MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Project
Vision Memo. No. 11C.

Memorandum MAC-M-328
September 1966

Figure Boundary Description Routines for the PDP-6 Vision Project

John White

As a step in the direction of "computer vision," several programs have been written which transform the output of a vidissector into some mathematical descriptions of the boundaries enclosing the objects in the field of view. Most of the discussion concerns the techniques used to transform a sequence of points, presumably representing a curve in the two-dimensional plane of view, into the best-fit conic-curve segment, or best-fit straight line. The resultant output of this stage is a list of such segments, one list for each boundary found.

As a step in the direction of "computer vision," several programs have been written which, applied successively, transform the point-view from the vidissector into a mathematical description of the boundaries of the objects in the field of view. An executive program will co-ordinate the actions of the various programs described below in order to find each and every object in the view. For example, several attempts may have to be made in order to be sure that a single object has been seen, using such clues as color contiguity, graininess, or texture, light intensity, etc.

The programs REGIONS1 and BNDSORT1 examine the points in the current subview of the vidissector and determine which ones belong to the object or region being sought. The executive determines which section of the vidissector to look at, the density of points to be interviewed, and the predicates that define a region. The end result is an ordered list of points consituting the boundary of the region. (Each point is a list of two LISP numbers, which are the x- and y -coordinates on the vidissector screen of the geometric point represented.) Further information on these routines may be obtained from Gerald Sussman.

POLYSEG takes the output of BNDSORT1 and a straight-line approximation to the boundary of the figure, using techniques adapted from the work of Roberts (see Lincoln Laboratory Technical Report No. 315, "Machine Perception of Three-Dimensional Solids"). The purpose of this routine is twofold: (1) to smooth out minor pieces of noise in the data, and (2) to reduce the total number of points representing the boundary. Hopefully the error tolerance will insure that a sufficient number of short straight lines are used to approximate a curved line segment on the boundary.

Perhaps more work could be put into this routine in the form of a better heuristic determination of corners as they appear in a boundary. POLYSEG returns a list of n+1 points for an n-sided polygonal approximation to the boundary of the region. The first point of the list is duplicated as the last point also, thus explaining the additional point.

ANGLES takes the output of POLYSEG and returns a list of four items

(1) Augmented polygon list [see below],

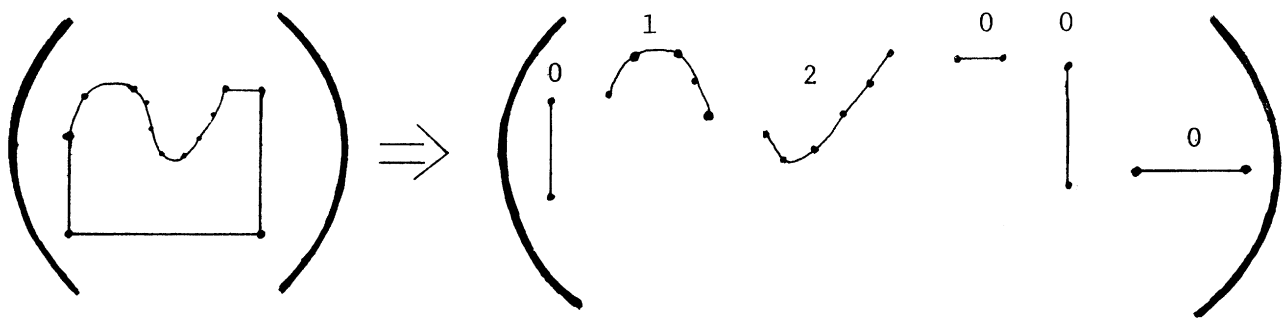
(2) Reasonable approximation to the diameter of the region described by the input polygon list,

(3) "CLCK" or "CNTK" indicating clockwise or counterclockwise ordering of the boundary points,

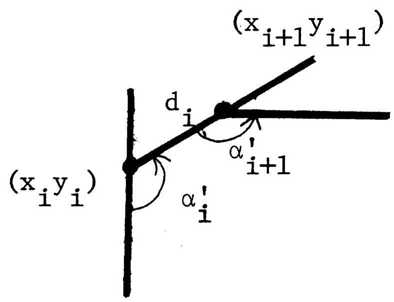
(4) Area of the approximating polygon (positive, floating point).

