8 Hough Transform for Higher Orders

Hough Transform for Higher Orders

Although the Hough transform has usually been used for straight lines, it has also been proposed for higher order sections as well. We will consider here the problem of finding ellipses. An ellipse at the origin and parallel to the axes is characterised by an equation

$$\left(\frac{x}{a}\right)^2+\left(\frac{y}{b}\right)^2=1$$

In general we need to consider the rotated ellipse:

$$\left(\frac{x\cos\theta-y\sin\theta}{a}\right)^2+\left(\frac{x\sin\theta+y\cos\theta}{b}\right)^2=1$$

and we need the ellipse to be at any position in the image (xc, yc):

$$\left(\frac{(x-x_c)\cos\theta-(y-y_c)\sin\theta}{a}\right)^2+\left(\frac{(x-x_c)\sin\theta+(y-y_c)\cos\theta}{b}\right)^2=1$$

So we can deduce that our parameter space requires five parameters to be identified {xc,yc,a,b, }. Thus the histogram array will be five dimensional. Clearly, we can no longer adopt the same strategy used for the straight line Hough transform, since, if we are to quantize each parameter into n levels we will require n^5 histogram points. The maximum feasible size is about n=32, which is too small for typical accuracy, still requires about 32Mbytes for the histogram. Accordingly, the usual approach is to decompose the problem into two parts namely the identification of the centre, followed by detection of the major and minor axes and the angle.

Estimation of the Centre

One way to find the centre is to use the symmetry property of the ellipse. That is to say, each point on the ellipse has an opposite point, which should have the same edge point gradient. We can apply the Hough method by finding all pairs of edge points with similar gradient, and calculating their mid points. A two dimensional histogram is constructed to count these mid points, and the maxima are selected as candidates for ellipse centres. This is a time consuming process, since each edge point must be checked against all others for similarity in edge direction. If memory size permits, the computation time can be reduced by sorting the edge points into order of direction.

This method fails if part of the ellipse is obscured in the image. An alternative strategy is to construct the centre from the tangent directions of a number of edge points. The way in which the construction is made is shown in diagram 8.1. The centre lies on the line joining the intersection point, r, of two tangents at the points p and q, to the midpoint of p and q. The tangent directions are given as:

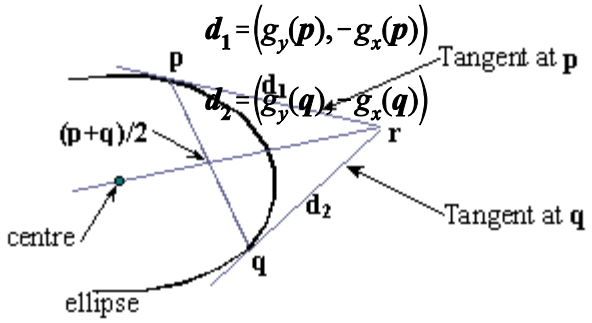

Diagram 8.1: Estimating an ellipse centre from its tangents

The point r can be found from:

$$r=p+\alpha d_1=q+\beta d_2$$

which can be solved for a and b using the two Cartesian components. The line through the centre therefore has equation:

$$c=\frac{1}{2}m(p+q)+(1-m)r$$

where c is any point on the line, and m is a general parameter. Separating this equation into the two Cartesian components, we can construct a two dimensional histogram in x-y, by stepping through y, computing the value of m from the y-component equation, then substituting the value into the x-component equation to calculate x and hence determine which histogram location to estimate. Again, we need to carry out the process for each pair of edge points in the edge map, after which the histogram maxima give us an estimate of the ellipse centre. The

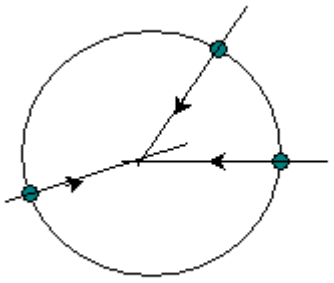

Diagram 8.2
Estimating the centre of a circle from the edge point gradient directions

corresponding construction method for the circle is rather simpler (Diagram 8.2).

