

BÀI TẬP THỰC HÀNH 3

Các phép toán cơ bản trên ảnh số

Nội dung

I. Lọc thông thấp trên miền không gian.....	1
II. Lọc thông cao trên miền không gian.....	1
III. Lọc tăng cường.....	1
IV. Gợi ý cách cài đặt.....	2
V. Tài liệu tham khảo.....	4

Qui định chung:

Sinh viên tự viết code thực hiện yêu cầu sau, có thể sử dụng hàm có sẵn để so sánh kết quả bài làm. Cho phép tham khảo nhưng phải chú thích rõ nguồn gốc.

Cú pháp lệnh command line:

<tên chương trình> <đường dẫn đến file ảnh> <chỉ thị lệnh> <tham số kèm theo>

I. Lọc thông thấp trên miền không gian

❖ ~~Lọc trung bình~~

~~**<tên chương trình> <đường dẫn đến file ảnh> filtlowpavg**~~

❖ Lọc trung vị

<tên chương trình> <đường dẫn đến file ảnh> filtmedian

❖ Lọc thông thấp tùy chỉnh bộ lọc

<tên chương trình> <đường dẫn đến file ảnh> filtlowpass <file chứa bộ lọc>

Với file chứa bộ lọc là một file text có định dạng sau:

N
w1 w2 w3
w4 w5 w6
w7 w8 w9

- N là kích thước bộ lọc (ma trận NxN)
- Các w_i là các giá trị nguyên trong bộ lọc

II. Lọc thông cao trên miền không gian

❖ Lọc trung bình

<tên chương trình> <đường dẫn đến file ảnh> filthighpavg

❖ Lọc sử dụng bộ lọc Roberts

<tên chương trình> <đường dẫn đến file ảnh> filtroberts

❖ Lọc sử dụng bộ lọc Prewitt

<tên chương trình> <đường dẫn đến file ảnh> filtprewitt

❖ Lọc sử dụng bộ lọc Sobel

<tên chương trình> <đường dẫn đến file ảnh> fildsobel

III. Lọc tăng cường

- ❖ Lọc sử dụng bộ lọc Hightboost

<tên chương trình> <đường dẫn đến file ảnh> filthhighboost <hệ số>

IV. Gợi ý cách cài đặt

Lưu ý: Gợi ý dưới đây nhằm mục đích dễ hiểu, chương trình sẽ chạy không hiệu quả do đoạn mã nguồn viết như vậy sẽ tốn nhiều thời gian để thực thi. Sinh viên xem để hình dung cách cài đặt và tìm cách tối ưu hóa mã nguồn.

Chương trình dưới đây viết bằng cú pháp OpenCV phiên bản cũ. Sinh viên dựa theo và sử dụng cú pháp OpenCV 2 để viết chương trình.

- ❖ Hàm tính tích chập:

```
//Tính tích chập giữa mặt nạ và ảnh tại vị trí x, y
//Input:
//    img: ảnh gốc
//    x, y: tọa độ của một pixel trên ảnh gốc, nơi sẽ đặt tâm của mặt nạ lên và thực
hiện phép tích chập
//    mask: mặt nạ, để đơn giản, mặc định kích thước 3*3
//Output:
//    giá trị mới của pixel ở vị trí x, y
int tinhTichChap(IplImage* img, int x, int y, int** mask, int size)
{
    float s = 0;
    CvScalar value;
    int d = 0;

    int xtam[9] = {-1, -1, -1, 0, 0, 0, 1, 1, 1};
    int ytam[9] = {-1, 0, 1, -1, 0, 1, -1, 0, 1};

    for (int i = 0 ; i < size ; i++)
        for (int j = 0 ; j < size ; j++)
        {
            //lấy giá trị pixel trong ảnh ở từng vị trí tương ứng với mask[i][j]
            value = cvGet2D(img, x+ xtam[d], y+ ytam[d]);
            d++;

            //nhân với giá trị của mask[i][j] và cộng dồn vào kết quả
            s+= mask[i][j] * value.val[0];
        }
    return (int)s;
}
```

- ❖ Hàm lọc thông thấp tổng quát:

```
//Input:
//    img : ảnh ban đầu
//    mask: bộ lọc, là một mảng 2 chiều
```

```
// size: kích thước bộ lọc
//Output: ảnh sau khi thực hiện phép tích chập giữa img và mask
IplImage* doAverageConv(IplImage* img, int** mask, int size)
{
    //Tạo ảnh kết quả
    IplImage* img_kq = cvCreateImage(cvSize(img->width, img->height),
                                     img->depth, img->nChannels);

    int newGrayValue;
    CvScalar value;

    //free-boundary
    for (int x = 1 ; x <= img->width-2 ; x++)
        for (int y = 1 ; y <= img->height-2; y++)
        {
            //tính tích chập giữa mặt nạ và ảnh tại vị trí x, y
            newGrayValue = tinhTichChap(img,x,y,mask, size);
            value.val[0] = newGrayValue;

            //gán giá trị mới cho ảnh kết quả
            cvSet2D(img_kq,x,y,value);
        }
    return img_kq;
}
```

❖ Hàm lọc thông cao tổng quát

```
//Input:
// img: ảnh ban đầu
// mask1, mask2 : 2 bộ lọc theo phương dọc và ngang, là một mảng 2 chiều
// size: kích thước bộ lọc
//Output: ảnh sau khi thực hiện phép gradient trên ảnh img bằng mask1 và mask2
IplImage* doDerivativeConv(IplImage* img, int** mask1, int** mask2, int size)
{
    //Tạo ảnh kết quả
    IplImage* img_kq = cvCreateImage(cvSize(img->width, img->height),
                                     img->depth, img->nChannels);

    float fGradient1, fGradient2;
    CvScalar value;

    //free-boundary
    for (int x = 1 ; x <= img->width-2 ; x++)
        for (int y = 1 ; y <= img->height-2; y++)
        {
            //tính tích chập giữa mặt nạ và ảnh tại vị trí x, y
            fGradient1 = tinhTichChap(img,x,y,mask1, size);
            fGradient2 = tinhTichChap(img,x,y,mask2, size);
            value.val[0] = (int) sqrt( fGradient1*fGradient1 +
fGradient2*fGradient2 );

            //gán giá trị mới cho ảnh kết quả
            cvSet2D(img_kq,x,y,value);
        }
    return img_kq;
}
```

V. Tài liệu tham khảo

- [1] Lý Quốc Ngọc, *Bài giảng Xử lý ảnh số và video số*, Khoa CNTT, ĐH Khoa Học Tự Nhiên, Tp HCM, ?.
- [2] Nguyễn Quốc Bình, Đặng Đăng Khoa, *Đồ án cao học môn Xử lý ảnh số*, ĐH Khoa Học Tự Nhiên, Tp HCM, 2010.