Modification of Canny Edge Detector

DIP Course Project – C3

Digital Image Processing (1001) | 2020.5.26

* Andrew · Jingyu LIU · 1630003037
* Laurence · Zichen LUO · 1630003040
* White · Zimian FENG · 1730026020

# Introduction

## Project Abstraction

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. And there are three principle of the detector.

1. Low error rate edge detection: The detection algorithm should accurately find as many edges as possible in the image to minimize missed detections and false detections.
2. Optimal positioning: The detected edge point should be accurately positioned at the center of the edge.
3. Edges in the image should be marked only once, and the image noise should not produce pseudo edges.

And usually, canny edge detector has 4 steps.

1. Gaussian blur. Edge detection is very sensitive to noise, you need to use Gaussian blur to remove noise

2. Find the intensity gradient in the image.

3. Non-maximum suppression. Smooth the image and obtain the first derivate of the image in the horizontal and vertical directions. Find the edge gradient and direction of each pixel.

4.Hysteresis Thresholding. Two thresholds will be used to decide which is true edge.

*Processed by original canny edge detector (threshold of 70~140)*

Disadvantage: Around Lena eyes and nose, we can see many unnecessary edges.

Our group project (CP 3.) aims to search for an improved version of Canny Edge Detector, with better edge-detecting result.

* The Gaussian filter, that is normally used in Canny Edge Detector to filter out noise and smooth the image, will be replaced with anisotropic/directional filter.

The original 4 directions used in Canny Edge Detector for deciding the high thresholder edges will be added to 8 directions.

We will compare the revised Canny Edge Detector with the original one, for the discussion of which is the better/more optimized version. We will also take into consideration of the optimized setting of threshold values and other parameters that may affect the result of the edge detection.

# Implementation of Anisotropic Filter

## Smoothing / Blurring Process

The second step in the handling of Canny Edge Detection, as introduced in the introduction part above, is the “Noise Reduction”. Smoothing, also called blurring, is a simple and frequently used image processing operation. Although there are many application cases of image blurring, but here in Canny Edge Detection and many other edge detection handling cases, blurring is used to reduce noise. As stated by the OpenCV official tutorial, the normal method is:

Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter. We have already seen this in previous chapters.

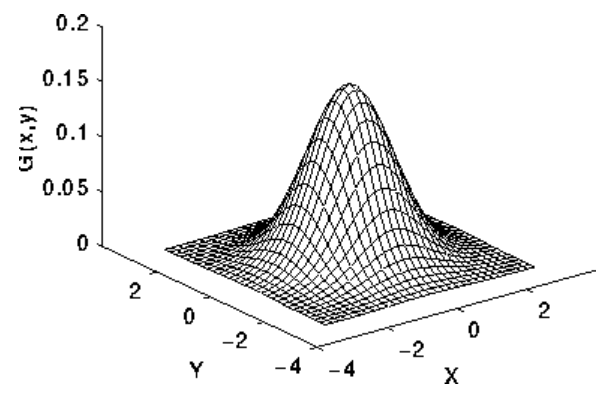
To remove noise in the source image, OpenCV recommends using a Gaussian filter. Gaussian filtering is done by convolving each point in the input array with a Gaussian kernel and then summing them all to produce the output array.

**Gaussian filtering** is done by convolving each point in the input array with a Gaussian kernel and then summing them all to produce the output array.

A 2D Gaussian can be represented as

where is the mean (the peak) and represents the standard deviation (per each of the variables and ).

The 2-D Gaussian distribution with mean and is shown as:



One of the examples of a Gaussian kernel provided by OpenCV is:

K = \dfrac{1}{159}\begin{bmatrix}
          2 & 4 & 5 & 4 & 2 \\
          4 & 9 & 12 & 9 & 4 \\
          5 & 12 & 15 & 12 & 5 \\
          4 & 9 & 12 & 9 & 4 \\
          2 & 4 & 5 & 4 & 2
                  \end{bmatrix}

However, in the example source code from OpenCV, we can see that it uses **Homogeneous Blur**, also called **Normalized Block Filter**.

