Introduction to Robotics

Manipulation and Programming

Unit 3: Sensors and Vision

CAMERA AND COLOR – PART 2: COLOR SPACE

DR. ERIC CHOU

IEEE SENIOR MEMBER

Objectives

- Taking Still Image Picture
- Extracting R, G, B Images
- Calculate the Gray Image

Still Image Picture Versus Video

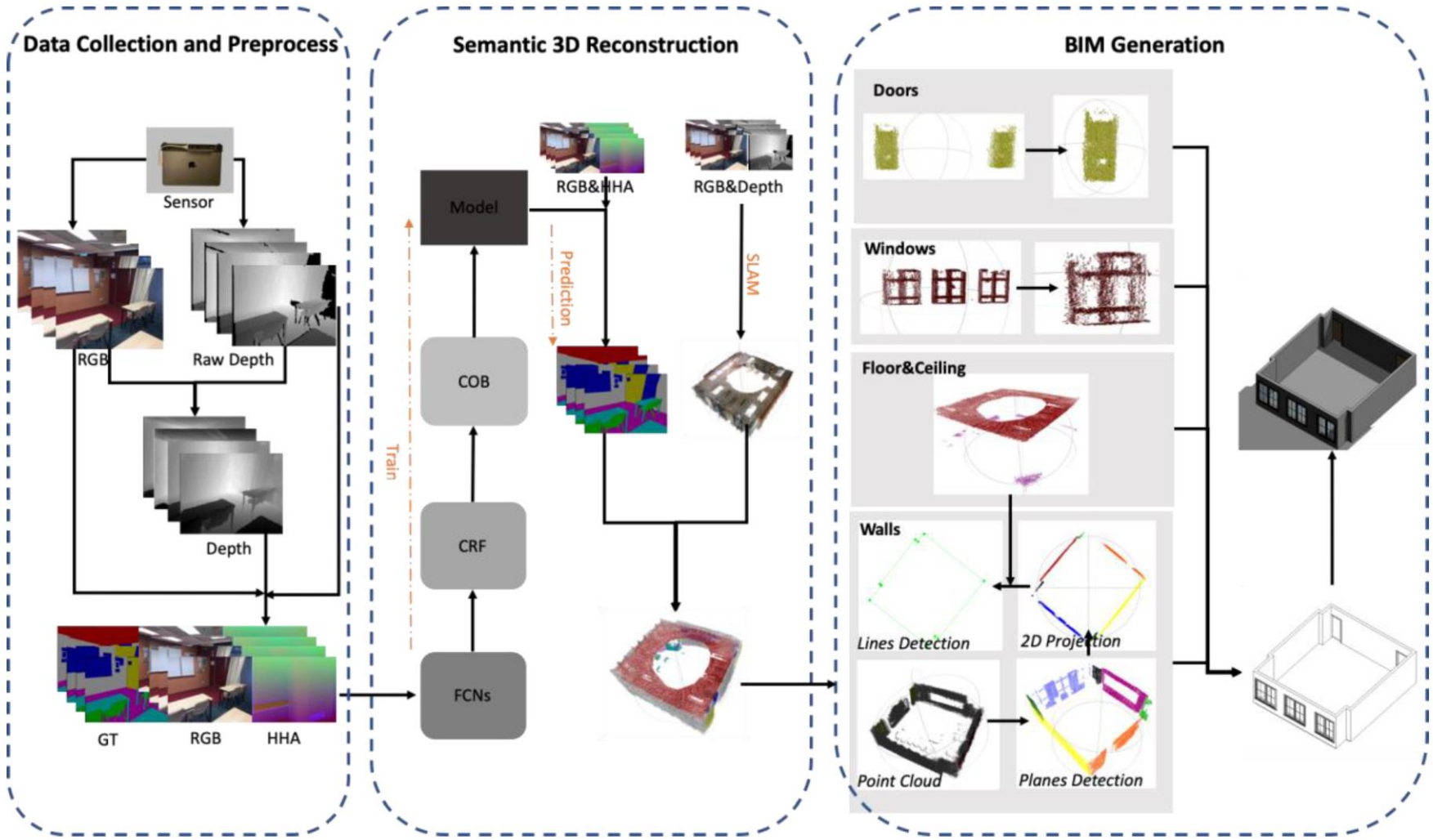
SECTION 1

Taking CMOS Images

- Electronics Shutter
- Exposure Time
- Gamma Correction
