Lecture Node for Edge Detection

mijialong
mijlong@mail2.sysu.edu.cn

Abstract

This learning report is used to record my learning in edge detection. In this report, I will record some common algorithms used in edge dection. For example, The **Sobel operator***, sometimes called the Sobel–Feldman operator or Sobel filter,* **Scharr operator***,* **Canny edge detector** *and* **Deriche edge detector**.

In addition to the algorithm principle and specific steps, I will also use the code corresponding to each algorithm to process some test image to see the effect of each edge detection algorithm.

1. Introduction

Now computer vision developing rapidly, image processing technology has been applied to all aspects of our lives. As one of the more basic directions in computer vision, edge detection has a history of decades of development.

1.1. Edge detection

Edge detection is a basic problem in computer vision and image processing, which is aim at identifying points that brightness in image changes sharply, or more formally. This problem includes a varity of mathematical methods and most of methods are suitable for using programming language to write code and hand it over to the computer for implementation.

1.2. Edge

The sharp changes between adjacent points can be used to shown important events and changes. These changes of image brightness are likely to correspond to:[1]

- discontinuities in depth
- discontinuities in surface orientation
- changes in material properties
- variations in scene illumination

The edge can be classified as viewpoint-dependent or viewpoint-independent, which extracted from a 2D image of a 3D scene. A viewpoint-dependent edge may change with the different viewpoints and offen shows the grometry of the scene. AAnd the properties of 3D objects usually can be reflected by a viewpoint-independent edge.

1.3. Effect

Applying edge detection to a image process may result a set of connected curves that the indicate the boundaries so that can typically reduce the amount of the data in the image that need to be processed. And it can filter out much more information, which is considered as less as relevant, while save the structural charateristics that is important of the image.

Edge detection has been applied into image processing, computer vision and machine vision in the academic field, which is in the fileds of feature detection and feature extraction. It is also a basic step in image analeysis, image pattern recognition and image processing.

2. Edge detector

There are lots of methods for edge detection, and most of them can be classified into 2 catagories: a) Based on search; b) based on zero crossing.

The search based method usually calculates the measures of edge strength with a first-order derivative expression such as gradient magnitude firstly, and then uses the calculated estimate of the local orientation of the edge, usually the gradient direction, to search for the local directional maximum of the gradient magnitude. [3]

The zero-crossing based methods search for zero-crossing in a second-order derivative expression to find edges which calculated from the image. The common zero-crossing expressions are the zero-crossing of the Laplacian operator or the zero-crossing of the nonlinear differential expression.

A smoothing stage is almost applied as a pre-processing step for edge detection, which is usually Gaussian smoothing, to reduce image noise.

2.1. Image gradient

Define the gradient of an image:

$$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$$

The ∇f is a vector, and it points the direction of the most rapid increase in intensity. And the gradient direction is given:

$$\begin{aligned}\theta &= \tan^{-1}(\nabla f) \\ &= \tan^{-1}(\frac{\partial f}{\partial x}/\frac{\partial f}{\partial y})\end{aligned}$$

So the edge strength is given by the gradient magnitude:

$$\| f \| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$$

It is not hard to find that the more noise in image, the harder in finding a effective image detector if we do not preprocess the image. So smoothing is a common solution tu reduce the noise of image, and Gaussian smoothing which is mentioned above is one of the most methods to smoothing the image for noise reduction.

There is a set of images(Figure 1-4) with different arguments of Gaussian smoothing that processed by Canny detector.

Figure 1. Origin picture with some noise.

2.2. Sobel operator

Sobel operator is a simple edge detector. It is relatively cheap in terms of calculations because of using small, sparable integer-valued filter to convolving image in the horizontal and vertical directions. But the gradient approximation it produces is relatively rough for high-frequency changes in the image.

2.2.1 Formulation

This operator uses two 3*3 kernel which are convolved with the original image to compute the approximations[4]. Define A as the source image and G_x and G_y are two images

Figure 2. Using Canny detector without Guassian smooting.