The original polygon list, i.e. the input list, has only the x-and y-coordinates for each vertex; the augmented list, (1) above, provides also the inside angle at each vertex and the distance to the next-in-order vertex. Thus each member $(x_i y_i)$ of the input becomes $(x_i y_i \alpha_i d_i)$ in the augmented output list.

SLPCS takes the output of ANGLES (providing the points are ordered clockwise) and attempts to segment the boundary into true straight lines or curved pieces. By "piece" we mean a segment of the boundary which, according to the following criteria, is a straight line, or a straight-line approximation to some curve without inflections. The output is then a listing of the pieces of the boundary in order, and the listing of the points of each peice is ordered the same as in the input. In addition to the points of the relevant boundary segment, the first member of each piece (which is otherwise a list of points) is a fixed point integer 0, 1, or 2 corresponding to "straight line," "curved out," and "curved in" respectively. Example:



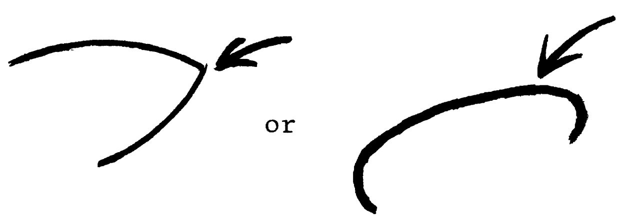
Notice that each piece is complete in itself so that its endpoints are the same points as will be found on one end of either adjoining piece. Several heuristics are applied to decide where one piece should end and another begin. Clearly an inflection point will be a breakpoint. Some thresholds are computed: LNTSH1 is such that at any point on the boundary, if the distance to the next point is greater than LNTSH1, then this segment is automatically taken to be a straight line piece. If the said distnace is less than LNTSH2 then it is assumed that this will not be a straight line piece, but rather part of some curve piece. The other cases are resolved by some additional formulae: consider point i with associated distance d_i and inside angle α_i' (outside angle $\alpha_i = \pi - \alpha_i'$).



We will consider the segment (x_iy_i) - $(x_{i+1}y_{i+1})$ to be a straight line piece if d_i > LNTSH1 or if

$$[d_i > \mathrm{LNTSH2} \wedge (\alpha_{i+1} \cdot d_i > \mathrm{CRVF} \vee \alpha_i^2 \cdot d_i > \mathrm{CRVB})].$$

Any curved piece is extended until an inflection point or straight line is reached. From the above formula, it is clear that straight lines are usually found by the local property of "sharp corner". The formula involves the two endpoints of a straight line in slightly different ways because it was found to be efficacious in discerning certain odd types of corners while using a one-way scan of the points from ANGLES. For example



Experiments were carried out to determine suitable values for the above thresholds, with the following results:

LNTSH1 = 0.25 * DIAMETER

LNTSH2 = 0.03 * DIAMETER

CRVB = 0.022 * DIAMETER

CRVF = 0.02068 * DIAMETER

The curved pieces output by SLPCS may be represented as some segment of a conic (general second degree curve) since there are no inflection points in such curves (some pathological cases do arise however, such as egg--shaped closed curves and spiral sections, but for the moment these are ignored, and probably will not be parts of viewed objects anyway). BSTCRV accepts any piece from the output of SLPCS, including straight lines which it leaves unaltered, and returns a list of eight members representing a conic-curve segment. That is, for straight lines

$$(0\ (x_1 y_1)\quad (x_2 y_2))\ \xrightarrow{\text{BSTCRV}}\ (0\ (x_1 y_1)\ (x_2 y_2))$$

and for curves which can be approximated by a circular or elliptical arc

$$(\text{IND}\ (x_1 y_1)\ \ldots\ (x_2 y_2))\ \xrightarrow{\text{BSTCRV}}\ (\text{IND 'ELLPSE'}\ P_0\ \ell_1\ \ell_2\ \alpha\ P_1 \beta)$$

