Numerical Methods and Machine Learning for Image Processing

Week 1, Class 1: Introduction

September 13, 2021

Damon M. Chandler and Yi Zhang

Introduction, Part 1

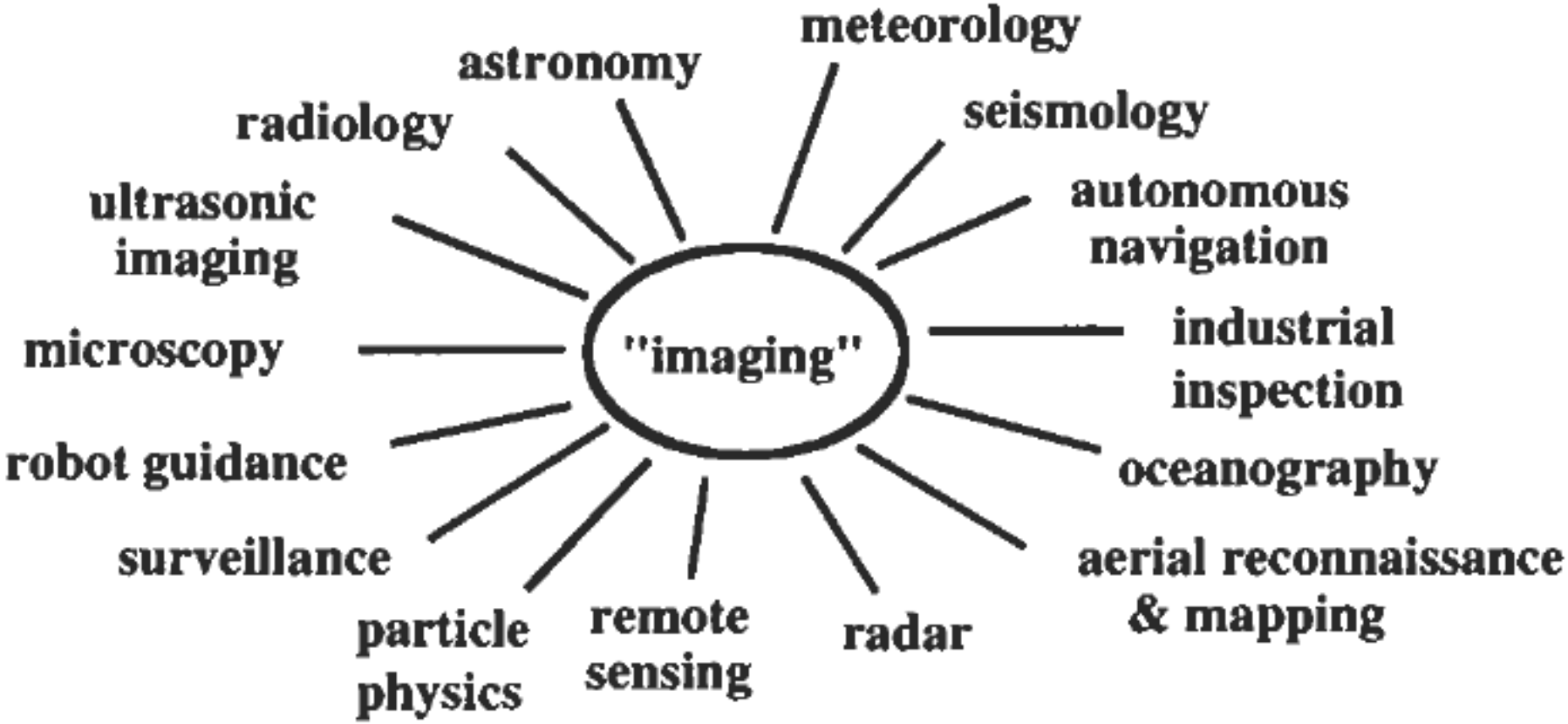
1. Overview of IP and CV
   - What is an image?
   - How are images formed?
   - What is IP and CV?
2. Python primer
   - Recommended software
   - Basic language primer

Today's goal: *Be able to write simple 1D numerical programs in Python*

Introduction, Part 1

1. Overview of IP and CV
   - What is an image?
   - How are images formed?
   - What is IP and CV?
2. Python primer
   - Recommended software
   - Basic language primer

What is IP and CV?

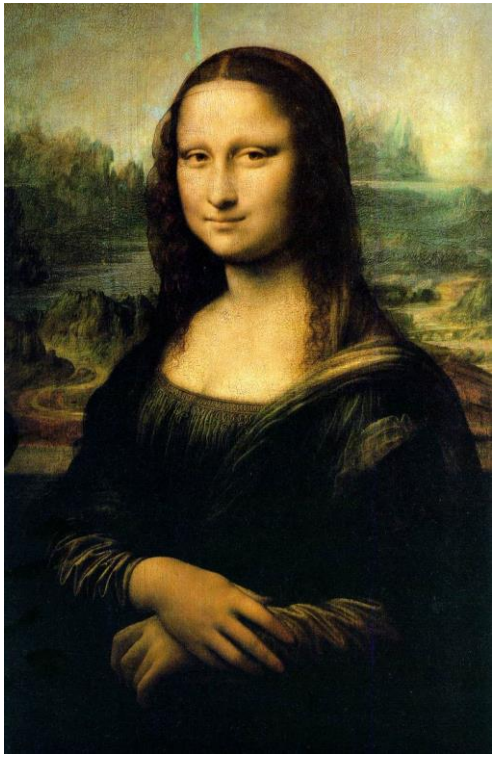


What *is* an image?

- An image as a **noun**
  - A **visual representation** of some measurable property of a person, object, or phenomenon

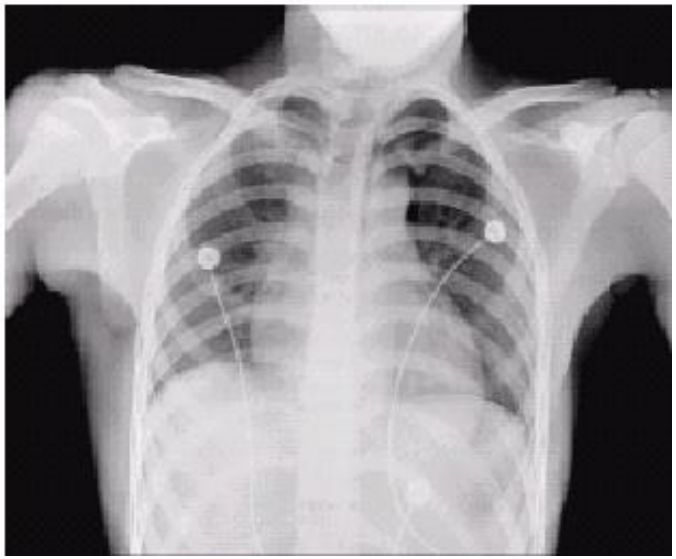
What *is* an image?

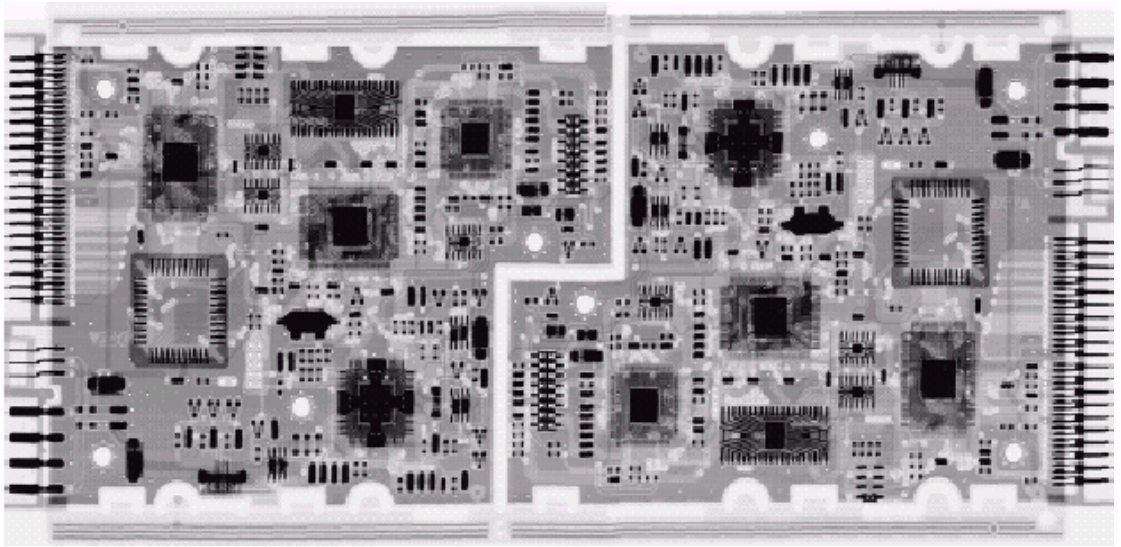
- An image as a **noun**
  - a **visual representation** of some measurable property of a person, object, or phenomenon



What *is* an image?

- An image as a **noun**
  - A **visual representation** of some measurable property of a person, object, or phenomenon





What *is* an image?