Figure 3. Using Canny detector with Guassian smooting (3×3).

that at each point contain the vertical and horizontal derivative approximations.

$$\begin{aligned}G_x &= \begin{bmatrix}+1 & 0 & -1 \\ +2 & 0 & -1 \\ +2 & 0 & -2\end{bmatrix} * A \\ &= \begin{bmatrix}1 \\ 2 \\ 1\end{bmatrix} * (\begin{bmatrix}+1 & 0 & -1\end{bmatrix} * A) \\ G_y &= \begin{bmatrix}+1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1\end{bmatrix} * A \\ &= \begin{bmatrix}+1 \\ 0 \\ -1\end{bmatrix} * (\begin{bmatrix}1 & 2 & 1\end{bmatrix} * A)\end{aligned}$$

Based on the definity of image gradient, the gradient pro-

Figure 4. Using Canny detector with Guassian smooting (9×9).

duced by sobel operator is

$$G = \sqrt{G_x^2 + G_y^2}$$

And the direction of gradient is

$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

The pseudocode implementation is:

Algorithm 1 sobel operator

Require: A: 2D image array
Ensure: $image$

```
function SOBEL(A)
    Gx = [-1 0 1; -2 0 2; -1 0 1]
    Gy = [-1 -2 -1; 0 0 0; 1 2 1]

    rows = size(A, 1)
    columns = size(A, 2)
    mag = zeros(A)

    for each i ∈ [1, rows − 2] do
        for each j ∈ [1, columns − 2 do
            S1 = sum(sum(Gx .* A (i:i+2, j:j+2)))
            S2 = sum(sum(Gy .* A (i:i+2, j:j+2)))

            mag(i + 1, j + 1) = sqrt(S1 .^ 2 + S2 .^ 2)
        end for
    end for

    threshold = 70 % threshold ∈ [0, 255]
    outputImage = max(mag, threshold)
    outputImage(outputImage == round(threshold)) = 0

    return outputImage
end function
```

2.2.2 Extension to other demision

Sobel operator can be divided into 2 operations[5]:

- Smoothing perpendicular to the derivative direction with a filter: $h(-1) = 1, h(0) = 2, h(1) = 1$
- Simple central difference in the derivative direction: $h'(-1) = 1, h'(0) = 0, h'(1) = -1$

From 3D to 4D, suppose that $x, y, z, t \in \{-1, 0, 1\}$:
3D:

$$h'_x(x, y, z) = h'(x)h(y)h(z)$$
$$h'_y(x, y, z) = h(x)h'(y)h(z)$$
$$h'_z(x, y, z) = h(x)h(y)h'(z)$$

4D:

$$h'_x(x, y, z) = h'(x)h(y)h(z)h(t)$$
$$h'_y(x, y, z) = h(x)h'(y)h(z)h(t)$$
$$h'_z(x, y, z) = h(x)h(y)h'(z)h(t)$$
$$h'_t(x, y, z) = h(x)h(y)h(z)h'(t)$$

An example in z-direction in 3D space:

$$h'_z(:, :, -1) = \begin{bmatrix} +1 & +2 & +1 \\ +2 & +4 & +2 \\ +1 & +2 & +1 \end{bmatrix}$$
$$h'_z(:, :, 0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
$$h'_z(:, :, 1) = \begin{bmatrix} -1 & -2 & -1 \\ -2 & -4 & -2 \\ -1 & -2 & -1 \end{bmatrix}$$

2.2.3 Optimized operators

Using central differences, sobel operator does not have perfect rotational symmetry. So Scharr try to improve this operator.[6, 7] This is the most frequently used:

$$G_x = \begin{bmatrix} +3 & 0 & -3 \\ +10 & 0 & -10 \\ +3 & 0 & -3 \end{bmatrix}$$
$$G_y = \begin{bmatrix} +3 & +10 & +3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

Kayyali operator was generated from Sobel operator, which is a perfect rotational symmetry based convolution

filter 3×3.[8]

