Lecture Node for Hough transform

Anonymous CVPR submission

Paper ID 18329015 Yuwei Hao

Abstract

This lecture note discusses the related content of Hough transform. Firstly, it introduces the basic principle of Hough transform. Secondly, it uses the code to verify the scene where the basic Hough transform to detect straight lines. Then it discusses the application extensions of the Hough transform, such as circles and ellipses.It also uses the relative codes to verify the correctness. Finally, this lecture note analyzes the generalized Hough transform, in contrast to the previous classical Hough transform..

1. Introduction

This part will explain some backgrounds and concepts related to Hough transform.

1.1. Background

The development history of Hough transform can be divided into three stages:

(1) Initial stage of development

In 1959, Paul Hough invented the Hough transform based on the need for bubble chamber image analysis, and published a patent called "Method and Means for Recognizing Complex Patterns" (Chinese name: methods and means for recognizing complex patterns) in 1962, in which What is described is the Hough transform of the original form. In this patent, any straight line in the Cartesian coordinate system can be represented by a slope and an intercept, and then the problem of unbounded transformation space is caused by the infinite slope of the vertical line.

(2) Promotion and application stage

In 1972, the Hough transform was popularized and used by Richard Duda & Peter Hart, and the Hough transform was extended to the recognition of objects of any shape, mostly circles and ellipses. Hough transform uses the transformation between two coordinate spaces to map a curve or straight line with the same shape in one space to a point in another coordinate space to form a peak, thereby transforming the problem of detecting arbitrary shapes into a statistical peak problem.

(3) Start to be popular

The Hough transform became popular in the computer vision community in 1981 because of a journal paper by Dana H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes." This paper describes a modification of the Hough transform using the principle of template matching. By knowing its shape and aiming it to find its position and orientation in an image, this modification enables the Hough transform to detect arbitrary objects described by its model, by Find the location of the model in the image to solve the problem of finding objects in the image. With the generalized Hough transform, the problem of finding the position of the model is transformed into the problem of finding the parameters of the transformation that maps the model into the image. Given the values of the transformation parameters, the position of the model in the image can be determined.

1.2. Line representation

A straight line can be represented by two variables in the two-dimensional space of the image, in the following two cases:

(1) In Cartesian coordinates: it can be represented by the parameters slope and intercept (k, b).

$$y = kx + b$$

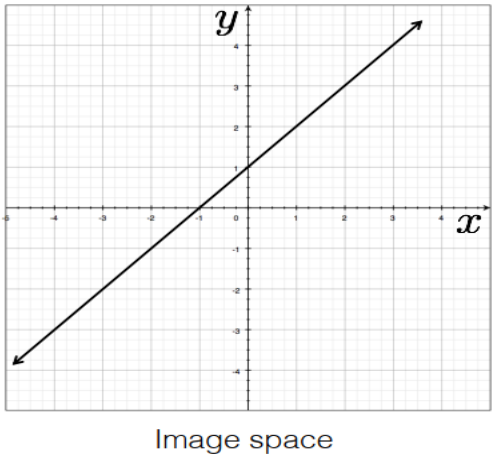

Figure 1: Schematic diagram of a straight line in the Cartesian coordinate system

(2) In the polar coordinate system: it can be represented by the parameters polar longitude and polar angle (r, θ).

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases}$$

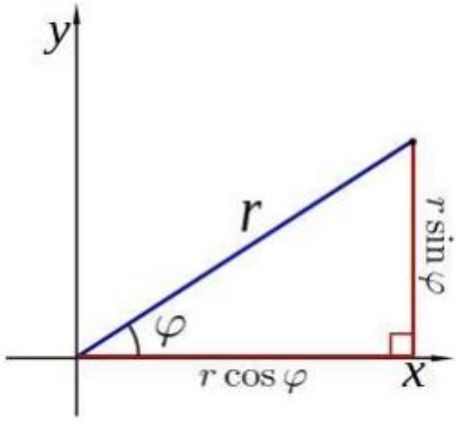

Figure 2: Polar coordinate system

2. Hough line detection

This part will specifically analyze the principle of Hough line detection.