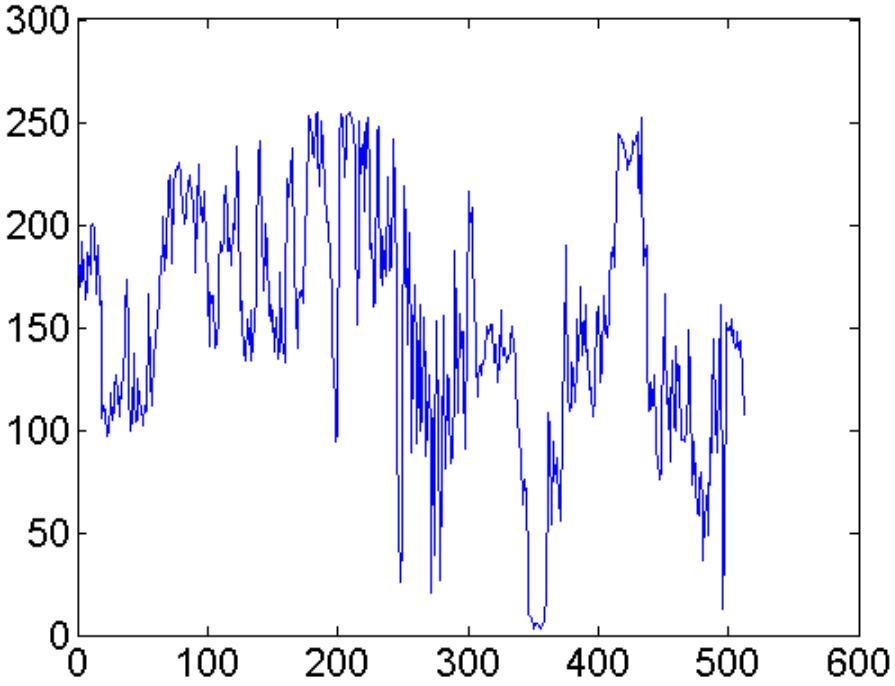
- An image as a **noun**
  - A **visual representation** of some measurable property of a person, object, or phenomenon



What *is* an image?

- An image as a **noun**
  - A **visual representation** of some measurable property of a person, object, or phenomenon
- An image as a mathematical **function** or **signal**

