



TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

NHẬP MÔN XỬ LÝ ẢNH

TS. ĐOÀN THỊ HƯƠNG GIANG

Email: giangdth@epu.edu.vn

Mobil: 0372630593

Thông tin môn học

- **Số tín chỉ: 02**

- Lý thuyết: 30 tiết

- **Đánh giá môn học:**

- a) **Kiểm tra – đánh giá thường xuyên, kiểm tra định kỳ (30%) tính bằng trung bình cộng các bài kiểm tra**

- Bài 1: Báo cáo chuyên đề (30 phút)

- Bài 2: Báo cáo chuyên đề (30 phút)

- b) **Thi cuối kỳ (70%)**

- Báo cáo chuyên đề (45 phút)

Học liệu

- **1. Tài liệu học tập (Sách, giáo trình chính)**

[1] Đỗ Khánh Vân, *Xử lý ảnh bằng kỹ thuật số*, NXB Khoa học kỹ thuật, 2005. **VL1103097-3100**

- **2. Tài liệu tham khảo**

[2] Rafael C. Gonzalez & Richard E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

Nội dung môn học

- Chương 1: Giới thiệu về xử lý ảnh
- Chương 2: Tăng cường ảnh qua xử lý lược đồ
- Chương 3: Tăng cường ảnh qua lọc không gian
- Chương 4: Phân đoạn ảnh

Chương 1: Giới thiệu về xử lý ảnh

1. Khái niệm ảnh

- Thông tin về vật thể hay quang cảnh được chiếu sáng mà con người quan sát và cảm nhận được bằng mắt và hệ thống thần kinh thị giác
- Biểu diễn ảnh về mặt toán học:

$$I = F(x, y)$$

Trong đó:

- ✓ x, y là tọa độ không gian hai chiều
- ✓ F là độ lớn của độ chói (ảnh đơn sắc), màu (đối với ảnh màu)
- ✓ x, y biến thiên liên tục và F cũng liên tục

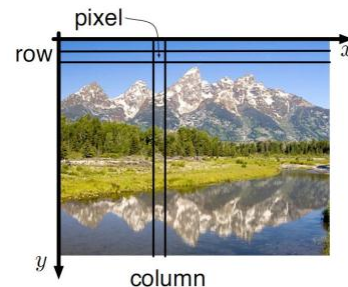
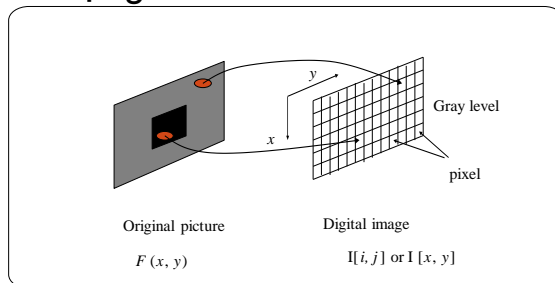
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

5

Khái niệm ảnh số

- Ảnh số là ảnh thu được từ ảnh liên tục bằng phép lấy mẫu và lượng tử hóa



Một ảnh số thường được biểu diễn như một ma trận các điểm ảnh. Trong đó mỗi điểm ảnh có thể được biểu diễn bằng:

- 1 bit (ảnh nhị phân)
- 8 bit (ảnh đa mức xám)
- 16, 24 bit (ảnh màu)

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

6

Ảnh được biểu diễn ảnh dưới dạng ma trận các điểm ảnh gọi là ảnh bitmap

Ảnh bitmap

- **Ảnh Bitmap (Raster Image):**

- Là loại ảnh dùng một loạt các pixel (điểm ảnh) để biểu thị hình ảnh.
- Mỗi pixel (điểm ảnh) là một hình vuông được gán một vị trí và giá trị màu cụ thể tạo nên hình ảnh.

Ảnh vector

- **Ảnh Vector** là loại ảnh vector không dựa trên mẫu pixel mà được tạo bởi các đoạn thẳng và đường cong được định nghĩa bằng các đối tượng toán học gọi là Vector, từ đó tạo nên các hình tròn hoặc hình đa giác.

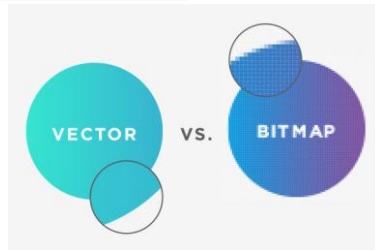
So sánh ảnh bitmap với ảnh vector

• Bitmap

- Biểu diễn các hình phức tạp hơn
- Tính toán chậm
- Hạn chế khi zoom, các phép biến hình
- Đuôi file: **PNG, JPG, BMP, JPEG và GIF...**

• Vector

- Biểu diễn các hình đơn giản
- Tính toán nhanh
- Chất lượng ảnh tốt
- Đuôi file: ***.EPS, *.AI, PDF *.CDR, or *.DWG.**



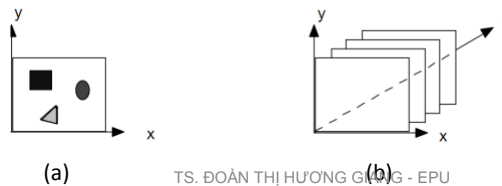
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

9

Ảnh tĩnh - Ảnh động

- Ảnh tĩnh (Image) là chỉ có một ảnh duy nhất – một khung hình duy nhất để biểu diễn đối tượng. Biểu diễn bởi hàm mô tả các giá trị độ chói của các điểm ảnh trong mặt phẳng ảnh $I(x,y)$ – hình (a)
- Ảnh động (Sequence Images) là tập hợp một số ảnh liên tiếp – một số khung hình liên tiếp để biểu diễn đối tượng. Biểu diễn bởi hàm mô tả các giá trị độ chói của các điểm ảnh trong mặt phẳng ảnh và biến thời gian $I(x,y,t)$ – hình (b)



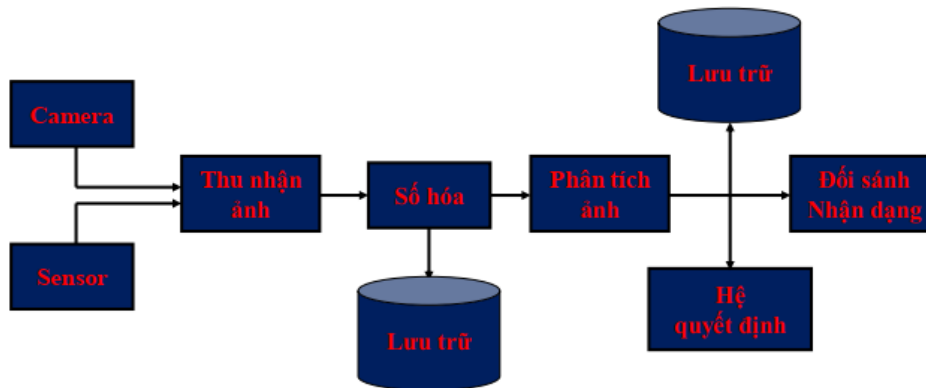
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