- Dynamic Range

Camera Models

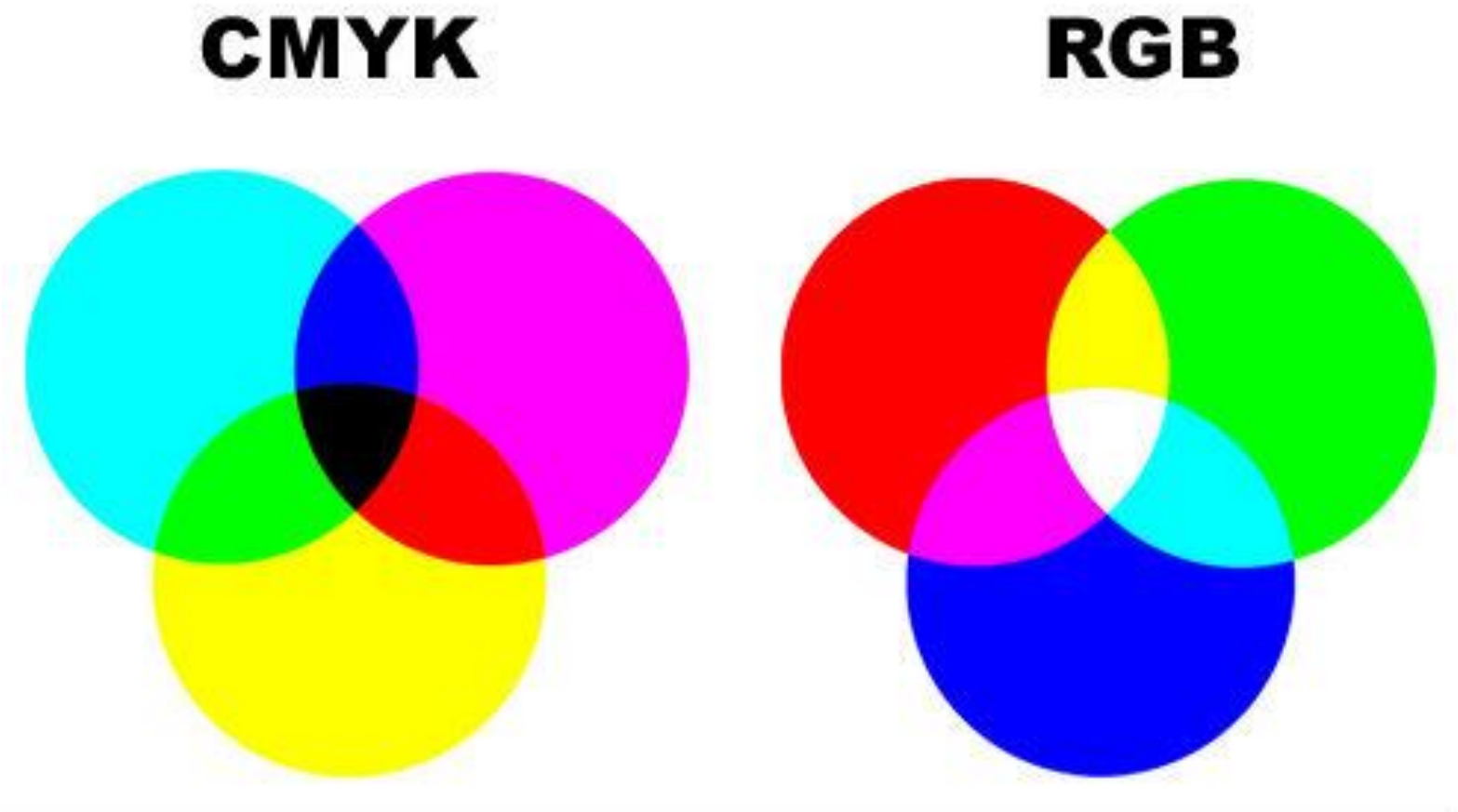
```
import numpy as np
import cv2
# start a camera
cam = cv2.VideoCapture(0)

# frame capturing loop
while True:
    _, frame = cam.read()
    cv2.imshow("Figure 1", frame)
    # new frame is here.

    key = cv2.waitKey(5)
    if key == 27:
        break
cv2.destroyAllWindows()

# print out the frame
print(frame)
print(frame.shape)
print(frame[0][0])
```