2.1. Image space and Parameter space

The pixels in the image space can be embodied in the Cartesian coordinate system.

$$y = mx + b$$

Obviously we can convert it to an expression with *k* and *b* as variables.

$$y - mx = b$$

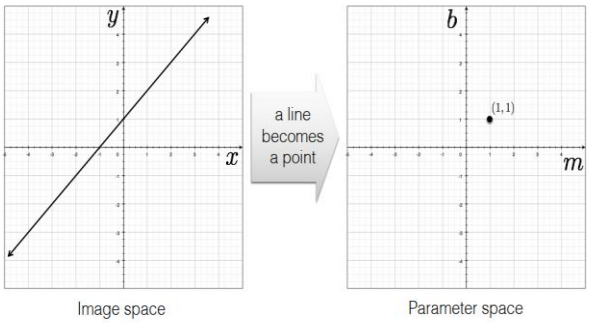

Figure 3: Interconversion between image space and parameter space(line to point)

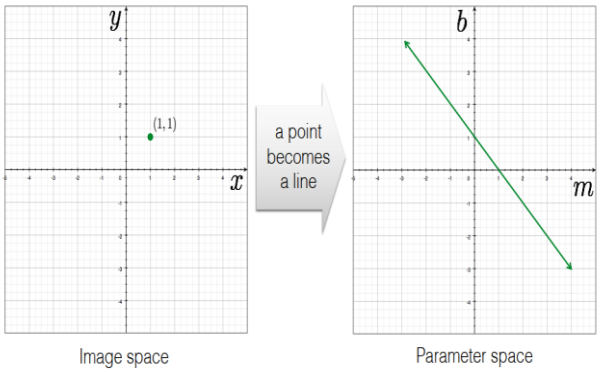

Figure 4: Interconversion between image space and parameter space(point to line)

As can be seen from the above figures, a straight line in the image space corresponds to a point in the parameter space. A point in the image space corresponds to a line in parameter space.

From the above concepts, we can know that if there are two different points in the image space (obviously these two points can determine a unique straight line), then in the parameter space, it can be transformed into two straight lines intersecting at a unique point. The value of k and b at the intersection point is the parameter value of the corresponding line in the image space.

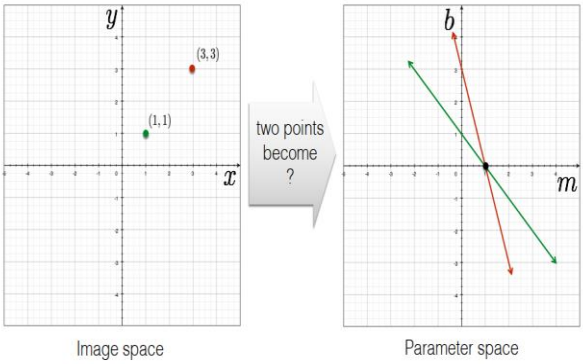

Figure 5: Using image space and parameter space to determine straight line parameters

2.2. Choice of parameter space

A parameter space with *k* and *b* as variables cannot represent all straight lines. For example, a line perpendicular to the x-axis in image space cannot be represented in parameter space. (the slope k is infinite)

So we can change the parameter space to the following representation:

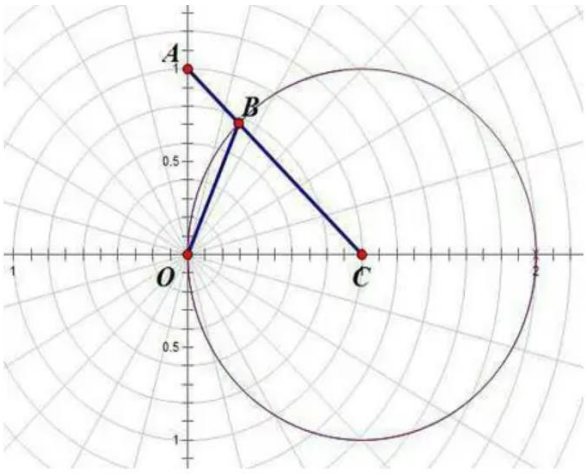

Figure 6: Polar coordinate system

The expression for a straight line is:

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{\rho}{\sin\theta}\right)$$

After simplification:

$$x\cos\theta + y\sin\theta = \rho$$

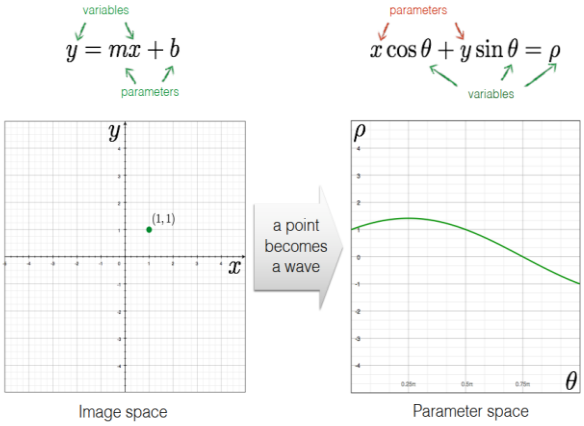

Figure 7: Interconversion between image space and parameter space(polar coordinate system)

At this time, a point in the image space (the Cartesian coordinate system x-y) corresponds to a curve in the parameter space (the polar coordinate system ρ-θ), to be precise, a sinusoidal curve.