10

Mô hình hệ thống xử lý ảnh

- Các giai đoạn trong xử lý ảnh



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

11

Các vấn đề trong xử lý ảnh

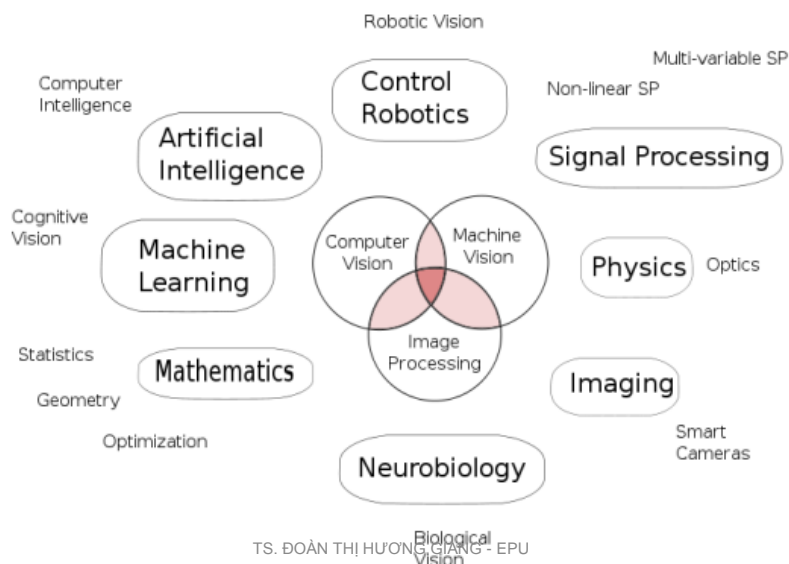
- Image acquisition (thu nhận ảnh, lấy mẫu, lượng tử hóa, nén)
- Image to Image (tăng cường ảnh, khôi phục ảnh, phân đoạn ảnh, lọc nhiễu)
- Image to parameter (trích chọn đặc trưng: feature extraction, feature selection).
- Parameter to decision (Nhận dạng: recognition, interpretation)

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

12

Các lĩnh vực liên quan



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

13

Các ứng dụng của xử lý ảnh

- Xử lý ảnh vệ tinh, ảnh viễn thám
- Thiên văn, nghiên cứu không gian, vũ trụ
- Thăm dò địa chất
- Lĩnh vực y tế
- Robot, tự động hóa
- Giám sát, phát hiện chuyển động
- Image và video retrieval
-

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

14

Matrix và vector

- Các phép xử lý ảnh thực chất là các phép tính toán trên:
 - Các ma trận
 - Các vectors

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

15

Một số khái niệm trong toán học về matrix và vector

- Khái niệm ma trận: m dòng, n cột
- A là vuông (square) nếu $m = n$
- A là ma trận đường chéo (diagonal):
 - là ma trận có các phần tử không nằm trên đường chéo $= 0$
 - và có ít nhất một phần tử trên đường chéo $\neq 0$
- A là ma trận đơn vị (identity - I): nếu ma trận đường chéo có tất cả các phần tử trên đường chéo đều $= 1$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 13 \\ 0 & 0 & 0 & 17 \end{pmatrix}$$

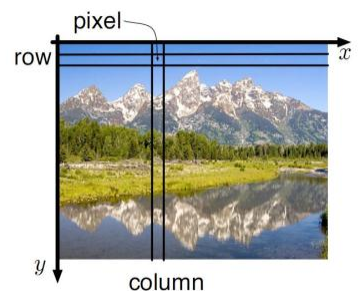
diagonal

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

identity

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

x
y



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

16

Lưu ý cài đặt python + openCV

- Các thư viện opencv-python, PIL, numpy
- Cài đặt:
 - **pip install opencv-python**
 - **pip install Pillow**
 - **Pip install numpy**
- Website tham khảo:
 - <https://pypi.org/project/opencv-python/>
 - <https://pypi.org/project/Pillow/>

Đọc và ghi ảnh

- Đọc video từ một file
- Đọc một chuỗi hình ảnh
- Đọc video từ webcam
- Ghi video

Đọc ảnh từ một ảnh từ file bằng python

Cú pháp:

```
img = cv2.imread(path, flag)
```

• Tham số:

✓ **path:** Đường dẫn chỉ tới thư mục chứa ảnh.

✓ **flag:** Loại ảnh được đọc về

- Mặc định là ảnh màu **cv2.IMREAD_COLOR**; có thể nhập tương đương giá trị **1**
- **cv2.IMREAD_GRAYSCALE**: ảnh xám - grayscale mode. Tương đương giá trị **0**.
- **cv2.IMREAD_UNCHANGED**: loại alpha channel. Tương đương giá trị **-1**.

✓ **Giá trị trả về:** Ảnh

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

19

Áp dụng

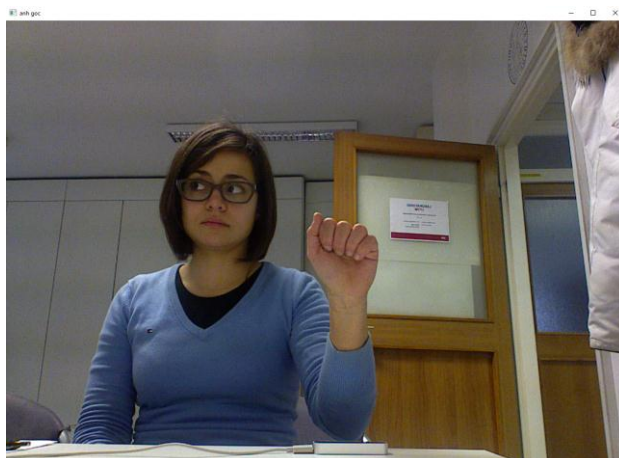
```
import cv2
img = cv2.imread("images\\1_rgb.png")
cv2.imshow("anh goc",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

20

Kết quả chạy chương trình



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

21

Đọc ảnh từ camera

- Đọc video từ webcam cũng gần tương tự như đọc từ file.
 - Khai báo chỉ số thiết bị webcam như sau:
 - Nếu thiết bị có một webcam tích hợp, thì chỉ số thiết bị sẽ là '0'.
 - Nếu có nhiều webcam được kết nối với thiết bị, thì chỉ số thiết bị được liên kết với mỗi webcam bổ sung sẽ tăng lên (ví dụ: 1, 2, v.v.).

- **Cú pháp:**

`Vid/cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)`

Trong đó: `CAP_DSHOW` là tùy chọn API quay video khác

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

22

Đọc ảnh từ camera của PC bằng python (tt)

```
import cv2
cap = cv2.VideoCapture(0)
while(True):
    ret, frame = cap.read()
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

23

Đọc ảnh từ một video ảnh từ file bằng python

- Cú pháp:

VideoCapture(Path, apiPreference)

- Trong đó:

- Path: là tên và đường dẫn tới video.
- “apiPreference”: chọn API.

- Python code:

Đọc video từ file lưu trên ổ cứng

```
vid_capture = cv2.VideoCapture('Resources/Cars.mp4')
```

```
if (vid_capture.isOpened() == False):
    print("Error opening the video file")
else:
    # Get frame rate information

    fps = int(vid_capture.get(5))
    print("Frame Rate : ",fps,"frames per second")

    # Get frame count
    frame_count = vid_capture.get(7)
    print("Frame count : ", frame_count)
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

24

Đọc một chuỗi hình ảnh

- Tương tự với việc xử lý khung hình từ một luồng video. Sử dụng ký hiệu được hiển thị bên dưới (Cars%04d.jpg)
- Cú pháp:


```
vid = cv2.VideoCapture('Resources/Image_sequence/Cars%04d.jpg')
```

 - Trong đó:
 - “%04d” là quy ước đặt tên theo dãy gồm bốn chữ số
 - Ví dụ: Cars0001.jpg, Cars0002.jpg, Cars0003.jpg, ...
 - Ví dụ: “Race_Cars_%02d.jpg” thì sẽ tìm kiếm các file có dạng:
 - Race_Cars_01.jpg, Race_Cars_02.jpg, Race_Cars_03.jpg, ...

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

25

Ghi video

- Để ghi video cần tạo một đối tượng ghi video từ lớp VideoWriter () với cú pháp như sau:


