Fitting Ellipses and General Second-Order Curves

Gerald J. Agin
July 31, 1981

The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pa. 15213

Preparation of this paper was supported by Westinghouse Corporation under a grant to the Robotics Institute at Carnegie-Mellon University.

Abstract

Many methods exist for fitting ellipses and other second-order curves to sets of points on the plane. Different methods use different measures for the goodness of fit of a given curve to a set of points. The method most frequently used, minimization based on the general quadratic form, has serious deficiencies. Two alternative methods are proposed: the first, based on an error measure divided by its average gradient, uses an eigenvalue solution; the second is based on an error measure divided by individual gradients, and requires hill climbing for its solution.

As a corollary, a new method for fitting straight lines to data points on the plane is presented.

Introduction

This paper discusses the following problem: Given some set of data points on the plane, how should we fit an ellipse to these points? In more precise terms, let curves be represented by some equation (7(JC,J)=0. We restrict *G(x,y)* to be a polynomial in *x* and *y* of degree not greater than 2. The curves generated by such a function are the conic sections: ellipses, hyperbolas, and parabolas. In the special case where *G(x,y)* is of degree 1, the curve represented is a straight line. Now, given a set of data pairs *{(x?y?* ż=l,.../i), what is the function *G(x,y)* such that the curve described by the equation best describes or fits the data? The answer to this question depends upon how we define "best"

The primary motivation for studying this problem is to deal with systems that use light stripes to measure depth information [Agin 76] [Shirai] [Popplestone]. When a plane of light cuts a cylindrical surface it generates a half ellipse in the plane of the illumination. When this ellipse is viewed in perspective it gives rise to another partial ellipse in the image plane. The incomplete nature of this curve segment makes it difficult to measure its intrinsic shape.

A similar problem often arises in scene analysis [Render] [Tsuji]. A circle viewed in perspective generates an ellipse on the image plane. If some scene-understanding procedure can identify the points that lie on the perimeter of the ellipse, these points may be used as the data points in a curve-fitting process to identifying the dimensions of the ellipse. The relative lengths of the major and minor axes and the orientation of these axes will then be sufficient to determine the plane of the ellipse relative to the camera.

Fitting ellipses and other second-order curves to data points can be useful in interpreting physical or statistical experiments. For example, particles in bubble-chamber photographs may follow elliptical paths, the dimensions of which must be inferred.

It is easy to see how a fitter of ellipses would be useful in an interactive graphics or a computer-aided drawing package: Le^ the user could indicate a rough approximation to the ellipse or circle he wants, and the system could infer the best-fitting approximation. This kind of capability is currently handled by fitting with splines [Smith] [Baudelaire].

It is important to distinguish among the *extraction* of points that may represent the boundary of an ellipse; the *segmentation* of collections of points into distinct curves; and the *fitting* of these points once they have been extracted. This paper does not purport to describe how to determine which points do or do not belong to any ellipse or ellipse segment Curve fitting *can* be of use in segmentation and extraction to evaluate the reasonableness of a f^vea hypothesis; however this discussion is limited to methods for determining the equation of the curve that best fits a given set of data points.

Representing Second-Order Curves

An ellipse in "standard position", such as the one in Figure 1, may be represented by the equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{1}$$

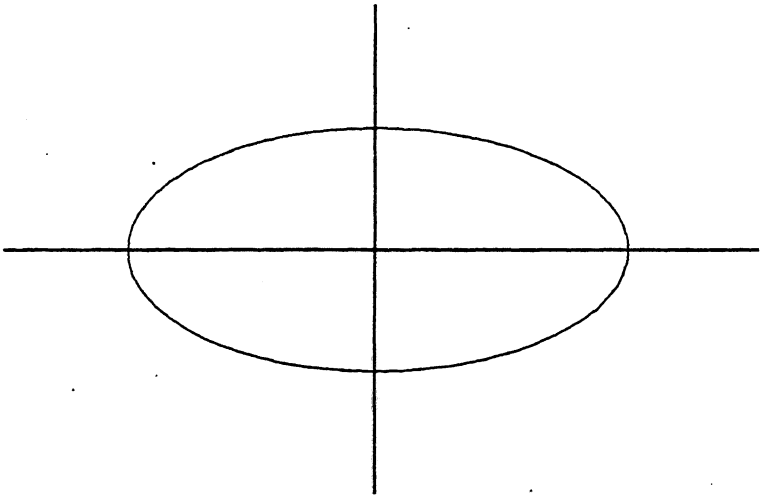

Figure 1: An Ellipse in Standard Position

Such an ellipse has its center at the origin of coordinates and its principal axes parallel to the coordinate axes. If parameter a is greater than parameter b, then a represents the length of the semi-major axis and b represents the length of the semi-minor axis. The *eccentricity* (e) of the ellipse is defined by the formula

$$e = \sqrt{1 - \frac{b^2}{a^2}},$$

where e must be positive, and between zero and 1. If $a=b$, then equation 1 represents a circle, and e is zero. If $a<b$ then b represents the semi-major axis and a the semi-minor, and e is defined as

$$e = \sqrt{1 - \frac{a^2}{b^2}}.$$

A shift of coordinates allows us to represent an ellipse centered on a point other than the origin, say (h,k), as in Figure 2. If we let

$$x' = x - h \tag{2}$$
$$\text{and} \quad y' = y - k$$

then the equation of the ellipse of Figure 2 is

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1, \tag{3}$$

or,

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 \tag{4}$$

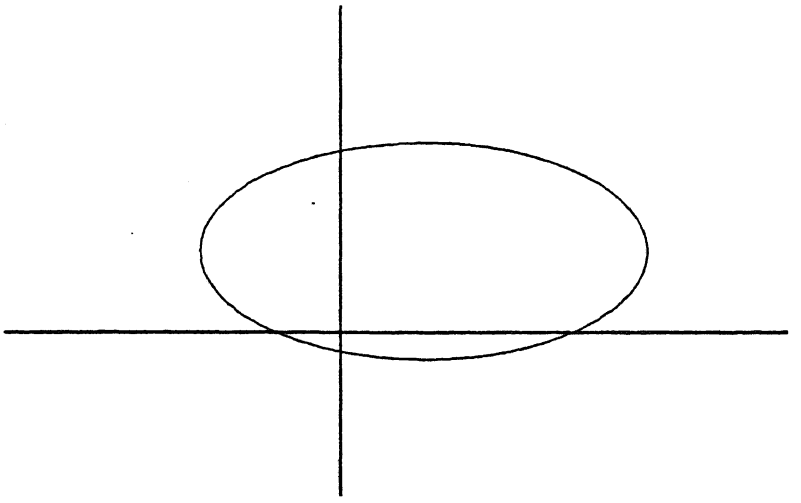

Figure 2: An Ellipse off the Origin of Coordinates

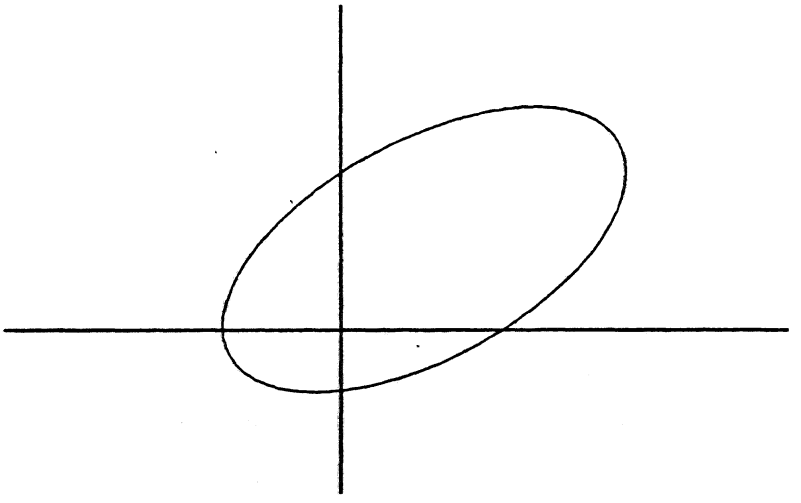

Figure 3: An Ellipse Rotated and Moved

A rotation of the ellipse, as in Figure 3, can be accounted for by the transformation

$$x'' = x'\cos\theta + y'\sin\theta$$
and $$y'' = -x'\sin\theta + y'\cos\theta.$$

These transformations can be substituted directly into the equation for an ellipse, but we prefer the implicit form:

$$\frac{x''^2}{a^2} + \frac{y''^2}{b^2} = 1 \tag{5}$$

where $x'' = (x-h)\cos\theta + (y-k)\sin\theta$
and $y'' = -(x-h)\sin\theta + (y-k)\cos\theta$.

Equation 5 can represent any ellipse in any orientation. A total of five parameters are involved: *a* and *b* represent the dimensions of the ellipse, *h* and *k* represent its center, and *0* represents its rotation.

