ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

TRUY VẤN THÔNG TIN THỊ GIÁC

Lab 01

Giảng viên hướng dẫn Thầy Võ Hoài Việt

Thầy Nguyễn Trọng Việt

Thầy Phạm Minh Hoàng

Sinh viên thực hiện Võ Nguyễn Hoàng Kim

21127090

Phụ lục

I. Biểu đồ Histogram

- Tổng quan, histogram là một loại biểu đồ thể hiện tần suất theo dạng cột. Dữ liệu được biểu thị bằng các cột trên biểu đồ có độ cao khác nhau tuỳ thuộc vào tần suất mà dữ liệu xuất hiện.
- Trong lĩnh vực thị giác máy tính, biểu đồ histogram là một công cụ thống kê được sử dụng để biểu diễn phân bố của các giá trị pixel trong một hình ảnh. Nó thể hiện tần suất xuất hiện của các giá trị màu khác nhau trong ảnh. Biểu đồ này có thể chia làm 2 loại:
    - **Histogram của ảnh xám**: Đối với ảnh xám (grayscale), histogram cung cấp thông tin về tần số, cho biết só lượng các pixel ở mỗi mức độ xám từ 0 đến 255.
    - **Histogram của ảnh màu**: Đối với ảnh màu (RGB), thường có ba biểu đồ histogram riêng biệt tương ứng cho ba kênh màu trong ảnh là đỏ, xanh lá, xanh lam. Mỗi biểu đồ thể hiện sự phân bố tần suất của kênh màu tương ứng.
- Trong mã nguồn bài này, ta thực hiện vẽ biểu đồ histogram cho ảnh màu, và chỉ vẽ duy nhất một ảnh histogram. Kết quả của biểu đồ histogram này sẽ cho thấy sự phân bố của tần suất ba kênh màu trong cùng một biểu đồ. Giá trị của nó sẽ được hiển thị bằng các đường thẳng có màu tương ứng.

II. Thực nghiệm

1. Xây dựng mã nguồn

a. Tổng quan mã nguồn

- Mã nguồn được xây dựng theo kiến trúc hướng đối tượng (OOP) với lớp chính là Video. Trong lớp này, các thuộc tính và phương thức bao gồm:
    - Thuộc tính:
        - ***frame***: dùng để lưu trữ khung hình (dưới dạng Mat) của video hoặc webcam
        - ***video_filePath***: chứa đường dẫn đến vị trí lưu trữ video
        - ***video***: là kiểu dữ liệu VideoCapture từ thư viện OpenCV, nó được sử dụng để mở và đọc video, đồng thời hỗ trợ cho việc truy cập các khung hình có trong video hoặc webcam
        - ***usingWebcam***: là biến cờ để xác định việc sử dụng webcam hay không.
    - Phương thức:
        - ***Constructor (Video) và Destructor (~Video)***: Khởi động và giải phóng cho đối tượng Video.
        - ***getFrame***: được sử dụng lấy một khung hình từ đối tượng Video.
        - ***openWebCam***:được sử dụng để mở webcam.
        - ***openVideo***: được sử dụng để mở video theo đường dẫn.
        - ***drawHistogram_RGB***: dùng để vẽ biểu đồ histogram tương ứng cho từng khung hình (frame) được truyền vào.
        - ***showResult***:hiển thị khung hình và biểu đồ histogram tương ứng.
- Chương trình được chia làm 3 file chính là **main.cpp**, **image.cpp** và **function.h**, trong đó
    - **main.cpp**: chứa mã nguồn chính để thực thi chương trình.
    - **image.cpp**: chứa các nội dung xây dựng các phương thức của đối tượng Video
    - **function.h**: chứa các khai báo thư viện cần thiết cũng như định nghĩa đối tượng Video.
- Chương trình được thực thi trên Terminal dưới lệnh sau:

<ten_file_thuc_thi.exe> <0/video_filePath>

Trong đó,

- ten_file_thuc_thi.exe: là file thực thi
- <0/video_filePath>: là dấu hiệu để chương trình nhận biết rằng sẽ thực hiện trên video hay webcam.
  - Nếu truyền vào 0, chương trình sẽ hiểu rằng sử dụng webcam từ máy, việc vẽ biểu đồ cũng sẽ sử dụng trên các frame tương ứng từ webcam
  - Nếu truyền vào ***video_filePath*** (là một đường dẫn đến vị trí của video), chương trình sẽ mở và đọc video tương ứng.