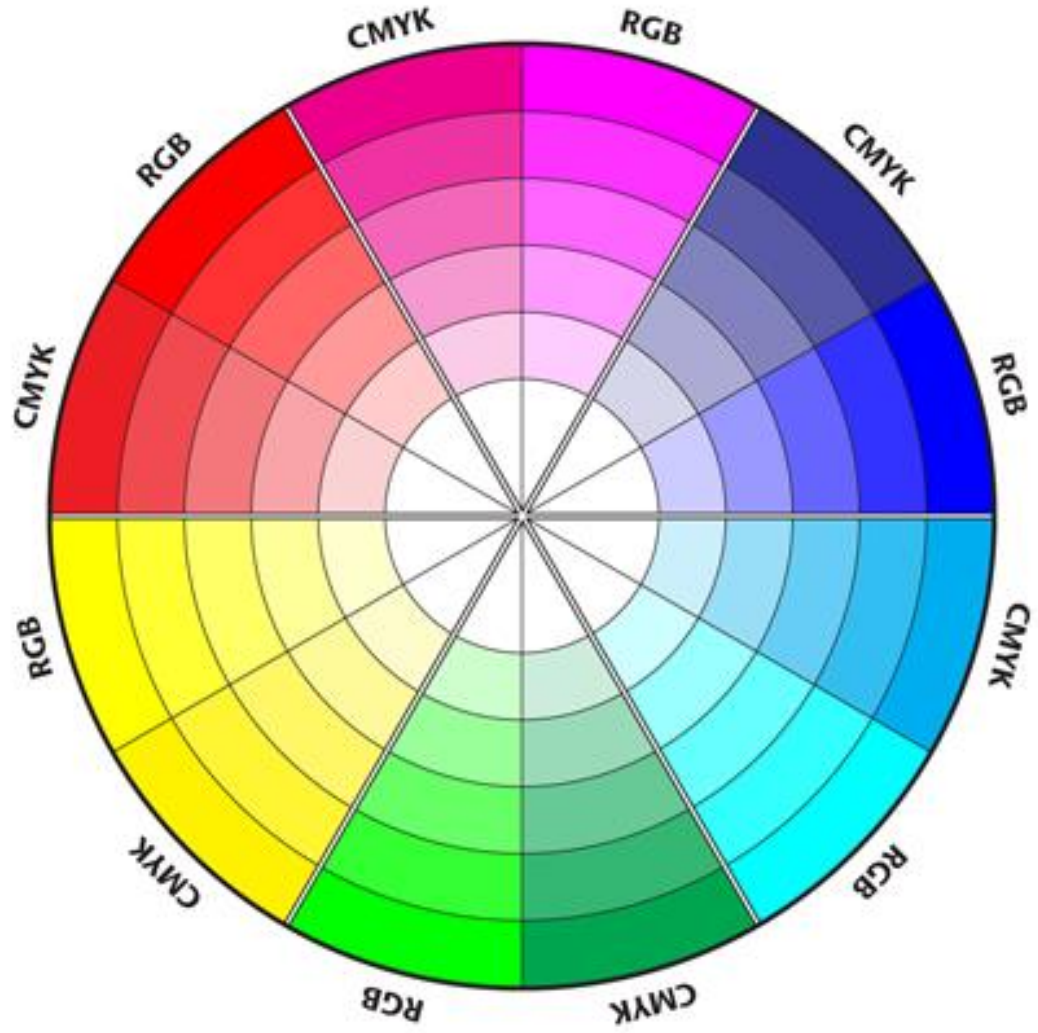

Data Collection and Preprocess
Sensor
RGB
Raw Depth
Depth
GT
RGB
HHA
Semantic 3D Reconstruction
Model
RGB&HHA
RGB&Depth
Prediction
SLAM
Train
COB
CRF
FCNs
BIM Generation
Doors
Windows
Floor&Ceiling
Walls
Lines Detection
2D Projection
Point Cloud
Planes Detection