The equation of a hyperbola in standard position is similar to that of an ellipse, but with a sign change:

$$\frac{x^2}{a^2} - \frac{}{b^2} = 1$$

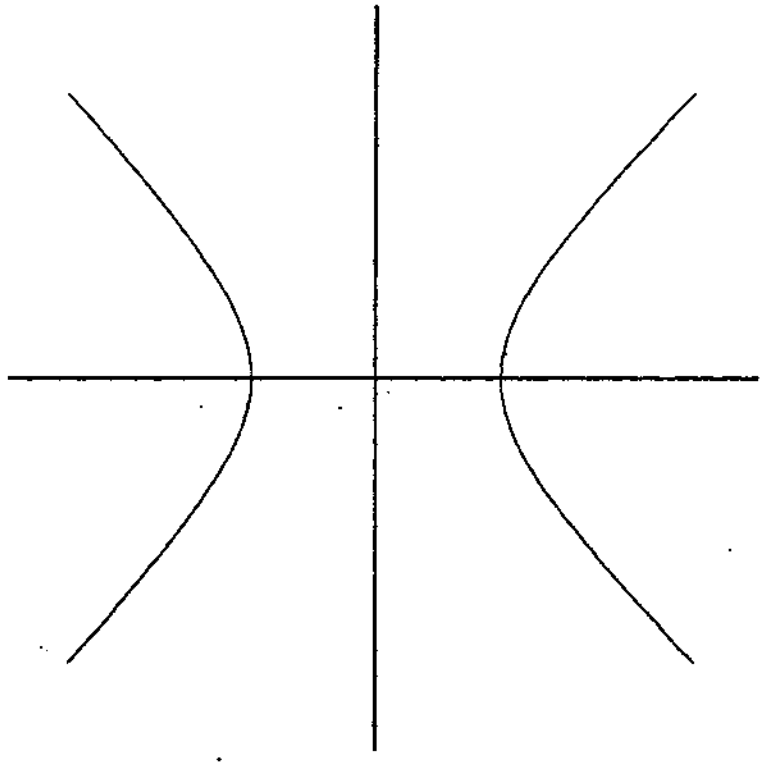

Figure 4: A Hyperbola in Standard Position

A hyperbola is shown in Figure 4. Its eccentricity is given by

$$e = \sqrt{1 + \frac{b^2}{a^2}}\ .$$

The center of the hyperbola can be moved and *ks* axes rotated by transforms similar to those we used fir ellipses. We can represent ellipses and hyperbolas by the same equation or set of equations if-we let eccentricity into (he equation, iw central conic (ellipse *m* hyperbola) can be represented as:

$$\frac{x^{m2}}{a^2} + \frac{}{a^2} \quad \frac{}{)} = 1 \qquad (6)$$

where $x^m = (x - h)\cos \& \ 4 - 0 - k)\sin 6$

and $f = —(x \sim h)\dot{m}n\$ \ 4 - (y \sim k)\cos \theta$

II should be soled here that an ellipse can also be represented parantetrically. For an ellipse in standanl

orientation, points on its perimeter are given by

$$x = h + a\cos\phi \qquad (7)$$
$$y = k + b\sin\phi,$$

where ϕ varies between 0 and 2π. The rotation θ of the ellipse can be taken care of by rewriting equation 7 as follows:

$$x = h + a\cos\phi\cos\theta - b\sin\phi\sin\theta$$
$$y = k + a\cos\phi\sin\theta + b\sin\phi\cos\theta$$

A hyperbola may also be represented parametrically, using hyperbolic functions. Points on a hyperbola in standard orientation with its center at (h,k) are given by

$$x = h \pm a\cosh\zeta$$
$$y = k \pm b\sinh\zeta.$$

The value of ζ may vary from zero to an arbitrary upper limit. The various permutations of the $\pm$ signs give rise to the four branches of the hyperbola.

A parabola is actually a conic section with eccentricity 1, but if we try to represent it in the form of Equation 6 a division by zero results. It is better to represent the parabola by the equation

$$y = a x^2$$

A shift of origin and a rotation give the form:

$$y'' = a x''^2 \qquad (8)$$
where $x'' = (x-h)\cos\theta + (y-k)\sin\theta$
and $y'' = -(x-h)\sin\theta + (y-k)\cos\theta$

Given any parameters of size, position, and orientation, Equation 6 or Equation 8 can be rewritten in the form

$$G(x,y) = A x^2 + B xy + C y^2 + D x + E y + F = 0 \qquad (9)$$

It may be shown that all conics may be represented in the form of Equation 9.

Purcell [Purcell, p. 130] shows that Equation 9 represents a hyperbola if the *indicator*, $B^2 - 4AC$ is positive, a parabola if it is zero, or an ellipse if it is negative.

Furthermore, the parameters of Equations 6 or 8 may be recovered by the following procedure: Apply a rotation θ in which $\theta = 45$ degrees if $A = C$ and

$$\tan 2\theta = \frac{B}{A - C}$$

if $A \neq C$. This transforms Equation 9 into an equivalent form in which B (the coefficient of the xy term) is zero. It is then a straightforward matter to extract the other four parameters.

Minimization and Approximation Theory

Approximation theory is a mathematical discipline that addresses curve fitting [Rivlin]. Usually, a set of n data points are specified as pairs of the form $\{ (x_i,y_i), i=1, ..., n \}$, where x is regarded as an independent variable and y_i represents values measured at n particular values of x. Let the symbol v denote the set of given data pairs. Let V be the set of all functions defined on $\{x_i, i=1, ... ,n\}$. V is thus an n-dimensional linear space, and $v \in V$.

Admissible solutions to curve fitting problems are usually represented in the form $y=f(x)$. The set of all admissible solutions constitutes a subspace W of V, whose dimensionality corresponds to the number parameters used to characterize f. For example, the set of all quadratic functions of one variable constitutes a space of dimensionality three. Given some $w \in W$ we need a measure of the difference between w and v, which we denote as $|w-v|$, the norm of $w-v$. The norm may be defined in the Euclidean manner as the square root of the sum of the squares of $w-v$, where summation is over all values of x for which both $w(x)$ and $v(x)$ are defined. Another norm in frequent use is the maximum of all elements of $w-v$, again over all points where both functions are defined.

A central theorem of approximation theory states that there exists some w^* such that

$$|w^*-v| \leq |w-v|$$

for all $w \in W$. When we use the Euclidean norm, we say the minimizing w^* is the best approximation in the *least-squares* sense. If the norm is the maximum of all elements of $w-v$, the minimizing w^* is referred to as the best *uniform* approximation.

The paradigm outlined above can be generalized to several dimensions. For example, given triples of the form $\{ x_i, y_i, z_i, i=1, ..., n \}$ and a space of functions $w(x,y)$ we may find w^* that minimizes (in the appropriate sense) the difference between $w(x_i,y_i)$ and z_i. But however many dimensions there are, the basic assumption remains: that w is a single valued function of one or more independent variables.

It is difficult to represent an ellipse as a single-valued function. Therefore, the "difference" between a data point and an ellipse is not uniquely and unambiguously defined. Intuitively, the difference should represent the perpendicular distance from the point to the curve. If an ellipse were represented in the form $y=f(x)$, then f would be multivalued over some range of x, and have no value elsewhere. Usually ellipses are represented *implicitly* by equations of the form $g(x,y)=0$. We might choose a norm that estimates the magnitude of g itself, (i.e., it measures the difference between g and zero,) and search for a g^* that minimizes that norm. But the "classical" techniques of approximation theory are no longer applicable, so we must develop other techniques.

Choosing an Error Function

The basic paradigm for ellipse fitting is as follows: First, choose a method of estimating the "error" of a point with respect to any given second-order curve; second, choose a method of calculating an aggregate error from all the individual errors; third, systematically search for the ellipse that minimizes the aggregate error. The choice of an error measure and an aggregating rule affects not only the solution, but also the computational effort needed to obtain the solution.

It should be noted that any five arbitrary points on the plane are sufficient to specify a second-order curve. As long as no three of the five points are coplanar, there exists a unique second-order curve that passes exactly through each of the five points. An algebraic procedure exists for finding this curve [Bolles]. . More sophisticated methods become necessary only when there are more than five data points to be fit.

If all the data points lie on, or very close to, a mathematically perfect curve, then almost any method for fitting ellipses will give acceptable results. In practice, problems usually arise when the data become noisy and dispersed. Very eccentric ellipses are harder to fit than nearly circular ones. Cases where only a portion of the complete curve is represented by data points generally create problems: the less complete the perimeter the greater the difficulty of estimating the curve to represent it.

For the rest of this discussion, we will consider only a Euclidean norm. In other words, we are restricting our attention to least-squares methods. This reflects a desire to let the solution represent an "average" of all the data, rather than being influenced primarily by the outlying points, as would be the case if we used a uniform norm.

