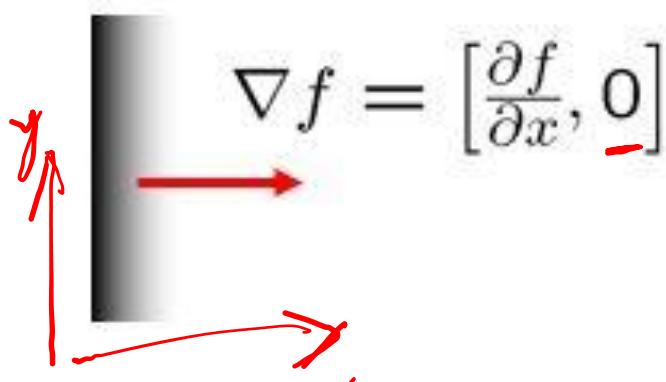
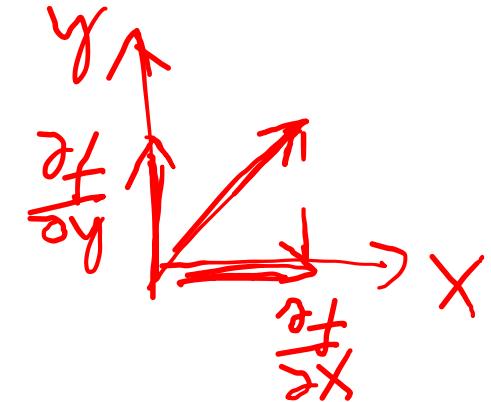
- Sebuah batas adalah tempat terjadinya perubahan drastis pada intensitas gambar



# Gradien Gambar

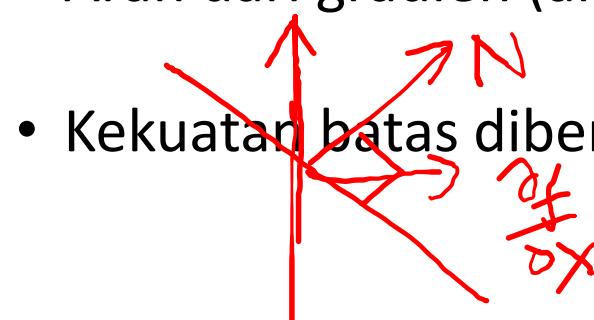
- Gradien dari sebuah gambar:  $\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$
- Gradien mengarah pada perubahan intensitas yang paling besar

→ turunan parsial



- Arah dari gradien (arah normal dari batas) dideskripsikan oleh:  $\theta = \tan^{-1} \left( \frac{\partial f}{\partial x} / \frac{\partial f}{\partial y} \right)$

- Kekuatan batas diberikan oleh nilai dari gradien  $||\nabla f|| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}$



# Turunan dan Konvolusi

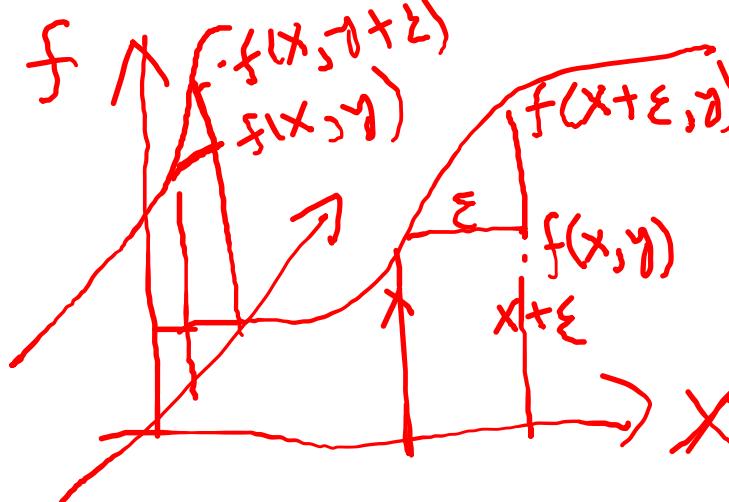
- Untuk fungsi 2D,  $f(x, y)$ , turunan parsial adalah:

$$\frac{\partial f(x, y)}{\partial x} = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon, y) - f(x, y)}{\epsilon}$$

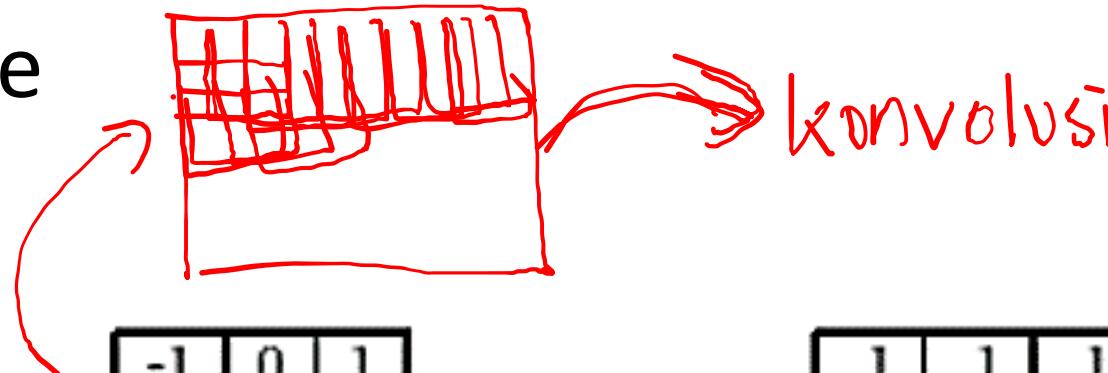
- Untuk data diskrit, kita dapat menggunakan finite difference:

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

- Untuk mengimplementasikannya sebagai konvolusi, filter apakah yang sesuai?



# Filter finite difference



Prewitt:  $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$  ;  $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

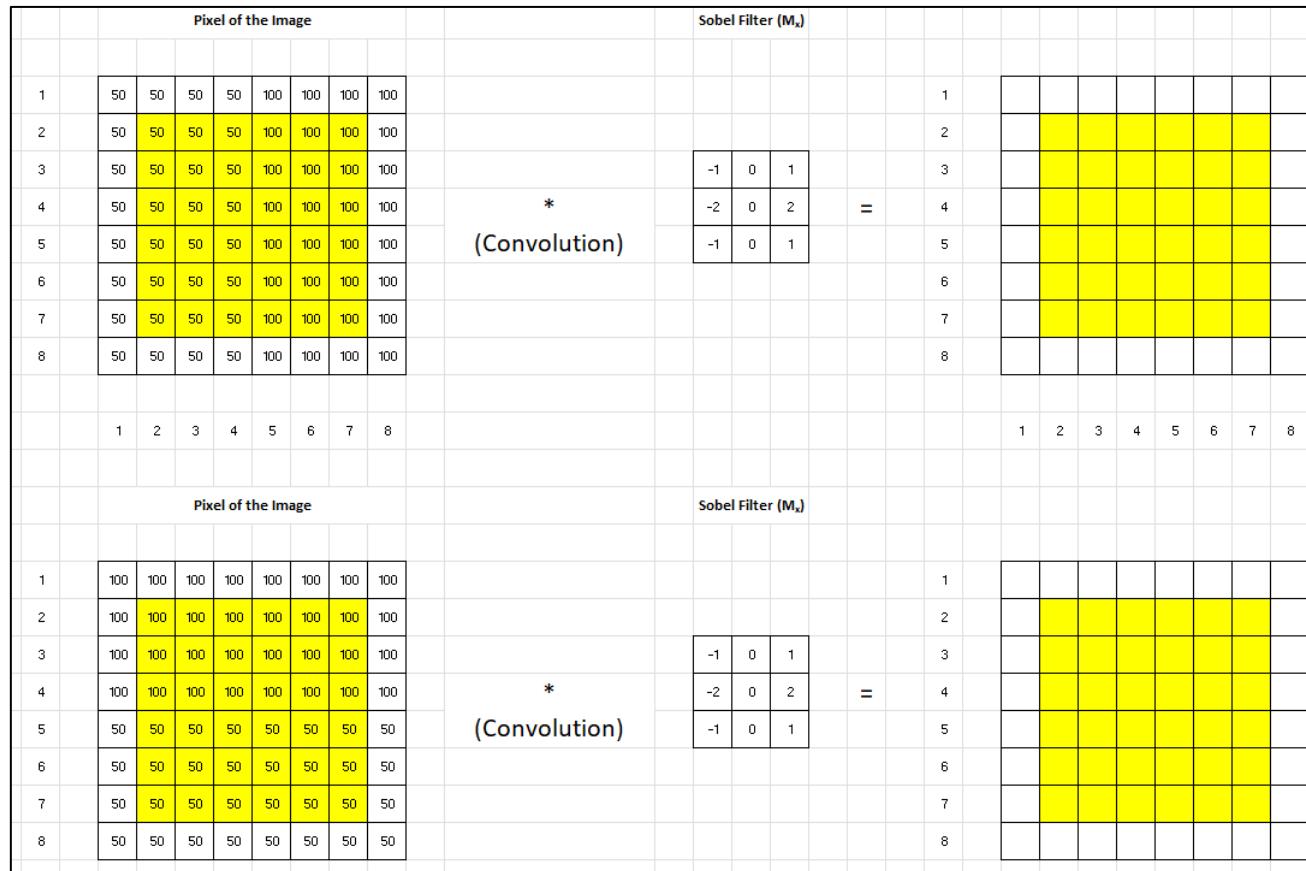
Sobel:  $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$  ;  $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Roberts:  $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$  ;  $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$



# Soal Latihan

- Kerjakanlah soal di dalam file SobelFilter.xlsx untuk lebih memahami mengenai konvolusi dan cara kerja filter sobel.

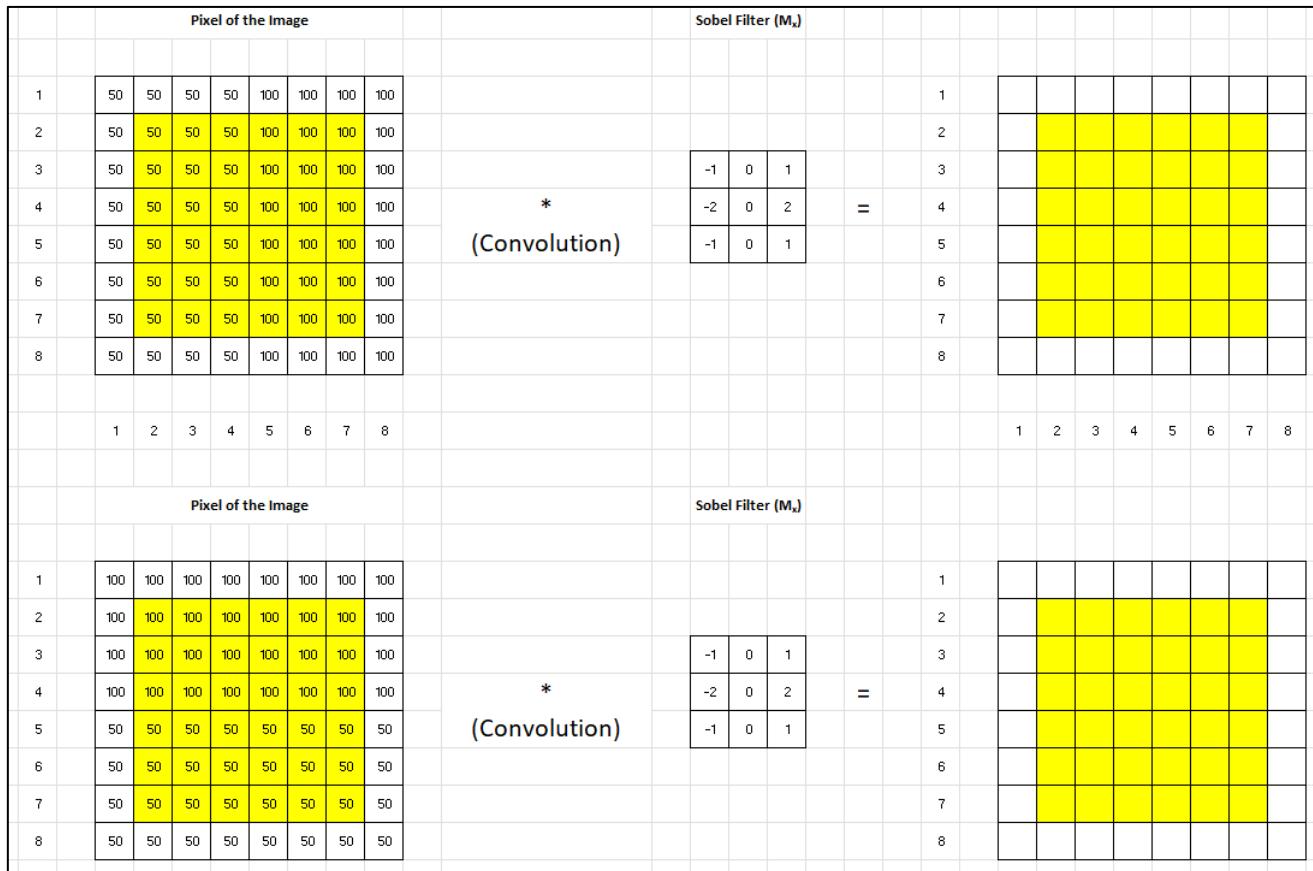


**Sobel:**

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$
$$M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

# Jawaban Latihan

- 1 orang mahasiswa/i mempresentasikan hasilnya



**Sobel:**

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

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

# Uji Pemahaman

- Batas apakah yang ditangkap filter-filter sebelumnya?

Prewitt:

$$M_x =$$

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

$$; \quad M_y =$$

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

Sobel:

$$M_x =$$

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

$$; \quad M_y =$$

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

Roberts:

$$M_x =$$

0	1
-1	0

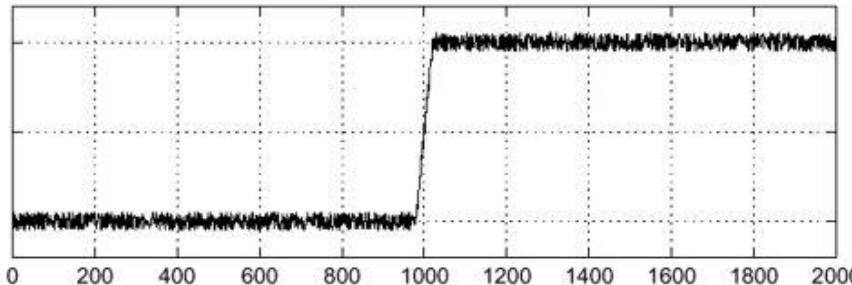
$$; \quad M_y =$$

1	0
0	-1

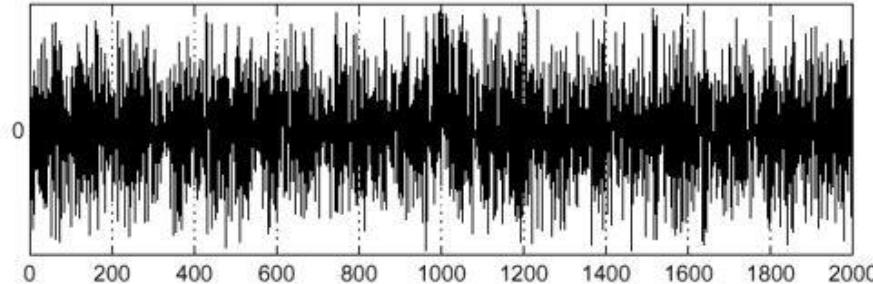
# Noise

- Misalkan kita mengambil satu baris dari gambar, lalu plot intensitas sebagai fungsi dari posisi:

$$f(x)$$



$$\frac{d}{dx} f(x)$$



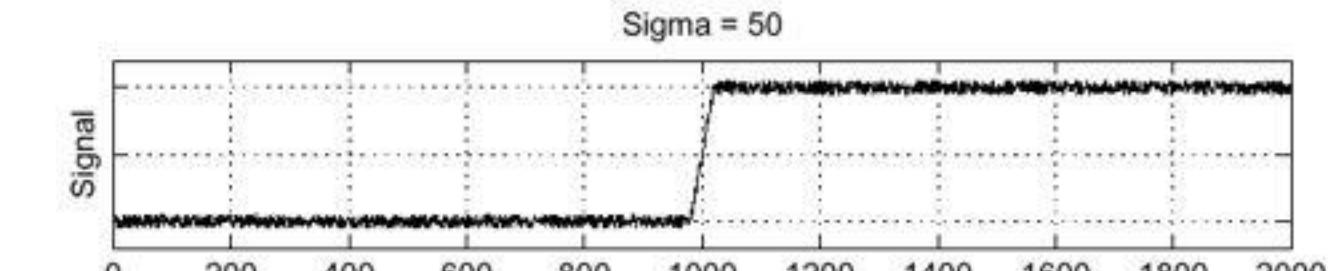
Inilah hasil turunan dari  $f(x)$  ->  
Dimanakah edge/batasnya?

- Finite difference berespon terlalu kuat terhadap noise (image noise berbentuk pixel yang sangat berbeda dengan pixel tetangganya)
- Apa yang harus dilakukan?

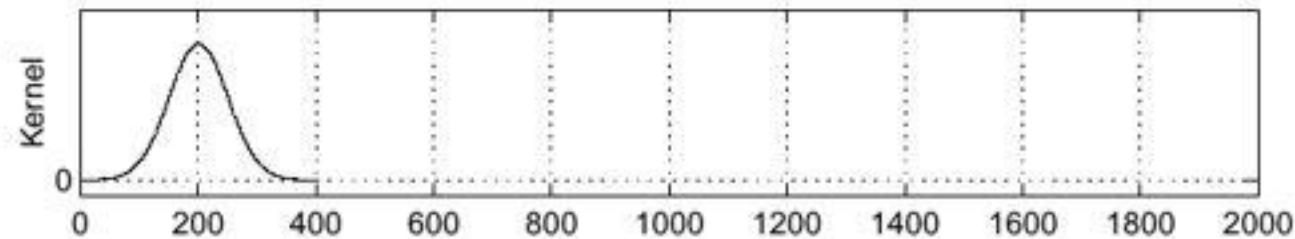
# Smoothing

2 Step

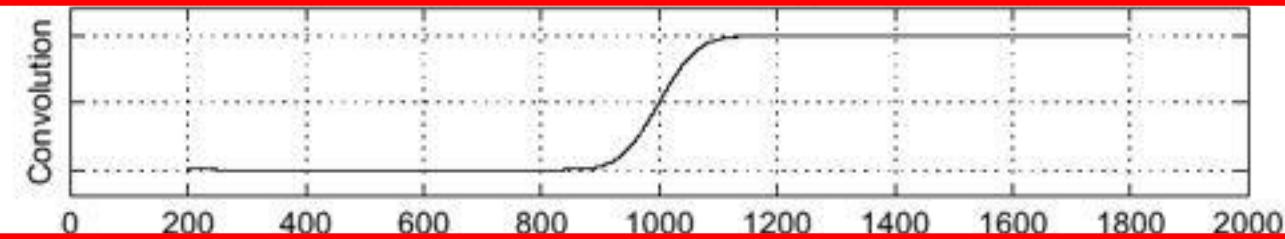
$f$



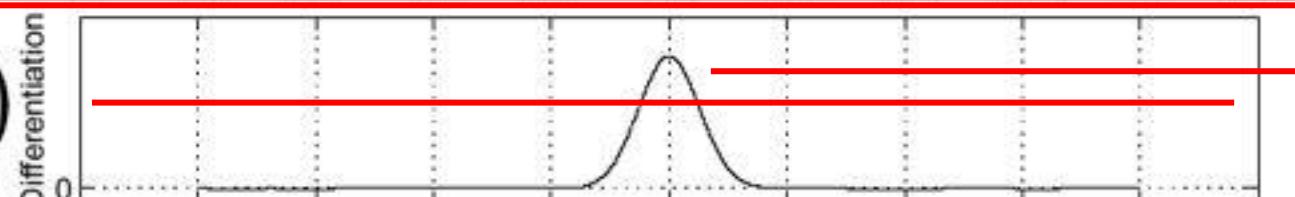
$g$



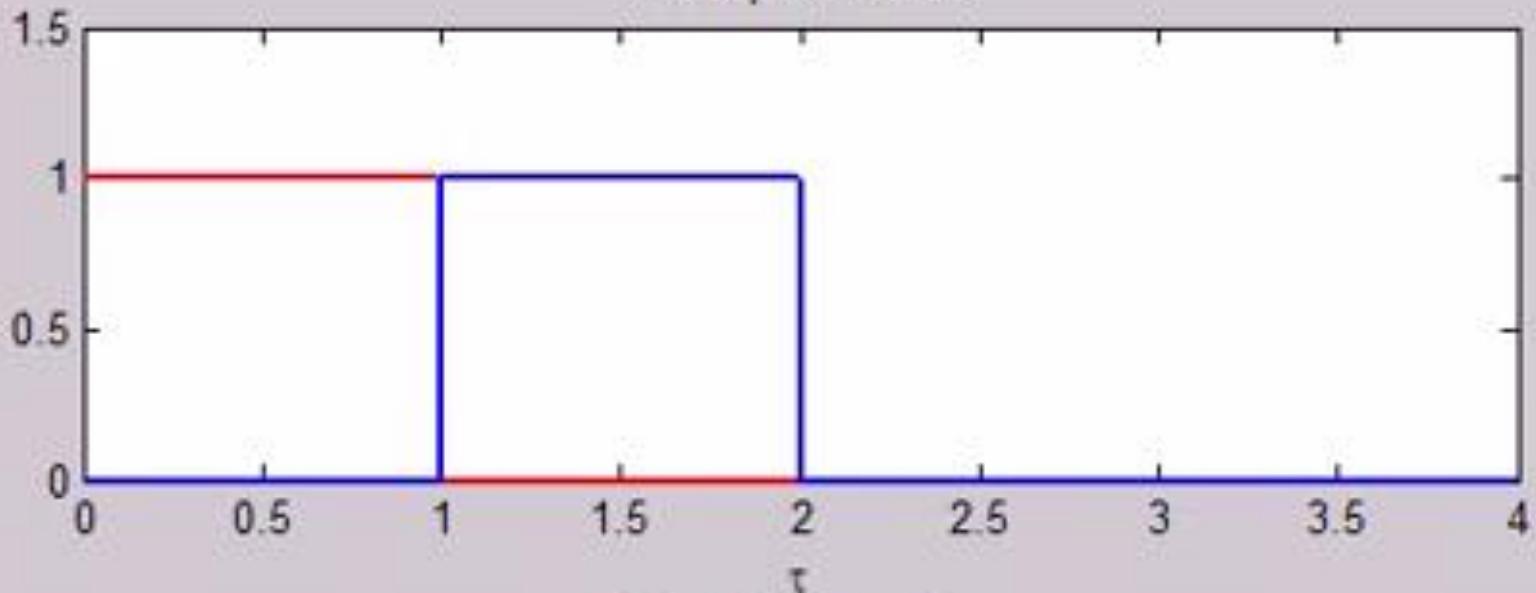
$f * g$



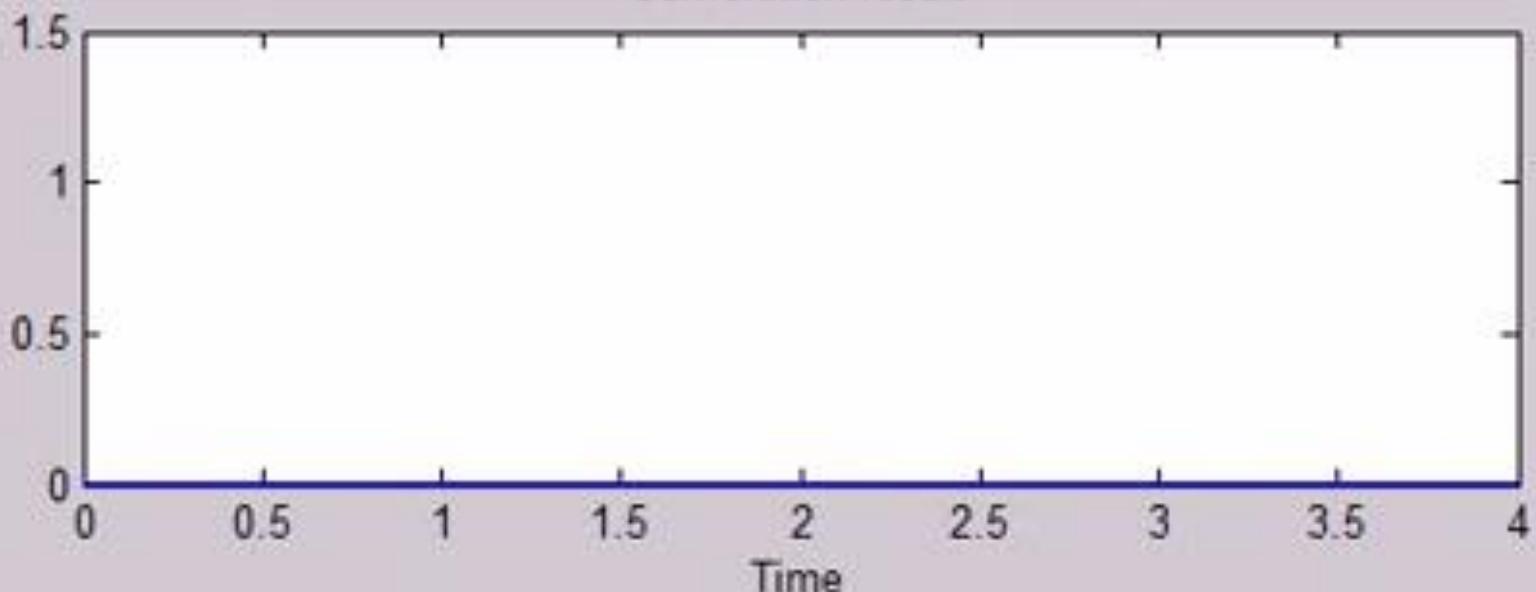
$\frac{d}{dx}(f * g)$



Two square waves



Convolution result



# Animation of Convolution of Two Time Signals

# Turunan dari konvolusi



1 Step

- Sifat turunan dari konvolusi:

$$\frac{\partial(f*g)}{\partial x} = f * \frac{\partial}{\partial x} g$$

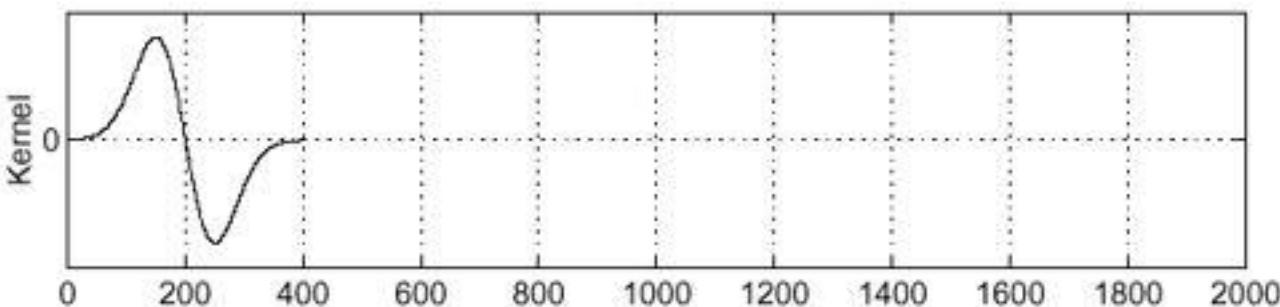
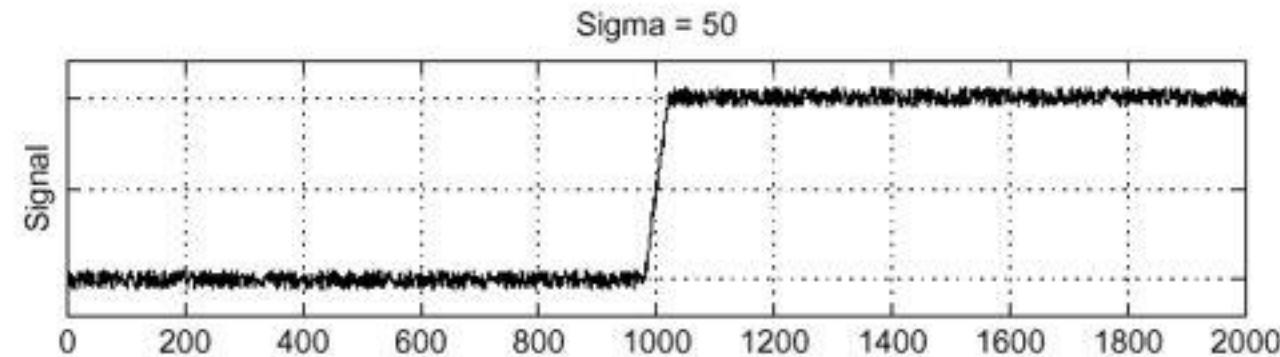
$$(f*g)' = \frac{d}{dt} \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

$$= \int_{-\infty}^{\infty} f(\tau) \frac{\partial}{\partial t} g(t - \tau) d\tau$$

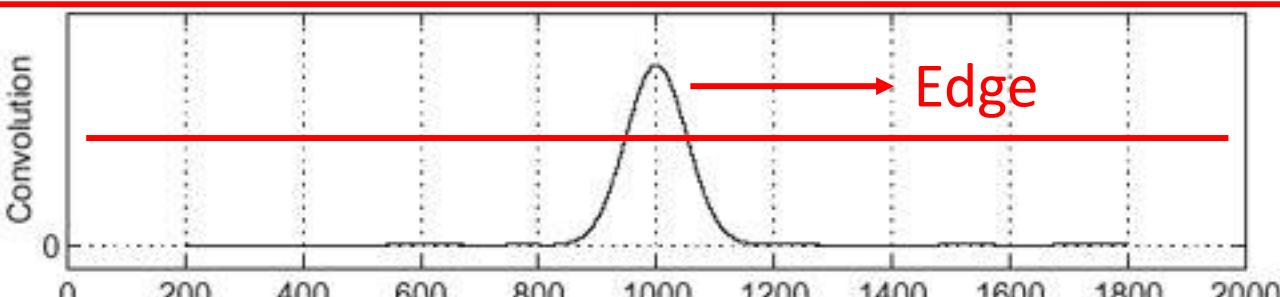
$$= \int_{-\infty}^{\infty} f(\tau)g'(\tau - \tau) d\tau = f * g'.$$

$f$

$$\frac{d}{dx} g$$

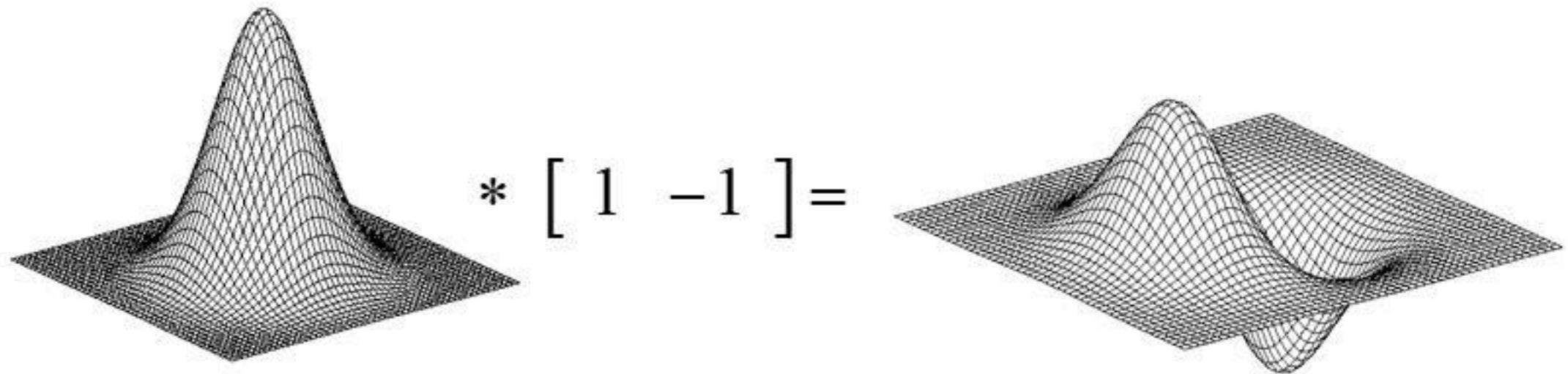


$$f * \frac{d}{dx} g$$



# Turunan dari filter gaussian

- Apakah filter ini dapat dipisahkan?



