McCULLOCH LABORATORY

vision group

FLASH #2

THE LINE VERIFIER GVERIFY1

A. K. Griffith

SUMMARY: A line verifier is presented which, given the co-ordinates of the end points of the hypothesized line, returns a (possibly) more accurate version of the end points, together with an estimate of the probability that there is a line in the region between the two end points given. No estimate is given as to the actual extent of the line: the increased accuracy of the returned end points lies in the accuracy of the slope and intercept of the line through them.

NOTE: Only the unstarred sections need be studied by the user. The starred (*) sections are for reference.

I. OUTLINE

To use the verifier, the user need only understand:

1) The use of the calling sequence of the function GVERIFY1;

2) The proper procedure for setting the top-level variables FPS0 and FPS1 to feature point rasters;

3) The use of the function FPCREATE to generate feature point rasters.

II. THE FUNCTION GVERIFY1

This is the verification function itself. The calling sequence is:

(GVERIFY1 X),

where X is in the form:

(X_1 Y_1 X_2 Y_2),

representing a line segment between points (X_1,Y_1) and (X_2,Y_2) expressed on Horn-Line-Finder co-ordinates. The numbers X_1, X_2, Y_1 and Y_2 may be in fixed point of floating point mode.

The value returned by the verifier is in the form:

(X_1' Y_1' X_2' Y_2' P),

where the points (X_1',Y_1') and (X_2',Y_2') are (hopefully) better versions of the end points of the line in the sense that the slope and intercept of the line defined by the latter points is quite accurace. The value P is an estimate of the probability of the existence of a line through the given points, and is in the range 0 to 1.

III. SETTING FPSO AND FPS1

Prior to any calls to GVERIFY1, the top level variables FPSO and FPS1 should be set to the values of:

(F%FEATUREPOINTS N 0), and

(F%FEATUREPOINTS N 1),

respectively (or (F%DFEATUREPOINTS N 0), (F%DFEATUREPOINTS N 1)). For information on how to use these functions, see the write-up on the F%FPOINTS package. The setting of these variables is done automatically when the FPCREATE function is executed.

IV. USING THE FUNCTION FPCREATE

The first of two alternative forms of the calling sequence of this function is:

(FPCREATE NAME1 NAME2 DEV USER).

The arguments are exactly those used in a call to UREAD in LISP: two file names, device name, user name. The file so referenced must be an intensity file created by Horn's intensity file creator for 125_{10} scan lines in each direction. This call will automatically set FPSO and FPS1 with feature point rasters obtained from the intensity information on the given file. If the file referenced does not exist, or some other error occurs, FPCREATE will return an appropriate error message.

The alternative call:

(FPCREATE VID)

reads information directly from the vidissector, and creates a pair of feature point rasters. Again, FPSO and FPS1 are set to the appropriate values.

The function FPCREATE requires that the F%FPOINTS package be in core at the time of execution.

V. AVAILABILITY

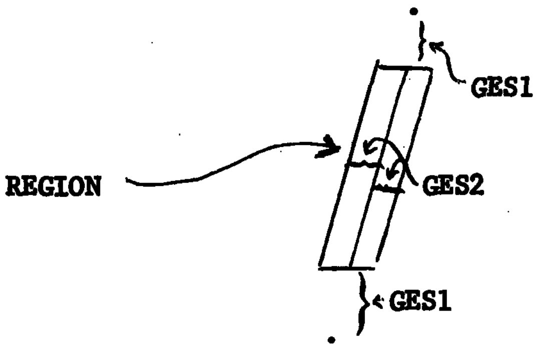
Link to or obtain the file V%> and read it into a LISP or NLISP with a LAP (q.v.). The file may be compiled for greater speed.

VI. *HORN CO-ORDINATES VS. GRIFFITH CO-ORDINATES

The image plane co-ordinate system relative to the (new) vidissector deflection field is the same in all my programs (L%LINES, F%FPOINTS, J%JOINES and GVERIFY1), but differs from that used by Horn. The latter nominally assumes that the (new) vidissector field is $1,000_{10}$ x $1,000_{10}$ units with origin in the lower left corner. The co-ordinate system used in my programs is described in detail in the L%LINES and F%FPOINTS write ups. In effect the scale is the same as that used by Horn, but my origin is (nominally) at (250.,250.) in the Horn system. More exactly, one may transform from Horn co-ordinates to Griffith co-ordinates by subtracting 258_{10} from both X and Y. In case the foregoing offset is not quite right, the user may modify the Griffith/Horn conversion which takes place in the GVERIFY1 program by modifying the values of the top level variables XOFFSET and YOFFSET from their original values of 258_{10} and 258_{10}.

VII. *HOW THE VERIFIER WORKS

For a line making angles between -45° and $+45^{\circ}$ with the vertical, the feature point raster FPS0 is examined for points lying within a region defined by the end points of the line, and by the values of the top-level variables GES1 and GES2:



The feature points lying in this region are extracted by the function GETSEGMENT. (The corresponding geometry for a line making an angle between -45° and +45° with the horizontal is as above with X and Y interchanged; the points are extracted from FPS1.)

For about 500 lines covering this region the function

$$F(L_i) = \sum_j D(L_i P_j)$$

is computed for the extracted points $\{P_j\}$, where:

$$D(L_i,P_j) = F(\text{Distance from } L_i \text{ to } P_j)$$

$$F(X) = \begin{cases} 3 & X \leq 1 \\ 2 & 1 \leq X \leq 2 \\ 1 & 2 \leq X \leq 3 \\ 0 & 3 \geq X. \end{cases}$$

The line with the maximum value of F is selected, and the corresponding maximum value, M, is compared with the value T given by:

T = (PLUS NV3 (TIMES NV2 S))

S = (MAX(ABS(*DIF X_1 X_2))(ABS(*DIF Y_1 Y_2)).

The probability, P, returned by GVERIFY1 is:

P = (*DIF 1. (MAX 0 (*QUO T (TIMES 2. M))).