where

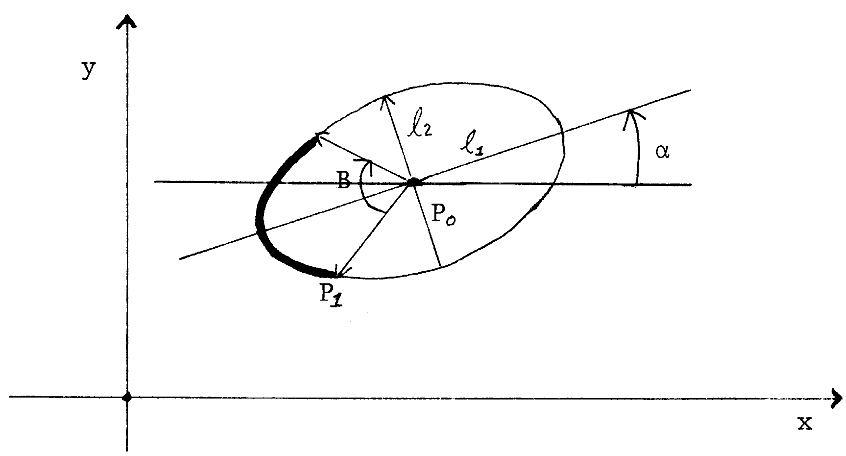
IND	is	1 or 2 for curved out or in
'ELLPSE'	is	the type of curve, as opposed to hyberbolic
P_0	is	a point representing the center of the ellipse
ℓ_1	is	a LISP number for the length of the principal semi-axis
ℓ_2	is	for the secondary semi-axis

α is an angle $-\pi/2 < \alpha \leq \pi/2$ measuring the tilt of the principal axis from the horizontal, about the center of the ellipse.

P_1, β specify the segment of the curve in the following manner: "beginning at point P_1 and using P_0 as center point, take B radians about P_0, in the clockwise direction if IND = 1 otherwise in the counterclockwise direction. This is the relevant segment.

Clearly for a circle $\ell_1 = \ell_2$, and for any complete circle or ellipse, $\beta = 2\pi$ and P_1 is any point on the curve.

<u>Example</u>



For second-degree curves which are not ellipses, i.e. hyperbolae or parabolae we simply return the general-equation coefficients and the first and last point on the arc.

$(IND\ (x_1y_1)\ \ldots\ (x_ny_n)) \xrightarrow{BSTCRV} (IND\ \text{'HYPER'}\ (A\ B)\ C\ D\ E\ (x_1y_1)\ (x_ny_n))$

where the best fit curve is

$Ax^2 + Bxy + Cy^2 + Dx + Ey = 1$

Objects which have no inflection points or straight line pieces are returned by SLPCES as all one piece. As mentioned earlier, egg-shaped objects fall into this category, but they cannot be well-fit to a conic. Except for a few nemeses, BSTCRV can be applied to the output of SLPCS by means of MAPCAR -- NIL being the [redacted] output from BSTCRV.

Appendix I

Atoms Used as Free Variables

Throughout the Vision Programs

PI, TWOPI — Set to π and 2π by a top level call to SETQ, these atoms provide convenient reference to some familiar numerical values.

CTOL, ETOL — BSTCRV requires the mean·square error in an approximation to be less than some tolerance. Since circles are intrinsically more simple than general conics, a circular fit is first examined: if the computed circle has center (a,b) and radius v, then it is accepted iff

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{(x_i-a)^2}{r^2}+\frac{(y_i-b)^2}{r^2}-\left(1+\frac{a^2+b^2}{r^2}\right)\right]^2 < \text{CTOL}$$

If the circle is not accepted, a general second-degree curve fit is tried and accepted iff

$$\frac{1}{n}\sum_{i=1}^{n}\left[Ax_i^2+Bx_iy_i+Cy_i^2+Dx_i+Ey_i-1\right]^2 < \text{ETOL}$$

See Appendix II for the formulae by which the best fit curve coefficients are determined.

Suitable values for these tolerances are ETOL = .01
CTOL = .0015

but some applications may want to vary them.

TOL, N1, N2: POLYSEG determines a straight line approximation to a curve by initially picking up N1 points and extending by N2 points at a time until the normalized mean-square error of the best-fit straight line to these points is greater than TOL. "BEST-FIT" means least-squares best fit in which

$$\frac{1}{n}\sum_{i=1}^{n}(ax_i + by_i - c)^2$$ is minimized by appropriate choice of a,b, and c.