- Ya, kita dapat memisahkannya menjadi 2 buah filter yang urutannya tidak penting
- Median filter adalah contoh sebuah filter yang tidak dapat dipisahkan

# Median Filter

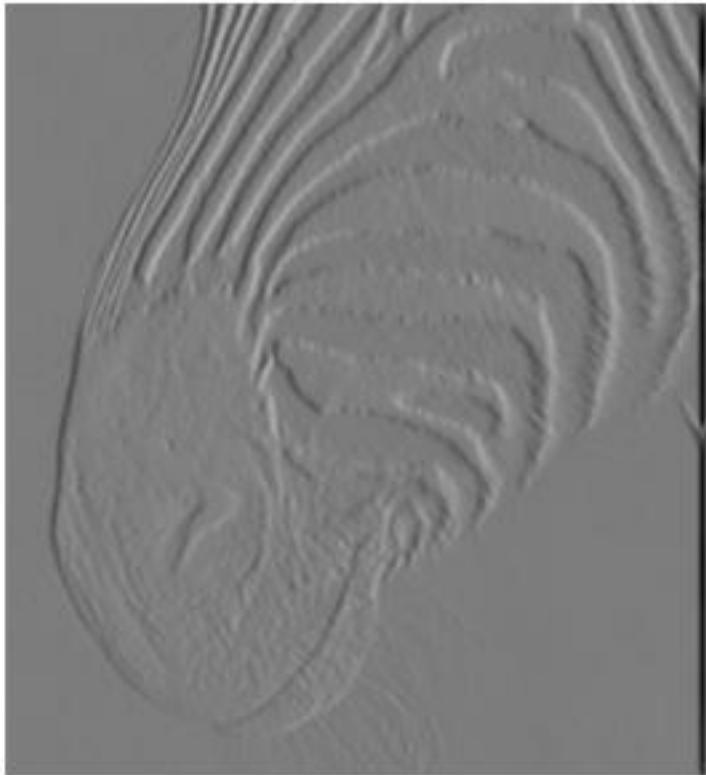
8	9	10	11	12
12	8	9	10	11
13	12	8	9	10
14	12	12	8	9
15	14	13	12	8

# Median Filter

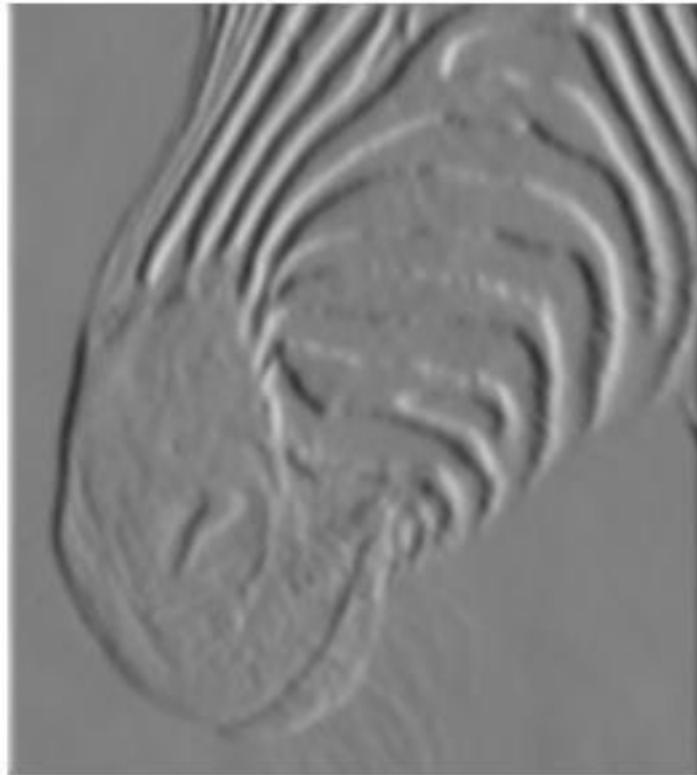
- Median filter adalah contoh dari filter yang tidak dapat dipisahkan karena ia menggunakan informasi spasial dari data inputnya untuk menghasilkan outputnya.
- Filter yang tidak dapat dipisahkan adalah jenis filter yang tidak dapat diwakilkan oleh fungsi matematis tunggal. Sebaliknya, jenis filter ini memerlukan pemrosesan data yang bergantung pada konteks lokal di sekitarnya.

# Smoothing tradeoffs

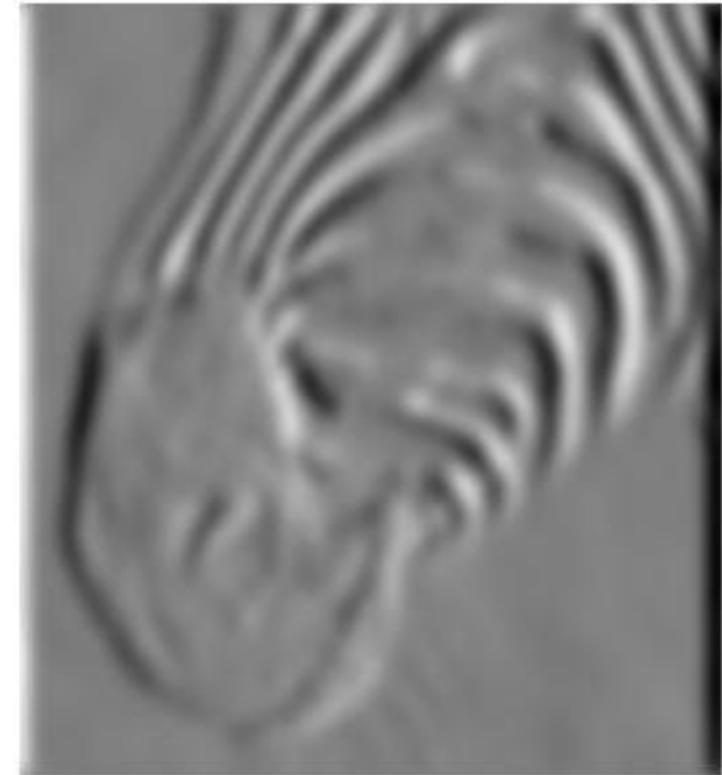
- Turunan yang dihaluskan menghilangkan noise, tapi blur batas
- Batas dapat ditemukan dalam skala yang berbeda



1 pixel

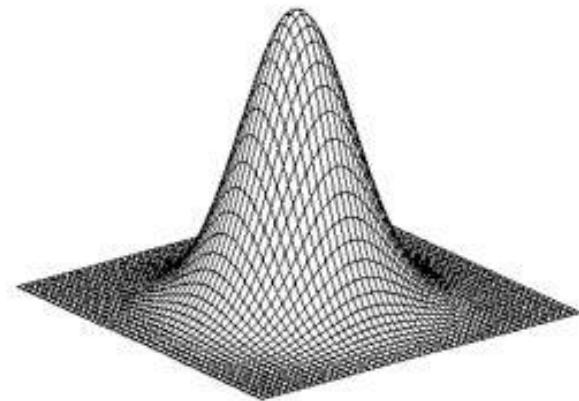


3 pixels

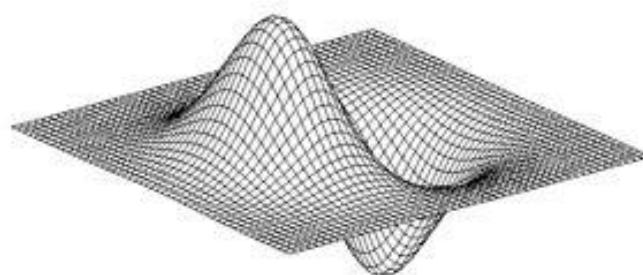


7 pixels

# Filter deteksi batas

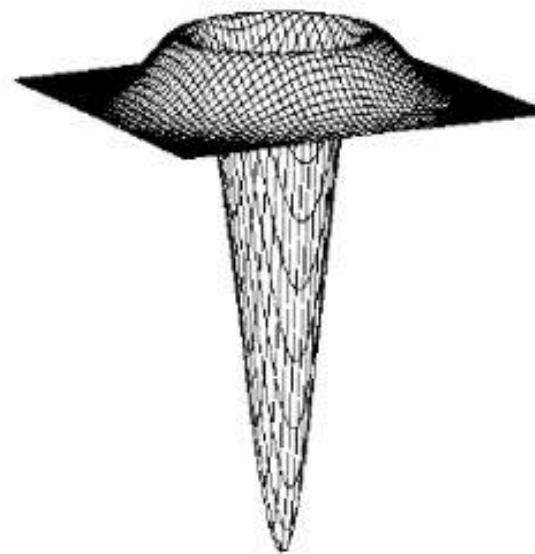


Gaussian



derivative of Gaussian

Laplacian of Gaussian



$g$

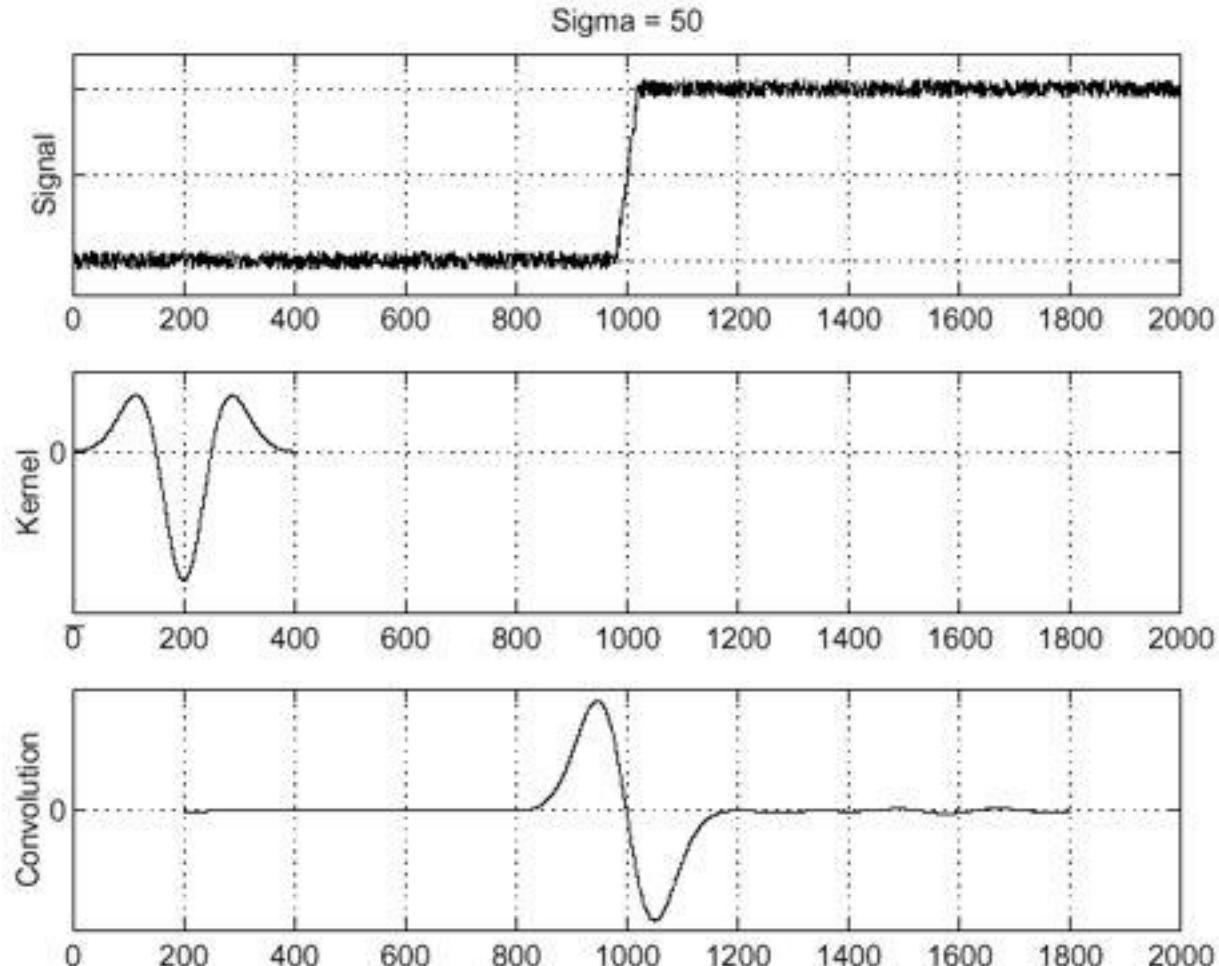
$$\frac{dg}{dx}$$

$$\frac{d^2g}{dx^2} + \frac{d^2g}{dy^2}$$

# Laplacian of Gaussian (LoG)

- Manakah **batas** menurut mekanisme ini?

$$f * \frac{d^2}{dx^2} g$$



1	1	1
1	-8	1
1	1	1

# Difference of Gaussians

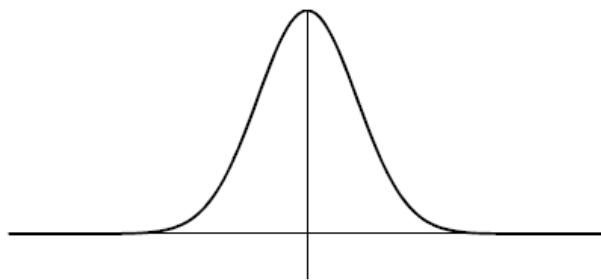
MATLAB



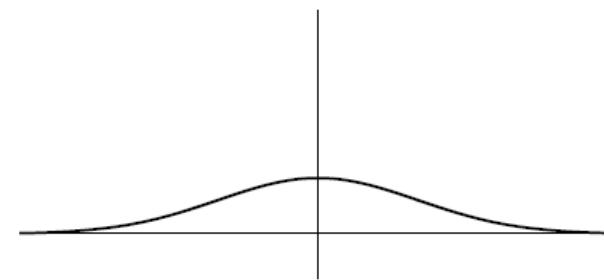
# Difference of Gaussians (DoG)

- DoG dapat mengaproksimasi LoG secara cukup baik
  - LoG (dashed blue) vs DoG (black)

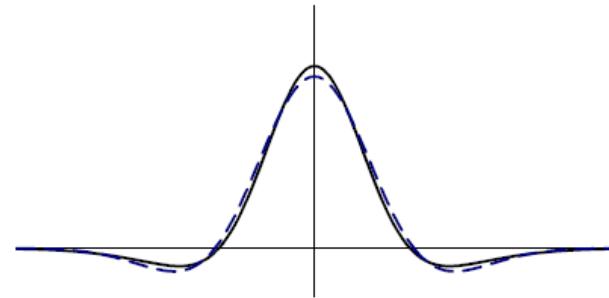
space:



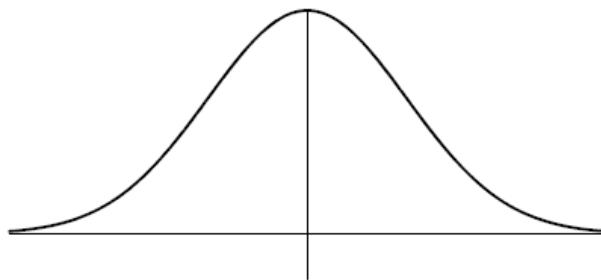
-



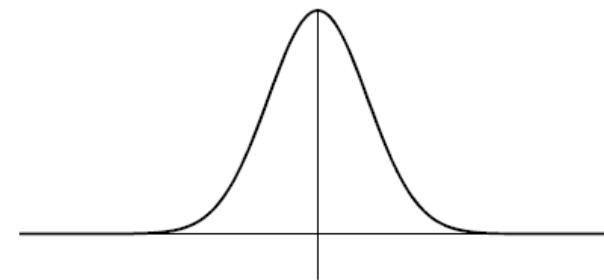
=



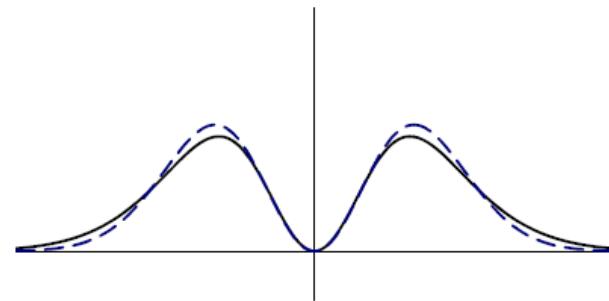
frequency:



-

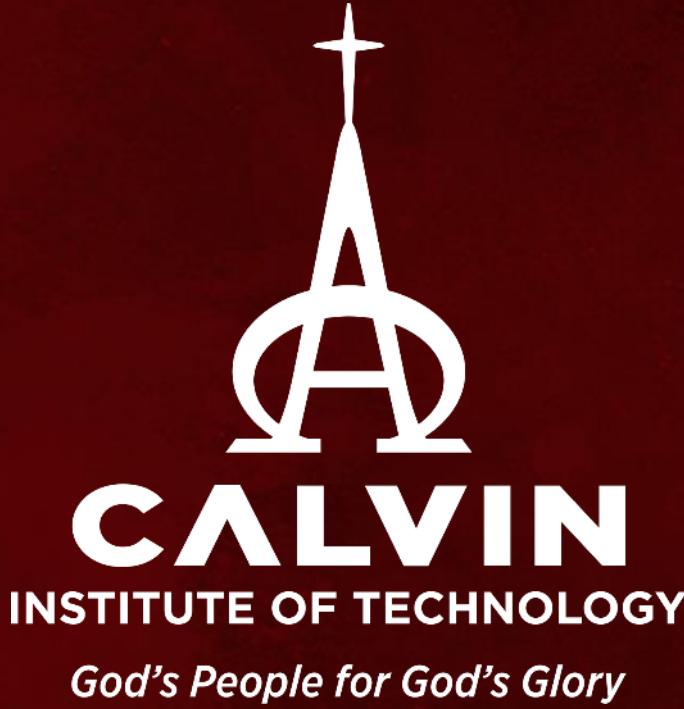


=



low-pass

lower-pass

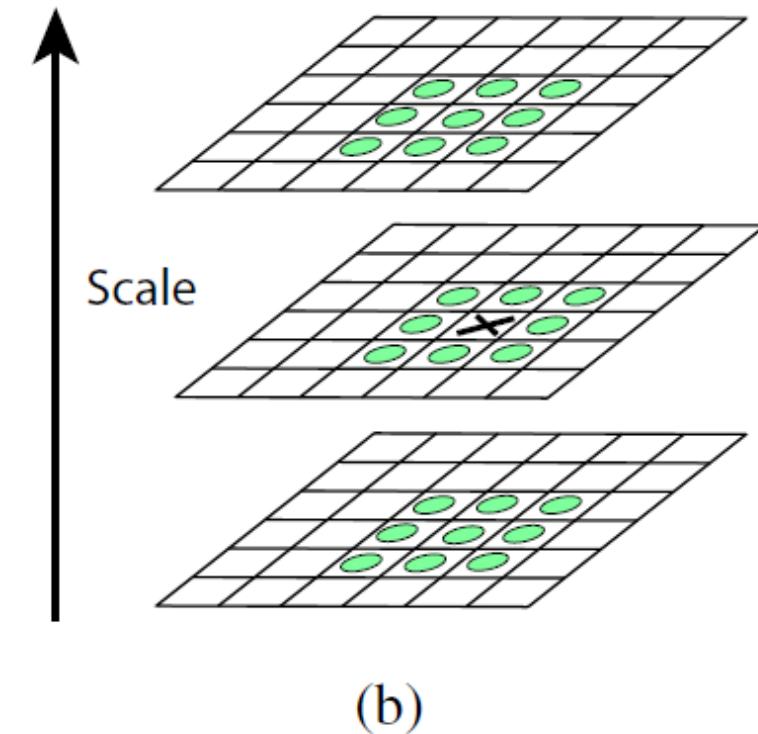
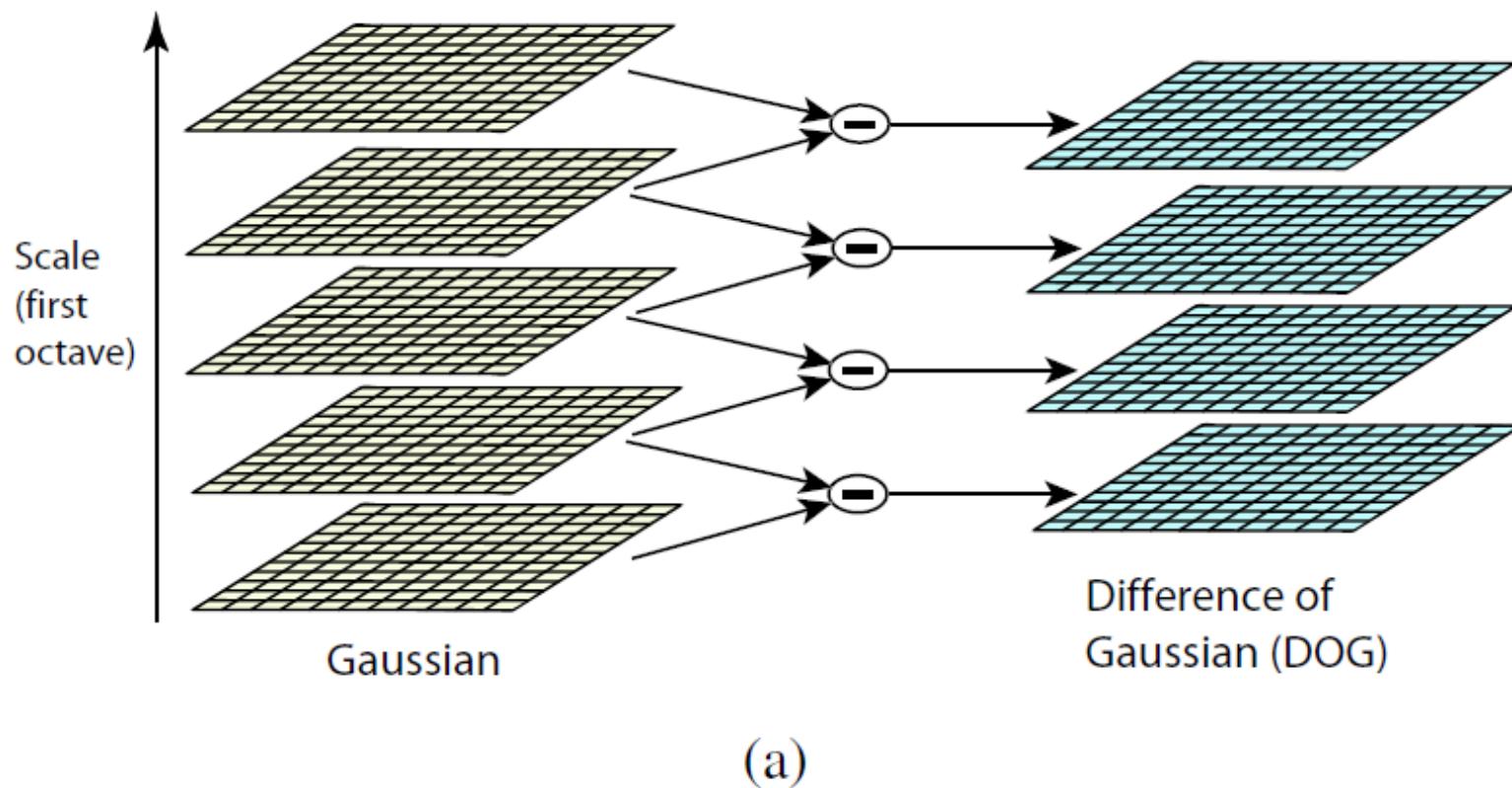


# Deteksi Batas dan Garis

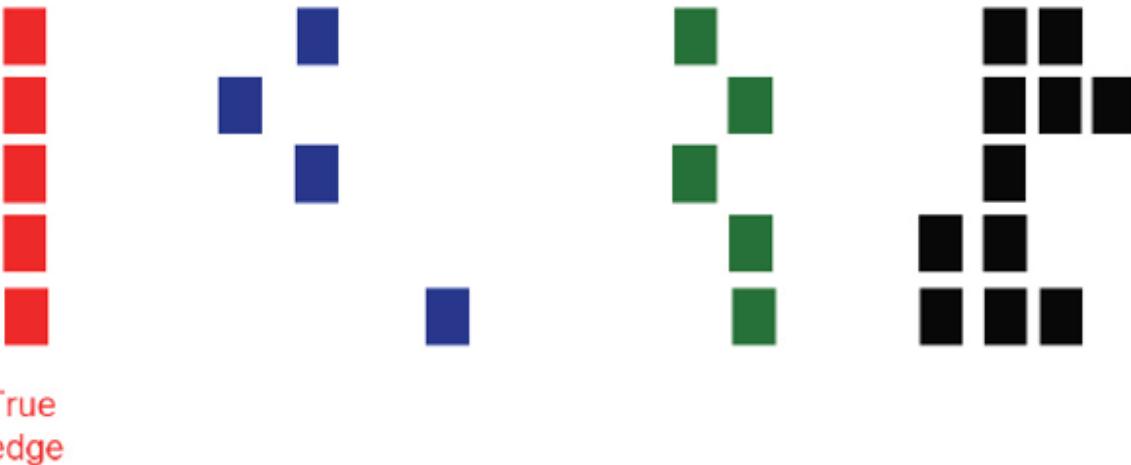
IBDA4311 – *Computer Vision*

# Piramida Gaussian dan DoG

- DoG biasanya digunakan untuk piramida gaussian untuk deteksi fitur yang bersifat scale-invariant



# Apakah deteksi batas yang baik?



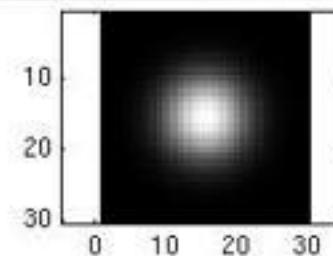
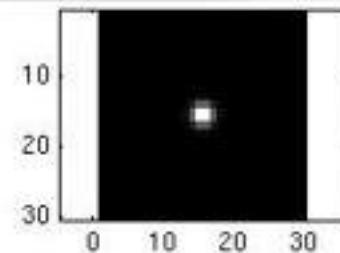
- Deteksi:
  - Meminimalkan false positives (deteksi yang salah)
  - Meminimalkan false negatives (gagal mendeksi batas sebenarnya)
  - Memaksimalkan deteksi yang benar
- Lokalisasi:
  - Batas yang terdeteksi harus sedekat mungkin dengan batas yang benar
- Respon tunggal:
  - Menghasilkan satu respons untuk setiap posisi batas yang benar
- Menghubungkan deteksi ke garis

# Apa sajakah parameter dalam deteksi batas?

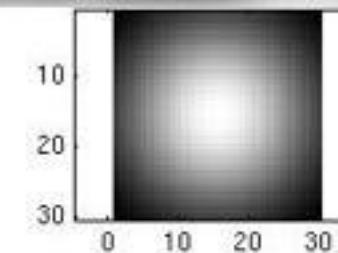
- Skala
- Threshold

# Pemilihan skala

- Kita menghaluskan image dengan kernel gaussian untuk mengurangi noise
- Skala Gaussian menentukan seberapa halus

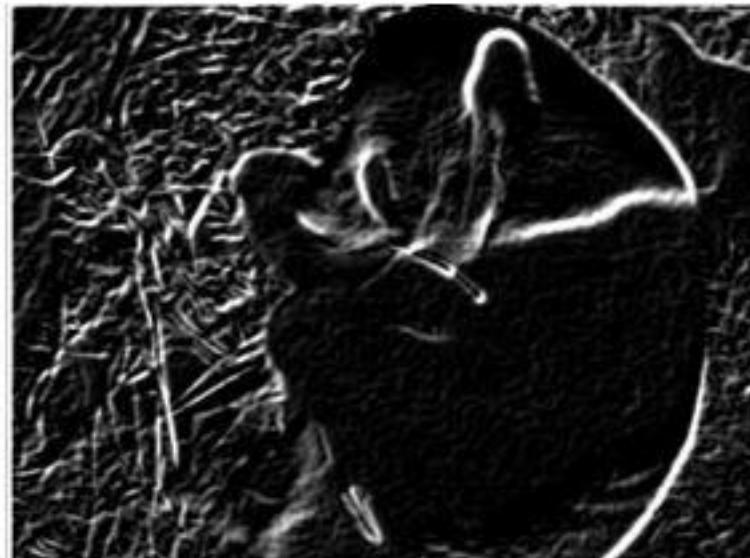


...