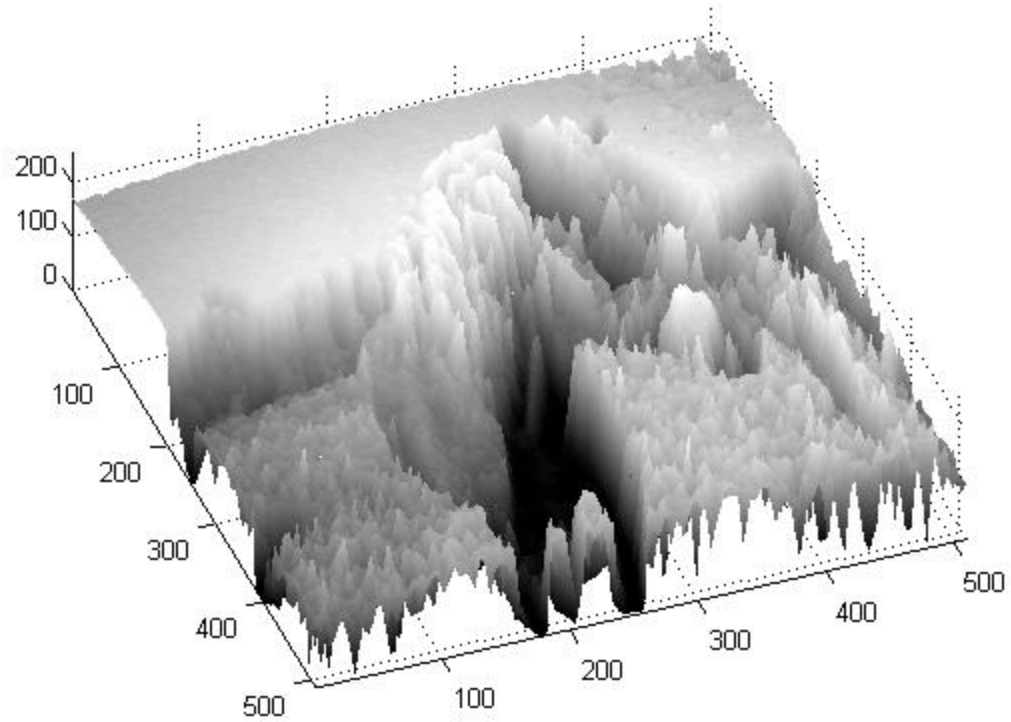
Example of a 1D signal



Continuous-time 1D signal

$f(t)$

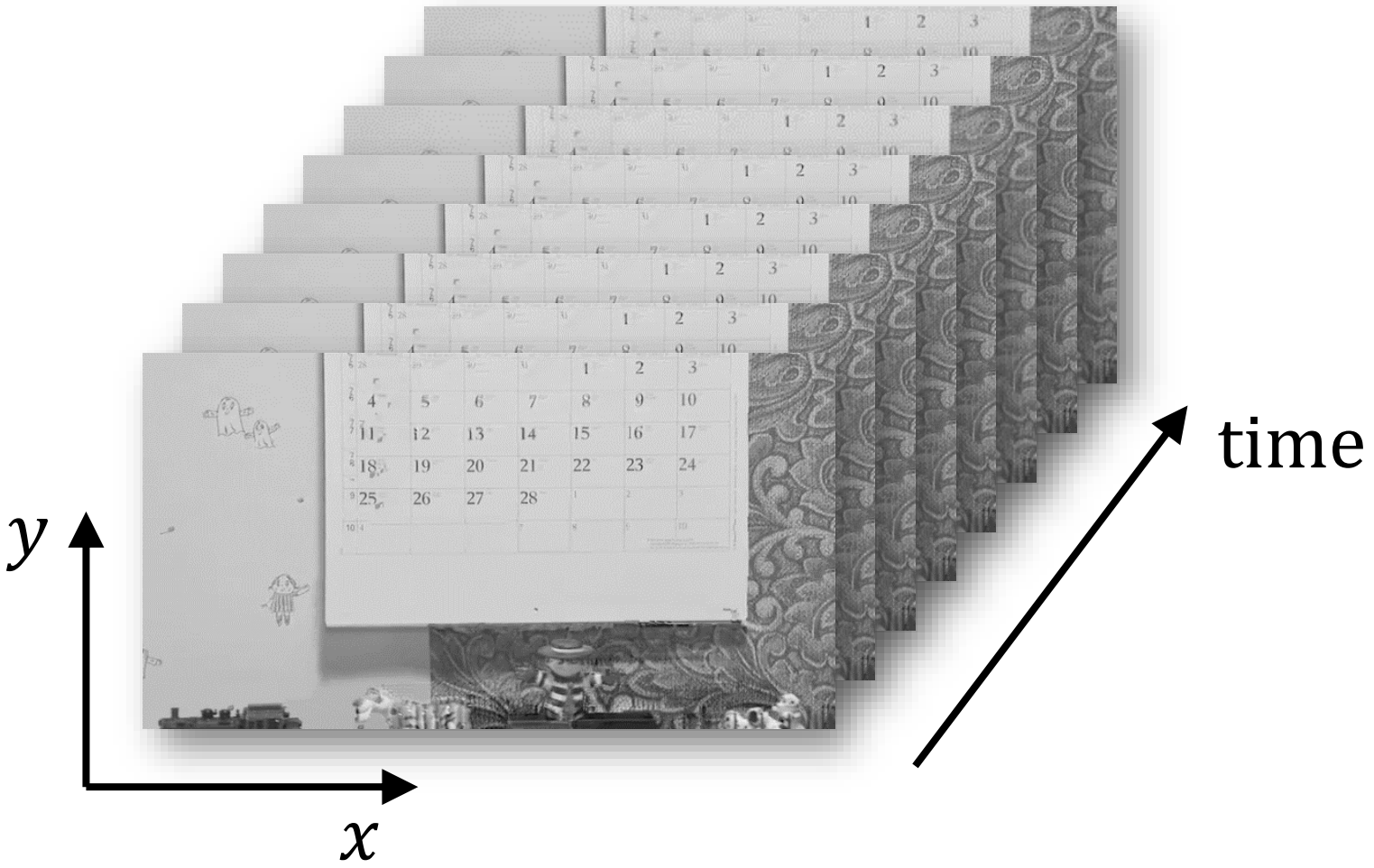
Example of a 2D signal



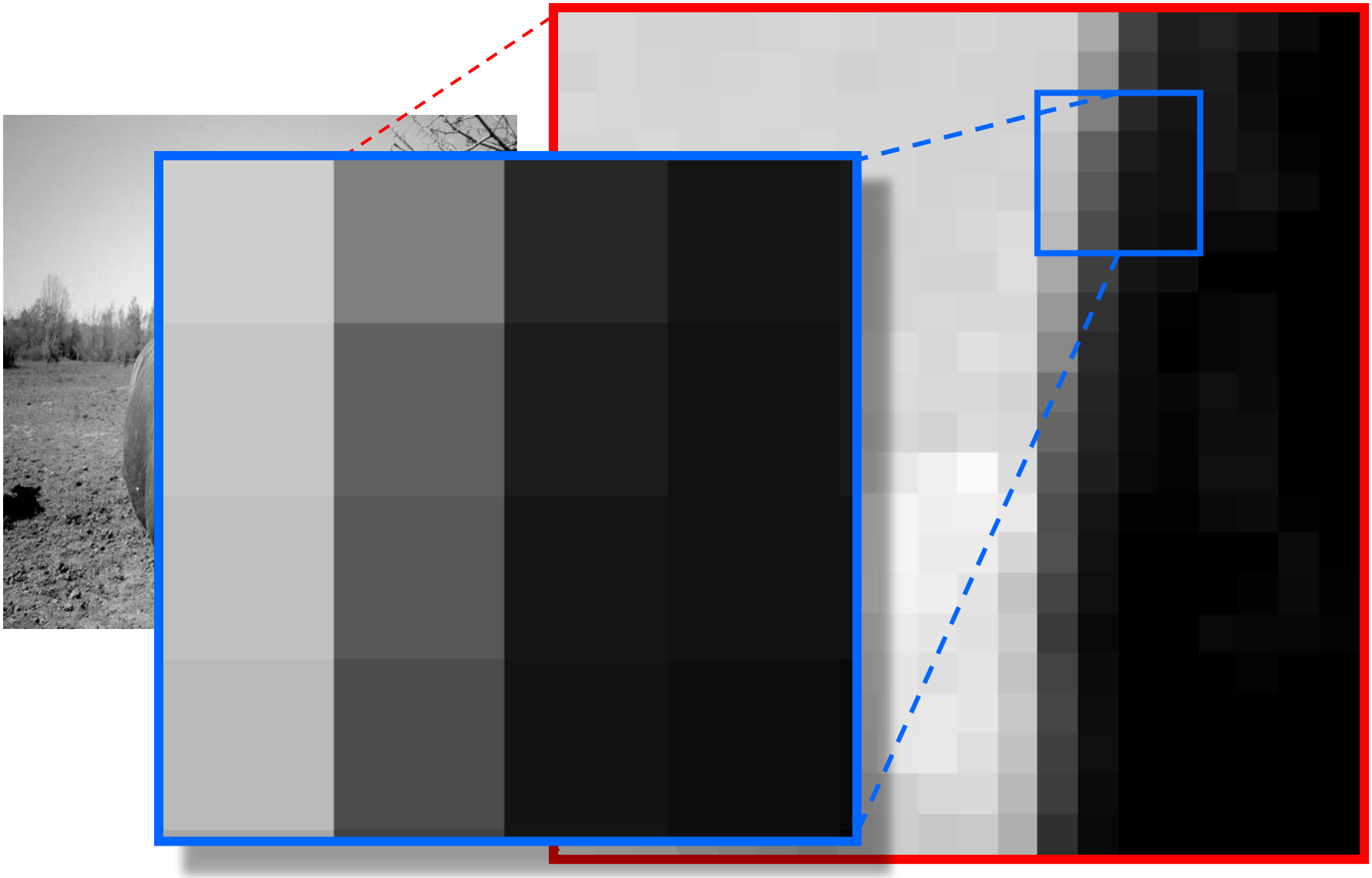
Continuous-space 2D signal

$f(x, y)$

Example of a 3D signal



What is an image?



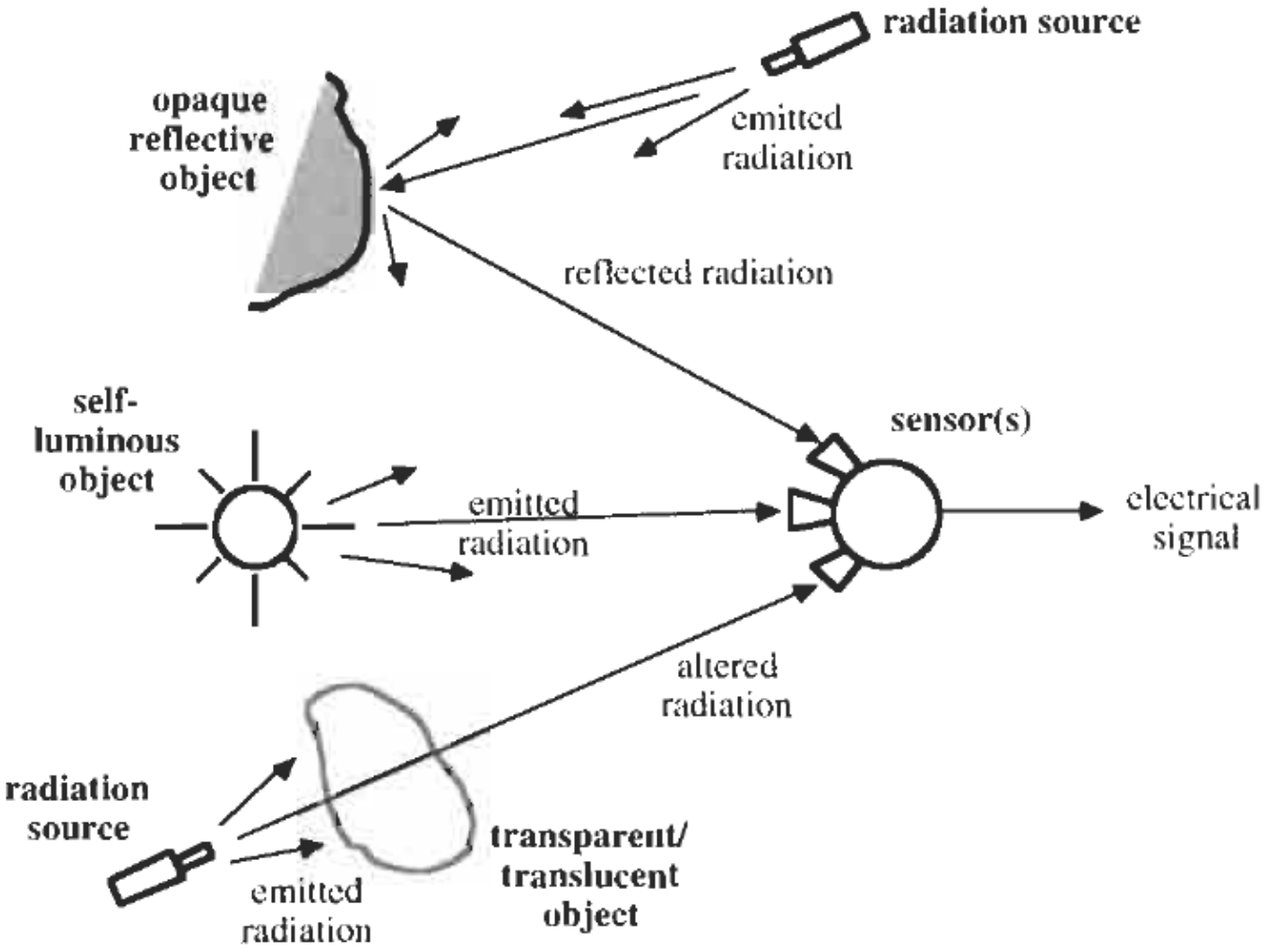
An image as a matrix

$$\begin{bmatrix} 170 & 160 & 157 & 158 & 160 & 162 & \\ 166 & 161 & 160 & 160 & 157 & 161 & \\ 168 & 158 & 160 & 163 & 158 & 159 & \\ 165 & 157 & 165 & 164 & 157 & 161 & \\ 165 & 158 & 161 & 159 & 163 & 166 & \cdots \\ 164 & 159 & 160 & 163 & 163 & 159 & \\ 167 & 160 & 160 & 161 & 162 & 165 & \\ 166 & 160 & 164 & 162 & 163 & 158 & \\ 165 & 163 & 160 & 159 & 158 & 161 & \\ 168 & 162 & 162 & 164 & 164 & 166 & \ddots \\ & & \vdots & & & & \end{bmatrix}$$

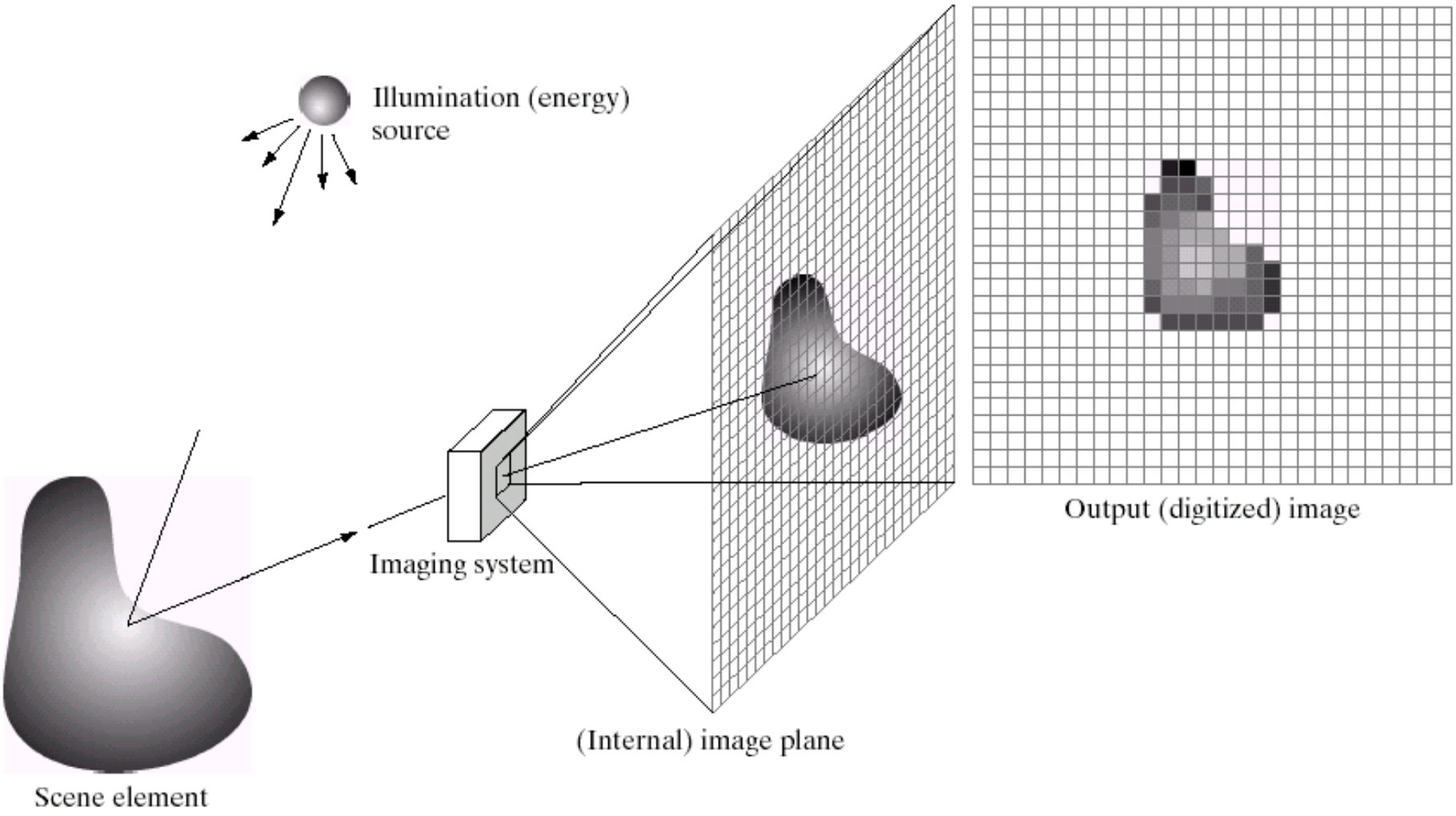
Matrix $A(r, c)$

How are images created?

