

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

**Tháng 4/2013**

# **CANNY EDGE DETECTION**



Bộ môn TGMT và KH Rô-bốt  
Khoa Công nghệ thông tin  
ĐH Khoa học tự nhiên TP HCM



## MỤC LỤC

MỤC LỤC .....	1
1 Thuật toán Phát hiện cạnh Canny .....	3
2 Cài đặt thuật toán phát hiện cạnh bằng OpenCV .....	4
3 Bài tập.....	6
Tài liệu tham khảo .....	7

# 1 Thuật toán Phát hiện cạnh Canny

## Bước 1: Giảm nhiễu

Thông thường để giảm nhiễu sử dụng các bộ lọc làm mờ. Có thể sử dụng bộ lọc Gaussian để tích chập với ảnh:

$$\frac{1}{159} * \begin{array}{|c|c|c|c|c|} \hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

## Bước 2: Tính độ lớn và góc của Gradient

Tính đạo hàm  $D_x(x, y)$  và  $D_y(x, y)$  theo chiều x và y của ảnh. Một số bộ lọc như: Roberts, Prewitt, Sobel.

Bộ lọc Sobel 3x3:

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

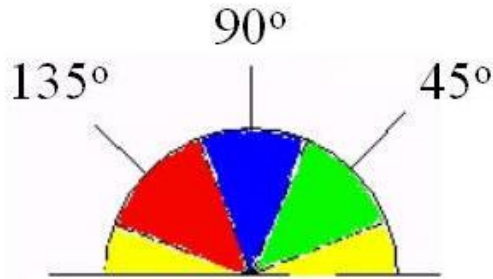
Độ lớn Gradient:

$$D = \sqrt{D_x^2(x, y) + D_y^2(x, y)}$$

Góc Gradient:

$$\theta = \arctan\left(\frac{D_x(x, y)}{D_y(x, y)}\right)$$

Tính  $\theta'$  bằng cách làm tròn từ góc  $\theta$  vào một trong bốn hướng:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ .



### Bước 3: chặn không cực đại (Non-Maximum Supression)

Bước này chỉ giữ lại những pixel thuộc cạnh mà có độ lớn gradient lớn nhất

Xem xét 3 pixel trong vùng 3 x 3 xung quanh pixel (x,y):

- Nếu  $\theta(x, y) = 0^\circ$  thì (x+1, y), (x, y) và (x-1, y) được xem xét.
- Nếu  $\theta(x, y) = 90^\circ$  thì (x, y+1), (x, y) và (x, y-1).
- Nếu  $\theta(x, y) = 45^\circ$  thì (x+1, y+1), (x, y) và (x-1, y-1).
- Nếu  $\theta(x, y) = 135^\circ$  thì (x-1, y+1), (x, y) và (x+1, y-1).

Nếu pixel (x, y) có gradient lớn nhất của 3 pixel xem xét thì pixel đó là cạnh.

### Bước 4: Ngưỡng Hysteresis (Hysteresis Thresholding)

Hysteresis sử dụng 2 ngưỡng, ngưỡng  $t_{\text{high}}$  và  $t_{\text{low}}$ . Pixel mà có độ lớn gradient  $D < t_{\text{low}}$  thì được loại ngay lập tức. Những pixel  $t_{\text{low}} < D < t_{\text{high}}$  được giữ lại nếu là một cạnh liên tục với những pixel có độ lớn gradient  $D > t_{\text{high}}$ .

## 2 Cài đặt thuật toán phát hiện cạnh bằng OpenCV

Xây dựng class Canny như sau:

```
class Canny
{
protected:
    IplImage *_srcImg, *_destImg;
    int _width, _height;
    int _lowThreshold, _maxThreshold;

public:
    Canny(int low, int max)
    {
        _srcImg = NULL;
    }
}
```

```

        _destImg = NULL;
        _lowThreshold = low;
        _maxThreshold = max;
    }

    int CannyEdgeDetection(char *path);
};