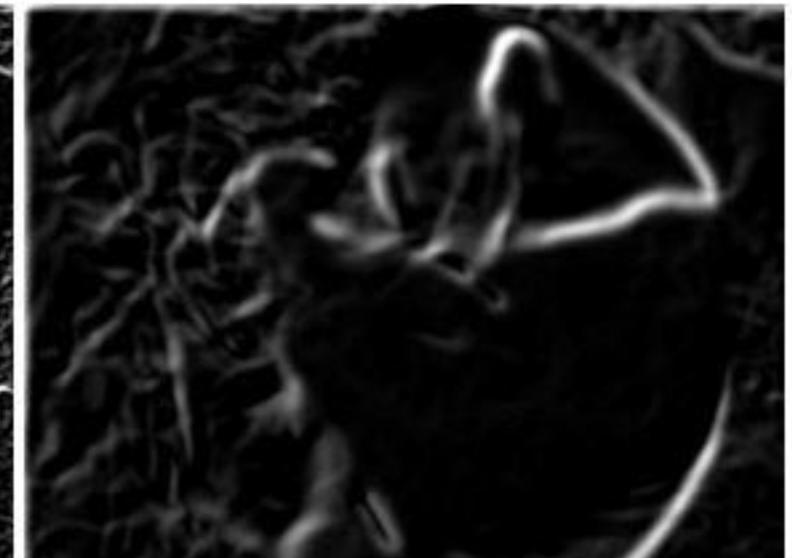


# Efek dari skala terhadap turunan

- Hasil berbeda tergantung skala Gaussian
- Skala besar: deteksi batas berskala besar
- Skala kecil: deteksi batas yang lebih tipis



$\sigma = 1$  pixel



$\sigma = 3$  pixels

# Bagaimana memilih skala?

- Tergantung dari apa yang kita cari
- Skala terlalu kecil: tidak bisa melihat hal besar
- Skala terlalu besar: tidak dapat membedakan hal-hal tipis

# Threshold

- Pilih sebuah nilai threshold
- Tentukan gradien apapun diantara 0 sampai nilai tersebut (off)
- Tentukan gradien apapun diantara nilai tersebut sampai 1 (on)

# Canny edge detector

- Deteksi batas yang paling umum digunakan di computer vision
- Deteksi step-edges yang dirusak oleh gaussian noise

# Canny edge detector

- Filter gambar dengan turunan gaussian
- Menemukan orientasi dan besar dari gradien
- Non-maximum suppression: (lokalisasi)
  - Dimulai dari riak dengan lebar multi-pixel sampai lebar 1 pixel
- Linking and thresholding:
  - Tentukan 2 thresholds: low dan high
  - Gunakan threshold yang lebih tinggi untuk membuat garis batas lalu lanjutkan dengan threshold yang lebih rendah
- Cv2.Canny()

# Canny edge detector



The original image

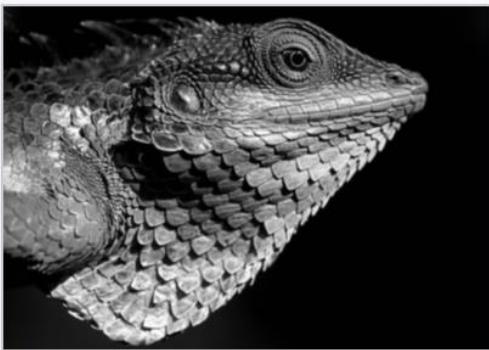
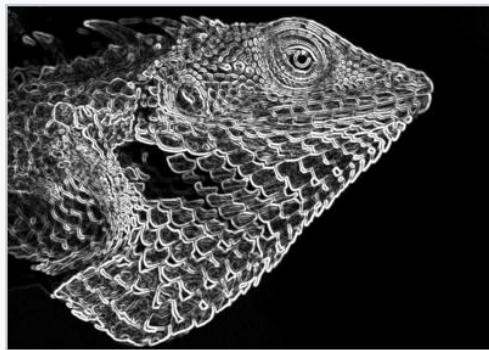
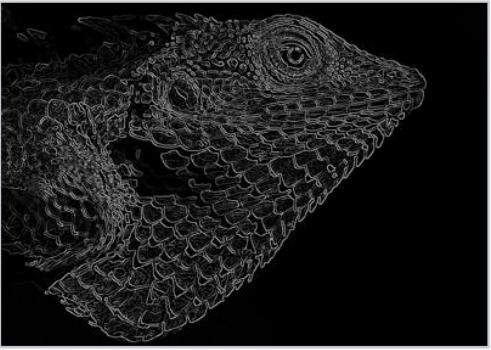


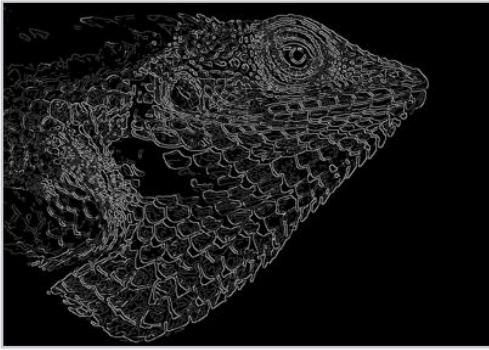
Image has been reduced to grayscale, and a 5x5 Gaussian filter with  $\sigma=1.4$  has been applied



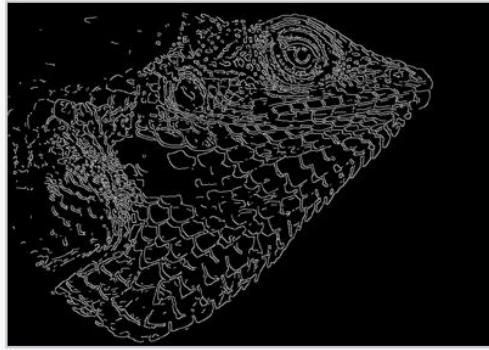
The intensity gradient of the previous image. The edges of the image have been handled by replicating.



Non-maximum suppression applied to the previous image.

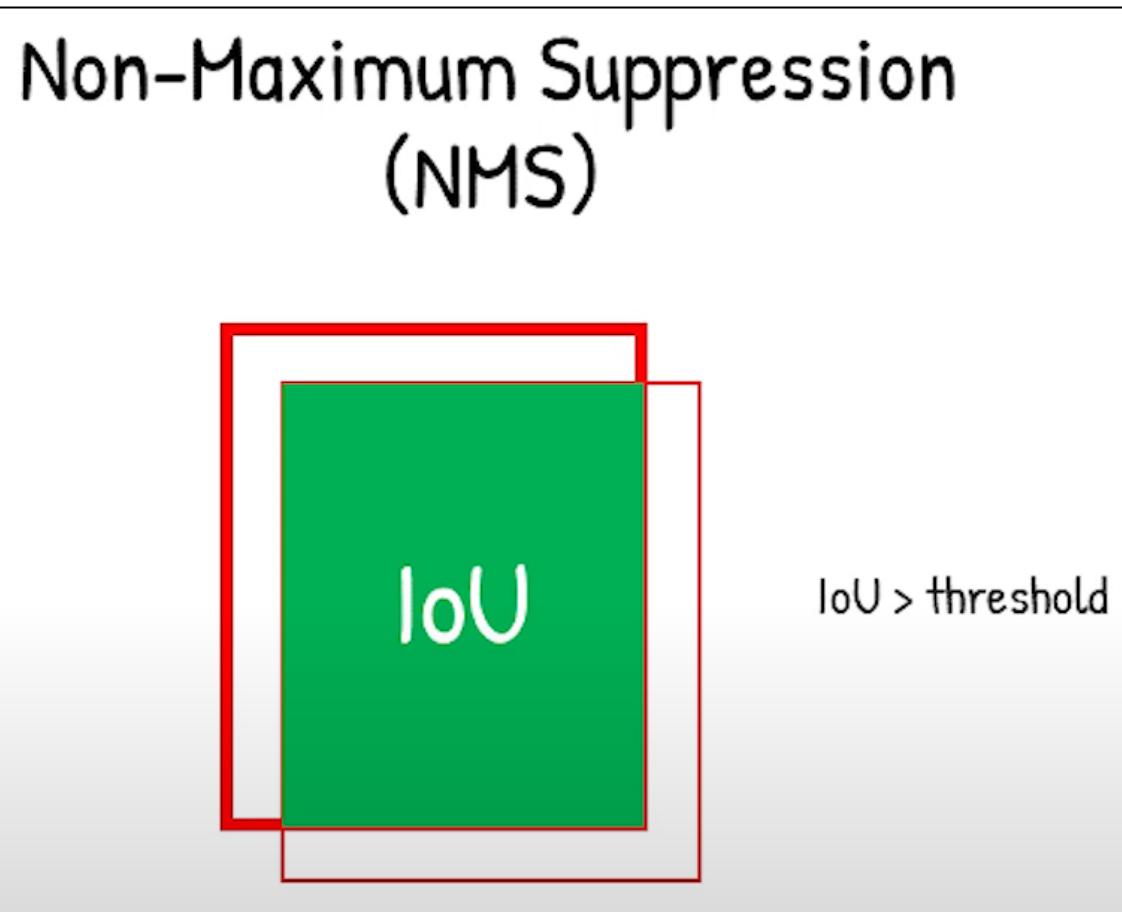


Double thresholding applied to the previous image. Weak pixels are those with a gradient value between 0.1 and 0.3. Strong pixels have a gradient value greater than 0.3

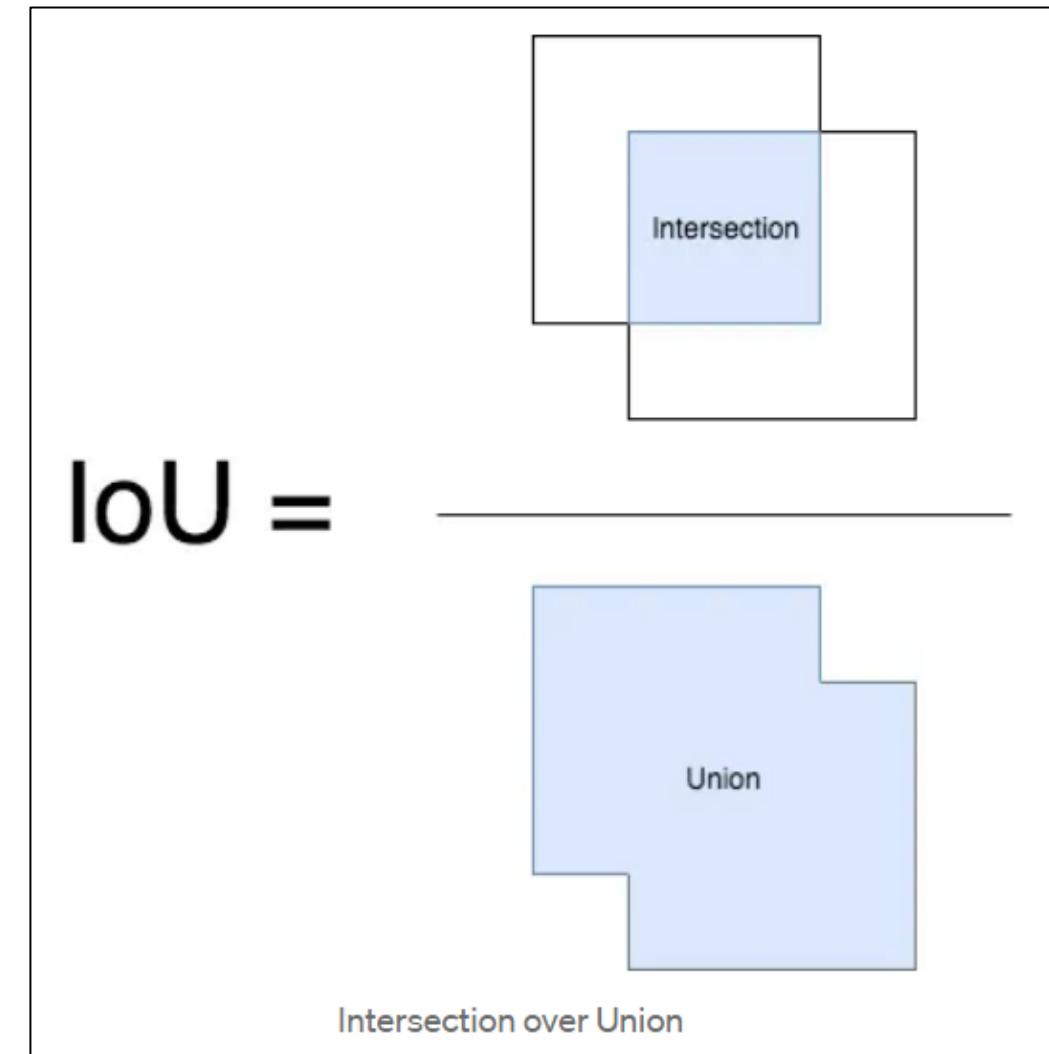


Hysteresis applied to the previous image

# Non-Maximum Suppression (NMS)



[https://www.youtube.com/watch?v=lYLyA1aQmrU&ab\\_channel=AnakAI](https://www.youtube.com/watch?v=lYLyA1aQmrU&ab_channel=AnakAI)



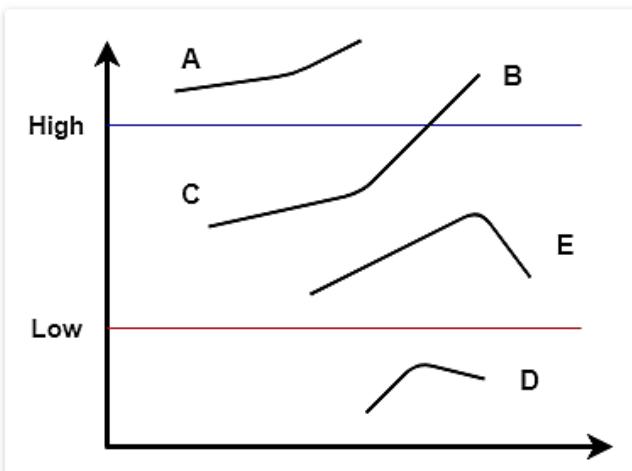
<https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c>

# Hysteresis Thresholding

Non-max suppression outputs a more accurate representation of real edges in an image. But you can see that some edges are more bright than others. The brighter ones can be considered as strong edges but the lighter ones can actually be edges or they can be because of noise. To solve the problem of "which edges are really edges and which are not" Canny uses the Hysteresis thresholding. In this, we set two thresholds 'High' and 'Low'.

- Any edges with intensity greater than 'High' are the sure edges.
- Any edges with intensity less than 'Low' are sure to be non-edges.
- The edges between 'High' and 'Low' thresholds are classified as edges only if they are connected to a sure edge otherwise discarded.

Let's take an example to understand



Here, A and B are sure-edges as they are above 'High' threshold. Similarly, D is a sure non-edge. Both 'E' and 'C' are weak edges but since 'C' is connected to 'B' which is a sure edge, 'C' is also considered as a strong edge. Using the same logic 'E' is discarded. This way we will get only the strong edges in the image.

This is based on the assumption that the edges are long lines.

<https://theailearner.com/tag/hysteresis-thresholding/>

# Deteksi Garis

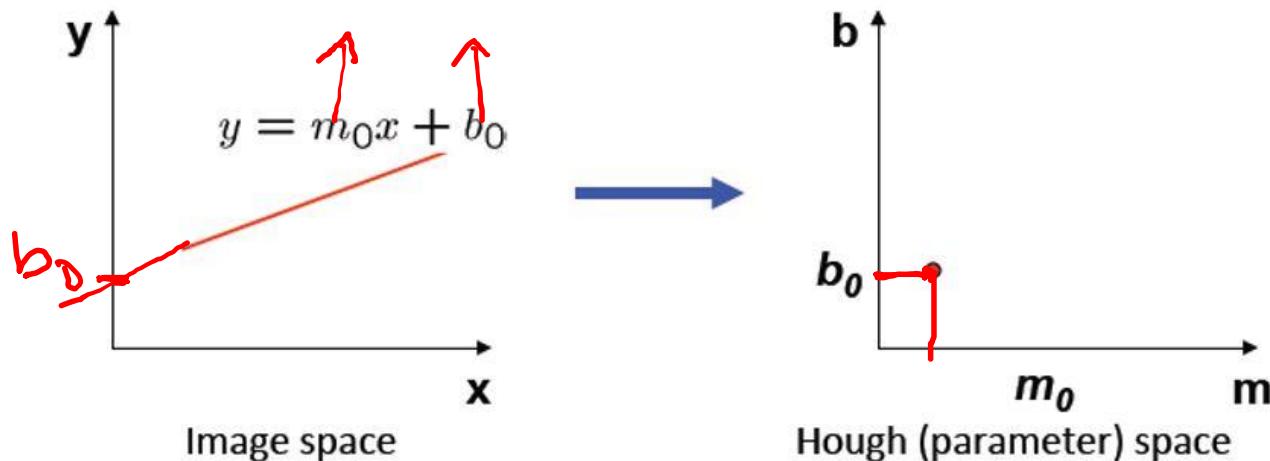
# Line Fitting

Metode-metode yang dapat digunakan:

1. Hough Transform
2. RANSAC → *deep learning*
3. SCNN

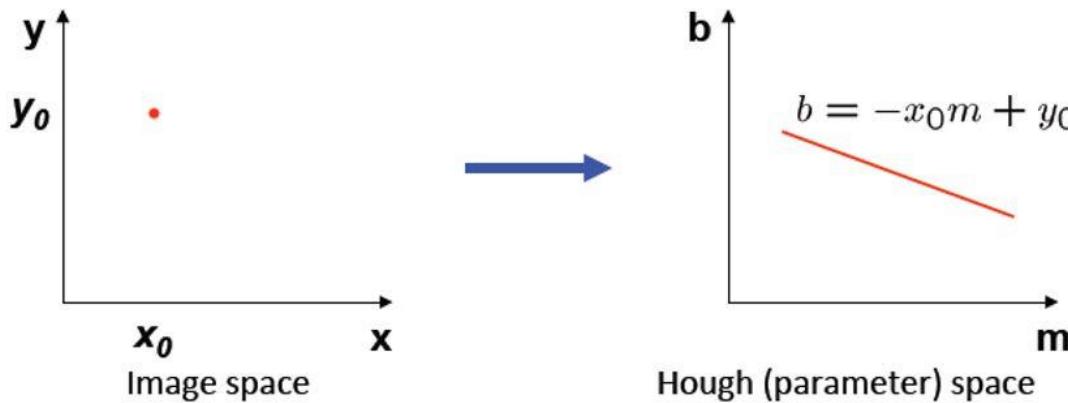
# Hough Transform

- Transformasi image space ( $x,y$ ) menuju Hough space ( $m,b$ )
- Sebuah garis pada gambar adalah sebuah titik pada hough space
- Hough: untuk himpunan poin ( $x,y$ ) temukan semua ( $m,b$ ) dimana  $y=mx+b$

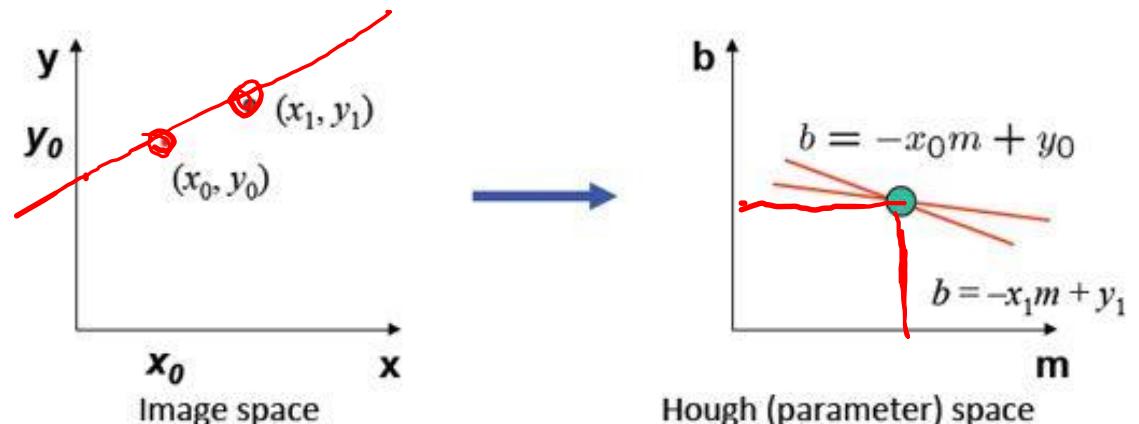


# Hough Transform

- Sebuah titik  $(x_0, y_0)$  pada gambar adalah solusi dari  $b = -x_0m + y_0$  (Titik -> Garis)

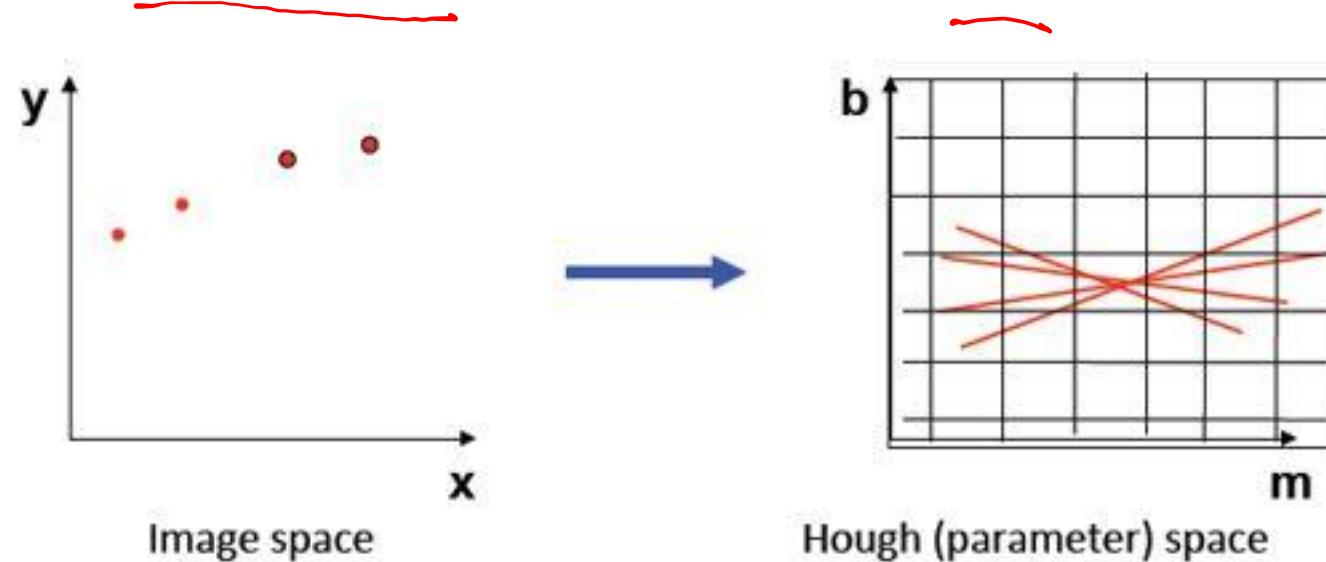


- Sebuah garis yang berisi titik  $(x_0, y_0)$  dan  $(x_1, y_1)$  adalah pertemuan garis  $b = -x_0m + y_0$  dan garis  $b = -x_1m + y_1$



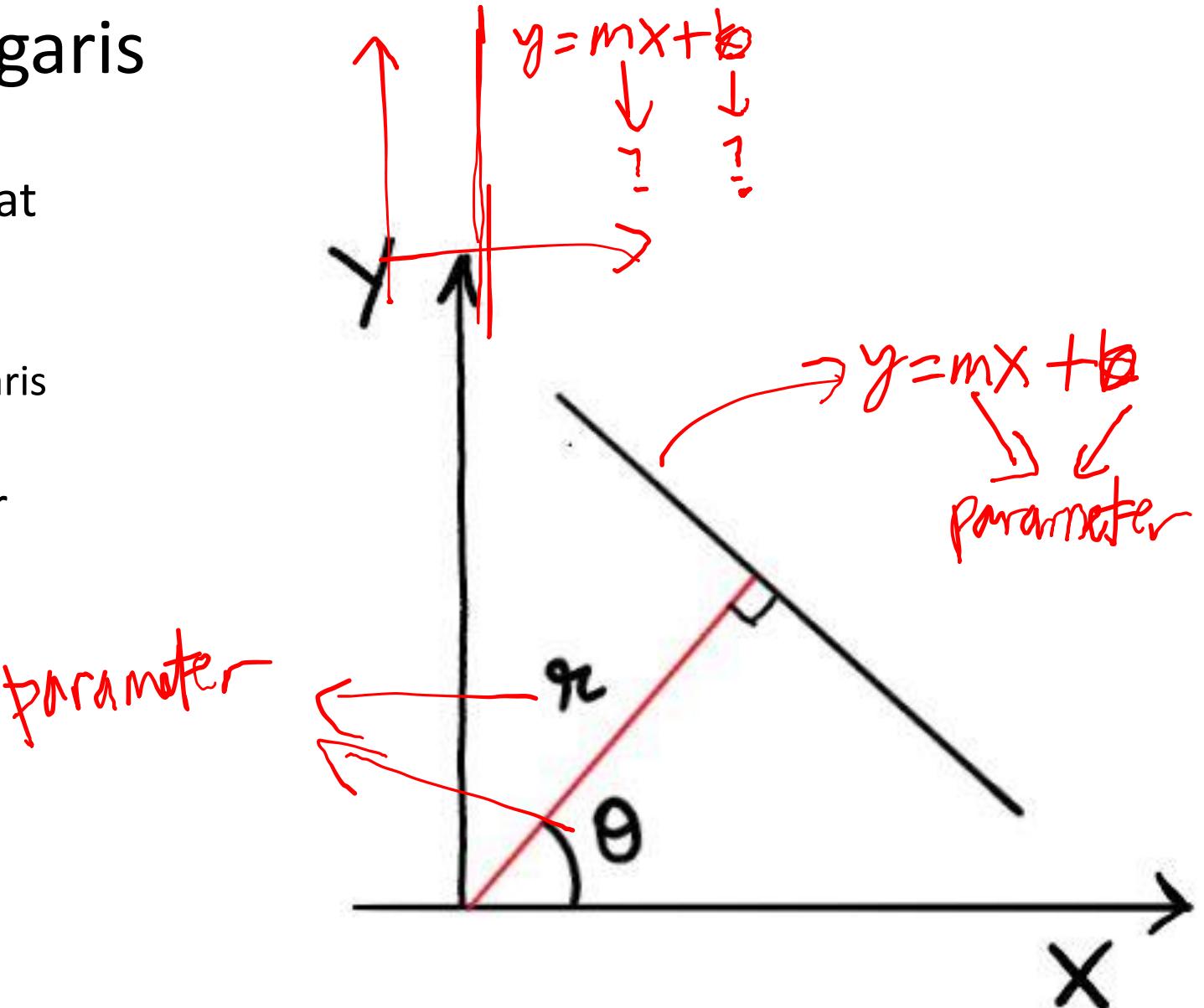
# Menemukan garis dengan transformasi Hough

- Setiap titik batas voting untuk himpunan parameter mungkin pada Hough space
- Hitung votes pada sebuah himpunan diskrit
- Parameter dengan voting banyak mengindikasikan garis pada image space



# Representasi polar dari garis

- Persamaan garis  $y = mx + b$  dapat bermasalah, misalnya:
  - Dapat bernilai tak hingga
  - Tidak dapat didefinisikan untuk garis vertikal
- Solusi: gunakan representasi polar