A box blur (also known as a box linear filter) is a spatial domain linear filter in which each pixel in the resulting image has a value equal to the average value of its neighboring pixels in the input image. It is a form of low-pass ("blurring") filter. In this technique, each pixel value is calculated as the average value of the neighborhood of the pixel defined by the kernel.

Kernels used in the homogeneous blur is called normalized box filter. In the case of Canny Edge Detection, the kernel size is preferably to be .

## Fallback of Isotropic/Linear Filtering

Box linear filter is optimal for reducing random noise in spatial domain (image space). However, it is the worst filter for frequency domain, with little ability to separate one band of frequencies from another. While Gaussian filter has better performance in frequency domain. The Gaussian smoothing operator is a 2-D convolution operator that is used to `blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump.

Gaussian filter, also, runs at a slower speed comparing to the box linear filter. And if the box filter run convolutionally, it can produce result similar/close to that of Gaussian filter.

The 2-D Gaussian filter, based on the 1-D/2-D Gaussian distribution, is an isotropic filter because the distribution is isotropic.

The reason is that Gaussian filters are isotropic in the sense that all surrounding pixels affect the center one in a similar fashion regardless their intensity variations. Hence, edges and noise are treated in the same way, which yields noise reduction as well as edge blurring. And edge blurring during the process of an edge detection is obviously not helpful to the result.

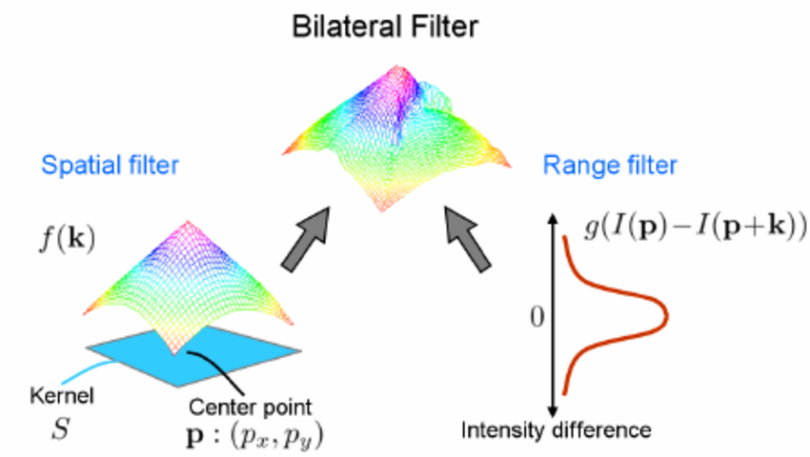
## Anisotropic Filtering

To remedy the problem of traditional Gaussian filtering, anisotropic filtering has drawn more and more attention, and several categories of anisotropic filtering have been proposed. The famous methods are, for example, Bilateral Filtering, Wavelet Transformation, the Mean-median method, and Anisotropic Diffusion. We will focus on studying the Bilateral Filtering and Anisotropic Diffusion in out report.

Bilateral Filtering

A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

Anisotropic Diffusion



Combined domain and range filtering will be denoted as bilateral filtering. It replaces the pixel value at x with an average of similar and nearby pixel values. In smooth regions, pixel values in a small neighborhood are similar to each other, and the bilateral filter acts essentially as a standard domain filter, averaging away the small, weakly correlated differences between pixel values caused by noise.

Anisotropic Diffusion

In image processing and computer vision, anisotropic diffusion, also called Perona–Malik diffusion, is a technique aiming at reducing image noise without removing significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image. Anisotropic diffusion resembles the process that creates a scale space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. Each of the resulting images in this family are given as a convolution between the image and a 2D isotropic Gaussian filter, where the width of the filter increases with the parameter. This diffusion process is a linear and space-invariant transformation of the original image. Anisotropic diffusion is a generalization of this diffusion process: it produces a family of parameterized images, but each resulting image is a combination between the original image and a filter that depends on the local content of the original image. As a consequence, anisotropic diffusion is a non-linear and space-variant transformation of the original image.