Kroon designed a differential filter by using the numerical method optimization.[9] Kroon use a quasi Newton optimizer from called FMINLBFGS. And he found the radio between the kernel values p_1 and p_2 is 3.5887, which is close to the radio of Scharrs kernel $3\frac{1}{3}$. The operator is:

$$\begin{bmatrix} +17 & 0 & -17 \\ +61 & 0 & -61 \\ +17 & 0 & -17 \end{bmatrix} \quad \begin{bmatrix} +17 & +61 & +17 \\ 0 & 0 & 0 \\ -17 & -61 & -17 \end{bmatrix}$$

These matrice are equal to that of a truncated Gaussian derivative of $\sigma = 0.548$, which toprevent large aliasing errors with area sampling of the kernel.

Instead of using a 3×3 derivative kernel, using a 5×5 kernel can optimized parameter σ to $\sigma = 0.6769$ and give the minimal angle erroris:

$$\begin{bmatrix} 0.0007 & 0.0053 & 0 & -0.0053 & -0.0007 \\ 0.0108 & 0.0863 & 0 & -0.0863 & -0.0108 \\ 0.0270 & 0.2150 & 0 & -0.2150 & -0.0270 \\ 0.0108 & 0.0863 & 0 & -0.0863 & -0.0053 \\ 0.0007 & 0.0053 & 0 & -0.0108 & -0.0007 \end{bmatrix}$$

$$\begin{bmatrix} 0.0007 & 0.0108 & 0.0270 & 0.0108 & 0.0007 \\ 0.0053 & 0.0863 & 0.2150 & 0.0863 & 0.0053 \\ 0 & 0 & 0 & 0 & 0 \\ -0.0053 & -0.0863 & -0.2150 & -0.0863 & -0.0053 \\ -0.0007 & -0.0108 & -0.0270 & -0.0108 & -0.0007 \end{bmatrix}$$

Figure 5-6 show examples processed with the Sobel operator and the Scharr operator.

2.3. Canny operator

Canny edge operator is a widely used edge detection method, which uses the a technique that finds the function which optimizes a given functional - calculus of variations.[10] Compared with Sobel operator, Canny edge detector can better meet the following requirements in edge detection:

- Low error rate in edge detection (catch as many edges as possible accurately).
- The edge point which detected can be located on the center of the edge accurately.
- Ignore or reduce image noise as much as possible in edge detection to preventing create error edge.

There are 5 steps in Canny edge detection.

1. Noise reduction. Usually the Gaussian filter is chose to finish it.
2. Get the intensiy gradient of the image.

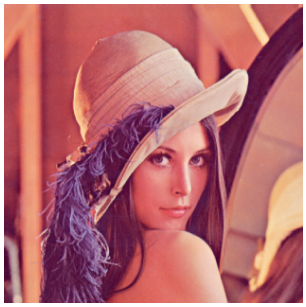

Figure 5. Origin picture without any process.

Figure 6. Origin picture processed by Sobel detector at X direction.

Figure 7. Origin picture processed by Sobel detector at Y direction.

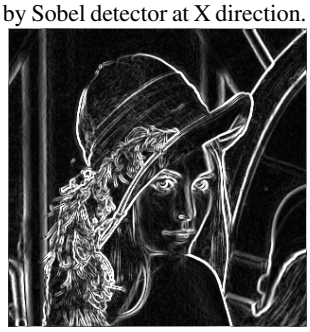

Figure 8. Origin picture processed by Sobel detector at X and Y direction.

3. Using non-maximum supression to get rid of spurious response.
4. Using double threshold to determin potential edges.
5. Edge tracking by hysteresis.

