MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

FINDING LINE SEGMENTS BY SURFACE FITTING TO THE HOUGH TRANSFORM

Wayne Niblack and Tai Truong
IBM Almaden Research Center

Dept. K54/802 650 Harry Road
San Jose, CA 95120

Abstract

We describe a method based on the Hough transform to compute a set of parameters that completely describe line segments. By complete, we mean the description includes the endpoints of each segment as well as the number of points along the segment, and, assuming a noise model N, the parameter(s) of N. The method involves fitting a surface locally to the Hough array, where the variables of the fit are the parameters we are looking for. In addition to providing a complete description for each segment, the computed estimates of ρ and θ are better than the resolution of the Hough array $H(\rho,\theta)$ so that a relatively low resolution H may be used.

1 Introduction

The Hough transform is often used to find instances of a parameterized object in an image. The most common use is to find lines, parameterized by m and b in $y = mx + b$ or, as many authors prefer, ρ and θ in $\rho = x\cos\theta + y\sin\theta$. In an earlier paper[1], we described a method for achieving very high accuracies in ρ and θ parameter estimation by using a series of one dimensional interpolations on the values in $H(\rho,\theta)$. In this paper, we extend this to a full two dimensional interpolation, with three main results: (1) the interpolated values for ρ and θ maintain their high accuracy; (2) in addition to ρ and θ, we compute parameters that provide a complete description of each line segment (endpoints, number of points in the segment, and noise characteristic about the segment); and (3) the interpolation is valid over a wide range of values of the Hough space resolution $\Delta\rho$ and $\Delta\theta$, and in particular, relatively coarse resolution may be used.

Other authors have used properties of the Hough array other than the location of the peak to gain more information about the lines in an image (for example, [2] uses the shape of the "butterfly" to better design a line segment detection filter), but ours is the first case we know where a complete line segment description is derived.

Applications that use the Hough transform and require the line segment endpoints (the most interesting additional parameters) typically first compute ρ and θ from $H(\rho,\theta)$, and then inspect the swath along ρ and θ in the image, applying various heuristics to find the beginning and end of the segment(s)[3].

2 The model

Consider a line segment S in an image with parameters ρ_0, θ_0 of length L_0 centered at (x_0, y_0). (Note: y_0 is not an independent parameter, but can be computed from ρ_0, θ_0, and x_0.) Suppose that in the image, the edgels detected about S are at locations that have a Gaussian distribution with mean 0, standard deviation σ_0 in the direction orthogonal to S, and that the average number of edgels per unit length along S is μ_0. See Figure 1. Suppose that we compute a Hough transform with with ρ and θ resolutions $\Delta\rho$ and $\Delta\theta$. We approximate the expected number of points in a cell of $H(\rho,\theta)$ (this is by definition the expected number of points in a swath of width $\Delta\rho$ centered along the line S' from u to v in the figure) as the area A' under the slice along S' times the width $\Delta\rho$ of the swath, times the density of points along S'. Then

$$A' = \frac{\int_{v'}^{u} e^{-\frac{t^2}{2\sigma^2}}\,dt}{\sin\phi}$$

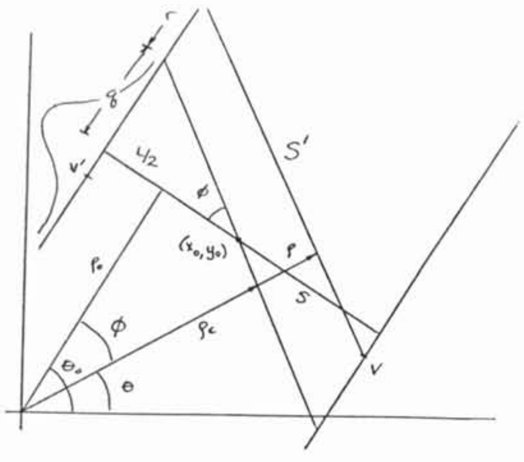

Figure 1: The parameters of a line segment.

where $\phi = \theta_0 - \theta$, $u = q + r$, $v = v\prime = q - r$, $q = (L/2)\tan\phi$, $r = (\rho - \rho_c)/\cos\phi$, and $\rho_c = x_0 \cos\theta + y_0 \sin\theta$. The density of points along $S\prime$ is $\mu_0 \cos\phi$. Combining these gives:

$$\begin{aligned} \hat{H}(\rho,\theta) &= \hat{H}(\rho,\theta; x_0, y_0, \theta_0, L_0, \mu_0, \sigma_0, \Delta\rho) \\ &= A\prime\Delta\rho(\mu_0 \cos\phi) \\ &= \frac{\Delta\rho\mu_0 \cos\phi}{\sqrt{2\pi}\sigma \sin\phi} \int_{z_1}^{z_2} e^{-\frac{t^2}{2\sigma^2}} dt \end{aligned}$$

where we have regrouped terms as $z_1 = (z - a)/\cos\phi$, $z_2 = (z + a)/\cos\phi$, $z = \rho - \rho_c$, and $a = (L_0/2)\sin\phi$.

