OpenCV-C++ canny() & scharr() - images

Ubuntu 22.04 LTS - Intel Core i5 CPU 2,20GHz x 4 - 16GiB RAM

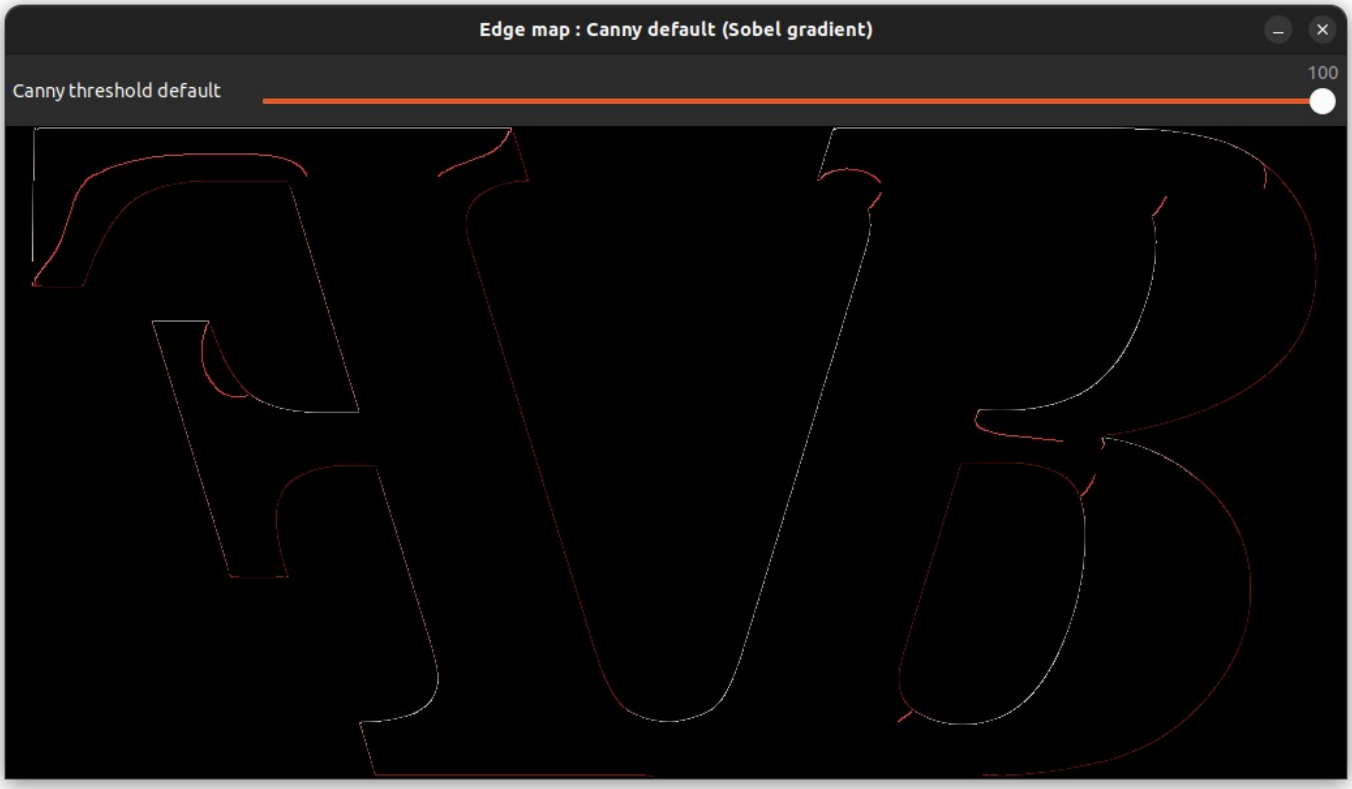
```
//edge.cpp
#include "opencv2/core/utility.hpp"
#include "opencv2/imgproc.hpp"
#include "opencv2/imgcodecs.hpp"
#include "opencv2/highgui.hpp"

#include<iostream>

int edgeThresh = 1;
int edgeThreshScharr=1;

cv::Mat image, gray, blurImage, edge1, edge2, cedge;

const char* window_name1 = "Edge map : Canny default (Sobel gradient)";
const char* window_name2 = "Edge map : Canny with custom gradient (Scharr)";

// define a trackbar callback
static void onTrackbar(int, void*)
{
    cv::blur(gray, blurImage, cv::Size(3,3));

    // Run the edge detector on grayscale
    cv::Canny(blurImage, edge1, edgeThresh, edgeThresh*3, 3);
    cedge = cv::Scalar::all(0);

    image.copyTo(cedge, edge1);
    cv::imshow(window_name1, cedge);

    // Canny detector with scharr
    cv::Mat dx,dy;
    cv::Scharr(blurImage,dx,CV_16S,1,0);
    cv::Scharr(blurImage,dy,CV_16S,0,1);
    cv::Canny( dx,dy, edge2, edgeThreshScharr, edgeThreshScharr*3 );

    // Using Canny's output as a mask, we display our result
    cedge = cv::Scalar::all(0);
    image.copyTo(cedge, edge2);
    cv::imshow(window_name2, cedge);
}

static void help(const char** argv)
{
    std::cout << '\n' << "This sample demonstrates Canny edge detection" << '\n'
             << "Call: " << '\n' << argv[0]
             << "image_name -- Default is logo.png" << "\n\n";
}

const char* keys = {"{help h||}{@image |logo.png|input image name}"};
```