More experimentation should be done to find suitable values for these parameters -- a good first guess might be TOL = .05
N1 = 4
N2 = 4

Appendix II

BEST FIT CIRCLE

For a data set $\{(x_1 y_1) \ldots (x_n y_n)\}$ the best fit circle, with equation $ax^2 + bx + ay^2 + cy = 1.0$, is the one which minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \left[ax_i^2 + bx_i + ay_i^2 + cy_i - 1.0 \right]^2 .$$

Differentiating the error expression above with respect to a, b, and c, leads to three simultaneous linear equations, which then can easily be solved. The center of the circle is at

$(- \frac{b}{2a}, - \frac{c}{2a})$ and it has radius $\sqrt{\frac{1}{a} + A^2 + B^2}$

BEST FIT CONIC

As above, we solve five simultanious linear equations in order to minimize

$$\frac{1}{n} \sum [ax^2 + bxy + cy^2 + dx + ey - 1.0]^2 .$$

Actually, a parametric form of the second-degree equation would be more revealing,

$$ax^2 + bxy + cy^2 + dx + ey = f.$$

One must place some additional constraint, however, since there are only five degrees of freedom in a conic curve, but the parametric equation has six unknowns. Imposing

$$\sqrt{a^2 + b^2 + c^2 + d^2 + e^2 + f^2} = 1$$

looks plausible, but the resulting simultaneous equations are horribly non-linear, so the much simpler condition f = 1 has been chosen. Cases in which this latter assumption leads to numerical nonsense rarely occur in practice.

DETERMINATION OF ELLIPSE PARAMETERS

If $b^2 - 4ac < 0$, then the curve represented by

$$ax^2 + bxy + cy^2 + dx + ey = 1.0$$

is an ellipse. Otherwise it is a parabola or hyperbola. If the whole plane is rotated about the origin through an angle of $-\alpha$ radians, the axes of the ellipse will be parallel to the x- and y-axes. Thus consider the curve

$$Ax^2 + Cy^2 + Dx + Ey = 1.0$$

where

$$\alpha = \text{sign (b)} \cdot \text{sign(a-c)} \cdot \frac{1}{2} \tan^{-1}\left|\frac{b}{a-c}\right| , \quad \text{sign}(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ -1 \text{ if } x < 0 \end{cases}$$

$$A = a \cos^2\alpha + b \cos\alpha \sin\alpha + c \sin^2\alpha$$

$$B = b(\cos^2\alpha - \sin^2\alpha) + 2(c-a) \sin\alpha \cos\alpha = 0$$

$$C = a \cdot \sin^2\alpha - b \cdot \cos\alpha \cdot \sin\alpha + c \cdot \cos^2\alpha$$

$$D = d \cos\alpha + e \sin\alpha$$

$$E = -d \sin\alpha + e \cos\alpha$$

This represents an ellipse with center at $(-\frac{D}{2A}, -\frac{E}{2C})$, whose horizontal semi-axis length is

$$\ell_1 = \frac{\sqrt{4AC^2 + D^2C^2 + ACE^2}}{2AC}$$

and whose vertical semi-axis length is $\ell_2 = \frac{A}{C} \cdot \ell_1$.

Thus the original ellipse has lengths ℓ_1 and ℓ_2, tilt α, and center

$$\left(\frac{1}{2} \left[-\frac{DD}{A} \cos\alpha + \frac{E}{C} \sin\alpha \right], \frac{1}{2} \left[-\frac{D}{A} \sin\alpha - \frac{E}{C} \cos\alpha \right] \right).$$

Reference: Thomas, G.B. Jr. Calculus and Analytic Geometry, Addison-Wesley (1960). Chapter 9.

CS-TR Scanning Project
Document Control Form

Date : 12/14/95

Report # AIM-110

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:

Document Information

Number of pages: 13 (17-images)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☒ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:

Check each if included with document:

☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other:

Page Data:

Blank Pages (by page number):

Photographs/Tonal Material (by page number):

Other (note description/page number):

Description : Page Number:

(A) IMAGE MAP! (1-13) UN#'ED TITLE PAGE, 2-13
(14-17) SCANCONTROL, TRGT'S (3)
(B) CUT&PASTE MATERIAL ON PAGES 2-3, 5-9, 12
(C) PAGES 3, 5 & 6 ARE STAIN BECAUSE OF PASTE.

Scanning Agent Signoff:
Date Received: 12/14/95 Date Scanned: 1/23/96 Date Returned: 1/25/96

Scanning Agent Signature: Michael W. Cook

Rev 9/94 DS/LCS Document Control Form cstrform.vsd

ORG