```

Hàm CannyEdgedetection:

```

int Canny::CannyEdgeDetection(char *path)
{
    _srcImg = cvLoadImage(path);
    if(_srcImg == NULL)
        return 0;
    cvNamedWindow("Image");
    cvShowImage("Image", _srcImg);

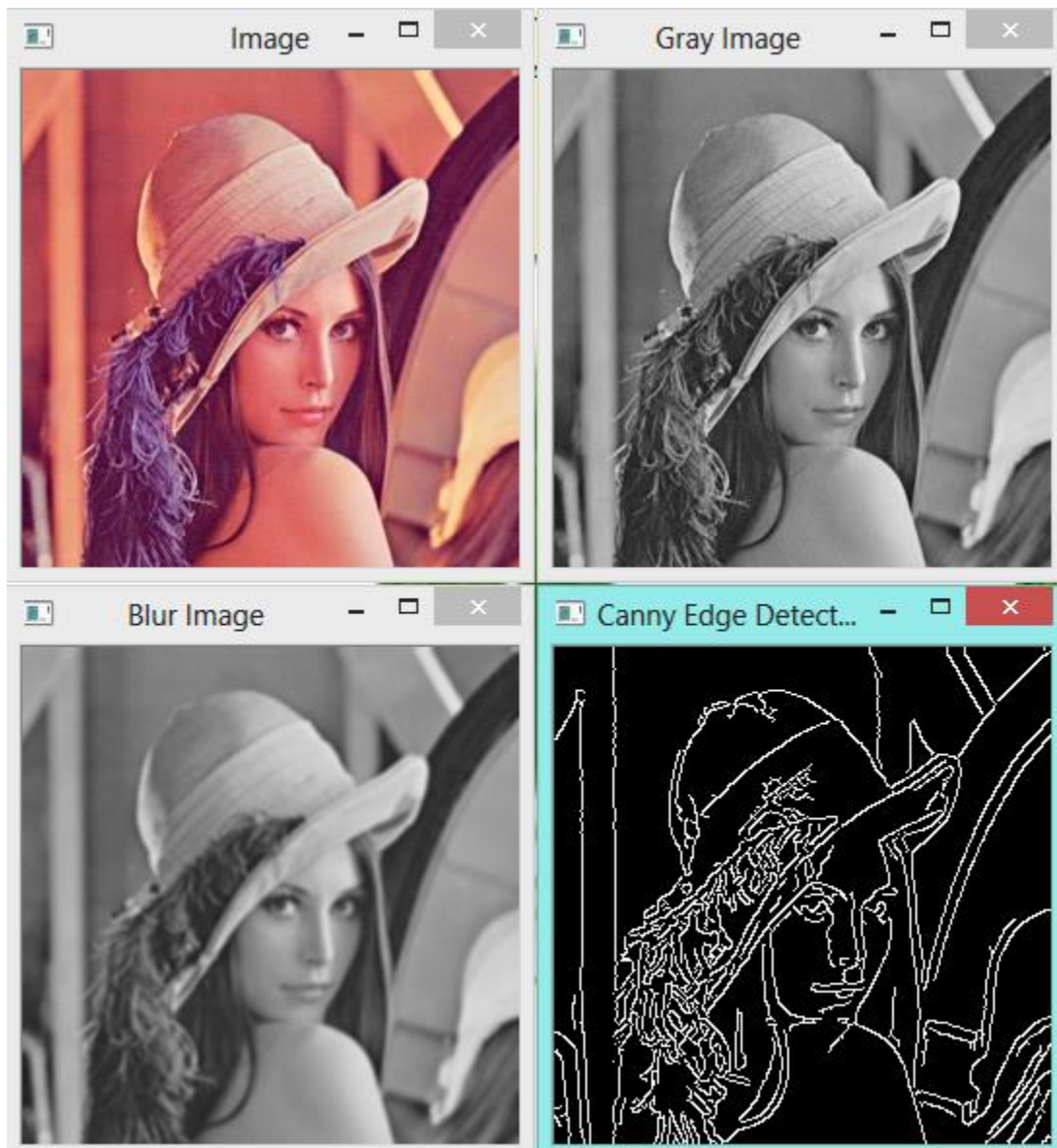
    IplImage *grayImg, *blurImg;
    grayImg = cvCreateImage(cvGetSize(_srcImg), IPL_DEPTH_8U, 1);
    cvCvtColor(_srcImg, grayImg, CV_BGR2GRAY);
    cvNamedWindow("Gray Image");
    cvShowImage("Gray Image", grayImg);

    blurImg = cvCreateImage(cvGetSize(_srcImg), IPL_DEPTH_8U, 1);
    cvSmooth(grayImg, blurImg, CV_GAUSSIAN, 5, 5);
    cvNamedWindow("Blur Image");
    cvShowImage("Blur Image", blurImg);

    _destImg = cvCreateImage(cvGetSize(_srcImg), IPL_DEPTH_8U, 1);
    cvCanny(blurImg, _destImg, _lowThreshold, _maxThreshold, 3);

    cvNamedWindow("Canny Edge Detection");
    cvShowImage("Canny Edge Detection", _destImg);
    return 1;
}

```



### 3 Bài tập

Viết chương trình phát hiện biên cạnh dựa vào thuật toán Canny như trong mục 1.

## **Tài liệu tham khảo**

- [1] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.