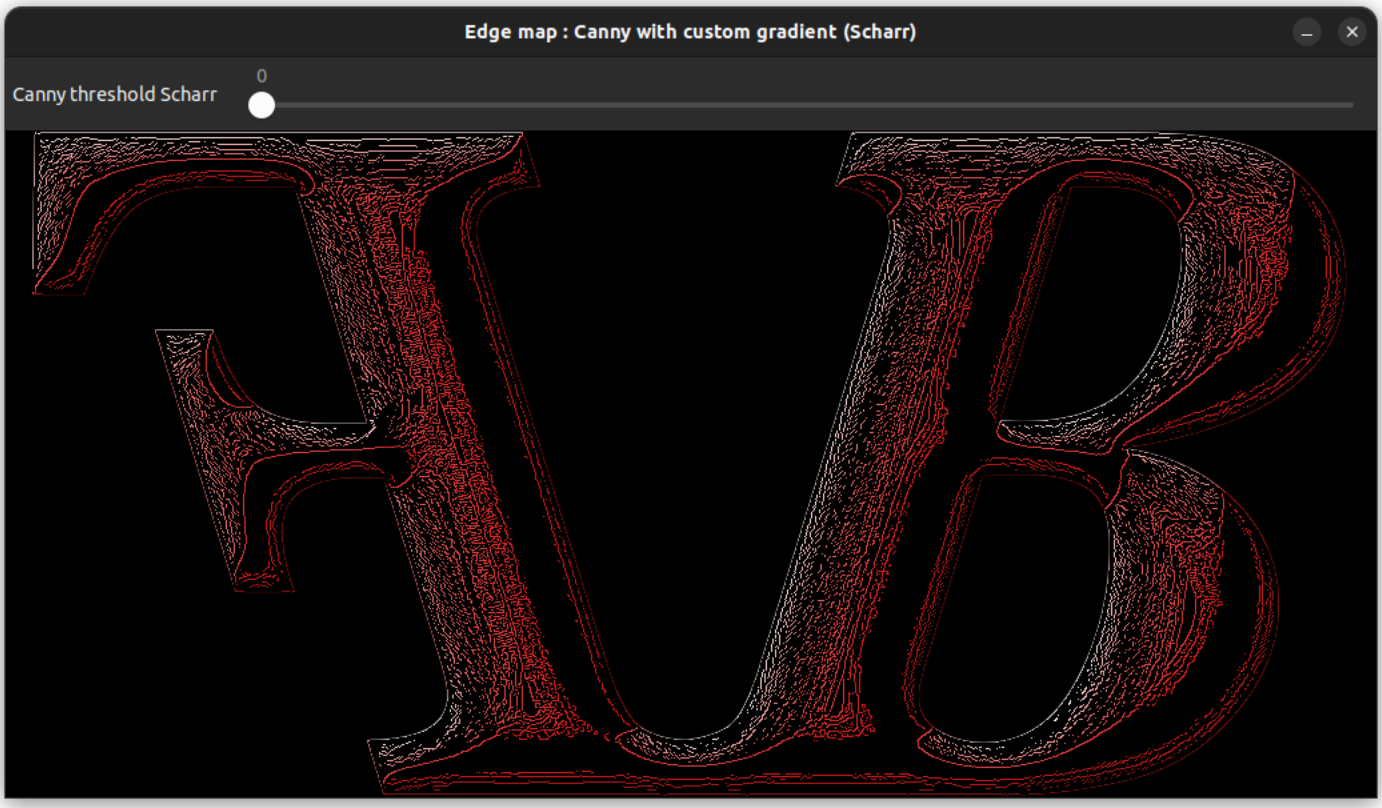
```
int main( int argc, const char** argv )
{
    help(argv);
    cv::CommandLineParser parser(argc, argv, keys);
    std::string filename = parser.get<std::string>(0);

    image = cv::imread(cv::samples::findFile(filename), cv::IMREAD_COLOR);
    if(image.empty())
    {
        std::cout << "Cannot read image file " << filename.c_str() << '\n';
        help(argv);
        return -1;
    }

    cedge.create(image.size(), image.type());
    cv::cvtColor(image, gray, cv::COLOR_BGR2GRAY);

    // Create a window
    cv::namedWindow(window_name1, 1);
    cv::namedWindow(window_name2, 1);

    // create a toolbar
    cv::createTrackbar("Canny threshold default", window_name1, \
                                                    &edgeThresh, 100, onTrackbar);
    cv::createTrackbar("Canny threshold Scharr", window_name2, \
                                                  &edgeThreshScharr, 400, onTrackbar);

    // Show the image
    onTrackbar(0, 0);

    // Wait for a key stroke; the same function arranges events processing
    cv::waitKey(0);

    //destroy all opened windows
    cv::detroyAllWindows();

    return 0;
}
```