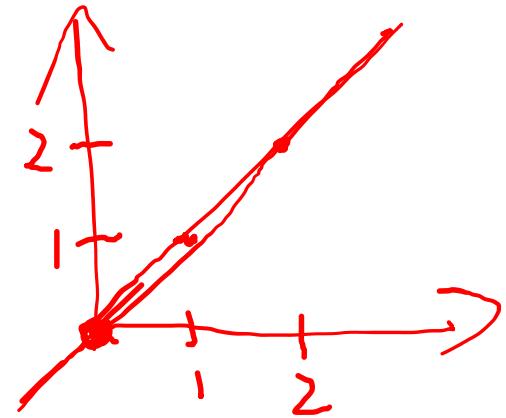


# Uji Pemahaman

- Apakah representasi polar dari garis  $y = x$

$$r \cos \theta = r \sin \theta$$
$$\cos \theta = \sin \theta$$
$$\theta = 45^\circ$$

$$y = mx + b$$
$$= 1x + 0$$

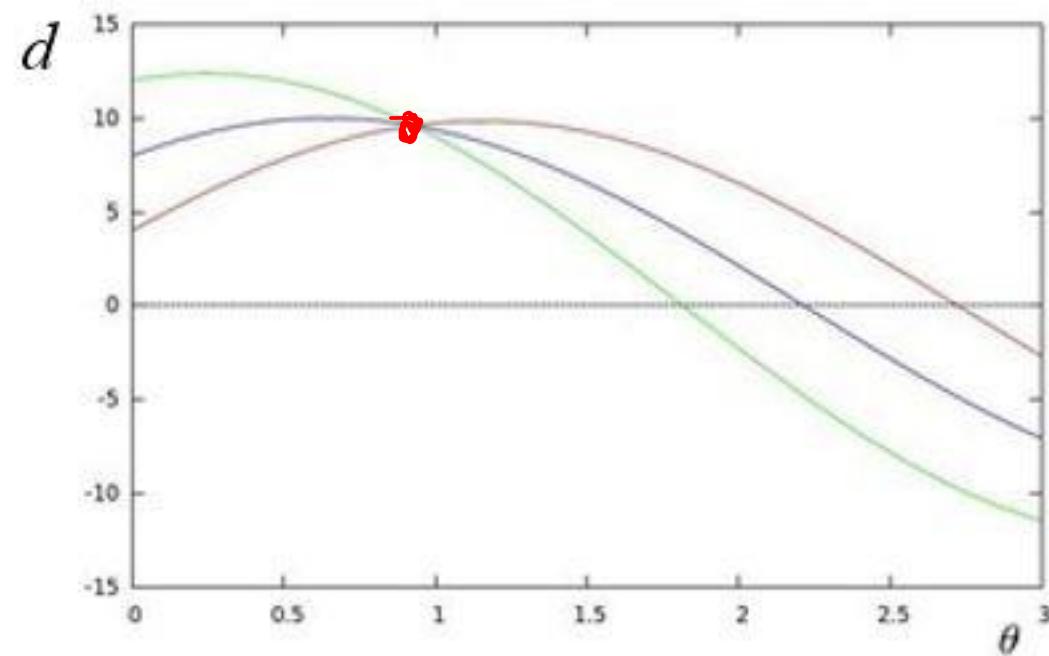
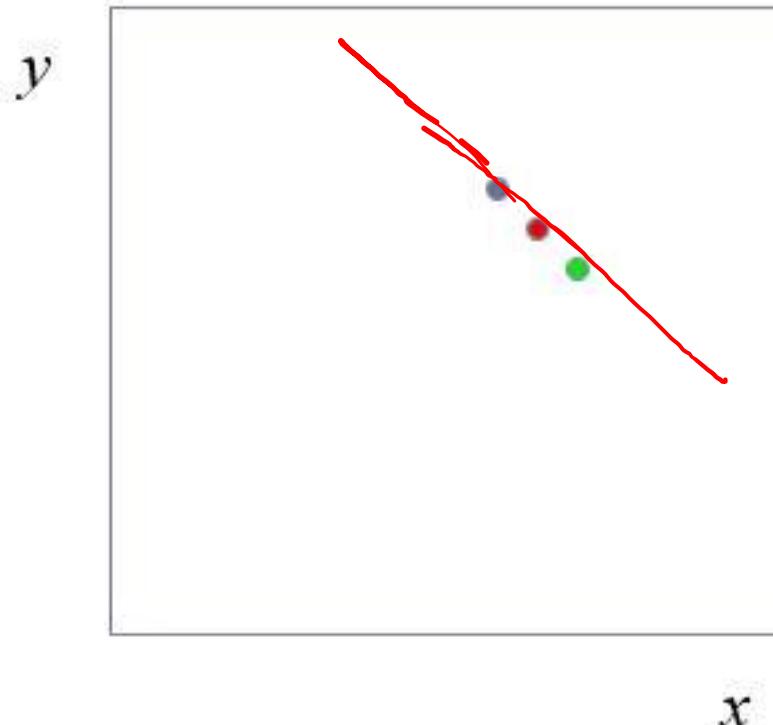


# Contoh Transformasi Hough

- Untuk suatu titik  $(x_0, y_0)$

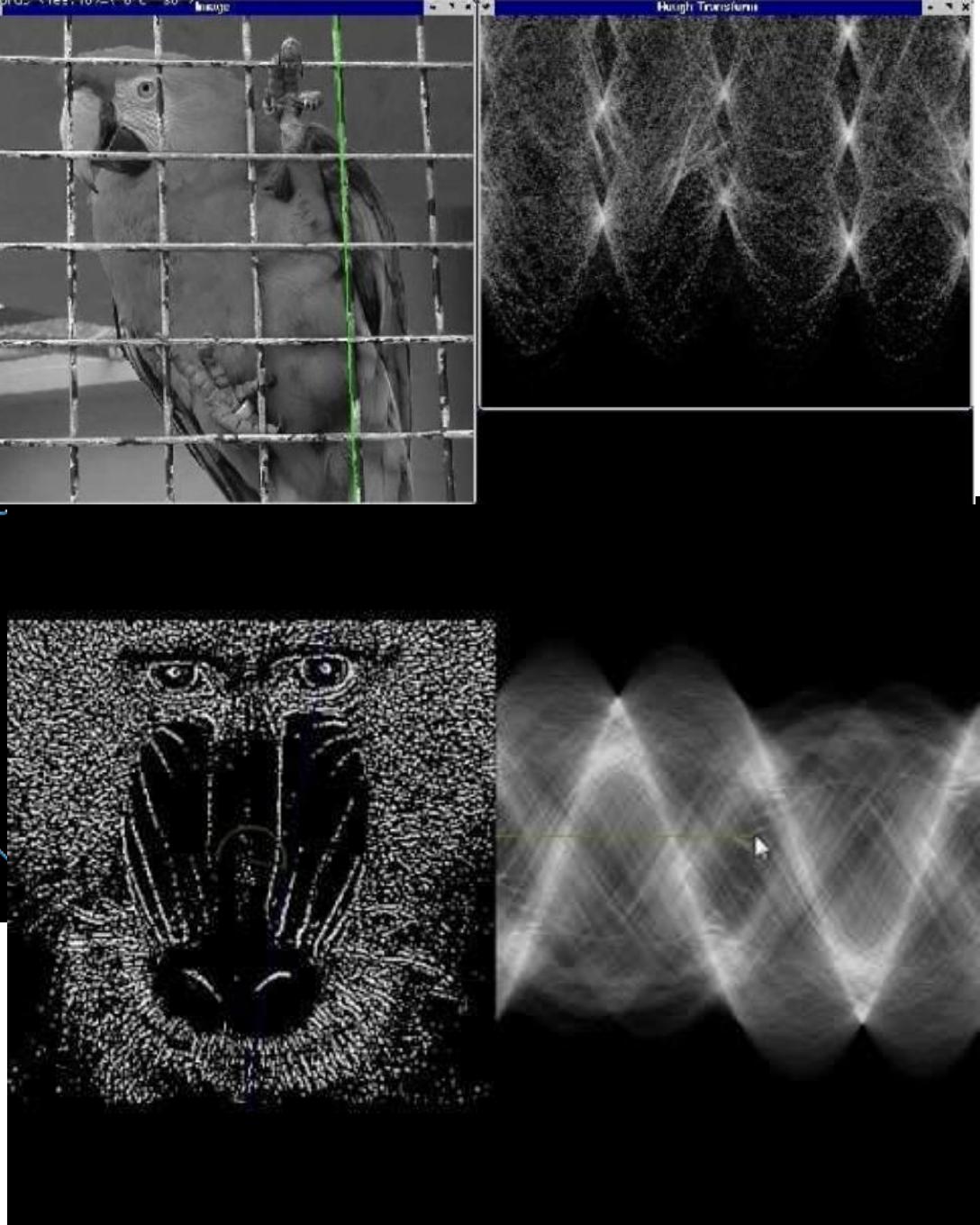
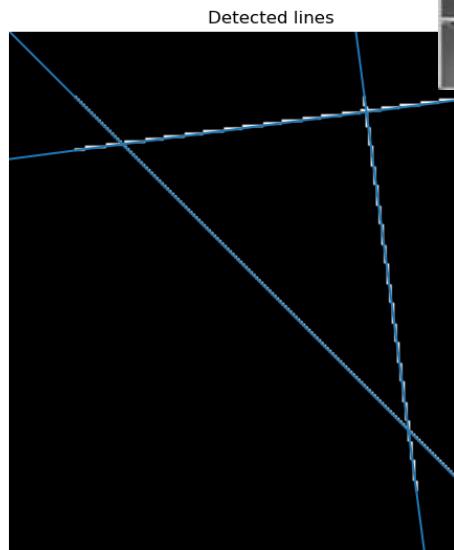
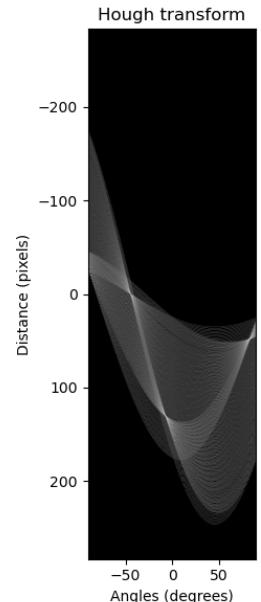
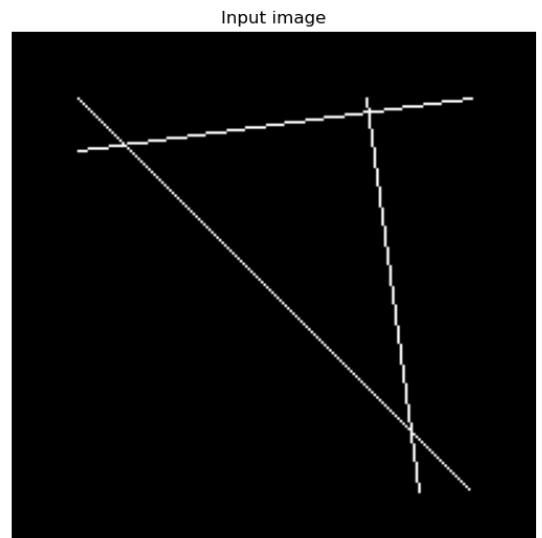
$$r(\theta) = x_0 \cos \theta + y_0 \sin \theta$$

- Untuk 3 titik sepanjang garis ( $d \equiv r$ )



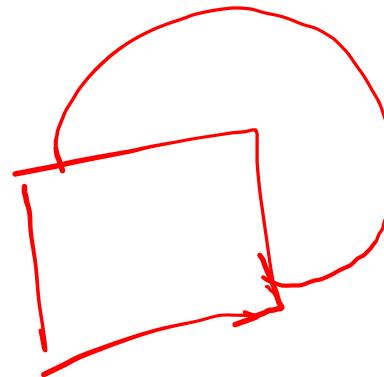
# Contoh pada gambar

- Image space vs parameter space

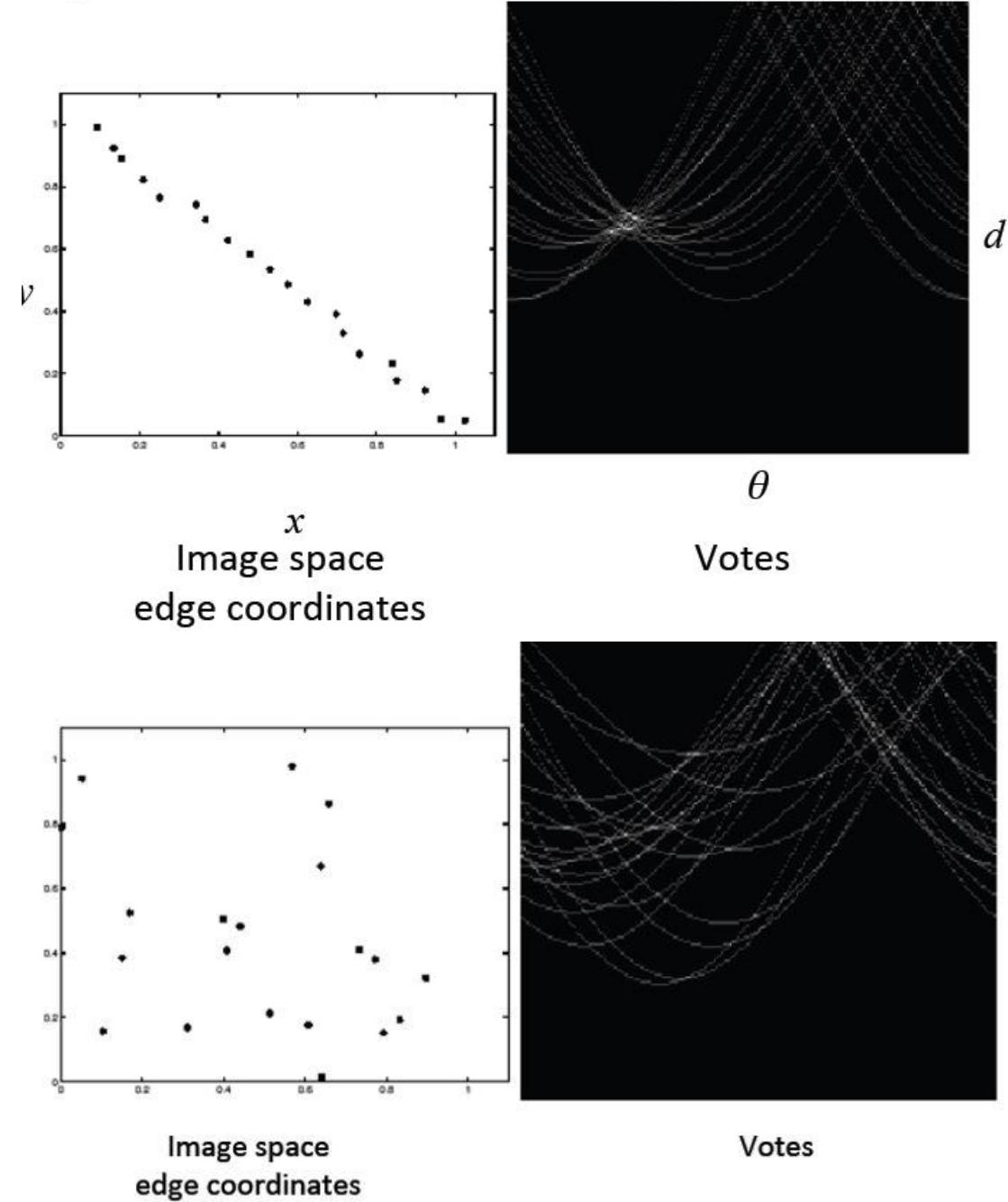
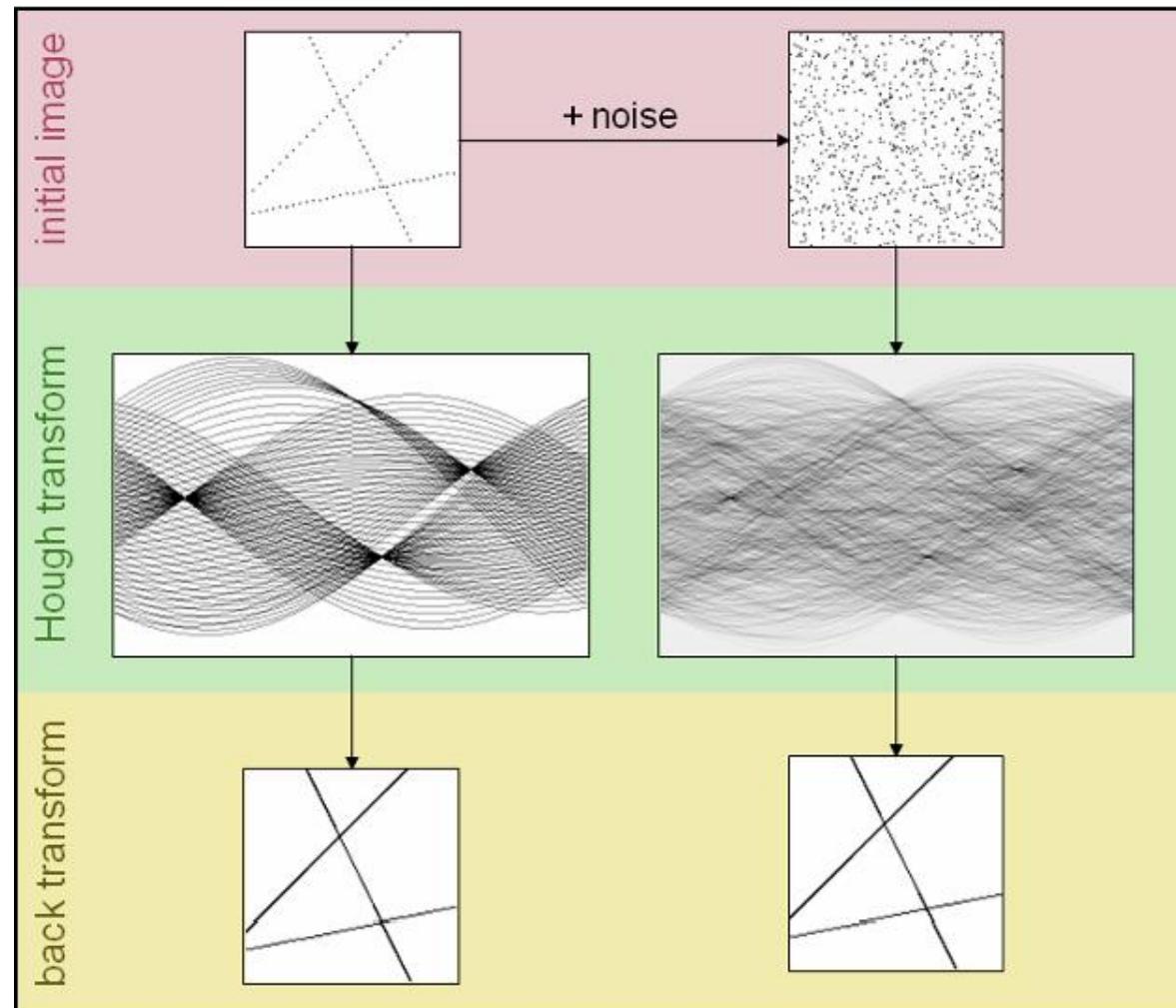


# Sifat Transformasi Hough

- Noise dan clutter votes tidak konsisten sehingga tidak akan terakumulasi
- Dapat mengatur occlusion selama model menerima cukup banyak votes
- Efisien



# Noise pada Transformasi Hough



# Kelebihan dan Kekurangan Transformasi Hough

- **Kelebihan**

- Dapat menerima occlusions
- Robust terhadap noise
- Dapat mendeteksi beberapa garis sekaligus dalam sekali proses gambar

- **Kekurangan**

- Kompleksitas meningkat eksponensial sesuai jumlah parameter

garis  $\rightarrow \Sigma P$

# Ransac

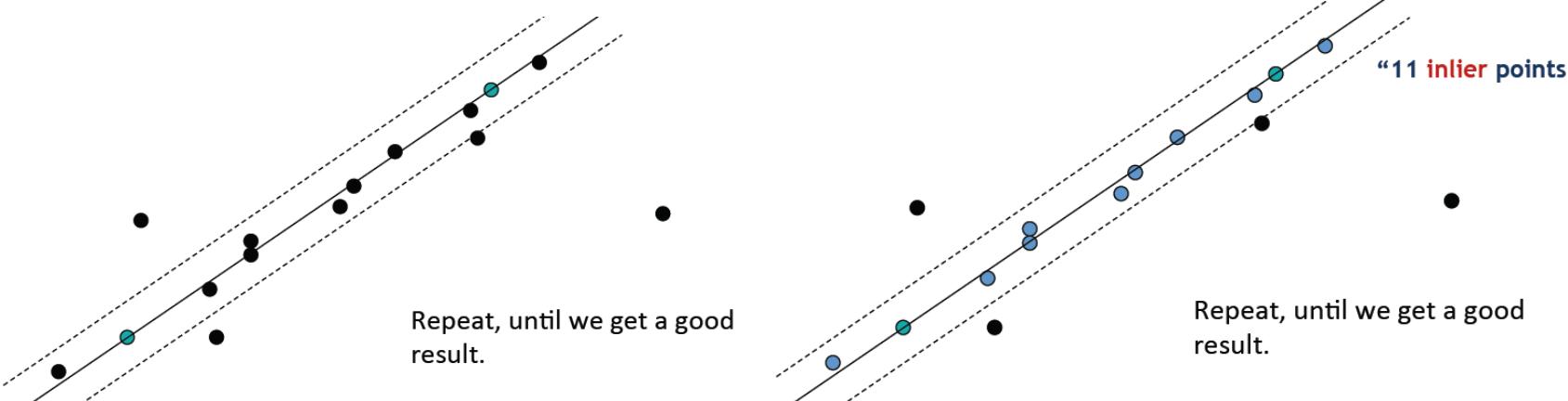
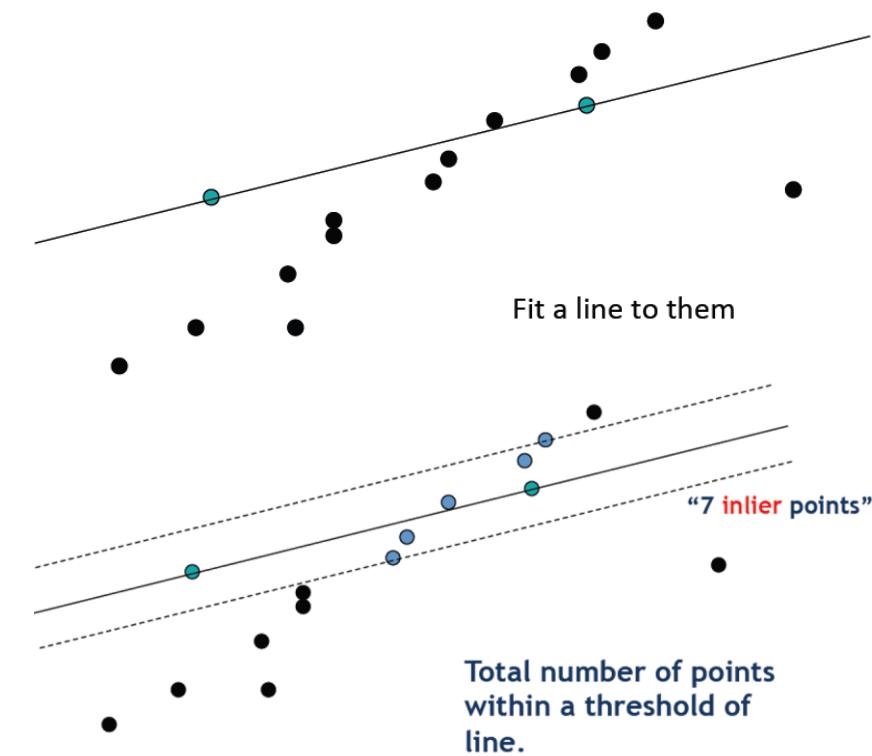
- **RAN**dom **S**ample **C**onsensus [Fischler & Bolles 1981]
- Cari “inliers” dan hanya gunakan ini
- Jika kita fit sebuah model pada “outliers” kita tidak akan memperoleh fit yang sesuai

# Algoritma Ransac

1. Loop:
  - a. Pilih secara acak kumpulan titik
  - b. Fit sebuah model terhadap pilihan tersebut
  - c. Temukan inliers dari model yang sudah dihitung
  - d. Jika jumlah inliers sudah cukup besar, hitung ulang model dengan hanya menggunakan inliers
  - e. Hitung jumlah inliers dari model yang sudah diperbaharui
2. Yang dipilih: model dengan jumlah inliers terbanyak

# Contoh Algoritma RANSAC

1. Data input: kumpulan titik
2. Step 1: pilih 2 titik
3. Step 2: fit garis pada titik-titik tersebut
4. Step 3: temukan inliers
5. Step 4: ulangi step 1-3 sampe konvergen



# Kriteria Stop

Opsi 1: Ketika model sudah cukup baik

- Menurut jumlah inliers
- Menurut fitting error

Opsi 2: menurut peluang

- Misalkan  $K$  adalah jumlah iterasi
- Misalkan  $n$  adalah jumlah titik yang diperlukan untuk menghitung model
- Misalkan  $f$  adalah fraksi inliers dari sebuah model
- Maka peluang sebuah sampel benar adalah:  $f_n$
- Peluang bahwa semua sampel  $K$  gagal adalah:  $(1 - f_n)^K$
- Pilih nilai  $K$  cukup tinggi untuk menjaga tingkat kegagalan rendah

# Implementasi

- OpenCV menggunakan Homografi: `cv2.findHomography(src_pts, dst_pts, cv2_RANSAC)`
- Scikit-image menggunakan: `skimage.measure.ransac()`
- Scikit-learn menggunakan: `sklearn.linear_model.RANSACRegressor()`

# Kelebihan dan Kekurangan

## Kelebihan

- Metode umum yang bekerja cukup baik untuk berbagai jenis masalah model fitting
- Mudah diimplementasikan

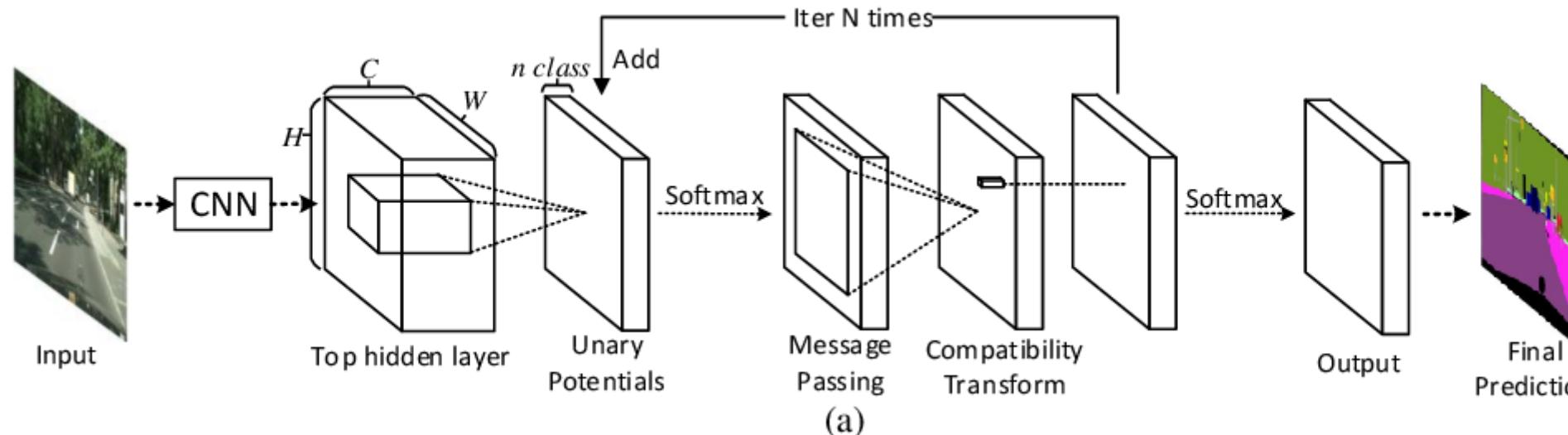
## Kekurangan

- Situasi dimana persentase outliers banyak sehingga butuh banyak iterasi dan tingkat kegagalan meningkat

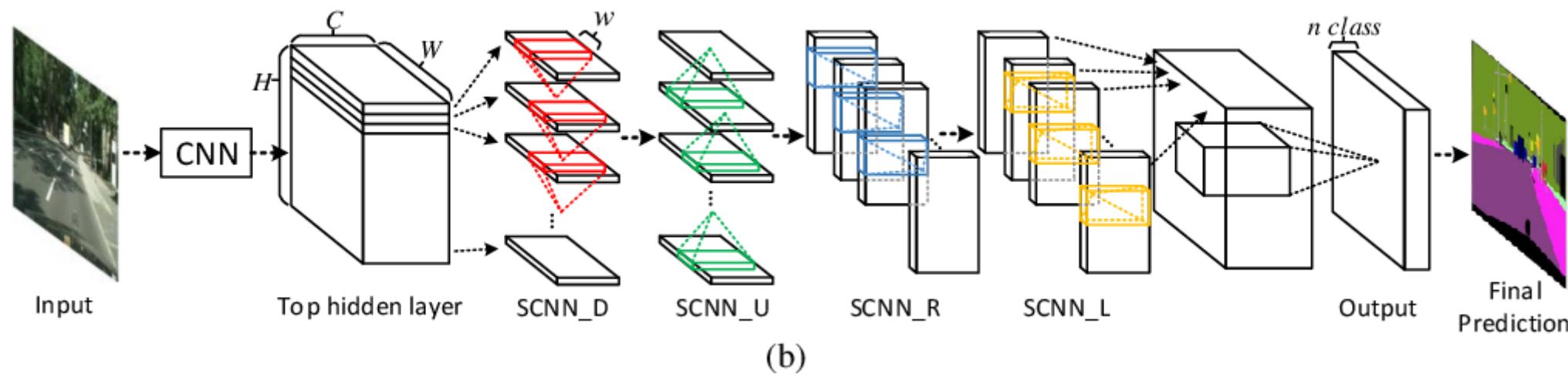
# SCNN

- Spatial Convolutional Neural Network
- CNN Klasik tidak memiliki pengertian spasial: tidak dapat menangkap relasi spasial (rotasi dan translasi)
- Relasi spatial memiliki banyak kegunaan:
  - Lampu lalu lintas umumnya memiliki relasi spasial seperti berdiri vertikal dan diletakan di kiri atau kanan jalan
  - Identifikasi lajur