2.3.1 Noise reduction

In image detection, image noise can easily affects the result as we talked above. So a smoothing filter is needed in deed image pre-processing usually, and Gaussian filter is a good choise in most scenarios. The Gaussian filter kernel is convolved with the image can sightly reduce the effects of obvious image noise. The equation (size = $(2k+1) \times (2k+1)$ and suppose that $1 \leq i, j \leq (2k+1)$) is:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{[i\,(k+1)]^2 + [j\,(k+1)]^2}{2\sigma^2}\right)$$

And a 5×5 size Guassian filter is better for most cases (but we most use 3×3 because it is easily) although it also vary depending on specific situations. The is the example:

$$B = \frac{1}{159} \begin{bmatrix} 1 & 4 & 5 & 4 & 2 \\ 2 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 2 & 9 & 12 & 9 & 4 \\ 1 & 4 & 5 & 4 & 2 \end{bmatrix} \times A$$

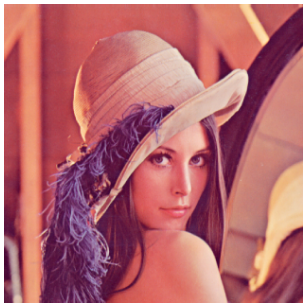

Figure 9. Origin picture without any process.

Figure 10. Origin picture processed by Scharr detector at X direction.

Figure 11. Origin picture processed by Scharr detector at Y direction.

Figure 12. Origin picture processed by Scharr detector at X and Y direction.

2.3.2 Get the intensity gradient

Canny algorithm needs filter to detect horizontal, vertical and diagonal edges (There are much more style of edges in real scenes) because an edge may has a variety of directions. So using edge detection operator such as Sobel operator or Scharr operator can get the value for first derivative in horizontal (suppose $G_x = \frac{\partial f}{\partial x}$) and vertical (suppose $G_y = \frac{\partial f}{\partial x}$) direction. The gradient and direction are:

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\Theta = \arctan\left(\frac{G_y}{G_x}\right)$$

2.3.3 Non-maximum suppression

Non-maximum suppression is applied to find the edge point which have the sharpest change in locale area. Operations will be done in each pixel in the image:

1. Compare the edge intensity of the current pixel with the edge intensity of the pixel in the positive and negative gradient directions.
2. Preserve the value if the edge intensity of the current pixel is the largest in the mask compared to another pixels with the same direction. Otherwise, suppressed the value.

Suppose that x' and x'' are the neighbors of pixel x along normal direction to an edge.[13] And $|\nabla G|(x, y)$ is the gradient at pixel (x, y), Non-maximum suppression can be written as:

$$M(x,y) + \begin{cases} |\nabla G|(x,y) & ,|\nabla G|(x,y) > |\nabla G|(x',y') \\ & \&|\nabla G|(x,y) > |\nabla G|(x'',y'') \\ 0 & ,otherwise \end{cases}$$

The pseudo code is

Algorithm 2 Non-maximum suppression

Require: G: the gradient at each pixel in the image
Require: (x', y'), (x'', y''): the neighbors of pixel (x, y)
Ensure: I_N: result image

```
for each pixel (x, y) do
    if G(x, y) > G(x', y') and G(x, y) > G(x'', y'') then
        I_N(x, y) = G(x, y)
    else
        I_N(x, y) = 0
    end if
end for
```

2.3.4 Double threshold

A more accurae representation of real edges can be shown by using non-maximum suppression. But some pixels may remain which caused by noise or color variation. In Sobel operator, we use a threshold to distinguish whether the pixel is in edge or not. It can be easliy used but likely ignore some potential edges.

So Canny use double threshold[10], which can find more potential details of edge and also increase the complexity of calculating. Suppose $p_1 < p_2$ are 2 values of threshold, they divides gradients we get in prvious steps into 3 intervals. Suppose G_{xy} as the gredient at pixel(x, y), the 3 interval can be given as:

$$\text{point } (x,y) = \begin{cases} \text{not a edge}, & G_{xy} < p_1 \\ \text{weak edge}, & p_1 \le G_{xy} < p_2 \\ \text{strong edge}, & p_2 \le G_{xy} \end{cases}$$

The “weak edge” need further processing, and it would be done in the next step.

2.3.5 Track by hystersis

Weak edge pixels can either be extracted from the true edge, or the noise/color variations. So it is necessary to remove

the weak edges that caused by the latter reasons to approach a more accurate result.