2.3. The core idea of Hough transform: Statistics

The line detection of Hough transform is to transform the line in the image space to the point in the parameter space, and solve the detection problem through statistical characteristics.

Specifically, if the pixels in an image form a straight line, then the curve corresponding to these pixel coordinate values (x, y) in the parameter space must intersect at a point, so we only need to convert all the pixels in the image (coordinates value) into a curve in the parameter space, and then find the intersection of all the curves in the parameter space to determine the straight line.

In theory, a point corresponds to an infinite number of straight lines or straight lines in any direction. But in practice, we have to limit the number of lines (that is, a finite number of directions) to be able to calculate.

Therefore, we discretize the direction θ of the straight line into a finite number of equally spaced discrete values, then the parameter ρ is correspondingly discretized into a finite number of discrete values. Therefore, the parameter space will be discretely quantized into grid cells of equal size. After the coordinate value of each pixel point in the image space (rectangular coordinate system) is transformed into the parameter space (polar coordinate system), the pixel point will fall in a certain grid, and the accumulation counter of the grid unit will be incremented by 1. When all the pixels in the image space have undergone Hough transform, we count all the grid cells in the parameter space. After finding the grid with the largest accumulated count value, its coordinate value (ρ, θ) corresponds to the desired line in image space.

2.4. Code verification for Hough Line detection

The algorithm pseudo code flow is as follows:

1. Initialize accumulator H to all zeros
2. For each edge point (x,y) in the image
   For θ = 0 to 180
   ρ = x cos θ + y sin θ
   H(θ, ρ) = H(θ, ρ) + 1
   end
   end
3. Find the value(s) of (θ, ρ) where H(θ, ρ) is a local maximum
4. The detected line in the image is given by
   ρ = x cos θ + y sin θ

The experimental results are as follows:

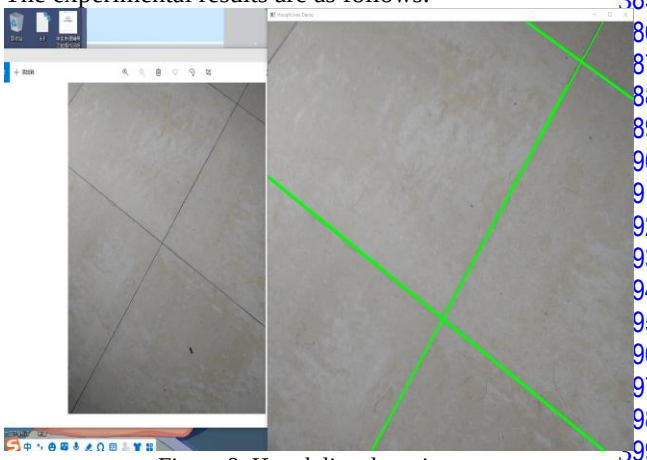

Figure 8: Hough line detection

2.5. Advantages and disadvantages of Hough line detection

Advantages:

(1) All points are processed independently, so occlusion can be handled;

(2) It has a certain robustness to noise. Noise points are unlikely to be consistent with any single box;

(3) The algorithm can detect multiple instances of a model in one pass;

(4) The algorithm has strong anti-interference ability and is not sensitive to the incomplete part of the straight line and other coexisting nonlinear structures in the image.

Disadvantages:

(1) With the increase of model parameters, the search time complexity increases exponentially;

(2) Non-target shapes can create false peaks in the parameter space, affecting the final result;

(3) For the selection of the unit grid size, multiple experiments are required to filter.

(4) The time complexity and space complexity of the algorithm are high, and in the detection process, only the direction of the straight line can be determined, and the length information of the line segment is lost.

3. Hough circle detection

This part will specifically analyze the principle of Hough circle detection.