Using the General Quadratic Form

One possible choice of an error function is the general quadratic form of a second-order curve as given in Equation 9. We must avoid the trivial solution $A = B = C = D = E = F = 0$, so we arbitrarily assign $F = 1$. This gives

$$G(x,y) = A\,x^2 + B\,xy + C\,y^2 + D\,x + E\,y + 1 = 0. \tag{10}$$

Given a data point (x_i,y_i), we let the pointwise error ξ_i be given by

$$\xi_i = G(x_i,y_i) = A\,x_i^2 + B\,x_i\,y_i + C\,y_i^2 + D\,x_i + E\,y_i + 1.$$

The aggregate error is given by

$$\begin{aligned} \Xi &= \Sigma\,\xi_i^2 \\ &= \Sigma\,(A\,x_i^2 + B\,x_i\,y_i + C\,y_i^2 + D\,x_i + E\,y_i + 1)^2 \end{aligned} \tag{11}$$

Obtaining the partial derivatives of Equation 11 with respect to A, B, C, D, and E, and setting these to zero,

we obtain the following system of equations:

$$\begin{aligned}
&A\ \Sigma x^4 + B\ \Sigma x^3 y + C\ \Sigma x^2 y^2 + D\ \Sigma x^3 + E\ \Sigma x^2 y + \Sigma x^2 = 0 \\
&A\ \Sigma x^3 y + B\ \Sigma x^2 y^2 + C\ \Sigma x y^3 + D\ \Sigma x^2 y + E\ \Sigma x y^2 + \Sigma x y = 0 \\
&A\ \Sigma x^2 y^2 + B\ \Sigma x y^3 + C\ \Sigma y^4 + D\ \Sigma x y^2 + E\ \Sigma y^3 + \Sigma y^2 = 0 \\
&A\ \Sigma x^3 + B\ \Sigma x^2 y + C\ \Sigma x y^2 + D\ \Sigma x^2 + E\ \Sigma x y + \Sigma x = 0 \\
&A\ \Sigma x^2 y + B\ \Sigma x y^2 + C\ \Sigma y^3 + D\ \Sigma x y + E\ \Sigma y^2 + \Sigma y = 0
\end{aligned} \tag{12}$$

The solution to these equations represents the the ellipse that minimizes the error function given in Equation 11.

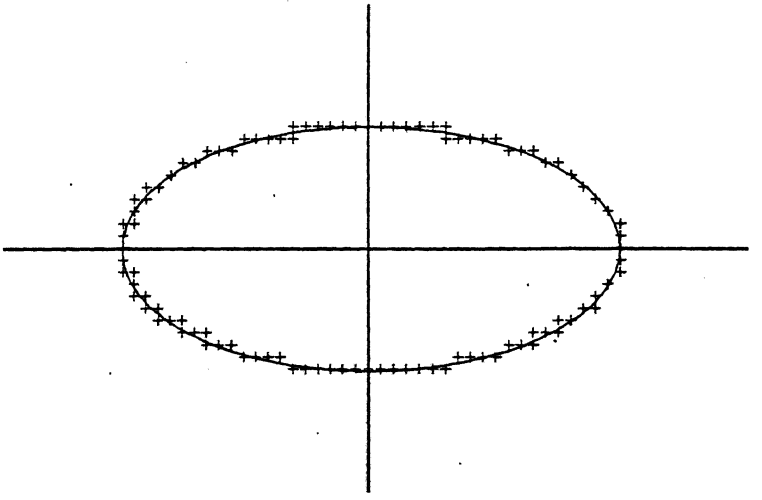

Figure 5: Fit Obtained by Minimizing Equation 11

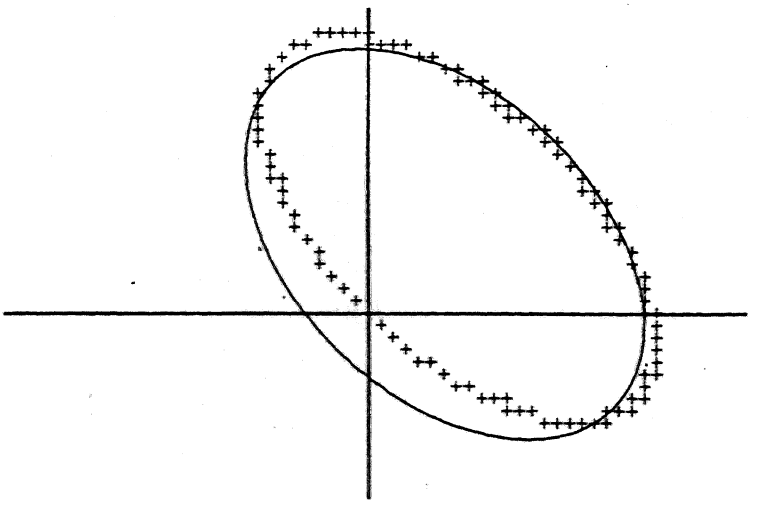

Figure 6: Fit Obtained by Minimizing Equation 11

Figure 5 shows a set of computer-generated data points and the curve generated by this method to fit it.

The method appears to work adequately in this c But Figure 6 shows another case, where the minimizing ellipse clearly misses the data points near the origin of coordinates. What we are seeing is the result of a poor choice of error function. When we went from the ellipse representation of Equation 9 to that of Equation 10 by fixing *F* to be 1, we allowed the representation to become degenerate; we lost the ability to represent an ellipse that passes through the origin. An ellipse as represented by Equation 10 that passes close to the origin must have large coefficients *A, B,* C, *D,* and *E*\ hence the error measure *Z* of Equation 11 will be large. Therefore, minimizing *Z* implies keeping the curve away from the origin.

A requirement of a useful curve fitting method is that it should be independent of scaling, translation, or rotation of the data points. That is, the choice of a coordinate system should not affect the solution curve; except, of course, that the solution curve should be scaled, moved, or rotated along with the data points.

The Average Gradient Constraint

Ideally, the error function we choose to minimize should be related to the distance from a point to the curve. Suppose we were to choose some primitive error measure such as the *G(x,y)* given in Equation 9. *G* is zero along the curve, and its magnitude increases when we measure *G* at points farther and farther from the curve. For a point in a small neighborhood the curve, *G* is proportional to the perpendicular distance from the point to the curve. The constant of proportionality is the reciprocal of the magnitude of the gradient of G.

We will choose a constraint on the coefficients of Equation 9 such that the average gradient is unity. Then the resulting error function will be directly related to the distances from points to curves.

A shift in notation will make the following mathematics easier. Define the vectors X and V to be

$$\mathbf{X} = \begin{bmatrix} x^2 \\ 7 \\ x \\ y \\ 1 \end{bmatrix} \text{ and } \quad V = \begin{bmatrix} A \\ B \\ C \\ D \\ E \\ F \end{bmatrix}$$

Then we may rewrite Equation 9 as

$$\mathfrak{L}(jy) = V^T X = X^T V.$$

Using the Euclidean norm, our aggregate error $\bar{A}$ is given by

$$\Xi = \Sigma \xi_i^2 = \Sigma G^2 = \Sigma (V^T X X^T V) = V^T \Sigma (X X^T) V = V^T P V. \qquad (13)$$

P = 2 X X^T is a matrix of sums of powers of JC and *y,* whose first five rows and columns are, in fact, the coefficients of *A B, C, D,* and *E* in Equation 12 and whose last column provides the constant terms.

Be magnitude of the gradient of *G*, |v *G*\ may be determined from the partial derivatives of *G* w to jtandj.

3(7 _ V^T 3X = y^T x

$$\frac{\partial G}{\partial u} = V^T \frac{\partial X}{\partial \lambda} = V^T X_{\lambda}$$

where

$$X_x = \begin{bmatrix} 2x \\ y \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad X_y = \begin{bmatrix} 0 \\ x \\ 2y \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$(\nabla G)^2 = \left(\frac{\partial G}{dx}\right)^2 + \left(\frac{\partial G}{dy}\right)^2 = V^T (X_x X_x^T + X_y X_y^T)\, V$$

$$\Sigma (\nabla G)^2 = V^T\, 2C X_x X_x^T + X_y X_y^T)\, V$$

$$= V^T Q V$$

Q= Ss;X_X_^T+XX_^T)is another matrix summed from powers of *x* and *y*. The mean-square gradi m^«rr«iataBdinpofaa{(x_^), 1=l_T.»,n}*kZ(vG)²/ n*. Requiring this "average gradient ma is equivalent to specifying

$$V^T Q V = a.$$

We wish It find the vector ¥ that minimizes the matrix product V^T P V, under the constraint that * & Us well ticwra {Cwr«it} that at the cmstråaed minimum there exists some LaGrange *miii* such that

P V * X Q V

This «pafi§t would be easy to solve by nonnadl €%envalue methods were it not for the fact thai ntplai mimx_f as4 P » searfy sopltr, (It ^wns that the closer the data points approximate a crate ttit closer P jpf imfw singularity.) The appendii gives a method for solving Equation 15 that *fk* eigenvs&\6G% |f_r i*l ^ i | comspondtng 10 five eigenvectors {V_., i= 1,,..,5}.

li? dcwm«8W the aggregate error, *Hqmtkn IX* we may *urn* Equations 15 and 14 to produce the rest*

$$\Sigma = V^T P V = \lambda V^T Q V = \lambda a.$$

Then we know that the coefficients of the quadratic function giving the minimum aggregate error under the given constraint are given by the eigenvector corresponding to the smallest eigenvalue.