Color Space

SECTION 2

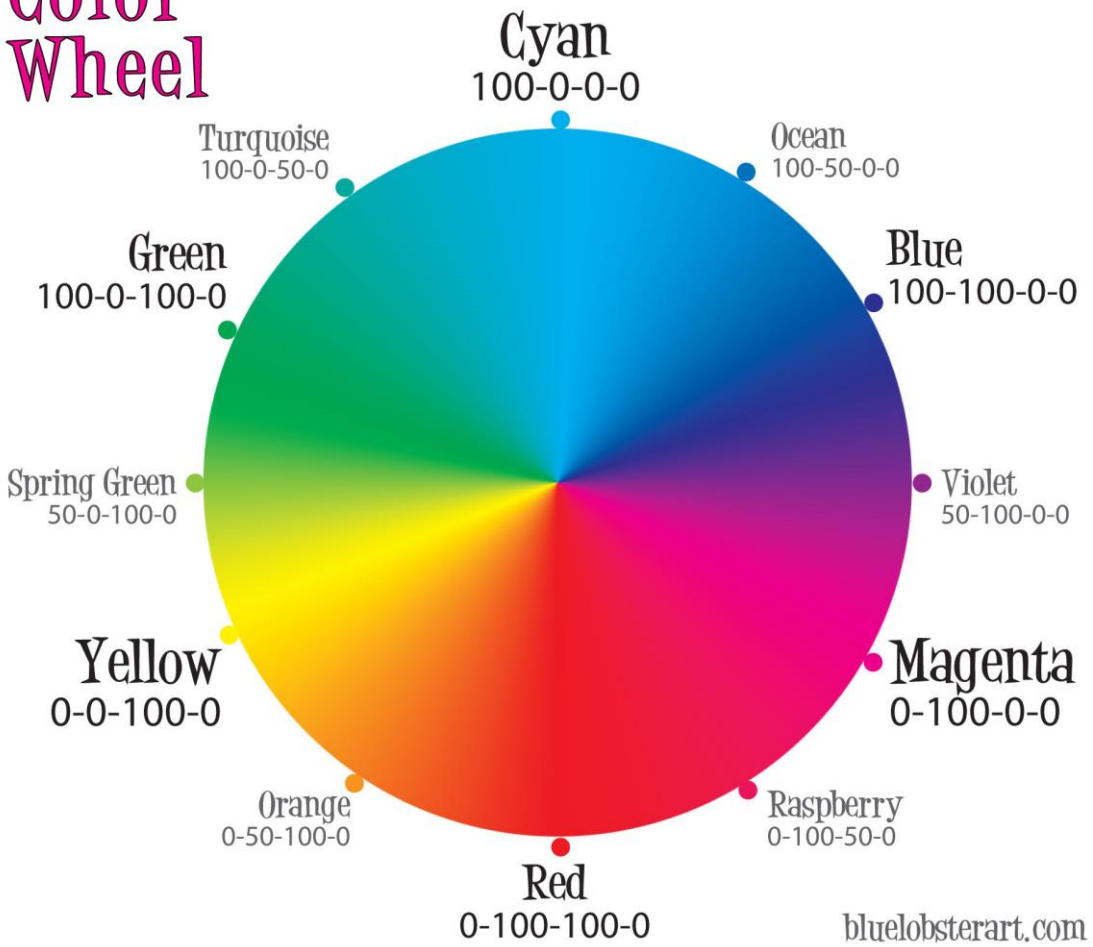
RGB vs CMYK

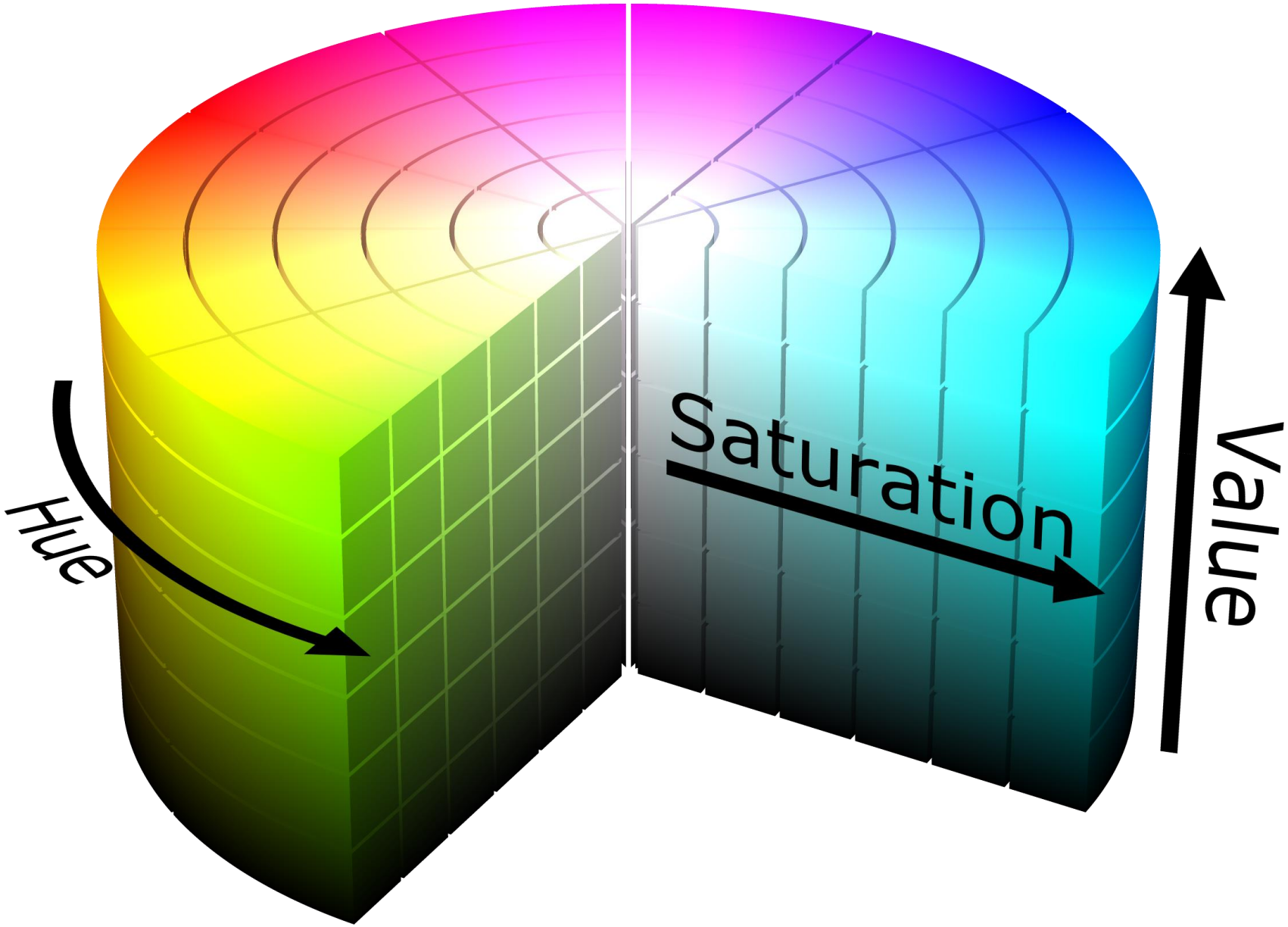


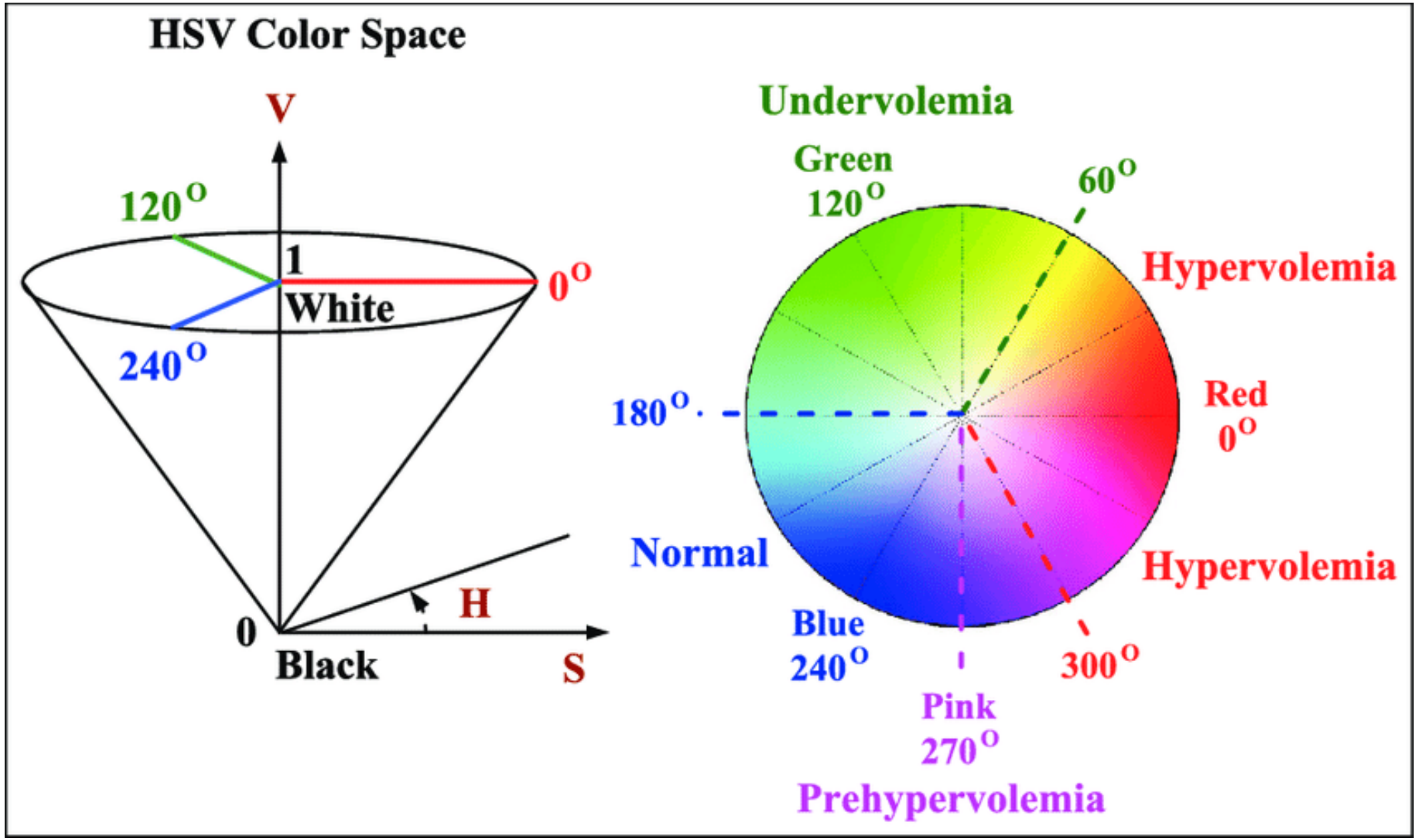


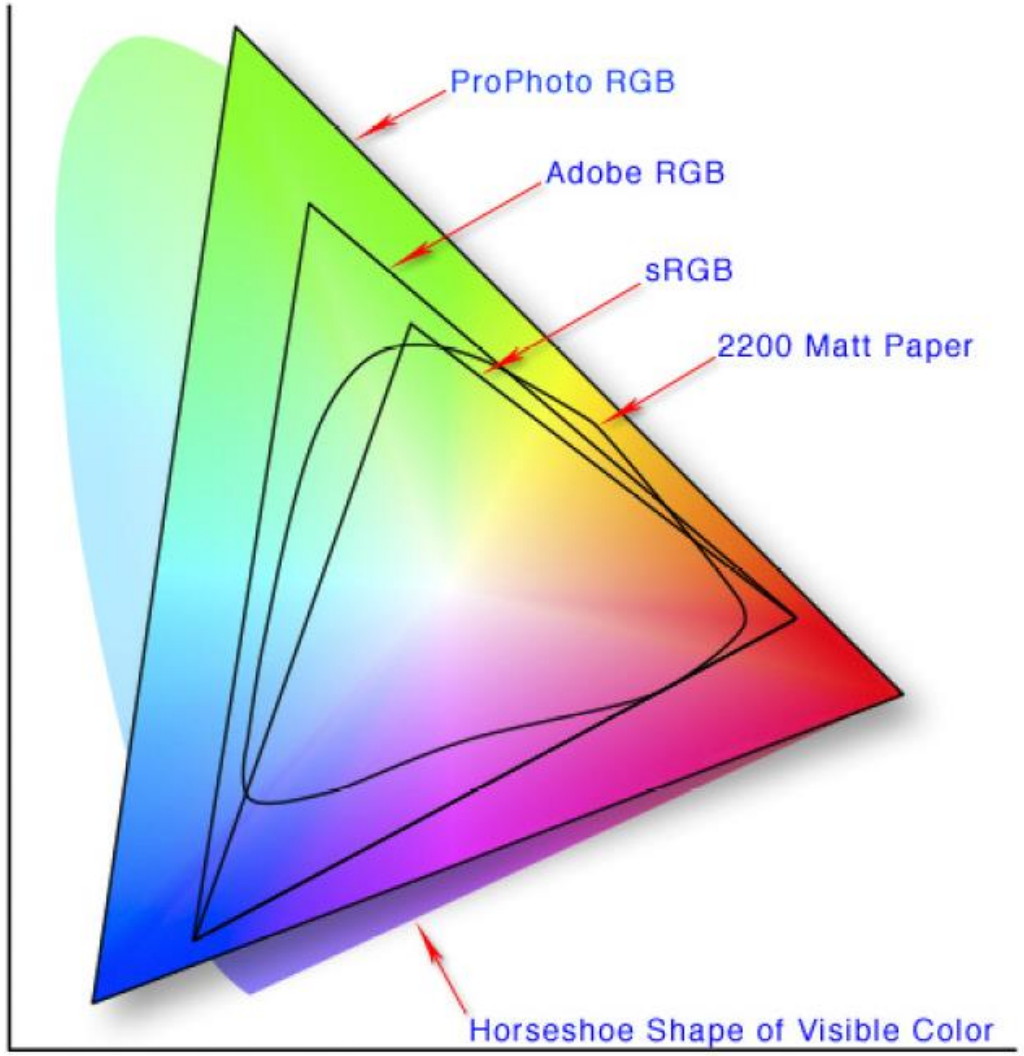
Color Wheel

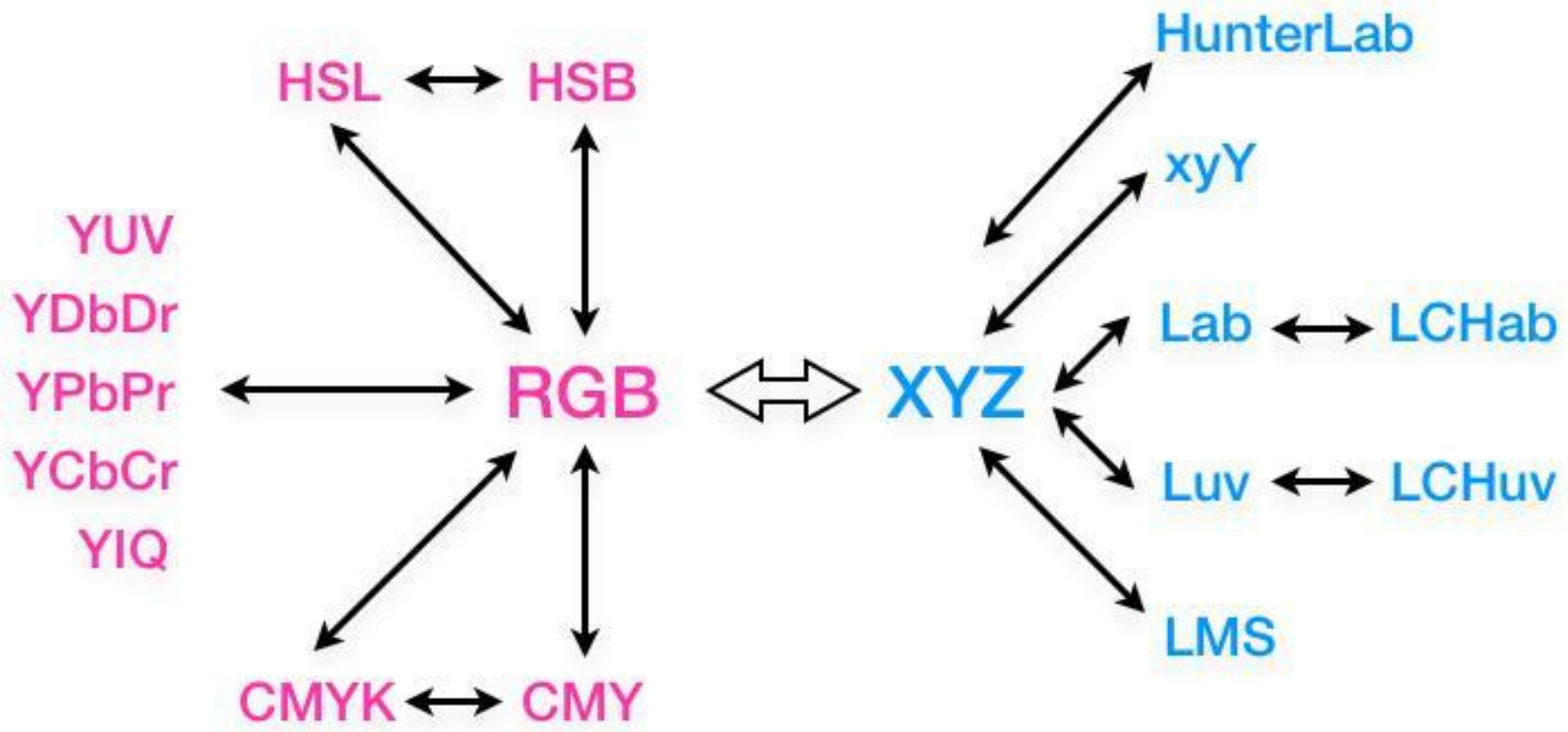
CMYK Color Wheel




Hue
Saturation
Value


HSV Color Space
V
120°
1
0°
White