```
VideoWriter(filename, apiPreference, fourcc, fps, frameSize[, isColor])
```
- Lớp VideoWriter () có các tham số:
 - ✓ filename: tên đường dẫn lưu video
 - ✓ apiPreference: Mã API
 - ✓ Fourcc: Mã 4 ký tự của codec, được sử dụng để nén ảnh
 - ✓ fps: Tốc độ khung hình của luồng video
 - ✓ frame_size: Kích thước ảnh trong video
 - ✓ isColor:
 - ✓ ≠ 0 ảnh màu.
 - ✓ = 0: ảnh xám (hiện chỉ được hỗ trợ trên Windows).
 - ✓ Codec video: cách nén luồng video, chuyển đổi video không nén sang định dạng nén hoặc ngược lại. Để tạo AVI hoặc MP4 thì sử dụng fourcc sau:
 - ✓ AVI: cv2.VideoWriter_fourcc ('M', 'J', 'P', 'G')
 - ✓ MP4: cv2.VideoWriter_fourcc (* 'XVID') Hai đối số đầu vào tiếp theo chỉ định tốc độ khung hình FPS và kích thước khung hình (chiều rộng, chiều cao).

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

26

```

1.  fps = 20
2.  vid_capture = cv2.VideoCapture(0)
3.  # sử dụng phương thức get () để lấy chiều rộng và chiều cao khung hình video
4.  frame_width = int(vid_capture.get(3))
5.  frame_height = int(vid_capture.get(4))
6.  frame_size = (frame_width, frame_height)
7.  # Khởi tạo đối tượng ghi video
8.  output = cv2.VideoWriter('Resources/output_video_from_file.avi',
    cv2.VideoWriter_fourcc('M', 'J', 'P', 'G'), 20, frame_size)
9.  while(vid_capture.isOpened()):
10.     ret, frame = vid_capture.read()
11.     if ret == True:
12.         # Ghi frame vừa đọc được từ camera
13.         output.write(frame)
14.     else:
15.         print('Stream disconnected')
16.         break

```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

27

Cách truy cập một điểm ảnh trong python

• **Cú pháp:**

```
Pval = Img[idx_h, idx_w, [,channels]]
```

• **Tham số:**

- ✓ idx_h: chỉ số hàng
- ✓ idx_w: chỉ số cột
- ✓ channels: Các kênh màu
- ✓ **Giá trị trả về:** giá trị điểm ảnh

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

28

Áp dụng

Lấy và thay đổi giá trị pixel của ảnh màu

```
import cv2
img = cv2.imread("images\\1_rgb.png")
cv2.imshow("anh goc", img)
px = img[0,0]           # lay gia tri diem anh tai toa do (0,0)
for i in range(500):
    img[0:30,i] = (0,255,0) # thay doi gia tri diem anh
print('pixel:',px)       # hien thi gia tri pixel
cv2.imshow("anh thay doi",img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

29

Kết quả chạy chương trình



Ảnh gốc



Ảnh thay đổi giá trị pixel

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

30

Lấy kích thước ảnh trong python

• **Cú pháp:**

`Dim = Img.shape`

Lấy thông tin tổng hợp cả ba thành phần:
(0,1,2) <-> (height, width, channel)

Giá trị trả về:

• **Dim: Số chiều ảnh**

- ✓ Height: chiều cao ảnh
- ✓ Width: chiều rộng ảnh
- ✓ Channels: Số kênh ảnh

`Heigh = Img.shape[0]`
`Width = Img.shape[1]`
`Channel = Img.shape[2]`

Áp dụng

```
import cv2
img = cv2.imread("images\\1_rgb.png")
dim = img.shape
height = img.shape[0]
width = img.shape[1]
channels = img.shape[2]
print('Dimension : ',dim)
print('Height : ',height)
print('Width : ',width)
print('Channels : ',channels)
```


Một số khái niệm (tt)

- $\text{Trace}(A) = \sum$ Các phần tử trên đường chéo chính
- Ma trận chuyển vị (**transpose**):
 - dòng \rightarrow cột, cột \rightarrow dòng,
 - ký hiệu: A^T
- Ma trận vuông A đối xứng (**symetric**) nếu: **Ko có**
 - $A = A^T$
- Ma trận nghịch đảo (**Inverse**): X là inverse của A nếu: **Ko có**
 - $XA = I$ và $AX = I$

Hàm Trace trong python

- **Cú pháp:**
`cv2.trace(mtx)`
- **Tham số:**
 - `mtx`: ma trận đầu vào
- **Giá trị trả về:**
 - Trả về tổng các thành phần thuộc đường chéo của ma trận `mtx`.

Áp dụng

```
import cv2
img = cv2.imread("images\\1_rgb.png")
retval = cv2.trace(img)
print('tong gia tri duong cheo cua ma tran:',retval)
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

35

Hàm Transpose trong python

- **Có pháp:**

`cv2.transpose(src[, dst])`

- **Tham số:**

- *src*: ma trận đầu vào
- *dst*: ma trận đầu ra

- **Trả về:**

- Trả về một ảnh

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

36

Áp dụng

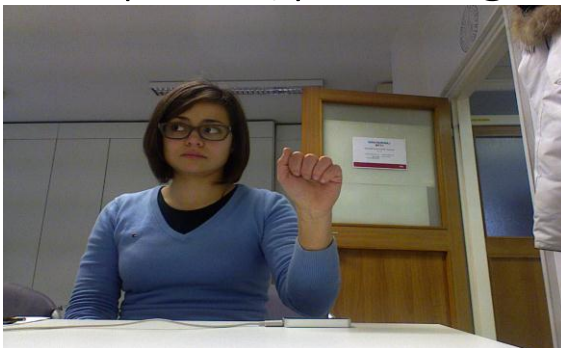
```
import cv2
img = cv2.imread("images\\1_rgb.png")
img2 = cv2.transpose(img)
cv2.imshow("anh goc",img)
cv2.imshow("anh goc transpose",img2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

37

Kết quả chạy chương trình



Ảnh gốc



Ảnh sau transpose

05/01/2023

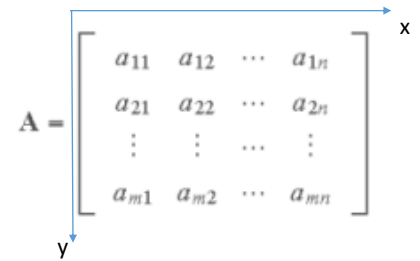
TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

38

Một số khái niệm (tt)

- Vector cột (column vector) là ma trận $m \times 1$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$$



$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

- Vector hàng (row vector) là ma trận $1 \times m$

$$\mathbf{b} = [b_1, b_2, \dots, b_n]$$

Lấy một cột hoặc một hàng của ảnh trong python

- Lấy một hàng:
row=image[idx,:] – idx: chỉ số hàng
- Lấy một cột:
col=image[:,idx] – idx: chỉ số cột