Solutions to the curve fitting problem are invariant with translation, rotation, and scaling of the input data. A proof of this is presented in Appendix B.

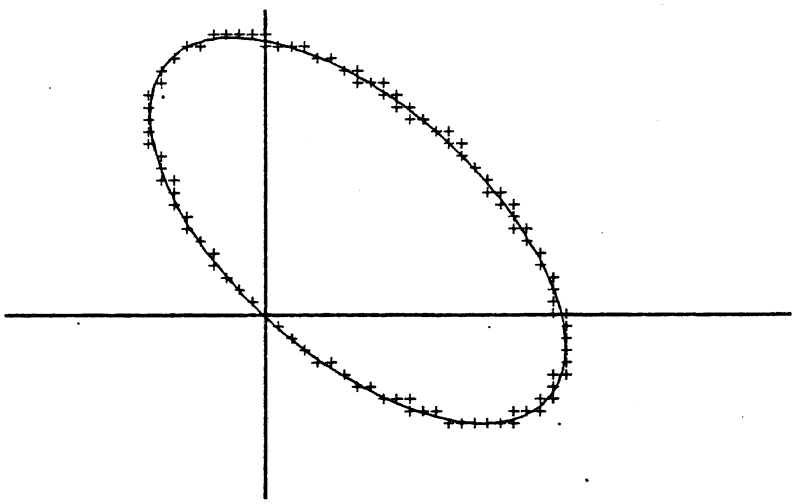

Figure 7: Curve Fitting with Average Gradient Constraint

Figure 7 show the same data points that were used for Figure 6 fit using the "eigenvalue" method described above. Comparing figures 6 and 7, shows that the new method gives superior results.

Some Difficulties

The problem of curve fitting gets worse when the points to be fit represent only part of an ellipse. Noise and digitization error accentuate the problem.

Figures 8 through 10 show increasingly difficult cases. The data points for Figure 8 are a subset of those used to generate Figures 6 and 7. There is a noticeable flattening of the solution curve, but not so much that if we had no knowledge of how the points were generated we would say the fit was "wrong." The misfit in Figure 9 is more apparent. The same ideal ellipse as before was used to generate the points, but a "fattening" of the data points has been simulated. Figure 10 represents an extreme case. The data points were not generated theoretically, but are from an actual light-stripe experiment [Agin 72].

What we are seeing is a systematic tendency for the solutions to flatten, becoming elongated ellipses parallel to the general linear trend of the data points. The tendency arises from the fact that, all other things

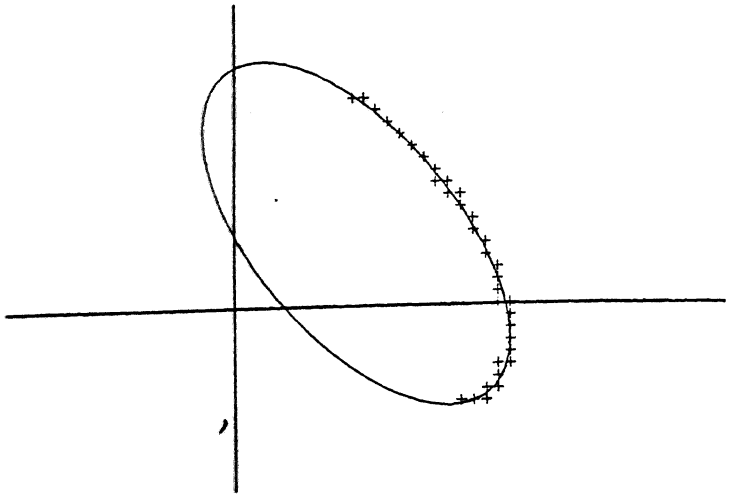

Figure 8: Curve Fit to a Short Segment

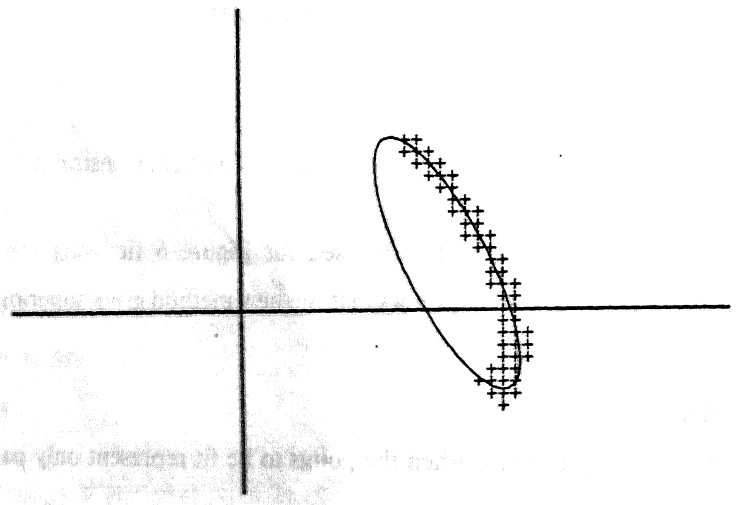

Figure 9: Curve Fit to a Short, Fattened Segment

being equal, the error of a scatter of points about a curve $G(x,y) = 0$ depends on the second derivati error function G. That is, a function whose gradient varies rapidly tends to "fit" better, in a normaliz squares sense, than a function with a constant gradient. Flattened ellipses and hyperbolas are chara by a high second derivative of their defining function. The curve fitting solution chooses these s curves over the more intuitive curves we would prefer.

The problem is not limited to fitting with the average gradient constraint. Lyle Smith [Smith] n same phenomenon using the general quadratic form, i.e., minimizing Equation 11.

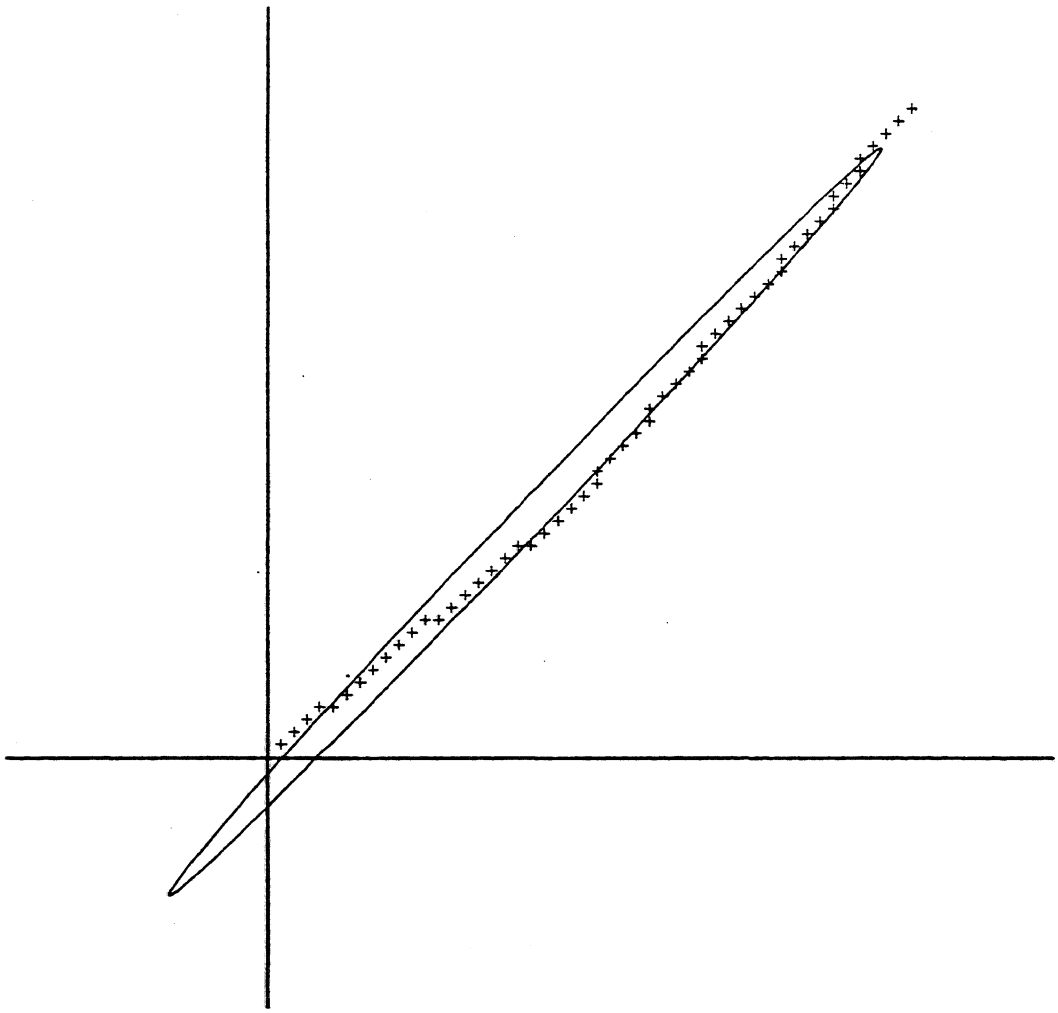

Figure 10: Curve Fit to a Gently-Curving Segment

It is tempting to try some method that would keep the general idea of constraining the average gradient, for example by computing that average over the entire curve instead of over all the data points. This would amount to a constraint on the coefficients A through F of Equation 9 independent of the data points. A little thought will show that this approach will not work at all. The RMS error can be made arbitrarily small by choosing a very large and very elongated ellipse with a gradient magnitude near unity along most of its length, but a vanishingly small gradient magnitude in the vicinity of the data points.