3.1. Circle representation

In image space:

$$(x-a)^2+(y-b)^2=r^2$$

In parameter space:

$$a=x-r\cos\theta$$
$$b=y-r\sin\theta$$

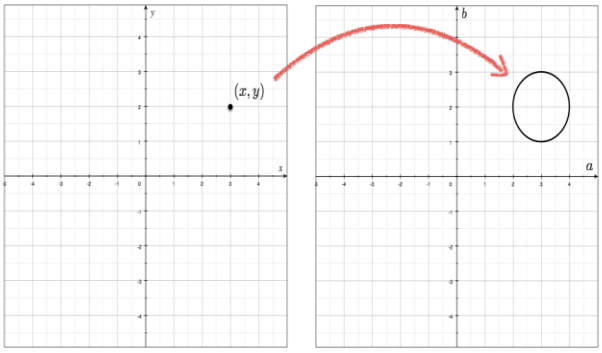

Figure 9: Interconversion between image space and parameter space(point to circle)

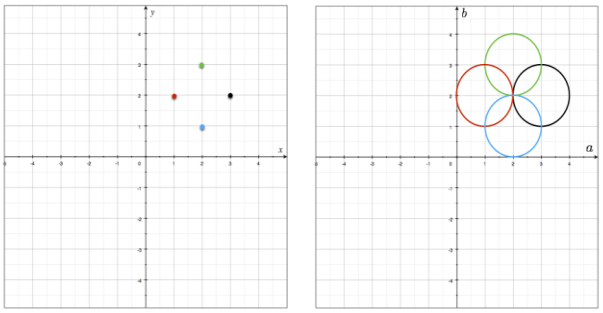

Figure 10: Using image space and parameter space to determine circle parameters

As can be seen from the above figure, the determination method for a circle is similar to that for a straight line. A circle in image space corresponds to a point in parameter space. A point in image space corresponds to a circle in parameter space.

From the above concept, we can know that if there are more than three different points in the image space, then in the parameter space, it can be transformed into more than three circles intersecting at the same point. The value of a and b at the only intersection point is the coordinate value of the center of the corresponding circle in the image space.

3.2. Algorithm details

Because the formula for a circle in image space is:

$$(x-a)^2+(y-b)^2=r^2$$

the formula for a circle in parameter space is:

$$a=x-r\cos\theta$$
$$b=y-r\sin\theta$$

For (x_0,y_0), we can uniformly define all circles passing through this point as:

$$a=x_0-r\cos\theta$$
$$b=y_0-r\sin\theta$$

This means that each group (a, b, r) represents a circle passing through the point (x_0,y_0).

(1) For a given point (x_0,y_0) , we can draw all circles passing through it in 3D Cartesian coordinates. In the end we will get a three-dimensional curve:

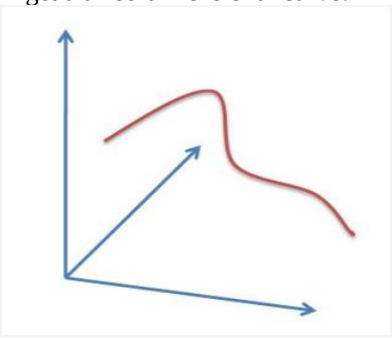

Figure 11: All circles passing through a point

(2) We can do the above operation for all the points in the image.. If the curves obtained by the above operations at two different points intersect in the space a-b-r, that is, they have a common set of (a,b,r), which means that they are on the same circle.
(3) The more curves intersect at a point, the more points the circle represented by the intersection means. We can set a threshold to determine how many curves intersect at a point before we consider a circle detected.
(4) This is what the Hough circle detection does. It traces the intersection between the curves corresponding to each point in the image. If the number of curves intersecting at a point exceeds the threshold, it can be considered that the parameter (a,b,r) represented by this intersection is a circle in the original image.

The above is the Hough circle detection algorithm. The problem is that its accumulation surface (the space where the 3D curve is drawn) is a 3D space, which means that it requires more computational consumption than the Hough line transform. The OpenCV Hough Circle Detection optimizes the operation of the standard Hough Circle Detection. It uses the "Hough gradient method".

3.3. Hough gradient method

1. Estimate the center of the circle
(1) Do a Canny edge detection on the original image to get the binary image of edge detection.
(2) Execute the Sobel operator once on the original image to calculate the neighborhood gradient values of all pixels.
(3) Initialize the center space N(a,b), so that all N(a,b)=0.
(4) Traverse all non-zero pixel points in the Canny edge binary image, draw a line along the gradient direction (the vertical direction of the tangent), and add N(a,b)+ =1.
(5) Statistically sort N(a,b) to get the possible center of the circle (the larger the N(a,b), the more likely it is the center of the circle).

