McCULLOCH LABORATORY

vision group

FLASH #3

FEATURE POINT GENERATION PROGRAMS

A. K. Griffith

Summary: The programs in this set extract, from a raster of intensity values over some scene, a set of points which are adjudged to lie along the boundaries of objects in the scene. Intensities may be obtained directly from the new vidissector, or from a previously created file of intensity values.

I. CALLING SEQUENCE, VIDI INPUT

To generate a set of "feature points" from a suitably lighted scene in the field of view of the new vidissector, the user need understand only the following calling sequence:

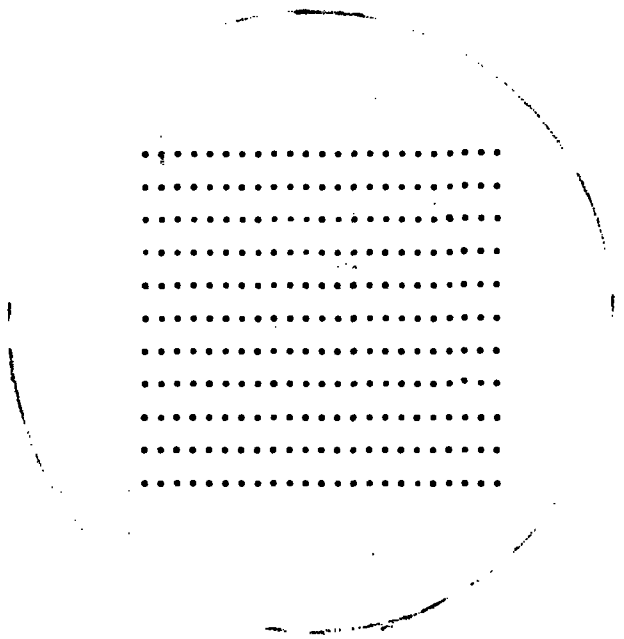
(F%FEATUREPOINTS N I),

where:

N is the number of intensity scans to be taken across the visual field; and

I is 0 or 1 depending on whether the scans are to be made horizontally or vertically respectively.

The scanning pattern covers a square centered in, and comprising a large part of, the field of the new vidissector. For N=11, and I=0, for example, the pattern would be as follows, where the circle represents the outer useful limit of the new vidissector field, and a line of dots indicates a scan of 500 intensities:



The value returned by (F%FEATUREPOINTS N I) is in the form:

$$\begin{array}{l} ((Y_0\ (PX_{01}\ PX_{02}\ \ldots..)) \\ \ (Y_1\ (PX_{11}\ PX_{12}\ \ldots..)) \\ \quad \ldots\ldots\ldots\ldots \\ \ (Y_n\ (PX_{n1}\ PX_{n2}\ \ldots..))\ ) \qquad (1.1) \end{array}$$

where the Y's range in value from 0 to approximately 500_{10}, and are the vertical (or horizontal) coordinates of the various scans; and a $PX_{i,j}$ corresponds to the j-th feature point along the i-th scan; i.e. the j-th point along the scan where the scan is adjudged to intersect with an edge in the actual scene at hand. Besides containing the distance along the scan at which the point in question occurs, the element $PX_{i,j}$ also contains information about what sort of edge is supposed to exist, and how distinct it is.

Note: To convert from the co-ordinate system of the previous paragraph to absolute vidissector coordinates, add 250. and multiply by 16. To convert to the coordinate system used by Horn, add 258. to both X and Y coordinates.

II. CALLING SEQUENCE, STORED INTENSITY INPUT

The following call behaves like the call to F%FEATUREPOINTS:

(F%DFEATUREPOINTS),

except that intensities are read from a selected input file. More specifically, generating a pair of featurepoint rasters requires the following sequence of calls:

```
(FH%UREAD name1 anme2 dev usr)
(FH%INIT)
(F%DFEATUREPOINTS)
(F%DFEATUREPOINTS).
```

The values of the latter two calls are featurepoints from horizontal-scan intensities and vertical-scan intensities respectively. The file referred to by the FH%UREAD command must be in the format as produced by the Horn/Binford intensity file creator, for 128 scans in the vertical and horizontal directions.