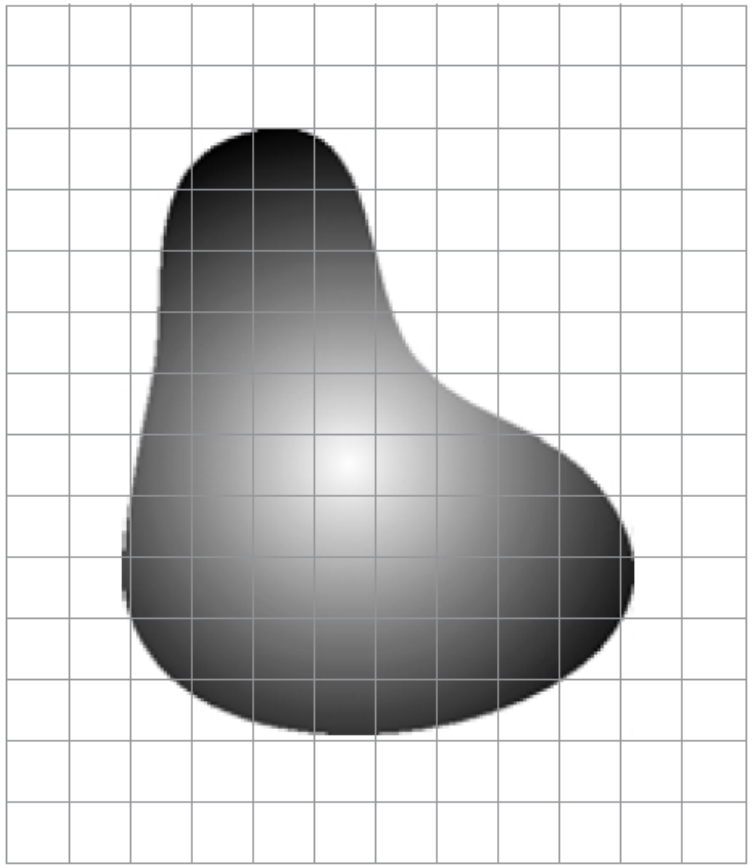
- **Reflection** images
- **Emission** images
- **Absorption** images



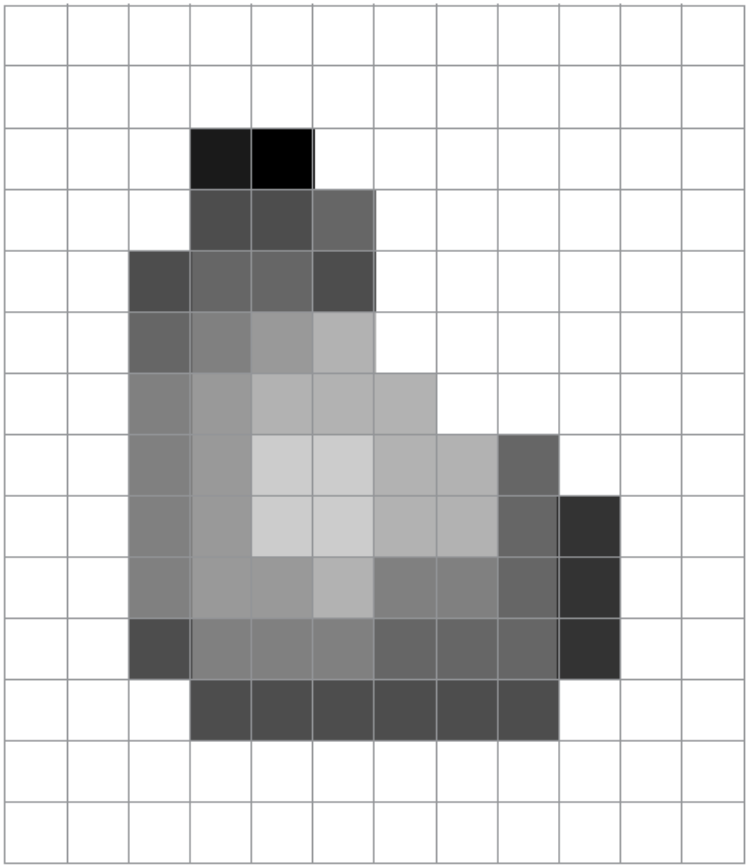
How are images created?



Two basic image properties



Continuous image



Digital image

Two basic image properties

1. **Number of pixels** in each direction
   - Determined by spatial sampling density and extent
   - Often called "**spatial resolution**"
2. **Number of bits per pixel**
   - Determined by quantization of intensity values
   - Number of shades of gray: 8-bpp → 256 shades
   - Often called "**intensity resolution**" or "gray/color resolution"

