

Trường đại học Khoa Học Tự Nhiên

Thành phố Hồ Chí Minh

XLAV

Document Lab3

Nguyễn Thanh Tùng
19127618

Contents

I. Thông tin sinh viên	1
II. Mô tả các hàm	1
III. Hướng dẫn sử dụng.....	3
IV. Tham khảo:.....	4

I. Thông tin sinh viên

1. MSSV: 19127618
2. Họ tên: Nguyễn Thanh Tùng

II. Mô tả các hàm

Class	Function	Description
Convolution	GetKernel	- Trả về kernel
	SetKernel	- Tạo ra kernel, hàm này được sửa thành kiểu virtual để các class con kế thừa và đa hình hóa.
	DoConvolution	<ul style="list-style-type: none"> - Thực hiện tính tích chập cho các kernel (của Average và Gaussian). - Phương pháp tính tích chập là tính tổng các giá trị của các điểm ảnh quanh điểm trung tâm (điểm cần tính) khi nhân với kernel của nó.
AverageKernel	SetKernel	<ul style="list-style-type: none"> - Cài đặt kernel cho phương pháp average (giống hướng dẫn của thầy trong clip) + Bước 1: ràng buộc giá trị cho chiều rộng và cao của kernel luôn là lẻ và lớn hơn 3. + Bước 2: đưa giá trị vào kernel. Phương pháp trung bình nên giá trị này là $\frac{1}{kích thước kernel}$
GaussianKernel	SetKernel	<ul style="list-style-type: none"> - Bước 1: ràng buộc giá trị cho chiều rộng và cao của kernel luôn là lẻ và lớn hơn 3. - Bước 2: Tính giá trị mỗi ô kernel dựa theo công thức sau: $h(i,j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2+j^2}{2\sigma^2}}$

		<ul style="list-style-type: none"> Bước 3: chuẩn hoá bằng công thức: $k(x) = \frac{k(x)}{\sum_{i=0}^{kernel size-1} k(i)}$
MedianKernel	SetKernel	<ul style="list-style-type: none"> Hàm này không có công dụng vì phương pháp này không cần kernel, sự có mặt của nó để chương trình không gặp lỗi ngoài mong muốn.
	DoConvolution	<ul style="list-style-type: none"> Hàm này hoạt động trên nguyên lý giá trị của ô cần tính là giá trị trung vị của ô đó và các ô lân cận. Vì vậy việc xây dựng hàm này gồm 2 bước: <ul style="list-style-type: none"> + Bước 1: lấy giá trị các ô lân cận và sắp xếp lại từ nhỏ tới lớn (hoặc lớn tới nhỏ). + Bước 2: giá trị cần tìm là giá trị ở giữa dãy trên.
Blur	BlurImage	<ul style="list-style-type: none"> Có vẻ mọi việc cần làm đều đã được thực hiện ở class Convolution và các class con của nó. Hàm này và class này đóng vai trò như một menu. Khả năng khá cao là em đã hiểu sai vai trò của các hàm bên Convolution mong thầy bỏ qua cho sự sai sót này.
EdgeDetector	DetectEdge	<ul style="list-style-type: none"> Hàm đóng vai trò như một menu.
	xGradient_Sobel	<ul style="list-style-type: none"> Tính ma trận gradian của Sobel theo trục x: $G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
	yGradient_Sobel	<ul style="list-style-type: none"> Tính ma trận gradian của Sobel theo trục y: $G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
	xGradient_Prewitt	<ul style="list-style-type: none"> Tính ma trận gradian của Prewitt theo trục x: $G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
	yGradient_Prewitt	<ul style="list-style-type: none"> Tính ma trận gradian của Prewitt theo trục y: $G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

	CalcPrewittAndSobel	<ul style="list-style-type: none"> Bước 1: tại mỗi điểm ảnh tính giá trị Prewitt hoặc Sobel bằng công thức: $Value = xGradian + yGradian$ Bước 2: ràng buộc giá trị trong khoảng [0, 255]. Bước 3: gán giá trị vừa tính vào điểm ảnh đang xét. Cách tính của Sobel và Prewitt là giống nhau chỉ khác nhau về kernel.
	Laplace	<ul style="list-style-type: none"> Tính giá trị tại một điểm theo ma trận: $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$ <ul style="list-style-type: none"> Đây là phương pháp triển khai đơn giản nhất trong 3 phương pháp. Các đường biên của phương pháp này mỏng hơn 2 phương pháp phía trên.