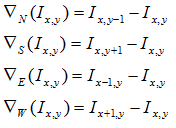
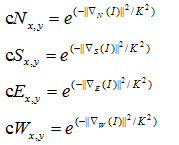
The anisotropic diffusion filtering algorithm has excellent smoothing performance for medical images, but the normal diffusion filtering algorithm will blur the edges and details. While anisotropic diffusion filtering can preserve the edges and at the same time smoothing the image.

Given a gray scale image :

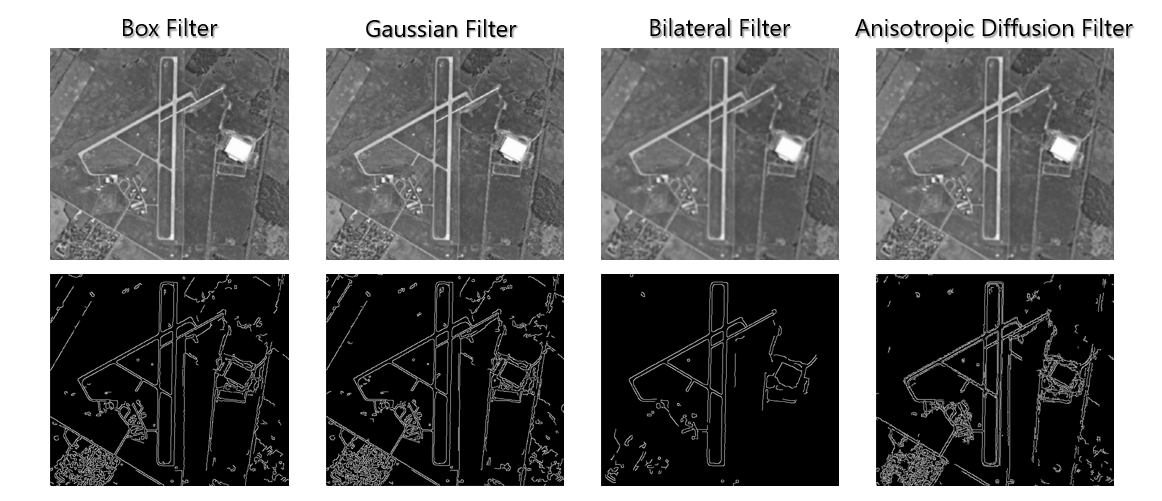
where is the Laplacian operator, and is the gradient operator. is the diffusion-coefficient and is the coefficient of heat conduction.

The basic idea of anisotropic diffusion filtering is to see an image as a thermal field, where every pixel in it is a heat flow. The flow will depend on the relation between pixels. And if the neighboring pixels are edge pixels, the “flow” tends to be weak and slow. Otherwise, the diffusion-coefficient changes while flows, and area will become smoother. In this way, while preserving the edge in the image, the noise is suppressed.

Where is the iteration times.

Effects in Practice



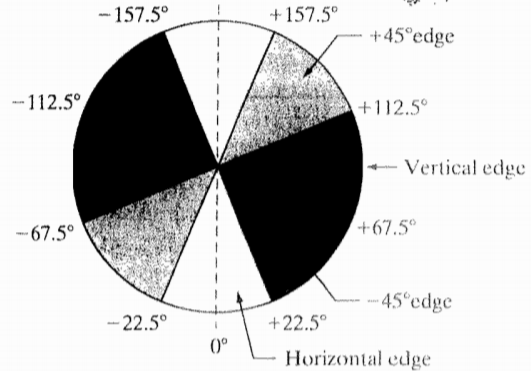
As for the parameter in Canny Edge Detection, we set the low threshold to be 70 and high threshold to be 140, and the size of aperture to be 3.