Pixel(x, y) in the image has 8 neighbors and is a weak edge. We consider this pixel is a true edge if there is at least one neighbor is a strong edge. Otherwise, it is not a true edge. It can be written as

$$\text{point } (x, y) = \begin{cases} \text{not a edge,} & \text{none of neighbors} \\ & \text{was a strong edge} \\ \text{true edge,} & \text{at least one neighbors} \\ & \text{is a string edge} \end{cases}$$

2.3.6 Development

In original Canny edge detector, non-maximum suppression only apples at 4 angle, which are 0°, 45°, 90° and 135° that define 8 direction.[10] The 8 directions are:

- The point will be considered to be in the **east and west** directions if the rounded gradient angle is 0°.
- The point will be considered to be in the **north east and south west** directions if the rounded gradient angle is 45°.
- The point will be considered to be in the **north and south** directions if the rounded gradient angle is 90°.
- The point will be considered to be in the **sourth east and north west** directions if the rounded gradient angle is 135°.

It is definitly that use this simplified method can get edges in more directions and effective in image processing. But someone proposed to use interpolation to imporve Canny detector at this step and here are the reasons:

a) Edge gradient directions in the natural image are not necessarily along the 8 directions. In order to find the gradient direction that best matches its location on a pixel, the pixel value on both sides of must be interploated;

b) Also because that the pixels in the actual digital image are discrete two-dimensional matrices, the point on both sides of the gradient directions at the true center position, or be called as sub-pixels, do not necessarily exist. The non-existent point and the gradient value of this point can only be obtained by interpolating the points on both side of it.

There are 2 set of pictures reflect the result about Canny edge detection and compression between non-maximum supression without interpolation and non-maximum suppresion with interpolation.

By comparing the pictures generated at each step, here are some differeces:

Figure 13. orgin picture

Figure 14. Step 1: Process with Gaussian filter.

Figure 15. Step 2: Process with Sobel detector.

Figure 16. Step 3: Non-maximum suppression without interpolation.

Figure 17. Step 3: Non-maximum suppression with interpolation.

Figure 18. Step 4-5: Hysteresisi thresholding without interpolation at step 3.

Figure 19. Step 4-5: Hysteresisi thresholding with interpolation at step 3.

Figure 20. orgin picture

Figure 21. Step 1: Process with Gaussian filter.

Figure 22. Step 2: Process with Sobel detector.

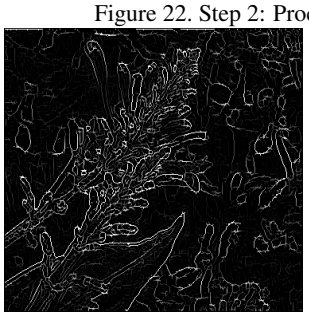

Figure 23. Step 3: Non-maximum suppression without interpolation.

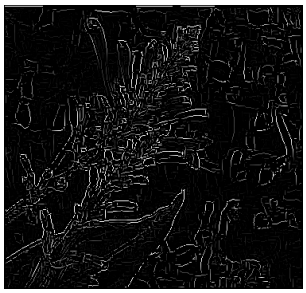

Figure 24. Step 3: Non-maximum suppression with interpolation.

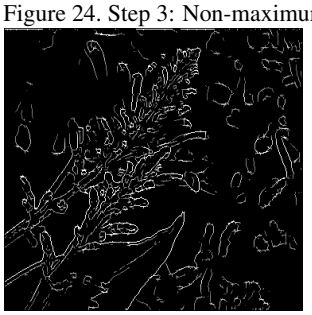

Figure 25. Step 4-5: Hysteresisi thresholding without interpolation at step 3.

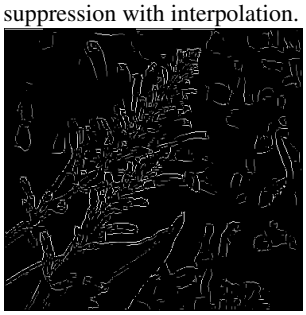

Figure 26. Step 4-5: Hysteresisi thresholding with interpolation at step 3.