Given an actual Hough array H resulting from a set of edgels, we estimate the parameters $x_0, y_0, \theta_0, L_0, \mu_0$, and σ_0 as those that minimize $\chi^2 = \sum(\hat{H} - H)^2$ in the vicinity of each peak. This is a non-linear least squares minimization, and we use Levenberg-Marquardt iteration[4].

Initial estimates. Initial estimates of the parameters are necessary for the iteration, and these can be made quite reasonably using values of $H(\rho,\theta)$ near the peak. To compute estimates of ρ_0 and θ_0, we smooth $H(\rho,\theta)$ in the ρ direction with a window of length ρ_W (we use 5), and then take the peak location. For n_0, the initial estimate is the number of points in this same window, and for σ_0 it is their standard deviation. To compute an initial guess for L_0, we estimate the endpoints of the segment by looking in the columns of $H(\rho,\theta)$ at $\pm T_\theta$ columns (we use 2) from our guess of θ_0. In these columns, we find the minimum and maximum ρ values enclosing the same number of points as are in the peak. We assume these are the endpoints of the segment. Thus we have two sets of ρ and θ values for each endpoint, and can solve for their x, y locations and thus an estimate of the length L_0.

3 Discussion

Iterations. The iteration is stopped when χ^2 decreases by less than 0.01 on two successive iterations. Convergence is rapid, and typically occurs in about six iterations.

Effect of window size. The minimization of χ^2 is done over a local window centered (approximately) on a peak in $H(\rho,\theta)$. Results from windows of sizes 5x5, 7x7, and 11x11 are shown in Table 1. Results in this and the next table are for input data generated by adding Gaussian noise of $\sigma = 1.2$ in the orthogonal direction to points along the line $\rho = 40.7, \theta = 47.3, x_0 = 82, n = 92$, and $L = 123.82$. The 11x11 and 7x7 windows are similar, but at 5x5 the accuracy begins to degrade.

Effect of $\Delta\rho$ and $\Delta\theta$ resolution. The resolutions $\Delta\rho$ and $\Delta\theta$ at which the Hough array is accumulated affect both the processing resources (speed and memory), and the accuracy of the fitting. Table 2 shows results from runs using a 7x7 window in $H(\rho,\theta)$ where $\Delta\rho = 2$ and $\Delta\theta = 2$.

Sensitivity. The parameters ρ and θ shift the surface $H(\rho,\theta)$, x rotates it, and n scales it. These effects are essentially orthogonal and are accurately computed by the fitting procedure. L and σ are more difficult. They have related effects on the surface, both tending to spread it out. The model is less sensitive to this type of change, with the result that computed values for $\hat{L}$ and $\hat{\sigma}$, as shown in the tables, are less accurate. An area for further study is a re-parameterization of the surface model of $H(\rho,\theta)$ to increase the sensitivity for all parameters.

Limitations. The main limitation of the method is that only one segment at a given ρ and θ can be modelled. Separate, co-linear segments would not be detected.

4 Multiple Line Segments

We generated a test case with 6 line segments on a background of uniform noise. $H(\rho,\theta)$ was

	Run 1	Run 2	Run 3	Run 4
		5 x 5		
$\hat{\rho}$	40.52	41.07	40.85	40.99
$\hat{\theta}$	47.15	47.66	48.01	47.09
$\hat{x_0}$	84.33	85.14	80.97	79.78
$\hat{n_0}$	93.96	88.41	88.80	94.83
$\hat{L_0}$	114.51	108.53	112.59	137.07
$\hat{\sigma_0}$	1.34	1.15	1.02	1.12
		7 x 7		
$\hat{\rho}$	40.79	40.90	40.53	40.51
$\hat{\theta}$	47.40	47.39	47.04	47.16
$\hat{x_0}$	80.91	81.61	80.66	83.15
$\hat{n_0}$	91.41	88.75	91.89	93.55
$\hat{L_0}$	130.54	118.51	114.75	125.80
$\hat{\sigma_0}$	1.14	1.09	1.26	1.28
		11 x 11		
$\hat{\rho}$	40.68	40.71	40.74	40.81
$\hat{\theta}$	47.33	47.31	47.48	47.45
$\hat{x_0}$	79.00	79.99	81.20	81.43
$\hat{n_0}$	94.98	91.66	91.31	92.01
$\hat{L_0}$	131.56	127.25	123.14	128.08
$\hat{\sigma_0}$	1.54	1.09	1.11	1.17

Table 1: The effect of window size. The top results are from 4 runs (one per column) interpolating over a 5x5 window, the middle over a 7x7 window, and the bottom an 11x11 window. Every run uses a new random noise sample added to the input points. (For these runs, the resolution $\Delta\rho$ and $\Delta\theta$ of $H(\rho,\theta)$ were both 1.)