We use the marion\_airport as input image.

As result, Box Filter and Gaussian Filter not only marked out the runways as the edge, but also marked the fields and roads as pseudo-edge. It’s obvious that by edge detection, our goal is to clearly mark out the airport runways, which is the main object, and isolated them from other things and the background. The Box Filter and Gaussian Filter fails to do so, but Bilateral Filter and Anisotropic Diffusion Filter have improved performance and better result.

# Directions in Canny Edge Detector

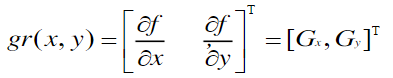
As we can see from the following figure that it has 4 directions in order to thin these edges using *nonmaximal suppression*. These 4 directions are, 0° (horizontal), -45°, 90° (vertical) and +45°, respectively.



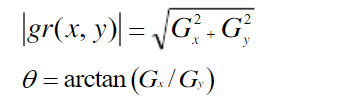
Directions in improved Canny Edge Detector

The directions in the original detector are powerful to thin those edges. However, to thin them more precisely, we have divided those gradients into 8 directions.

In order to realize this function, we used a new algorithm where it is extended from traditional Sobel operators. The definition equation of gradient is below:

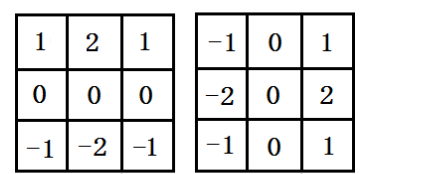


And the equations of amplitude and directions are below:

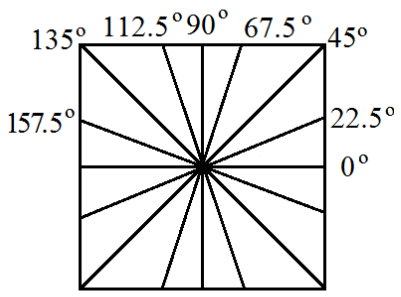


We can see that if Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. [1]

By using vertical (right) and horizontal (left) template to do the convolution, it can not only get the image gradient well, but also suppress the noise to some extent.



In this case, the middle angle directions of each interval are extracted. The 8 directions are shown below the figure.



Similar with 4 directions, the notation of each direction is not merely a single angle. Instead they are a range of intervals and we get the middle direction of each interval. The horizontal direction ranges from -11.25° to 11.25°, the new inserted direction of 22.5° ranges from 11.25°to 33.75°, by parity of reasoning.

The contribution of directional gradients depends on the distance between each gradient and the central point and the angle of gradient and the central point. The nearer the gradient is, or the smaller the angle is, the greater the contribution will be.

The results of images using 4 directions (left) and 8 directions are below (right) (both using threshold of 70~140):

Where we can see that, compared with the left figure, the right one can remove more pseudo edges more effectively.

Thresholding

As we know, setting a proper threshold value is more than important. If too low, there will still have some pseudo edges remaining. If too high, the image will lose some key details.

In this project, we set up the threshold with low threshold of 70 and high threshold of 140. From the result, we can see that the main role is much more stood out, also owing to the better effect of the smoothed image. For instance, there are few edges on the person’s hat as the hat looks much smoother after filtering. The main edges we can see are the whole face, hair, and its background.



Summary

In general, our modification has two main advantages comparing with original canny edge detector.

1. Gaussian blur sometimes filters out effective edges, causing images to lose detail. Anisotropic diffusion filter can reduce noise while preserving detailed features of the image.
2. Decide the high thresholder edges by 4 directions will keep a part of the pseudo edges, but as our example shows, 8 directions detector can reduce more pseudo edges.

References

[1] Green, B. (2002). Canny edge detection tutorial. *Retrieved: March*, *6*, 2005.