2. Estimated radius (for a certain center (a,b))
(1) Calculates the distance from the center of the circle for all non-zero points in the Canny plot.
(2) The distance is sorted from small to large, and the appropriate possible radius is selected according to the threshold (for example, 3 and 3.5 are both classified as radius value 3).
(3) Initialize the radius space r, N(r)=0.
(4) Traverse non-zero points in the Canny graph, N(distance)+=1.
(5) The possible radius values are obtained by statistics (the larger the N(r), the more times the distance value occurs, and the more likely it is the radius value).

3.4. OpenCV: HoughCircles Function

The corresponding function in Python is cv2.HoughCircles.
The function prototype is:

```
cv2.HoughCircles(gray,cv2.HOUGH_GRADIENT,1,
30,param1=100,param2=30,minRadius=3,maxRadius=97)
```

Parameter Description(in order):
(1) Parameter 1 is an 8-bit grayscale image for the input image
(2) The detection method used is currently the Hough gradient method in OpenCV. Its identifier is CV_HOUGH_GRADIENT, and this identifier can be filled in this parameter.
(3) The reciprocal of the ratio of the resolution of the accumulator image that detects the center of the circle to the input image. If dp=1, the accumulator and the input image have the same resolution. If dp=2, the accumulator has half the width and height of the input image.
(4) The minimum distance between the centers of the circles detected by the Hough transform, that is, the minimum distance between two different circles that our algorithm can clearly distinguish. If this parameter is too small, multiple adjacent circles may be mistakenly detected as a coincident circle. Conversely, if this parameter is set too large, some circles cannot be detected.
(5) param1, has a default value of 100. It is the corresponding parameter of the detection method set by the third parameter method. For the currently only method Hough gradient method CV_HOUGH_GRADIENT, it represents the high threshold passed to the canny edge detection operator, and the low threshold is half of the high threshold.
(6) param2, also has a default value of 100. It is the corresponding parameter of the detection method set by the third parameter method. For the currently only method Hough gradient method CV_HOUGH_GRADIENT, it represents the accumulator threshold value of the circle center in the detection stage. The smaller it is, the more circles that do not exist at all can be detected, and the larger it is, the more perfect circles will pass.
(7) The minRadius of the int type has a default value of 0, which represents the minimum value of the circle radius.
(8) The maxRadius of the int type also has a default value of 0, which represents the maximum value of the radius of the circle.
This function can easily detect the center of the circle, but may not find a suitable circle radius. We can use the minRadius and maxRadius parameters to specify the maximum and minimum circle radii to assist the results of the circle detection. Alternatively, you can simply ignore the returned radius, leave both default values, use only the HoughCircles function to detect the center of the circle, and use additional steps to further determine the radius.

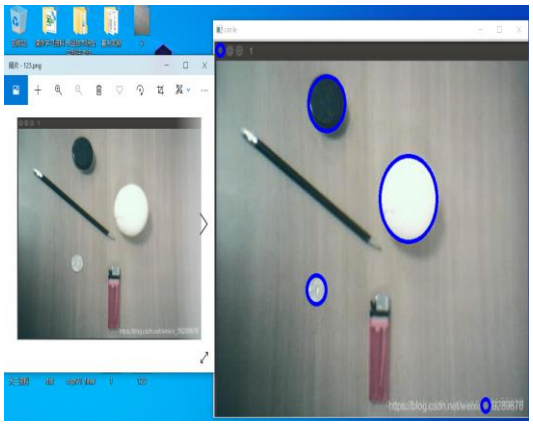

Figure 12: Hough circle detection

3.5. Advantages and disadvantages of Hough gradient method

Advantages:

(1) Reduced computational consumption and optimized algorithm performance.

Disadvantages:

(1) In the Hough gradient method, we use the Sobel derivative to calculate the local gradient, then the consequent assumption is that it can be treated as equivalent to a local tangent, and this is not a numerically stable approach. This will give correct results in most cases, but may produce some noise in the output.

(2) The entire set of non-zero pixels in the edge image is considered as a candidate for each center. Therefore, if the threshold of the accumulator is set low, the algorithm will take a long time.