Corve Fitting by Hill Climbing

The bc^i measure of the goodness of fit of a point or set of points to a given mathematical curve $G(x,y)$ Is provided by measuring the perpendicular distance from each point to the curve. A reason* approximation to that distance may be had by dividing the error function Gfa,j-,) by the magnitude of gradient of (7 measured at (x_i, y_i). With such a definition, aggregate error S is given by

$$= = 1\left(\frac{}{|\nabla G(x,y)|}\right)^2$$

$$= \sum \frac{V^T X X^T V}{V^T (X_x X_x^T + X_y X_y^T) V},$$

where V, X, X_x, and X_y arc the same as in the previous settioa*

The peiRt-by-point division males It impossible to move the summation sign inside the matrix produc ic did a the previous section. Minimizing Equation 16 will require a hill-climbing approach. We ir postulate a coefficient vector V, use it to evaluate 3f, then choose another V to see whether or not It Impre tie error S* esc

Even though there arc six elements in the vector V, there arc really only fi\c independent paramc necessary f§ specif)[1] an ellipse* The hill climbing algorithm will manipulate these five. We are free to spa these parameters in any way we choose. We *only* require that *it* be possible to derive V uniquely from th parameters* For «anpl«» we could chouse *to* optimize over a, e, f. A, and *k* ghen in Equation 6. seine*hat better approach is to repreiesi the ellipse in the form

$$\alpha (x-h)^2 + \beta (x-h)(y-k) + \gamma (y-k)^2 = 1$$

andi optimum ©i»er & jB, *y> h Mi k 'This* formulation avoids *iegmetxy in 0* (orientation) when the clips

H#'iV!:!^ji*! must start with some initiai guess as to the approximating ellipse. The easiest way to do t :s -v-t h:r*2 ^:i data points, preferably jt both ends and nor the middle, and calculate the circle that pas Ar f;i^€:^ £:?£ *p^rm. HifaUTtxi** tends to preserve the form -:f ihc inxrJ guess, ff the initial gui rcpttrfc^ti:i in c;]r^e ;he r;^xi *;^ a ;:^verge t> *n* hjperholic >}Iution. A rcughl> circular ellipse will r **be transformed to a drastically elongated one.**

The minimiz problem is fi^.c? I1i^^s^n^ Cure must be ^t2rc;sed to use the correct *numcrd* **technique, or the results will be poor.** 'Af'»:r.e ui^erj^nc^Otts, I::urr.sv:it tJ^tiiic xethxd_^f^l^pi **descent with accelerated convergen** \r^]cttf ve. !l>T;v lake:r.ar,> minutes:f:onputcr Unc f **the method to converge, if at all. Evaluation of the gradient of *S* does not appear** to help appreciably. T **only method that gives acceptable results requires evaluating the matrix of second partial derivatives of**

then finding the eigenvectors of that matrix. The complete method is given in Appendix C.

We shall not attempt to prove formally that results obtained from hill climbing on the expression given by Equation 16 are independent of position, orientation, and scale. Instead we shall appeal to an intuitive understanding of an error function and its gradient. The error function should not be affected by changes of coordinates, nor should its gradient. A change of scale will affect the error function and its gradient, but should multiply them by the same constant value everywhere. Hence, a local minimum will stay a local minimum under translation, rotation, and scaling. Depending on the particular hill-climbing method used, there *may* be some dependence of convergence properties on scaling and rotation.