# Arsitektur SCNN

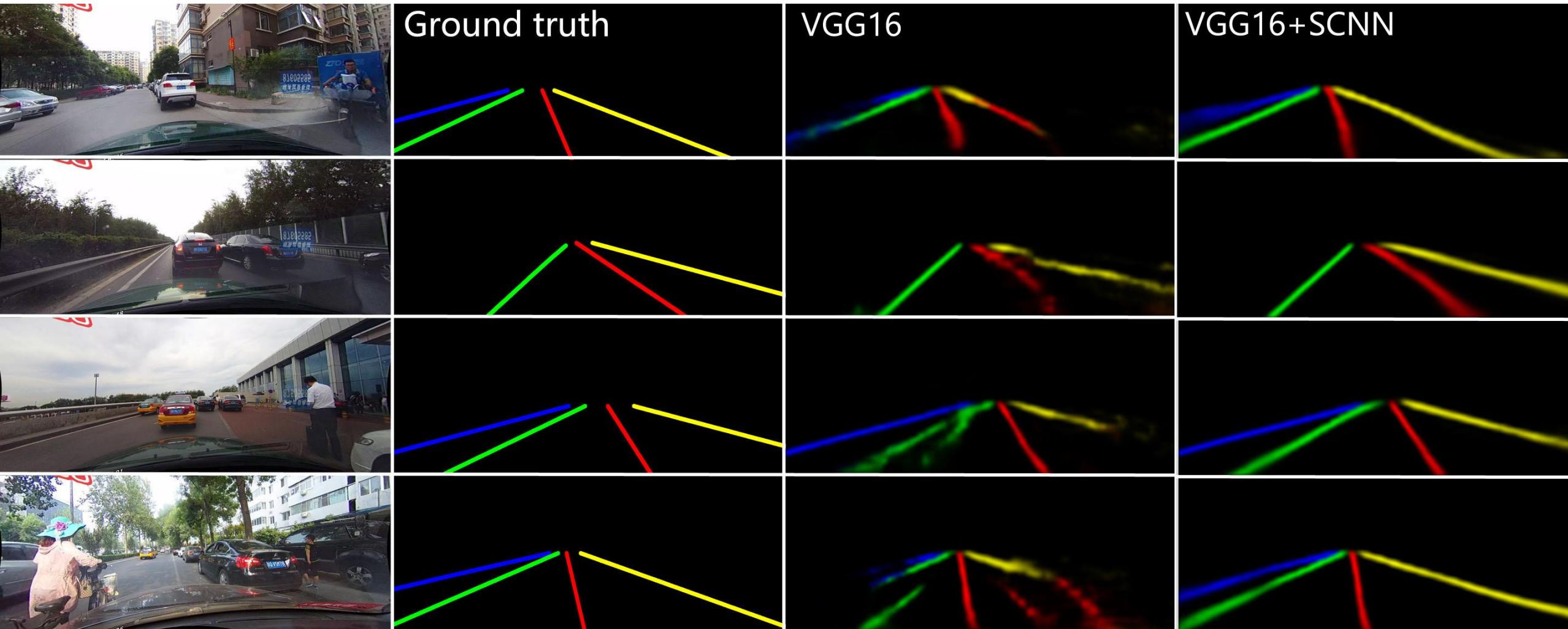


(a)



(b)

# Contoh SCNN



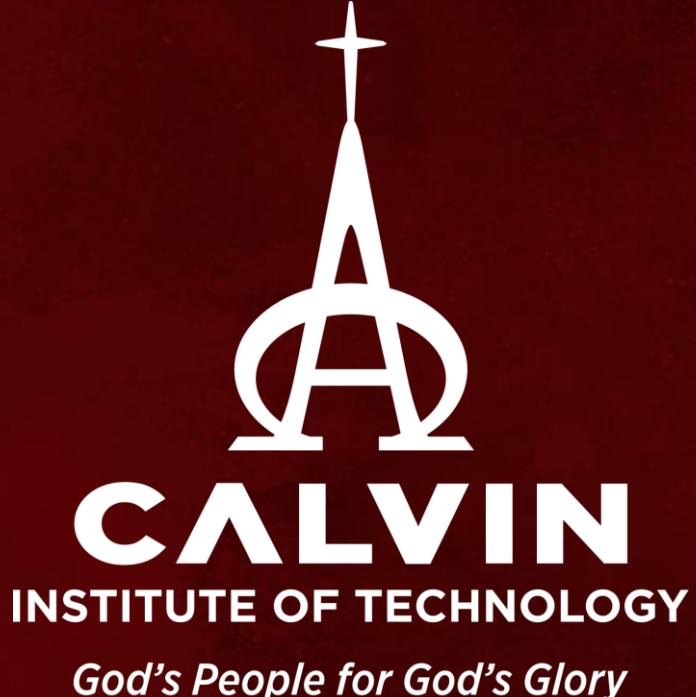
# Summary

## Edge detection

- Filter
- Turunan, konvolusi, Gaussian blur, LoG, DoG
- Canny Detection

## Line detection

- Hough transform
- RANSAC
- SCNN



# IBDA4311 Computer Vision

---

# Topik Minggu Ini dan Capaian Pembelajaran

## Topik minggu ini:

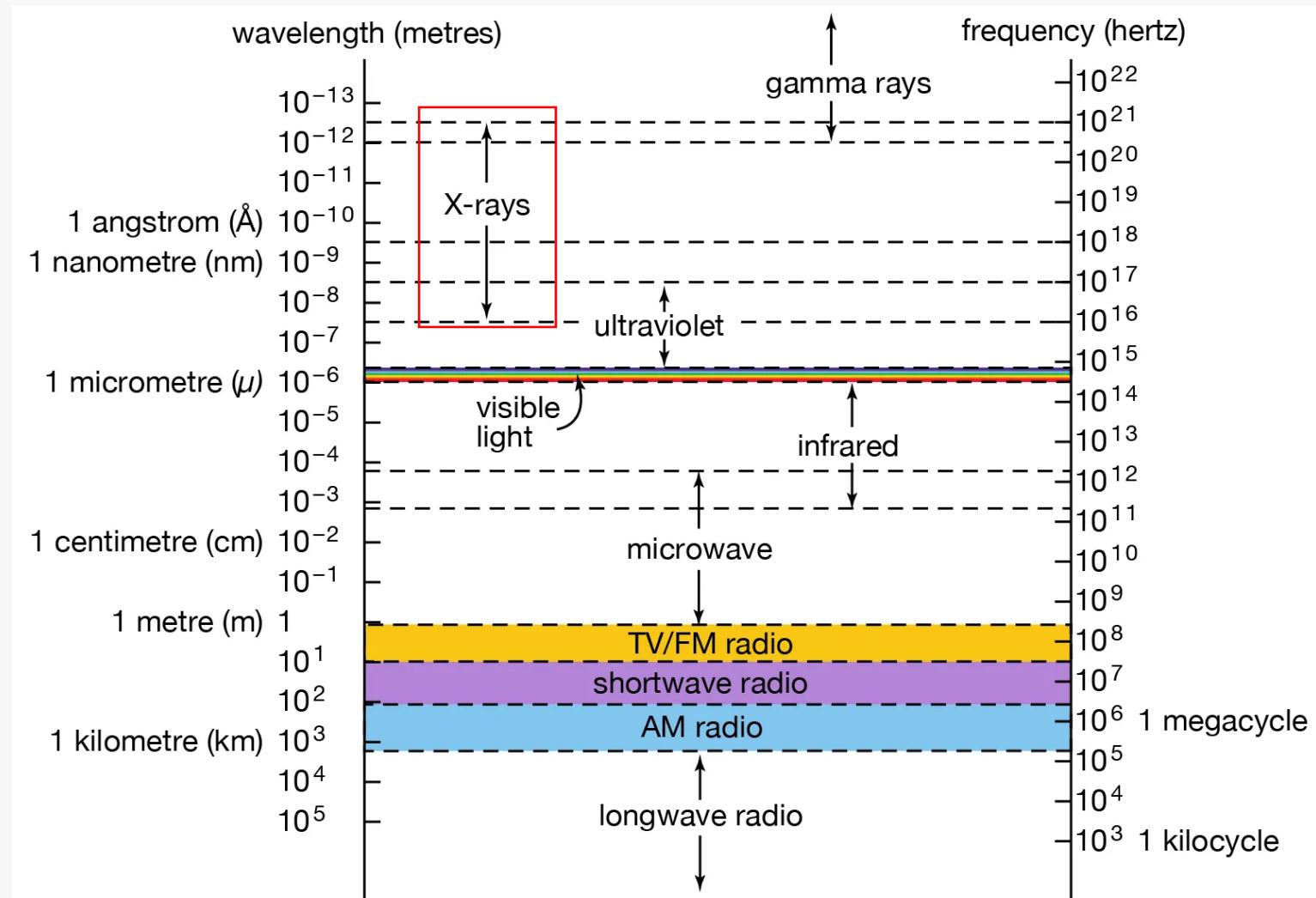
1. Pemrosesan Gambar Medis (*Medical Image Processing*)

## Capaian Pembelajaran:

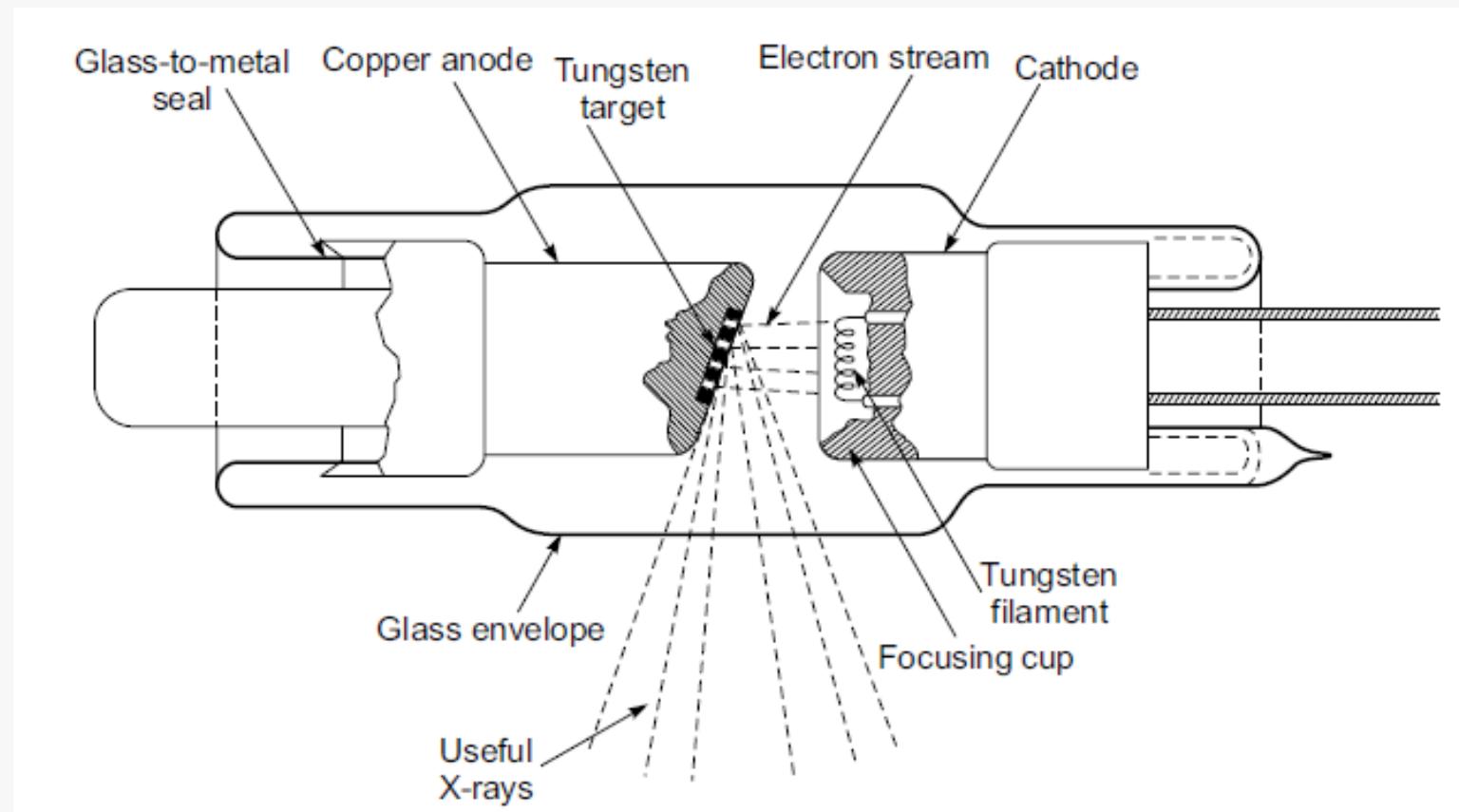
1. Mahasiswa mampu mengaplikasikan konsep-konsep pemrosesan gambar medis

# X-Ray

# X-Rays Wavelength

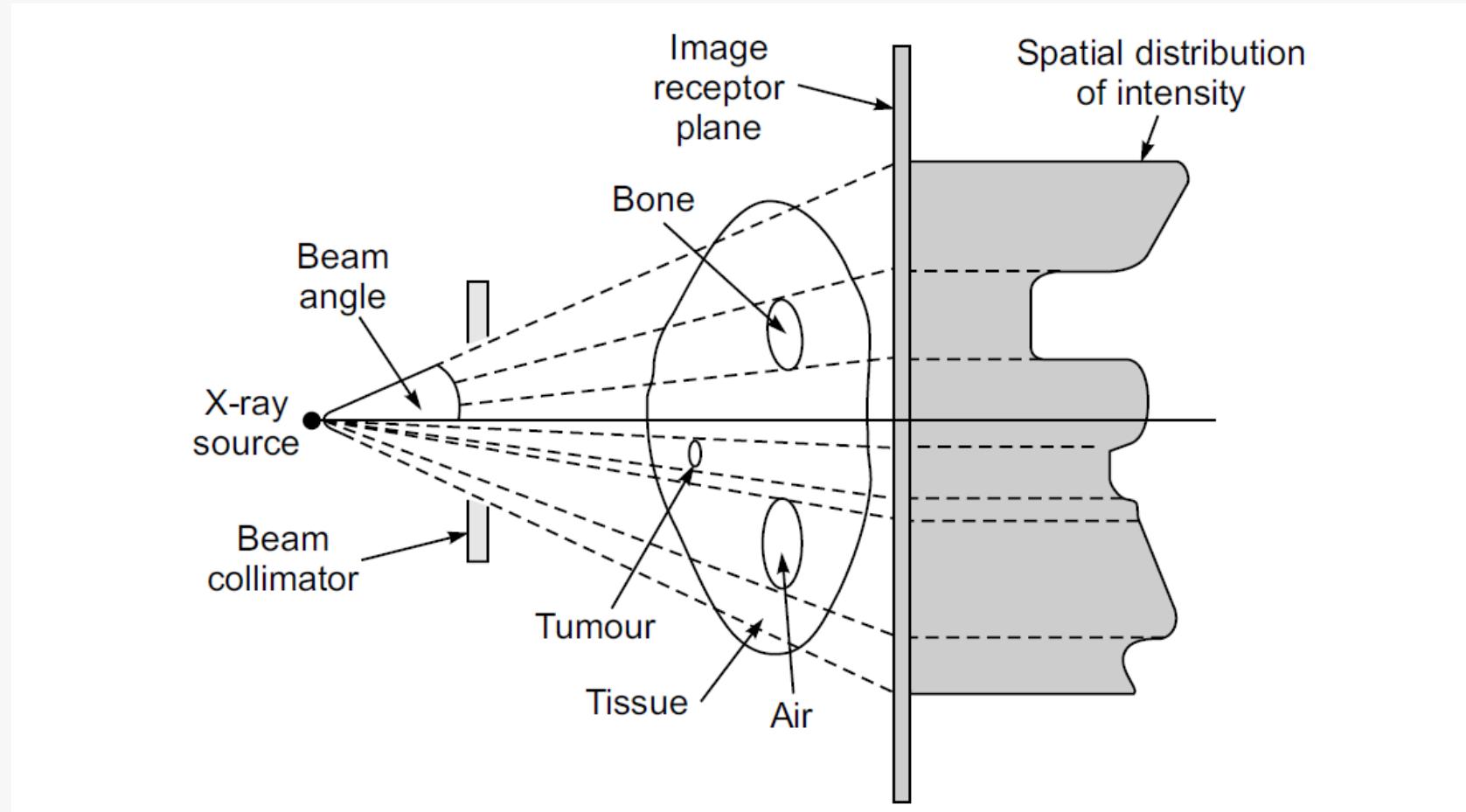


# Construction Of Stationary X-Ray Anode Tube



Sumber: "Handbook of Biomedical Instrumentation", Third Edition, Author: Dr. R. S. Khandpur, Publisher McGraw Hill Education, ISBN: 9789339205430.

# Basis Of Diagnostic Radiology



Gambar: Basic set up for a diagnostic radiology image formation process. Sumber: "Handbook of Biomedical Instrumentation", Third Edition.

# X-Ray Images



(a) Normal



(b) Bacterial Pneumonia



(c) Viral Pneumonia



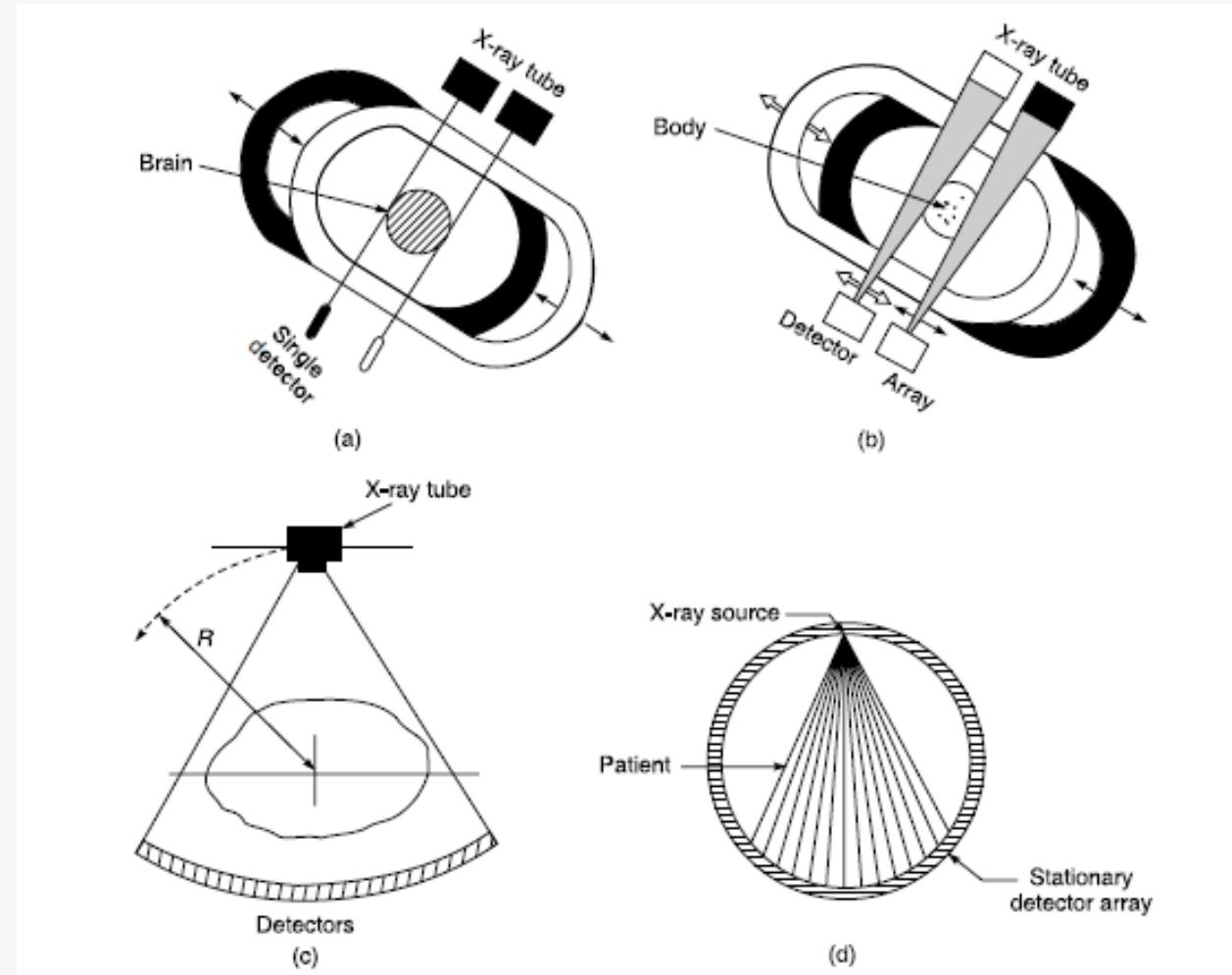
(d) COVID-19 Pneumonia



<https://images.app.goo.gl/21ZPmYHbuyPo5DCe6>  
<https://eastportdentalaz.com/common-types-of-dental-x-rays-and-why-you-need-them/>

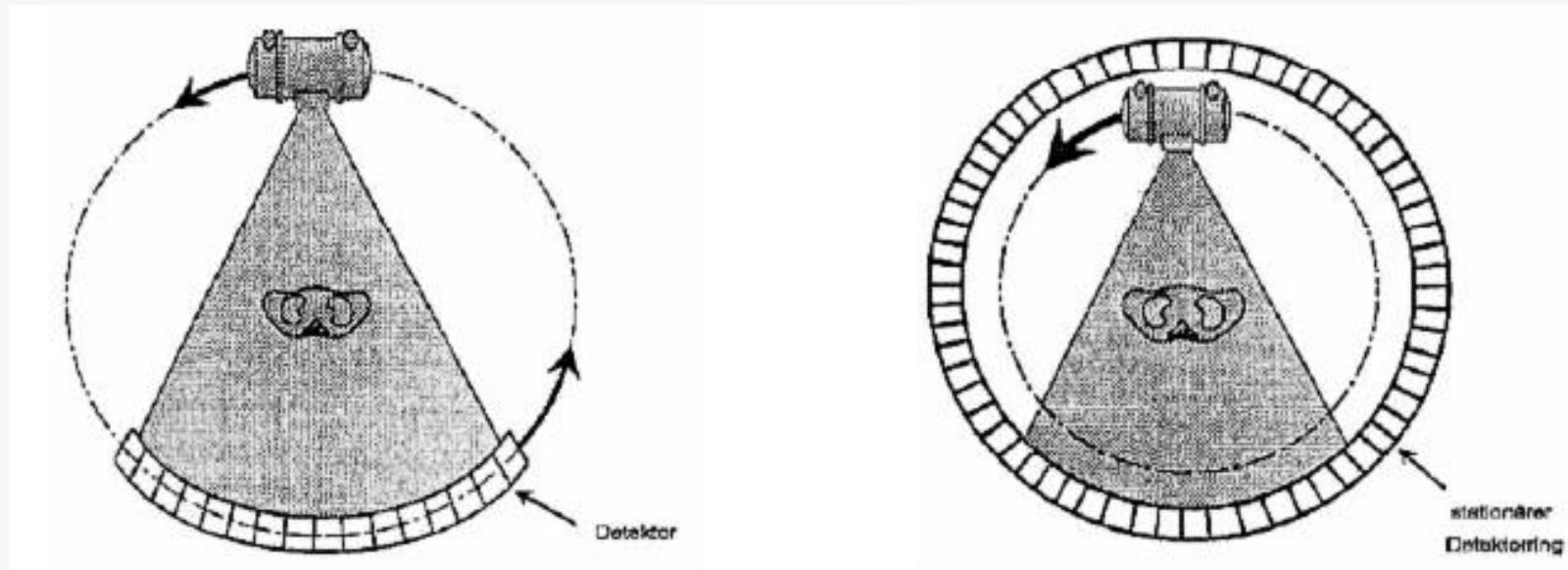
# Computed Tomography (CT)

# I. Scanning System: Scanning Arrangements



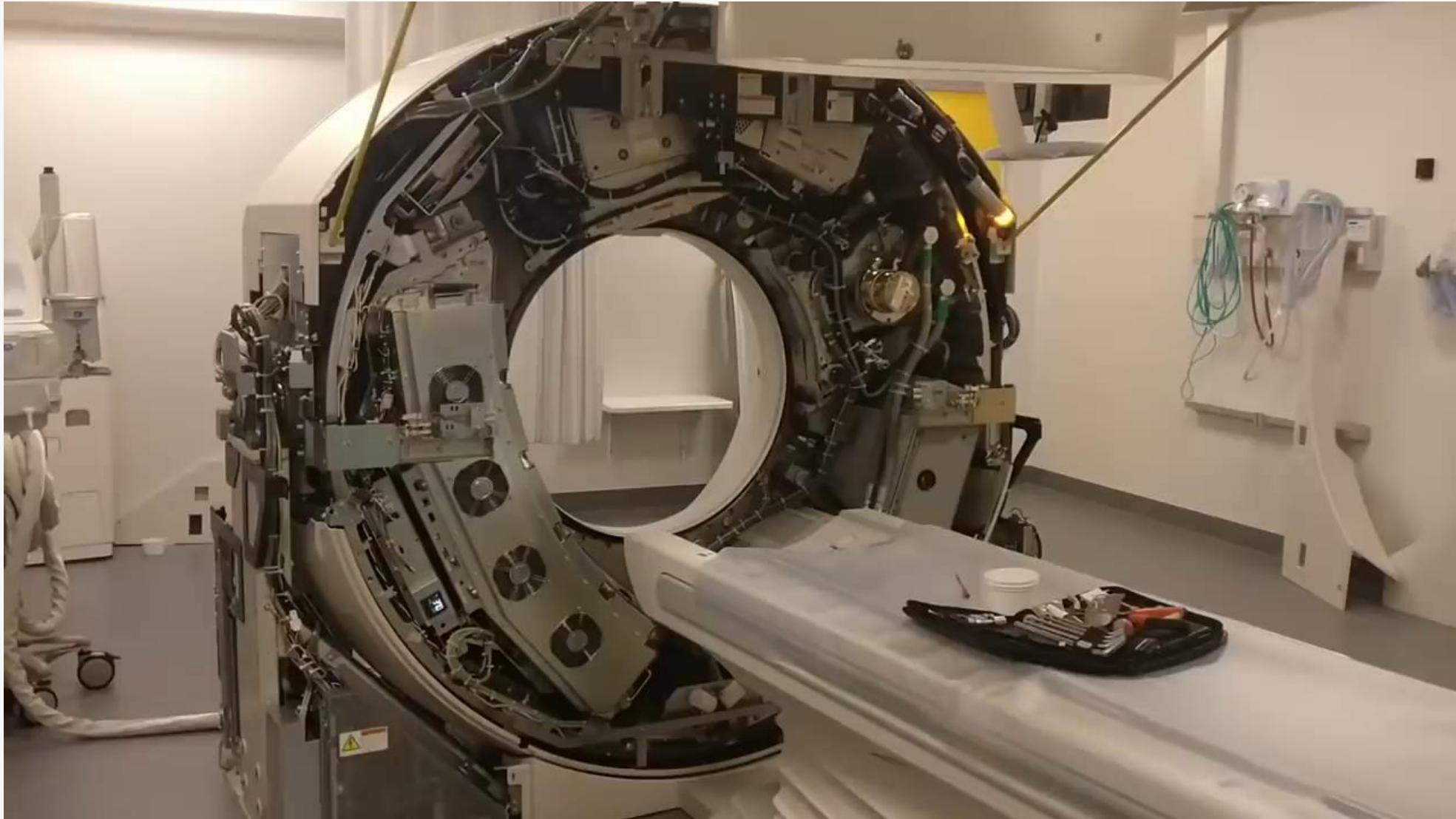
Gambar: Scanning arrangement of the early CT machines. Sumber: "Handbook of Biomedical Instrumentation", Third Edition.

## II. Processing System: Gantry Geometry



Sumber: Friedrich Ueberle

# Video: Inside Of CT Scanner At Full Speed

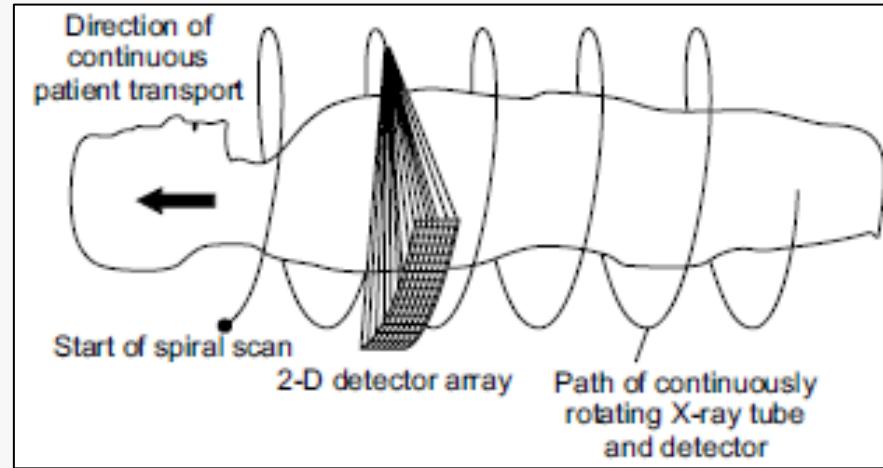


Sumber: [https://www.youtube.com/watch?v=pLajmU4TQuI&ab\\_channel=Newsflare](https://www.youtube.com/watch?v=pLajmU4TQuI&ab_channel=Newsflare)

September 26, 2023

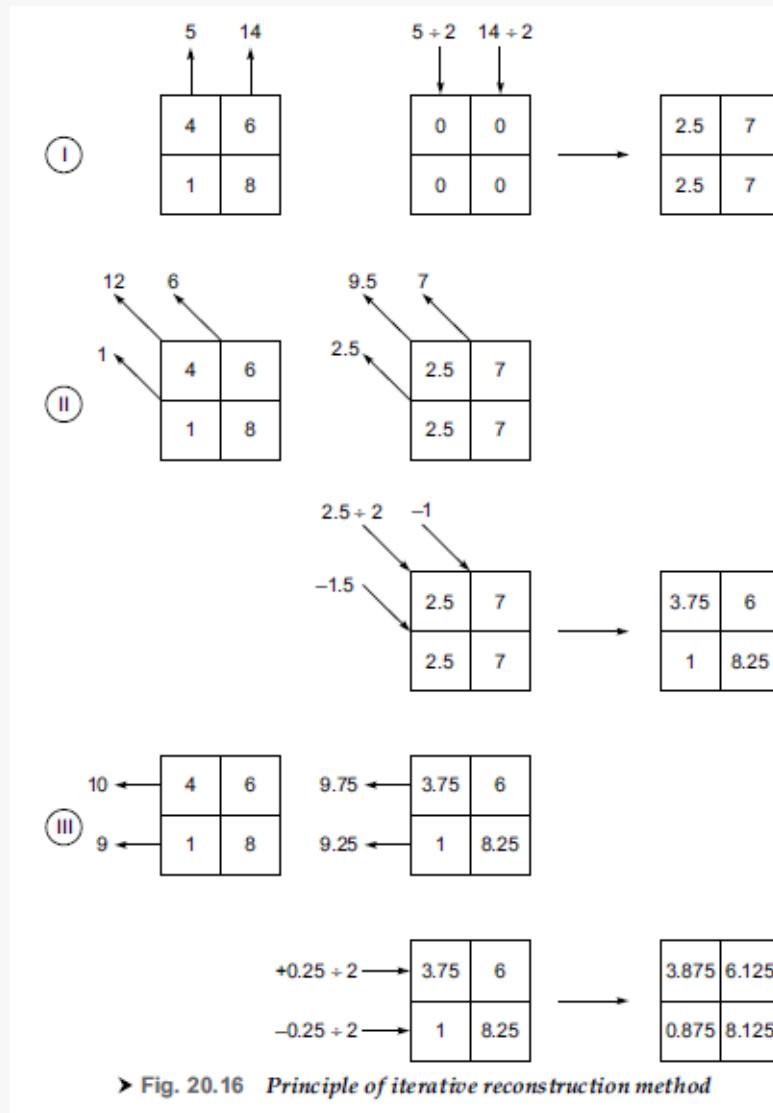
Confidential. Not to be copied / distributed

# I. Scanning System: Scanning Arrangements



Gambar: The spiral CT scan principle in multislice scanning. Sumber: "Handbook of Biomedical Instrumentation", Third Edition.

