

Image and Video Processing – Individual Lab 2

Problem

In this assignment, you must implement some geometric image processing using OpenCV in C++. Your program is called by command line arguments and perform the following functions

1. Thông tin cá nhân

Họ và tên: Hồ Ngọc Quyền

MSSV: 19120635

2. Báo cáo

2.1 Thông tin

- Chương trình sử dụng OpenCV trên C++
- Sử dụng tham số dòng lệnh để thực hiện một số tác vụ xử lý ảnh đơn giản

2.2 Chức năng:

- Đọc file ảnh từ <InputFilePath> và thực hiện xử lý ảnh, lưu ảnh ra file <OutputFilePath>

Một số chức năng cụ thể:

- Phóng to, thu nhỏ một ảnh
- Thay đổi kích thước ảnh
- Xoay ảnh xung quanh tâm, không bảo toàn nội dung ảnh
- Xoay ảnh xung quanh tâm, bảo toàn nội dung ảnh
- Lấy đối xứng ảnh theo trục ngang, trục dọc

3. Cài đặt

3.1. Class PixelInterpolate

- Lớp nội suy màu của từng pixel
- Hàm tính giá trị màu của ảnh kết quả từ nội suy màu trong ảnh gốc
- Có 2 phương pháp nội suy được cài đặt, gồm:

Nội suy theo phương pháp song tuyến tính:

- Với mỗi giá trị (tx,ty) thực truyền vào, thực hiện tìm 4 giá trị màu của 4 pixel là 4 góc vuông của hình chữ nhật chứa (tx,ty). Màu của pixel đích là sự đóng góp của 4 giá trị màu xung quanh, được tính theo công thức:

$$F(x,y) = z00*(1-x)*(1-y) + z10*x*(1-y) + z01*(1-x)*y + z11*x*y$$

Với x,y thuộc khoảng [0;1] là vị trí thực của pixel trên trục tọa độ nguyên, bảo đảm nằm trong hình chữ nhật nói trên.

z00, z10, z01, z11 là giá trị màu tại 4 đỉnh hình chữ nhật

Nội suy theo phương pháp láng giềng gần:

- Giá trị (tx,ty) thực được truyền vào được gán bằng giá trị màu của pixel gần nhất trên trục tọa độ nguyên

3.2. Class AffineTransform

- Lớp biểu diễn các phép biến đổi affine cơ bản, gồm:
 - Phép tịnh tiến theo vector (Translate)
 - Phép xoay 1 góc angle (Rotate)
 - Phép tỉ lệ theo hệ số (Scale)

Xây dựng ma trận chuyển đổi của ba phép trên theo ma trận bên dưới

Rotation matrix

| | | |
|-------|--------|---|
| cos α | -sin α | 0 |
| sin α | cos α | 0 |
| 0 | 0 | 1 |

Scaling matrix

| | | |
|----|----|---|
| Sx | 0 | 0 |
| 0 | Sy | 0 |
| 0 | 0 | 1 |

Translation matrix

| | | |
|---|---|----|
| 1 | 0 | dx |
| 0 | 1 | dy |
| 0 | 0 | 1 |

Bên cạnh đó, có hàm con TransformPoint nhằm chuyển đổi 1 điểm (x,y) theo ma trận chuyển đổi trên.

3.3. Class GeometricTransformer

- Lớp giúp thực hiện một số phép biến đổi hình học trên ảnh, theo phép biến đổi affine đã có.

a) Phóng to, thu nhỏ ảnh

- Sử dụng phép affine ngược (Scale) với tỉ lệ phóng to, thu nhỏ là $1/s_x$, $1/s_y$.
- Xây dựng ảnh đích có kích thước tỉ lệ với kích thước ảnh gốc theo hệ số phóng to, thu nhỏ
- Input: Ảnh muốn phóng to, thu nhỏ, hệ số phóng to s_x , s_y , phép nội suy sử dụng
- Output: Ảnh có kích thước lớn gấp s_x lần theo chiều ngang, s_y lần theo chiều cao so với ảnh gốc

b) Thay đổi kích thước ảnh

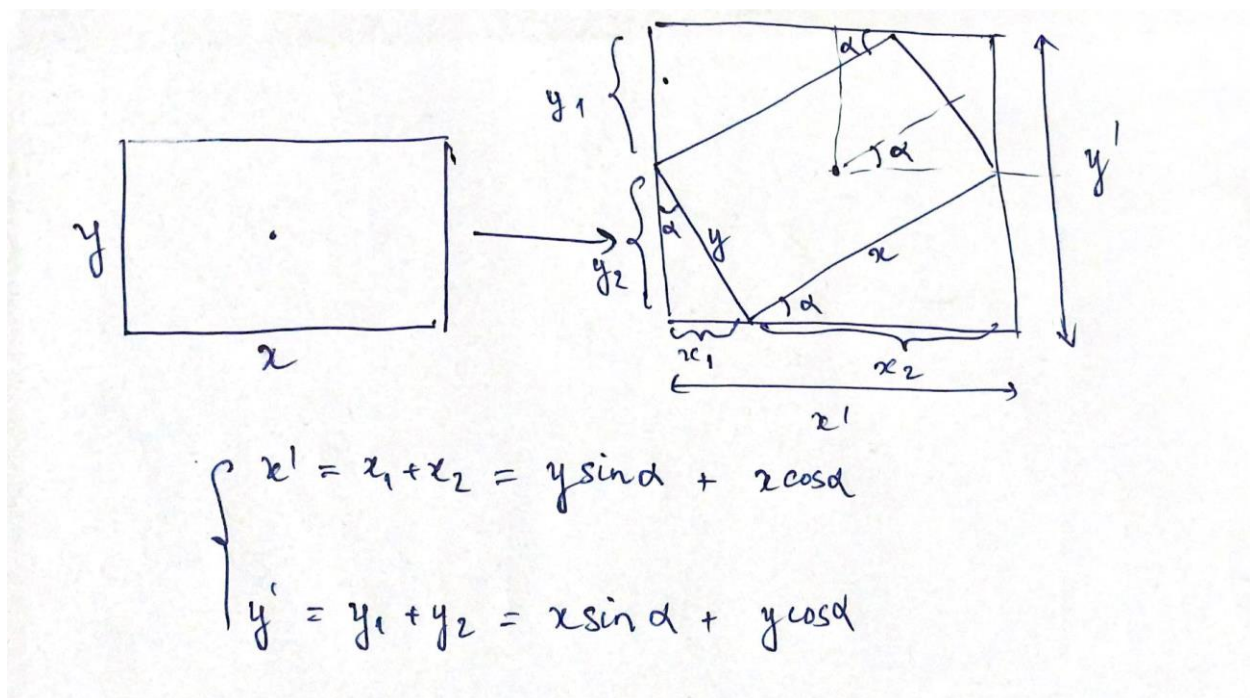
- Tương tự phép phóng to, thu nhỏ ảnh
- Ảnh đích có kích thước là kích thước mới được truyền vào
- Input, Ảnh muốn thay đổi kích thước, chiều rộng mới, chiều cao mới, phép nội suy sử dụng
- Output: Ảnh có kích thước mới