240°
0
Black
H
S
Undervolemia
Green
120°
60°
Hypervolemia
180°
Red
0°
Normal
Hypervolemia
Blue
240°
300°
Pink
270°
Prehypervolemia


ProPhoto RGB
Adobe RGB
sRGB
2200 Matt Paper
Horseshoe Shape of Visible Color


HSL
HSB
YUV
YDbDr
YPbPr
YCbCr
YIQ
RGB
CMYK
CMY
XYZ
HunterLab
xyY
Lab
LCHab
Luv
LCHuv
LMS

Python Code Extraction of Single Color Images

SECTION 3

Create Single Color Image

```
def copyImage(f):
    g = copy.copy(f)
    return g

def allRedImage(f):
    redImage = copyImage(f)
    for i in range(len(redImage)):
        for j in range(len(redImage[0])):
            redImage[i][j][1] = 0
            redImage[i][j][0] = 0
    return redImage
```

RGB in cv2 Frame

pixel[2] pixel[1] pixel[0]

Gray Color

SECTION 4

Compositive Color Space

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.159 & -0.332 & 0.050 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} x \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Calculation of the Illuminance level.

Get Illuminance Level

```
def getGrayImage(f):
    grayImage = copyImage(f)
    for i in range(len(grayImage)):
        for j in range(len(grayImage[0])):
            gray = np.uint8(0.299*f[i][j][2]+0.587*f[i][j][1]+0.144*f[i][j][0])
            grayImage[i][j][2] = gray
            grayImage[i][j][1] = gray
            grayImage[i][j][0] = gray
    return grayImage
```