## II. Processing System: Processing Unit



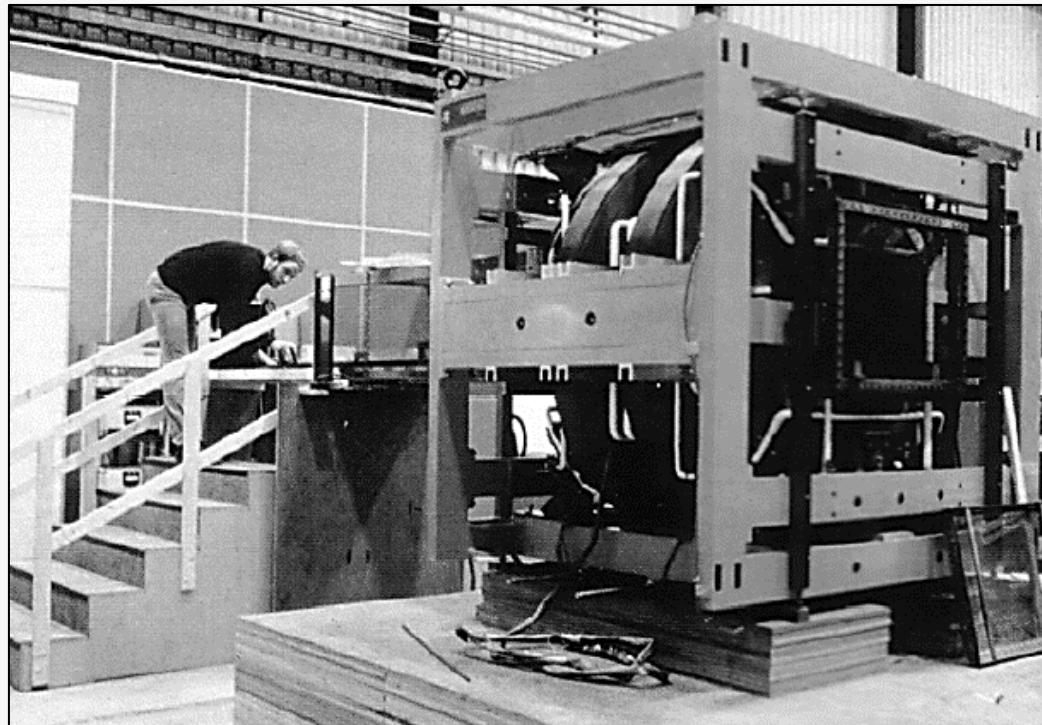
Gambar: Principle of the iterative reconstruction method. Sumber: "Handbook of Biomedical Instrumentation", Third Edition.

# CT Images



# Magnetic Resonance Imaging (MRI)

# History Of MRI



Gambar: The 1<sup>st</sup> 0.15 Tesla MRI Prototype from Philips was put into operation in 1979. Sumber: Friedrich Ueberle

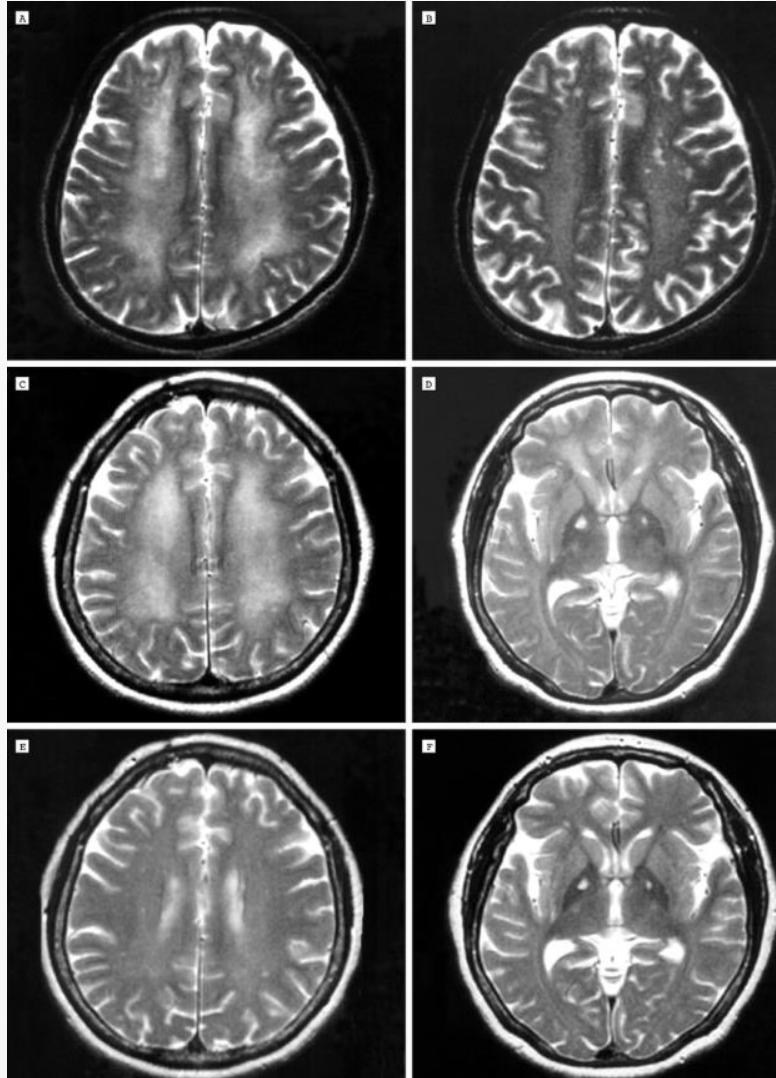


Gambar: 4th Generation and High Power 3 Tesla MRI scanner Skyra Magnetom from Siemens.  
Sumber: <https://www.metropolitan-general.gr/en/media2/photos/2016-ultra-modern-3-tesla-m-r-i-machine>

# Video: How MRI Machine Works



# MRI Images



[https://www.researchgate.net/figure/A-A-T2-weighted-magnetic-resonance-imaging-MRI-scan-obtained-1-month-after-carbon\\_fig2\\_12392282](https://www.researchgate.net/figure/A-A-T2-weighted-magnetic-resonance-imaging-MRI-scan-obtained-1-month-after-carbon_fig2_12392282)

# Ultrasonic Imaging Systems

# Ultrasound Procedures



<https://www.medicalnewstoday.com/articles/245491>

# Ultrasound Images



# Impedansi Karakteristik

Impedansi karakteristik atau impedansi akustik spesifik suatu medium didefinisikan sebagai produk kerapatan medium dengan kecepatan suara dalam medium yang sama.

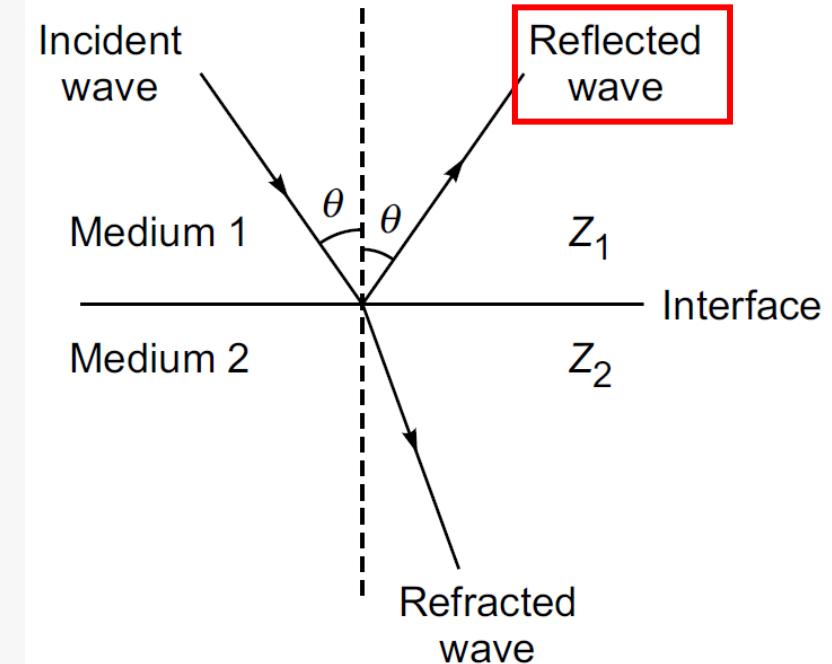
$$z = \rho V$$

where  $z$  = specific acoustic impedance  
 $\rho$  = density of the medium  
 $V$  = velocity of sound in the medium.

Impedansi karakteristik menentukan derajat pemantulan dan pembiasaan pada antarmuka antara dua media. Persentase energi gelombang datang yang dipantulkan digambarkan oleh rumus berikut ini:

$$\left[ \frac{z_1 - z_2}{z_1 + z_2} \right]^2 \times 100\%$$

where  $z_1$  = acoustic impedance of medium 1  
 $z_2$  = acoustic impedance of medium 2

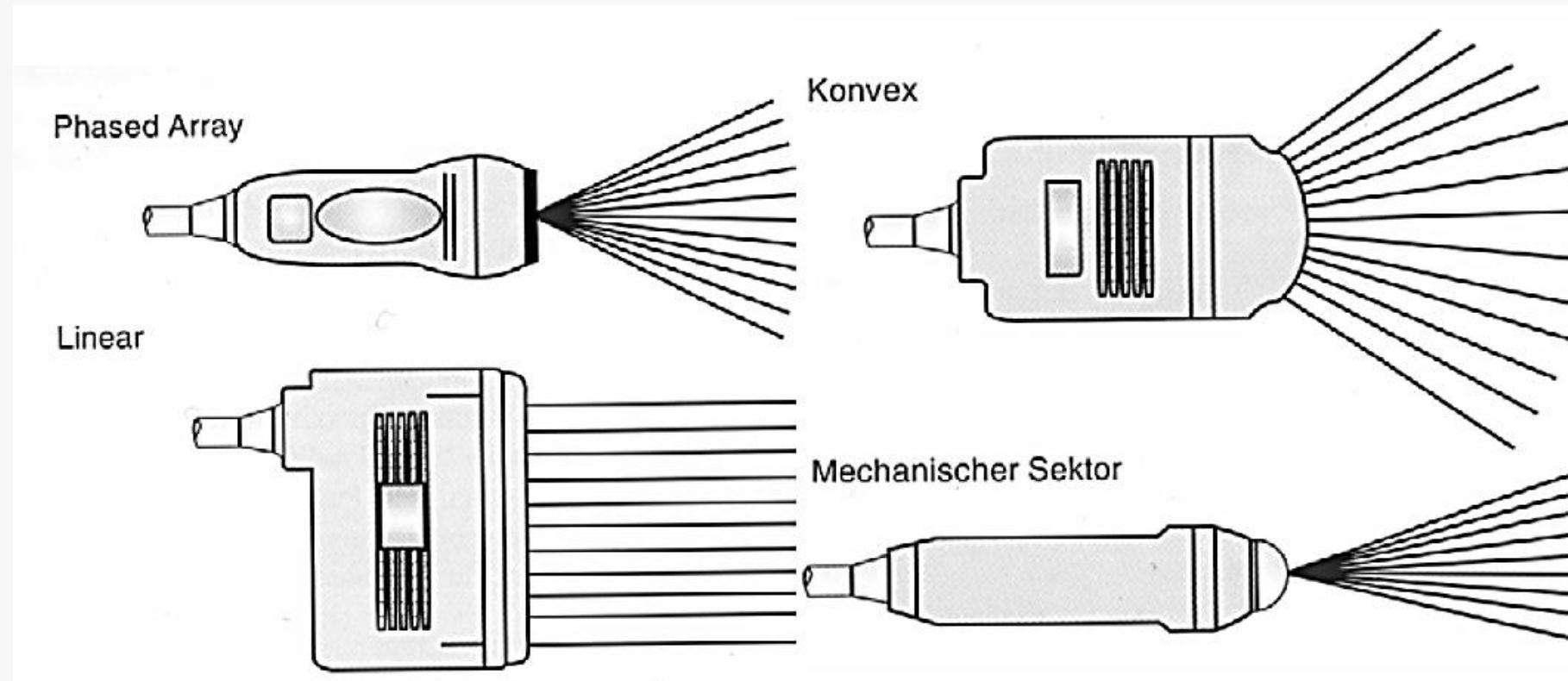


Gambar: Pemantulan dan pembiasaan ultrasonik pada antarmuka antara dua media yang memiliki impedansi akustik yang berbeda.  
Sumber: Handbook of Biomedical Instrumentation

# Anatomy of USG

**Probe dengan prinsip konverter yang berbeda:**

- **Phased Array:** Kontrol fokus dan arah dilakukan sepenuhnya oleh elektronik. Semua elemen dinyalakan bersama-sama sekaligus
- **Linear Array:** Grup-grup dari elemen dinyalakan secara bergantian
- **Konvex Array:** Grup-grup dari elemen dinyalakan secara bergantian seperti Linear Array, tetapi pancaran berbentuk kipas
- **Mechanical Sector:** Elemen tunggal berporos motor



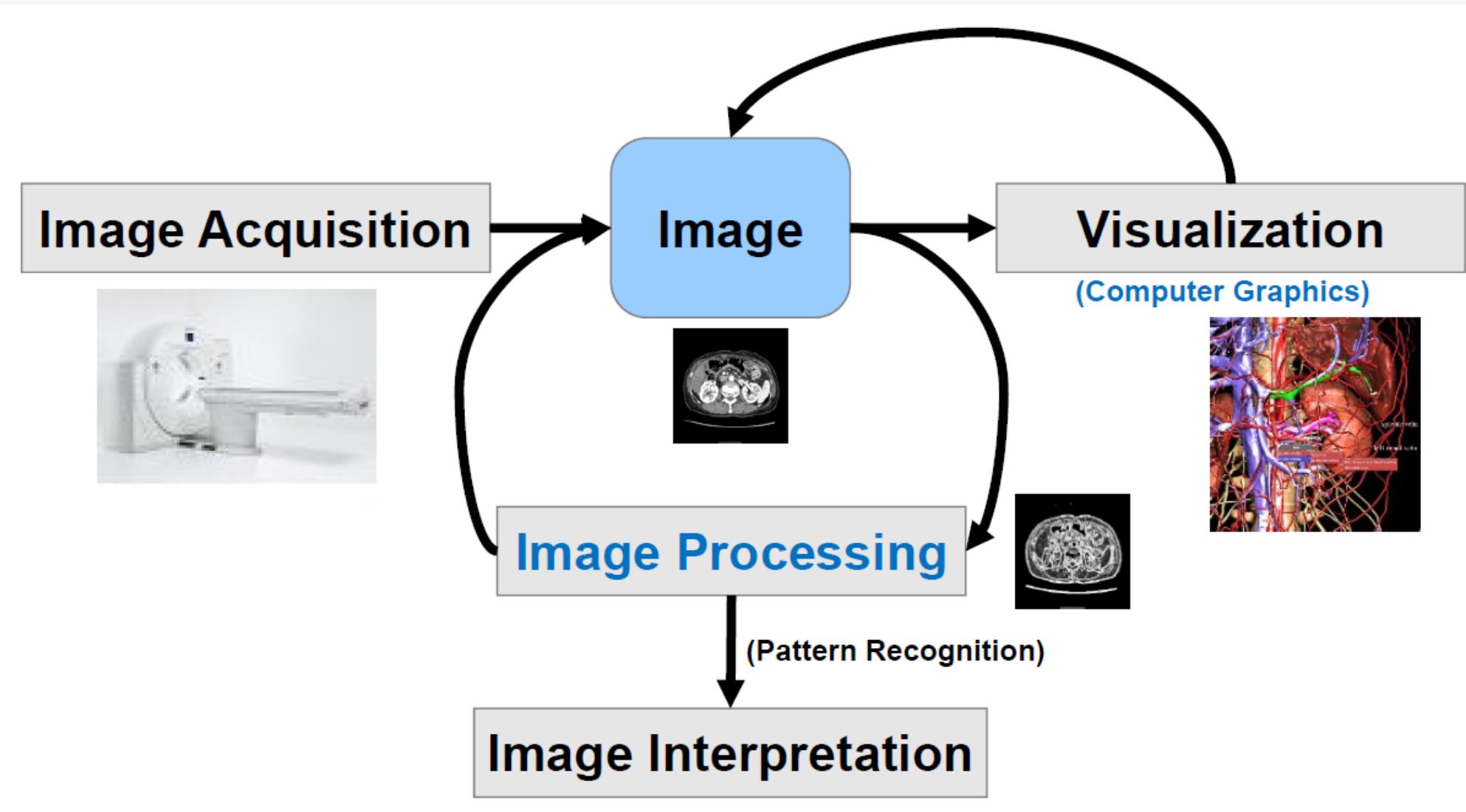
Sumber: Friedrich Ueberle

September 26, 2023

Confidential. Not to be copied / distributed

# Medical Image Processing

# Image Workflow



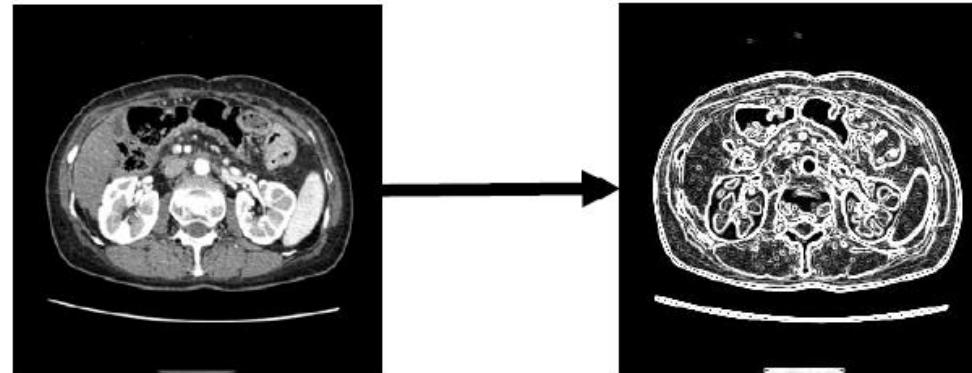
Courtesy: Prof. Dr.-Ing. Thomas Schiemann

September 26, 2023

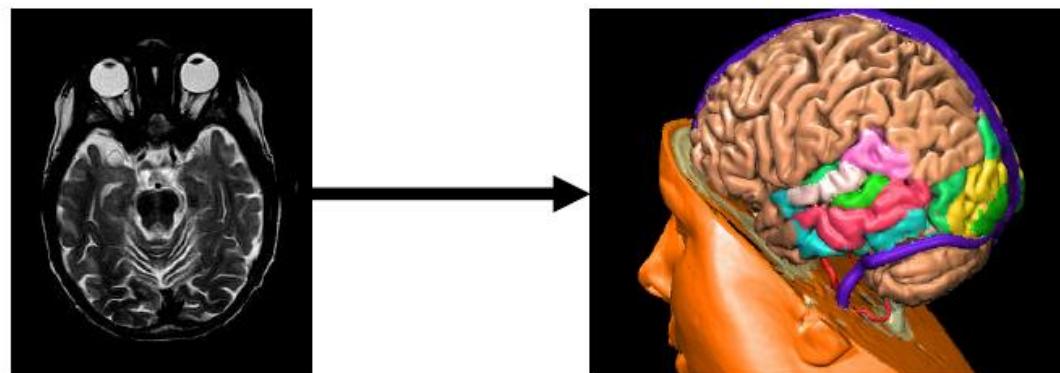
Confidential. Not to be copied / distributed

## 2 Examples

- Image Processing: Edge Detection



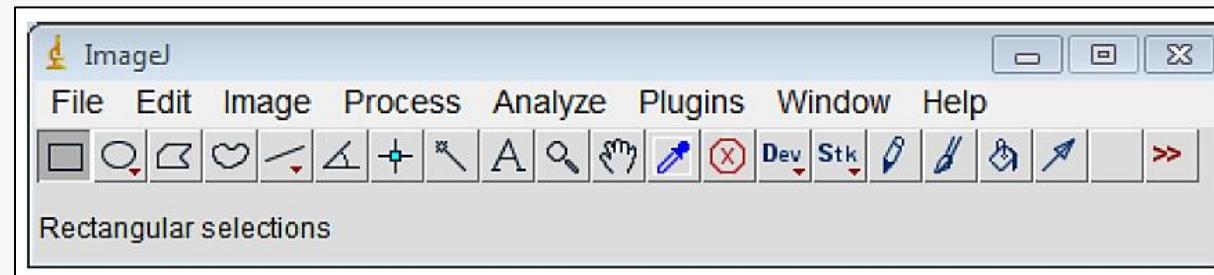
- Computer Graphics: 3D Visualization of MRI



VOXEL-MAN, University Hospital Hamburg-Eppendorf

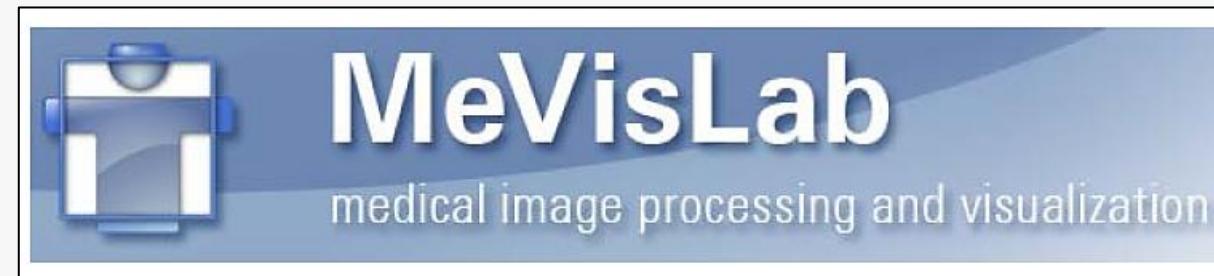
# ImageJ -> Install

- Image processing software with emphasis on medical and scientific applications
- Set of basic functions available
- Written in Java, offers Programming interface for own methods (**Plugins**)
- Open source: <http://rsbweb.nih.gov/ij>
- Download 64-bit Java for Windows: <https://www.java.com/en/download/>

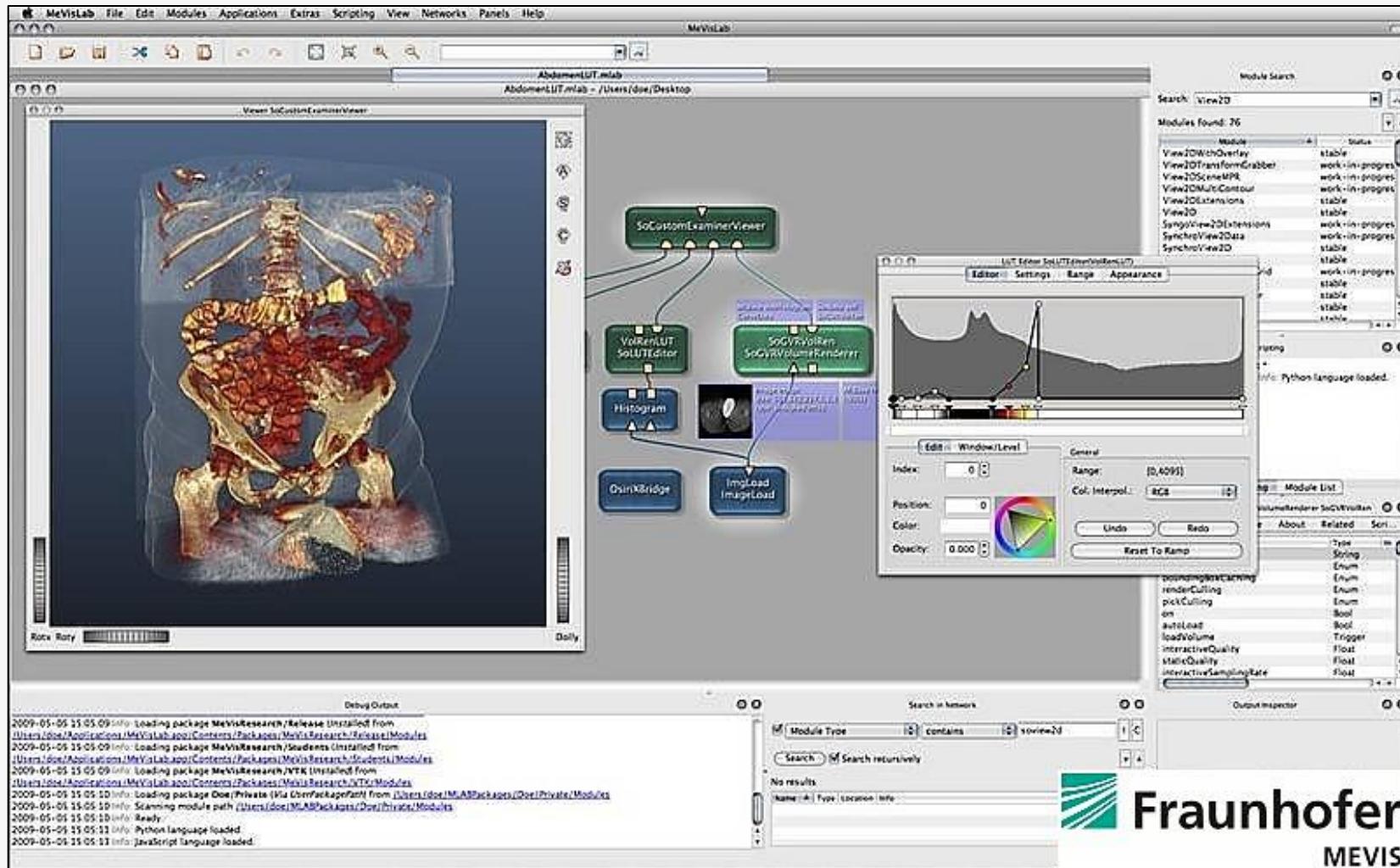


# MeVisLab -> Install

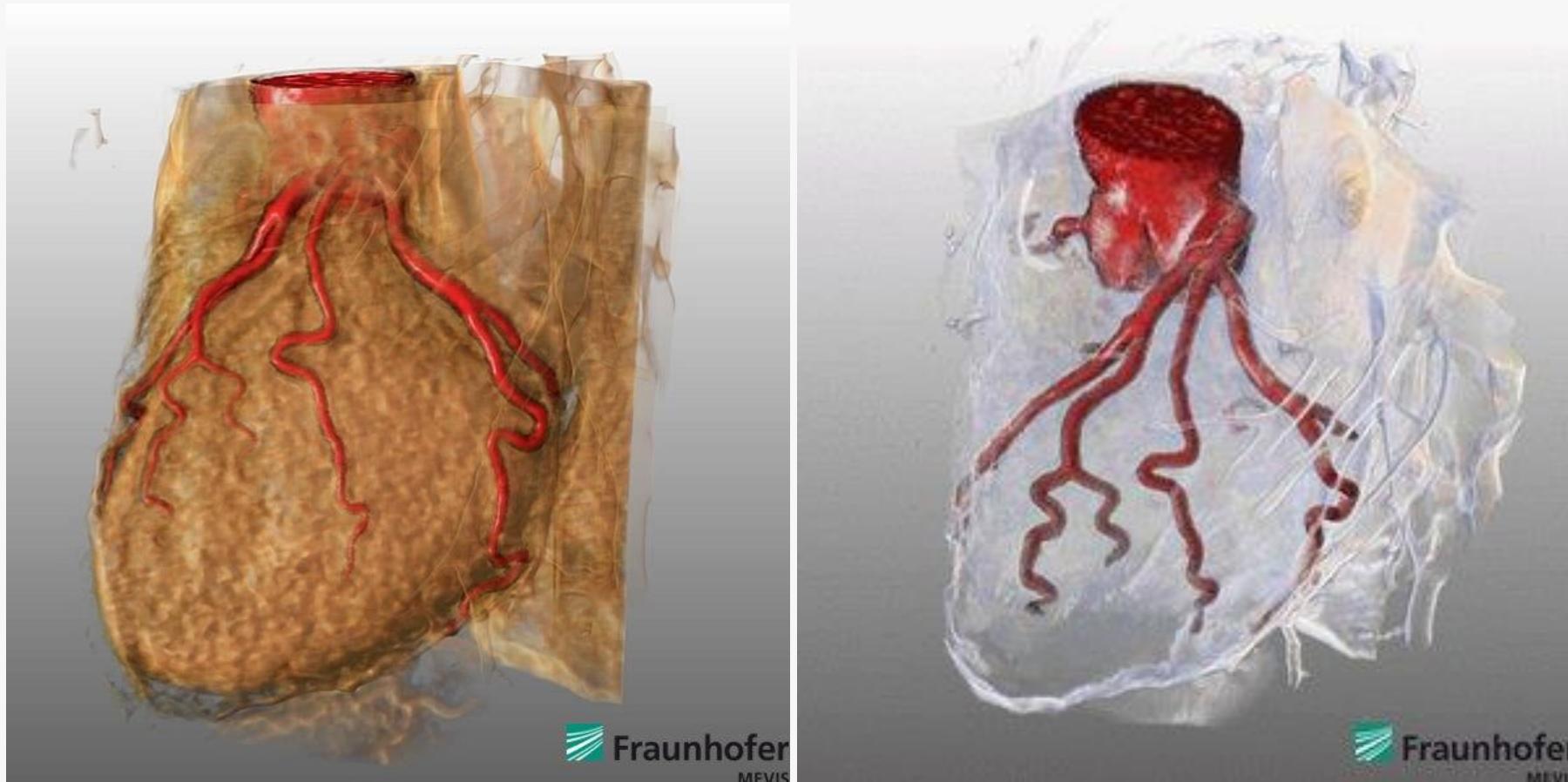
- Professional software for medical image processing and visualization.
- Programming interface for own methods (**modules**) in C++.
- Graphical programming of module networks
- Free license: <http://www.mevislab.de/download>
- <https://ij.imjoy.io/>