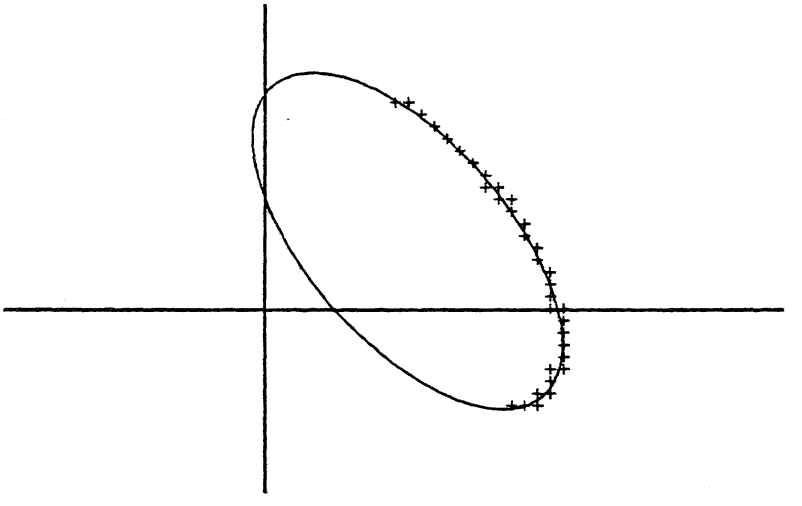

Figure 11: Hill-Climbing Curve Fit

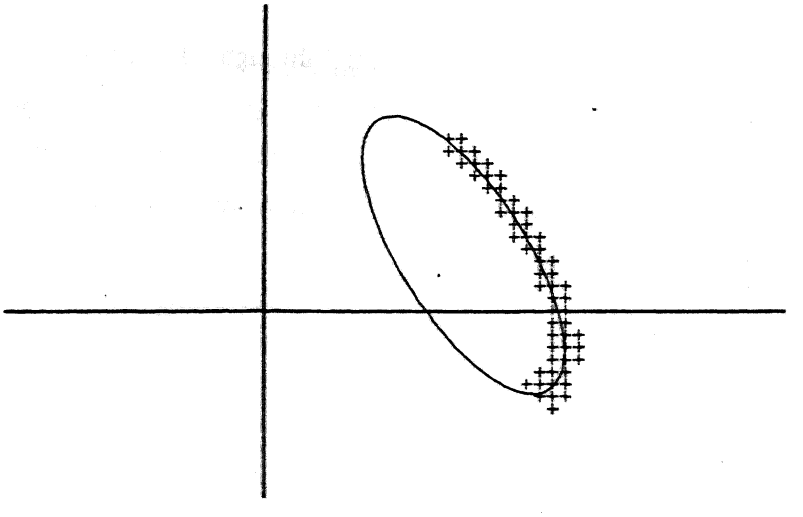

Figure 12: Hill-Climbing Curve Fit

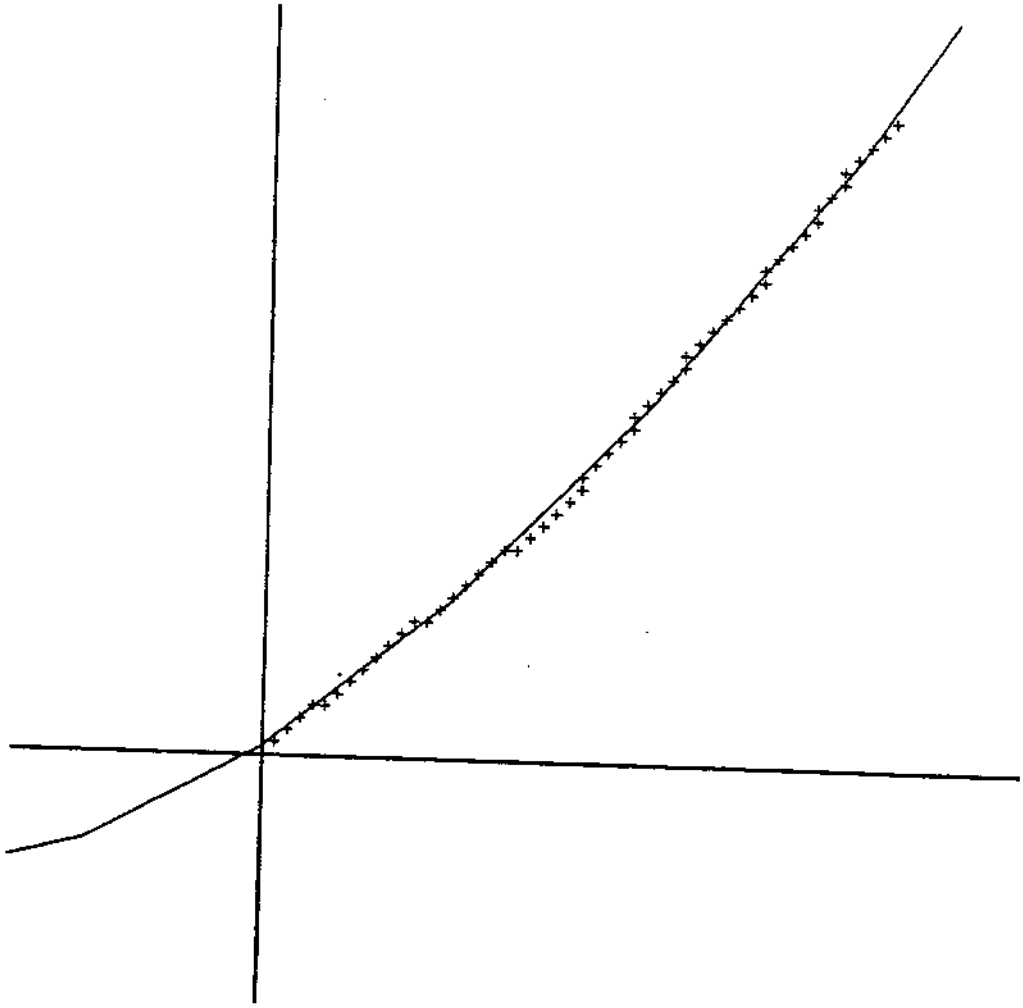

Figure 13: Hill-Climbing Curve Fit

Figures 11 through 13 show die date points of Figures 8 through 10 fitted by hill climbing with an initial . circular estimate. Figure 11 is approximately equivalent to Figure 8. Figure 12 shows a more noticeable improvement with *respect* to Fipre 9. While the result doesn't come near the ellipse from which the data points were generated (cf. Figure 7), the It at the tower end of die data points is more 'Intuitive/' *In the* case of Figure 13, the improvement is dramatic

Applying the Gradient Constraint to Straight Lines

The following section is a digression from the main topic of fitting second-order curves. A new formulation of straight-line fitting is obtained when we apply the methods developed here to the linear case.

A straight line is defined by the equation

$$G(x,y) = A\,x + B\,y + C = 0\,. \tag{18}$$

We define

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{ and } \mathbf{V} = \begin{bmatrix} A \\ B \\ C \end{bmatrix},$$

so that we may rewrite Equation 18 as

$$G(x,y) = \mathbf{V}^T \mathbf{X} = \mathbf{X}^T \mathbf{V} = 0\,.$$

We seek to minimize the error function

$$\Xi = \Sigma\, \xi_i^{\,2} = \Sigma\, G^2 = \mathbf{V}^T \mathbf{P}\, \mathbf{V}$$

where

$$\mathbf{P} = \Sigma\, \mathbf{X}\, \mathbf{X}^T = \begin{bmatrix} \Sigma x^2 & \Sigma xy & \Sigma x \\ \Sigma xy & \Sigma y^2 & \Sigma y \\ \Sigma x & \Sigma y & n \end{bmatrix}.$$

The magnitude of the gradient of G is constant for all x and y, and is equal to the square root of $A^2 + B^2$.

$$(\nabla G)^2 = \mathbf{V}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V} = \mathbf{V}^T\, \mathbf{Q}\, \mathbf{V}.$$

If the gradient is constrained to unity, then the error function $G(x,y)$ will be precisely equal to the perpendicular distance from (x,y) to the line $G = 0$.

Just as in the second-order case, the vector $\mathbf{V}$ that minimizes Ξ subject to the given constraint must be a solution to the eigenvalue equation

$$\mathbf{P}\, \mathbf{V} = \lambda\, \mathbf{Q}\, \mathbf{V}.$$

Some algebra yields the pair of solutions

$$V = \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \frac{1}{\sqrt{s^2 + (r-\lambda)^2}} \times \begin{bmatrix} -s \\ r - \lambda \\ (s\Sigma x - (r-\lambda)\Sigma y)/n \end{bmatrix} \tag{19}$$

$$= \frac{1}{\sqrt{(t-\lambda)^2 + s^2}} \times \begin{bmatrix} t - \lambda \\ -s \\ (-(t-\lambda)\Sigma x + s\Sigma y)/n \end{bmatrix} \tag{20}$$

where

$$r = \Sigma x^2 - (\Sigma x)^2 / n$$
$$s = \Sigma xy - \Sigma x \Sigma y / n$$
$$t = \Sigma y^2 - (\Sigma y)^2 / n$$

$$\lambda = \tfrac{1}{2}(r + t - \sqrt{(r-t)^2 + 4s^2}\ ).$$

The two forms are mathematically equivalent unless $s=0$, in which case one form or the other will division by zero. For this reason, Equation 19 is to be preferred whenever r is greater than t, and Eq when the reverse is true. Once A and B have been computed using either form, C may be easily con $-(A\Sigma x + B\Sigma y)/n$. The mean-square error of the fit is equal to λ/n.

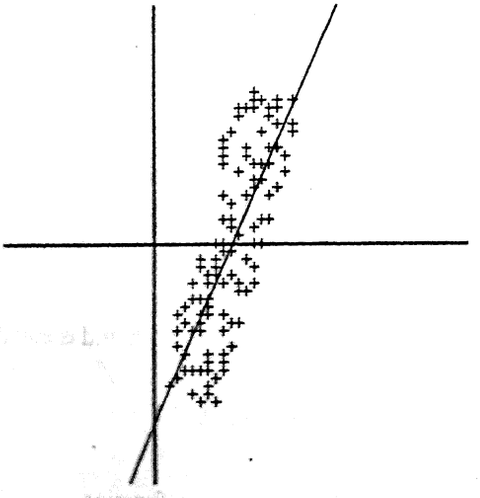

Figure 14: Straight Line Fit Minimizing Vertical Distances

Figures 14 and 15 show a startling comparison between the traditional method of fitting straight lin the method presented above. The data points show a wide scatter about a nearly-vertical line. The l Figure 14 was fit using the traditional linear regression formulas, where a line is represented by the equat

$$y = Mx + B$$

and M and B are calculated as

$$M = \frac{N\Sigma xy - \Sigma x \Sigma y}{N\Sigma x^2 - (\Sigma x)^2}$$

$$B = \frac{\Sigma x^2 \Sigma y - \Sigma x \Sigma xy}{N\Sigma x^2 - (\Sigma x)^2}.$$

The straight line of Figure 15 was based on the line representation of Equation 18 and the solutio Equation 19.

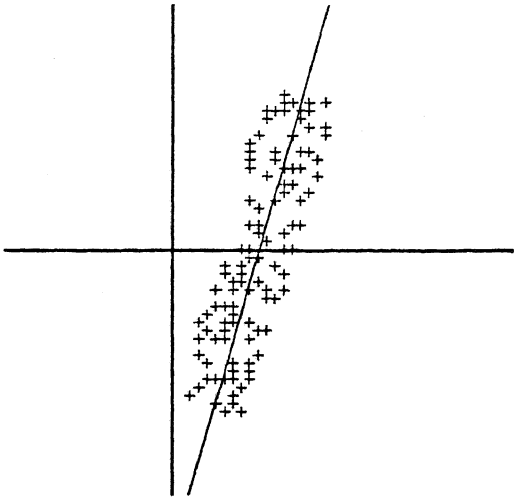

Figure 15: Straight Line Fit Minimizing Perpendicular Distances

A failure of a "tried and true" method deserves some analysis and discussion. In this case, the failure is traceable to the assumption that x is the independent variable, that y depends on x. But when the trend of the data is nearly vertical, it may be that x is more a function of y. A vertical line is degenerate using the regression formulas. If it makes sense for a collection of points on the plane to approximate a vertical line, then we should not use linear regression.

I have not seen this formulation published anywhere else. I would appreciate anyone who has seen this result published elsewhere letting me know.

Conclusions

Three methods for fitting second-order curves to sets of data points on the plane have been presented and analyzed. These methods are distinguished principally by the way they measure the amount of misfit between a given curve and a given set of points. The three measures are:

1. the quadratic form, with the constant term set equal to 1 (Equation 11),
2. the quadratic form (Equation 13) subject to the *average* gradient value being held to 1 (Equation 14),
3. the quadratic form divided by the gradient magnitude at each point (Equation 16).