(3) Because only one circle is selected for each center, if there are concentric circles, only one of them can be selected.

(4) Because the centers are sorted in ascending order of their associated accumulator values, and if the new center is too close to the previously accepted center, it will not be preserved. And when there are many concentric circles or approximate concentric circles, the tendency of the Hough gradient method is to retain the largest circle. It can be said that this is an extreme approach, because the default Sobel derivative here will generate noise, which is necessary for smooth images with infinite resolution.

4. Hough ellipse detection

4.1. Algorithm preparation

The core of the algorithm is to determine the inclination angle θ of the ellipse, and the idea of the cumulative count of the Hough transform is also used here. Suppose first that the center (O_x, O_y) of the ellipse has been found, and by the way, all the two points that contribute to the voting of this center point (almost all of these points are on the ellipse, of course, there may be some point pairs not on the ellipse, but about this center point point symmetry) are placed in a set of edge points *Q*. Since the major axis passes through the center of the ellipse, and then set the slope of the major axis to be k_0, the equation of the straight line where the major axis is located is $y - O_y = k_0(x - O_x)$. In analytic geometry, if the equation of the line is *Ax + By + C = 0*, and the coordinates of point *P* are (x_0, y_0), then the distance *d* from point P to the line is:

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

From the above formula, it can be known that the distance d_0 from the point *P* to the straight line where the long axis is located is:

$$d_0 = \frac{|k_0 x_0 - y_0 + O_y - k_0 O_x|}{\sqrt{k_0^2 + 1}}$$

Since the ellipse is symmetrical about the long axis, the strings l_i perpendicular to the long axis in the ellipse will be bisected by the long axis. Assuming that the slope of a major axis is k_0, find out the chord in the ellipse perpendicular to k_0 (two straight lines are perpendicular, and the slope meets $k_1 k_2 = -1$), and then calculate the distance from the two ends of the chord to the major axis to see whether the chord is evenly divided by the major axis. If you can find many chords bisected by the long axis of the hypothesis (the number of occurrences is greater than a certain threshold *T*), it can be considered that the long axis slope k_0 initially assumed is correct, that is, the inclination angle is determined θ.Otherwise, repeat the above process with another slope. It should be noted that in general, the long axis is near the straight line of the point pair with the largest or larger distance in the point set *Q*, so you can first calculate the slope k_s of the straight line of the point pair with the larger distance in *Q*, and then let the slope of the long axis gradually change in a certain step from k_s, so that it can quickly find the correct long axis slope.

4.2. Algorithm details

The specific algorithm flow is as follows:

(1) Preprocess the original image, including filtering, grayscale equalization and edge extraction, convert the image into a binary image, and the edge pixels are the feature points.

(2) Determine the center of the ellipse (O_x, O_y). Put the two points that contribute to the vote of this central point into a set of edge points *Q*. If multiple center points are detected (corresponding to multiple edge point sets Q_i), there may be multiple ellipses in the figure, and multiple

threads are opened. One thread is responsible for the processing of one center point, and the following steps are continued in each thread.

(3) Calculate the slope k_i between each two points in the edge point set *Q* and store it in an array.

(4) The inclination angle of the ellipse is determined by using the symmetry of the ellipse about the major axis (or minor axis) θ ,That is, first find the direction of the long axis. Assuming the slope k_0 of a long axis, the slope of the straight line perpendicular to the long axis can be obtained through simple calculation, so that all point pairs with the slope perpendicular to k_0 can be found (that is, the two ends of those strings perpendicular to the assumed long axis, but the found point pairs may not be on the ellipse, and may also be bisected by the assumed long axis, resulting in a false accumulation, but this situation is rare and can be ignored). Then according to the method described above, find out θ .

(5) After obtaining the inclination angle θ , starting from the center point, select a point on the long axis as the focus F_1 of the ellipse (the focus is on the long axis), then another point symmetrical with F_1 about the center point is another focus F_2 .

