

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH BUỔI 3

Học Phần : Xử Lý Ảnh

1) Sửa bài tập về nhà buổi 2

Câu hỏi 1 : Xây dựng tính năng biến đổi tuyến tính tổ chức đồ

Sinh viên tham khảo đoạn code sau :

```
QImage MainWindow::SuaChuaHistogram(const QImage &imgin)
{
    int max=0;
    int min = 255;
    QImage image_in(imgin);
    QImage image_out(image_in.width(),image_in.height(),QImage::For

    for (int x=0; x<imgin.width(); x++)
    {
        for (int y=0; y<imgin.height(); y++)
        {
            QRgb color = imgin.pixel(x,y);
            int gray = qGray(color);
            QRgb color_in = qRgb(gray,gray,gray);
            image_in.setPixel(x,y,color_in);
            if (gray > max) max = gray;
            if (gray < min) min = gray;
        }
    }

    for (int x=0; x<imgin.width(); x++)
    {
        for (int y=0; y<imgin.height(); y++)
        {
            int newgray = 0;
            QRgb color = imgin.pixel(x,y);
            int gray = qGray(color);
            newgray = ((gray - min)*255)/(max-min);
            QRgb color_out = qRgb(newgray,newgray,newgray);
            image_out.setPixel(x,y,color_out);
        }
    }

    return image_out;
}

void MainWindow::on_btSuaChua_clicked()
{
    QImage img(ui->LineFileName->text());
    Histogram_GrayImage(img, 200);

    QImage new_img = SuaChuaHistogram(img);
    Histogram_GrayImage(new_img, 200);
}
```

Câu hỏi 2 : Xây dựng tính năng cân bằng tự động tổ chức đồ

Sinh viên tham khảo đoạn code sau :

```

 QImage MainWindow::CanBangHistogram(const QImage &imgin)
 {
     double h[256];
     for (int i=0; i<256; i++)
         h[i]=0;

     QImage image_in(imgin);
     QImage image_out(image_in.width(),image_in.height(),QImage::Format_RGB32);
     for (int x=0; x<imgin.width(); x++)
     {
         for (int y=0; y<imgin.height(); y++)
         {
             QRgb color = imgin.pixel(x,y);
             int gray = qGray(color);
             h[gray]++;
         }
     }
     double hn[256];
     int numpix = imgin.width() * imgin.height();
     for (int i=0; i<256; i++)
         hn[i]= h[i]/(numpix);

     double c[256];
     for (int i=0; i<256; i++)
         c[i]=0;
     for (int i = 0; i < 256; i++)
     {
         for (int j = 0; j < i; j++)
         {
             c[i] = c[i] + hn[j];
         }
     }

     for (int x=0; x<imgin.width(); x++)
     {
         for (int y=0; y<imgin.height(); y++)
         {
             QRgb color = imgin.pixel(x,y);
             int gray = qGray(color);
             int newgray = (int)(c[gray] * 255);
             QRgb color_out = qRgb(newgray,newgray,newgray);
             image_out.setPixel(x,y,color_out);
         }
     }

     return image_out;
 }

 void MainWindow::on_btCanBang_clicked()
 {
     QImage img(ui->LineFileName->text());
     Histogram_GrayImage(img, 200);

     QImage new_img = CanBangHistogram(img);
     Histogram_GrayImage(new_img, 200);
 }

```

2) Nội dung thực hành buổi 3

Nội dung buổi thực hành thứ 3 sẽ xoay quanh việc tính tích chập của ảnh với các bộ lọc hạ thông như bộ lọc trung bình và bộ lọc trung vị. Sau đó xây dựng tính năng thêm nhiễu muối tiêu vào trong ảnh, và sử dụng bộ lọc Median để khử nhiễu muối tiêu cho ảnh.

Sinh viên tiến hành xây dựng giao diện chương trình như sau :

The screenshot shows a Windows application window titled "MainWindow" with standard minimize, maximize, and close buttons. The interface is organized into two main panels, both titled "Xử lý ảnh" (Image Processing).

