

[CTT451] – [Nhập môn Thị giác Máy tính]

Tháng 3/2013

LẬP TRÌNH VỚI OPENCV



MỤC LỤC

1.	Một số hàm thông dụng trong OpenCV	2
2.	Tải và hiển thị ảnh	3
3.	Chuyển ảnh từ hệ màu sang ảnh độ xám	5
4.	Tách các kênh màu của ảnh RGB.....	6
5.	Bài tập.....	7

LẬP TRÌNH VỚI OPENCV

1. Một số hàm thông dụng trong OpenCV

Hàm đọc ảnh từ file:

```
IplImage* cvLoadImage(string strPathName);
```

Hàm sao chép ảnh:

```
void cvCopyImage(IplImage* src, IplImage* dst);  
IplImage* cvCloneImage(const IplImage* image);
```

Hàm hủy đối tượng ảnh:

```
void cvReleaseImage ( IplImage** image);
```

Hàm tạo cửa sổ:

```
cvNamedWindow(char* strWindowName, int flag);  
flag nếu là số lẻ thì hiển thị đúng kích thước ảnh.
```

Hiển thị ảnh trên cửa sổ:

```
cvShowImage (char* strWindowName, IplImage* img);
```

Hàm chuyển đổi hệ màu:

```
void cvCvtColor(IplImage* src, IplImage* dst, int  
code);
```

Hằng số code quy định cách chuyển đổi có dạng:

CV_<Hệ Màu Nguồn>2<Hệ Màu Đích>

VD:

CV_BGR2HSV
CV_RGB2GRAY
CV_HSV2BGR

Tách các kênh màu:

```
cvCvtPixToPlane ( IplImage* src,  
IplImage* img1, IplImage* img2,  
IplImage* img3, IplImage* img4);
```

Trộn các kênh màu:

```
void cvCvtPlaneToPix( const CvArr* src0,  
const CvArr* src1,  
const CvArr* src2,  
const CvArr* src3,  
CvArr* dst);
```

2. Tải và hiển thị ảnh

Tạo lớp `IplImageWrapper` như sau:

```
class IplImageWrapper
{
protected:
    IplImage* _srcImg;
    IplImage* _destImg;
    int _width, _height;
    IplImage *r_plane, *b_plane, *g_plane;
public:
    IplImageWrapper();
    ~IplImageWrapper();

    void LoadImage(char* path);
    void ShowImage(char* windowName, int img = 0);
    void RGB2GRAY1();
    void RGB2GRAY2();
    void PixToPlane();
    void PixToPlane2();

};
```

Lớp gồm có 2 biến `_srcImg` và `_destImg` là ảnh nguồn và ảnh đích, hai biến `_width`, `_height` lưu lại kích thước ảnh và các biến `r_plane`, `b_plane`, `g_plane` chứa từng kênh màu của ảnh.

Hàm constructor và destructor:

```
IplImageWrapper::IplImageWrapper()
{
    _srcImg = NULL;
    _destImg = NULL;
}

IplImageWrapper::~IplImageWrapper()
{
    if(_srcImg != NULL)
        cvReleaseImage(&_srcImg);
    if(_destImg != NULL)
        cvReleaseImage(&_destImg);
}
```

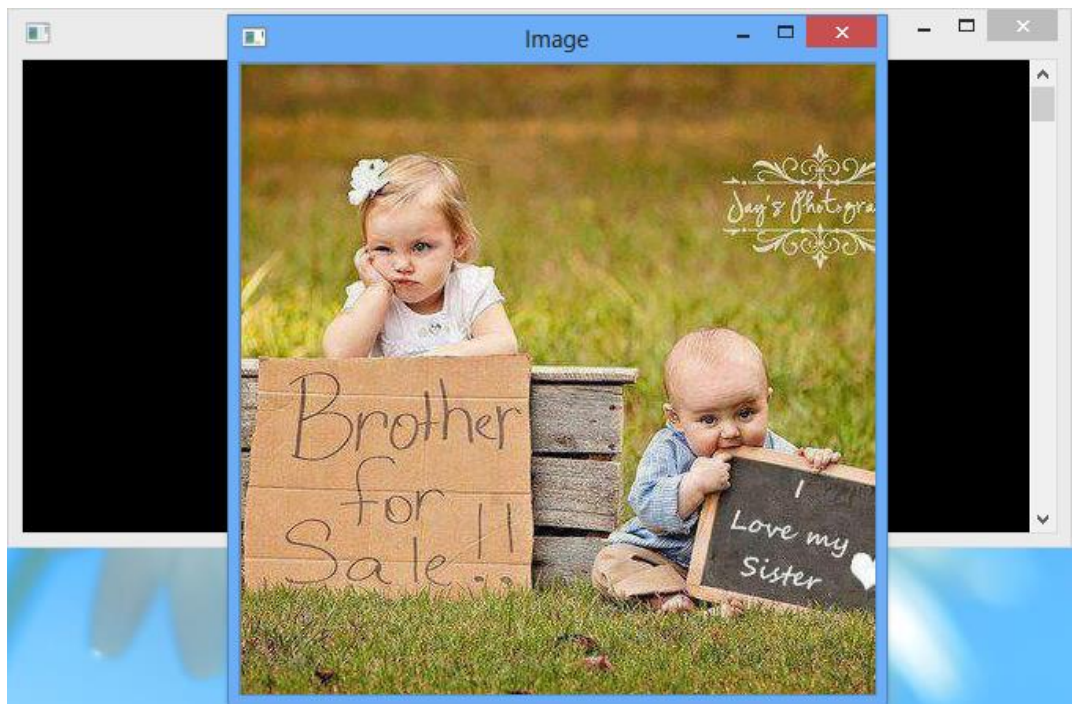
Xây dựng hàm `LoadImage` và `ShowImage` như sau:

```

void IplImageWrapper::LoadImage(char* path)
{
    if(_srcImg != NULL)
        cvReleaseImage(&_srcImg);
    _srcImg = cvLoadImage(path);
    if(_srcImg != NULL)
    {
        _width = _srcImg->width;
        _height = _srcImg->height;
    }
}