As may be expected, the three measures lead to different results when minimized. The first measure has been shown to be sensitive to translation in the plane, and to give grossly incorrect results under certain conditions. The second measure has been formally shown to be insensitive to translation, rotation, and

scaling, and reasons have been given why the third measure ought to be the same. The third measure has been shown to give somewhat better results than the second, particularly in difficult cases with small angular arcs and widely scattered data points.

The three measures also lead to very different computational procedures for their minimization. Minimizing measures 1 and 2 both require summing products of x and y up to the 4th power; in this summation they are O(n), where n is the number of data points. But for fewer than 100 data points, the major use of computation time is in solution of the simultaneous linear equations (for measure 1), or the eigenvalue solution (for measure 2). On a Digital Equipment Corporation 2060 computer, generation of Figures such as 6 and 7 typically require about 50 milliseconds.

On the other hand, measure 3 is very expensive computationally. Computation time is a direct function not only of the number of data points, but also of the initial solution estimate and the accuracy required. Generation of Figures 11 and 12 required 24 and 42 seconds respectively. Hence hill climbing is to be recommended only when all other methods prove inadequate.

Appendix A: Solution of the Generalized Eigenvalue Equation

We wish to solve the generalized eigenvalue equation

$$PV = \lambda QV,$$

given that Q is singular and P may be close to singular. The following method was derived by Richard Underwood.

We know that the Last row and the last column of matrix Q are zero. Q may be represented by the partitioned matrix

$$Q = \left[\begin{array}{c|c} Q^* & 0 \\ \hline 0 & 0 \end{array}\right].$$

We may usually expect the 5x5 matrix Q* to be positive definite. We may use a Cholesky decomposition [Forsythe and Moler] to factor Q* into a lower diagonal matrix L* and its transpose L^{*T}, so that

$$Q^* = L^* L^{*T}$$

If we let L represent the augmented matrix

$$L = \left[\begin{array}{c|c} L^* & 0 \\ \hline 0 & 1 \end{array}\right]$$

then we have the result

$$L^{-1} Q L^{-T} = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & 0 \end{array}\right]$$

where I denotes the five-by-five unit matrix and L^{-T} is the transpose of L^{-1}.

The original generalized eigenvalue equation, Equation 15, may be transformed into

$$L^{-1} P L^{-T} L^T V = L^{-1} Q L^{-T} L^T \lambda V.$$

Applying the substitution

$$C = L^{-1} P L^{-T} \sim \left[\begin{array}{c|c} C^* & U \\ \hline V & T \end{array}\right]$$

and letting Y be the partitioned column vector

$$Y = L^{-T} V = \left[\begin{array}{c} Z \\ \hline W \end{array}\right]$$

yields the representation

$$\left[\begin{array}{c|c} C^* & U \\ \hline V & T \end{array}\right] \left[\begin{array}{c} Z \\ \hline W \end{array}\right] = \lambda \left[\begin{array}{c|c} I & 0 \\ \hline 0 & 0 \end{array}\right] \left[\begin{array}{c} Z \\ \hline W \end{array}\right]. \qquad (21)$$

The bottom row of this result represents the scalar equation

$$U^T Z + aW = O.$$

This may be solved to give

$$W = -\frac{U^T Z}{a}. \qquad (22)$$

The top five rows of Equation 21 represent the vector equation

$$C'Z + U W = \lambda Z$$

into which we may substitute our result for *W*, Equation 22, to yield

$$(\tilde{C} - \frac{1}{a} U U^T) Z = \lambda Z. \qquad (23)$$

Equation 23 may be solved by usual eigenvalue methods, such as the Q-R algorithm [Isaacson]. Given a particular solution Z_i, the corresponding V_1 is given by

$$V, = L\text{-}T \begin{bmatrix} Z_1 \\ \hline W \end{bmatrix}$$

where *W* is given by Equation 22.

Appendix B: Rotational, Translational, and Scaling Invariance

The rotational, translational, and scaling invariance of the method may be shown as follows: Let there be a coordinate system (i/,i?) related to (x j) by the transformations

$$u = ax + by + e \qquad (24)$$
$$v = dx + ey + f.$$

We may define a vector U analogous to X such that

$$U = \begin{bmatrix} 1? \\ uv \\ v^2 \\ u \\ v \\ 1 \end{bmatrix} = HX = \begin{bmatrix} a^2 & lab & b^1 & lac & 2bc & c^2 \\ ad & ae+be & be & af+cd & bf+ee & cf \\ £\bullet & Ide & e^2 & Idf & 2ef & f \\ 0 & 0 & 0 & a & b & c \\ 0 & 0 & 0 & d & e & f \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{bmatrix}.$$

Just as a vector V defines an error function $Gixj) = V^T X$, a vector V 'can define an error function $G^* = V'^T U$. Equating the two error functions (for all X) yields the relationship

$$V = H^T V'.$$

Now, suppose that for some collection of data points $\{X_i, f=1,.../i\}$ V minimizes the aggregate error given by Equation 13. We have shown that V satisfies Equation 15, and that the corresponding eigenvalue X 5s the smallest of all eigenvalues that satisfy the equation.

Rewriting Equation 15 and performing some **algebra gives**

$$P V = \lambda Q V$$

$$\Sigma (X X^T) V = \lambda \Sigma (X_x X_x^T + X_y X_y^T) V$$

$$\Sigma (X X^T H^T V') = \lambda \Sigma (X_x X_x^T H^T V' + X_y X_y^T H^T V')$$

$$\Sigma (H X X^T H^T V') = \lambda \Sigma (H X_x X_x^T H^T V' + H X_y X_y^T H^T V')$$

$$\Sigma (U U^T V') = \lambda \Sigma (U_x U_x^T V' + U_y U_y^T V')$$

where U_x and U_y denote the partial derivatives of U with respect to x and y, respectively. If the transformation of Equation 24 is an *orthonormal* transformation, that is, if

$$a^2 + b^2 = c^2 + d^2$$

and

$$ad + be = 0,$$

then it may be shown that

$$U_x U_x^T + U_y U_y^T = (a^2 + b^2)(U_u U_u^T + U_v U_v^T).$$

Substituting this result in the above,

$$\Sigma (U U^T) V' = (a^2 + b^2) \lambda \Sigma (U_u U_u^T + U_v U_v^T) V'$$

yields the result we seek: any solution to Equation 15 in one coordinate system is also a solution in any orthonormally related coordinate system. Since the eigenvalues are proportional, the *smallest* eigenvalues in the two coordinate systems correspond.

Appendix C: Hill Climbing Method

Hill climbing refers to a class of numerical methods that minimize a function $G(U)$, where U may be a n-dimensional vector. For our purposes, we may assume the existence of a subroutine MIN1 that minimizes G along a straight line. It accepts an initial estimate U_0 and an increment ΔU, finds a value of k that locally minimizes $G(U_0 + k\,\Delta U)$, and updates U_0 to the new minimizing value. Different hill-climbing strategies consist of different means of selecting a sequence of ΔU vectors. The sequence terminates when no further improvement in G can be obtained.

A method that requires no knowledge about the function G is to search sequentially along the n dimensions of U, i.e., to apply the sequence

$$\Delta U = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \cdots, \quad \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \cdots$$

This is one form of the *method of steepest descent*. For some functions this method will suffice. But if the function G is ill-conditioned, that is if the elements of U interact to a great degree in their influence on $G(U)$,

or if the equipotentials of G tend to form squashed ellipsoids, then this simple approach will converge very slowly. Figure 16 shows a hypothetical sequence of iterations in minimizing a function of two variables.

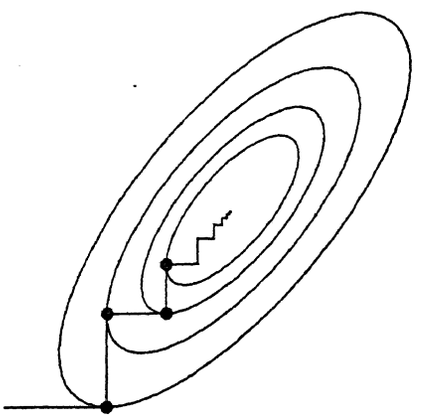

Figure 16: Iterations in Method of Steepest Descent

Convergence can be enhanced by keeping track of the cumulative change in U as the minimization proceeds. After n minimizations along the n coordinate directions of U, an additional minimization step can be attempted along the direction indicated by the sum of the individual $k_i \Delta U_i$ terms measured in the preceding n calls to MIN1. This is called the *method of steepest descent with accelerated convergence.*

Some improvement in performance can be obtained if it is possible to evaluate the gradient of G, that is, the n partial derivatives of G with respect to the elements of U. At each minimization step let ΔU point in the direction of the gradient. Use of the gradient can give a computational advantage in reducing the number of calls to MIN1, but it is doubtful whether this technique affects overall convergence properties.

