Comp 254, Image Processing and Analysis
Spring 2002, Prof. Guido Gerig

Assignment 3: Hough Transform

Out: Tuesday April-02-2002
Due: Tuesday April-16-2002 (theoretical and practical part)
Please provide a paper copy of essential results of the practical part

Required Readings: Generalizing the Hough Transform ..., D.H. Ballard 1981
Use of Hough Transformation to detect Lines ..., Duda and Hart
Linking Image Space and Accumulator-Space, G. Gerig, ICCV'87

I. Theoretical Problems

Use the Hough Transform to recognize straight lines in contour images. The parametrization $y = ax + b$ is suboptimal (why?), so you should use the parametric form $\rho = x \cos\theta + y \sin\theta$.

Problem 1: HT as a point to curve transformation

Show that points that lie on the same curve $\rho = x_0 \cos\theta + y_0 \sin\theta$ in parameter space with axes (ρ, θ) all represent straight lines in image space which intersect at a specific image point (x_0, y_0).

Problem 2: Compensation for errors

Given a straight line g_0 passing through two opposite corner points of a quadratic image (lower right to upper left). The quantization of cells in parameter space is given by $\Delta\rho$ and $\Delta\theta$. We would like to get a feeling for the error region in image space (region for all points in image space) which contributes to the cell (ρ_0, θ_0) representing the straight line g_0. You can think of a cell in parameter space centered at (ρ_0, θ_0) with widths $\Delta\rho$ and $\Delta\theta$.

a) Do the calculations b) and c) for putting the image origin (0,0) lower left and also for putting the image origin at the center of the image. Discuss the differences.

b) Which is the longest straight line piece in the image orthogonal to g_0 that could "erroneously" contribute with its full length to the parameter cell centered at (ρ_0, θ_0)?

c) Calculate a numerical example for an image size of 512x512 pixels and two quantizations with $(\Delta\theta = 1^o, \Delta\rho = 2$ pixels), and $(\Delta\theta = 5^o, \Delta\rho = 5$ pixels).

Problem 3: Detection of ellipses

Describe how you would approach the problem of finding ellipses of arbitrary size and orientation if a) only contours of elliptical figures are available and b) if additionally you can make use of the local contour direction provided by an edge detection method.

I. Practical Problem

Problem 3: Hough Transform for straight lines

Write a program that calculates the Hough Transform for straight lines using the parametric form $\rho = x \cos\theta + y \sin\theta$. (Hint: Use the result of Theoretical Problem 2 to decide where to put the origin most efficiently).

Each point in image space generates a cos-curve which is accumulated in parameter space. The increment can be chosen as 1 for each point, but could also be proportional to the edge-strength which is the brightness attribute for each contour point (see contour images described below), so that the accumulation reflects the "importance of edge points".

Images can be found on the web-page. There are sets of three images (original, contours and angles) for the "telephone" image and for a simulated scene containing edges and lines "edges-lines". The contour images contain pre-calculated, thinned contours on a black background. The brightness attribute for each contour point is the edge strength, or local gradient magnitude. The angle images represent the local orientation of the gradient at each pixel of the image. If you read the two images into your program, you can think of overlaying the contour image with the angle image and get the angle attribute for each contour point. The brightness value in the angle image is $2\pi * 40$ to optimally scale the 2π within the byte range.

a) Implement the Hough-Transform and create an accumulator that corresponds to a 2-D image array with axes θ and ρ (similarly to the scatter-plot in assignment 1). Please be careful about the discretization, or sampling, of the cos-curve to avoid getting a dotted line like in the PC-demonstration in the course. (The curve $A * cos(\theta - \delta)$ can have a gradient larger than 1!). Don't use the angle information at this stage. Choose the cell size $\Delta\rho$ and $\Delta\theta$ as a parameter in your program for doing your own tests.

b) Detect the 10 largest maxima and write the pairs (ρ_j, θ_j) to a file for control.

c) Calculate the corresponding 10 straight lines and draw these lines back into the image space for comparison with the original image.

d) Use the angle information provided by the angle-image to reduce the point to curve transformation to a point to single-cell transformation.

e) Detect the 10 largest maxima and compare it with b) and c).

f**) Write a procedure that detects the cell of maximum count for each curve related to each contour point $\underline{x}_i$ and accumulates this cell in a second accumulator. This would be the simplification of the accumulator as described in the paper by Gerig87 (see handout). Detect again the 10 largest maxima and compare with b) and c), and with e). Which method would you prefer, and why?

g**) Can you think about a method to find corners as intersections between two lines? Would you have an idea how to implement it, maybe using the simplification concept?

** Voluntarily, Bonus Questions