Continuous image

Digital image

Spatial resolution


1024


512

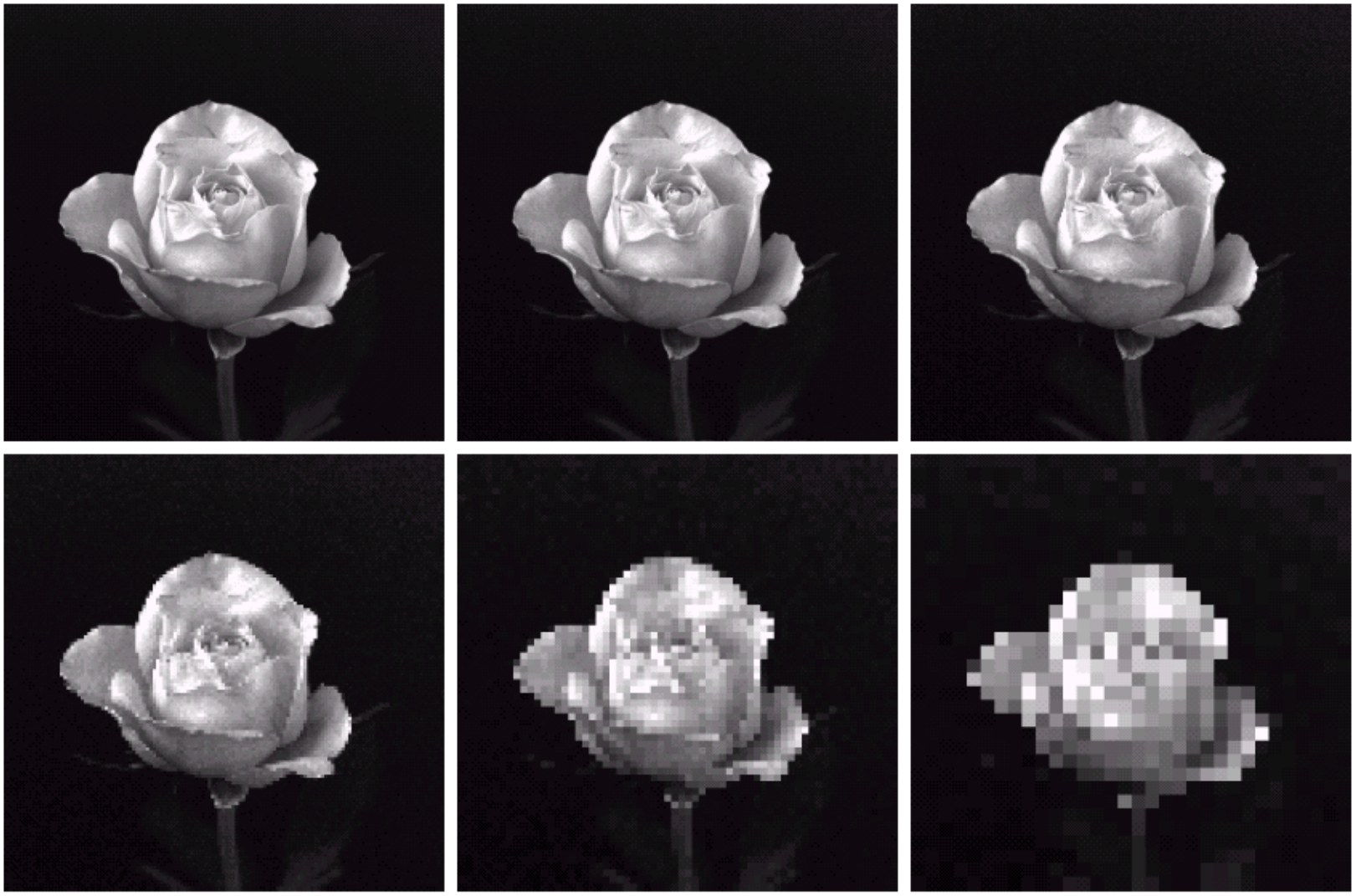

256


128


64

32

Spatial resolution



Intensity resolution

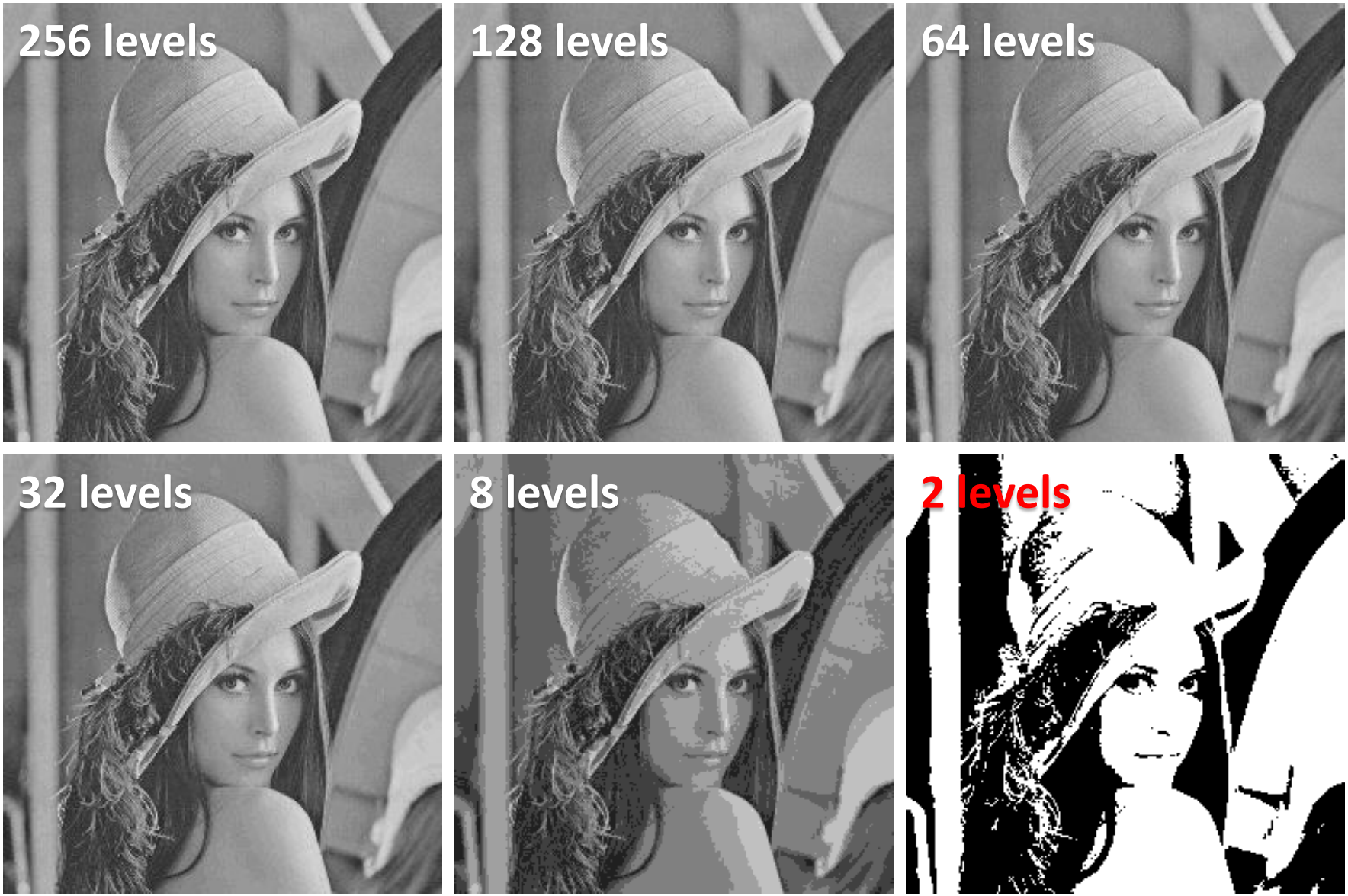
256 levels

128 levels

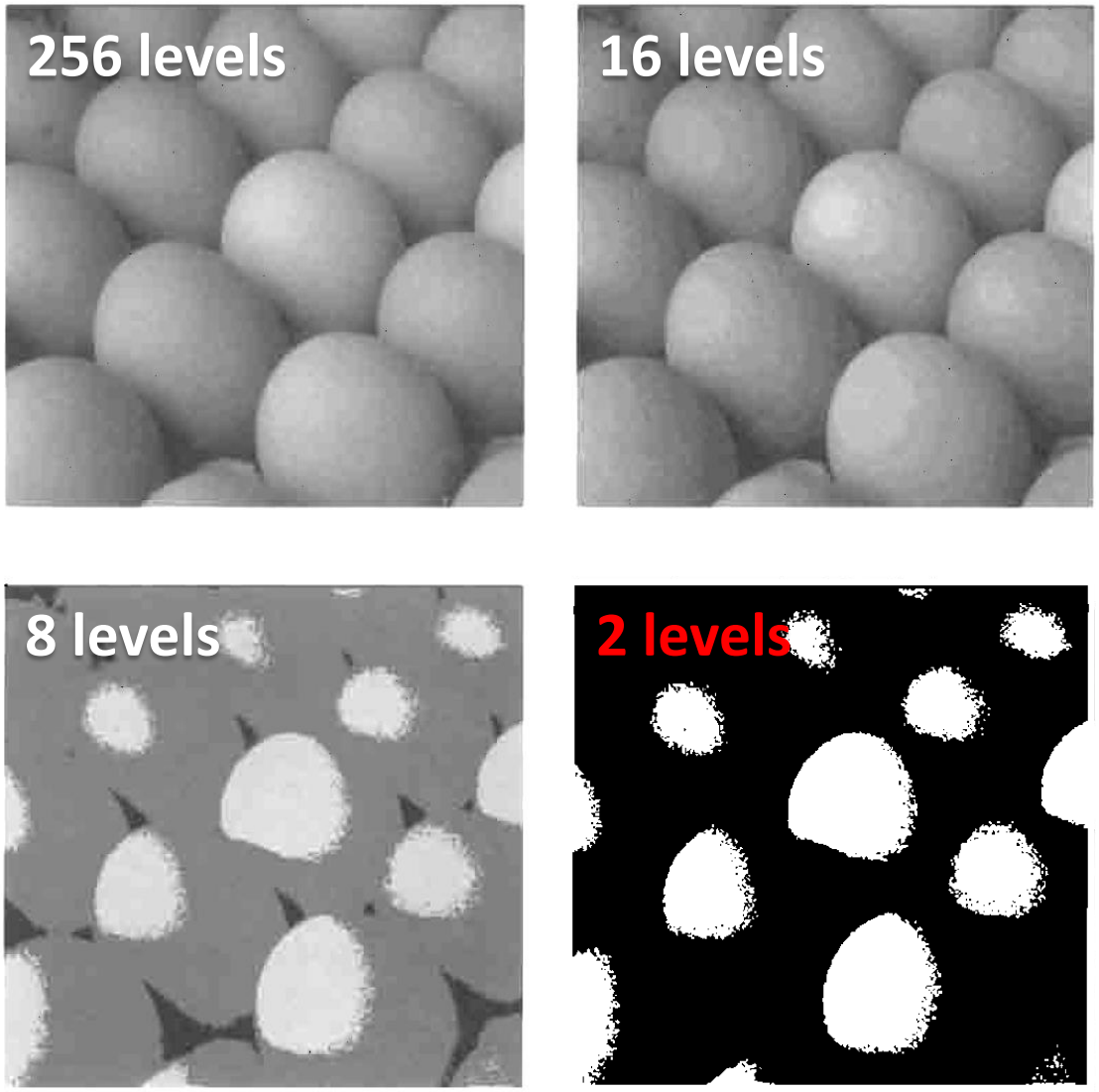
64 levels

32 levels

Intensity resolution

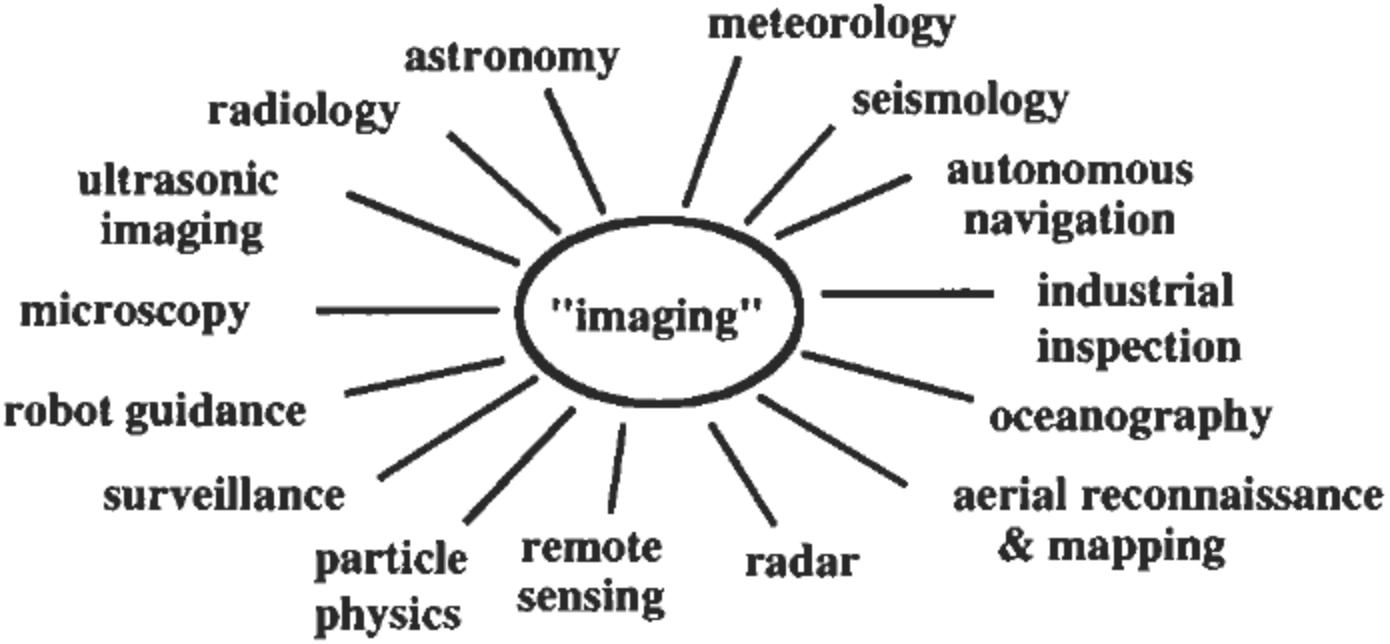


Intensity resolution



What is IP and CV?