c) Xoay ảnh quanh tâm, không bảo toàn ảnh

- Ảnh đích sau khi thực hiện phép biến đổi sẽ không thay đổi kích thước.
- Việc xoay quanh tâm ban đầu sẽ chuyển về bài toán xoay quanh gốc tọa độ. Em thực hiện việc đó bằng phép tịnh tiến (Translate) theo vector CO (với C là tâm của ảnh gốc có thể tìm được).
- Sau đó áp dụng phép affine ngược (Rotate) với góc quay là $-angle$. Cuối cùng thực hiện phép affine tịnh tiến theo vector OC' (với C' là tâm của ảnh đích có thể tìm được). Thực hiện chuyển đổi với ma trận chuyển đổi tìm được sau khi thực hiện các phép affine trước đó.
- Input: Ảnh muốn xoay, góc xoay, phép nội suy sử dụng
- Output: Ảnh sau khi xoay quanh tâm, không bảo toàn ảnh

d) Xoay ảnh quanh tâm, bảo toàn ảnh

- Cách tìm ma trận chuyển đổi tương tự với phép xoay ảnh quanh tâm, không bảo toàn ảnh được nêu ở trên. Tuy nhiên, ảnh đích sau khi xoay sẽ được bảo toàn toàn bộ ảnh, do đó kích thước sẽ thay đổi. Có thể tính được kích thước mới dựa vào lượng giác, cụ thể như sau:



Với x, y là kích thước của ảnh gốc, x', y' là kích thước của ảnh kết quả

- Input: Ảnh muốn xoay, góc xoay, phép nội suy sử dụng
- Output: Ảnh sau khi xoay quanh tâm, bảo toàn ảnh

e) Đối xứng ảnh theo trục nằm dọc, nằm ngang

- Việc đối xứng ảnh thực chất là phép tịnh tiến với ma trận chuyển có phần khác so với ma trận chuyển đổi đã trình bày ở trên. Cụ thể là:

| | | |
|---|----|----|
| 1 | 0 | dx |
| 0 | -1 | dy |
| 0 | 0 | 1 |

FlipV

| | | |
|----|---|----|
| -1 | 0 | dx |
| 0 | 1 | dy |
| 0 | 0 | 1 |

FlipH

- Với đối xứng theo trục nằm ngang: $(dx; dy) = (\text{height}-1; 0)$
- Với đối xứng theo trục nằm dọc: $(dx; dy) = (0; \text{width}-1)$
- Input: Ảnh muốn đối xứng, hướng đối xứng, phép nội suy sử dụng
- Output: Ảnh sau khi được đối xứng

4. Hướng dẫn sử dụng

- Sử dụng tham số dòng lệnh để thực hiện hàm, cụ thể:
 - Phóng to, thu nhỏ ảnh

OpenCV.exe -zoom <interp> <s_x> <s_y> <InputFilePath> <OutputFilePath>

- Thay đổi kích thước ảnh

OpenCV.exe -resize <interp> <n_w> <n_h> <InputFilePath> <OutputFilePath>

- Xoay ảnh xung quanh tâm, không bảo toàn nội dung ảnh

OpenCV.exe -rotK <interp> <angle> <InputFilePath> <OutputFilePath>

- Xoay ảnh xung quanh tâm, bảo toàn nội dung ảnh

OpenCV.exe -rotP <interp> <angle> <InputFilePath> <OutputFilePath>

- Lấy đối xứng ảnh theo trục ngang

OpenCV.exe -flipH <interp> <InputFilePath> <OutputFilePath>

- Lấy đối xứng ảnh theo trục dọc

OpenCV.exe -flipV <interp> <InputFilePath> <OutputFilePath>

5. Tài liệu tham khảo

- ✓ High quality-Digital Image Processing – Rafael C. Gonzales
- ✓ <https://blogs.sas.com/content/iml/2020/05/18/what-is-bilinear-interpolation.html>
- ✓ <https://stackoverflow.com/questions/57863376/why-i-cannot-flip-an-image-with-opencv-affine-transform>
- ✓ <https://people.gnome.org/~mathieu/libart/libart-affine-transformation-matrices.html#ART-AFFINE-ROTATE>
- ✓ <https://stackoverflow.com/questions/22041699/rotate-an-image-without-cropping-in-opencv-in-c>