Áp dụng

```
import cv2
import numpy as np
img = cv2.imread("images\\1_rgb.png") # read image
img_b,_,_ = cv2.split(img) # split into 3 channels of image
idx=0
row = img_b[idx,:] # get a row = 0
col = img_b[:,idx] # get a col = 0
print('row size:',row.shape)
print('col size:',col.shape)
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

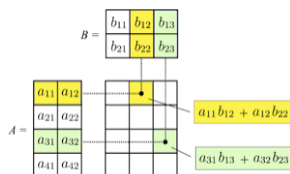
41

Một số khái niệm (tt)

$$A + B = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

- Các phép tính trong ma trận
- A, B cùng kích thước m x n.
 - $C = A + B \rightarrow C$ kích thước m x n và $C_{ij} = A_{ij} + B_{ij}$
 - $D = A - B \rightarrow D$ kích thước m x n và $D_{ij} = A_{ij} - B_{ij}$
- $A(m, n) = A_{ij}(m, n); B(n, q) = B_{ij}(n, q)$
 - $C = A(m,n) \times B(n,q) \rightarrow C$ kích thước m x q và $C_{ij} = \sum_{k=0}^n (A_{ik} * B_{kj})$



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

42

$$A = \begin{pmatrix} 1 & 2 & 5 \\ 0 & -1 & 1 \end{pmatrix}; B = \begin{pmatrix} 4 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 2 & -2 \end{pmatrix} \Rightarrow AB = \begin{pmatrix} 9 & 12 & -11 \\ 1 & 1 & -1 \end{pmatrix}.$$

Một số khái niệm (tt)

• Các phép tính trong ma trận

- Cho 2 vector $A(1 \times m)$, $B(1 \times m)$ cùng kích thước.
- Tích vô hướng 2 vector (inner product – dot product) được định nghĩa như sau:

$$A^T * B = A * B^T = \sum_{i=1}^m (a_i * b_i)$$

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

43

Một số khái niệm (tt)

• Chuẩn của vector (vector norm)

- Vector norm của vector x ký hiệu $\|x\|$, còn gọi là 2-norm hay khoảng cách Euclidean

$$\|x\| = [x_1^2 + x_2^2 + \dots + x_m^2]^{1/2}$$

$$\|x\| = [x^T x]^{1/2}$$

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

44

Một số khái niệm (tt)

- Pixel:
 - Picture element là đơn vị nhỏ nhất cấu tạo nên ảnh số
 - Mỗi pixel có tọa độ (x,y) và giá trị cường độ sáng hoặc màu sắc tại điểm đó
- Độ phân giải của ảnh:
 - Số pixel có trong ảnh để tạo nên bức ảnh đó
 - Thường ghi dưới dạng: m x n
 - m: số pixel trên chiều rộng ảnh
 - n: số pixel trên chiều cao ảnh
 - Độ phân giải càng cao, ảnh càng sắc nét

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

45

Một số khái niệm (tt)

- Mức xám (gray)
 - Mức xám là kết quả của việc mã hoá ứng với một cường độ sáng của mỗi điểm ảnh với một giá trị số.
 - Thông thường ảnh được mã hoá dưới dạng 16, 32, 64 hay 256 mức.
- Ví dụ:
 - Tại điểm ảnh tọa độ (20, 40) có mức xám là 60,
 - tại điểm ảnh tọa độ (30, 40) có mức xám là 23, ...

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

46

Hàm chuyển từ ảnh ba kênh màu sang ba kênh xám trong python

- **Cú pháp:**

img = cv2.cvtColor(*src*, *code*[, *dst*[, *dstCn*]])

- **Tham số:**

- **src:**

- Ảnh đầu vào

- Thuộc loại 8 bit unsigned (CV_8U) hoặc 16 bit unsigned (CV_16U)

- **code:** quy định mã màu sẽ chuyển đổi

- **dst:** Ảnh đầu ra có cùng kích thước với đầu vào.

- **dstCn:**

- Số lượng kênh trong hình ảnh đích.

- Nếu tham số là 0 thì số kênh được lấy tự động từ src và code.

- Nó là một tham số tùy chọn.

- **Giá trị trả về:** Ảnh.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

47

Áp dụng

1. import cv2
2. img = cv2.imread("images\\1_rgb.png")
3. **img_gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)**
4. cv2.imshow('anh goc',img)
5. cv2.imshow('anh gray',img_gray)
6. cv2.waitKey(0)
7. cv2.destroyAllWindows()

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

48

Kết quả chạy chương trình



Ảnh gốc



Ảnh xám

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

49

Một số khái niệm (tt)

- Lân cận (neighbours): Một điểm ảnh p tại tọa độ (x, y) có
- 4 lân cận ngang - dọc của p :

Ký hiệu là $N_4(p)$

$(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$

- 4 lân cận chéo của p :

Ký hiệu là $ND(p)$

$(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$

- 8 lân cận của p :

Ký hiệu $N_8(p)$ là sự kết hợp của $N_4(p)$ và $ND(p)$

$(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$,
 $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

50

Một số khái niệm (tt)

- Liên thông: Các điểm trong ảnh gọi là liên thông với nhau nếu
 - Là lân cận của nhau
 - Và có cùng giá trị mức xám

05/01/2023

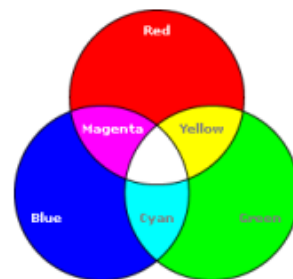
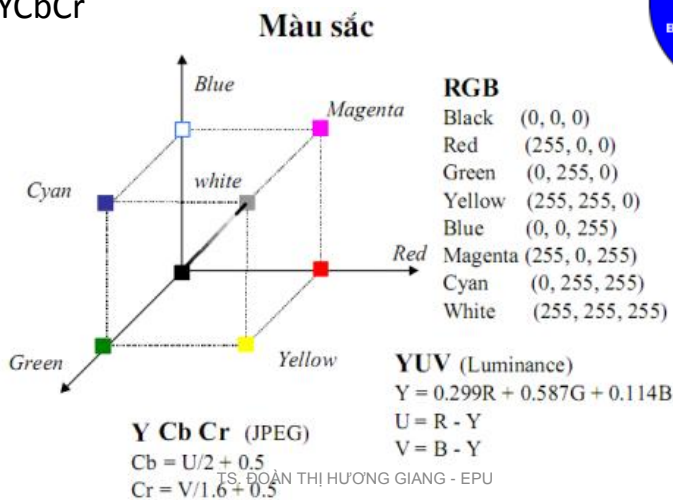
TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

51

<https://aiots.vn/phan-8-cac-khong-gian-mau-trong-opencv/>

Các hệ màu cơ bản

- RGB, YUV, YCbCr



05/01/2023

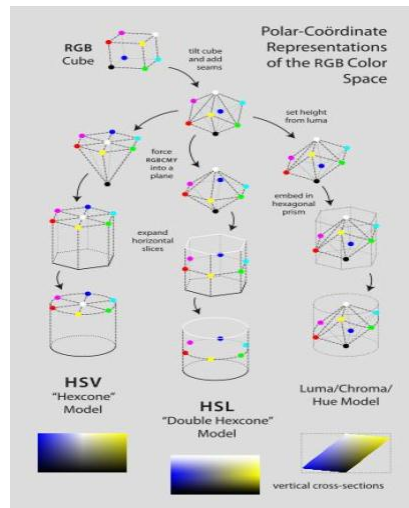
TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

52

Chuyển đổi giữa các hệ màu