- Image processing = processing images!
- Computer vision = vision of the computer!
- Virtually every branch of science has subdisciplines that use recording devices or sensors to collect image data



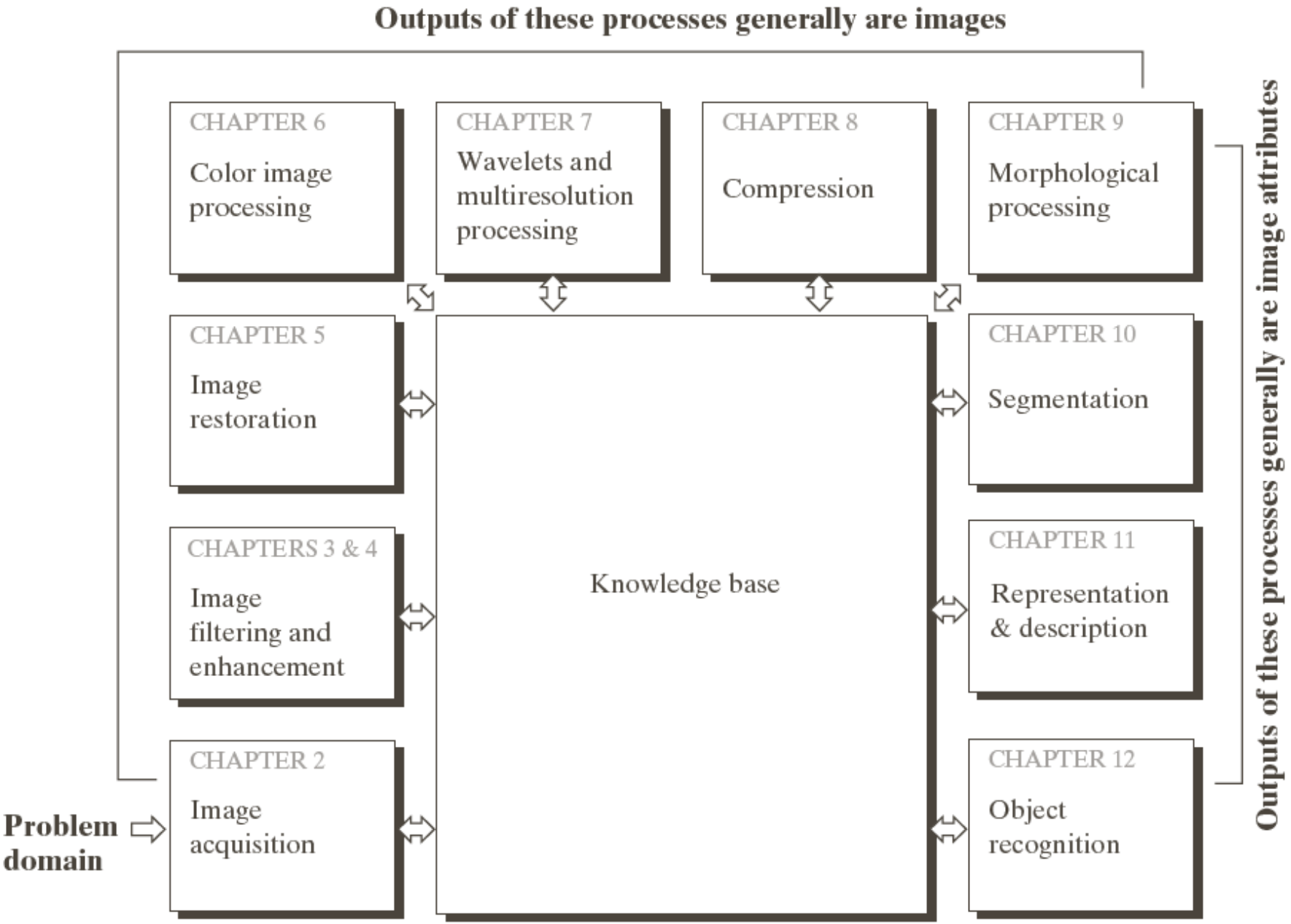

Outputs of these processes generally are images
CHAPTER 6
Color image processing
CHAPTER 7
Wavelets and multiresolution processing
CHAPTER 8
Compression
CHAPTER 9
Morphological processing
CHAPTER 5
Image restoration
Knowledge base
CHAPTER 10
Segmentation
CHAPTERS 3 & 4
Image filtering and enhancement
CHAPTER 11
Representation & description
CHAPTER 2
Image acquisition
CHAPTER 12
Object recognition
Problem domain
Outputs of these processes generally are image attributes

Image Enhancement

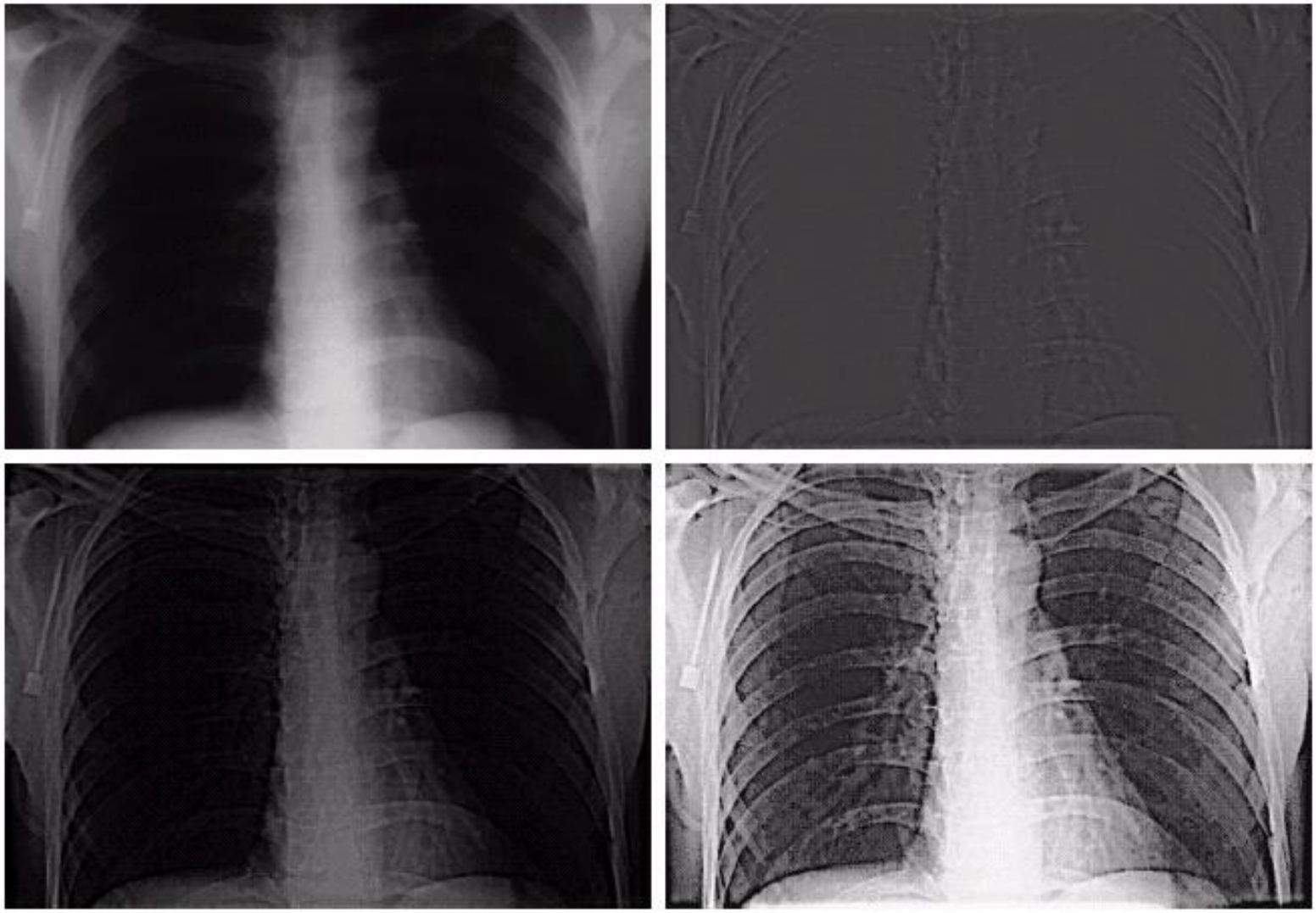
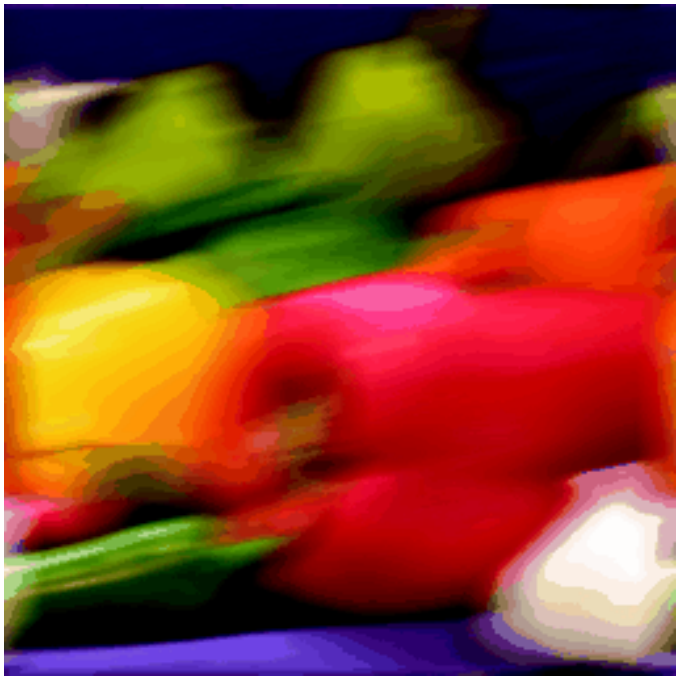