# MeVisLab

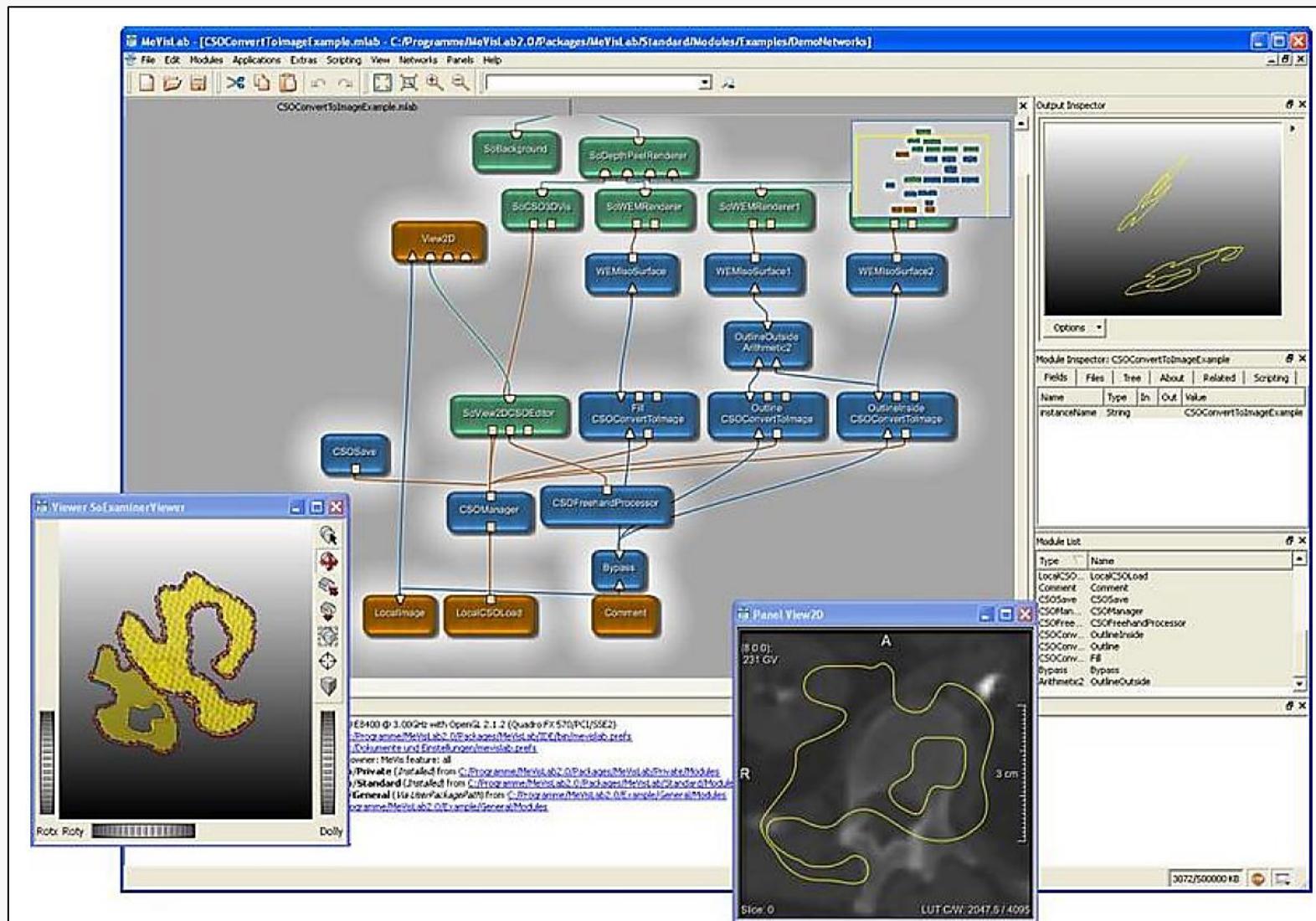


# MeVisLab



<https://en.wikipedia.org/wiki/MeVisLab>

# MeVisLab

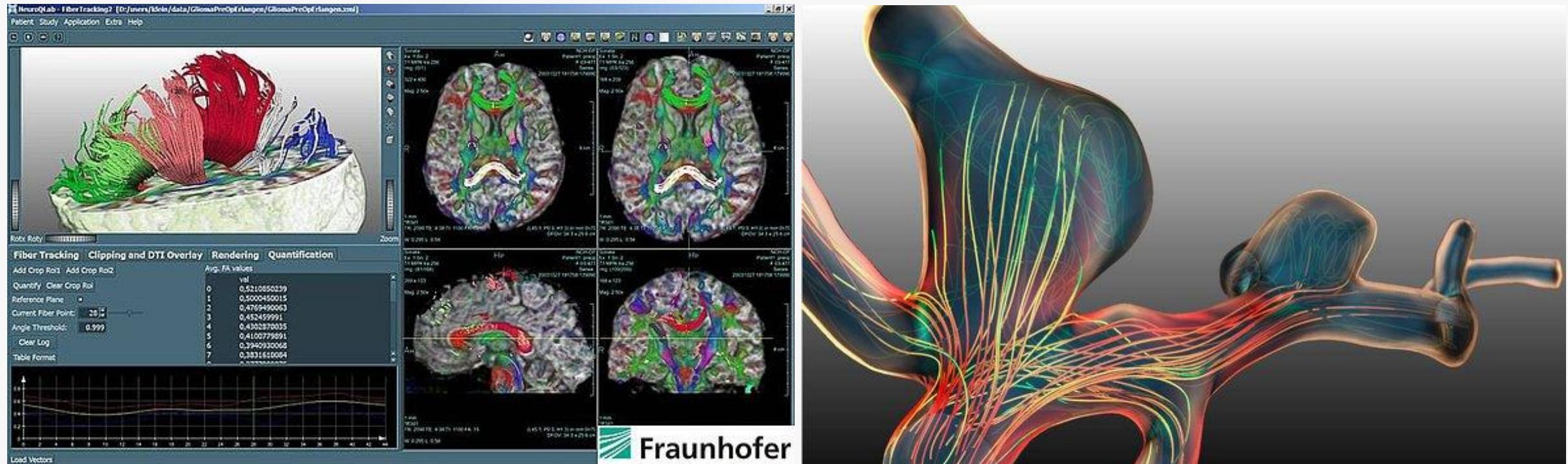


Source: Mevis Medical Solutions AG

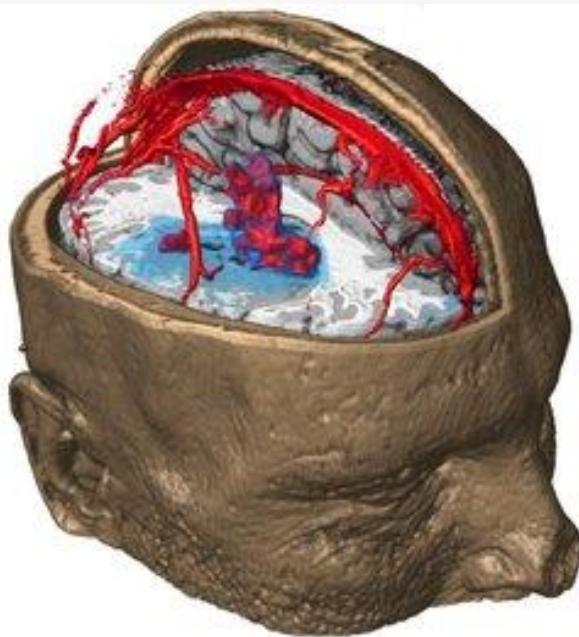
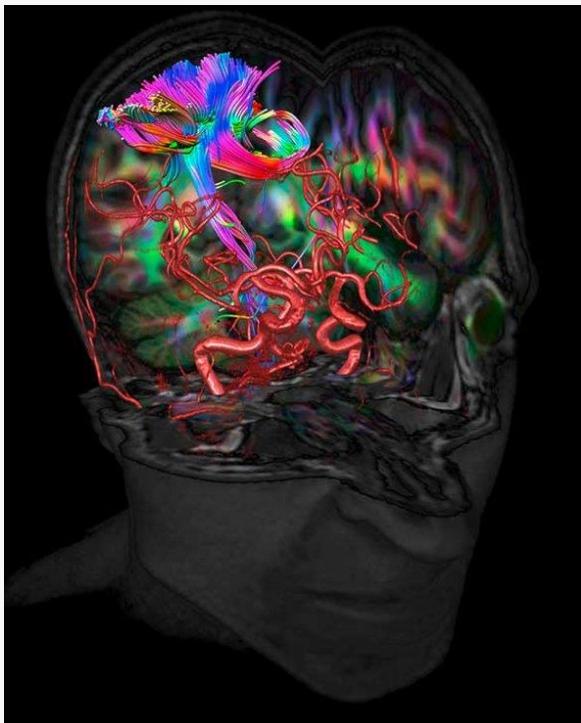
September 26, 2023

Confidential. Not to be copied / distributed

# MeVisLab



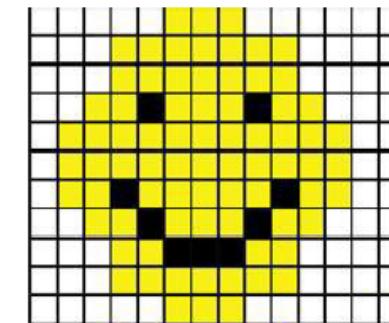
# MeVisLab



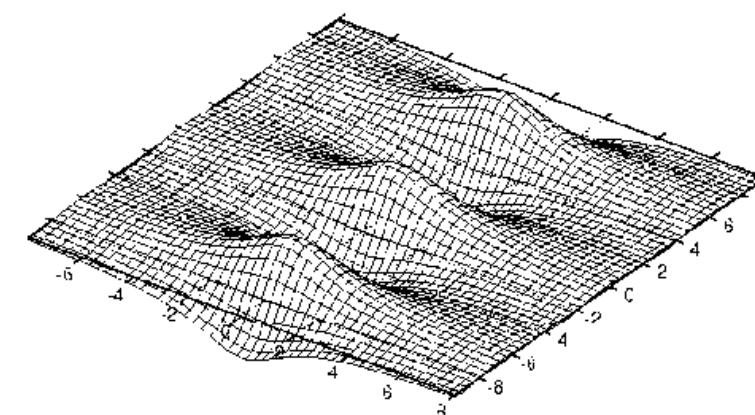
Fraunhofer  
MEVIS

# What is an image?

- Grid or array of pixels (picture elements)
  - each pixel carries intensity or color information



- 2D (diskrete) function  $f(x,y)$ 
  - function value: intensity or color information



# Binary and gray value images

- 1 Bit: black and white = binary images



- 8 Bit: 256 gray values (0-255)
- 12 Bit: 4096 gray values  
used in CT: Hounsfield units -1024...3071
- 16 Bit: 65536 gray values



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Histograms

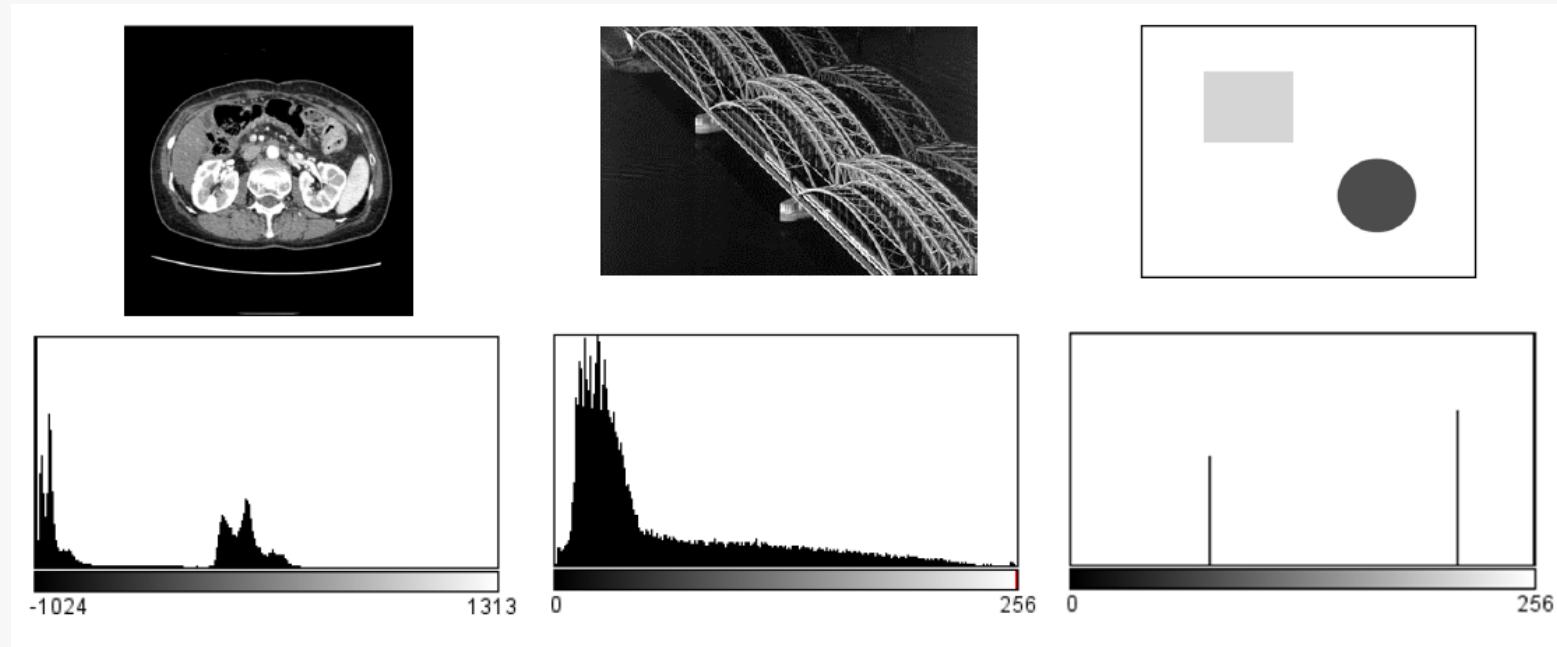
- Histograms show frequencies of intensities (“How many pixels are in the image with a certain intensity?”)
- The information of a histogram can basically be represented in a list:

Intensity	Number of pixels
0	25898
1	7834
2	35
...	...

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Examples

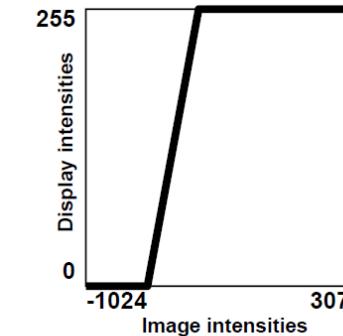
- Usually, histograms are displayed as bar-diagrams



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

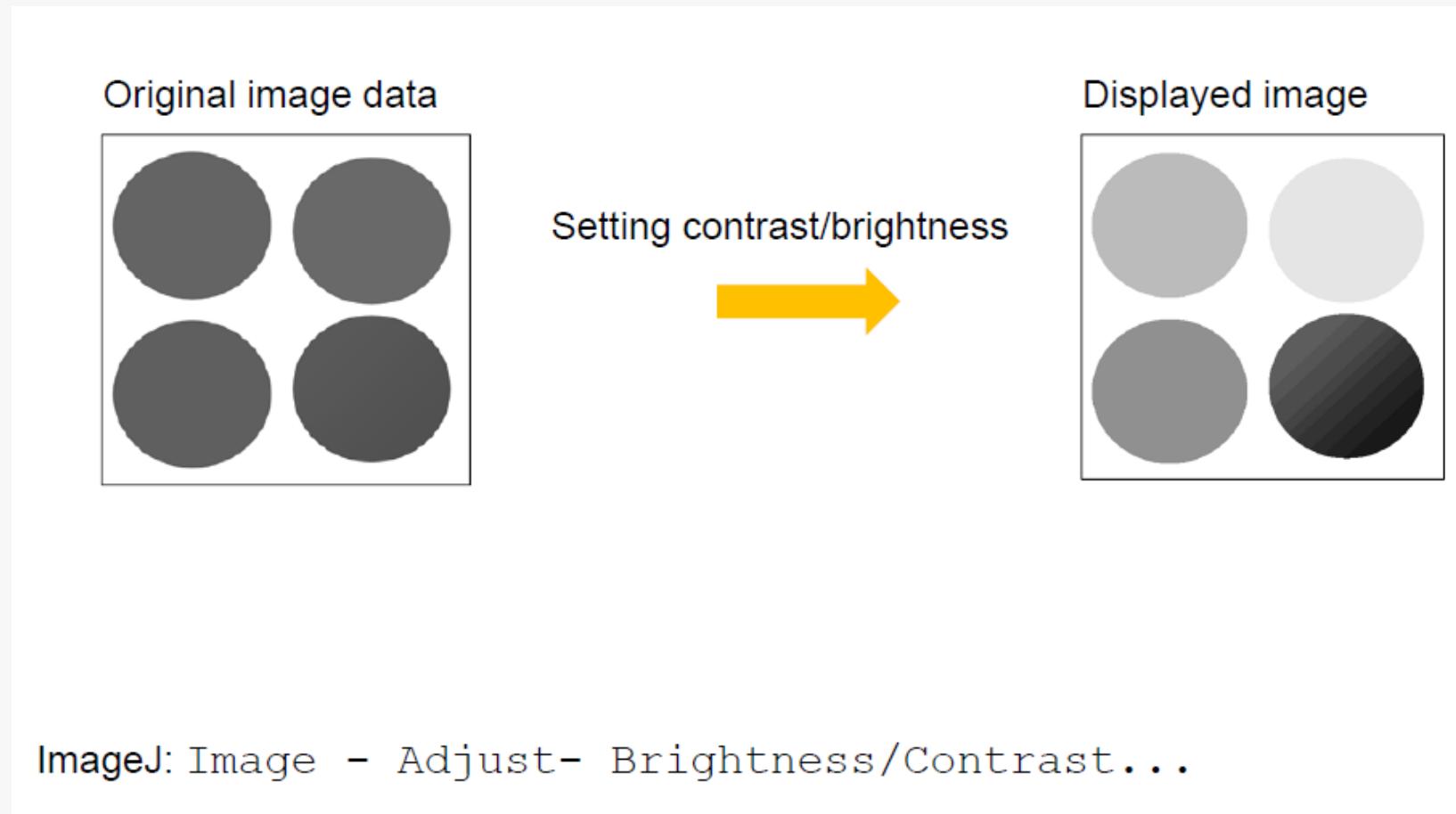
# Histogram based operations

- Selection of an intensity interval to be mapped onto the maximum and minimum intensities, which can be displayed,  
usually  $\text{max}=0$ ,  $\text{min}=255$ .
- Setting
  - maximum and minimum
  - brightness and contrast
  - level and window (“medical dictron”)

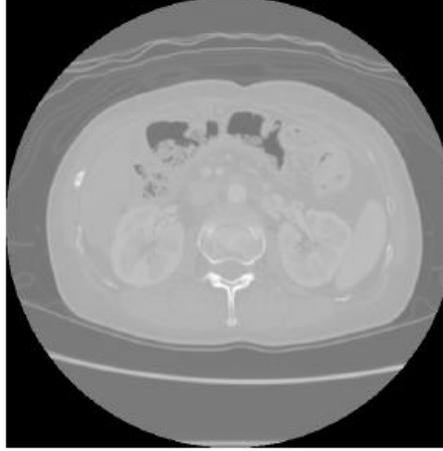


Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Example



# Computer Tomography



Original  
data



„clothes window“



soft-tissue window



bone window

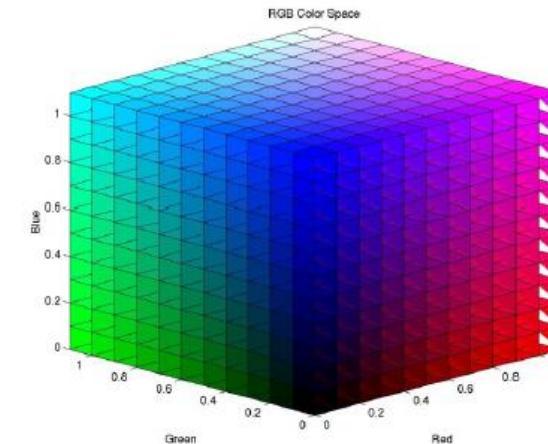
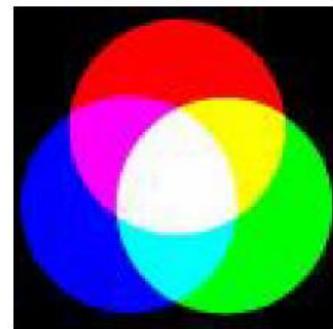
Courtesy: Prof. Dr.-Ing. Thomas Schiemann

September 26, 2023

Confidential. Not to be copied / distributed

# RGB-Color-Model

- Colors are defined by addition of the three basic colors red, green and blue
- grey/white = red+green+blue in equal percentages
- colors =  $p_1^*\text{red} + p_2^*\text{green} + p_3^*\text{blue}$

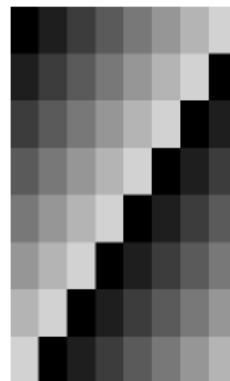


# Lookup-Tables (LUT)

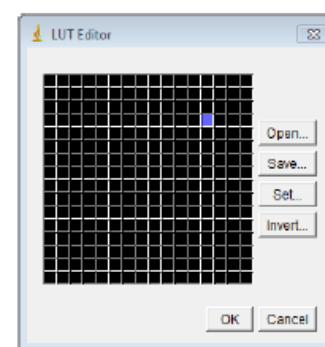
- Lookup-tables assign a new intensity or color to each original intensity

Example 1:

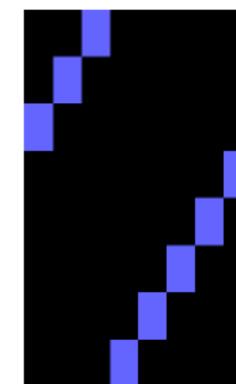
Original image data



Lookup table

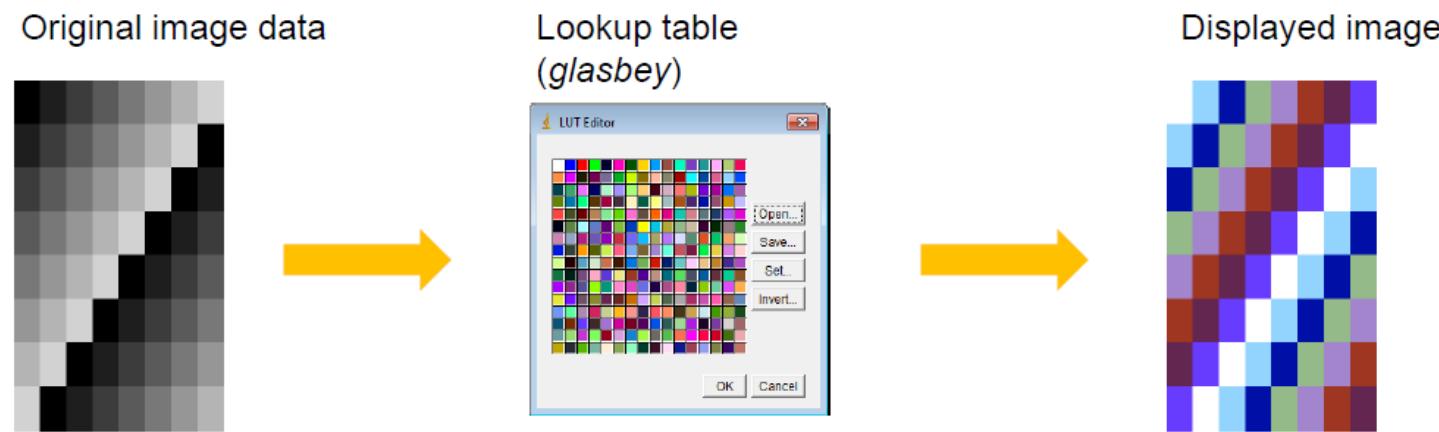
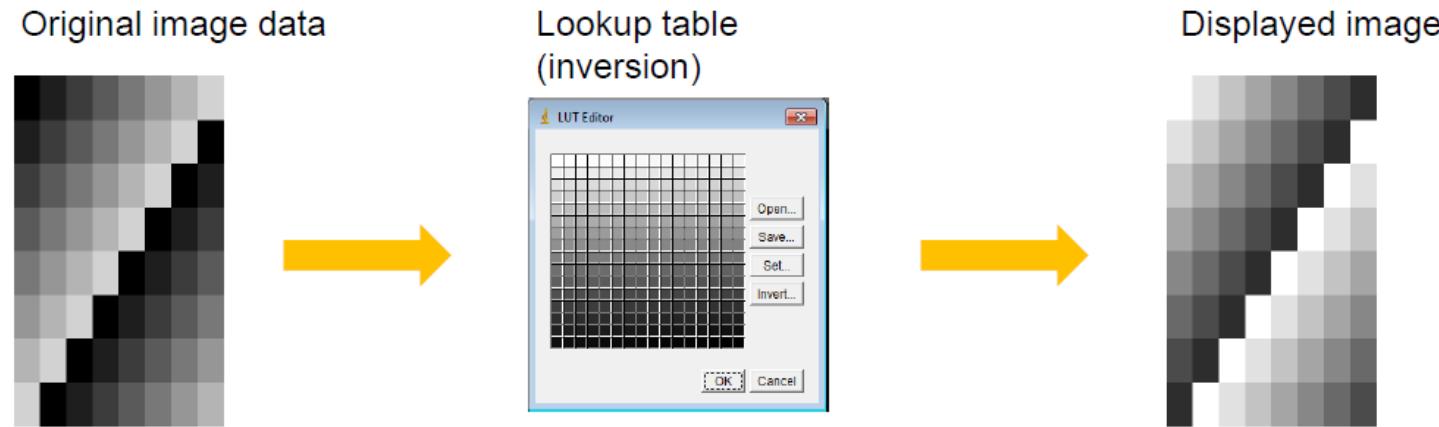


Displayed image

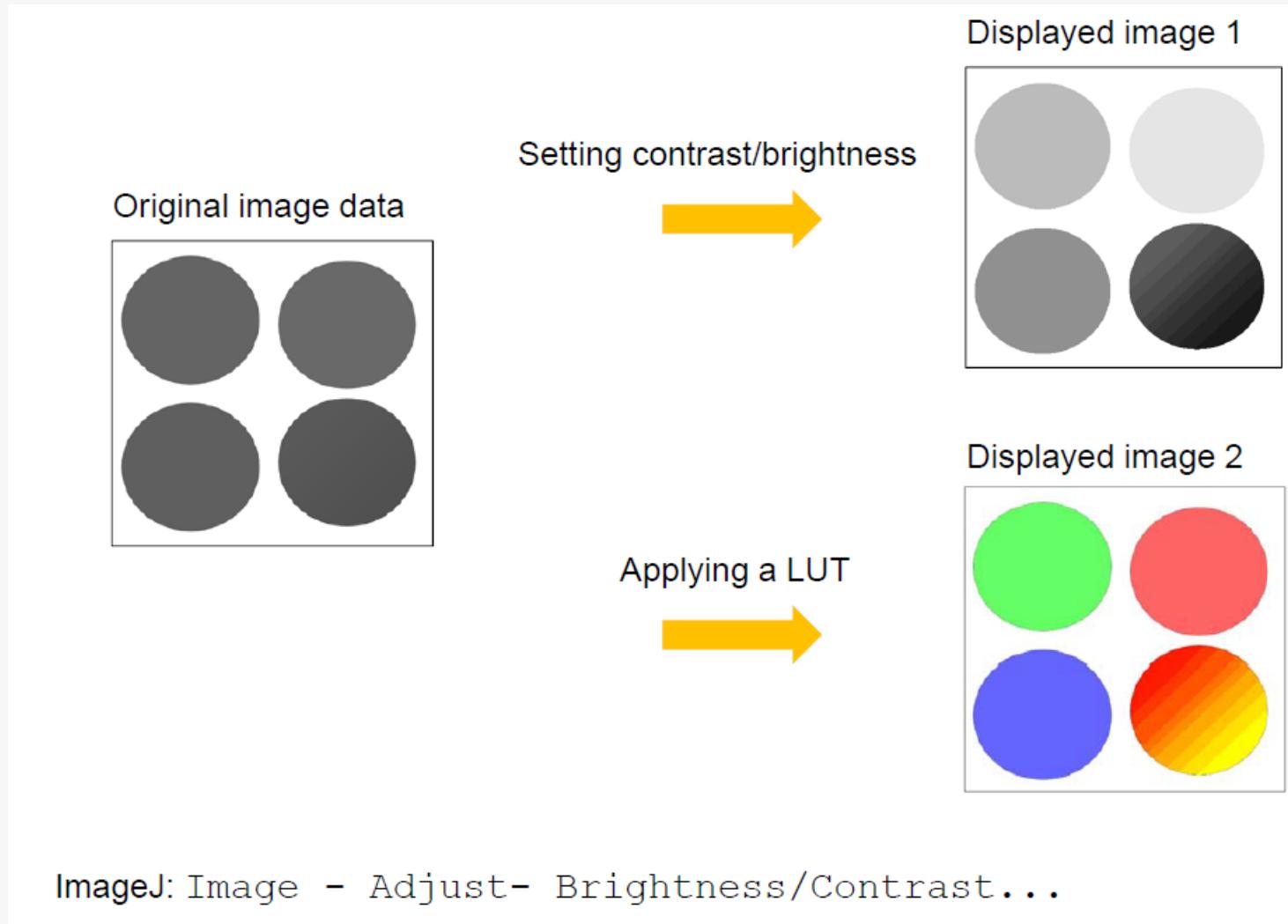


ImageJ: Image - Color - Edit LUT...