- Chuyển đổi giữa các không gian màu khác nhau bằng cách sử dụng hàm:

cvtColor()



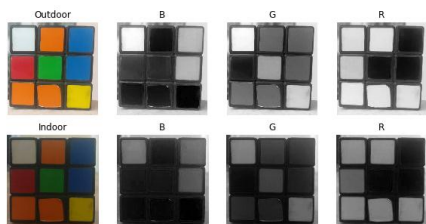
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

53

Không gian màu RGB

- Là một không gian màu kết hợp khi mà tất cả các màu thu được bằng sự kết hợp tuyến tính của các giá trị Đỏ(red), Xanh lục (green) và Xanh lam (blue).
- Ba kênh tương quan với nhau bởi lượng ánh sáng chiếu vào bề mặt.



Các kênh khác nhau Xanh lam (B), Xanh lục (G), Đỏ (R) của không gian màu RGB được hiển thị riêng biệt

Nhận xét:

- ✓ Quan sát kênh B, có thể thấy rằng mặt màu xanh và trắng trông giống nhau trong điều kiện ánh sáng trong nhà nhưng khác biệt rõ ràng với điều kiện ngoài trời.
- ✓ Sự không đồng nhất này làm cho việc phân đoạn ảnh dựa trên màu sắc rất khó khăn trong không gian màu này.
- ✓ Không gian màu RGB là sự trộn lẫn thông tin của từng kênh gồm thông tin liên quan đến màu sắc (sắc độ) thông tin liên quan đến cường độ (độ chói).

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

54

Không gian màu LAB

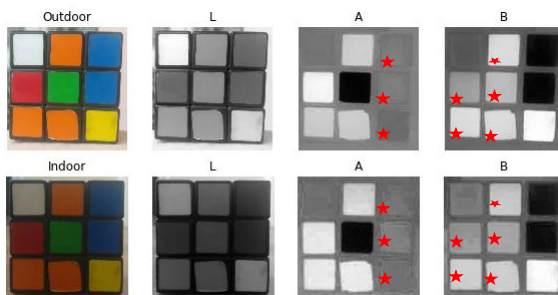
- Không gian màu Lab có ba thành phần.
 - **L: Độ sáng (Cường độ).**
 - a: thành phần màu từ Xanh lục đến Đỏ tươi.
 - b: thành phần màu từ Xanh lam đến Vàng.
- Không gian màu Lab khá khác với không gian màu RGB.
 - Trong không gian màu RGB, thông tin màu được tách thành ba kênh nhưng ba kênh giống nhau **cũng mã hóa thông tin độ sáng**.
 - Trong khi đó không gian màu Lab, **kênh L độc lập với thông tin màu và chỉ mã hóa độ sáng. Hai kênh còn lại mã hóa màu.**
- Không gian màu Lab có các thuộc tính sau:
 - Không gian màu đồng nhất về mặt tri giác, gần đúng với cách chúng ta cảm nhận màu sắc.
 - Độc lập với thiết bị (chụp hoặc hiển thị).
 - Được sử dụng nhiều trong Adobe Photoshop.
 - Có liên quan đến không gian màu RGB bằng một phương trình biến đổi phức tạp.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

55

Không gian màu LAB (tt)



Hình ảnh trong không gian màu Lab được tách thành ba kênh

Nhận xét:

- Có thể thấy rằng sự thay đổi về độ chiếu sáng chủ yếu ảnh hưởng đến kênh L.
- Các kênh A và B chứa thông tin về màu sắc không có thay đổi lớn khi so sánh giữa điều kiện trong nhà hay ngoài sáng.
- Các giá trị tương ứng của Xanh lục, Cam và Đỏ (nằm trong dải của kênh A) không thay đổi trong kênh B và tương tự, các giá trị tương ứng của Xanh lam và Vàng (nằm trong dải của kênh B) không thay đổi trong kênh A.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

56

Không gian màu YCrCb

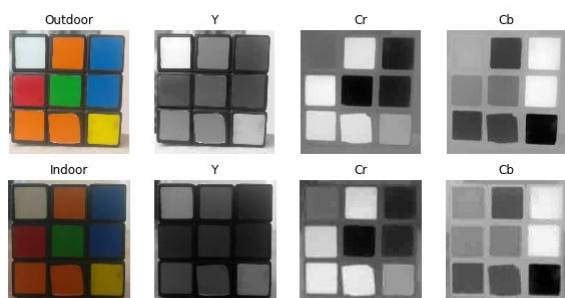
- Không gian màu YCrCb có nguồn gốc từ không gian màu RGB và có ba thành phần sau đây.
 - Y: Độ chói
 - $Cr = R - Y$: thành phần màu đỏ
 - $Cb = B - Y$: thành phần màu xanh lam
- Không gian màu này có các đặc tính sau:
 - Tách các thành phần độ chói và độ sắc thành các kênh khác nhau.
 - Chủ yếu được sử dụng trong nén (của các thành phần Cr và Cb) cho truyền hình.
 - Phụ thuộc vào thiết bị.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

57

Không gian màu YCrCb (tt)



hình ảnh gốc trong không gian màu YCrCb được tách thành các kênh riêng biệt

Nhận xét:

- Khi thay đổi độ sáng giữa 2 hình thì:
 - + các thành phần màu sắc và cường độ có đặc điểm như LAB.
 - + Chỉ ảnh hưởng đến Y, không ảnh hưởng tới Cr và Cb.
- So với LAB, sự khác biệt giữa màu đỏ và cam của YCrCb là không rõ ràng.
- Màu trắng ở 3 kênh đều khác nhau.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

58

Không gian màu HSV

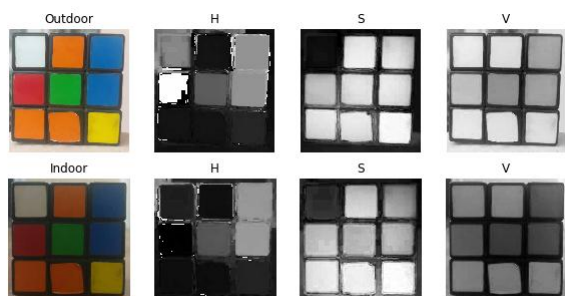
- Không gian màu HSV có ba thành phần sau
 - H: Hue (Bước sóng chi phối).
 - S: Độ bão hòa (Độ tinh khiết / sắc thái của màu).
 - V: Giá trị (Cường độ).
- Một số thuộc tính của HSV
 - Chỉ sử dụng một kênh để mô tả màu sắc (H), dẫn đến rất trực quan để biểu diễn màu sắc.
 - Phụ thuộc vào thiết bị.

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

59

Không gian màu HSV (tt)



Nhận xét:

- Kênh H rất giống nhau trong cả hai hình ảnh cho biết thông tin màu sắc vẫn còn nguyên vẹn ngay cả khi thay đổi ánh sáng.
- Thành phần S cũng rất giống nhau trong cả hai hình ảnh.
- Thành phần V phản ánh lượng ánh sáng chiếu vào nó, do đó nó thay đổi khi có sự thay đổi độ chiếu sáng.
- Có sự khác biệt giữa các giá trị của màu đỏ ở kênh H giữa 2 hình ảnh.

Các thành phần H, S và V
của hai hình ảnh

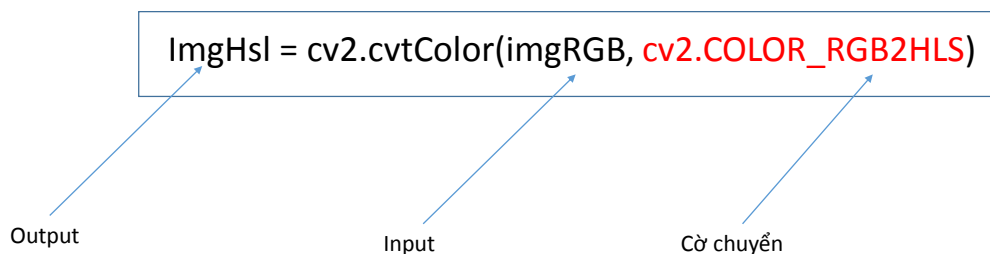
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

60

Hàm chuyển đổi giữa các hệ màu trong python

- Chuyển giữa hệ màu RGB → HSL trong python
- Cú pháp:



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

61

Áp dụng RGB → HLS