Image Restoration

Good shot



Blurred shot

Corrected (deblurred) image

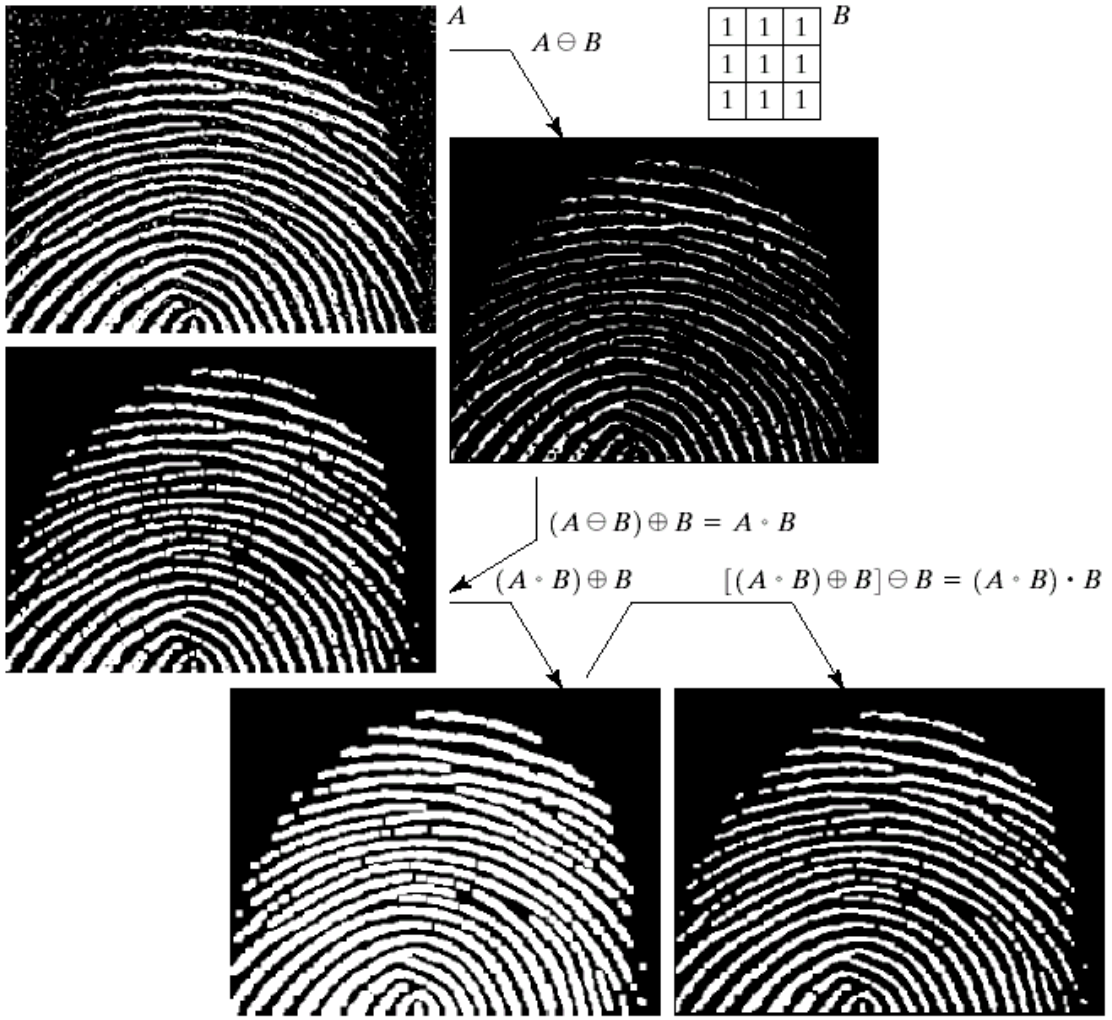
Image Compression

JPEG COMPRESSED
6.1 Kbytes

ORIGINAL IMAGE SIZE
700 Kbytes

JPEG 2000 COMPRESSED
6.1 Kbytes

Fingerprint Recognition



Introduction, Part 1

1. Overview of IP and CV
   - What is an image?
   - How are images formed?
   - What is IP and CV?
2. Python primer
   - Recommended software
   - Basic language primer

Recommended Software

- Python + package manager:
  - https://docs.anaconda.com/anaconda/install/
- Development Environment:
  - Visual Studio Code
    - https://code.visualstudio.com/docs/python/python-tutorial
      (follow download and installation instructions)
  - Spyder
    - Installed with Anaconda
  - PyCharm
    - https://www.jetbrains.com/pycharm/

```
a = 1
b = 2.5
c = "This is text"

print(a)
print(type(a), "\n")
print()
print(b)
print(type(b), "\n")
print()
print(c)
print(type(c), "\n")
```

```
1
<class 'int’>

2.5
<class 'float’>

This is text
<class 'str'>
```

```python
a = 1
b = 2.5
c = "This is text"

d = a + b
e = str(a) + c
f = a + int(b)

print(d)
print(type(d), "\n")
print()
print(e)
print(type(e), "\n")
print()
print(f)
print(type(f), "\n")
```

```
3.5
<class 'float'>

1This is text
<class 'str'>

3
<class 'int'>
```

```
a = 1
b = 2.5
c = "This is text"

d = a + b
e = str(a) + c
f = a + int(b)

%whos
```

```
Variable   Type       Data/Info
-----------------------------
a          int        1
b          float      2.5
c          str        This is text
d          float      3.5
e          str        1This is text
f          int        3
```

```
x = 10
y = x + 100
z = x / 100
w = 3*x - 1.5

print(x)
print(y)
print(z)
print(w)
```

```
10
110
0.1
28.5
```

```
x = 10
y = x*2
z = x**2

print(x)
print(y)
print(z)
```

```
10
20
100
```

```python
t = (10, 20, "apple")

print(t)
print(len(t))
print(type(t))
```

```
(10, 20, 'apple')
3
<class 'tuple'>
```

```python
t = (10, 20, "apple")

print(t[0])
print(t[1])
print(t[2])
```

```
10
20
apple
```

```
t = (10, 20, "apple")

t[1] = 5
print(t)
```

```
TypeError: 'tuple' object does not support item
assignment
```

```python
my_list = [10, 20, 30, 40, 50]

print(my_list)
print(len(my_list))
print(type(my_list))
print()
print(my_list[0])
print(my_list[1])
print(my_list[2])
print(my_list[3])
print(my_list[4])
print()
print(my_list[-1])
print(my_list[-2])
print(my_list[-3])
```

```
[10, 20, 30, 40, 50]
5
<class 'list’>

10
20
30
40
50

50
40
30
```

```python
my_list = [10, 20, 30, 40, 50]

for v in my_list:
  print(v)
```

```
10
20
30
40
50
```

```
my_list = [10, 20.5,
  "some text"]

print(my_list[0])
print(my_list[1])
print(my_list[2])
```

```
10
20.5
some text
```

```python
list1 = [10, 20, 30]
list2 = [40, 50, 60]
list3 = list1 + list2
list4 = [list1, list2]

print(list1)
print(list2)
print(list3)
print(list4)
print()
for y in list4:
  print(y)
  for x in y:
    print(x)
```

```
[10, 20, 30]
[40, 50, 60]
[10, 20, 30, 40, 50, 60]
[[10, 20, 30], [40, 50, 60]]

[10, 20, 30]
10
20
30
[40, 50, 60]
40
50
60
```

```
list1 = [10, 20, 30]
list2 = [40, 50, 60]

list3 = list1
list3[1] = 100

print(list1)
print(list2)
print(list3)
```

```
[10, 100, 30]
[40, 50, 60]
[10, 100, 30]
```

```
list1 = [10, 20, 30]
list2 = [40, 50, 60]

list3 = list1.copy()
list3[1] = 100

print(list1)
print(list2)
print(list3)
```

```
[10, 20, 30]
[40, 50, 60]
[10, 100, 30]
```

```python
list1 = [100, 200]
list2 = [400, 500, 600]

print(len(list1))
print(len(list2))
```

```
2
3
```

```python
nums = [10, 20, 30, 40,
  50, 60, 70, 80, 90]

print(nums)
print(nums[2:5])
print()
print(nums[0:9])
print(nums[:9])
print(nums[:])
print()
print(nums[0:-1])
print(nums[0:-2])
print()
print(nums[0:9:2])
print(nums[::2])
print()
print(nums[9::-1])
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90]
[30, 40, 50]

[10, 20, 30, 40, 50, 60, 70, 80, 90]
[10, 20, 30, 40, 50, 60, 70, 80, 90]
[10, 20, 30, 40, 50, 60, 70, 80, 90]

[10, 20, 30, 40, 50, 60, 70, 80]
[10, 20, 30, 40, 50, 60, 70]

[10, 30, 50, 70, 90]
[10, 30, 50, 70, 90]

[90, 80, 70, 60, 50, 40, 30, 20, 10]
```

```
x = 10
if x == 2:
  print("x is 2")
elif x == 1 or x == 1.5:
  print("x is either 1 or 1.5")
elif x > 30 and x < 100:
  print("x is greater than 30 and less than 100")
else:
  print("x is something else")
```

```
x is something else
```

```
x = 2
if x == 2:
  print("x is 2")
elif x == 1 or x == 1.5:
  print("x is either 1 or 1.5")
elif x > 30 and x < 100:
  print("x is greater than 30 and less than 100")
else:
  print("x is something else")
```

```
x is 2
```

```
x = 1.5
if x == 2:
  print("x is 2")
elif x == 1 or x == 1.5:
  print("x is either 1 or 1.5")
elif x > 30 and x < 100:
  print("x is greater than 30 and less than 100")
else:
  print("x is something else")
```

```
x is either 1 or 1.5
```

```
x = 99
if x == 2:
  print("x is 2")
elif x == 1 or x == 1.5:
  print("x is either 1 or 1.5")
elif x > 30 and x < 100:
  print("x is greater than 30 and less than 100")
else:
  print("x is something else")
```

```
x is greater than 30 and less than 100
```

```python
a = 1
b = 2.5
c = 100 if a+b < 4 else -100
print(c)

a = 2
b = 2.5
c = 100 if a+b < 4 else -100
print(c)
```

```
100
-100
```

```python
nums = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

for i in nums:
  print(i, end=", " if i<=8 else "\n")
```