	Run 1	Run 2	Run 3	Run 4
$\hat{\rho}$	41.05	40.66	40.68	40.70
$\hat{\theta}$	47.59	47.35	47.05	47.11
$\hat{x_0}$	82.83	80.63	82.65	79.99
$\hat{n_0}$	91.87	92.37	91.36	93.24
$\hat{L_0}$	116.75	126.41	125.00	131.13
$\hat{\sigma_0}$	1.37	1.41	1.12	1.46

Table 2: The effect of $\Delta\rho$ and $\Delta\rho$ resolution. The figures show results from 4 runs (one per column) interpolating over a 7x7 window in $H(\rho,\theta)$ where $\Delta\rho = 2$ and $\Delta\theta = 2$. Compare these with the middle results of the previous table.

accumulated, and the fitting procedure applied to each peak. To improve the accuracy by avoiding the influence from nearby peaks, when the i-th line segment was found, the points in the swath at $\hat{\rho_i} \pm 2\hat{\sigma_i}$ were removed from $H(\rho,\theta)$ prior to finding and fitting about the next peak. This process continued until the maximum peak in $H(\rho,\theta)$ had less than 10 points. Results are shown in Figure 2.

5 Conclusions

We have described a method for computing a complete description of a line segment based on fitting a surface to the Hough transform. The method provides high accuracy in the estimates of ρ and θ, as well as good estimates of the location and length of, the number of points in, and the noise characteristics of the points in the line. The method is computationally feasible, requiring only a few iterations, and is applicable to cases of both single and multiple lines.

References

[1] Wayne Niblack and Dragutin Petkovic. On improving the accuracy of the hough transform. *International Journal of Machine Vision and Applications*, 3(2):87–196, 1989.

[2] J. F. Boyce, G. A. Jones, and V. F. Leavers. An implementation of the hough transform for line and circle detection. *SPIE Inverse Problems in Optics*, 808:69–75, 1987.

[3] Jacqui Skingley and A. J. Rye. The hough transform applied to sar images for thin line detection. *Pattern Recognition Letters*, 6:61–67, 1987.

[4] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1988.

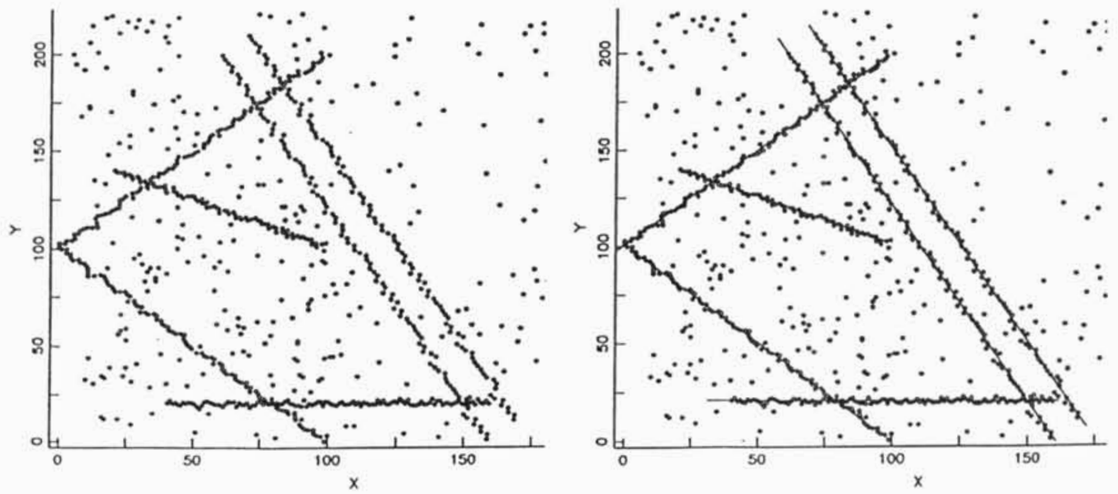

Figure 2: Multiple line segment detection example. The input data is on the left. The data with computed line segments overlaid is on the right.