Xử lý 1 ảnh (Process 1 image):

- Tên file ảnh:** A text input field followed by a "Chọn" (Select) button.
- Ảnh âm bản:** A button labeled "Xử lý ảnh xám" (Process grayscale image).
- Mức sáng:** A text input field with a label "> 0 tăng; < 0 giảm" (Increase/Decrease). Buttons for "Xử lý ảnh xám" and "Xử lý ảnh màu" (Process color image) are present.
- Độ tương phản:** A text input field with a label "> 1 tăng; 0 < c < 1 giảm" (Increase/Decrease). Buttons for "Xử lý ảnh xám" and "Xử lý ảnh màu" are present.
- Chọn ngưỡng:** A text input field with a label "> n trắng; < n đen" (White/Black threshold). Buttons for "Xử lý phân ngưỡng" (Process thresholding), "Giải thuật tuần tự" (Sequential algorithm), and "Giải thuật Otsu" (Otsu algorithm) are present.
- Histograms:** Buttons for "Vẽ histogram ảnh xám" (Draw grayscale histogram), "Vẽ histogram ảnh màu" (Draw color histogram), "Biến đổi tuyến tính histogram" (Linear histogram transformation), and "Cân bằng histogram" (Histogram equalization).
- Red-bordered box:** Contains buttons for "Kích thước mặt nạ" (Mask size), "Lọc trung bình" (Average filter), "Lọc trung vị" (Median filter), "Tỷ Lệ nhiễu (%)" (Noise ratio (%)), "Thêm nhiễu muối tiêu" (Add salt and pepper noise), and "Thêm nhiễu ảnh màu" (Add color noise).
- Filters:** Buttons for "Lọc Gradient", "Lọc Robert", "Lọc Sobel", "Lọc Prewitt", and "Lọc Laplace".

Xử lý 2 ảnh (Process 2 images):

- Chọn ảnh 1:** A text input field followed by a "Chọn ảnh 1" (Select image 1) button.
- Chọn ảnh 2:** A text input field followed by a "Chọn ảnh 2" (Select image 2) button.
- Opacity ảnh 1:** A text input field.
- Opacity ảnh 2:** A text input field.
- Operations:** Buttons for "Cộng 2 ảnh" (Add 2 images) and "Trừ 2 ảnh" (Subtract 2 images).

2.1) Bộ lọc trung bình :

Line Edit “Kích thước mặt nạ” sẽ do người dùng nhập vào. Nếu muốn kích thước mặt nạ là 3x3 thì người dùng nhập số 3 vào. Nếu là 5x5 thì nhập số 5 vào. Sau đó bấm vào nút lọc trung bình hoặc lọc trung vị thì chương trình sẽ đi tính tích chập của ảnh đầu vào với mặt nạ có kích thước đã chọn. Hàm để tính tích chập của ảnh với bộ lọc trung bình được viết như sau :

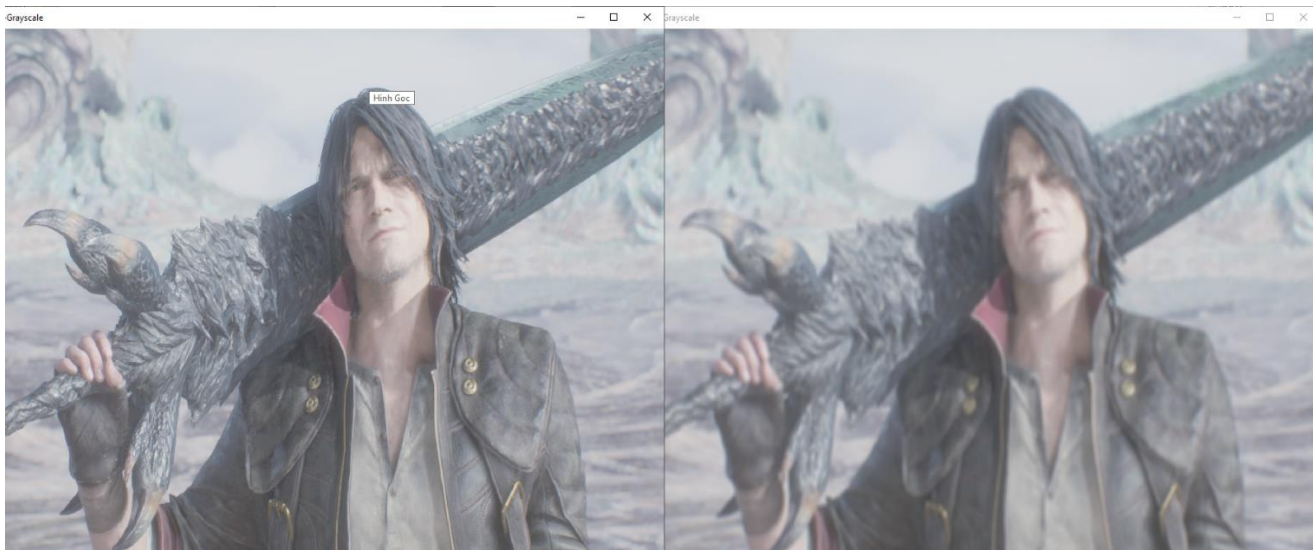
```
void MainWindow::on_btn_meanfilter_clicked()
{
    QImage image_in(ui->LineFileName->text());
    // khai báo cho mặt nạ chập 3x3
    int size = ui->ln_masksize->text().toInt();
    int margin = (size - 1) / 2;
    int cells = size * size;

    QImage image_out(image_in.width() - margin, image_in.height() - margin, QImage::Format_RGB32);
    int sumR, sumG, sumB;
    QColor color;
    for (int x = margin; x < image_in.width() - margin; x++)
        for (int y = margin; y < image_in.height() - margin; y++) {
            sumR = sumG = sumB = 0;
            for (int i = -margin; i <= margin; i++)
                for (int j = -margin; j <= margin; j++) {
                    color = image_in.pixel(x + i, y + j);
                    sumR += color.red();
                    sumG += color.green();
                    sumB += color.blue();
                }
            image_out.setPixel(x, y, qRgb(sumR / cells, sumG / cells, sumB / cells));
        }

    QLabel* label_in = new QLabel();
    label_in->setPixmap(QPixmap::fromImage(image_in.scaled(1024, 768)));
    label_in->setToolTip("Hình Gốc");
    label_in->show();

    QLabel *label = new QLabel();
    label->setPixmap(QPixmap::fromImage(image_out.scaled(1024, 768)));
    label->setToolTip("Hình đã xử lý");
    label->show();
    QString fname = QFileDialog::getSaveFileName();
    image_out.save(fname);
}
```