- Ví dụ, để chạy chương trình, ta tiến hành mở terminal, truy cập đến thư mục chứa mã nguồn, và thực hiện gọi lệnh:
  - Nếu ta muốn sử dụng dữ liệu từ webcam, ta sẽ thực thi lệnh sau

*21127090.exe **0***

  - Ngược lại, nếu ta muốn sử dụng video có sẵn với đường dẫn là ***video_filePath***, ta sẽ thực thi lệnh sau

*21127090.exe **video_filePath***

- Để dừng chương trình, người dùng nhấn phím **q**.

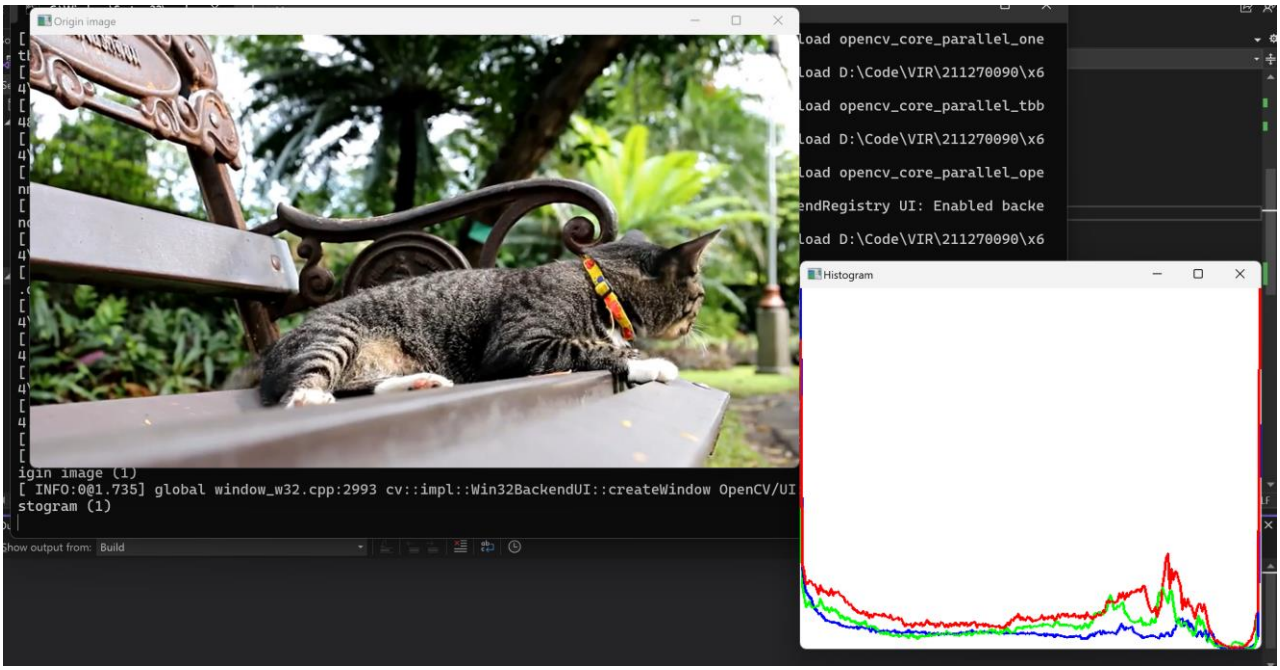
b. Chi tiết mã nguồn

- Trong chương trình này, trọng tâm chính là cách xây dựng biểu đồ histogram cho khung hình (frame) tương ứng, do đó, phưởng thức **drawHistogram_RGB** và hàm **main** sẽ được phân tích chi tiết trong mục này.
- **drawHistogram_RGB()**
  - Tham số truyền vào: phương thức này nhận vào 3 tham số là ***image*** (là khung ảnh cần vẽ biểu đồ histogram), ***hist_w*** và ***hist_h*** đại diện cho chiều rộng và cao của biểu đồ histogram.
  - Cách thức hoạt động:
    - Do mặc định việc phân tích này được thực hiện trên video/ảnh có 3 kênh màu là đỏ, xanh lá, xanh lam (RGB) nên tổng quan rằng ta sẽ phải tính toán giá trị histogram cho từng kênh màu riêng biệt (các kênh màu được tính toán histogram với 256 bin, giới hạn từ 0 đến 255). Việc tính toán này được hỗ trợ bởi hàm **cv::calcHist()**.
    - Để các giá trị này có thể nằm gọn trong biểu đồ histogram với kích thước mong muốn, việc chuẩn hoá các giá trị là điều cần thiết.
    - Cuối cùng, thực hiện vẽ các cột giá trị histogram lên biểu đồ bằng cách vẽ các đường thẳng cho mỗi giá trị histogram của từng kênh màu, các đường này được phân biệt bằng cách sử dụng chính màu sắc tương ứng của từng kênh.
    - Kết quả trả về sẽ là hình ảnh biểu đồ histogram hoàn chỉnh.
- **main()**
  - Tham số truyền vào: hàm này nhận vào 2 tham số truyền từ terminal ***n_args*** (đại diện cho số lượng biến truyền vào) và ***args*** (là tập các biến truyền vào).
    - Trong mã nguồn này, như cấu trúc lệnh đã được đề cập đến ở mục a, số lượng biến hợp lệ của chương trình luôn là 2, với biến đầu tiên luôn là tên chương trình thực thi, và biến còn lại để đánh dấu cho việc sử dụng webcam hay video với đường dẫn được cung cấp