(6) We do a one-dimensional accumulation of α. After the two focal points are selected, a point *M* can be taken arbitrarily. Assuming that M is on the ellipse, according to the properties of the ellipse $F_1M + F_2M = 2\alpha$, an α value can be calculated. Then take a lot of *M* points, and perform one-dimensional accumulation on α. When the same α value is calculated many times (greater than a certain threshold *T*), it means that the previously selected focus is correct, so that the focal length *c* and the semi-major axis α are determined. , use the formula $c^2 + \beta^2 = \alpha^2$ to calculate the semi-minor axis length β; otherwise, start from the center point and take 2 pixels as a step outward along the long axis direction (the smaller the step size, the more accurate, but the longer the running time.), take the next point as the focus F_1, and return to step (5). Since the focal length *c* is the longest, that is, α, the distance *L* between the two points with the longest distance in the edge point set *Q* can be calculated (which is close to α). When the distance between the focal point F_1 and the center point is greater than *L*, At the end of the algorithm, if you still cannot find an α value with a number of occurrences greater than the threshold, it means that a false ellipse center is found at the beginning, and there is no ellipse in this place.

5. Generalized Hough transform

5.1. Introduction

Generalized Hough Transform (GHT) was proposed by DanaH.Ballard in 1981, which was adjusted based on the principle of template matching on the basis of Hough transform. The generalized Hough transform does not require an analytical formula that can give the shape to be detected, it can detect any given shape.

Through the previous study, we have been able to extract an essence of the Hough transform - to give a description mode of the graph, such as the equation, function, table, etc. of the graph point set, and then use this mode to traverse the parameter space, The pattern's graphic point set is projected to a point in the parameter space (the actual discrete situation is generally not perfectly concentrated to one point), and this point records the number of graphic points.

The reason why the generalized Hough transform can handle graphics of any shape is not to find an equation that can represent any graphics (which is impossible), but to describe a graphics in the form of a table, and save the coordinates of the edge points of the graphics in a table. , then the graph is determined.

So in fact, whether it is a straight line (in fact, a line segment), a circle, an ellipse or other geometric shapes, the same method can be used to process them. The difference is that the shapes at this time are customized and real, while the algebraic equations The pattern is continuous and abstract, there is only one equation for the circle, but the custom circle is infinite, as long as you think it is round enough.

Of course both representations have their own advantages and limitations. After you have the table, you need to find a conversion algorithm that can project the graph point set to a point in the parameter space. For example, the line and circle Hough transform project the point set through equations (functions) and traversal, so that it belongs to a certain line or circle. The points are concentrated to a point; then how to project with only one table describing the coordinate points of the edge of the graph? We can regard this table as a template and perform template matching. If most of the points are successfully matched, it can be understood that these points are projected to a point, but at this time there is no need to search for peaks in the parameter space. This mode It can be considered that there is no relationship between the parameters, so it is a complete traversal.

However, in the case of rotation and scaling, the template matching Hough transform is very time-consuming, and it can also be imagined that the traversal search time is increased because the parameters are irrelevant. The most subtle feature of Ballard's (1981) generalized Hough transform is that two associations are added to the parameters, so that in the case of translation and rotation

(without scaling), only one parameter needs to be traversed, and the three parameters are the center coordinates of the graph (horizontal coordinates). Vertical), rotation angle (relative to the reference graph), Ballard's algorithm saves the radial value of the edge point of the reference graph to the center in advance, and uses the gradient direction of the edge point of the graph to be searched (represented by the angle relative to the coordinate axis) as an index to find the corresponding The radial amount of , and the projection is completed after adding this amount, so the parameter to be traversed is only the rotation angle, so there are two associations. Of course, if you add scaling, you need to traverse two parameters, which is only the same as the scale of the Hough detection circle. The graph table of this generalized Hough transform no longer directly saves the coordinates, but the gradient of the edge point plus the radial quantity, and a graph can be represented by giving these quantities.

5.2. Advantages and disadvantages of Generalized Hough transform

Advantages:

(1) It is robust to partially or slightly deformed shapes (ie, robust to recognition under occlusion).

(2) Good robustness to interference from other structures (i.e. other lines, curves, etc.) in the image.

(3) It has strong anti-noise ability.

(4) Multiple targets of the same type can be found in a single pass.

Disadvantages:

(1) The dimension of the parameter space is high, and there is an exhaustive process. So it requires a lot of storage and a lot of computation.

References

[1] Chen Yugen, Yang Yan. Two improved algorithms for ellipse detection based on Hough transform [J]. Semiconductor Optoelectronics, 2017, 38(05): 745-750. DOI: 10.16818/j.issn1001-5868.2017.05.026.

[2] Hough V, Paul C. Method and means for recognizing complex

[3] XuL, OjaE. Randomized Hough transform (RHT): basic Mechanisms, algorithms and components [J].Computer Vision Graphic Image Proc. : Image Understanding,1993, 57(2): 131-154