1. Edge pixels filtered out by the interpolation method are less than edge processed without interpolation, so image processed with interpolation looks not very continuous in some place, which also make the loss of image details.
2. However, non-maximum suppression without interpolation is more likely to get an edge with a width greater than 1 pixel in terms of edge width. So in case of high-precision single-point response, non-maximum suppression with interpolation may have a better result.

2.3.7 Deriche edge detector

Deriche edge detector is a edge detection operetor that based on Canny edge detector developed by Rachid Deriche in 1987 (Canny edge detector was published in 1986). There are 4 steps in the algorithm, some of which is similar as Canny detector:

1. Smoothing. This step using IIR filter instead of Guassian filter.
2. Calculation of magnitude and gradient direction
3. Non-maximum suppression
4. Double threshold
5. Hystersis threshold with double thresholds

The difference between Canny edge detector and Deriche edge detector are in step 1 and step 2. IIR (**I**nfinite **I**mpulse **R**esponse) filter is used to replace Gussian filter. Deriche propose 2 filters[12] to optimize the Canny criteria.

The first is

$$f(x) = \frac{S}{\omega} e^{\alpha|x|} \sin \omega x$$

Through mathematical inference and calculations and experiments[12], Deriche obtain the most effective filter when the value of ω approaches 0.By that the second formula can be given as

$$f(x) = Sxe^{-\alpha|x|}$$

Using the filter, Deriche edge detector can be able to process image in a short constant time independent of the desired amount of smoothing when the processed image is very noisy or requires a lots of smoothing. And these two filters have only one parameter so that changing the characteristics of image processed can be more easily.

3. Conclusion

By reading relative papers and materials, I understand these edge detectors more deeply with learning their priciple, specific steps and running code to precess example picture. It is not a easy journey for me, but I was glad to see these different pictures that computer shown when my computer run code successfully.

These edge detectors definitly are not perfect. With the image processed by edge detector the loss of amount of data reduce the possibility of human identification, but the remaining features can be stored by computer and be used to detect objects.

Whether it is the Sobel, Scharr, Canny and Deriche edge detector, they are first-order edge detector and proposed before 1990s, and now there are many kinds of methods that be different with these "traditional" approaches, which named deep-learning based approaches.

Deep-learning based approaches can be divided into 5 classes. They are general edge detection, object contour detection, semantic edge detection(Category-Aware) occlusion boundary detection and edge detection from multi-frames.[13]

The world is developing rapidly and I also should learn these cutting-edge technologies hard.

References

[1] H.G. Barrow and J.M. Tenenbaum (1981) "Interpreting line drawings as three-dimensional surfaces", Artificial Intelligence, vol 17, issues 1–3, pages 75–116.

[2] Lindeberg, Tony (2001) [1994],"Edge detection", Encyclopedia of Mathematics, EMS Press.

[3] Irwin Sobel, 2014, History and Definition of the Sobel Operator.

[4] K. Engel (2006). Real-time volume graphics. pp.

[5] Scharr, Hanno, 2000, Dissertation (in German), Optimal Operators in Digital Image Processing.

[6] B. Jähne, H. Scharr, and S. Körkel. Principles of filter design. In Handbook of Computer Vision and Applications. Academic Press, 1999.

[7] Dim, Jules R.; Takamura, Tamio (2013-12-11). "Alternative Approach for Satellite Cloud Classification: Edge Gradient Application". Advances in Meteorology. 2013: 1–8. doi:10.1155/2013/584816. ISSN 1687-9309.

[8] D. Kroon, 2009, Short Paper University Twente, Numerical Optimization of Kernel Based Image Derivatives.

[9] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[10] A. Neubeck and L. Van Gool, "Efficient Non-Maximum Suppression," 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 2006, pp. 850-855, doi: 10.1109/ICPR.2006.479.

[11] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, Int. J. Computer Vision, Vol. 1, pp. 167–187, April 1987.

[12] Alper Yilmaz, Mubarak Shah Fall 2012, UCF

[13] MarkMoHR, Awesome-Edge-Detection-Papers(`https://github.com/MarkMoHR/Awesome-Edge-Detection-Papers`), github