Lưu ý : Sinh viên cần nắm rõ ý nghĩa của các biến size, margin và cells. Chủ động thay đổi giá trị của kích thước mặt nạ để thấy rõ sự khác biệt.



Ảnh gốc

Ảnh đã chập với bộ lọc trung bình 11x11

2.2) Bộ lọc trung vị :

Sinh viên cài đặt tính năng này như sau :

```
void MainWindow::on_btn_medianfilter_clicked()
{
    QImage image_in(ui->LineFileName->text());
    // khai báo cho mat na chap
    int size = ui->ln_masksize->text().toInt();
    int margin = (size - 1)/2;
    int cells = size* size;

    QImage image_out(image_in.width()-margin,image_in.height()-margin,QImage::Format_RGB32);

    for (int x=margin; x<image_in.width()-margin; x++)
        for (int y=margin; y<image_in.height()-margin; y++) {
            int h[cells];
            int k=0;
            for (int i=-margin; i<=margin; i++)
                for (int j=-margin; j<=margin; j++){
                    QRgb color = image_in.pixel(x+i,y+j);
                    int gray = qGray(color);
                    h[k] = gray;
                    k++;
                }

            qSort(h, h + cells);
            int meanV = h[(cells - 1)/2];
            image_out.setPixel(x,y,qRgb(meanV,meanV,meanV));
        }

    QLabel* label_in = new QLabel();
    label_in->setPixmap(QPixmap::fromImage(image_in.scaled(1024,768)));
    label_in->setToolTip("Hình Goc");
    label_in->show();

    QLabel *label = new QLabel();
    label->setPixmap(QPixmap::fromImage(image_out.scaled(1024,768)));
    label->setToolTip("Hình đã xử lý");
    label->show();
    QString fname = QFileDialog::getSaveFileName();
    image_out.save(fname);
}
```

Lưu ý : Cần xem lại phần lý thuyết về bộ lọc Median.

Kết quả thu được



Ảnh bị nhiễu muối tiêu

Ảnh kết quả đã được khử nhiễu

2.3) Thêm nhiễu muối tiêu vào ảnh

Sinh viên cần khai báo thêm 2 hàm sau vào file mainwindow.h và mainwindow.cpp

```
int MainWindow::Random(int n)
{
    return rand()%n;
}

QImage MainWindow::imNoise_Gray(QImage &image_in, float level)
{
    QImage image_out = image_in;

    int noisePoint = image_in.width() * image_in.height() * level;

    for (int i = 0; i < noisePoint; i++)
    {
        int x = Random(image_in.width());
        int y = Random(image_in.height());
        int a = Random(2);
        if(a == 0)
            image_out.setPixel(x,y,qRgb(0,0,0));
        else
            image_out.setPixel(x,y,qRgb(255,255,255));
    }

    return image_out;
}
```