III. Hướng dẫn sử dụng

1. Cấu trúc thư mục:

```

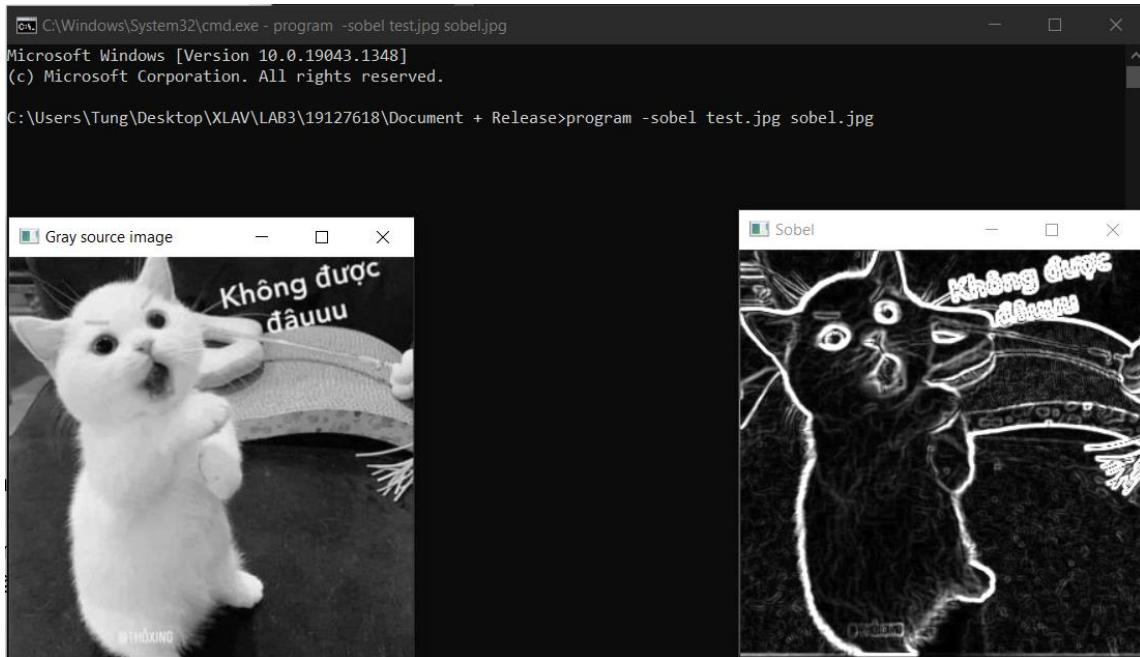
|-19127618:
|  -Document+Release:
|    -19127618_lab3.docx
|    -19127618_lab3.pdf
|    -program.exe
|    -test.jpg
|  -source:
|    -Lib.h
|    -Lib.cpp
|    -Blur.h
|    -Blur.cpp
|    -Convolution.h
|    -Convolution.cpp
|    -EdgeDetector.h
|    -EdgeDetector.cpp
|    -Source.cpp

```

2. Cách sử dụng:

- Khởi chạy bằng cmd.

- Nhập theo hướng dẫn trong đề.
- Nếu nhập đúng sẽ có hình ảnh gốc (dạng grayscale) và ảnh preview sau đó ảnh sau xử lý sẽ được lưu xuống đường dẫn đã nhập. Nếu thất bại sẽ báo lỗi trên console.
- Ví dụ:



3. Lưu ý:

- Phải nhập path chính xác nếu không chương trình sẽ không hoạt động.
- Những giá trị truyền vào bắt buộc phải lớn hơn 0.

IV. Tham khảo:

1. <https://aishack.in/tutorials/sobel-laplacian-edge-detectors/>

—HẾT—

Chúc thầy một ngày vui vẻ!