  - Cách thức hoạt động
    - Để chắc chắn rằng các biến được truyền vào hợp lệ, trước hết, ta cần kiểm tra số lượng biến. Nếu như số lượng biến truyền vào khác 2, chương trình sẽ in ra thông báo không hợp lệ và dừng ngay lập tức.
    - Nếu như số lượng biến hợp lệ, việc còn lại là xử lý video để vẽ ra biểu đồ histogram tương ứng cho từng frame.

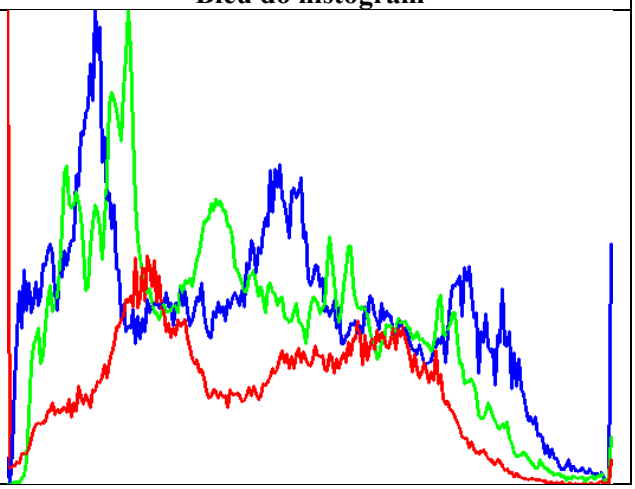
- Thực hiện khởi tạo đối tượng Video trong mã nguồn và truyền vào các tham số cần thiết. Nếu như từ biến truyền vào xác định rằng chương trình sẽ sử dụng webcam để xử lý, nó sẽ gọi hàm để mở webcam từ máy tính lên. Ngược lại, nếu như dữ liệu từ biến truyền vào là một đường dẫn đến video, chương trình sẽ mở video tương ứng lên.
    - Tuy nhiên trong bước này, nếu như không mở được video (do đường dẫn sai, hoặc video không tồn tại hoặc xảy ra lỗi khiến cho video không mở được, …), chương trình sẽ in ra thông báo và thay thế bằng cách mở webcam từ máy tính lên.
- Sau khi đã mở được video/webcam, sử dụng vòng lặp while để có thể duyệt hết toàn bộ video:
    - Tại mỗi lần lặp, chương trình sẽ lấy một frame từ video thông qua phương thức **getFrame()**.
    - Cần kiểm tra xem frame vừa lấy được có phải là một giá trị rỗng không, nếu phải, điều này có nghĩa rằng video đã kết thúc. Khi đó, sẽ thoát khỏi vòng lặp.
    - Nếu không phải là một frame rỗng, chương trình sẽ tiến hành vẽ biểu đồ histogram tương ứng cho frame đó thông qua phương thức **drawHistogram()**.
    - Biểu đồ histogram và frame tương ứng sẽ được hiển thị thông qua phương thức **showResult()**. Mỗi frame được dừng (delay) khoảng 500ms (tức 0.5s) để giúp cho người dùng quan sát dễ hơn.
    - Đồng thời, khởi tạo phím tắt kết thúc vòng lặp (cũng như dừng chương trình) bằng hàm **waitKey()**. Phím tắt được sử dụng để dừng chương trình là **q**.

2. Kết quả đạt được

- Khi thực thi chương trình như các bước đã nêu trên, quá trình vẽ biểu đồ histogram cho từng ảnh được thể hiện bằng hình bên dưới.



a. Webcam

Khung ảnh	Biểu đồ histogram
	

b. Video

Khung ảnh	Biểu đồ histogram