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

```python
for i in range(0, 10):
  print(i, end=", " if i<=8 else "\n")
```

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

```
for i in range(0, 10):
  print(i, end=", " if i<=8 else "\n")

for i in range(10):
  print(i, end=", " if i<=8 else "\n")

for i in range(0, 10, 2):
  print(i, end=", " if i<=6 else "\n")
```

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
0, 2, 4, 6, 8
```

```python
my_list = [10, 20, 30, 40, 50]

for v in my_list:
  print(v)

print()

for idx in range(len(my_list)):
  print(my_list[idx])
```

```
10
20
30
40
50

10
20
30
40
50
```

```
i = 0
while i < 10:
  print(i, end=", " if i<=8 else "\n")
  i += 1
```

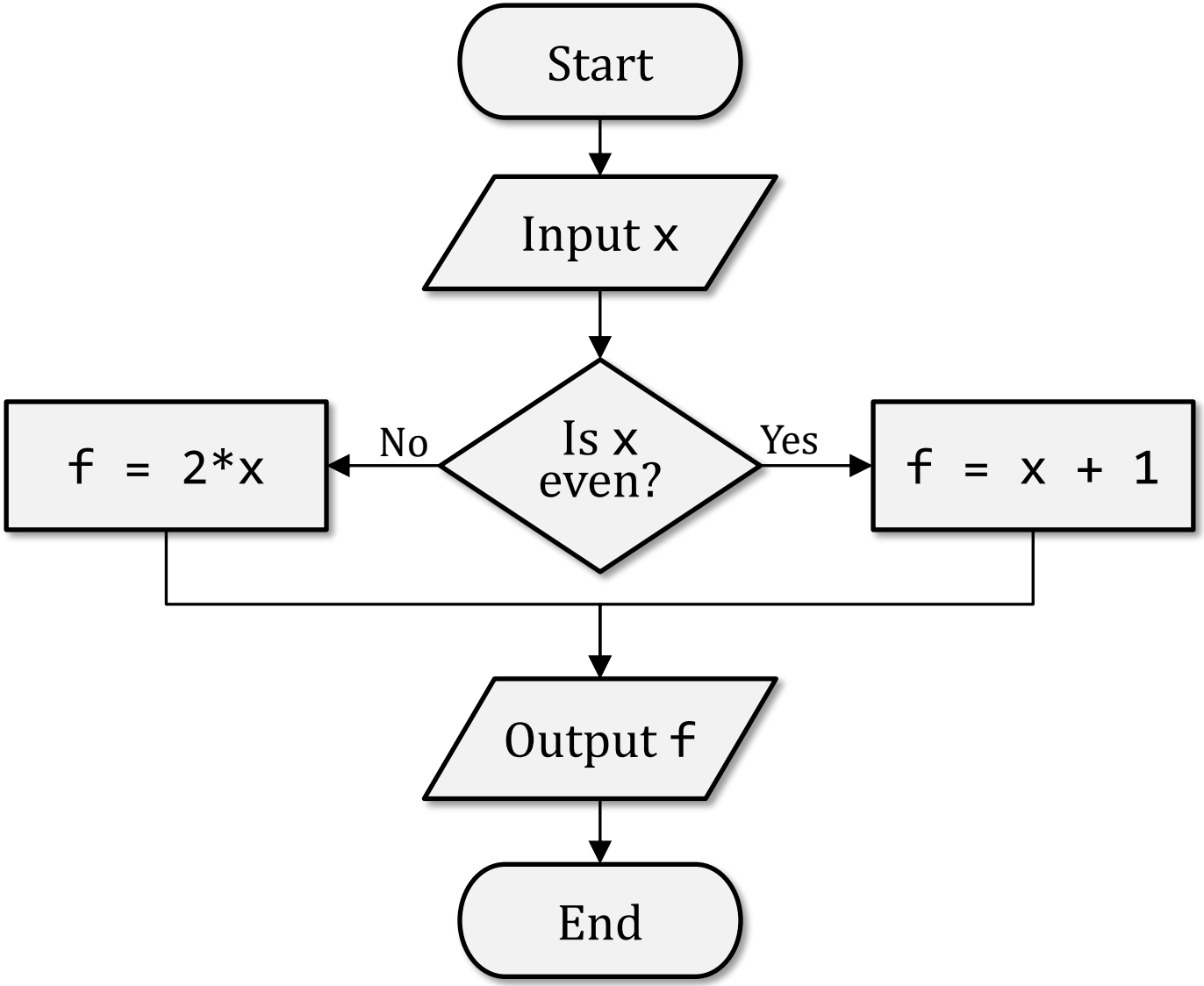
```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

```
i = 0
while True:
  print(i, end=", " if i<=8 else "\n")
  i += 1
  if i == 10:
    break
```

```
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
```

A simple example

$$f(x) = \begin{cases} x + 1, & x \text{ even} \\ 2x, & x \text{ odd} \end{cases}$$



```
print("Enter a number:")
x = input()

try:
  x = int(x)
  if x % 2 == 0:
    x += 1
  else:
    x *= 2
  print(x)

except ValueError:
  print("You entered a non-integer!")
```

```python
# Processing for even numbers -------------------------------
def do_if_even(x):
  x += 1
  return x
#-----------------------------------------------------------

# Processing for odd numbers --------------------------------
def do_if_odd(x):
  x *= 2
  return x
#-----------------------------------------------------------

print("Enter a number:")
x = input()

try:
  x = int(x)
  if x % 2 == 0:
    x = do_if_even(x)
  else:
    x = do_if_odd(x)
  print(x)

except ValueError:
  print("You entered a non-integer!")
```

```python
# Processing for even numbers --------------------------------
def do_if_even(x):
  return x + 1
#-------------------------------------------------------------

# Processing for odd numbers ---------------------------------
def do_if_odd(x):
  return  x * 2
#-------------------------------------------------------------

print("Enter a number:")
x = input()

try:
  x = int(x)
  if x % 2 == 0:
    x = do_if_even(x)
  else:
    x = do_if_odd(x)
  print(x)

except ValueError:
  print("You entered a non-integer!")
```

```python
# Conversion from str to int ------------------------------------
def str2int(str_val):
  try:
    return int(str_val)
  except ValueError:
    return None
#--------------------------------------------------------------


print("Enter a number:")
x = input()

x = str2int(x)
if x:
  if x % 2 == 0:
    x = do_if_even(x)
  else:
    x = do_if_odd(x)
  print(x)
else:
  print("You entered a non-integer!")
```

```python
# Processing function ------------------
def proc_numbers(x, y, op="add"):
  if op == "add":
    return x + y
  elif op == "subtract":
    return x - y
  elif op == "multiply":
    return x * y
  elif op == "divide":
    return x / y
# ------------------------------------

z = proc_numbers(2.5, 100)
print(z)
z = proc_numbers(2.5, 100, "add")
print(z)
z = proc_numbers(2.5, 100, "subtract")
print(z)
z = proc_numbers(2.5, 100, "multiply")
print(z)
z = proc_numbers(2.5, 100, "divide")
print(z)
z = proc_numbers(2.5, 100, "hello")
print(z)
```

```
102.5
102.5
-97.5
250.0
0.025
None
```

```python
# Processing function -------------------
def proc_numbers(x, y, op="add"):
  if op == "add":
    return x + y
  elif op == "subtract":
    return x - y
  elif op == "multiply":
    return x * y
  elif op == "divide":
    return x / y
# ---------------------------------------

z = proc_numbers(x=2.5, y=100, op="add")
print(z)
z = proc_numbers(y=100, op="add", x=2.5)
print(z)
```

```
102.5
102.5
```