```
img = cv2.imread("images\\1_rgb.png")
img_hsl = cv2.cvtColor(img, cv2.COLOR_RGB2HLS)
cv2.imshow('anh goc',img)
cv2.imshow('anh hsl',img_hsl)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

62

Kết quả chạy chương trình



Ảnh gốc



Ảnh HSL

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

63

Hàm chuyển đổi giữa các hệ màu trong python (tt)

- Chuyển đổi giữa hệ màu RGB → HSV
- Cú pháp

```
ImgHSV = cv2.cvtColor(ImgRGB, cv2.COLOR_RGB2HSV)
```

Output

Input

Cờ chuyển

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

64

Áp dụng RGB → HSV

```
import cv2
img = cv2.imread("images\\1_rgb.png")
img_hsv = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
cv2.imshow('anh goc',img)
cv2.imshow('anh hsv',img_hsv)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

65

Kết quả chạy chương trình RGB → HSV



Ảnh gốc



Ảnh HSV

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

66

Hàm chuyển đổi giữa các hệ màu trong python

- Chuyển từ ảnh RGB → CMY
- Cú pháp:

```
ImgCMY = cv2.cvtColor(ImgRGB, cv2.COLOR_RGB2YCR_CB)
```

Output

Input

Cờ chuyển

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

67

Áp dụng RGB → CMY

```
import cv2
import numpy as np
img = cv2.imread("images\\1_rgb.png")
img = img.astype(np.float64)/255.
K = 1 - np.max(img, axis=2)
C = (1-img[...,2] - K)/(1-K)
M = (1-img[...,1] - K)/(1-K)
Y = (1-img[...,0] - K)/(1-K)
img_cmy= (np.dstack((C,M,Y,K)) * 255).astype(np.uint8)
cv2.imshow('anh goc',img)
cv2.imshow('anh cmy',img_cmy)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

68

Kết quả chạy chương trình RGB \rightarrow CMY



Ảnh gốc



Ảnh CMY

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

69

Hàm chuyển đổi giữa các hệ màu trong python

- Chuyển từ hệ màu RGB \rightarrow YUV
- Cú pháp:

```
ImgYUV = cv2.cvtColor(ImgRGB, cv2.COLOR_RGB2YUV)
```

Output

Input

Cờ chuyển

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

70

Áp dụng RGB \rightarrow YUV

```
import cv2
img = cv2.imread("images\\1_rgb.png")
img_yuv = cv2.cvtColor(img, cv2.COLOR_RGB2YUV)
cv2.imwrite('images\\1_rgb_cmy.png',img_yuv)
cv2.imshow('anh goc',img)
cv2.imshow('anh yuv',img_yuv)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

71

Kết quả chạy chương trình RGB \rightarrow YUV



Ảnh gốc



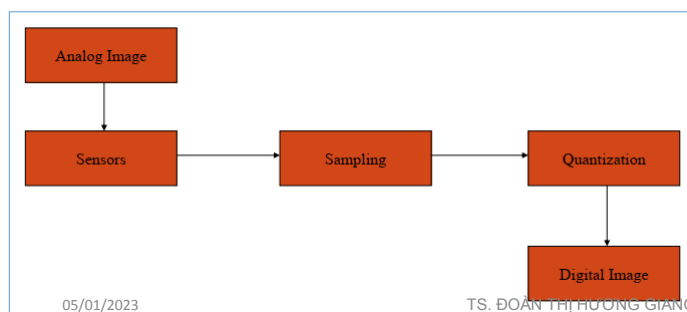
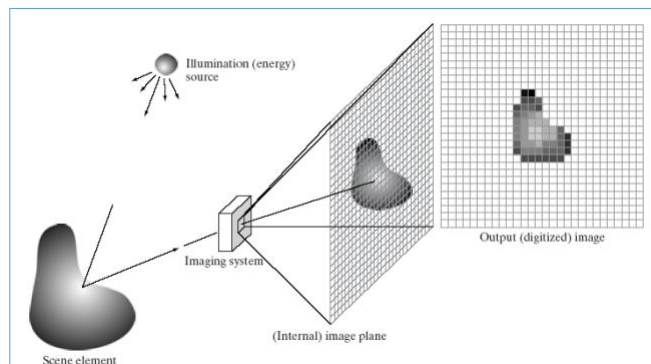
Ảnh YUV

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

72

Số hóa ảnh



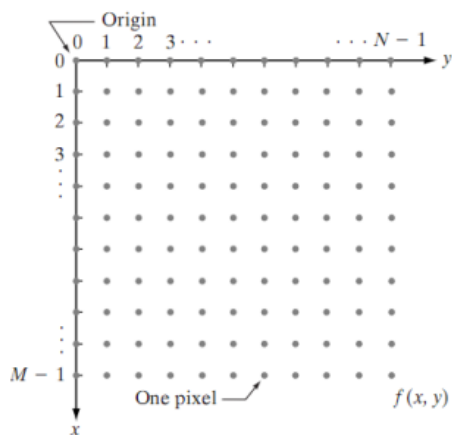
05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

73

Biểu diễn ảnh số

- $F(x, y)$, $F(l, j)$ hay $F(m, n)$



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

74

Ma trận dữ liệu ảnh số

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}.$$

$a_{ij} = f(x=i, y=j) = f(i, j)$ gọi là cấp xám của điểm ảnh tại tọa độ (i, j)

Dung lượng ảnh số

- L là số cấp xám sử dụng trong ảnh (số bước lượng tử hóa) $L = 2^k$
- $M \times N$: Kích thước của ma trận ảnh số (số mẫu khi lấy mẫu)
- Kích thước ảnh số: $M \times N \times k$ (bít)
- $M = N$ thì kích thước: $N^2 * k$

Tính toán dung lượng ảnh số

N/k	1 (L=2)	2 (L=4)	3 (L=8)	4 (L=16)	5 (L=32)	6 (L=64)	7 (L=128)	8 (L=256)
32	1.024	2.048	3.072	4.096	5.120	6.144	7.168	8.192
64	4.096	8.192	12.288	16.384	20.480	24.576	28.672	32.768
128	16.384	32.768	49.152	65.536	81.920	98.304	114.688	131.072
256	65.536	131.072	196.608	262.144	327.680	393.216	458.752	524.288
512	262.144	524.288	786.432	1.048.576	1.310.720	1.572.864	1.835.008	2.097.152
1024	1.048.576	2.097.152	3.145.728	4.194.304	5.242.880	6.291.456	7.340.032	8.388.608
2048	4.194.304	8.388.608	12.582.912	16.777.216	20.971.520	25.165.824	29.360.128	33.554.432
4096	16.777.216	33.554.432	50.331.648	67.108.864	83.886.080	100.663.296	117.440.512	134.217.728
8192	67.108.864	134.217.728	201.326.592	268.435.456	335.544.320	402.653.184	469.762.048	536.870.912

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

77

Độ phân giải không gian và độ phân giải mức xám

- Quá trình lấy mẫu → độ phân giải không gian trong ảnh
- Quá trình lượng tử hóa → độ phân giải mức xám trong ảnh

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

78

Độ phân giải ảnh

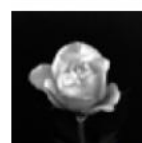
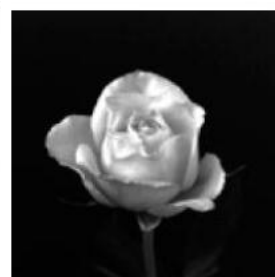


- Tiến hành lấy mẫu thưa dần để được các ảnh có kích thước nhỏ dần để được các ảnh có kích thước nhỏ dần:
- 1024x1024
- 512x512
- 256x256
- 128x128
- 64x64
- 32x32

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

79

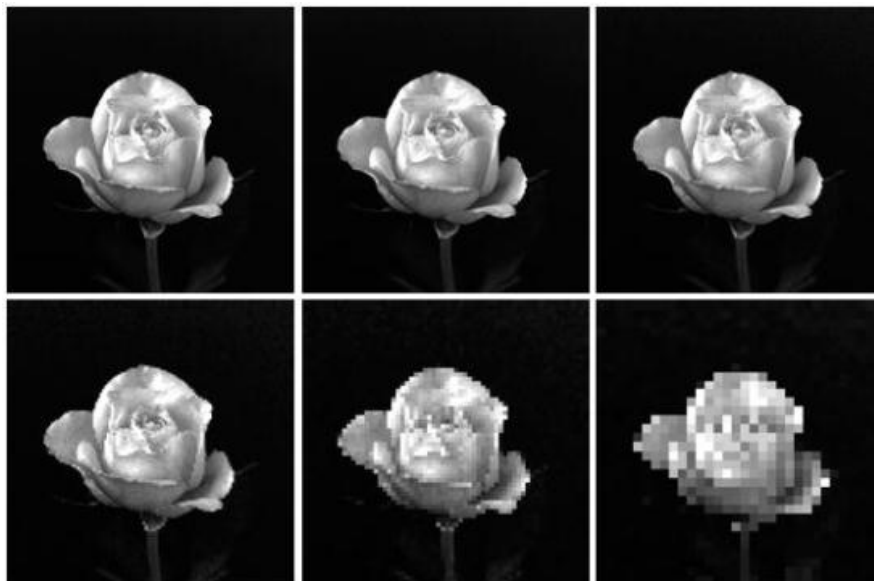


Ảnh được lấy mẫu theo các kích thước khác nhau

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

80



Các hình ảnh
trong Slide
trước được
phóng to thành
kích thước
1024×1024

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

81

Hàm thay đổi độ phân giải ảnh trong python

- `cv2.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])`

Tham số	Mô tả
src	[required] source/input image
dsize	[required] desired size for the output image
fx	[optional] scale factor along the horizontal axis
fy	[optional] scale factor along the vertical axis
interpolation	[optional] - INTER_NEAREST - INTER_LINEAR - INTER_AREA - INTER_NEAREST - INTER_CUBIC - INTER_LANCZOS4

Độ phân giải cấp xám

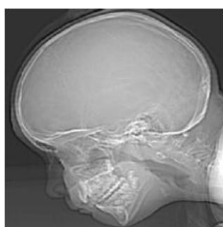


- Xét ảnh liên tục như hình bên.
- Chúng ta tiến hành lấy mẫu ảnh cùng một kích thước nhưng với số cấp xám nhỏ dần.
 - 128
 - 64
 - 32
 - 16
 - 8
 - 4
 - 2

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

83



(a) Ảnh gốc



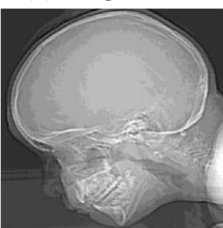
(b) Ảnh 128 cấp



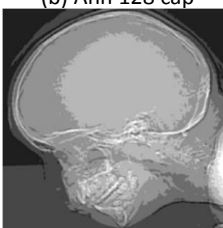
(c) Ảnh 64 cấp



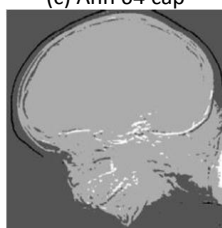
(d) Ảnh 32 cấp



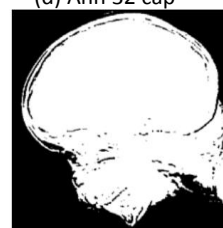
(e) Ảnh 16 cấp xám



(f) Ảnh 8 cấp xám



(g) Ảnh 4 cấp xám



(h) Ảnh 2 cấp xám

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

84

Phóng to và thu nhỏ ảnh số (Zooming & Shrinking)

- Bản chất của phóng to và thu nhỏ ảnh cũng giống như sampling
 - Zooming: oversampling
 - Shrinking: undersampling
- Zooming và shrinking: gồm 2 bước
 - Bước 1. Tạo ra các vị trí điểm ảnh mới
 - Bước 2. Gán giá trị mức xám cho các điểm ảnh mới

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

85

Zooming

- Bước 1. Tạo ra các vị trí điểm ảnh mới: đơn giản → tạo ra grid kích thước bằng kích thước muốn zoom lên
- Bước 2. Gán giá trị mức xám, có nhiều phương pháp:
 - o Nearest neighbor interpolation (nội suy gần nhất)
 - o Bilinear interpolation
 - o Pixel replication (nhân bản pixel)

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

86

Hàm phóng to thu nhỏ ảnh trong python