The situation illustrated in Figure 16 can be completely avoided if the second partials of G are available. In the neighborhood of U_0, $G(U)$ may be approximated by the expression

$$G(U) \approx G_0 + D^T (U - U_0) + (U - U_0)^T P (U - U_0) \qquad (25)$$

where $D = D(U_0)$ is the gradient vector, or vector of first partial derivatives, and $P = P(U_0)$ is the matrix of second partial derivatives. P is a symmetric matrix. An eigenvalue analysis of P will give n linearly independent eigenvectors $\{U_i, i=1,...,n\}$ and associated eigenvalues $\{\lambda_i, i=1,...,n\}$ such that

$$P U_i = \lambda_i U_i .$$

These eigenvectors point in the directions of the principal axes of the equipotential ellipsoids of G. Function minimization may take place in the eigenvector directions independently without cross-coupling or co-

variance effects. Convergence will be quite rapid.

If the eigenvectors are normalized to unit magnitude, and we let $\mathbf{U} = \mathbf{U}_0 + k\,\mathbf{U}_i$, then Equation 25 becomes

$$G(\mathbf{U}) \approx G_0 + k\,\mathbf{D}^T\mathbf{U}_i + \lambda_i k^2 . \tag{26}$$

Taking the derivative with respect to k and setting the result equal to zero, we find that the minimum ought to occur when $k = \mathbf{D}^T\mathbf{U}_i / (2\,k\,\lambda_i)$. If λ is negative, as it frequently turns out to be, then the k above actually points to a relative *maximum.* This result can be used to guide the minimization by subroutine MIN1, to suggest initial step size for the search, but experience shows that the use of MIN1 should not be bypassed.

For the case at hand, ellipses are represented by

$$\alpha(x-h)^2 + \beta(x-h)(y-k) + \gamma(y-k)^2 = 1,$$

or

$$\alpha x^2 + \beta xy + \gamma y^2 - (2\alpha h+\beta k)x - (\beta h+2\gamma k)y + \alpha h^2 + \beta hk + \gamma k^2 - 1 = 0.$$

Therefore let

$$\mathbf{U} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ h \\ k \end{bmatrix} \quad \text{and} \quad \mathbf{V}(\mathbf{U}) = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ -2\alpha h-\beta k \\ -\beta h-2\gamma k \\ \alpha h^2+\beta hk+\gamma k^2-1 \end{bmatrix},$$

and let $\mathbf{X}$, $\mathbf{X}_x$, and $\mathbf{X}_y$ be defined as before. $G(\mathbf{U})$ is given by

$$G(\mathbf{U}) = \Xi = \Sigma\,\xi_i^2 = \Sigma\frac{N}{D}$$

where

$$N = \mathbf{V}^T\mathbf{X}\,\mathbf{X}^T\mathbf{V} = (\mathbf{X}^T\mathbf{V})^2$$
$$D = \mathbf{V}^T(\mathbf{X}_x\mathbf{X}_x^T + \mathbf{X}_y\mathbf{X}_y^T)\mathbf{V} = (\mathbf{X}_x^T\mathbf{V})^2 + (\mathbf{X}_y^T\mathbf{V})^2 .$$

The first and second partial derivatives of N and D with respect to the elements of $\mathbf{U}$ are:

$$\nabla\mathbf{N} = 2\begin{bmatrix} Fx'^2 \\ Fx'y' \\ Fy'^2 \\ -FF_x \\ -FF_y \end{bmatrix}$$

$$V^2N = 2\begin{bmatrix} x'^4 & x'^3y' & x'^2y'^2 & -F_xx'^2-2Fx' & -x'^2 \\ x'^3y' & x'^2y'^2 & x'y'^3 & -F_yx'y-FFx' & -F_yx'y-FFx' \\ \cdot\ ^2 & x'y'^3 & y'^4 & \cdot & -F_yy'^2-2Fy \\ -Fx'^2-2Fx' & -Fx'y'-Fy' & -F_yy'^2 & F_x^2+2Fa & \\ V & -F_yx'/-Fx' & -F_yy'^2-2Fy' & {}_y+F\beta & F_y^2+2F\gamma \end{bmatrix}$$

$$VD = 2\begin{bmatrix} 2F/ \\ F_xy'+F_yx' \\ 2F \\ -2F/-FJ \\ L-Fj-lF_y \end{bmatrix}$$

$$V^2D = 2\begin{bmatrix} 4x'^2 & 4x'y' & 0 & -4x'\alpha-2F_x & -2x'\beta \\ 2JC>' & y'^2+x'^2 & 2x'y' & -2y'\alpha-x'\beta-F_y & -y'\beta-F\ \ x'\gamma \\ 0 & 2x'y' & 4y'^2 & -2y'\beta & -4/Y-2F_y \\ -4x'a-2F_x & -2y'\alpha-x'\beta-2F_y & -2y'\beta & 4\alpha^2+2\beta^2 & 2\alpha\beta+2\beta\gamma \\ -2x'\beta & -y'\beta-F_x-2x'\gamma & -4y_Y-2F_y & 2\alpha\beta+2\beta\gamma & 2\beta^2+4\gamma^2 \end{bmatrix},$$

where

$$x' = x - h$$
$$y' = y - k$$
$$F = \alpha x'^2 + \beta x'y' + \gamma y'^2 - 1$$
$$F_x = 2\alpha x' + \beta y'$$
$$F_y = \beta x' + 2\gamma y'.$$

The first and second partials of $\mathfrak{L}_1$ can be derived from the partials of N and D by use of the formulas

$$\frac{3}{3j?\ 2)}\frac{N}{} = \frac{DN_p - ND_p}{D^1}$$

$$\frac{d^1N}{dpdq\,D} = \frac{2ND_pD_q - N_pDD_q - N_qDD_p + N_{pq}D^2 + NDD_{pq}}{if}$$

where p and q stand for any of the set {a, /}, 7, A, and &}, and subscripting denotes taking the partial derivative. The derivative of a sum is equal to the sum of the derivatives of the terms. Hence, the partial derivatives of 2 are the sums of the partial derivatives of the individual $|_t$ terms.

References

[Agin 72] Gerald Jacob Agin, "Representation and Description of Curved Objects," Memo AIM-173, Stanford Artificial Intelligence Project, Computer Science Department, Stanford University, Stanford, California, October 1972.

[Agin 76] Gerald J. Agin and Thomas O. Binford, "Computer Description of Curved Objects," *IEEE Transactions on Computers*, Vol. C-25, No. 4, April 1976, pp. 439-449.

[Baudelaire] Patrick C. Baudelaire, "DRAW", from *Alto User's Handbook*, Xerox Palo Alto Research Center, Palo Alto, California, September, 1979, pp. 98-128.

[Bolles] Robert C. Bolles and martin A. Fischler, "A RANSAC-Based Approach to Model Fitting and its Application to Finding Cylindrs in Range Data," Seventh International Joint Conference on Artificial Intelligence, Vancouver, August, 1981.

[Courant] R. Courant, *Differential and Integral Calculus*, Interscience, 1936, Volume 2, pp. 190-199.

[Coxeter] H. S. M. Coxeter, *Introduction to Geometry*, John Wiley and Sons, Inc., New York, 1961.

[Forsythe and Moler] George E. Forsythe and Cleve B. Moler, *Computer Solution of Linear Algebraic Equations*, Prentice-Hall, Englewood Cliffs, N.J., pp. 114-119.

[Isaacson] Eugene Isaacson and Herbert Bishop Keller, *Analysis of Numerical Methods*, John Wiley and Sons, 1966, pages 173-174.

[Kender] John Kender, "Shape from Texture," Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, 1980.

[Popplestone] R. J. Popplestone, C. M. Brown, A. P. Ambler, and G. F. Crawford, "Forming Models of Plane-and-Cylinder Faceted Bodies from Light Stripes," Fourth International Joint Conference on Artificial Intelligence, Tbilisi, Georgia, USSR, August 1975, pp.664-668.

[Purcell] Edwin J. Purcell, *Analytic Geometry*, Appleton-Century-Crofts, Inc., New York, 1958.

[Rivlin] Theodore J. Rivlin, *An Introduction to the Approximation of Functions*, Blaisdell Publishing Company, Waltham, Mass, 1969.

[Shirai] Yoshiaki Shirai and Motoi Suwa, "Recognition of Polyhedrons with a Range Finder," Second International Joint Conference on Artificial Intelligence, London, September 1971.

[Smith] Lyle B. Smith, "The Use of Man-Machine Interaction in Data-Fitting Problems," SLAC Report No. 96, Stanford Linear Accelerator Center, Stanford, California, March 1969.

[Tsuji] Saburo Tsuji and Fumio Matsumoto, "Detecting Elliptic and Linear Edges by Searching Two Parameter Spaces," Fifth International Joint Conference on Artificial Intelligence, Cambridge, Mass., August 1977, pp. 700-705.