Estimating the Other Parameters

The further three parameters may now be estimated directly from the equation:

$$\left(\frac{x\cos\theta - y\sin\theta}{a}\right)^2 + \left(\frac{x\sin\theta + y\cos\theta}{b}\right)^2 = 1$$

with all points first translated to the estimated centre [xc, yc]. However, we still require a histogram array of size n^3 for quantization into n levels. A quantization into 128 levels still requires an array with 2M entries, which may be beyond current resources. Accordingly a further decomposition step is required. If we rearrange the equation by multiplying by a^2 we get:

$$(x\cos\theta - y\sin\theta)^2 + \left(\frac{a}{b}\right)^2 (x\sin\theta + y\cos\theta)^2 = a^2$$

letting h=a/b and differentiating we get:

$$2(x\cos\theta - y\sin\theta)(\cos\theta - y'\sin\theta) + 2(x\sin\theta + y\cos\theta)(\sin\theta + y'\cos\theta)h^2 = 0$$

where y' = dy/dx, and can be obtained from the edge direction.

This equation has now only two parameters, which can be estimated by the normal Hough method without any additional computation. That is to say, we construct a histogram array in h^2 and , and for each edge point, substitute the values of x y and y' into the equation to get an equation for h^2 in terms of . For each quantized value of f we compute h^2 and increment the appropriate histogram entry. The space is limited to the range of [0..2] in , but theoretically unlimited in h. In practice, it will be necessary to place limits on h, that will reflect the degree to which the ellipses can be squashed. Once these two parameters have been estimated, the value of a^2 can be determined by using a one dimensional Hough transform with the original equation.

The Hough transform for template matching

One possibility is to use the Hough transform for template matching. This is applicable in cases where we have prior knowledge of the shape of the object we are trying to find, but do not know its location, orientation or even size. The problem is to parameterise the shape that we wish to identify in a way that can be easily identified. One way is found in Ballard and Brown's book. The process is easier if we know the objects orientation. The first stage is to define an arbitrary reference point inside the object, which we loosely call the centre [xc,yc]. For each edge point, we can define a line which goes through the centre. This may most efficiently be stored as the angle between the edge point (radial) direction, and the line joining that point to the centre. For faster computation, it is convenient to store the components of the direction vector **r**. The geometry is shown in diagram 8.3. We construct a table of **r** (or) against the absolute direction . Now, for each image point **p** we can obtain the equation of a radial line as:

$$\boldsymbol{c} = \boldsymbol{p} + \mu \boldsymbol{r}[\theta(\boldsymbol{p})]$$

and accumulate a two dimensional histogram to estimate the likely centre, exactly as we did for the ellipse. If we do not know the orientation of the object, the equation now needs an additional parameter :

$$\boldsymbol{c} = \boldsymbol{p} + \mu \boldsymbol{r}[\theta(\boldsymbol{p}) + \phi] \quad (1)$$

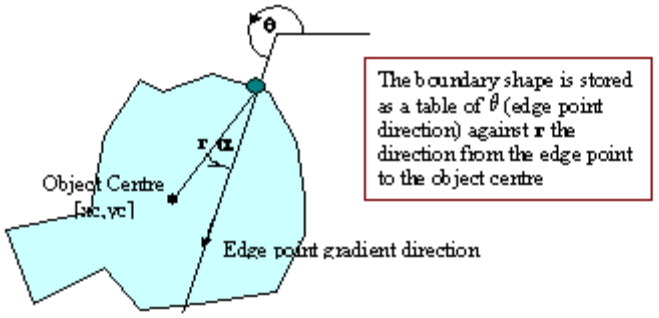

Diagram 8.3 Defining an object's shape numerically

Which will reduce the accuracy with which we can quantize the space. Having identified xc,yc and it becomes a trivial problem to estimate the size. For the formulation in which stored the shape in terms of the vector **r**, we can now take the parameter in the equation of the centre line as a measure of size, and accumulate it in a one dimensional histogram using the edge point data with equation 1.