# Examples



# Examples



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

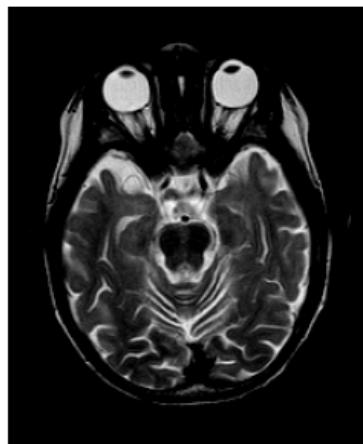
September 26, 2023

Confidential. Not to be copied / distributed

# Threshold

- Gray value images  $I(x,y)$  can be converted to binary images  $B(x,y)$  by setting a lower threshold  $t_1$  and an upper threshold  $t_2$  ( $t_1 \leq t_2$ ):

$$B(x,y) = \begin{cases} 1, & \text{if } t_1 \leq I(x,y) \leq t_2 \\ 0, & \text{otherwise} \end{cases}$$



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

September 26, 2023

Confidential. Not to be copied / distributed

# Latihan

Kerjakanlah Latihan Minggu 5\_1

---

# Medical Image Processing: Segmentation Problem

# The Segmentation Problem

- Questions on an image of the abdomen:
  - Does pixel (40,320) belong to the liver?
  - What is the area/volume of the liver?
  - What is the center of mass of the liver?
  - What is the distance of the liver from the spine?
  - How many different anatomical structures can be seen on the image?
- We need a layer of additional information.

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# The Segmentation Problem

- Basis: Assigning labels (numbers) to groups of connected pixels.

**HAW**

Binary image

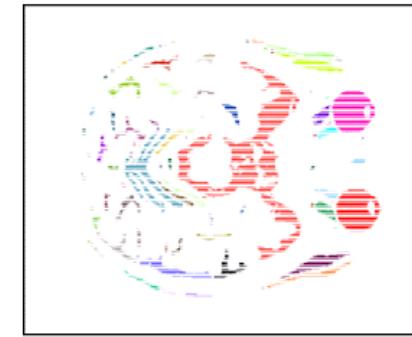
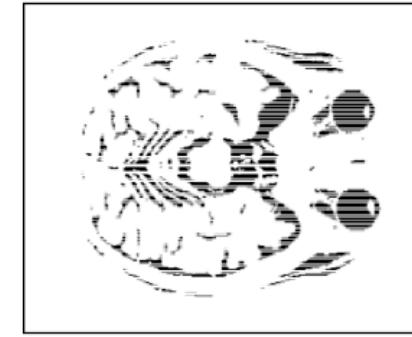
**HAW**

Labels shown as grey values

**HAW**

Labels shown with LUT

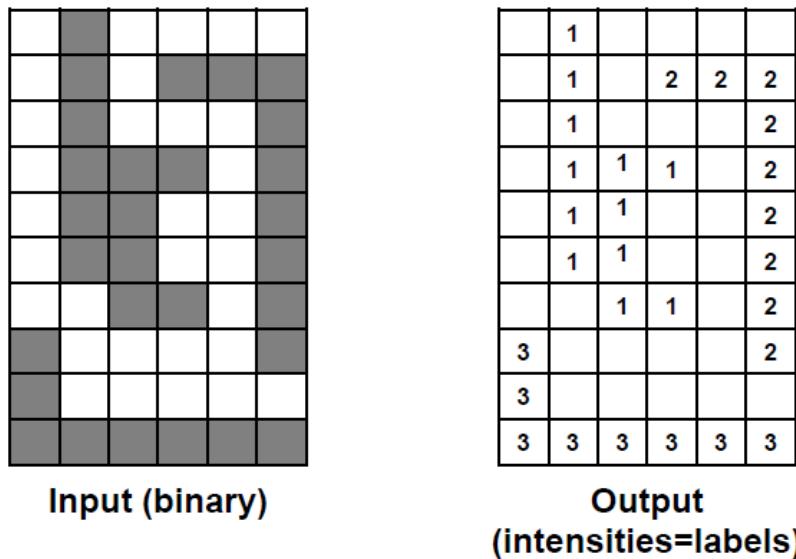
- Possible statements after segmentation:
  - There are three letters.
  - The letter with label 1 (red) is H.
  - The letter with label 2 (green) is A.
  - The letter with label 3 (blue) is W.
  - ...



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Region Labelling

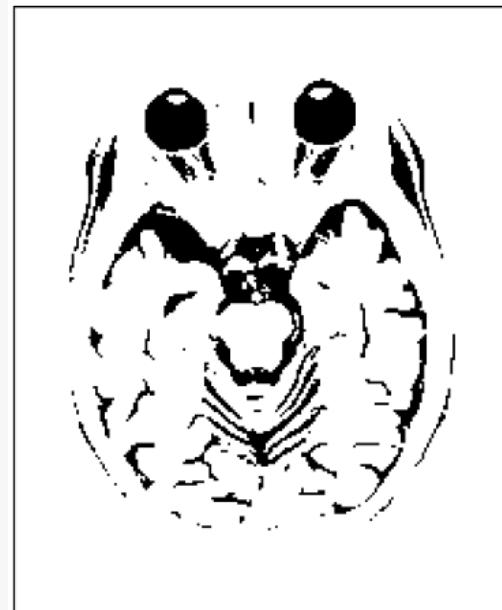
- Methods for region labeling use a binary input-image and create a grey-value output-image, where groups of connected pixels (=regions) have the same intensity (=label).
- Example:



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Example

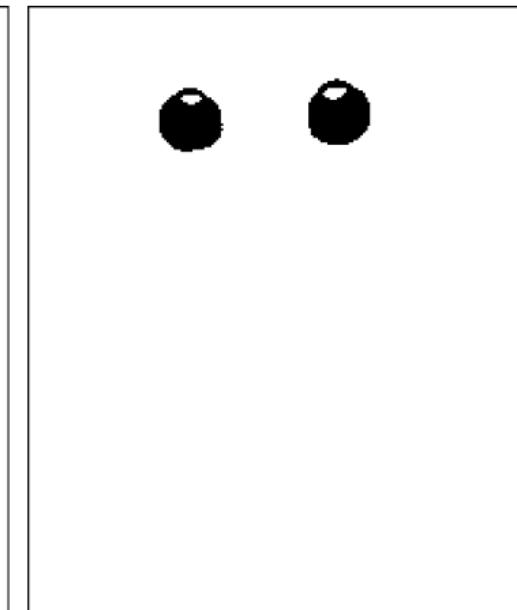
## ➤ Example: Segmentation of the eyes



1. binary image



2. labeled regions

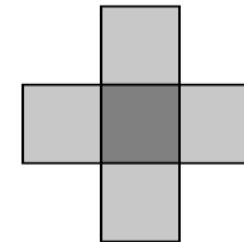


3. selected regions

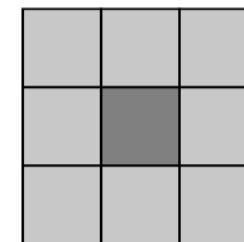
Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Connectivity

- There are 2 important types of connectivity:
  1. Only connectivity via an edge is considered.  
Then an inner pixel has 4 neighbors.
  2. Also connectivity via a corner is considered.  
Then an inner pixel has 8 neighbors.
    - Boundary pixels have less neighbors.
    - Other types of connectivity may also consider neighbors with a distance larger than 1.
- Using the 4-neighborhood is quicker and results in more separated regions, which is usually preferable in segmentation problems.



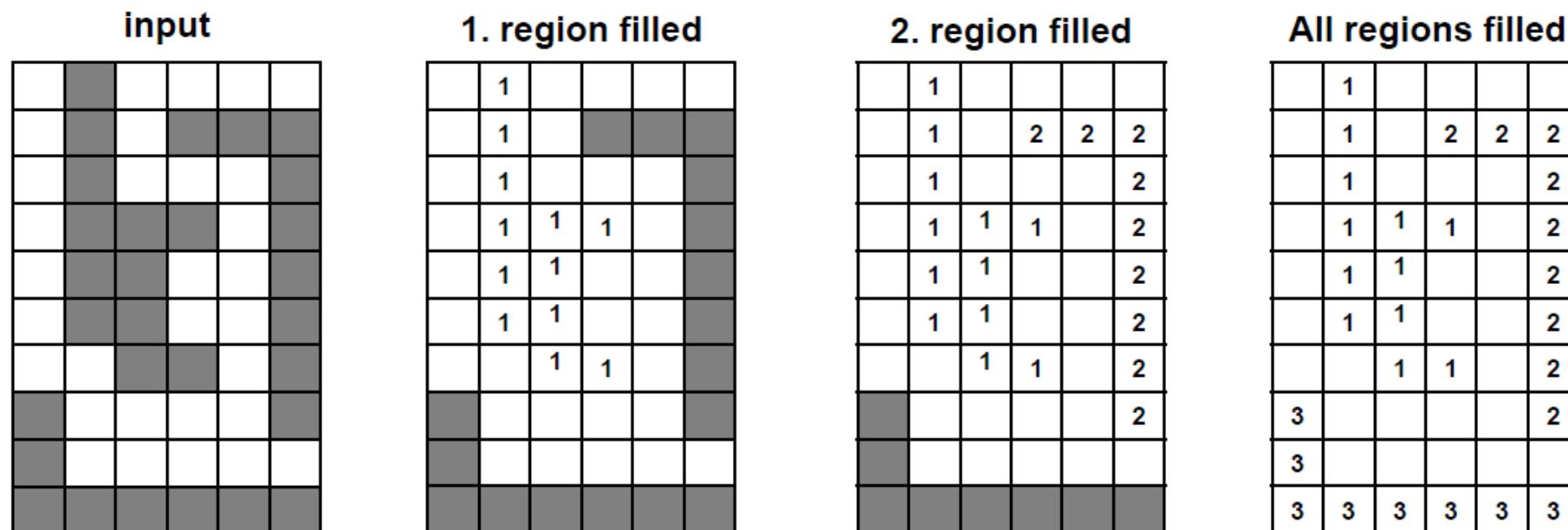
4 neighbors



8 neighbors

# Method 1: Flood Filling

- Find the next set pixel without a label
- Label this pixel and all connected pixels with the same label
- Recursive programming or administration of a stack



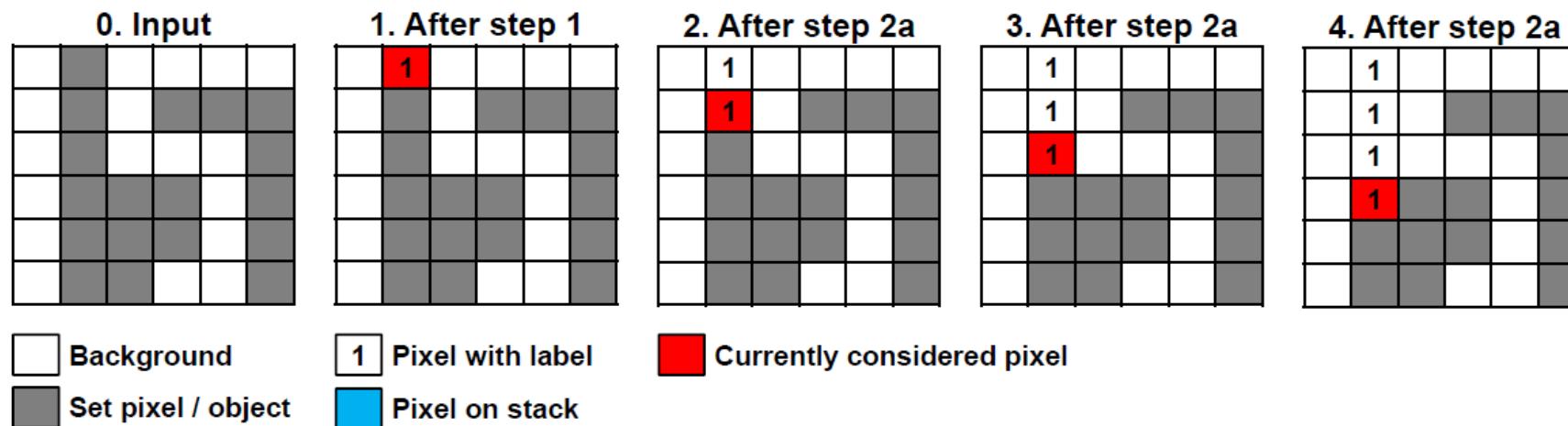
Courtesy: Prof. Dr.-Ing. Thomas Schiemann

September 26, 2023

Confidential. Not to be copied / distributed

# Steps

0. Set the current label-value to 1 and start at pixel (0,0)
1. Find the first (or next) set and unlabeled pixel and assign the current label-value to this pixel
2. Check all neighbors:
  - a: **Just one neighbor** is set, unlabeled and not stored on stack  
Move to this neighbor, assign the current label and continue at 2.



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Steps

b:  $N \geq 2$  **neighbors** are set, unlabeled and not stored on stack

Put  $N-2$  neighbors on stack

Move to 1 neighbor, assign the current label and continue at 2.

c: No set neighbors and unlabeled neighbors, but **stack filled**

Take pixel from stack

Move to this pixel, assign the current label and continue at 2

d: No set and unlabeled neighbors, and **stack empty**

Increase the current label-value and continue at 1

5. After step 2b

1				
1				
1				
1				
1				

6. After step 2b

1				
1				
1				
1				
1				

7. After step 2a

1				
1				
1				
1				
1				

8. After step 2c

1				
1				
1				
1				
1				

9. After step 2a

1				
1				
1				
1				
1				

9. After step 2a

1				
1				
1				
1				
1				

10. After step 2c

1				
1				
1				
1				
1				

11. After steps 2d and 1

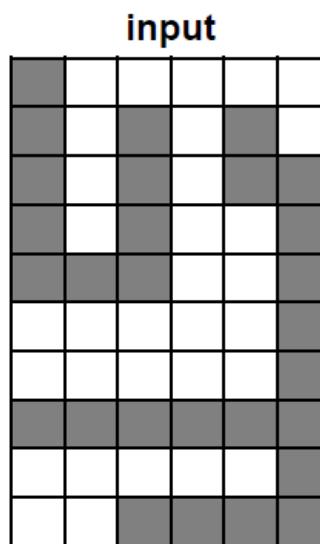
1				
1				
1				
1				
1				

... 17. Final result

1				
1				
1				
1				
1				

## Method 2: Sequential Filling

- Move through the whole image in a sequential order
- Assign the lowest label of the neighborhood to each pixel or assign a new label, if all neighbors are unlabeled.
- administration of a connection table
- 2 passes through the image



after 1. pass

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

connection table

region	connected to
2	1
4	3
5	3
...	...

after 2. pass

1					
1	1		3		
1	1		3	3	
1	1			3	
1	1	1		3	
				3	
3	3	3	3	3	3
					3
			3	3	3
				3	

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Binary Morphological Operations

- Basic operations: dilation and erosion
- Composed operations: opening and closing
- The operations are controlled by a structuring element, which can be regarded as a (small) binary image
  - typical structuring elements:

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

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

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

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Dilation

- The structuring element is positioned at every 1-pixel of the (binary) image.
- Every 0-pixel of the image, which is hit by a 1 of the structuring element is set to 1.
- Dilation widens structures.

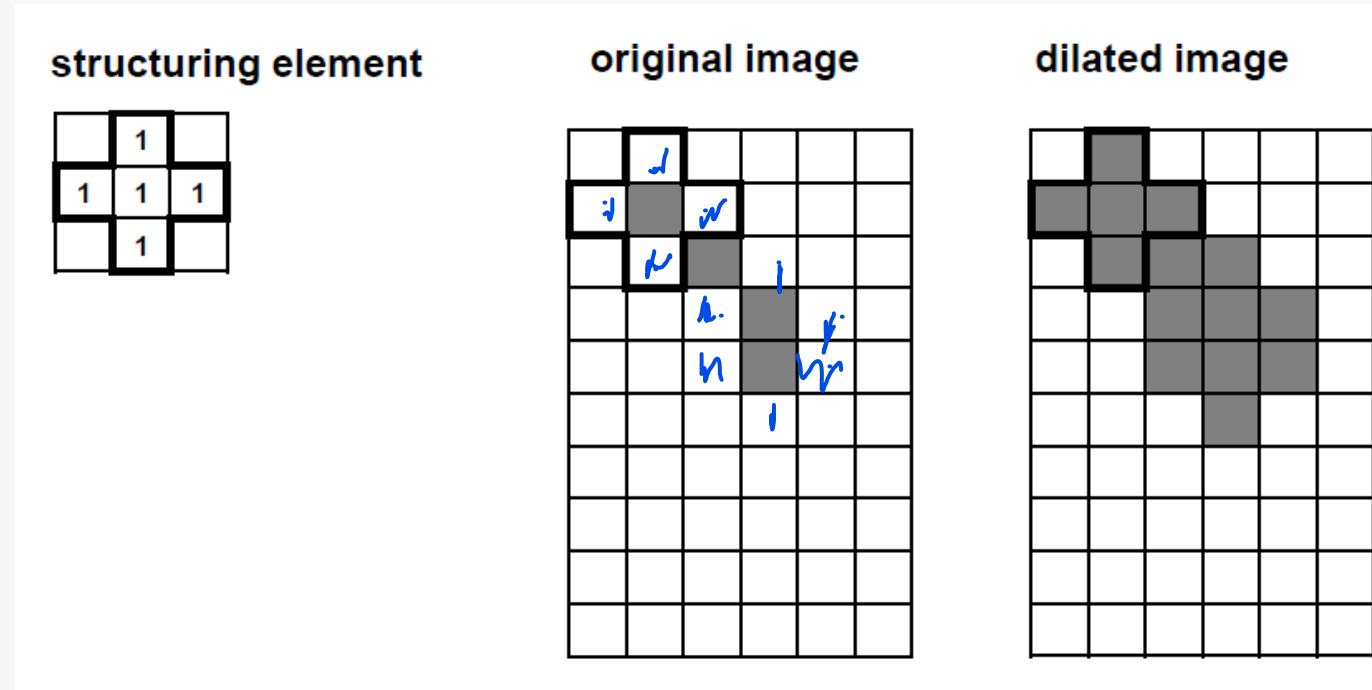
Example: Dilation by 3x3 box



**letters    letters**

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Dilation Example



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Erosion

- The structuring element is positioned at every 1-pixel of the (binary) image.
- If a 1 of the structuring element hits a 0 of the image, the image is set to 0 at the center of the structuring element.
- “Dilation of the background”
- Erosion narrows structures.

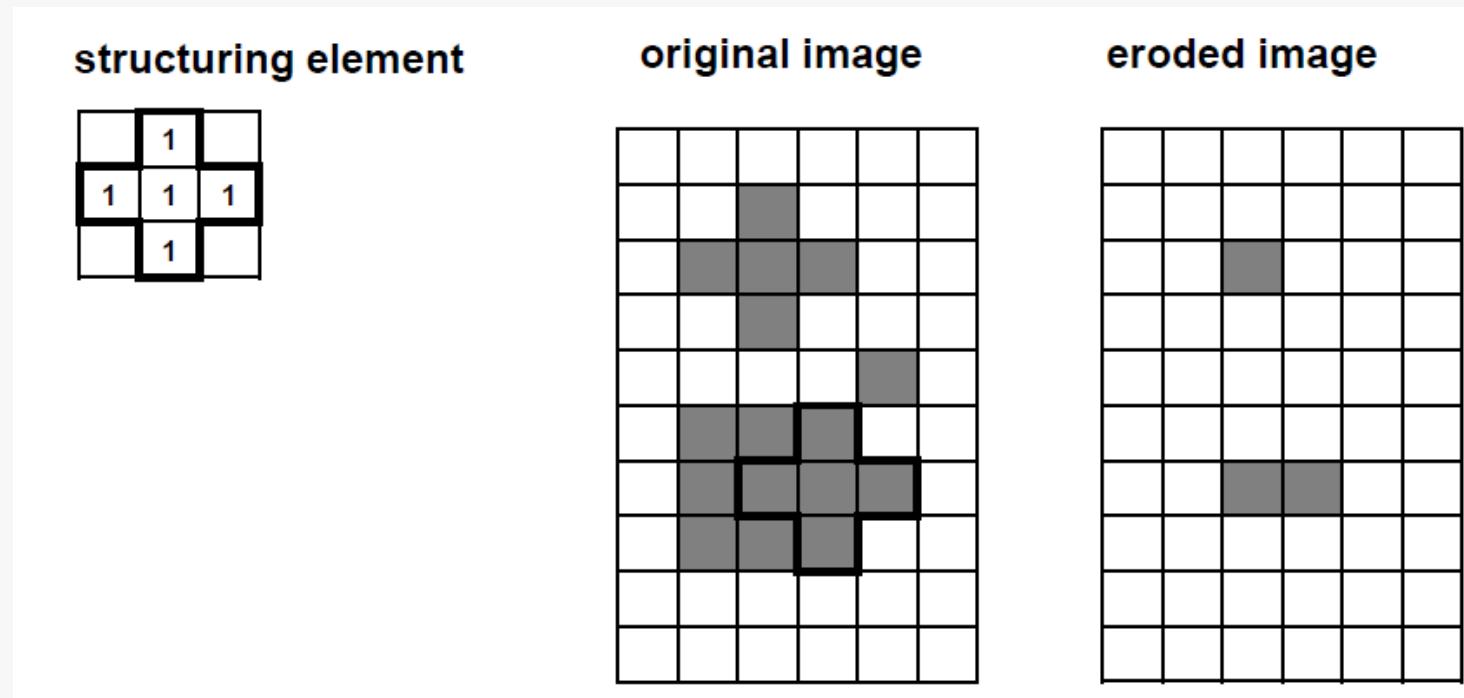
Example: Erosion by 3x3 box



letters    letters

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Erosion Example



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Opening

- Opening = 1. Erosion, 2. Dilation
- Erosion and Dilation are “opposite” operations, but they are not inverse.
- Only those pixels are left in the image, which are covered by a 1 of the structuring element, while it is completely located on 1-pixels of the image.
- The structuring element stamps its shape onto the original structures.

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Opening

Example: Opening by 3x3 box

**letters letters**

Example: Opening of a square object by a circular structuring element



Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Closing

- Closing = 1. Dilation, 2. Erosion
- Opening of the background
- Filling of holes, which are smaller or equal to the structuring element.

Example: Closing by 3x3 box



letters    letters

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Applications

- Finding boundaries
  - 1. Erosion
  - 2. Subtract erosion-output from input
- Separating objects from each other (Segmentation)
- Removal of small objects (opening)
- Filling of small holes (closing)
- (Finding objects of a certain shape)

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Morphological Operations in Formulas

Let  $I$  be the image,  $S$  the structuring element, and  $I_{ij}$  the image shifted  $i$  pixels in x-direction and  $j$  pixels in y-direction.

(The origin of the coordinate-system is usually located in the top left corner)

Then the dilation can be defined as

$$I \oplus S = \bigcup_{(i,j) \in S} I_{(-i,-j)} \quad (\text{Union of all shifted images})$$

and erosion as

$$I \ominus S = \bigcap_{(i,j) \in S} I_{(-i,-j)} \quad (\text{Intersection of all shifted images})$$

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Grey-Value Morphological Operations

The image  $I$  and the structuring element  $S$  both are containing grey-values now.

Then the dilation is defined as

$$(I \oplus S)(x, y) = \max_{(i,j) \in S} \{I(x + i, y + j) + S(i, j)\}$$

and erosion as

$$(I \ominus S)(x, y) = \min_{(i,j) \in S} \{I(x + i, y + j) - S(i, j)\}$$

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Example

Image <u>(surrounded by zeros)</u>	Structuring Element	Structuring element positioned at (1,1) for <u>dilation</u>	Dilation	Erosion																																																																									
<table border="1"><tr><td>7</td><td>6</td><td>3</td><td>5</td></tr><tr><td>5</td><td>6</td><td>2</td><td>3</td></tr><tr><td>5</td><td>9</td><td>3</td><td>2</td></tr><tr><td>3</td><td>5</td><td>7</td><td>7</td></tr></table>	7	6	3	5	5	6	2	3	5	9	3	2	3	5	7	7	<table border="1"><tr><td>2</td><td>1</td><td>0</td></tr><tr><td>0</td><td>3</td><td>1</td></tr><tr><td>0</td><td>0</td><td>4</td></tr></table>	2	1	0	0	3	1	0	0	4	<table border="1"><tr><td>9</td><td>7</td><td>3</td><td>5</td></tr><tr><td>5</td><td>9</td><td>3</td><td>3</td></tr><tr><td>5</td><td>9</td><td>7</td><td>2</td></tr><tr><td>3</td><td>5</td><td>7</td><td>7</td></tr></table>	9	7	3	5	5	9	3	3	5	9	7	2	3	5	7	7	<table border="1"><tr><td>10</td><td>9</td><td>7</td><td>8</td></tr><tr><td>13</td><td>9</td><td>8</td><td>6</td></tr><tr><td>10</td><td>12</td><td>11</td><td>5</td></tr><tr><td>6</td><td>10</td><td>11</td><td>10</td></tr></table>	10	9	7	8	13	9	8	6	10	12	11	5	6	10	11	10	<table border="1"><tr><td>2</td><td>-2</td><td>-1</td><td>2</td></tr><tr><td>2</td><td>-1</td><td>-2</td><td>0</td></tr><tr><td>1</td><td>2</td><td>0</td><td>-1</td></tr><tr><td>0</td><td>2</td><td>2</td><td>1</td></tr></table>	2	-2	-1	2	2	-1	-2	0	1	2	0	-1	0	2	2	1
7	6	3	5																																																																										
5	6	2	3																																																																										
5	9	3	2																																																																										
3	5	7	7																																																																										
2	1	0																																																																											
0	3	1																																																																											
0	0	4																																																																											
9	7	3	5																																																																										
5	9	3	3																																																																										
5	9	7	2																																																																										
3	5	7	7																																																																										
10	9	7	8																																																																										
13	9	8	6																																																																										
10	12	11	5																																																																										
6	10	11	10																																																																										
2	-2	-1	2																																																																										
2	-1	-2	0																																																																										
1	2	0	-1																																																																										
0	2	2	1																																																																										

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Special Case: Maximum-Filter

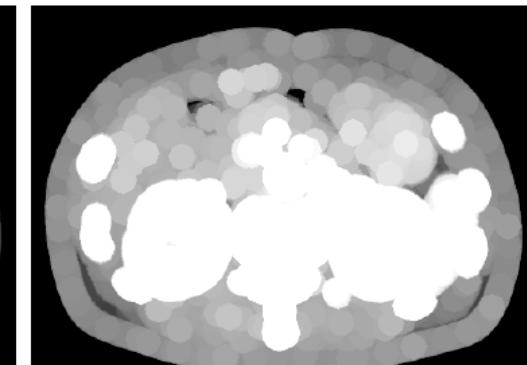
- The maximum filter is the grey-value *dilation* with a structuring element filled by zeros.
- Each pixel is replaced by the maximum intensity of its neighborhood, which is defined by the structuring element.



input



max-filter,  $r=2$



max-filter,  $r=12$

Courtesy: Prof. Dr.-Ing. Thomas Schiemann

# Special Case: Minimum-Filter

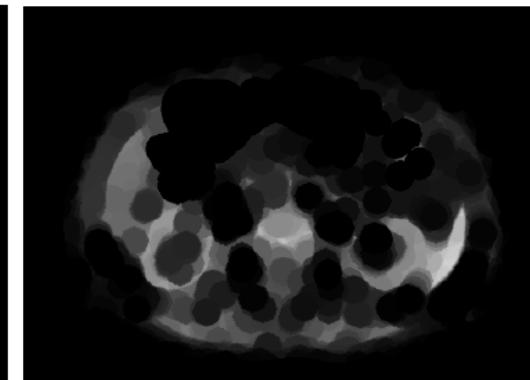
- The minimum filter is the grey-value *erosion* with a structuring element filled by zeros.
- Each pixel is replaced by the minimum intensity of its neighborhood, which is defined by the structuring element.



input



min-filter,  $r=2$



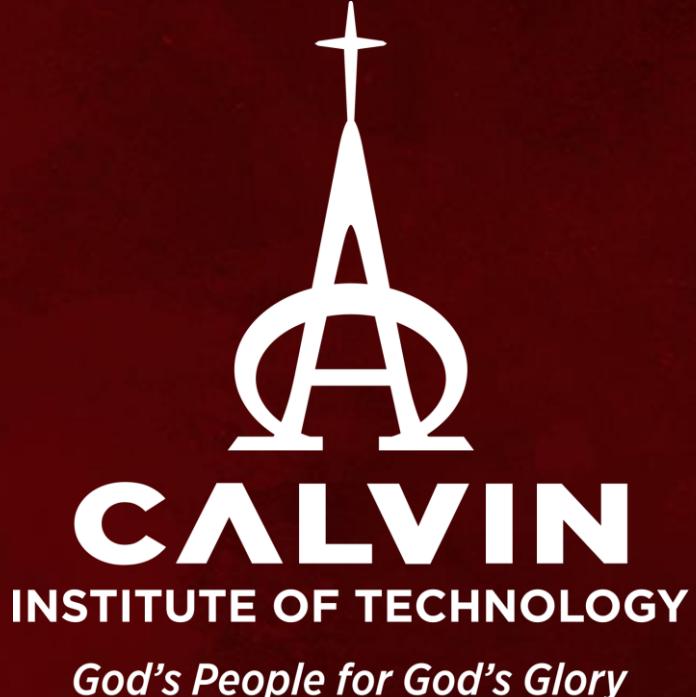
min-filter,  $r=12$

Courtesy: Prof. Dr.-Ing. Thomas Schiemann



# Latihan

Kerjakanlah Latihan Minggu 5\_2 dan kumpulkanlah jawaban anda dalam Canvas



Terima Kasih  
Tuhan Memberkati