```

Hàm có chức năng load ảnh từ đường dẫn path và được trả bởi con trỏ _srcImg.



```

void IplImageWrapper::ShowImage(char* windowName, int
img)
{
    IplImage* image;
    if(img == 0)
        image = _srcImg;
    else if(img = 1)
        image = _destImg;
    else if(img = 2)
        image = b_plane;
}

```

```

else if(img = 3)
    image = g_plane;
else if(img = 4)
    image = r_plane;
if(image != NULL)
{
    cvNamedWindow(windowName,1);
    cvShowImage(windowName,image);
}
}

```

Hàm tạo cửa sổ có tên windowName và hiển thị ảnh lên cửa sổ này.

3. Chuyển ảnh từ hệ màu sang ảnh độ xám

Chuyển ảnh từ hệ màu sang ảnh độ xám sử dụng hàm của OpenCV

```

void IplImageWrapper::RGB2GRAY1()
{
    if(_destImg != NULL)
        cvReleaseImage(&_destImg);
    _destImg = cvCreateImage(cvSize(_width, _height),
IPL_DEPTH_8U, 1);
    cvCvtColor(_srcImg, _destImg, CV_RGB2GRAY);
}

```

Chuyển ảnh từ hệ màu sang ảnh độ xám dựa vào công thức:

$$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$$

```

void IplImageWrapper::RGB2GRAY2()
{
    int step, channels;
    step = _srcImg->widthStep;
    channels = _srcImg->nChannels;

    _destImg = cvCloneImage(_srcImg);
    uchar* dataGray;
    dataGray = (uchar*)_destImg->imageData;
    int i, j;
    for(i = 0; i < _height; i++)
        for (j = 0; j < _width; j++)
        {
            uchar r,g,b,gray_value;
            b = dataGray[i*step+j*channels];
            g = dataGray[i*step+j*channels + 1];

```

```

        r = dataGray[i*step+j*channels + 2];
        gray_value = (int)(r*0.3 + g*0.59
                           + b*0.11);
        dataGray[i*step+j*channels] =
            dataGray[i*step+j*channels + 1] =
            dataGray[i*step+j*channels + 2] =
            gray_value;
    }
}

```



4. Tách các kênh màu của ảnh RGB

Tách các kênh màu sử dụng hàm của OpenCV

```

void IplImageWrapper::PixToPlane()
{
    r_plane = cvCreateImage(cvSize(_width,_height), 8,
1);
    g_plane = cvCreateImage(cvSize(_width,_height), 8,
1);
    b_plane = cvCreateImage(cvSize(_width,_height), 8,
1);
    cvCvtPixToPlane(_srcImg, b_plane, g_plane, r_plane,
NULL);
}

```

Tách các kênh màu (truy cập dữ liệu ảnh)

```

void IplImageWrapper::PixToPlane2()

```



```

{
    int i, j, step, channels;
    step = _srcImg->widthStep;
    channels = _srcImg->nChannels;

    b_plane = cvCloneImage(_srcImg);
    g_plane = cvCloneImage(_srcImg);
    r_plane = cvCloneImage(_srcImg);

    uchar *dataB, *dataG, *dataR;
    dataB = (uchar *)b_plane->imageData;
    dataG = (uchar *)g_plane->imageData;
    dataR = (uchar *)r_plane->imageData;

    for(i = 0; i < _height; i++)
        for (j = 0; j < _width; j++)
        {
            dataB[i*step+j*channels+1] = 0;
            dataB[i*step+j*channels+2] = 0;

            dataG[i*step+j*channels] = 0;
            dataG[i*step+j*channels+2] = 0;

            dataR[i*step+j*channels] = 0;
            dataR[i*step+j*channels+1] = 0;
        }
}

```



5. Bài tập

- Tạo ảnh âm bản (negative image) của ảnh mức xám
- Tạo ảnh âm bản của ảnh màu

- Thực hiện cộng thêm một hằng số vào giá trị tại mỗi điểm ảnh trên ảnh mức xám.
- Thực hiện với 3 hằng số, mỗi hằng số được dùng để cộng vào giá trị của kênh màu tương ứng tại mỗi điểm trên ảnh màu.