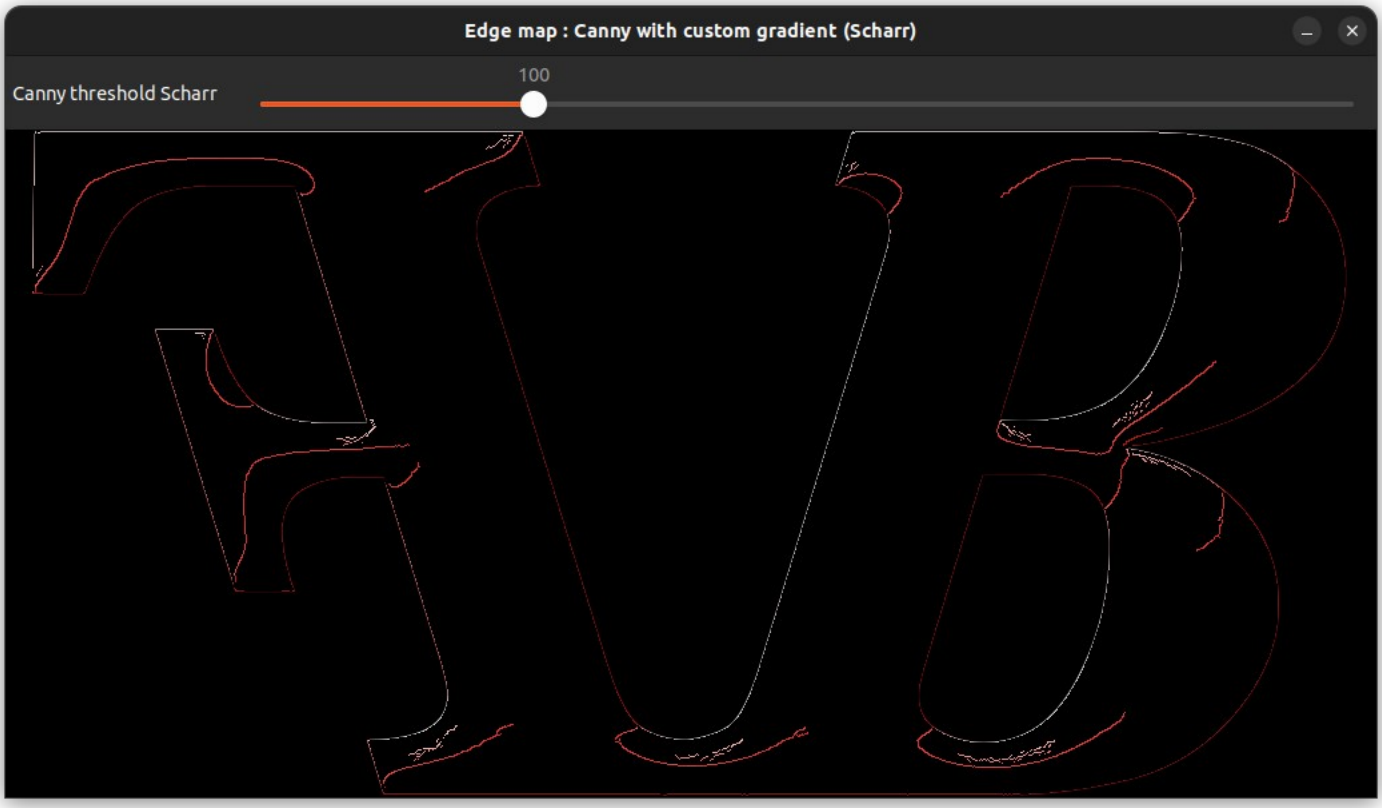
```
$ cd opencv/my_edge/ ⏎
$ ls ⏎
logo.png    edge.cpp
$ g++ -ggdb edge.cpp -o edge `pkg-config --cflags --libs opencv4` ⏎
$ ls ⏎
edge      logo.png     edge.cpp
$ ./edge ⏎
```

logo.png


Edge map : Canny default (Sobel gradient)
Canny threshold default
0


Edge map : Canny default (Sobel gradient)
Canny threshold default
100





Francesco Bonfissuto Munich, August 02.2022