Sau đó, tại sự kiện click của nút thêm nhiễu, ta thêm đoạn code như sau :

```
void MainWindow::on_btn_addnoise_clicked()
{
    QImage image_in(ui->LineFileName->text());
    QImage image_inGray(image_in.width(), image_in.height(), QImage::Format_RGB32);

    for(int i = 0; i < image_in.width(); i++)
    {
        for(int j = 0; j < image_in.height(); j++)
        {
            QRgb rgb = image_in.pixel(i,j);
            int gray = qGray(rgb);
            QRgb color_in = qRgb(gray,gray,gray);
            image_inGray.setPixel(i,j,color_in);
        }
    }

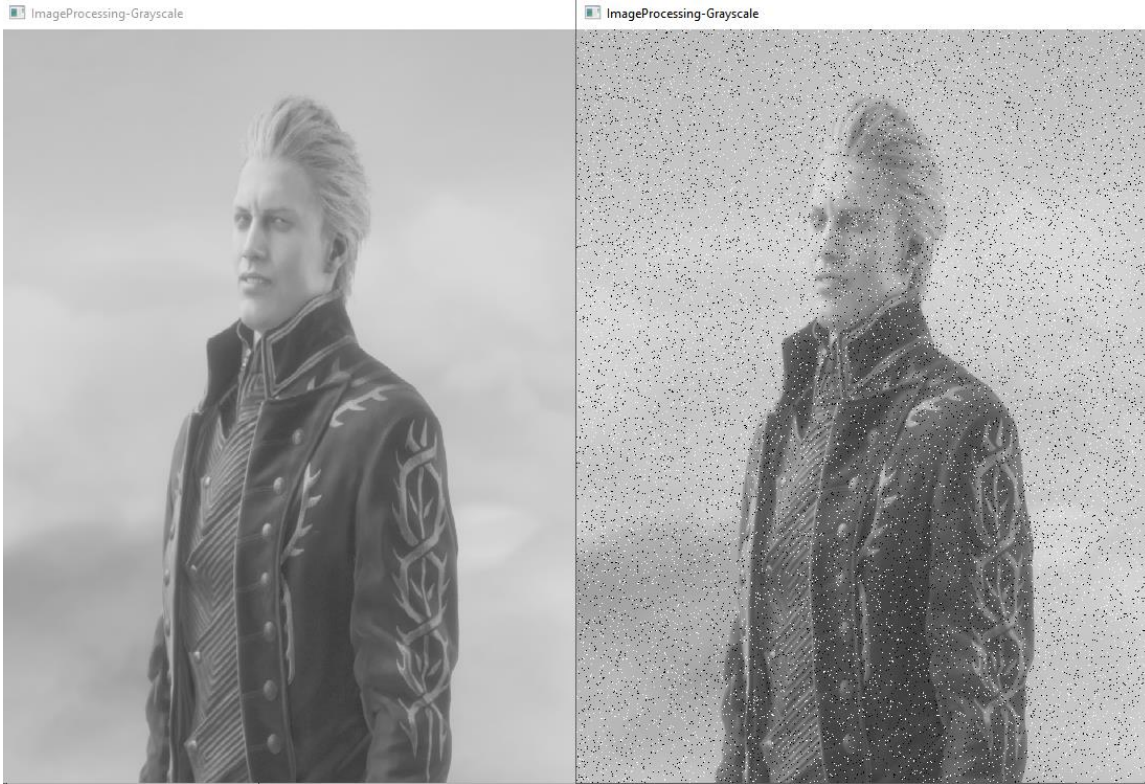
    float percentage = (ui->ln_percentage->text().toInt())/100;

    QImage image_out = imNoise_Gray(image_inGray, percentage);

    QLabel* label_in = new QLabel();
    label_in->setPixmap(QPixmap::fromImage(image_inGray.scaled(1024,768)));
    label_in->setToolTip("Hình Goc");
    label_in->show();

    QLabel *label = new QLabel();
    label->setPixmap(QPixmap::fromImage(image_out.scaled(1024,768)));
    label->setToolTip("Hình đã xử lý");
    label->show();
    QString fname = QFileDialog::getSaveFileName();
    image_out.save(fname);
}
```


Kết quả thu được



Ảnh trước khi thêm nhiễu

Ảnh sau khi thêm nhiễu

3) Bài tập về nhà

Bài 1 : Xây dựng tính năng lọc trung bình đối với ảnh xám

Bài 2 : Xây dựng tính năng thêm nhiễu trên ảnh màu.