III. AVAILABILITY

Link to or obtain the file F% >. Read it into a LISP, NLISP or DLISP containing a LAP (q.v.). The uncompiled version takes about $5,000_{10}$ words of binary program space and about $4,000_{10}$ words of free storage. When compiled, the program requires about $6,000_{10}$ words of binary program spage.

IV. DISPLAYING A SET OF FEATURE POINTS

The LISP object returned by F%FEATUREPOINTS (or F%DFEATUREPOINTS) may be displayed on the GE console or by the line printer by executing:

```
(GE X), or
(LP X),
```

respectively, where X is the object returned by either of the two feature point generating functions. The GE console or printer may be thought of as a "window" looking at the array of feature points with the following controls over the size and location;

1) (S A)

increases the scale of magnification through the window by a factor of A. For example, if the scale is originally such that the full array of feature points exactly fills the "window", then the execution of:

```
(S 2.)
```

will cause the center 1/4 of the feature point raster to be visible.

2) (SX A), and
(SY A)

are like (S A), except that only the X and Y scales are changed respectively.

3) (MX A)

moves the point of view of the "window" to the right by an amount A. If A is negative, the point of view will be moved to the left.

4) (MY A)

is similar to (MX A), except that the point of view is moved vertically.

5) (SQUARE)

resets the magnification and position so that a feature point raster exactly fills the "window".

Note: The functions GE and LP require that the feature point rasters be in "uncompact" mode. Thus it may be necessary to use the function UNCOMPACT (cf. the U%package) on the arguments. Also, arguments presented to these functions must have the property that CAR of their successive elements, which are numbers, must be in decreasing order.

V. *EXACT NATURE OF A FEATURE POINT, COMPACT AND UNCOMPACT FORMATS

A feature point, symbolized in the formula (1.1) by $PX_{i,j}$, may have either a "compact" or an "uncompact" form. In compact form, a feature point is represented by a list of four numbers:

(A B C D).

These four quantities refer to a point in the image plane with X coordinate B (or Y co ordinate, if the scan is vertical), appearing at an edge of type C, having amplitude D. The value of A gives a measure of the probability, in some sense, that there really is an edge at the point in question.

If in compact mode, then the feature point consists of a single full word, which may be thought of as consisting of four fields whose correspondence to the four elements of the uncompact version is given as follows:

(MAX 37777 A)	B	C	(MIN 777(MAX 0(PLUS D 400)))
14 bits	9 bits	4 bits	9 bits.

The format (compact or uncompact) produced by a call to either of the two feature point generating functions is determined by a previous execution of:

(F%COMPACT X),

where:

X = NIL causes the uncompact form to be used, and

X = T causes the compact form to be used. The latter is the default option.

VI. *MODIFYING THE PARADIGM PROFILE SET

Feature points are obtained by comparing all 16 element sub-segments of the intensity scan with paradigm 16-tuple profiles, and indicating a feature point where a particularly good match (mod absolute intensity and scale) is found. These profiles are quantized to have values in the range 1 to 10. The current version of F% uses 6 different profiles, corresponding to various flavors of edge and line. If it is desired to use some subset, or some other set, they must be in the form:

$((1\ V_{1,1} \ldots .\ V_{1,16})$

....

$(n\ V_{n,1} \cdots V_{n,16}))$.

This is fed to a compiler by the call:

(F%COMPILER L),

which automatically creates a file F%LAP >of magic LAP code. This code is then to be read into the LISP-with-F%-and LAP by a standard UREAD, ctl-Q sequence. The numbers preceding a 16-tuple in the profile list given to the compiler appear as the "type numbers" (discussed in in the previous section) in the feature points returned by calls to the feature point generation programs.