```
import cv2
img = cv2.imread('images\\1_rgb.png')
h,w = img.shape[:2]
alpha=1
img_z = cv2.resize(img,(int(alpha*w),int(alpha*h)))
# phong to
alpha = 1.5
img_z_out = cv2.resize(img,(int(alpha*w),int(alpha*h)))
#thu nho
alpha = 0.5
img_z_in = cv2.resize(img,(int(alpha*w),int(alpha*h)))
```

Các phép biến đổi ảnh

Các phép biến đổi ảnh dựa trên các phép toán logic

- Phép AND ảnh
- Phép OR ảnh
- Phép XOR
- Phép NOT

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

89

Phép AND ảnh

Cú pháp:

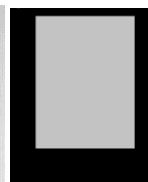
`cv2.bitwise_and(source1, source2, destination, mask)`

Tham số:

source1: Ảnh đầu vào 1; *Single-channel; 8-bit or floating-point*
source2: Ảnh đầu vào 2; *Single-channel; 8-bit or floating-point*
dest: Ảnh đầu ra,
mask: Mặt nạ, *8-bit single-channel*



Ảnh gốc



Mặt nạ AND



Phép AND ảnh

Python code:

```
img1 = cv2.imread('images\\palace.png',0)
img2 = cv2.imread('images\\background.png',0)
img_and = cv2.bitwise_and(img2, img1, mask = None)
cv2.imshow('Bitwise And', img_and)
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

90

Phép OR ảnh

Cú pháp:

`cv2.bitwise_or(source1, source2, destination, mask)`

Tham số:

source1: Đầu vào 1; Single-channel, 8-bit or floating-point

source2: Ảnh đầu vào 2; Single-channel, 8-bit or floating-point

dest: Ảnh đầu ra

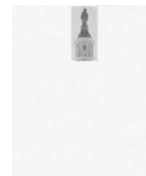
mask: Mặt nạ; 8-bit single-channel mask



Ảnh gốc



Mặt nạ OR



Phép OR ảnh

Python code:

```
img1 = cv2.imread('images\\palace_1.png',0)
img2 = cv2.imread('images\\background_3.png',0)
img_and = cv2.bitwise_or(img2, img1, mask = None)
cv2.imshow('Bitwise Or', img_and)
```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

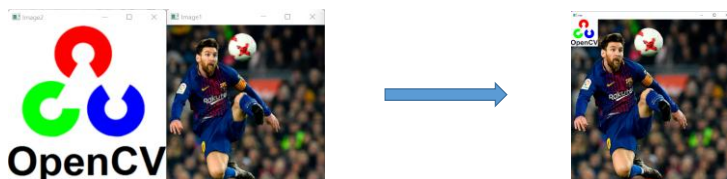
91

Bài tập

Bài 1: Hãy thực hiện các phép tính bit trên ảnh, giải thích:

- `cv2.bitwise_xor`
- `cv2.bitwise_not`

Bài 2: Yêu cầu là đặt logo OpenCV phía trên một hình ảnh.