Adaptive Hough Transform

A computational strategy, particularly useful for dealing with cases where three or more parameters need to be estimated, is to iteratively increase the resolution of the quantization. Thus, for equation (1) above, we could start by quantizing xc,yc and into 10 levels. The estimate is then made, and one or more values extracted. For each of these candidates, the histogram points to which they belong are then divided further into say a further 10 levels, and the process repeated. Three iterations gives us the equivalent of a resolution of 1000, which is sufficiently accurate for most purposes. This strategy reduces considerably the computation time and memory requirement, but has the disadvantage that thresholding is required, and consequently the correct histogram point may be hidden at high resolution by combinations of side lobes.

Representing Curves by Short line segments

Clearly there is a limit to the degree of curve that can be traced using the Hough transform. For second order and over, the problem must be decomposed, and it is not clear that a decomposition will always be available. Template matching can solve the problem in some cases, but only if we have prior knowledge of the boundary shape. One alternative which has proved successful is to divide the image into small square windows, and to extract some part of the boundary from each window, and link up the parts to form a curve. There are a number of advantages of using short line segments as an intermediate step for curve extraction.

1. The Hough transform carried out in a small window is far less prone to problems of side lobes than for a whole image. This is because, providing the windows are small, we can assume that in most cases only one line segment is present.

2. The extraction of a line segment from a small window can be performed very fast.

3. The windows can be processed independently if parallel processing is available.

4. Short line segments extracted in the presence of noise can compensate for missing information at the pixel level. ie we can extract an eight pixel line with say six voting edge points. However, two missing edge points may cause a direct search of the edge map to fail.

Of the Hough techniques already discussed, the Wallace parameterisation is most suitable for this application. Looking at Diagram 8.4, we have only sixty four pixels which could be edge points, and a range of 1 to 28 for the parameters. For each edge pixel we need to store an array indexed by the start parameter, in which the entry is the computed finish pixel. To do this simply requires less than 2000 bytes, and more complex structures can be used to save space, since the space is symmetric in s and f. Each point will vote for on average about 15 lines. Further refinements can be added such as extracting the line segments taking into account part of the adjacent image windows. This can be done by processing, for example, a 12 by 12

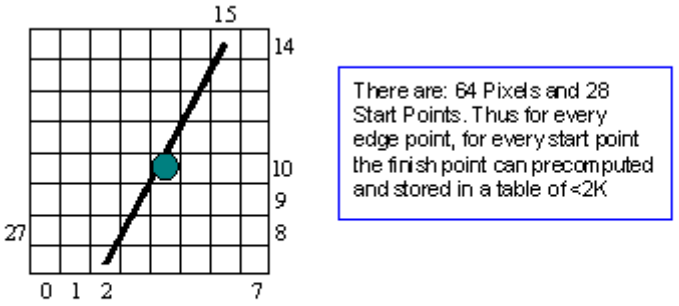

Diagram 8.4: Hough Transform by Table Lookup

window surrounding the 8 by 8 window for which we want to determine the best line segment. The idea is illustrated by Diagram 8.5, and it will be seen to provide some degree of immunity to both noise and bias effects.

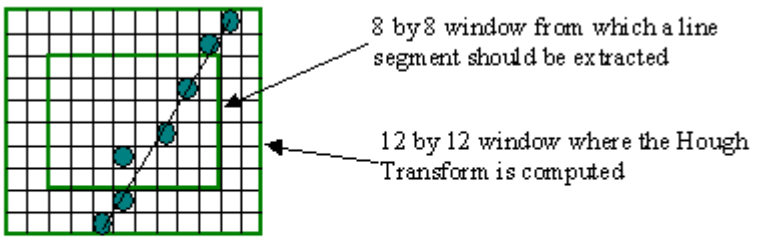

Diagram 8.5: Overlapped windows for extraction of line segments