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

92

Code

```

1.  # Load two images
2.  img1 = cv2.imread('messi.jpg')
3.  img2 = cv2.imread('opencv.png')
4.  # I want to put logo on top-left corner, So I create a ROI
5.  rows,cols,channels = img2.shape
6.  roi = img1[0:rows, 0:cols]
7.  # Now create a mask of logo and create its inverse mask also
8.  img2gray = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
9.  ret, mask = cv2.threshold(img2gray, 10, 255, cv2.THRESH_BINARY)
10. mask_inv = cv2.bitwise_not(mask)
11. # Now black-out the area of logo in ROI
12. img1_bg = cv2.bitwise_and(roi, roi, mask = mask_inv)
13. # Take only region of logo from logo image.
14. img2_fg = cv2.bitwise_and(img2, img2, mask = mask)
15. # Put logo in ROI and modify the main image
16. dst = cv2.add(img1_bg, img2_fg)
17. img1[0:rows, 0:cols] = dst
18. cv2.imshow('res', img1)
19. cv2.waitKey(0)
20. cv2.destroyAllWindows()

```

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

93

Các phép biến đổi ảnh dựa trên các phép toán số học

- Phép trừ: `cv2.subtract()`
- Phép cộng: `cv2.add()`
- Phép nhân: `cv2.multiply()`
- Phép chia: `cv2.divide()`
- Phép mũ bậc 2: `cv2.pow()`
- Phép căn bậc 2: `cv2.sqrt()`

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

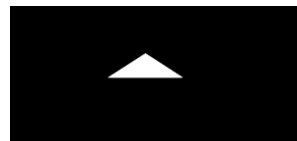
94

Phép trừ ảnh

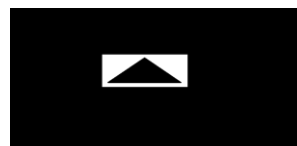
```
import cv2
import numpy as np
img = cv2.imread('images\\rectangle.png',0)
img1 = cv2.imread('images\\triangle.png',0)
img = cv2.resize(img, (640,480))
img1 = cv2.resize(img1, (640,480))
# subtract the images
img_sub = cv2.subtract(img1, img)
cv2.imshow('image', img_sub)
```



Ảnh 1



Ảnh 2



Ảnh trừ

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

95

Bài tập

Bài 1:

- Sinh viên lập trình và hiển thị các kết quả với các hàm còn lại?
- Giải thích?

Bài 2:

- Đọc ảnh từ camera, thực hiện trừ nền để có ảnh chỉ có người?

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

96

Trung bình ảnh

- $g(x,y)$ là ảnh nhiễu thu được bởi ảnh gốc $f(x,y)$ và nhiễu:

$$g(x,y) = f(x,y) + \eta(x,y)$$

- Với tập ảnh nhiễu $\{g(x,y)\}$:

$$\bar{g}(x,y) = \frac{1}{K} * \sum_{i=1}^K g_i(x,y)$$

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

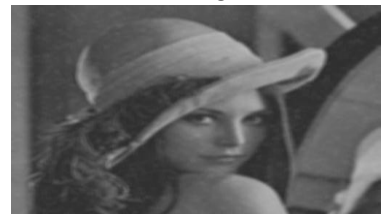
97

Ví dụ phép trung bình ảnh

```
import cv2
img = cv2.imread('images\\anhgoc_trungbinh.png')
img_mean = cv2.blur(img,(7,7))
cv2.imwrite('images\\anh_trungbinh.png',img_mean)
cv2.imshow('Anh goc',img)
cv2.imshow('Anh trung binh',img_mean)
cv2.waitKey(0)
cv2.destroyAllWindows();
```



Ảnh gốc



Ảnh trung bình

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

98

Bài tập

Cho ảnh

- 1) Tính và vẽ histogram trong các trường hợp: 8 bins, 16 bins, 32 bins
- 2) Nhận xét về độ tương phản của ảnh trên
- 3) Cân bằng histogram cho ảnh trên,
 - Vẽ histogram sau khi cân bằng
 - Tính toán lại các giá trị điểm ảnh ứng với histogram mới

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

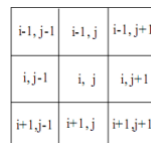
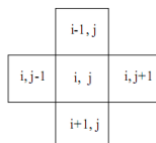
99

Mối quan hệ giữa các điểm ảnh

- **Lân cận (neighbor)**

Một pixel p tại vị trí (i, j) có thể có

- 4 lân cận tại các vị trí: (Ký hiệu $N4(p)$) $(i-1, j)$; $(i+1, j)$; $(i, j-1)$; $(i, j+1)$
- 8 lân cận tại các vị trí: (Ký hiệu $N8(p)$) $(i-1, j)$; $(i+1, j)$; $(i, j-1)$; $(i, j+1)$; $(i+1, j+1)$; $(i+1, j-1)$; $(i-1, j+1)$; $(i-1, j-1)$



05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

100

Liên kề (adjacency)

- V : tập các giá trị mức xám để xác định liên kề
- 2 pixel p, q có giá trị mức xám $\in V$ là
 - 4-adjacency: nếu $q \in N_4(p)$
 - 8-adjacency: nếu $q \in N_8(p)$
 - m-adjacency:
 - Nếu $q \in N_4(p)$, hoặc
 - $q \in N_8(p)$ và $N_4p \cap N_4q$ không có điểm nào có giá trị mức xám $\in V$

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

101

Ví dụ về liên kề

- $V = \{1\}$ - ảnh nhị phân

0	1	1
0	1	0
0	0	1

0	1	1
0	1	0
0	0	1

8 – adjacency

0	1	1
0	1	0
0	0	1

m - adjacency

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

102

Đường đi (path)

- Path từ điểm $p(x, y)$ đến $q(s, t)$ được định nghĩa là tập các pixel $(x_i, y_i), i = (1, \dots, m)$
- Sao cho: $(x_0, y_0) = (x, y); (x_n, y_n) = (s, t)$ và (x_i, y_i) và (x_{i-1}, y_{i-1}) là các điểm liền kề.
- Khái niệm: 4-paths, 8-paths, m-paths

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

103

Liên thông (connected)

- S : tập các pixel trong ảnh
- $p, q \in S$ gọi là liên thông nếu \exists 1 path nối giữa p và q
- Với mỗi pixel p , tập tất cả các điểm liên thông với nó gọi là vùng liên thông.
- Vùng, miền (region)
 - R là một tập các điểm ảnh: $R \in S$
 - R gọi là region nếu R là một vùng liên thông

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

104

Đường bao (boundary, border, contour)

- Đường bao của một region R là tập các điểm thuộc R mà có 1 hoặc nhiều điểm lân cận không thuộc R
- Chú ý:
 - Đường bao \neq đường biên (edge)
 - Đường bao \equiv đường biên: ảnh nhị phân

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

105

Khoảng cách giữa 2 điểm trong ảnh

- Khoảng cách Euclidean:

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{\frac{1}{2}}$$

- Khoảng cách D4

$$D_4(p, q) = |x - s| + |y - t|$$

- Khoảng cách D8

$$D_8(p, q) = \max(|x - s|, |y - t|)$$

05/01/2023

TS. ĐOÀN THỊ HƯƠNG GIANG - EPU

106

... To chapter 2